CSCI-6971 Lecture Notes: Monte Carlo integration*

Kristopher R. Beevers
Department of Computer Science
Rensselaer Polytechnic Institute
beevek@cs.rpi.edu

February 21, 2006

1 Overview

Consider the following equation which arises frequently in robotics:

$$\mathbf{E}\left[h(x)\right] = \int_{\mathcal{A}} h(x) f(x)\, dx \tag{1}$$

where $f(x)$ is a PDF. We are often interested in computing some expected value, for example as part of an estimation or inference process. More generally, we need to compute the value of an integral:

$$I = \int_{\mathcal{A}} g(x)\, dx \tag{2}$$

It is often the case that I is difficult or impossible to compute analytically. Furthermore, x is frequently many-dimensional. The question then arises: how can we *approximate* the value of I efficiently?

1.1 Riemann approximation

A simple technique for approximating the value of an integral is to divide the region $\mathcal{A}$ into many small hypercubes, compute the value of $g(x)$ at a point in each of these hypercubes, and sum the results. For example, if $\mathcal{A} = [0,1]^d$ and $I = \int_{\mathcal{A}} g(x)\, dx$:

$$\hat{I} = \epsilon \sum_{i=1}^{N} g(\epsilon \cdot \text{offset}_i) \tag{3}$$

where ϵ is size of the d-dimensional hypercubes and is small, offset_i is the d-dimensional "index" of the ith hypercube in the hypergrid, and $N = 1/\epsilon^d$. This technique is known as the *Riemann approximation*.

One problem with the Riemann approximation is that the estimation error $|\hat{I} - I|$ can be arbitrarily bad for functions with unbounded derivatives: the variance of g in an ϵ-sized

*The primary sources for most of this material are: "Numerical Methods in Finance," P. Brandimarte, John Wiley & Sons, New York, 2002; "Monte Carlo Methods Vol. 1: Basics," M.H. Kalos and P.A. Whitlock, John Wiley & Sons, New York, 1986; "Monte Carlo Strategies in Scientific Computing," J.S. Liu, Springer, 2002; and the author's own notes.

hypercube can be large, and thus the value of g at a single point in the hypercube may not be representative of the integral of g over that hypercube. If, however, the derivative of g is bounded, e.g.:

$$|g(x) - g(x')| \leq B||x - x'|| \tag{4}$$

then the error of the Riemann approximation is also bounded:

$$|\hat{I} - I| = O\left(\frac{B\sqrt{d}}{N^{1/d}}\right) \tag{5}$$

(We will leave this result unproven.) Note that for a fixed N, the error increases exponentially in the dimension of the space. This property is often referred to as the "curse of dimensionality" and is a consequence of the fact that the volume of a hypercube is ϵ^d.

From Equation 5 we see that as d increases, to guarantee a fixed error we must increase N exponentially (i.e. we must keep ϵ constant). Thus while the Riemann approximation works well for computing low-dimensional integrals, it quickly becomes intractable as dimension increases.

1.2 Monte Carlo integration

Consider the following alternative approach to estimating I. First, draw N samples $x_i \sim f(x)$ where $f(x)$ is some probability distribution over $\mathcal{A}$. Often $f(x)$ is taken to be uniform over $\mathcal{A}$. Let $X_i = g(x_i)$. Now compute:

$$\hat{I} = \frac{\text{vol}(\mathcal{A})}{N} \sum_{i=1}^{N} X_i \tag{6}$$

This technique is known as *Monte Carlo integration*.

Definition 1.1. A *Monte Carlo method* is a technique that makes use of random numbers to perform a calculation that can be modeled as a stochastic process.

It is not difficult to see that $\mathbf{E}\left[\hat{I}\right] = I$. Furthermore, we see that $\text{var}\left(\hat{I}\right) = \text{var}\left(X_i\right)/N$ which means that the error $|\hat{I} - I| = \sqrt{\text{var}\left(\hat{I}\right)}$ decreases as $1/\sqrt{N}$. This seems almost like a miracle because there is no dependence on d. Monte Carlo integration does not suffer from the curse of dimensionality.

