©Eric C. Anderson
(subbin' for E.A Thompson)

Monte Carlo Methods and Importance Sampling

History and definition: The term "Monte Carlo" was apparently first used by Ulam and von Neumann as a Los Alamos code word for the stochastic simulations they applied to building better atomic bombs. Their methods, involving the laws of chance, were aptly named after the international gaming destination; the moniker stuck and soon after the War a wide range of sticky problems yielded to the new techniques. Despite the widespread use of the methods, and numerous descriptions of them in articles and monographs, it is virtually impossible to find a succint definition of "Monte Carlo method" in the literature. Perhaps this is owing to the intuitive nature of the topic which spawns many definitions by way of specific examples. Some authors prefer to use the term "stochastic simulation" for almost everything, reserving "Monte Carlo" only for Monte Carlo Integration and Monte Carlo Tests (*cf.* Ripley 1987). Others seem less concerned about blurring the distinction between simulation studies and Monte Carlo methods.

Be that as it may, a suitable definition can be good to have, if for nothing other than to avoid the awkwardness of trying to define the Monte Carlo method by appealing to a whole bevy of examples of it. Since I am (so Elizabeth claims!) unduly influenced by my advisor's ways of thinking, I like to define Monte Carlo in the spirit of definitions she has used before. In particular, I use:

> Definition: Monte Carlo is the art of approximating an expectation by the sample mean of a function of simulated random variables.

We will find that this definition is broad enough to cover everything that has been called Monte Carlo, and yet makes clear its essence in very familiar terms: Monte Carlo is about invoking laws of large numbers to approximate expectations.[1]

While most Monte Carlo simulations are done by computer today, there were many applications of Monte Carlo methods using coin-flipping, card-drawing, or needle-tossing (rather than computer-generated pseudo-random numbers) as early as the turn of the century—long before the name Monte Carlo arose.

In more mathematical terms: Consider a (possibly multidimensional) random variable X having probability mass function or probability density function $f_X(x)$ which is greater than zero on a set of values $\mathcal{X}$. Then the expected value of a function g of X is

$$\mathbb{E}(g(X)) = \sum_{x \in \mathcal{X}} g(x) f_X(x) \tag{1}$$

if X is discrete, and

$$\mathbb{E}(g(X)) = \int_{x \in \mathcal{X}} g(x) f_X(x) dx \tag{2}$$

if X is continuous. Now, if we were to take an n-sample of X's, $(x_1, \ldots, x_n)$, and we computed the mean of $g(x)$ over the sample, then we would have the Monte Carlo *estimate*

$$\widetilde{g_n}(x) = \frac{1}{n} \sum_{i=1}^{n} g(x_i)$$

[1] This applies when the simulated variables are independent of one another, and might apply when they are correlated with one another (for example if they are states visited by an ergodic Markov chain). For now we will just deal with independent simulated random variables, but all of this extends to samples from Markov chains via the weak law of large numbers for the number of passages through a recurrent state in an ergodic Markov chain (see Feller 1957). You will encounter this later when talking about MCMC.

of $\mathbb{E}(g(X))$. We could, alternatively, speak of the random variable

$$\widetilde{g_n}(X) = \frac{1}{n}\sum_{i=1}^{n} g(X)$$

which we call the Monte Carlo *estimator* of $\mathbb{E}(g(X))$.

If $\mathbb{E}(g(X))$, exists, then the weak law of large numbers tells us that for any arbitrarily small ϵ

$$\lim_{n\to\infty} P\left(|\widetilde{g_n}(X) - \mathbb{E}(g(X))| \geq \epsilon\right) = 0.$$

This tells us that as n gets large, then there is small probability that $\widetilde{g_n}(X)$ deviates much from $\mathbb{E}(g(X))$. For our purposes, the strong law of large numbers says much the same thing—the important part being that so long as n is large enough, $\widetilde{g_n}(x)$ arising from a Monte Carlo experiment shall be close to $\mathbb{E}(g(X))$, as desired.

