MCMC

Azmain Biswas

April 2023

Contents

List of Figures

Chapter 1

Introduction

In the ever-evolving landscape of mathematical and statistical research and application, the integration of simulation techniques has emerged as a powerful tool to unravel complex phenomena, validate theoretical frameworks, and facilitate a deeper understanding of intricate mathematical structures. Simulation techniques are usually used when any problem is heard to calculate with the help of traditional analytical method like simulation of a Solitaire Card game, to analysis this traditionally we have to deal with the number like $52! = 8.06 \times 10^{67}$ this number is beyond astronomically large so, dealing with this number is very hard even for a high performant computer, to deal with this problem Polish-American mathematician and nuclear physicist Stanistaw Ulam with the help of Von Neumann develope Monte Carlo Simulation.

Chapter 2

Generating Random Variables

2.1 Generating Discrete Random Variables

Main component of a simulation study is the ability to generate random number, where a random number represents the value of random variable uniform distribution on $(0, 1)$.

2.1.1 Pseudorandom Number Generation

Random numbers were originally either manually or mechanically generated, by using spinning wheels or dice rolling or card shuffling but the modern approach is to use a computer to successively generate pseudorandom numbers.

One of the common approaches to generate pseudorandom numbers starts with an initial value x_0, called seed, and then recursively computes successive values $x_n, n \geq 1$, by letting

$$x_n = ax_{n-1} \text{ modulo } m \tag{2.1}$$

where a and m are given positive integers, and where the equation (2.1) means that ax_{n-1} is divided by m and remainder is taken as the value of x_n. Thus, each value of x_n is either $0, 1, \dots, m-1$ and the quantity x_n/m is pseudorandom number and follows an approximation to the value of a uniform $(0, 1)$ random variable.

The approach specified by equation (2.1) to generate random numbers is called the Multiplicative Congruential Method.

Another method is

$$x_n = (ax_{n-1} + c) \text{ modulo } m$$

this method is known as *Mixed Congruential Generators* or *Linear congruential Generations (LCGs)* where c is a non-negative integer.

2.1.2 The Inverse Transform Method

Suppose we want to generate the value of a discrete random variable X having probability mass function

$$P(X = x_i) = p_i, \ i = 0, 1, \dots, \ \sum_i p_i = 1$$

To do this, we generate a random number from a uniform distribution $(0, 1)$ U, and set

$$X = \begin{cases} x_0 \text{ if } U < p_0 \\ x_1 \text{ if } p_0 \leq U \leq p_0 + p_1 \\ \vdots \\ x_j \text{ if } \sum_{i=0}^{j-1} p_i \leq U \leq \sum_{i=0}^{j} p_i \\ \vdots \end{cases}$$

Since, for $0 < a < b < 1, P(a \leq U < b) = b - a$, we have,

$$P(X = x_j) = P\left(\sum_{i=0}^{j-1} p_i \leq U < \sum_{i=0}^{j} p_i\right) = p_j.$$

So, X has the desired distribution.

Example 2.1.1 (Bernoulli Distribution). Let, $X \sim Ber(p)$ where p is success probability i.e. $P(X = 0) = 1 - p$ and $P(X = 1) = p$ and $0 \leq p \leq 1$. Then, to generate X we first generate $U \sim U[0, 1]$ then, we set

$$X = \begin{cases} 1, \text{ if } U \leq p \\ 0, \text{ if } U > p \end{cases}$$

Hence, X follows Bernoulli Distribution with the parameter p.

Algorithm for Inverse Transform Algorithm for Generating Bernoulli Distribution:
STEP 1: Generate a random variable $U \sim U[0, 1]$.
STEP 2: If $U \leq p$ set $X = 1$ or set $X = 0$.
STEP 3: Go to STEP 1.