Let us examine this property more thoroughly. A useful tool is Hoeffding's inequality, which is a direct result of the Chernoff bound. If N x_i's are independently drawn from the same distribution such that $x_i \in [A, A']$ for all i, then:

$$P\left(\left|\sum_{i=1}^{N} x_i - \sum_{i=1}^{N} \mathbf{E}\left[x_i\right]\right| \geq \epsilon\right) \leq 2e^{\frac{-2\epsilon^2}{N(A-A')}} \tag{7}$$

Plugging in $\hat{I}$ and I we have:

$$P\left(|\hat{I} - I| \geq \epsilon\right) = P\left(\left|\sum_{i=1}^{N} g(x_i) - NI\right| \geq N\epsilon\right) \leq 2e^{\frac{-2N\epsilon^2}{B^2 d}} \tag{8}$$

The denominator in the exponent on the right hand side is a result of our assumption of a bounded derivative of g over the interval $[A, A']$ (Equation 4). Suppose we choose some desired confidence level $\eta = 2e^{-2N\epsilon^2/B^2 d}$. Then solving for ϵ:

$$\epsilon = \sqrt{\frac{B^2 d \log\left(\frac{2}{\eta}\right)}{2N}} \tag{9}$$

and it follows that with probability at least $1 - \eta$,

$$|\hat{I} - I| \leq c \left(\frac{B\sqrt{d}}{\sqrt{N}} \right) \tag{10}$$

where $c = \sqrt{\log(2/\eta)/2}$. Thus, as expected, $|\hat{I} - I| = O(B\sqrt{d}/\sqrt{N})$. Note that we can fix any two of N, ϵ, or η and solve for the other.

The primary attraction of Monte Carlo methods is their immunity to the curse of dimensionality. When computing low-dimensional integrals basic Monte Carlo techniques are not competitive with deterministic approaches such as the Riemann approximation. However, as dimension increases and given a fixed amount of computation time (i.e., a fixed N), the Monte Carlo approach quickly outpaces most deterministic methods.

Of course, another factor is that implementing Monte Carlo integration is extremely straightforward. In some cases, this aspect makes Monte Carlo methods useful even when more complicated deterministic approaches might lead to more accurate estimates.

2 Random number generation

We now briefly switch gears and discuss the problem of generating random numbers from a PDF $f(x)$. This will be helpful in understanding some improvements to Monte Carlo integration, which we will discuss next.

2.1 Inverse transform

Suppose we are able to analytically compute $F(x)$, the CDF associated with $f(x)$. Additionally, suppose $F(x)$ is easily invertible. Then we can use the following strategy to generate random numbers according to $f(x)$, given a simple uniform random number generator: draw $U \sim \text{UNIFORM}[0, 1]$, and return $X = F^{-1}(U)$. It is easy to see that X is indeed drawn from the desired distribution:

$$P(X \leq x) = P\left(F^{-1}(U) \leq x\right) = P(U \leq F(x)) = F(x) \tag{11}$$

The major limitation with the inverse transform is that it requires an invertible CDF. In some cases the requirement is easily met, as with the exponential distribution $X \sim e^{\mu}$, where $F(x) = 1 - e^{\mu x}$ so $F^{-1}(x) = -\ln(1 - U)/\mu$.

2.2 Acceptance-rejection

A more general technique can be used when $F(x)$ is hard to invert. Suppose we know some function $t(x)$ such that $t(x) \geq f(x)$ for all x in the support $\mathcal{A}$ of f. Since $f(x)$ is a PDF, $t(x)$ is clearly not, but we can define a PDF as follows:

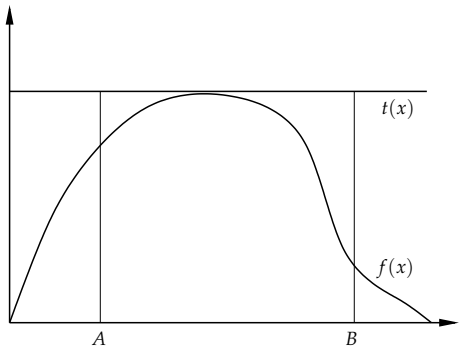
$$r(x) = \frac{t(x)}{\int_{\mathcal{A}} t(x)\, dx} \tag{12}$$

If it is easy for us to simulate $r(x)$ using Monte Carlo techniques then we can use the following procedure to generate random variables according to f:

1. Generate $Y \sim r(x)$
2. Generate $U \sim \text{UNIFORM}[0, 1]$

3. if $U \leq f(Y)/t(Y)$ return Y; otherwise repeat the process.

