A M.Sc Thesis

on

Simulation and Markov Chain Monte Carlo

by

Azmain Biswas

Enrollment No.: 2022MAM008

under the supervision of

Prof. M. Mitra

1856 १८५६

उत्तिष्ठत जाग्रत प्राप्य वरान् निबोधत

INDIAN INSTITUTE OF ENGINEERING SCIENCE AND TECHNOLOGY, SHIBPUR

भारतीय अभियांत्रिकी विज्ञान एवं प्रौद्योगिकी संस्थान, शिवपुर

Indian Institute of Engineering Science and Technology, Shibpur

Department of Mathematics

Submitted for the partial fulfillment of the requirements for the award of the degree of

M.Sc. in Applied Mathematics

CERTIFICATE

This is to certify that the M.Sc Thesis on **"Simulation and Markov Chains Monte Carlo"** submitted by Azmain Biswas to the Department of Mathematics, Indian Institute of Engineering Science and Technology, Shibpur, Howrah-711103, for final semester of Master of Science in Applied Mathematics is a record of project work carried out by him under my supervision.

The result of this project work or any part thereof has not been submitted for any degree or diploma.

Prof. M. Mitra
Department of Mathematics
Indian Institute of Engineering Science and Technology
Shibpur, Howrah.

Acknowledgment

Thanks to my supervisor, Prof. M. Mitra, for a very interesting project. It has been a great pleasure to work under his guidance, and I have learned a lot from his way of thinking and his approach to a problem.

I am also indebted to Prof. Pritha Das, Head of the Department of Mathematics, Indian Institute of Engineering Science and Technology, Shibpur, for her help.

At various stages, I received encouragement and inspiration from most of the faculty members of the department. I would like to thank all of them for their invaluable words of advice.

I have benefited greatly from the advantages of the TEX-group software typing systems, the use of which has contributed much to simplify the typing of long mathematical expressions involved in my work.

Additionally, I would like to thank the Python Software Foundation for their software, without which I could not have finished this project.

I would like to thank my parents and friends, for being so supportive of my career and life choices, as well as for making me happy.

Azmain Biswas

Contents

List of Figures

List of Tables

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 is a computer-based exploratory exercise that aids in understanding how the behavior of a random or even a deterministic process changes in response to changes in input or the environment. It is essentially the only option left when exact mathematical calculations are impossible, or require an amount of effort that the user is not willing to invest. Even when the mathematical calculations are quite doable, a preliminary simulation can be very helpful in guiding the researcher to theorems that were not a priori obvious or conjectured, and also to identify the more productive corners of a particular problem. Although simulation in itself is a machine-based exercise, credible simulation must be based on appropriate theory. A simulation algorithm must be theoretically justified before we use it.

The classic theory of simulation includes such time-tested methods as the original Monte Carlo, Inverse Transform method, Accept-Reject method, Bivariate techniques from standard distributions in common use. They involve a varied degree of sophistication. Markov chain Monte Carlo is the name for a collection of simulation algorithms for simulating from the distribution of very general types of random variables taking values in quite general spaces. MCMC methods have truly revolutionized simulation because of an inherent simplicity in applying them, the generality of their scopes, and the diversity of applied problems in which some suitable form of MCMC has helped in making useful practical advances. MCMC methods are the most useful when conventional Monte Carlo is difficult or impossible to use.

Simulation depend on various theoretical aspect such as The weak law of Large Number, The Central limit theory, The sample mean and sample variance etcetera.

There are various type of simulation technique in standard simulation theory such as,

1. The Inverse Transform Method
2. Accept-Reject Algorithm
3. Bivariate Techniques
4. Ordinary Monte Carlo
5. Importance Sampling
6. Markov Chain Monte Carlo

This project work focus mainly on these simulation techniques.

In Chapter 2, we discuss how to generate random variable both uniform and continuous, by using method like The Inverse Transform Method, Accept-Reject Algorithm, Bivariate Techniques.

In Chapter 3, we focus on Ordinary Monte Carlo and how to use it to solve problem like evaluating integration and evaluating the value of π. Then, the focus shifts to Importance Sampling and how it is beneficial from Ordinary Monte Carlo by some example. Learn about how to choose optimal Importance sample distribution.

In Chapter 4, we know different Markov Chain Monte Carlo method like Metropolis-Hastings Algorithm and Gibbs Sampler and discuss how to observe sample is reached stationary.

1.1 Mathematical Preliminaries

Definition 1.1.1 (Probability Space). *A probability specs is a triple $(\Omega, \mathcal{F}, P)$ consisting of:*

(a) the sample space Ω (an arbitrary non-empty set)

(b) a non-empty collection of subsets $\mathcal{F}$ of Ω, called sigma field of subspace of Ω, such that,

(i) $\Omega \in \mathcal{F}$

(ii) if $A \in \mathcal{F}$, then $A^c \in \mathcal{F}$

(iii) if $A_n \in \mathcal{F}$, $n = 1, 2, \ldots,$ then $\cup_n = 1^\infty A_n \in \mathcal{F}$

(c) a probability measure $p : \mathcal{F} \to [0, 1]$, which is a real valued function on $\mathcal{F}$ such that,

(i) $P(A) \geq 0$ for all $A \in \mathcal{F}$

(ii) $P(\Omega) = 1$

(iii) if $A_1, A_2, \ldots$ are disjoint sets in $\mathcal{F}$, then $P(\cup_n A_n) = \sum_n P(A_n)$.

Definition 1.1.2 (Conditional Probability). *Let A, B be general events with respect to some sample space Ω, and suppose $P(A) > 0$. The conditional probability of B given A is defined as*

$$P(B|A) = \frac{P(A \cap B)}{P(A)}.$$

Theorem 1.1.1 (Bayes's Theorem). *Let, $\{A_1, A_2, \ldots, A_n\}$ be a partition of sample space Ω. Let B be a some fixed event. Then*

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{\sum_{i=1}^{n} P(B|A_i)P(A_i)}.$$

Definition 1.1.3 (Random Variable). *Let, Ω be a sample space corresponding to some experiment and let $X : \Omega \to \mathbb{R}$ be a function from the sample space to the real line. Then X is called a random Variable*

Definition 1.1.4 (Cumulative Distribution Function). *The cumulative distribution function(CDF) or simply distributed function, F of a random variable X is defined for any real number x by*

$$F(x) = P(X \leq x).$$

Definition 1.1.5 (Probability Mass Function). *For a discrete random variable X we define its Probability mass function(pmf) $p(x)$ by*

$$p(x) = P(X = x)$$

and we have,

$$\sum_{i \in \Lambda} p(x_i) = 1. \text{ and } p_i \geq 0.$$

Definition 1.1.6 (Probability Density Function). *For a continuous random variable if there is a non-negative function $f(x)$ defined for all real number x and having the property that for any set $C \subset \mathbb{R}$,*

$$P(X \in C) = \int_C f(x)dx.$$

The function f is called probability density function(pdf) of the random variable X.