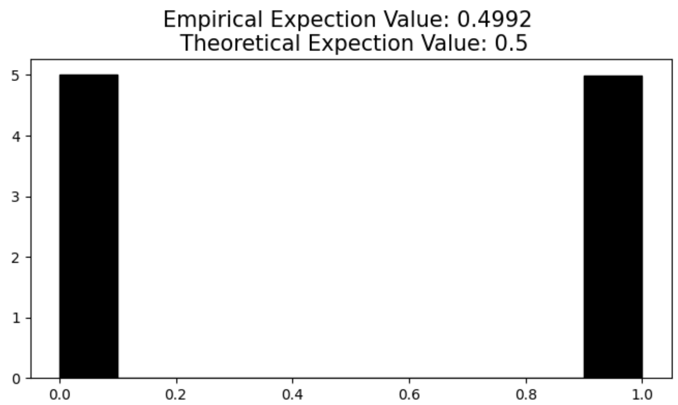

Figure 2.1: Inverse Transform method for generating Bernoulli random numbers with $p = 0.5$

Example 2.1.2 (Binomial Distribution). Let, $X \sim Bin(n, p)$ then, X has probability mass function

$$f(r) = P(X = r) = \binom{n}{r} p^r (1 - p)^{n-r}, \ i = 1, 2, \ldots$$

The generation of $X \sim Bin(n, p)$ by Inverse Transform Algorithm can be tedious. We can use the relation between Binomial and Bernoulli distribution. If $x_i \sim Ber(p), \forall i = 1, 2, \ldots, n$ then, $\sum_{i=1}^{n} x_i \sim Bin(n, p)$.

Hence, by generating x_i n independent random variable from Bernoulli distribution and summing them we get binomial distribution

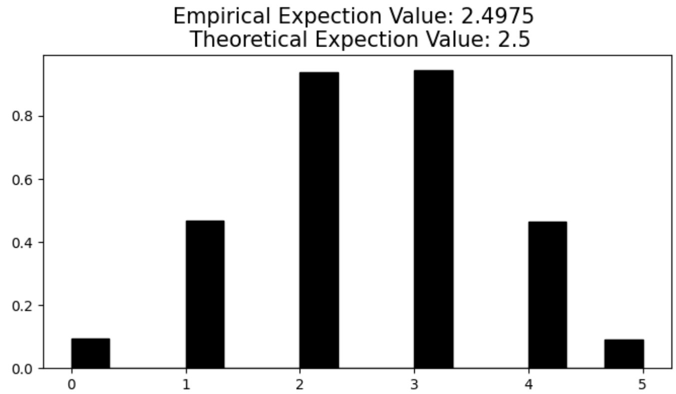

Figure 2.2: Generating binomial random numbers with $n = 5$ and $p = 0.5$

2.2 Generating Continuous Random Variables

2.2.1 The Inverse Transform Algorithm

To generate Continuous random variables The Inverse Transform Algorithm is very important method. It is based on a following theorem.

Theorem 2.2.1. *Let U be a uniform $(0,1)$ random variable. For any continuous distribution function F the random variable X defined by*

$$X = F^{-1}(U)$$

has distribution F.

Proof. Let, F_X denote the distribution function of $X = F^{-1}(U)$. Then,

$$\begin{aligned} F_X(x) &= P(X \leq x) \\ &= P(F^{-1}(U) \leq x) \end{aligned}$$

Since, F is a cumulative distribution function it follows that $F(x)$ is monotonic increasing function of x and range of $F(x)$ is $(0,1)$. Then,

$$\begin{aligned} F_X(x) &= P\left(F\left(F^{-1}(U)\right) \leq F(x)\right) \\ &= P(U \leq F(x)) \\ &= F(x) \text{ since } U \sim U(0,1) \end{aligned}$$

$\square$

The above theory tells us we can generate a random variable X from the continuous distribution function F by generating a random number $U \sim U(0,1)$ and setting $X = F^{-1}(U)$.