The following picture illustrates the idea:



Suppose we draw $Y = A$. Then the sample is likely to be "accepted" since $f(A)/t(A)$ is close to 1. On the other hand, if we draw $Y = B$ the sample is likely to be "rejected" since $f(B)/t(B)$ is small. This matches what we'd expect based on inspection of f.

The average number of iterations of the procedure required for a sample to be accepted is $\int_{\mathcal{A}} t(x)\,dx$. For this reason it is desirable for t to "fit" f as closely as possible while meeting the condition $t(x) \geq f(x)$ so that the number of iterations is minimized.

3 Variance reduction

We now turn to the problem of improving the accuracy of the Monte Carlo estimate $\hat{I}$ of an integral I. Recall that

$$|\hat{I} - I| = \sqrt{\frac{\text{var}\,(X_i)}{N}} \tag{13}$$

since $\hat{I}$ is the sum of N independent identically distributed random variables. Clearly we can decrease the error by increasing N, the number of samples. However this approach has diminishing returns since the error decreases as $O(1/\sqrt{N})$. An alternative is to decrease the variance of the samples. Much of the work on Monte Carlo methods has focused on so-called *variance reduction techniques*.

3.1 Antithetic variates

Suppose we were to generate $2N$ samples:

$$\begin{gathered} X_1, X_2, \ldots, X_N \\ Y_1, Y_2, \ldots, Y_N \end{gathered} \tag{14}$$

Now, define $Z_i = \frac{X_i + Y_i}{2}$ for all $i = 1 \ldots N$, and compute $\tilde{I} = \frac{1}{N}\sum_{i=1}^{N} Z_i$ as our Monte Carlo estimate. As before, $\text{var}\left(\tilde{I}\right) = \text{var}\,(Z_i)/N$. However, it is possible for $\tilde{I}$ to be a better estimate of I than the vanilla Monte Carlo estimate $\hat{I}$ computed using $2N$ samples!

The key insight is that by introducing correlation between X_i and Y_i we can affect the variance of Z_i since if X_i and Y_i are not independent:

$$\text{var}\,(Z_i) = \text{var}\left(\frac{X_i + Y_i}{2}\right) = \frac{1}{4}\left(\text{var}\,(X_i) + \text{var}\,(Y_i) + 2\text{cov}\,(X_i, Y_i)\right) \tag{15}$$

When X_i and Y_i are independent, the covariance term is zero so $\text{var}(Z_i) = \text{var}(X_i)/2$ and $\text{var}\left(\tilde{I}\right) = \text{var}\left(\hat{I}\right)$. However, if X_i and Y_i are negatively correlated, $\text{cov}(X_i, Y_i) < 0$ so $\text{var}(Z_i) < \text{var}(X_i)/2$ and thus $\text{var}\left(\tilde{I}\right) < \text{var}\left(\hat{I}\right)$.

This leaves the question of how to generate negatively correlated random variables. For certain types of random variables this is reasonably straightforward. For example, if we are trying to estimate $\int_0^1 g(x)\,dx$ we could generate N uniform random variables $U_i \sim \text{UNIFORM}[0,1]$ and then compute $Z_i = (g(U_i) + g(1-U_i))/2$. Similarly, to generate negatively correlated random variables from a standard Normal distribution, we can draw $X_i \sim N(0;1)$ and compute $Z_i = (g(X_i) + g(-X_i))/2$. Note that in fact for both of these approaches we have generated only N random variables but our estimate is better than we can achieve using $2N$ random variables with standard Monte Carlo!