The relation between CDF and pdf is express by,

$$F(a) = P(X \in (-\infty, a]) = \int_{-\infty}^{a} f(x)dx.$$

Definition 1.1.7 (Expectation). *If X is a random variable, then the exception or expected value of X, also called the mean of X and denoted by $E(X)$, is defined by*

$$E(X) = \int xdF(x)$$

Definition 1.1.8 (Conditional Expectation). *If X and Y are jointly discrete random variables, we define $E(X|Y = y)$ by,*

$$\begin{aligned} E(X|Y = y) &= \sum_x xP(X = x|Y = y) \\ &= \frac{\sum_x xP(X = x, Y = y)}{P(Y = y)} \end{aligned}$$

Similarly, if X and Y are jointly continuous r.v. with joint distribution function $f(x, y)$, then $E(X|Y)$ given by,

$$\begin{aligned} E(X|Y = y) &= \int_x xf_{X|Y}(x|y)dx \\ &= \frac{1}{f_Y(y)} \int_x f_{X,Y}(x, y)dx \end{aligned}$$

when, the denominator is zero, the expression is undefined.

We have, $E\left(E(X|Y)\right) = E(X)$.

Definition 1.1.9 (Variance). *If X is a random variable with mean $E(X)$, then the variance of X, denoted by $Var(X)$, is defined by,*

$$Var(X) = E\left((X - E(X))^2\right)$$

Now,

$$\begin{aligned}\mathrm{Var}(X) &= E\left((X-E(X))^2\right)\\ &= E\left(X^2-2XE(X)+(E(X))^2\right)\\ &= E\left(X^2\right)-2(E(X))^2+E(X)^2\\ &= E\left(X^2\right)-(E(X))^2\end{aligned}$$

We can also define the **conditional variance formula** by,

$$\mathrm{Var}(X) = E(\mathrm{Var}(X|Y)) + \mathrm{Var}(E(X|Y)).$$

Theorem 1.1.2 (The Weak Law of Large Numbers). *Let $X_1, X_2, \ldots$ be a sequence of in dependent and identically distributed random variables having mean μ, Then, for any $\epsilon > 0$,*

$$P\left(\left|\frac{X_1+X_2+\ldots+X_n}{n}-\mu\right|>\epsilon\right)\to 0$$

Theorem 1.1.3 (The Central Limit Theorem). *Suppose $X_1, X_2, \ldots$ is a sequence of i.i.d random variables with $E[X_i] = \mu$ and $Var[X_i] = \sigma^2 < \infty$. Then,*

$$\lim_{n\to\infty} P\left(\frac{X_1+\ldots+X_n-n\mu}{\sigma\sqrt{n}} < n\right) = \Phi(Z)$$

Were, $\Phi(Z)$ denote the distribution function of a standard normal distribution.

Definition 1.1.10 (Estimators). *Suppose now that we have an unknown real parameter θ taking values in a parameter space $T \subset \mathbb{R}$. A real-valued statistic $U = u(X)$ that is used to estimate θ is called, appropriately enough, an estimator of θ.*

When we actually run the experiment and observe the data x, the observed value $u = u(x)$ (a single number) is the estimate of the parameter θ.

Suppose, $X_1, X_2, \ldots, X_n$ are random variables form $F(x;\theta)$. To estimate a parameter function $\Psi(\theta)$ consider some some estimator $T(X_1, X_2, \ldots, X_n)$. Now **Mean Squared Error**(MSE) of the estimator $\Psi(\theta)$ is defined by,

$$\mathrm{MSE}(T) = E_\theta(T(X_1, X_2, \ldots, X_n) - \Psi(\theta))^2 \tag{1.1}$$

Now,

$$\begin{aligned}\mathrm{MSE}(T) &= E_\theta\left(T-\Psi(\theta)\right)^2\\ &= E_\theta\left(T-E_\theta(T)+E_\theta(T)-\Psi(\theta)\right)^2\\ &= \mathrm{Var}_\theta(T)+E_\theta(E_\theta(T)-\Psi(\theta))^2\\ &= \text{Variance of } T + (\text{Biased}(T))^2\end{aligned}$$

For, unbiased estimator the term Biased(T) is zero. Hence, for unbiased estimator variance is MSE.

Sample Mean and Sample Variance

Suppose that $X_1, X_2, \ldots, X_n$ are independent random variable having the same distribution function. Let μ and σ^2 denote, respectively their mean and variance that is,

$\mu = E(X_i)$ and $\sigma^2 = Var(X_i)$. The quantity

$$\bar{X} \equiv \sum_{i=1}^{n} \frac{X_i}{n}$$

which is the arithmetic average of the n data values, is called the *sample mean.* When the population mean μ is unknown, the sample mean is often used to estimate μ.

Because,

$$\begin{aligned} E(\bar{X}) &= E\left(\sum_{i=1}^{n} \frac{X_i}{n}\right) \\ &= \sum_{i=1}^{n} \frac{E(X_i)}{n} \\ &= \frac{n\mu}{n} = \mu \end{aligned}$$

If follows that $\bar{X}$ is an unbiased estimator of μ, where we say that an estimator of parameter is an unbiased estimator of that an estimator of a parameter is an unbiased estimator of that parameter if its expected value is equal to the parameter.

The quantity S^2, define by

$$S^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n-1}$$

is called the *sample variance.* Now,

$$\begin{aligned} E\left(S^2\right) &= \frac{1}{n-1} E\left(\sum_{i=1}^{n}(X_i - \bar{X})^2\right) \\ &= \frac{1}{n-1} E\left(\sum_{i=1}^{n} X_i^2 - 2\bar{X}\sum_{i=n}^{n} X_i + n\bar{X}^2\right) \\ &= \frac{1}{n-1} E\left(\sum_{i=1}^{n} X_i^2\right) - \frac{n}{n-1} E\left(\bar{X}^2\right) \\ &= \frac{1}{n-1} \sum_{i=1}^{n} E\left(X_i^2\right) - \frac{n}{n-1} E\left(\bar{X}^2\right) \end{aligned}$$

Now,

$$\begin{aligned} \text{Var}(X_i) &= E\left(X_i^2\right) - \left(E(X_i)\right)^2 \\ \text{or, } \sigma^2 &= E(X_i^2) - \mu^2 \\ \text{or, } E(X_i^2) &= \sigma^2 + \mu^2 \end{aligned}$$

And,

$$\begin{aligned}
\text{Var}(\bar{X}) &= E\left(\bar{X}^2\right) - \left(E(\bar{X})\right)^2 \\
\text{or, } E(\bar{X}^2) &= \text{Var}\left(\frac{X_1 + X_2 + \ldots + X_n}{n}\right) + \mu^2 \\
&= \frac{1}{n^2} n \text{Var}(X_i) + \mu^2 \text{ baecuse, all } X_i \text{ are IID} \\
&= \frac{1}{n}\sigma^2 + \mu^2
\end{aligned}$$

Now,

$$\begin{aligned}
E\left(S^2\right) &= \frac{1}{n-1} n\left(\sigma^2 + \mu^2\right) - \frac{n}{n-1}\left(\frac{1}{n}\sigma^2 + \mu^2\right) \\
&= \left(\frac{n}{n-1} - \frac{1}{n-1}\right)\sigma^2 \\
&= \frac{n-1}{n-1}\sigma^2 = \sigma^2.
\end{aligned}$$

Hence,S^2 is unbiased of σ^2

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 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

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

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

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

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 (Bernoulli Distribution). Let, $X \sim \text{Ber}(p)$ 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 $U \sim \text{U}[0,1]$ 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 $U \sim \text{U}[0,1]$.
STEP 2: If $U \leq p$ set $X = 1$ or set $X = 0$.
STEP 3: Go to STEP 1.