Example 2.2.1 (Exponentian Distribution). Suppose we want to generate a random variable $x \sim Exp(\lambda)$, then its probability density function is

$$f(x) = \lambda e^{-\lambda x}.$$

Hence, The cumulative distribution function is,

$$F(x) = 1 - e^{\lambda x}$$

if we let $x = F^{-1}(u)$, then,

$$\begin{aligned} u &= F(x) = 1 - e^{-\lambda x} \\ 1 - u &= e^{-\lambda x} \\ x &= -\frac{\ln(1-u)}{\lambda} \end{aligned}$$

Hence, we can generate an exponential random variable with parameter 1 by generating a uniform $(0, 1)$ random number U and then setting

$$X = F^{-1}(U) = -\frac{\ln(1-U)}{\lambda}.$$

We see that if $U \sim U(0, 1)$ then also $1 - U \sim U(0, 1)$ thus $\ln(1 - U)$ has the same distribution as $\ln(U)$ so,

$$X = F^{-1}(U) = -\frac{\ln(U)}{\lambda}.$$

will also work. If we use second expression then the algorithm will take less computing power hence less time.

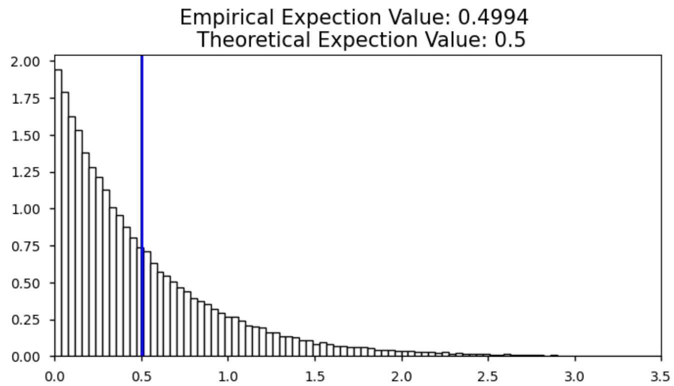

Figure 2.3: Inverse Transform method for generating $Exp(2)$

Example 2.2.2 (Gamma Distribution). Let $X \sim G(n, \lambda)$ Then, its probability mass function is given by,

$$f(x) = \frac{1}{\Gamma(n)} \lambda^n x^{n-1} e^{-\lambda x}$$

We know if $X_i \sim Exp(\lambda) \forall i = 1, 2, \ldots, n$ then $Y = \sum_i X_i \sim G(n, \lambda)$. As,

$$\begin{aligned} M_Y(t) = E\left[e^{tY}\right] = E\left[e^{\sum_{i=1}^{n} X_i t}\right] &= E\left[\prod_{i=1}^{n} e^{X_i t}\right] \\ &= \prod_{i=1}^{n} E\left[e^{X_i t}\right] \text{ As all } X_i \text{ are independent} \\ &= \prod_{i=1}^{n} \frac{\lambda}{\lambda - t} = \left(\frac{\lambda}{\lambda - t}\right)^n \end{aligned}$$

Then, Generating n number of $X_i \sim Exp(\lambda)$ and summing them we can easily generate a random variable which follows gamma distribution

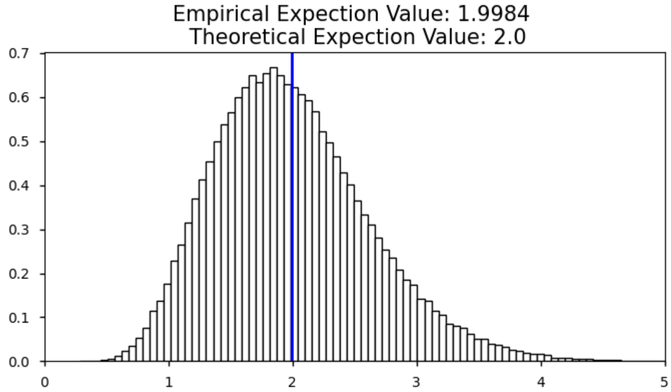

Figure 2.4: $G(10, 5)$ generated by summing of $Exp(5)$

2.2.2 Accept - Reject Method

The accept–reject method is useful when it is difficult to directly simulate $f(x)$ but we can generate another density $g(x)$ such that $f(x)/g(x)$ is uniformly bounded and it is much easier to simulate $g(x)$. We simulate X from g, and retain it or toss it according to a probability proportional to $f(x)/g(x)$. Because an X value is either retained or discarded, depending on whether it passes the admission rule, the method is called the accept–reject method. The density $g(x)$ is called the envelope density.

