

# Monte Carlo Integration and the Metropolis Algorithm

Cody Petrie

May 18, 2015

## 1 Background Information

### 1.1 Probability Theory

Most of these next few sections are pulled from [1]. Here are some of the basics of probability theory. The cumulative distribution function (CDF) is the probability that a random variable ( $X$ ) is less than or equal to some value ( $x$ ).

$$\text{CDF}(x) = \Pr\{X \leq x\} \quad (1)$$

Note that this is always decreasing with the maximum value being 1. The derivative of this is then the probability distribution function (PDF), which describes the probability of  $X$  being between  $x$  and  $dx$ , where  $dx$  is small.

$$\text{PDF}(x) = \frac{d}{dx} \text{CDF}(x) \quad (2)$$

Notice here that the PDF will always be positive because the CDF is monotonically increasing (always only increasing). Now using the definition of the PDF we can determine the probability that  $X$  is between the values  $a$  and  $b$  by integrating,

$$\Pr\{a \leq X \leq b\} = \int_a^b \text{PDF}(x) dx. \quad (3)$$

Now the expectation value,  $E[X]$ , of is the value that is most likely to be chosen when choosing random values from  $\text{PDF}(X)$ . Assuming  $X$  is continuous it can be calculated by

$$E[X] = \int_{-\infty}^{\infty} x \text{PDF}(x) dx. \quad (4)$$

Obviously this integral goes to a sum if  $X$  is discrete. Now if we define the random variable to be  $Y = f(X)$  over the domain  $[a, b]$ , the expectation value of  $Y$  becomes,

$$E[Y] = \int_a^b f(x) \text{PDF}(x) dx. \quad (5)$$

The variance  $\sigma^2[Y]$  is then given by

$$\sigma^2[Y] = E[(Y - E[Y])^2]. \quad (6)$$

Jarosz then goes on to list a few properties that can be found from the given information above. I will just list them here.

$$E[aY] = aE[Y] \quad (7)$$

$$\sigma^2[aY] = a^2\sigma^2[Y] \quad (8)$$

$$E\left[\sum_i Y_i\right] = \sum_i E[Y_i] \quad (9)$$

$$\sigma^2[Y] = E[Y^2] - E[Y]^2 \quad (10)$$

$$\sigma^2\left[\sum_i Y_i\right] = \sum_i \sigma^2[Y_i] \quad (11)$$

Equation 10 shows up a lot in physics so I will take the time here to show it using the properties above. Start with equation 6 and expand to get

$$\sigma^2[Y] = E[Y^2 - 2YE[Y] + E[Y]^2]. \quad (12)$$

Using equations 7 and 9 and noting the  $E[Y]$  is a constant we get

$$\sigma^2[Y] = E[E[Y^2] - 2YE[Y] + E[Y]^2] \quad (13)$$

$$= E[E[Y^2]] - 2E[Y]E[Y] + E[E[Y]^2] \quad (14)$$

$$= E[Y^2] - 2E[Y]^2 + E[Y]^2 \quad (15)$$

$$= E[Y^2] - E[Y]^2, \quad (16)$$

which is indeed the same as equation 10.

## 2 Monte Carlo Integration

Now I will apply what we have learned above to integrating a simple 1D function like  $f(x)$  over the range  $[a, b]$ . I will not try to obtain an approximation for the integral

$$I = \int_a^b f(x)dx. \quad (17)$$

Start with the expectation value of  $f(x)$ .

$$\langle f(x) \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(X_i) \quad (18)$$

$$= \int_a^b f(x)\text{PDF}(x)dx \quad (19)$$

Now if  $X_i$  are pulled from a uniform distribution  $U[a, b]$  then  $\text{PDF}(x) = 1/(b - a)$  so we get

$$I = \lim_{N \rightarrow \infty} \frac{(b - a)}{N} \sum_{i=1}^N f(X_i). \quad (20)$$

And thus we can approximate  $I$  with

$$I \approx \frac{(b - a)}{N} \sum_{i=1}^N f(X_i). \quad (21)$$

This makes sense since it is like taking the average value of  $f(x)$  and then multiplying it by the length of the integration interval. This can be generalized to pulling  $X_i$  from an arbitrary distribution by

$$I \approx \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{\text{PDF}(X_i)}. \quad (22)$$

Importance sampling is the technique of picking  $\text{PDF}(x)$  to be similar to  $f(x)$  thus randomly choosing samples close to where  $f(x)$  is “important”. It is probably important to mention here that the error goes as  $N^{-1/2}$  regardless of dimension.