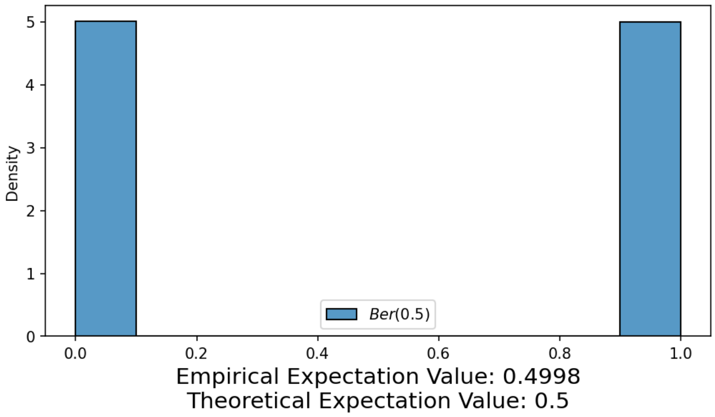

Figure 2.1: Inverse Transform method for generating Bernoulli random numbers with $p = 0.5$

Example 2.2 (Binomial Distribution). Let, $X \sim \text{Bin}(n,p)$ 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 $X \sim \text{Bin}(n,p)$ by Inverse Transform Algorithm can be tedious. We can use the relation between Binomial and Bernoulli distribution. If $x_i \sim \mathrm{Ber}(p), \forall i = 1, 2, \ldots, n$ then, $\sum_{i=1}^{n} x_i \sim \mathrm{Bin}(n, p)$.

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

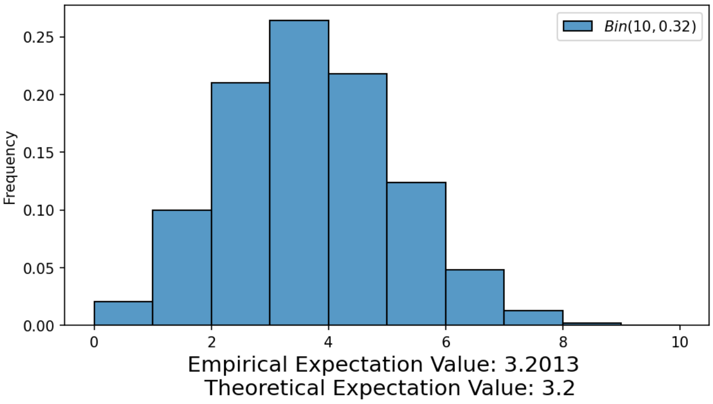

Figure 2.2: Generating binomial random numbers with $n = 10$ and $p = 0.32$

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,

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

□

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

Example 2.3 (Exponential Distribution). Suppose we want to generate a random variable $x \sim \text{Exp}(\lambda)$, 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 $U \sim \text{U}(0, 1)$ then also $1 - U \sim \text{U}(0, 1)$ 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.

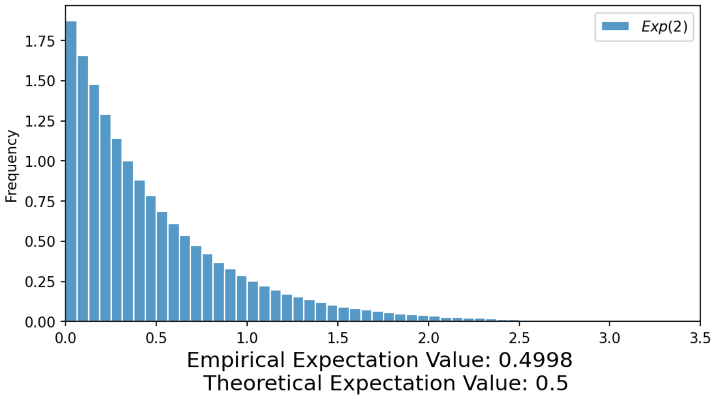

Figure 2.3: Inverse Transform method for generating Exp(2)

Example 2.4 (Gamma Distribution). Let $X \sim \text{G}(n, \lambda)$ 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 $X_i \sim \text{Exp}(\lambda) \forall i = 1, 2, \ldots, n$ then $Y = \sum_i X_i \sim \text{G}(n, \lambda)$. 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 $X_i \sim \text{Exp}(\lambda)$ and summing them we can easily generate a random variable which follows gamma distribution

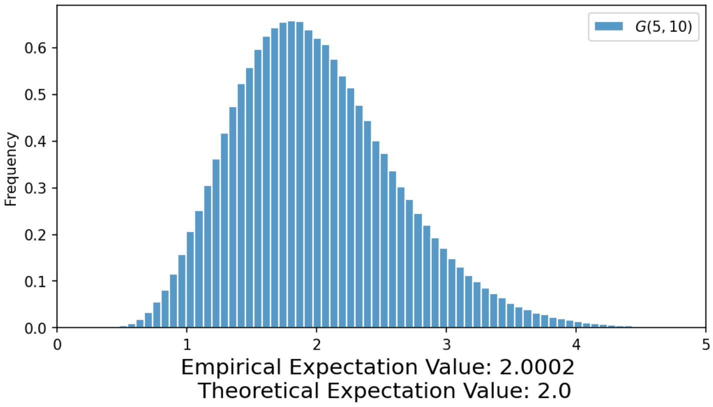

Figure 2.4: G(10, 5) generated by summing of Exp(5)

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 $U \sim \text{U}(0, 1)$, 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,

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

□

Example 2.5 (Generating a Normal Random Variable). To generate a standard normal variable Z i.e. $Z \sim \mathrm{N}(0,1)$, note first that the absolute value of Z has probability density function

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

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

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

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 $X \sim \mathrm{Exp}(1)$.
STEP 2: Generate $U \sim \mathrm{U}(0,1)$, 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 2.3 we have seen that $-\ln U \sim \mathrm{Exp}(1)$ When $U \sim \mathrm{U}(0,1)$.