There is an important limitation of this technique: *the function we are integrating must be monotonic*. If the function is nonmonotonic we cannot ensure that $g(X_i)$ and $g(Y_i)$ are negatively correlated, even if X_i and Y_i are negatively correlated. Thus, if g is nonmonotonic and $g(X_i)$ and $g(Y_i)$ are in fact positively correlated, the technique actually increases the variance of our estimate!

3.2 Common random variates

A similar technique can be applied when our goal is to estimate the value of a function that can be expressed as a difference between two random variables $Z_i = X_i - Y_i$. The only difference is that X_i and Y_i should be positively correlated. The same monotonicity requirement as for antithetic variates remains.

One problem for which this technique is useful is when our goal is to estimate the sensitivity of a parameterized function to its parameter. For example, suppose our goal is to estimate the sensitivity of $h(\alpha) = \mathbf{E}_\omega[f(\alpha;\omega)]$ to the parameter α. (Note that randomness arises only through the variable ω.) In other words we wish to estimate $\frac{dh(\alpha)}{d\alpha}$ which we cannot compute analytically. We can instead estimate

$$\frac{h(\alpha + \delta\alpha) - h(\alpha)}{\delta\alpha} \tag{16}$$

for a small value of $\delta\alpha$ by generating samples of the difference, i.e.:

$$Z_i \sim \frac{f(\alpha + \delta\alpha;\omega) - f(\alpha;\omega)}{\delta\alpha} \tag{17}$$

By introducing positive correlation between $f(\alpha + \delta\alpha;\omega)$ and $f(\alpha;\omega)$ the variance is reduced versus the default approach.

3.3 Control variates

Suppose we have access to some potentially useful outside information. In particular, suppose our goal is to estimate $\mathbf{E}[X]$ and we know $\mathbf{E}[Y] = \nu$ for some related random variable Y. Furthermore, we know that $\text{cov}(X, Y) \neq 0$, although we may not actually know the value. It seems that we should somehow be able to use our knowledge of Y to improve our estimate of $\mathbf{E}[X]$.

In fact we can employ the following strategy. Generate two samples X_i and Y_i. Suppose $Y_i > \nu$ and we know that $\text{cov}(X, Y) > 0$. Then X_i is probably also an overestimate of X, so we correct it:

$$X_i^c = X_i + c(Y_i - \nu) \tag{18}$$

where c is a control parameter we can choose. Then:

$$\begin{aligned} \mathbf{E}\left[X_i^c\right] &= \mathbf{E}\left[X_i\right] + c(\mathbf{E}\left[Y_i - \nu\right]) = \mathbf{E}\left[X_i\right] && (19) \\ \text{var}\left(X_i^c\right) &= \text{var}\left(X_i\right) + c^2\text{var}\left(Y_i - \nu\right) + 2c \cdot \text{cov}\left(X_i, Y_i - \nu\right) && (20) \\ &= \text{var}\left(X_i\right) + c^2\text{var}\left(Y_i\right) + 2c \cdot \text{cov}\left(X_i, Y_i\right) && (21) \end{aligned}$$

It remains to determine a value of c. We want to minimize the variance so we take its derivative:

$$\frac{d\text{var}\left(X_i^c\right)}{dc} = 2c \cdot \text{var}\left(Y_i\right) + 2\text{cov}\left(X_i, Y_i\right) \tag{22}$$

Setting this equal to zero we obtain:

$$c^* = \frac{-\text{cov}\left(X_i, Y_i\right)}{\text{var}\left(Y_i\right)} \tag{23}$$

and with some manipulation we see:

$$\frac{\text{var}\left(X_i^{c^*}\right)}{\text{var}\left(X_i\right)} = 1 - \rho_{XY}^2 \tag{24}$$

In other words, we have reduced the variance of our samples and thereby reduced the variance of our estimate. Of course we may not initially know $\text{cov}\left(X, Y\right)$ or $\text{var}\left(Y\right)$. A simple solution is the use some pilot samples to estimate these values, and then proceed as above.