One other thing to note at this point is that $\widetilde{g_n}(X)$ is unbiased for $\mathbb{E}(g(X))$:

$$\mathbb{E}(\widetilde{g_n}(X)) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} g(X_i)\right) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(g(X_i)) = \mathbb{E}(g(X)).$$

Making this useful: The preceding section comes to life and becomes useful when one realizes that very many quantities of interest may be cast as expectations. Most importantly for applications in statistical genetics, it is possible to express all probabilities, integrals, and summations as expectations:

Probabilities: Let Y be a random variable. The probability that Y takes on some value in a set A can be expressed as an expectation using the indicator function:

$$P(Y \in A) = \mathbb{E}(I_{\{A\}}(Y)) \tag{3}$$

where $I_{\{A\}}(Y)$ is the indicator function that takes the value 1 when $Y \in A$ and 0 when $Y \notin A$.

Integrals: Consider a problem now which is completely deterministic—integrating a function $q(x)$ from a to b (as in high-school calculus). So we have $\int_a^b q(x)dx$. This can be expressed as an expectation with respect to a uniformly distributed, continuous random variable U between a and b. U has density function $f_U(u) = 1/(b-a)$, so if we rewrite the integral we get

$$(b-a)\int_a^b q(x)\frac{1}{b-a}dx = (b-a)\int_a^b q(x)f_U(x)dx = (b-a)\mathbb{E}(q(U)) \ldots \text{voila!}$$

Discrete Sums: The discrete version of the above is just the sum of a function $q(x)$ over the countably many values of x in a set A. If we have a random variable W which takes values in A all with equal probability p (so that $\sum_{w\in A} p = 1$ then the sum may be cast as the expectation

$$\sum_{x\in A} q(x) = \frac{1}{p}\sum_{x\in A} q(x)p = \frac{1}{p}\mathbb{E}(q(W)).$$

The immediate consequence of this is that all probabilities, integrals, and summations can be approximated by the Monte Carlo method. A crucial thing to note, however, is that there is no restriction that says U or W above must have uniform distributions. This is just for easy illustration of the points above. We will explore this point more while considering importance sampling.

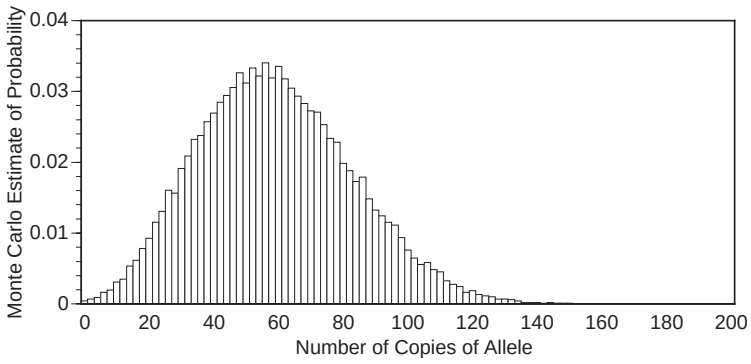
Example I: Approximating probabilities by Monte Carlo. Consider a Wright-Fisher population of size N diploid individuals in which X_t counts the numbers of copies of a certain allelic type in the population at time t. At time zero, there are x_0 copies of the allele. Given this, what is the probability that the allele will be lost from the population in t generations, *i.e.*, $P(X_t = 0|X_0 = x_0)$? This can be computed exactly by multiplying transition probability matrices together, or by employing the Baum (1972) algorithm (which you will learn about later), but it can also be approximated by Monte Carlo. It is simple to simulate genetic drift in a Wright-Fisher population; thus we can easily simulate values for X_t given $X_0 = x_0$. Then,

$$P(X_t = 0|X_0 = x_0) = \mathbb{E}(I_{\{0\}}(X_t)|X_0 = x_0)$$

where $I_{\{0\}}(X_t)$ takes the value 1 when $X_t = 0$ and 0 otherwise. Denoting the i^{th} simulated value of X_t by $x_t^{(i)}$ our Monte Carlo estimate would be

$$\widetilde{P}(X_t = 0|X_0 = x_0) \approx \frac{1}{n}\sum_{i=1}^{n} I_{\{0\}}(x_t^{(i)}).$$