## 2.1 Importance Sampling

I will now go over the statement made in equation 22 in more detail. If instead of pulling the samples  $X_i$  from a uniform distribution (some of the  $X_i$  may represent smaller parts of the integrand and be less important) it is wise to sample  $X_i$  from a distribution that resembles the integrand. First let the integrand be  $g(x) = f(x)/\text{PDF}(x)$  where  $p(x)$  is the probability distribution that you want to sample from. Now the integral becomes

$$I = \int_a^b f(x)dx = \int_a^b g(x)\text{PDF}(x)dx. \quad (23)$$

As before the last integral can be approximated as the average value of  $g(x)$  and thus

$$I \approx \frac{1}{N} \sum_{i=1}^N g(X_i) = \frac{1}{N} \sum_{i=1}^N \frac{f(X_i)}{\text{PDF}(X_i)}. \quad (24)$$

## 3 The Metropolis Algorithm

Much of this is pulled from [2] and [3]. Since we want  $X_i$  to be pulled from a specific distribution  $\text{PDF}(x)$  it is often hard or impossible to do this by inverting  $\text{PDF}(x)$  and so we use the Metropolis algorithm. The Metropolis algorithm creates a random walk (sequence) of points that are distributed according to the distribution that you want.

The basic idea is that you start with a random number (or set of numbers, this is often called a walker in the literature) within the range that you need and then you generate

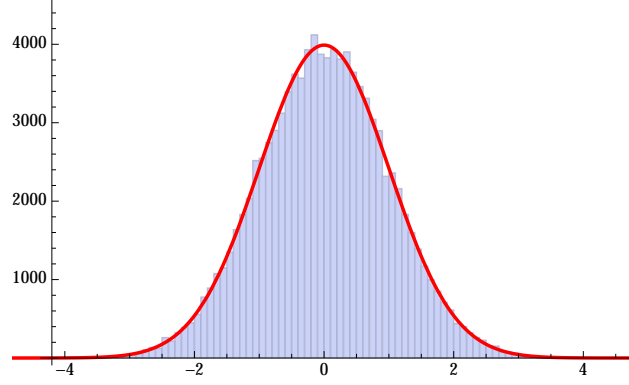


Figure 1: The histogram is of the random variables generated and the red curve is the original Normal distribution that I wanted to generate random variables from.

another number (walker) that you propose to move to. If the random numbers are chosen from a uniform distribution then the probability of accepting the move is given by

$$A(X' \rightarrow X) = \min \left( 1, \frac{\text{PDF}(X')}{\text{PDF}(X)} \right) \quad (25)$$

If the move is not accepted then the next walker in the sequence is the same as the previous.

A simple Mathematica code showing an example of this for a Normal distribution is shown below. The resulting distribution is shown in figure 1.

```
mu = 0; sig = 1;
P[x_] = 1/(sig*Sqrt[2 Pi])*Exp[-((x - mu)^2/(2*sig^2))];
num = 100000;
a = -5; b = 5;
dist = Table[0, {num}];
dist[[1]] = RandomReal[{a, b}];
For[i = 2, i <= num, i++, move = RandomReal[{a, b}];
  myrand = RandomReal[{0, 1}];
  If[P[move]/P[dist[[i - 1]]] >= myrand, dist[[i]] = move,
    dist[[i]] = dist[[i - 1]]];]
A = 10000;
p1 = Histogram[dist];
p2 = Plot[A*P[x], {x, -10, 10}, PlotRange -> All,
  PlotStyle -> {Thick, Red}];
Show[p1, p2]
```

## References

- [1] Wojciech Jarosz. *Efficient Monte Carlo Methods for Light Transport in Scattering Media*. PhD thesis, UC San Diego, September 2008.

- [2] Paul Richard Charles Kent. *Techniques and Applications of Quantum Monte Carlo*. PhD thesis, Robinson College, Cambridge, August 1999.
- [3] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal. Quantum monte carlo simulations of solids. *Rev. Mod. Phys.*, 73:33–83, 2001.