3.4 Rao-Blackwellization

Rao-Blackwellization, also known as "conditioning," is based on the idea that "one should carry out analytical computation as much as possible." In other words, Monte Carlo simulation should be used only where necessary.

If our goal is to estimate $I = \mathbf{E}\left[h(x)\right]$, the standard strategy is to compute $\hat{I} = \frac{1}{N}\sum_i X_i$ where the X_i are samples of $h(x)$. Suppose instead that we can decompose x into two parts, $x^{(a)}$ and $x^{(b)}$, and that we can compute $\mathbf{E}\left[h(x)|x^{(b)}\right]$ analytically. This suggests that we can instead simulate only $x^{(b)}$ and compute

$$\tilde{I} = \frac{1}{N}\sum_i \mathbf{E}\left[h(X_i|x_i^{(b)})\right] \tag{25}$$

From the Law of Iterated Expectations we have:

$$\mathbf{E}\left[h(x)\right] = \mathbf{E}\left[\mathbf{E}\left[h(x)|x^{(b)}\right]\right] \tag{26}$$

and from the Law of Total variance we know:

$$\text{var}\left(h(x)\right) = \text{var}\left(\mathbf{E}\left[h(x)|x^{(b)}\right]\right) + \mathbf{E}\left[\text{var}\left(h(x)|x^{(b)}\right)\right] \tag{27}$$

so we have:

$$\text{var}\left(\hat{I}\right) = \frac{\text{var}\left(h(x)\right)}{N} \geq \frac{\text{var}\left(\mathbf{E}\left[h(x)|x^{(b)}\right]\right)}{N} = \text{var}\left(\tilde{I}\right) \tag{28}$$

i.e. var $\left(\hat{I}\right) \geq$ var $\left(\tilde{I}\right)$ so we have achieved some variance reduction. The basic idea here is to break $h(x)$ into parts ("factor" it); if we can analytically compute some of the parts, we only need to use Monte Carlo simulation to estimate the other "subproblems."

A problem with Rao-Blackwellization is that the factorization can be very problem dependent. Another is that in some cases the analytical computation may be more expensive than just simulating the entire system. Often, Rao-Blackwellization can be coupled with importance sampling techniques to reduce the impact of analytical computations.

3.5 Stratified sampling

Suppose our goal is to compute the integral $I = \int_{\mathcal{A}} h(x)\,dx$ such that we can decompose $\mathcal{A}$ into disjoint subregions ("strata") with $h(x)$ being relatively homogeneous within each strata. The idea is to compute the integral over each strata and then combine the results, i.e.:

$$\tilde{I}_i = \frac{1}{N_i}\sum_{j=1}^{N_i} h(X_j^i) \tag{29}$$

$$\tilde{I} = \sum_{i=1}^{M} \tilde{I}_i \tag{30}$$

Then, we have

$$\text{var}\left(\tilde{I}\right) = \sum_{i=1}^{M} \frac{\text{var}\left(\tilde{I}_i\right)}{N_i} \tag{31}$$

where var $\left(\tilde{I}_i\right)$ is the variance of $h(x)$ in strata i. As long as we can guarantee relatively low variance within each strata, this approach can lead to a reduction in overall variance of our estimate since normal Monte Carlo sampling yields var $\left(\tilde{I}\right) = \text{var}_{\mathcal{A}}(h(x))$.

Stratification can simply be done by dividing $\mathcal{A}$ into uniform cells. In some cases most of the variation in $h(x)$ lies along a subset of the dimensions of the function in which case stratification need be done only on those dimensions. In choosing N_i, the number of samples to assign to strata i, we should generally allocate more samples to strata with higher variance. Some samples can be devoted to computing pilot estimates of the strata variances, and these estimates can be used in allocating samples by solving a nonlinear programming problem to minimize var $\left(\hat{I}\right)$ with respect to the N_is. In many cases this overcomplicates matters and uniform allocation of samples gives sufficient results.

