Physics through Computational Thinking

The Monte Carlo method

Auditya Sharma and Ambar Jain
Dept. of Physics, IISER Bhopal

Outline

In this module we look at

1. Importance sampling
2. An example.

Monte Carlo Integration

```
Clear["Global`*"]
```

Suppose we want to find the definite integral

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

by Monte Carlo integration. One strategy, as we have seen, is to generate a large number of random points x_i drawn from a uniform distribution in the interval [0, 1], and simply find the mean of $f(x_i)$ evaluated at these points. That is,

$$I \approx \frac{1}{N_{\text{samp}}} \sum_{i=1}^{N_{\text{samp}}} f(x_i), \tag{2}$$

where N_{samp} is the number of points sampled.

If we rewrite the definite integral we need to evaluate as

$$I = \int_0^1 \frac{f(x)}{p(x)}\, p(x)\, dx, \tag{3}$$

where $p(x)$ is some suitable probability distribution. We can still get an estimate of the same integral, by sampling points drawn from the distribution $p(x)$. That is, the idea is to compute

$$I \approx \frac{1}{N_{\text{samp}}} \sum_{i=1}^{N_{\text{samp}}} \frac{f(x_i)}{p(x_i)}, \tag{4}$$

where x_i are now points drawn not from the uniform distribution but from the distribution $p(x)$. To get an advantage out of this, we must choose $p(x)$ appropriately. *Importance* sampling is a way of thus choosing a distribution such that greater importance is given to points where the function has greater value. Thus, the convergence of the algorithm is speeded up. Finding an optimal $p(x)$ for a given context, is an art form. The general idea is to come up with a distribution that resembles the function itself.

Estimating π

Now, let us consider the integral

$$I = \int_0^1 \frac{4}{1+x^2}\, dx. \tag{5}$$

Let us try to come up with an exponentially dropping distribution.

$$p(x) = A\, e^{-x} \tag{6}$$

We can get *Mathematica* to normalize it:

```
Integrate[Exp[-x], {x, 0, 1}]
```

$$\frac{-1+e}{e}$$

So, let us choose our distribution to be:

$$p(x) = \frac{e^{1-x}}{e-1} \tag{7}$$

Now, we have to figure out a way to generate points that are drawn from this distribution. *Mathematica* can help us here.

```
p[x_] := Exp[1 - x]/(e - 1)

Dist = ProbabilityDistribution[p[x], {x, 0, 1}];

Plot[PDF[Dist, x], {x, -1, 2}];

f[x_] := 4/(1 + x^2)

Histogram[RandomVariate[Dist, 100]];
```

```
Table[xi = RandomVariate[Dist]; f[xi]/p[xi], {100}];

data = Table[n = 2^k;
    {n, Mean[Table[Mean[Table[xi = RandomVariate[Dist]; f[xi]/p[xi], {n}]], {1}]]}, {k, 5, 10}];
TableForm[data]
```

32	3.0817
64	3.16435
128	3.12875
256	3.12487
512	3.14129
1024	3.14549

```
ListPlot[data, Joined → True]
```