The method proceeds as follows,

STEP 1: Find a density g and a finite constant c such that $\frac{f(x)}{g(x)} \leq c \ \forall x$.
STEP 2: Generate $X \sim g$.
STEP 3: Generate $U \sim U(0, 1)$, independent of X.
STEP 4: Retain this generated value X if $U \leq \frac{f(x)}{cg(x)}$.
STEP 5: Repeat the same until the required number of n values of X has been obtained.

The following theorem supports the method.

Theorem 2.2.2. *Let $X \sim g$, and U, independent of, be a distributed as $U[0, 1]$. Then the conditional density of X given that $U \leq \frac{f(X)}{cg(X)}$ is f.*

Proof. Denote the CDF of f by F. Then,

$$\begin{aligned} P\left(X \leq x | U \leq \frac{f(X)}{cg(X)}\right) &= \frac{P\left(X \leq x, U \leq \frac{f(X)}{cg(X)}\right)}{P\left(U \leq \frac{f(x)}{cg(x)}\right)} \\ &= \frac{\int_{-\infty}^{x} \int_{0}^{\frac{f(t)}{cg(t)}} g(t) du dt}{\int_{-\infty}^{\infty} \int_{0}^{\frac{f(t)}{cg(t)}} g(t) du dt} \\ &= \frac{\int_{-\infty}^{x} f(t) dt}{\int_{-\infty}^{\infty} f(t) dt} = \frac{F(x)}{1} = F(x). \end{aligned}$$

□

Example 2.2.3 (Generating a Normal Random Variable)*.* To generate a standard normal variable Z i.e. $Z \sim N(0, 1)$, note first that the absolute value of Z has probability density function

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \ 0 \leq x \leq \infty. \tag{2.2}$$

Then, we can choose g as the exponential density function with mean 1 i.e.

$$g(x) = e^{-x} \ 0 \leq x \leq \infty$$

Now,

$$\frac{f(x)}{g(x)} = \sqrt{\frac{2}{\pi}} e^{x - \frac{x^2}{2}}$$

and so the maximum value of $f(x)/g(x)$ occurs at the value of x that maximize $x - x^2/2$ hence $x = 1$ so we take

$$c = \max_x \frac{f(x)}{g(x)} = \frac{f(1)}{g(1)} = \sqrt{\frac{2e}{\pi}}.$$

Now,

$$\frac{f(x)}{cg(x)} = \exp\left(x - \frac{x^2}{2} - \frac{1}{2}\right) = \exp\left(\frac{-(x-1)^2}{2}\right)$$

Then, its follows that we can generate the absolute value of a standard normal random variable as follows:
STEP 1: Generate $X \sim Exp(1)$.
STEP 2: Generate $U \sim U(0,1)$, independent of X.
STEP 3: If $U \leq \exp\left(-(X-1)^2/2\right)$, retain X, Otherwise, return to Step 1.

Once, we have simulated a random variable X having density function as in Equation (2.2) we can obtain a standard normal Z by letting Z be equally likely to be either X or $-X$. In Step 3, the value X is accepted if $U \leq \exp\left(-(X-1)^2/2\right)$, which is equivalent to $-\ln U \geq (X-1)^2/2$. However, in Example (2.2.1) we have seen that $-\ln U \sim Exp(1)$ When $U \sim U(0,1)$.

