MCMC

Azmain Biswas

April 2023

Contents

Chapter 1

Introduction

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

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

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 $0, 1, \ldots, m-1$ 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

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

this method is known as Mixed Congruential Generators 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

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

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.

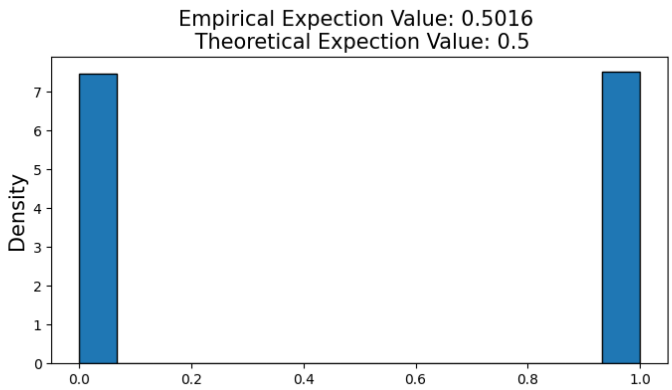

Figure 2.1: Inverse Transform plot for generating Bernoulli random numbers with $p = 0.5$

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

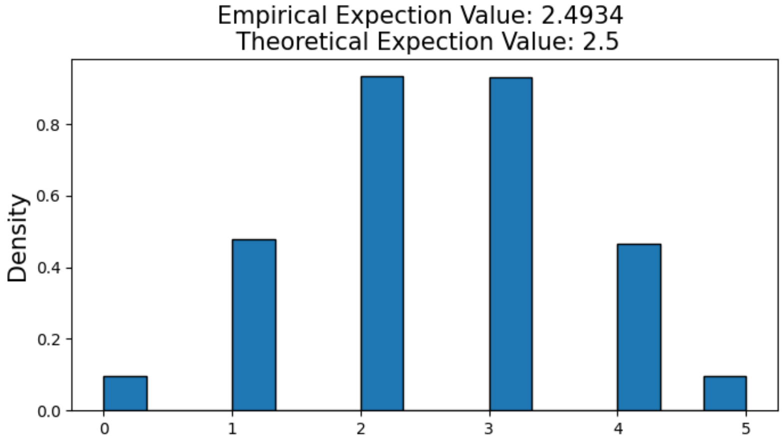

Figure 2.2: Generation of binomial random numbers with $n = 5$ and $p = 0.5$

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,

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

□

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.

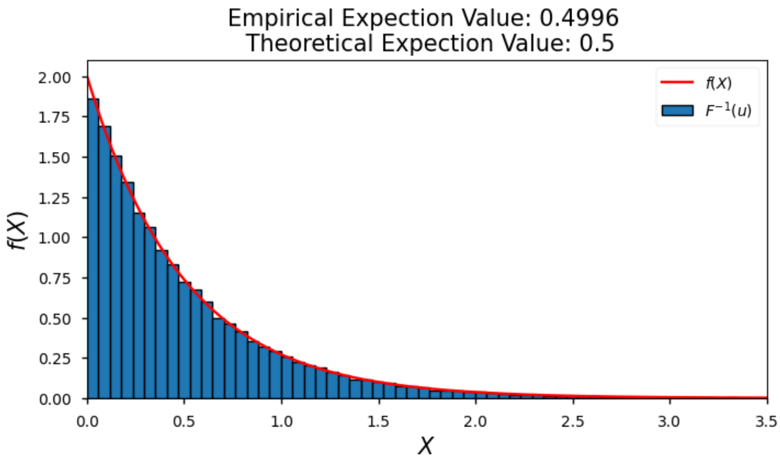

Figure 2.3: Inverse Transform plot for generating $Exp(2)$

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

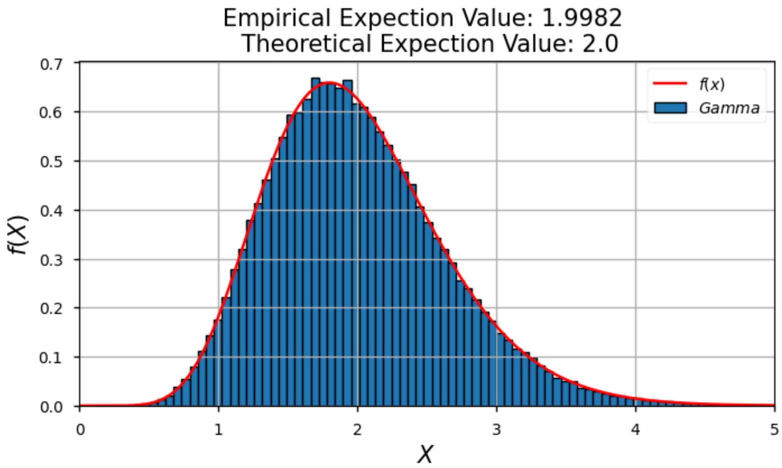

Figure 2.4: $G(10,5)$ generation by sum of $Exp(5)$