Monte Carlo Integration

1 Motivation

Up to this point, we have studied exact methods for performing numerical integration (Trapezoidal/Simpson's Rule). These methods involve evaluating the function at a certain number of points over the integration region. To estimate an integral in d dimensions, with N function evaluations in each dimension, the cost of these methods is

$$C_S = N^d. \tag{1}$$

It is clear that if d is very large, it is impossible to use these methods. We need a completely new approach to the problem which will avoid this scaling problem.

Monte Carlo Integration is a powerful tool for evaluating integrals of arbitrary dimension, up to some well defined uncertainty.

Given a sample set, ϵ_x, of N uniform variate random numbers, x_i, in the range $[0, 1]$,

$$\epsilon_x = \{x_1, x_2, \ldots, x_N\}, \tag{2}$$

let $f(x)$ be a function on x_i.

The *Sample Mean* is

$$\bar{f}(x)^{(N)} = \frac{1}{N} \sum_{i=1}^{N} f(x_i). \tag{3}$$

The expectation value of the function, $f(x)$, is

$$\langle f \rangle = \int_0^1 f(x) dx. \tag{4}$$

The **Strong Law of Large Numbers** says that

$$\mathcal{P}\left(\lim_{N \to \infty} \bar{f}(x)^{(N)} = \langle f \rangle \right) = 1. \tag{5}$$

This means that we can estimate the expected value of f by sampling it a large number of times and computing the sample mean. This prcess can be extended to integrals over arbitrary intervals, $[a, b]$, quite simply.

Firstly, generate ϵ_X, a set of uniform variate random numbers in the range $[a, b]$, of length $V = b - a$. Then evaluate

$$I = \int_a^b f(x)dx \approx \frac{V}{N} \sum_{i=1}^{N} f(x_i). \tag{6}$$

Thus we can estimate the integral by randomly sampling the integration range.

When performing a Monte Carlo Integration, it is essential to include an estimate of the error.

2 Error Estimation

Since Monte Carlo is a stochastic, or random, method the uncertainty in the estimate of the integral is very important. For a Monte Carlo Integration, the variance, $\mu(f)$, is given by

$$\begin{aligned}
\mu(f) &= \frac{1}{N}\left[\langle (f - \langle f \rangle)^2 \rangle\right] && (7)\\
&= \frac{1}{N}\left[\langle f^2 - 2f\langle f \rangle + \langle f \rangle^2 \rangle\right] && (8)\\
&= \frac{1}{N}\left[\langle f^2 \rangle - 2\langle f \rangle \langle f \rangle + \langle f \rangle^2\right] && (9)\\
&= \frac{1}{N}\left[\langle f^2 \rangle - \langle f \rangle^2\right] && (10)\\
&= \frac{1}{N}\left[\frac{1}{N}\sum_{i=1^N} f^2(x_i) - \left(\frac{1}{N}\sum_{i=1^N} f(x_i)\right)^2\right] && (11)
\end{aligned}$$

$\langle f \rangle^2$ comes directly from the Monte Carlo Integration. $\langle f^2 \rangle$ can be evaluated simultaneously. The standard deviation, σ, is related to $\mu(f)$ simply by

$$\sigma = \sqrt{|\mu(f)|}. \tag{12}$$

Thus it is clear that

$$\sigma \propto \sqrt{\frac{1}{N}}. \tag{13}$$

3 Algorithm

1. Generate uniform variate random number, x, in the range $[a, b]$.
2. Calculate $f(x)$.
3. Set $I = I + f(x)$, $I_2 = I_2 + f(x)^2$.
4. Repeat 1, 2 and 3 N times.
5. Set $I = I/N$, $I_2 = I_2/N$.
6. Set $\sigma = \sqrt{\frac{1}{N}|I^2 - I_2|}$.
7. The estimate of the integral is $I \pm \sigma$.