3.6 Importance sampling

Our goal as usual is to compute $\mathbf{E}\left[h(x)\right] = \int_{\mathcal{A}} h(x)f(x)\,dx$ where $f(x)$ is a PDF. Up until now we have mostly taken $f(x)$ to be uniform over $\mathcal{A}$ but it need not be. Furthermore, $f(x)$ (regardless of its distribution) may not be the best PDF for the purposes of Monte Carlo integration. Intuitively, we want a PDF with similar behavior to the entire integrand.

Let us introduce a different density $g(x)$ such that $g(x) = 0 \rightarrow f(x) = 0$. We will term $g(x)$ the *importance density* or the *proposal distribution*. We start with some manipulation:

$$\mathbf{E}\left[h(x)\right] = \int h(x)f(x)\,dx \tag{32}$$

$$= \int \frac{h(x)f(x)}{g(x)}g(x)\,dx \tag{33}$$

$$= \mathbf{E}_g\left[\frac{h(x)f(x)}{g(x)}\right] \tag{34}$$

Our approach is to generate samples $X_i \sim g(x)$ and compute:

$$\tilde{I} = \frac{1}{N} \sum_{i=1}^{N} \frac{h(X_i) f(X_i)}{g(X_i)} \tag{35}$$

Clearly, our choice of $g(x)$ is crucial in obtaining some reduction in variance. We illustrate the optimal choice with a discrete example. Suppose our goal is to estimate $F(N) = \sum_{i=1}^{N} h(x_i)$ where each x_i belongs to a set of N discrete points, with N so large that we cannot compute $F(N)$ outright. We can estimate $F(N)$ using Monte Carlo simulation: $\hat{F} = \frac{N}{M} \sum_{j=1}^{M} h(x_j)$. Suppose $p_i = 1/N$ is the probability of sampling point x_i. Then using the standard Monte Carlo approach:

$$\hat{F} = \frac{1}{M} \left(\frac{\sum_{j=1}^{M} h(x_k)}{1/N} \right) = \frac{1}{M} \sum_{j=1}^{M} \frac{h(x_j)}{p_j} \tag{36}$$

The idea of importance sampling is to pick p_i more intelligently. For example, suppose we pick:

$$p_i = \frac{h(x_i)}{F(N)} \tag{37}$$

Then our estimate is:

$$\tilde{F} = \frac{1}{M} \sum_{j=1}^{M} \frac{h(x_j)}{p_j} = \frac{1}{M} \sum_{j=1}^{M} \frac{h(x_j) F(N)}{h(x_j)} = F(N) \tag{38}$$

In other words by choosing our proposal distribution according to Equation 37 we always get the right answer, with zero variance, no matter how we sample! There is of course a serious problem: Equation 37 requires $F(N)$, which is the answer we seek, so we cannot actually compute p_i. Certainly, however, we can approximate it, e.g. using pilot samples.

Here is an algorithm for approximately optimal importance Monte Carlo integration:

1. Create N bins and compute $\overline{h_i}(x)$ (the mean value of $h(x)$) for each bin i
2. Repeat K times:
   (a) Generate $U_i \sim \text{UNIFORM}[0, 1]$
   (b) Pick bin k iff $F_{k-1} \leq U_i < F_k$ where $F_k = \sum_{i=1}^{k} p_i$ with $p_i = \overline{h_i} / \sum_{i=1}^{N} \overline{h_i}$
   (c) Generate $\omega_i \sim \text{UNIFORM}[0, \frac{1}{N}]$
   (d) Set $X_i = \frac{k-1}{N} + \omega_i$
3. Compute $\frac{1}{K} \sum_{i=1}^{K} \frac{h(X_i)}{N p(X_i)}$ where $p(X_i) = \overline{h}(X_i) / \sum_{j=1}^{N} \overline{h_j}(X_i)$

Importance sampling is useful for most situations where Monte Carlo integration is employed, but is particularly helpful when sampling from the tails of a distribution, i.e. when computing $\mathbf{E}[h(x)|x \in \mathcal{A}]$ where $x \in \mathcal{A}$ is a rare event. With importance sampling, we can use a PDF such that the event is more likely to occur.