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

STEP 3: Generate $U \sim \mathrm{U}(0,1)$ and set,

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

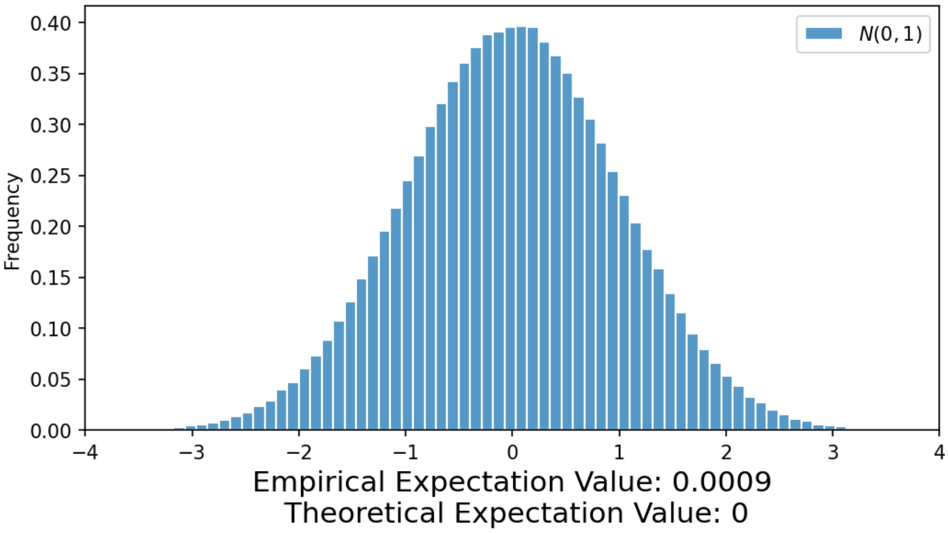

Figure 2.5: Generating $\mathrm{N}(0,1)$ with Accept - Reject method

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

Example 2.6 (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 $\mathrm{U}[0,1]$ density can be served as an envelope density for generating such Beta distribution by using accept-reject method. Precisely, generate U, $X \sim \mathrm{U}[0,1]$ (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 $U, X \sim \mathrm{U}[0,1]$.
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.

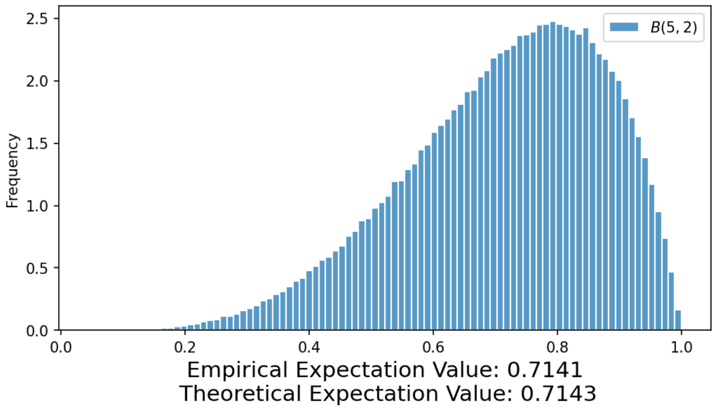

Figure 2.6: Generating Beta(5,2) with accept-reject method

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.

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

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 $c = \sup_x \frac{f(x)}{g(x)}$. □

In the Example 2.5 for N$(0, 1)$ the acceptance rate is $\sqrt{\frac{\pi}{2e}} = 0.7601$. And in the Example 2.6 for Beta(5,2) the acceptance rate is 0.4069

2.2.3 Bivariate Techniques

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,

$$
\begin{aligned}
R^2 &= X^2 + Y^2 \\
\tan\theta &= \frac{Y}{X}
\end{aligned}
$$

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

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

To determine the joint density of R^2 and Θ - call it $g(d,\theta)$ 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 d and Θ is,

$$\begin{aligned} g(d,\theta) &= |J| f(x,y) \\ &= |J| \frac{1}{2\pi} e^{-(x^2+y^2)/2} \end{aligned} \tag{2.3}$$

where,

$$J = \begin{vmatrix} \frac{\partial x}{\partial t} & \frac{\partial x}{\partial \theta} \\ \frac{\partial y}{\partial t} & \frac{\partial y}{\partial \theta} \end{vmatrix} = \frac{1}{2}.$$

Then, replacing the value of $x = \sqrt{d}\sin(\theta)$ and $y = \sqrt{d}\cos(\theta)$ in Equation (2.3) we get,

$$g(d,\theta) = \frac{1}{2}\frac{1}{2\pi}e^{-d/2}, \quad 0 < d < \infty, 0 < \theta < 2\pi. \tag{2.4}$$

As $g(d,\theta)$ is equal to product of the product of Exp(1/2) density and $\mathrm{U}(0,2\pi)$, it follows that,
R^2 and Θ are independent, with $R^2 \sim \mathrm{Exp}(1/2)$ and $\Theta \sim \mathrm{U}(0,2\pi)$
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 $U_1, U_2 \sim \mathrm{U}(0,1)$.
STEP 2: $R^2 = -2\ln U_1$ and $\Theta = 2\pi U_2$.
STEP 3: Now let,

$$X = R\cos\Theta = \sqrt{-2\ln U_1}\cos(2\pi U_2) \tag{2.5}$$

$$Y = R\sin\Theta = \sqrt{-2\ln U_1}\sin(2\pi U_2). \tag{2.6}$$

The transformation given by Equation (2.5) and Equation (2.6) are known as Box-Muller transformation.

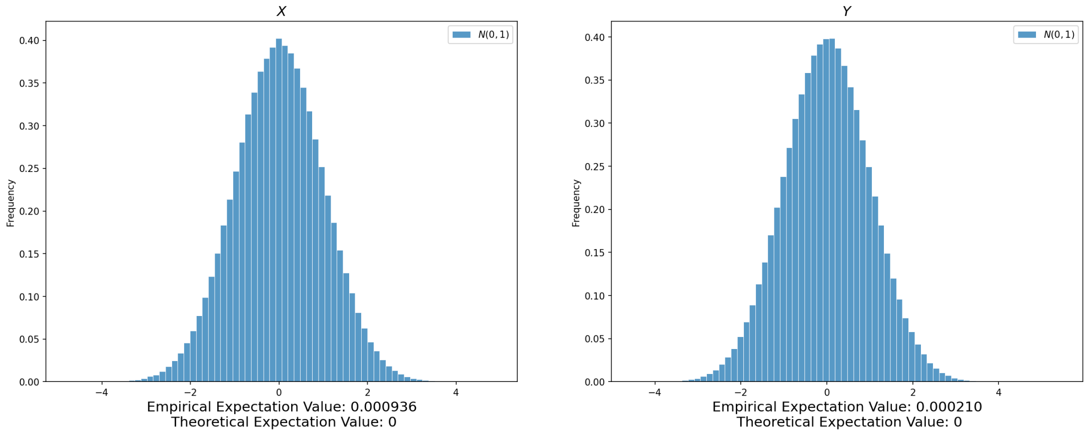

Figure 2.7: Generating independent $X, Y \sim \mathrm{N}(0,1)$ with polar method

Chapter 3

Monte Carlo Method

3.1 Ordinary Monte Carlo Simulation

Polish-American mathematician Stanislaw Ulam, recovering from an illness, was playing a lot of solitary card game. He wanted to calculate the probability of winning and quickly it is impossible to calculate analytically. Then he thought about playing lots of hands counting number of wins, but decided it will take years. After falling several times he asked Von Neumann to build a program to simulate solitary card game in ENIAC. Then Around 1940 they used Monte Carlo simulation in Manhattan Project in which physicists wanted to understand how the physical properties of neutrons would be affected by various possible scenarios following a collision with a nucleus.