Example II: Monte Carlo approximations to distributions. A simple extension of the above example is to approximate the whole probability distribution $P(X_t|X_0 = x_0)$ by Monte Carlo. Consider the histogram below:



It represents the results of simulations in which $n = 50,000$, $x_0 = 60$, $t = 14$, the Wright-Fisher population size $N = 100$ diploids, and each rectangle represents a Monte Carlo approximation to $P(a \leq X_{14} < a + 2|X_0 = 60)$, $a = 0, 2, 4, \ldots, 200$. For each such probability, the approximation follows from

$$P(a \leq X_{14} < a+2|X_0 = 60) = \mathbb{E}(I_{\{a\leq X<a+2\}}(X_{14})) \approx \frac{1}{n}\sum_{i=1}^{n} I_{\{a\leq X<a+2\}}(x_{14}^{(i)}),\ a = 0, 2, \ldots, 200$$

Example III: A discrete sum over latent variables. In many applications in statistical genetics, the probability $P(Y)$ of an observed event Y must be computed as the sum over very many latent variables X of the joint probability $P(Y, X)$. In such a case, Y is typically fixed, *i.e.*, we have observed $Y = y$ so we are interested in $P(Y = y)$, but we can't observe the values of the latent variables which may take values in the space $\mathcal{X}$. Though it follows from the laws of probability that

$$P(Y = y) = \sum_{x\in\mathcal{X}} P(Y = y, X = x),$$

quite often $\mathcal{X}$ is such a large space (contains so many elements) that it is impossible to compute the sum. Application of the law of conditional probability, however, gives

$$P(Y = y) = \sum_{x\in\mathcal{X}} P(Y = y, X = x) = \sum_{x\in\mathcal{X}} P(Y = y|X = x)P(X = x). \tag{4}$$

The term following the last equals sign is the sum over all x of a function of x [namely, $P(Y = y|X = x)$], weighted by the marginal probabilities $P(X = x)$. Clearly this is an expectation, and therefore may be approximated by Monte Carlo, giving us

$$P(Y = y) \approx \frac{1}{n}\sum_{i=1}^{n} P(Y = y|X = x_i)$$

where x_i is the i^{th} realization from the marginal distribution of X. You will see this sort of thing many times again in Stat 578C.

OK, these examples have all been presented as if the application of Monte Carlo to practically any problem is a soporific and trivial exercise. However, nothing could be further from the truth! Though it is typically easy to formulate a quantity as an expectation and to propose a "naive" Monte Carlo estimator, it is quite another thing to actually have the Monte Carlo estimator provide you with good estimates in a reasonable amount of computer time. For most problems, a number of Monte Carlo estimators may be proposed, however some Monte Carlo estimators are clearly better than others. Typically, a "better" Monte Carlo estimator has smaller variance (for the same amount of computational effort) than its competitors. Thus we turn to matters of...

Monte Carlo variance: Going back to our original notation, we have the random variable $\widetilde{g_n}(X)$, a Monte Carlo estimator of $\mathbb{E}(g(X))$. Like all random variables, we may compute its variance (if it exists) by the standard formulas:

$$\text{Var}(\widetilde{g_n}(X)) = \text{Var}\left(\frac{1}{n}\sum_{i=1}^{n} g(X_i)\right) = \frac{\text{Var}(g(X))}{n} = \frac{1}{n}\sum_{x\in\mathcal{X}}[g(x) - \mathbb{E}(g(X))]^2 f_X(x) \tag{5}$$

if X is discrete, and

$$\text{Var}(\widetilde{g_n}(X)) = \text{Var}\left(\frac{1}{n}\sum_{i=1}^{n} g(X_i)\right) = \frac{\text{Var}(g(X))}{n} = \frac{1}{n}\int_{x\in\mathcal{X}}[g(x) - \mathbb{E}(g(X))]^2 f_X(x)dx \tag{6}$$

if X is continuous. From here on out, let us do everything in terms of integrals over continuous variables, but it all applies equally well to sums over discrete random variables. There are numerous ways to reduce the variance of Monte Carlo estimators. Of these "variance-reduction" techniques, the one called "importance sampling" is particularly useful. I find that it is best introduced by describing its antithesis which I call "irrelevance sampling" or "barely relevant sampling," which we will turn to after a short digression.