So, summing up, we can generate the standard normal random variable Z as follows:
STEP 1: Generate independent $X_1, X_2 \sim Exp(1)$
STEP 2: If $X_2 \geq (X_1 - 1)^2/2$ retain X_1. Otherwise, return to Step 1.
STEP 3: Generate $U \sim U(0,1)$ and set,

$$Z = \begin{cases} X_1 \text{ if } U \leq \frac{1}{2}, \\ X_1 \text{ if } U > \frac{1}{2}. \end{cases}$$

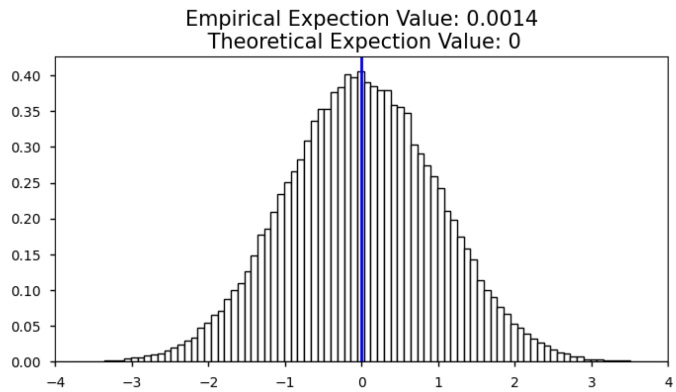

Figure 2.5: Generating $N(0,1)$ with Accept - Reject method

If we want to generate normal random variable to have mean μ and variance σ^2, just take $\mu + \sigma Z$.

Example 2.2.4 (Generating Beta Distribution). If α and β are both getter then 1, then Beta density is uniformly bounded and its maximum attain at $\frac{\alpha-1}{\alpha+\beta-2}$. As a result the $U[0,1]$ density can be served as an envelope density for generating such Beta distribution by using accept-reject method. Precisely, generate U, $X \sim U[0,1]$ (independently), and retain the value if $U \leq \frac{f(X)}{\sup_x f(X)}$, where,

$$f(X) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, 0 < x < 1.$$

Because

$$\sup_x f(X) = f\left(\frac{\alpha-1}{\alpha+\beta-2}\right) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{(\alpha-1)^{\alpha-1}(\beta-1)^{\beta-1}}{(\alpha+\beta-2)^{\alpha+\beta-2}}$$

The algorithm finally works out as follows:
STEP 1: Generate independent $U, X \sim U[0,1]$.
STEP 2: Retain the value X if,

$$U \leq \frac{X^{\alpha-1}(1-X)^{\beta-1}(\alpha+\beta-2)^{\alpha+\beta-2}}{(\alpha-1)^{\alpha-1}(\beta-1)^{\beta-1}}.$$

Otherwise, return to STEP 1.

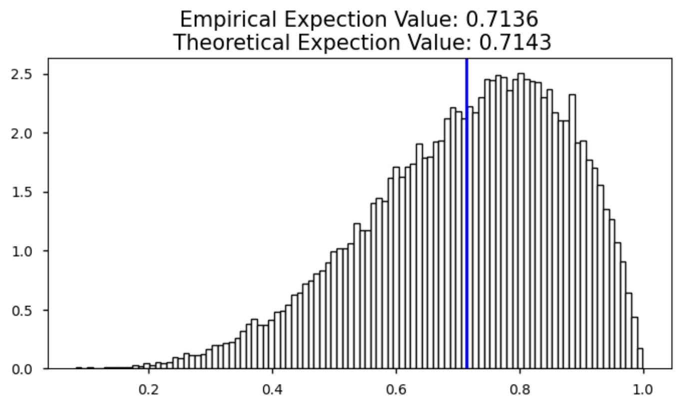

Figure 2.6: Generating Beta(5,2) with accept-reject method

An issue about an accept-reject method is the acceptance rate. Our goal make it as large as possible to increase the efficiency of the method. This can be achieved by choosing c to be smallest possible number, described in the result bellow.