The basis for Monte Carlo is Law of Large Number. If we simulate large number $X_1, X_2, \ldots$ IID copes of random variable X Then we can approximate the true value $E(f(X))$ by simple mean $\frac{1}{n}\sum_{i=1}^{n} f(X_i)$. Here in Monte Carlo Random Sampling play the key factor for good estimation of $E(f(X))$.

3.1.1 Examples

Evaluate Integrals using Monte Carlo simulation

The application of Monte Carlo is to computation of integrals. Let $g(x)$ be a function and suppose we wanted to compute I where

$$I = \int_0^1 g(x)dx$$

To compute the value of I, note that if $U \sim \mathrm{U}[0,1]$ then we can express I as

$$I = E[g(U)]$$

If $U_1, \ldots, U_n$ are independent uniform $(0,1)$ random variables, it thus follows that the random variables $g(U_1), \ldots, g(U_n)$ are independent and identically distributed random variable having mean I. Therefore, by law of large numbers, its follows that, with probability,

$$\sum_{i=0}^{n} \frac{g(U_i)}{n} \to E(g(U)) = I \text{ as } k \to \infty$$

Hence, we can approximate I by generating large numbers of random numbers U_i and taking as our approximation the average value of $g(U_i)$.

If we wanted to compute

$$I = \int_a^b g(x)dx$$

then, by taking the substitute $y = (x-a)/(b-a)$, $dy = dx/(b-a)$, we see that

$$\begin{aligned} I &= \int_0^1 g(a + [b-a]y)(b-a)dy \\ &= \int_0^1 h(y)dy \end{aligned}$$

Where $h(y) = (b-a)g(a + [b-a]y)$. Thus, we can approximate I by continually generating random numbers and then taking the average value of h evaluated at these random numbers.

Similarly, if we wanted

$$I = \int_0^\infty g(x)dx$$

we could apply the substitution $y = 1/(x+1)$, $dy = -dx/(x+1)^2 = -y^2dx$, to obtain the identity

$$I = \int_0^1 h(y)dy$$

where,

$$h(y) = \frac{g(\frac{1}{y} - 1)}{y^2}$$

Using this technique we can also evaluate multidimensional integrals. Suppose that g is a function with n-dimention argument and we are interested in computing

$$I = \int_0^1 \int_0^1 \dots \int_0^1 g(x_1, x_2, \dots, x_n)dx_1dx_2 \dots dx_2.$$

Then, we can express I as

$$I = E(g(U_1, U_2, \dots U_n))$$

where $U_1, U_2, \dots U_n \sim$ U$[0,1]$ Hence if we generate k independent sets, each consisting of n independent U$[0,1]$ random variable

$$\begin{gathered} U_1^1 \dots U_n^1 \\ U_1^2 \dots U_n^2 \\ \vdots \\ U_1^k, \dots, U_n^k \end{gathered}$$

then, since the random variables $g(U_1^i, \dots, U_n^i), i = 1, 2, \dots, k$ are all independent and identically distributed random variable with mean I, we can estimate I by $\sum_{i=1}^k g(U_1^i, \dots, U_n^i)/k$.

Example 3.1. Suppose we want to integrate,

$$I = \int_0^1 e^{-\frac{x^2}{2}}dx.$$

Then we can say that,

$$I = E(e^{-\frac{U^2}{2}})$$

where, $U \sim \mathrm{U}[0,1]$. Then simulating large number of $U_1, U_2, \dots, U_n \sim \mathrm{U}[0,1]$ and calculating,

$$\sum_{i=1}^{n} \frac{e^{-\frac{U_i^2}{2}}}{n}$$

we can evaluate I.

Hence, the algorithm is:

STEP 1: Generate $U \sim \mathrm{U}[0,1]$.

STEP 2: Calculate $e^{-\frac{U^2}{2}}$ and retain it and go to STEP 1.

STEP 3: After large number of iteration evaluate the average.

Monte Carlo sample size	Monte Carlo Estimate of I = 0.8556
50	0.8555
100	0.8558
1000	0.8555
10000	0.8556
100000	0.8556

Table 3.1: Monte Carlo Integration of $e^{-x^2/2}$.

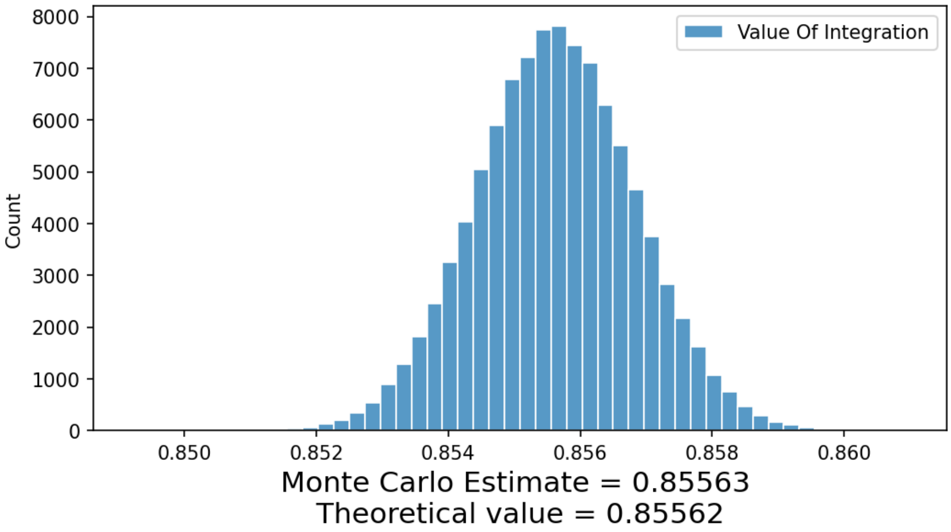

Figure 3.1: Monte Carlo Integration of $e^{-x^2/2}$

Here, we see by the time the Monte Carlo sample size is 100000, we get fairly accurate estimates for the value I.

The Estimation of π

Suppose that the random vector (X, Y) is uniformly distribution in the square of area 4 centered at the origin. That is, it is a random point in the region specified in Figure 3.2. Let us consider now the probability that this random point in the square in contained within the inscribed circle of radius 1 like the Figure 3.3.