Digression 1: Estimating $\text{Var}(\widetilde{g_n}(X))$*:* Since, typically $\mathbb{E}(\widetilde{g_n}(X))$ in (5) or (6) is unknown and the sum or the integral is not feasibly computed (that is why we would be doing Monte Carlo in the first place) the formulas in (5) and (6) are not useful for estimating the variance associated with your Monte Carlo estimate when you are actually doing Monte Carlo. Instead, just like approximating the variance from a sample à la our earliest statistics classes, we have an unbiased estimator for $\text{Var}(g(X))$:

$$\widehat{\text{Var}}(g(X)) = \frac{1}{n-1}\sum_{i=1}^{n}(g(x_i) - \widetilde{g_n}(x))^2 \tag{7}$$

(This is just the familiar s^2 from Statistics 1). The unbiased estimate of the variance of $\widetilde{g_n}(X)$ is $1/n$ of that:

$$\widehat{\text{Var}}(\widetilde{g_n}(X)) = \frac{\widehat{\text{Var}}(g(X))}{n} = \frac{1}{n(n-1)}\sum_{i=1}^{n}(g(x_i) - \widetilde{g_n}(x))^2. \tag{8}$$

The form given in (8) is not particularly satisfying if one does not want to wait until the end of the simulation (until n is reached) to compute the variance. To this end, the following formulas are extremely useful: The mean can be computed on the fly, recursively by:

$$\widetilde{g_{n+1}}(x) = \frac{1}{n+1}(n\widetilde{g_n}(x) + g(x_{n+1})). \tag{9}$$

If we also expend the effort to record the sum of the squares of the $g(x)$'s, $SS_{g_n} = \sum_{i=1}^{n}[g(x_i)]^2$, then a simple calculation gives $\widehat{\mathrm{Var}}(\widetilde{g_n}(X))$:

$$\widehat{\mathrm{Var}}(\widetilde{g_n}(X)) = \frac{1}{n}\left(\frac{1}{n-1}SS_{g_n} - \frac{n}{n-1}(\widetilde{g_n}(x))^2\right). \tag{10}$$

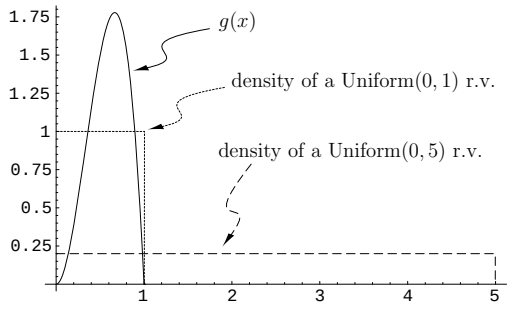
When programming in a language like C, using zero-base subscripting, I often confuse myself trying to implement the above recursion and formulas. So, primarily for my own benefit, (though this may be a useful reference for you if you program in C or C^{++}) I include a code snippet for implementing the above:

```
// variable declarations
double i;  // the index for counting number of reps (declaring as double
           // avoids having to cast it to a double all the time)
double gx;  // the values that will be realized
double mean_gx;    // the Monte Carlo average that we will accumulate (our Monte Carlo estimate)
double ss_gx;  // the sum of squares of gx
double var_of_mean_gx;  // the estimate of our monte carlo variance

// variable initializations:
mean_gx = 0.0;  ss_gx = 0.0;

// loop over the index i.  these are repetitions in the simulation:
for(i=0.0;i<n;i+=1.0)  {
     gx = "the value for gx realized on this repetion";
     mean_gx += (gx - mean_gx)/(i+1.0);  // the current Monte Carlo estimate
     ss_gx += gx * gx;  // the current sum of squares
     if(i>0) {
          var_of_mean_gx = (ss_gx - (i+1.0)*(mean_gx * mean_gx) ) / (i * (i+1.0) );
          // above is the current estimate of the variance of our Monte Carlo estimator
     }
}
```

Barely relevant sampling: Back to the task at hand. To introduce importance sampling we consider its opposite. Imagine that we want a Monte Carlo approximation to $\int_0^1 g(x)dx$ for $g(x)$ shown in the figure below. Note that $g(x) = 0$ for $x < 0$ and $x > 1$.