Theorem 2.2.3 (Acceptance Rate)**.** *For an accept-reject scheme, the probability that an $X \sim g$ is acceded is $\frac{1}{c}$, and is maximized when c is chosen to be $c = \sup_x \frac{f(x)}{g(x)}$.*

Proof.

$$\begin{aligned} P\left(U <= \frac{f(x)}{cg(x)}\right) &= \int_{-\infty}^{\infty} \int_0^{\frac{f(x)}{cg(x)}} g(t) du dt \\ &= \int_{-\infty}^{\infty} \frac{f(t)}{cg(t)} g(t) dt = \int_{-\infty}^{\infty} \frac{f(t)}{c} dt = \frac{1}{c}. \end{aligned}$$

Because any c that can be chosen must be at least as large as $\sup_x \frac{f(x)}{g(x)}$, obviously $1/c$ is maximized by choosing $c = \sup_x \frac{f(x)}{g(x)}$. $\square$

In the example (2.2.3) for $N(0,1)$ the acceptance rate is $\sqrt{\frac{\pi}{2e}} = 0.7601$. And for example (2.2.4) for Beta(5,2) the acceptance rate is 0.4069

2.3 The Polar Method for Generating Normal Random Variables

Let X and Y be independent slandered normal random variable and let R and θ denote the polar coordinates of vector (X, Y). That is,

$$R^2 = X^2 + Y^2$$
$$\tan\theta = \frac{Y}{X}$$

Since X and Y are independent, their joint density is the product of their individual densities and thus given by

$$\begin{aligned} f(x,y) &= \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \\ &= \frac{1}{\sqrt{2\pi}} e^{-(x^2+y^2)/2} \end{aligned}$$

To determine the joint density of R^2 and Θ - call it $f(d,\theta)$ we make the change of variables

$$d = x^2 + y^2, \quad \theta = \tan^{-1}\left(\frac{y}{x}\right)$$

Then the joint density function of R^2 and Θ is,

$$f(d,\theta) = \frac{1}{2}\frac{1}{2\pi} e^{-d/2}, \quad 0 < d < \infty, 0 < \theta < 2\pi. \tag{2.3}$$

As $f(d,\theta)$ is equal to product of the product of $Exp(1/2)$ density and $U(0, 2\pi)$, it follows that, R^2 and Θ are independent, with $R^2 \sim Exp(1/2)$ and $\Theta \sim U(0, 2\pi)$

Hence to generate a pair of independent slandered normal random variables X and Y by generating R^2 and Θ in polar coordinates and then transform back to rectangular coordinates. Hence the algorithm is:

STEP 1: Generate random number $U_1, U_2 \sim U(0, 1)$.

STEP 2: $R^2 = -2\ln U_1$ and $\Theta = 2\pi U_2$.

STEP 3: Now let,

$$\begin{aligned} X &= R\cos\Theta = \sqrt{-2\ln U_1}\cos(2\pi U_2) \\ Y &= R\sin\Theta = \sqrt{-2\ln U_1}\sin(2\pi U_2). \end{aligned} \tag{2.4}$$

The transformation given by equitations (2.4) are known as Box-Muller transformation.

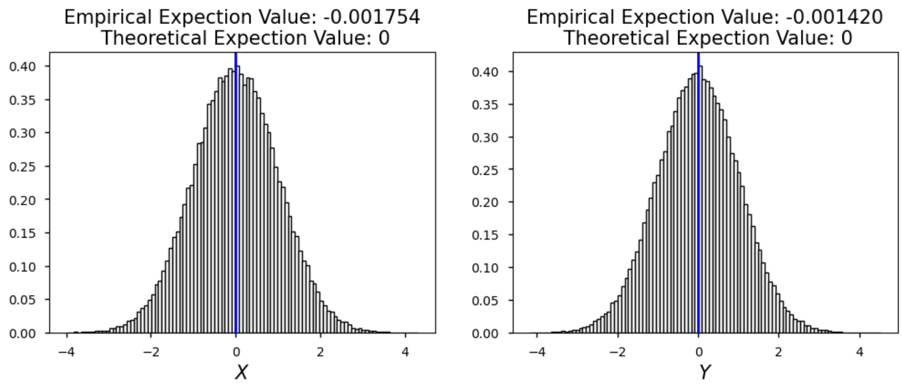

Figure 2.7: Generating independent $X, Y \sim N(0, 1)$ with polar method