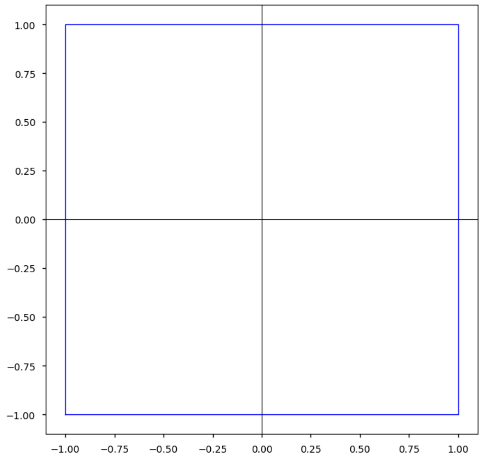

Figure 3.2: Square

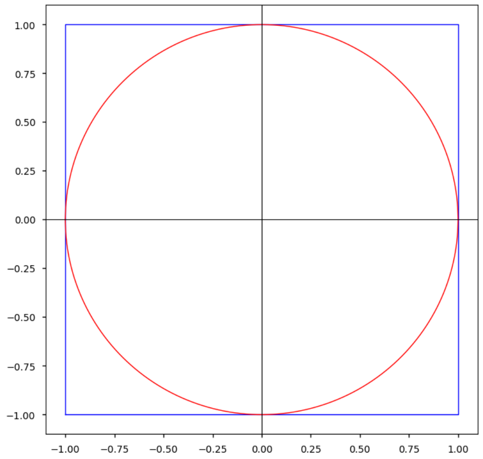

Figure 3.3: Circle within Square

Note that since (X, Y) is uniformly distributed in the square it follows that

$$\begin{aligned} P((X, Y) \text{ is in the circle }) &= P(X^2 + Y^2 \leq 1) \\ &= \frac{\text{Area of the circle}}{\text{Area of the square}} = \frac{\pi}{4} \end{aligned}$$

Hence, we generate large number of points in the square, the proportion of points that fall within the circle will be approximately $\pi/4$. Now, if X and Y were independent and both were uniformly distributed over $(-1, 1)$, their joint density would be

$$\begin{aligned} f(x, y) &= f(x)f(y) \\ &= \frac{1}{2} \times \frac{1}{2} \\ &= \frac{1}{4}, \ -1 \leq x \leq 1, \ -1 \leq y \leq 1 \end{aligned}$$

Since, the density function of (X, Y) is constant in the square, it thus follows that (X, Y) is uniformly distributed in the square. Now, if $U \sim$ U$[0, 1]$ then $2U \sim$ U$[0, 2]$ and so $2U - 1 \sim$ U$[-1, 1]$. Therefore, if we generate random numbers U_1 and U_2 and set $X = 2U_1 - 1$ and $Y = 2U_2 - 1$, and define,

$$I = \begin{cases} 1 \text{ if } X^2 + y^2 \leq 1 \\ 0 \text{ otherwise} \end{cases}$$

Then,

$$E(I) = P(X^2 + y^2 \leq 1) = \frac{\pi}{4}.$$

Hence, the Algorithm for estimating π is:
STEP 1: Set Circle = 1.
STEP 2: Generate $U_1, U_2 \sim$ U$[0, 1]$
STEP 3: If $(2U_1 - 1)^2 + (2U_2 - 1)^2 \leq 1$ Set Circle = Circle + 1, Otherwise return to STEP 2. STEP 4: After simulating N time, set Area of Circle = Circle/N

Monte Carlo sample	Monte Carlo Estimate of π
50	2.9600
100	3.0800
1000	3.1200
10000	3.1428
100000	3.1397
1000000	3.1394
10000000	3.1414

Table 3.2: Monte Carlo Estimates of π

Here, we see by the time the Monte Carlo sample size is 10000000, we get fairly accurate estimates for π

3.2 Importance Sampling

There are two different ways to think about importance sampling. The more traditional one is to go back to the primary problem that Monte Carlo wants to solve, namely to approximate the value of an expectation $\mu = \int \phi_0(x)dF_0(x)$ for some function ϕ_0 and some CDF F_0. However, (ϕ_0, F_0) is not the only pair (ϕ, F) for which $\int \phi(x)dF(x)$ equals the specific number μ. Indeed, given any other CDF F_1,

$$\begin{aligned}\mu &= \int \phi_0(x)dF_0(x)\\ &= \int \phi_0(x)\frac{dF_0}{dF_1}(x)dF_1(x)\\ &= \int \lambda(x)\phi_0(x)dF_1(x).\end{aligned}$$

Where $\lambda(x) = \frac{dF_0}{dF_1}(x)$. If F_0, F_1 have densities f_0, f_1, then $\lambda(x) = \frac{f_0(x)}{f_1(x)}$; if F_0, F_1 have respective pmfs f_0, f_1, then also $\lambda(x) = \frac{f_0(x)}{f_1(x)}$ This raises the interesting possibility that we can sample from a general F_1, and subsequently use the usual Monte Carlo estimate

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \lambda(X_i)\phi_0(X_i) = E_{F_1}[\lambda(X_i)\phi_0(X_i)].$$

Where $X_1, X_2, \ldots, X_n$ is Monte Carlo sample from F_1. Importance sampling poses the problem of finding an optimal choice of F_1 for which to sample, so that $\hat{\mu}$ has the smallest possible variance. The distribution F_1 hat ultimately gets chosen is called the *importance sampling distribution.*

We can visualize this method by an example.

Example 3.2. Suppose we want to evaluate

$$I = \int_0^{10} e^{-2|x-5|}dx.$$

Doing it analytically we get $I = 0.9999$.

Now, suppose $\phi(x) = e^{-2|x-5|}$ then we want to evaluate

$$I = \int_0^{10} \phi(x)dx$$

Now,

$$\begin{aligned}I &= \int_0^{10} \phi(x)dx\\ &= \int_0^{10} \phi(x)\frac{10}{10}dx\\ &= \int_0^{10} 10 \times \phi(x)\frac{1}{10}dx = \int_0^{10} 10\phi(x)f_0(x)dx \text{ where } f_0(x) \text{ pdf of U}(0,1) \qquad (3.1)\\ &= E_U[10 \times \phi(U)] \text{ where, } U \sim \text{U}(0,10).\end{aligned}$$

By Ordinary Monte Carlo technique we can estimate I by $\frac{1}{N}\sum_{i=1}^{N} 10 \times \phi(U_i)$ where

$U_i \sim \mathrm{U}(0, 10)$ for $i = 1, 2, \ldots, N$ for the large number of N.

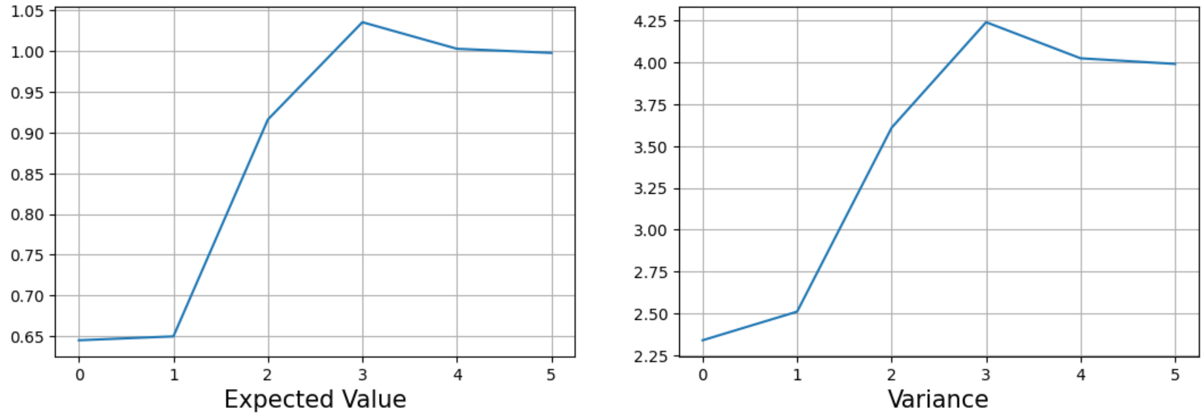

Figure 3.4: Monte Carlo integration of $\int_0^{10} e^{-2|x-5|}dx$.