If we have $U \sim \mathrm{Uniform}(0,1)$, then we can cast the integral as the expectation with respect to U: $\int_0^1 g(x)dx = \mathbb{E}(g(U))$, so we may approximate it by Monte Carlo: $\frac{1}{n}\sum_1^n g(u_i)$. This would work reasonably well.

The figure, however, suggests another possibility. One could use $W \sim \mathrm{Uniform}(0,5)$ giving $\int_0^1 g(x)dx = 5 \cdot \mathbb{E}(g(W))$ and hence the Monte Carlo estimator $\frac{5}{n}\sum_1^n g(w_i)$. Obviously such a

course would make no sense at all because, on average, 80% of the realized w_i's would tell you nothing substantial about the integral of $g(x)$ since $g(x) = 0$ for $1 < x < 5$. This would be "barely relevant sampling," and no one in their right mind would willingly do it. It does make clear that one's choice of distribution from which to draw their random variables will affect the quality of their Monte Carlo estimator.

Importance sampling: Importance sampling is choosing a good distribution from which to simulate one's random variables. It involves multiplying the integrand by 1 (usually dressed up in a "tricky fashion") to yield an expectation of a quantity that varies less than the original integrand over the region of integration. For example, let $h(x)$ be a density for the random variable X which takes values only in A so that $\int_{x\in A} h(x)dx = 1$. Then $\frac{h(x)}{h(x)} = 1$ and

$$\int_{x\in A} g(x)dx = \int_{x\in A} g(x)\frac{h(x)}{h(x)}dx = \int_{x\in A} \frac{g(x)}{h(x)}h(x)dx = \mathbb{E}_h\left(\frac{g(X)}{h(X)}\right), \tag{11}$$

so long as $h(x) \neq 0$ for any $x \in A$ for which $g(x) \neq 0$, and where $\mathbb{E}_h$ denotes the expectation with respect to the density h. This gives a Monte Carlo estimator:

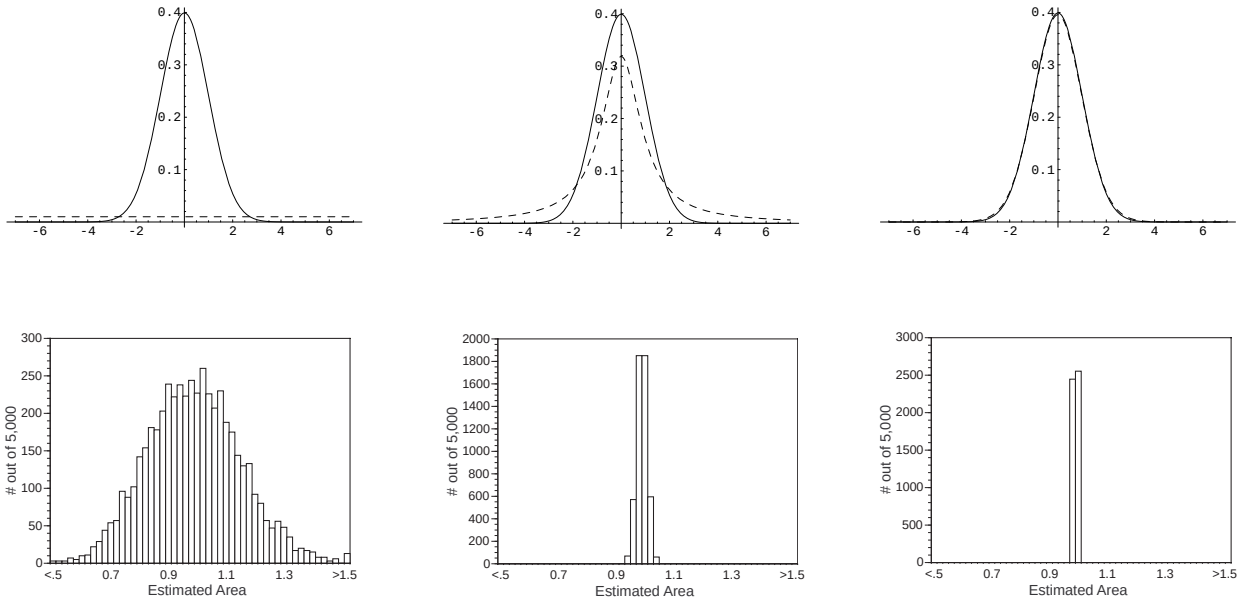
$$\widetilde{g_n^h}(X) = \frac{1}{n}\sum_{i=1}^{n} \frac{g(X_i)}{h(X_i)} \quad \text{where} \quad X_i \sim h(x). \tag{12}$$