4 Monte Carlo in Higher Dimensions

This process can be generalised quite simply to calculate integrals of arbitrary dimension;

$$I = \int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} dx_1 \dots dx_n f(\vec{x}) \tag{14}$$

Here, $\vec{x} = (x_1, \dots, x_n)$.

In this case, we randomly sample the n dimensional hypercube, by generating n uniform variate random numbers, $x_1, \dots, x_n$, in the range $[a_1, b_1], \dots, [a_n, b_n]$, to form a vector, $\vec{x}$.

A sample set of N such vectors, $\epsilon_x = \{\vec{x}_1, \dots, \vec{x}_n\}$, of such vectors can be created. (Here the subscript refers to the position of the vector in the sample set, rather than a component of the vector). The integral can then be estimated using the sample mean;

$$I \approx V \bar{f}(\vec{x})^{(N)} = \frac{V}{N} \sum_{i=1}^{N} f(\vec{x}_i). \tag{15}$$

In this expression, V is the volume of the hypercube;

$$V = \prod_{i=1}^{n} (b_i - a_i). \tag{16}$$

Again, the uncertainty in the estimate is crucial. The standard deviation is given by

$$\sigma = \sqrt{\frac{1}{N} \left| \frac{1}{N} \sum_{i=1^N} f^2(\vec{x}_i) - \left(\frac{1}{N} \sum_{i=1^N} f(\vec{x}_i) \right)^2 \right|}. \tag{17}$$

Note that while we have increased the dimensional of the integral, the number of function evaluations remains unchanged at N (c.f. N^d for Simpson's Rule). This means that Monte Carlo will be more effective for computing high dimension integrals.

5 Algorithm

1. Generate $\vec{x}$ with components, x_i, uniform variate random numbers in the range $[a_i, b_i]$.
2. Calculate $f(\vec{x})$.
3. Set $I = I + f(\vec{x})$, $I_2 = I_2 + f(\vec{x})^2$.
4. Repeat 1, 2 and 3 N times.
5. Set $I = I/N$, $I_2 = I_2/N$.
6. Set $\sigma = \sqrt{\frac{1}{N}|I^2 - I_2|}$.
7. The estimate of the integral is $I \pm \sigma$.

6 Comparison of Simpson's Rule and Monte Carlo

We demonstrated above how Monte Carlo might be more efficient than Simpson's Rule as the dimension of the integral increases. We must know pin down at exactly what dimension this crossover in efficiency takes place. In the discussion below, N is the number of function evaluations and d is the dimension of the integral. As was noted previously, the cost, C_S, and error, E_S, of Simpson's Rule are

$$C_S \sim N^d, \tag{18}$$

$$E_S \sim \frac{1}{N^2}. \tag{19}$$

This means that

$$C_S \sim E_S^{-d/2}. \tag{20}$$

From our discussions of Monte Carlo Integration, we see that the cost, C_{MC}, and error, E_{MC}, are

$$C_{MC} \sim N, \tag{21}$$
$$E_{MC} \sim \frac{1}{\sqrt{N}}. \tag{22}$$

Thus, we have

$$C_{MC} \sim E_{MC}^{-2}. \tag{23}$$

If we assume that the two methods have the same cost, C, then

$$C_{MC} = C_S = C \tag{24}$$
$$\Rightarrow \quad E_{MC}^{-2} = E_S^{-d/2}. \tag{25}$$

Next, we assume that at some dimension, d, the two methods produce the same error, E, from the same cost, i.e.

$$E_{MC} = E_S = E. \tag{26}$$

Putting all this together, we see that

$$E^{-2} = E^{-d/2} \tag{27}$$
$$\Rightarrow -2 = -d/2 \tag{28}$$
$$\Rightarrow d = 4. \tag{29}$$

Thus both methods are equally efficient at computing 4 dimensional integrals. For integrals of higher dimension, Monte Carlo will be more efficient, while below this, Simpson's Rule will be better.