Sample Size	Estimated Value I=0.9999	Variance
10	0.6450	2.3398
100	0.6498	2.5107
1000	0.9159	3.6097
10000	1.0353	4.2405
100000	1.0027	4.0247
1000000	0.9976	3.9913

Table 3.3: Monte Carlo integration of $\int_0^{10} e^{-2|x-5|}dx$.

Here we can see that for sample size 1M, we get a pretty good estimation of I with less then 0.6% error. But the variance is very high. If we see at left of Figure 3.5 we can see we are taking unnecessary value from low frequency part of $\phi(x)$ that is, from extreme left and right. If we choose an importance sampling distribution that has similar curve as $\phi(x)$, then we can estimate I with low variance. If we choose importance sampling distribution as $\mathrm{N}(5, 1)$ then we see from right of Figure 3.5 it has similar pattern as $\phi(x)$.

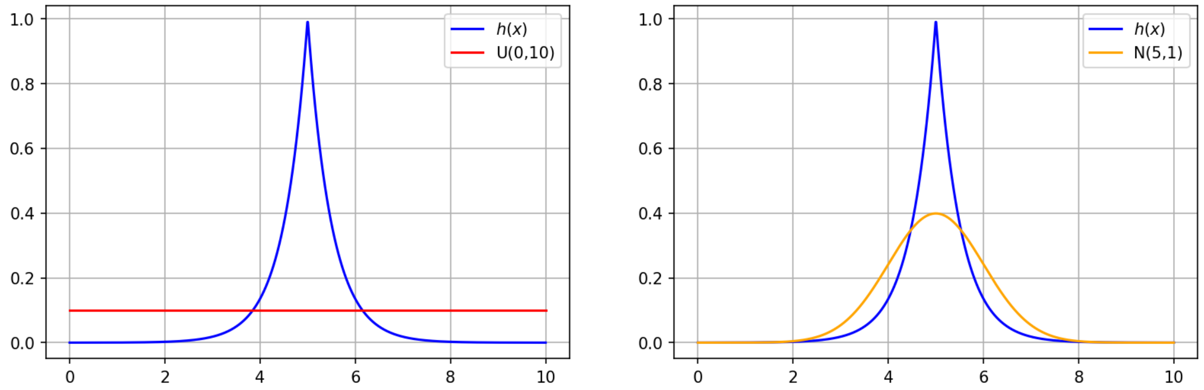

Figure 3.5: $h(x)$ with $U(0, 10)$ and $\mathrm{N}(5, 1)$

Let, $f_1(x)$ is pdf of $\mathrm{N}(0,1)$ then, from Equation (3.1)

$$
\begin{aligned}
I &= \int_0^{10} 10\phi(x) f_0(x) dx \\
&= \int_0^{10} 10\phi(x) \frac{f_0(x)}{f_1(x)} q(x) dx \\
&= E_X \left[10\phi(X) \frac{f_0(X)}{f_1(X)} \right] \text{ where } X \sim \mathrm{N}(5,1) \\
&= E_X \left[10\phi(X) \lambda(X) \right] \text{ where } X \sim \mathrm{N}(5,1)
\end{aligned}
$$

Where, $\lambda(x) = \frac{f_0(x)}{f_1(x)}$ here $f_0(x)$ is pdf of $U(0,1)$ and $f_1(x)$ is pdf of $\mathrm{N}(5,1)$. Now using usual Monte Carlo estimate

$$
I = \frac{1}{N} \sum_{i=1}^{N} 10\phi(X_i)\lambda(X_i)
$$

where $X_1, X_2, \ldots, X_N$ is Monte Carlo sample from $\mathrm{N}(5,1)$.

Sample Size	Estimated Value (I=0.9999)	Variance
10	1.2473	0.3812
100	0.9292	0.2593
1000	1.0018	0.3550
10000	1.0072	0.3603
100000	1.0039	0.3595
1000000	0.9999	0.3580

Table 3.4: Evaluating $\int_0^{10} e^{-2|x-5|} dx$ using Importance Sampling.

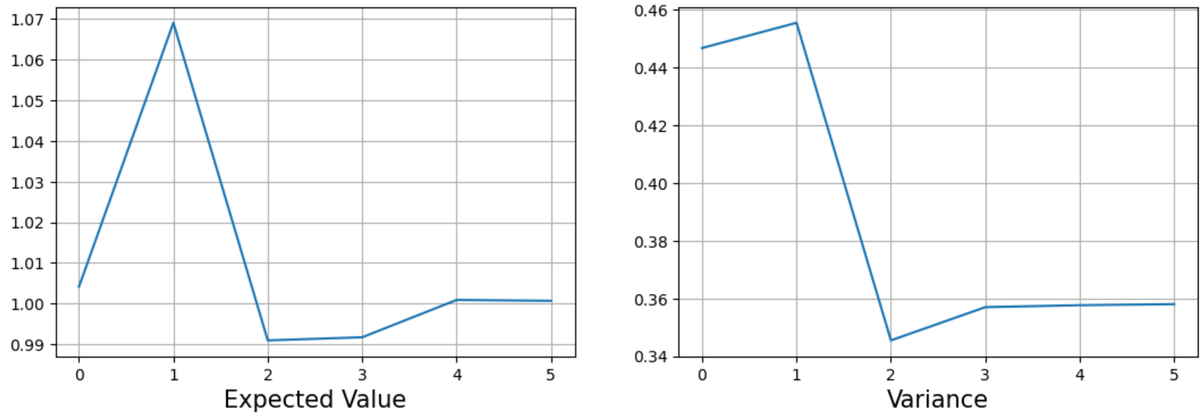

Figure 3.6: Evaluating $\int_0^{10} e^{-2|x-5|} dx$ using Importance Sampling.

Here we can see that the estimation of I is pretty close and variance is also lower the Original Monte Carlo method.

A more contemporary view of importance sampling is that we do not approach importance sampling as an optimization problem, but because the circumstances force us to consider different sampling distributions F.