Using (6) and the Cauchy-Schwarz inequality, it can be shown that $\mathrm{Var}(\widetilde{g_n^h}(X))$ is minimized when $h(x) \propto |g(x)|$ (see Rubinstein 1981, p. 123). If we restrict our attention to what for most of our purposes is the truly relevant case,[2] that is, $g(x) \geq 0\ \forall x \in A$, then it is immediately apparent that the choice of the density $h(x)$ which minimizes Monte Carlo variance is proportional to $g(x)$, *i.e.*, if $\alpha h(x) = g(x)$ where α is some constant of proportionality, then clearly we have $g(x)/h(x) = \alpha\ \forall\{x : h(x) > 0\}$ so $\mathbb{E}(g(X)/h(X)) = \alpha$ and hence the Monte Carlo variance would be zero by (6).

Wonderful! All we need to do to have a Monte Carlo estimator with zero variance is use (12) and make sure that our density h is proportional to the function g. The absurdity of this wishful thinking is that the ability to simulate *independent* random variables from $h(x)$, or the ability to compute the density $h(x)$, itself, implies that the normalizing constant of the distribution is computable, which in turn would imply that the original integral involving $g(x)$ is computable and there would hence be no reason to do Monte Carlo at all! Ultimately, however, it makes clear that a good importance sampling function (as h is called) will be one that is as close as possible to being proportional to $g(x)$—a point made by the following contrived example.

> Example V: A contrived demonstration. Let us approximate by Monte Carlo the area under a Normal$(0, 1)$ density curve from -50 to 50. This quantity will, of course, be *extremely* close to 1 (and we may as well call it 1). This is contrived because no one would ever do this in practice... Nonetheless, we will use a series of importance sampling functions: (a) a Uniform$(-50, 50)$ density, (b) a cauchy density (a t random variable on 1 df) truncated at -50 and 50, and (c) a truncated t random variable on 30 df. Figure 1 on Page 7 shows each importance sampling function as a dashed curve next to the normal curve. Below each of these is the histogram of 5,000 Monte Carlo estimates (using $n = 1,000$) of the area under the normal curve. As is clear from the progression from (a) to (c), the Monte Carlo estimates are less variable when the importance sampling function is closer to the shape of the normal density. (Note: the important feature is that the *shape* of the curves is closer. Obviously $g(x)$ and $h(x)$ will, in general, not be similar in height.)

[2] This is typically the relevant case because we are interested in non-negative quantities like probabilities.



(a) Uniform (b) t_1 Dist. (c) t_30 Dist.

Figure 1: Three different importance sampling functions (dotted lines) used to integrate the standard normal density (solid line) from -50 to 50. Top panels are the density curves and bottom panels are histograms of 5,000 Monte Carlo estimates of the area (which is exactly 1) using $n = 1,000$.

In summary, a good importance sampling function $h(x)$ should have the following properties:

1. $h(x) > 0$ whenever $g(x) \neq 0$
2. $h(x)$ should be close to being proportional to $|g(x)|$
3. it should be easy to simulate values from $h(x)$
4. it should be easy to compute the density $h(x)$ for any value x that you might realize.

Fulfilling this wish-list in high dimensional space (where Monte Carlo techniques are most useful) is quite often a tall task, and can provide hours of entertainment, not to mention dissertation chapters, *etc.*

Note also that $g(x)$ is any arbitrary function, so it certainly includes the integrand of a standard expectation. For example, with $X \sim f_X$ we might be interested in $\mathbb{E}(r(X))$ for some function r so we could use

$$\mathbb{E}(r(X)) = \int r(x) f_X(x) dx = \int \frac{r(x) f_X(x)}{h(x)} h(x) = \mathbb{E}_h \left(\frac{r(x) f_X(x)}{h(x)} \right)$$

and then go searching about for a suitable $h(x)$ that is close to proportional to $r(x) f_X(x)$.

Example VI: Latent variables and importance sampling Going back to Example III with the discrete sum over latent variables X it is clear that the optimal importance sampling function would be the conditional distribution of X given Y, *i.e.*,

$$P(Y = y) = \sum_{x \in \mathcal{X}} P(Y = y, X = x) = \sum_{x \in \mathcal{X}} \frac{P(Y = y, X = x)}{P(X|Y = y)} P(X|Y = y).$$

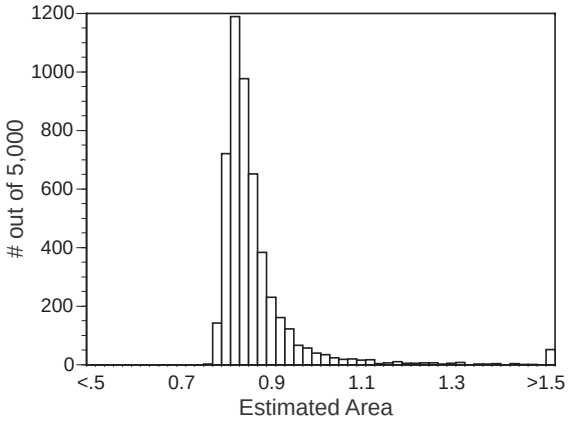


Figure 2: Histogram of 5,000 Monte Carlo estimates of the area under a truncated t distribution with one df using the standard normal density as the importance sampling function. The true area is 1. Note the several very high values (> 1.5).

> Note that the right side is a conditional expectation of a function of X. As before $P(X|Y)$ is not computable. So one must turn to finding some other distribution, say $P^*(X)$, that is close to $P(X|Y)$ but which is more easily sampled from and computed.

A common pitfall of importance sampling: As a final word on importance sampling, it should be pointed out that *the tails of the distributions matter*! While $h(x)$ might be roughly the same shape as $g(x)$, serious difficulties arise if $h(x)$ gets small much faster than $g(x)$ out in the tails. In such a case, though it is improbable (by definition) that you will realize a value x_i from the far tails of $h(x)$, if you do, then your Monte Carlo estimator will take a jolt—$g(x_i)/h(x_i)$ for such an improbable x_i may be orders of magnitude larger than the typical values $g(x)/h(x)$ that you see.

As an example of this phenomenon, investigate Figure 2 which shows the histogram of 5,000 Monte Carlo estimates of the area between -50 and 50 of a t_1 density (truncated at -50 and 50 and renormalized so the exact area is 1). The importance sampling function used for this was a standard, unit normal density, which obviously gets small in the tails much faster than a cauchy (see Figure 1(b)). Note in particular that about 15 of the 5,000 Monte Carlo estimates were greater than 1.5!

Further reading: A classic reference on Monte Carlo is Hammersley and Handscomb (1964). They describe several other variance-reduction techniques that you might find interesting.

References

Baum, L. E., 1972 An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In O. Shisha (Ed.), *Inequalities–III: Proceedings of the Third Symposium on Inequalities Held at the University of California, Los Angeles, September 1–9, 1969*, pp. 1–8. New York: Academic Press.

Feller, W., 1957 *An Introduction to Probability Theory and Its Applications, 2nd Edition*. New York: John Wiley & Sons.

Hammersley, J. M. and D. C. Handscomb, 1964 *Monte Carlo Methods*. London: Methuen & Co Ltd.

Ripley, B. D., 1987 *Stochastic Simulation*. New York: Wiley & Sons.

Rubinstein, B. Y., 1981 *Simulation and the Monte Carlo Method*. New York: Wiley & Sons.