Now, we also assume that F_0, F_1 both have densities, say f_0, f_1. If F_0, F_1 are both discrete then the notation only change, but the argument is same. Suppose then $f_i(x) = \frac{h_i(x)}{c_i}$, $i = 0, 1$, where the assumption is that h_0, h_1 are completely known and also computable, but c_0, c_1 are unknown and are not even computable. Then, as we showed above, for any function ϕ for which the expectation $E_{F_0}[\phi(X)]$ exist,

$$\begin{aligned}\mu = E_{F_0}[\phi(X)] &:= \int \frac{f_0(x)}{f_1(x)}\phi(x) f_1(x) dx \\ &= \frac{c_1}{c_0}\int \frac{\phi(x)h_0(x)}{h_1(x)} f_1(x)dx \\ &= \frac{c_1}{c_0} E_{F_1}\left(\frac{\phi(X)h_0(X)}{h_1(X)}\right).\end{aligned}$$

This is a useful reduction, but we have to deal with the fact that ratio $\frac{c_1}{c_0}$ is not known to us. Now, if we use the special function $\phi(x) \equiv 1$, the same representation above gives us

$$\begin{aligned}1 &= \frac{c_1}{c_0} E_{F_1}\left(\frac{h_0(X)}{h_1(X)}\right) \\ \implies \frac{c_1}{c_0} &= \frac{1}{E_{F_1}\left(\frac{h_0(X)}{h_1(X)}\right)}\end{aligned}$$

and because h_0, h_1 are explicitly known to us, we have a way to get rid of the quotient $\frac{c_1}{c_0}$ and write the final *importance sampling identity*

$$E_{F_0}[\phi(x)] = \frac{E_{F_1}\left(\frac{\phi(X)h_0(X)}{h_1(X)}\right)}{E_{F_1}\left(\frac{h_0(X)}{h_1(X)}\right)}$$

We can now use an available Monte Carlo sample $X_1, X_2, \ldots, X_n$ from F_1 to find Monte Carlo estimates for $\mu = E_{F_0}[\phi(x)]$

The basic plug-in estimate for μ is the so-called ratio estimate

$$\hat{\mu} = \frac{\sum_{i=1}^{n} \frac{\phi(X_i)h_0(X_i)}{h_1(X_i)}}{\sum_{i=1}^{n} \frac{h_0(X_i)}{h_1(X_i)}}.$$

Example 3.3 (Binomial Bayes problem with an Atypical Prior). Suppose $X \sim Bin(m, p)$ for some fixed m and p has the prior density $c \sin^2(\pi p)$, where c is a normalizing constant. Throughout the example, c denotes a generic constant, and is not intended to mean the same constant at every use.

The posterior density of p given $X = x$ is

$$\pi(p|X = x) = cp^x(1-p)^{m-x}\sin^2(\pi p), \quad 0 < p < 1.$$

The problem is to find the posterior mean

$$\mu = c\int_0^1 p[cp^x(1-p)^{m-x}\sin^2(\pi p)]dp.$$

We use importance sampling to approximate the value of μ. Towards this, choose

$$\phi(p) = p, \;\; h_0(p) = p^x(1-p)^{m-x}\sin^2(\pi p), \;\; h_1(p) = p^x(1-p)^{m-x},$$

so that if $p_1, p_2, \ldots, p_n$ are samples from F_1, (i.e. $p_i \sim \text{Beta}(x+1, m-x+1)$), then the importance sampling estimate of the posterior mean μ is

$$\begin{aligned}\hat{\mu} &= \frac{\sum_{i=1}^{n} \frac{\phi(p_i)h_0(p_i)}{h_1(p_i)}}{\sum_{i=1}^{n} \frac{h_0(p_i)}{h_1(p_i)}} \\ &= \frac{\sum_{i=1}^{n} p_i \sin^2(\pi p_i)}{\sum_{i=1}^{n} \sin^2(\pi p_i)}.\end{aligned}$$

Note that we did not need to calculate the normalizing constant in the posterior density. We take $m = 100$, $x = 45$ for specificity.

Sample Size	Importance Sampling Estimate of μ	Variance
20	0.4356	1.3757
50	0.4575	1.7399
100	0.4596	1.5194
250	0.4499	1.4991
500	0.4486	1.4250

Table 3.5: Importance Sampling Estimates of μ for Different Sample Sizes

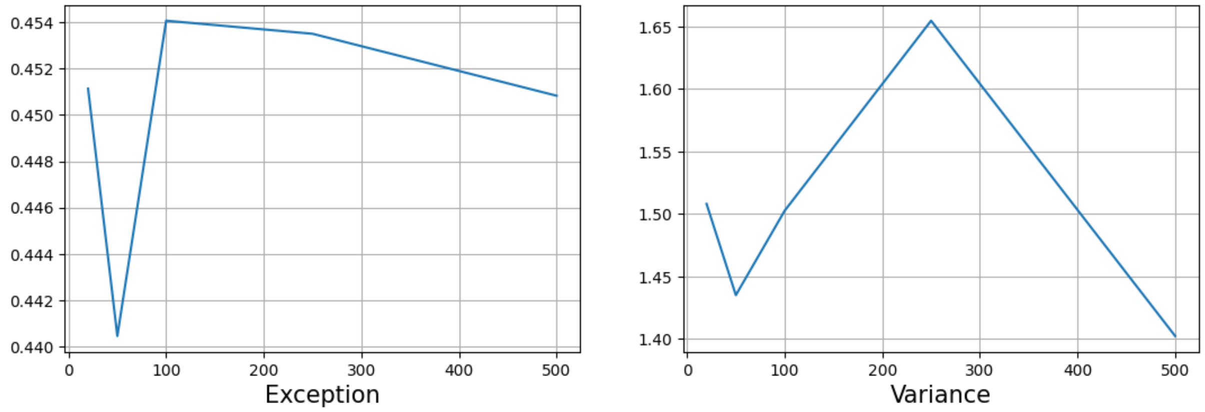

Figure 3.7: Exception and variance graph of Binomial Bayes problem with an Atypical Prior

3.2.1 Optimal Importance Sampling Distribution

We now address the question of the optimal choice of the importance sampling distribution. There is no unique way to define what an optimal choice means. We formulate one definition of optimality and provide an optimal importance sampling distribution. The optimal choice would not be practically usable, as we are shown. However, the solution still gives useful insight.

Theorem 3.2.1. *Consider the importance sampling estimator $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \lambda(X_i)\phi(X_i)$ for $\mu = \int \phi(x)f_0(x)dx$, where $\lambda(x) = \frac{f_0(x)}{f_1(x)}$, and $X_1, \ldots, X_n$ are IID observations from F_1 Assume that $\phi(x) \geq 0$, and $\mu > 0$. Then, $Var_{F_1}(\hat{\mu})$ is minimized when $f_1(x) = \frac{\phi(x)f_0(x)}{\mu}$.*

Proof. Because $X_1, \ldots, X_n$ is iid, so are $\lambda(X_1)\phi(X_1), \ldots, \lambda(X_n)\phi(X_n)$, and hence,

$$Var_{F_1}(\hat{\mu}) = \frac{1}{n} Var_{F_1}(\lambda(X_1)\phi(X_1)).$$

Clearly, this is minimized when with probability one under F_1, $\lambda(X_1)\phi(X_1)$ is constant, say k. The constant k must be equal to the mean of $\lambda(X_1)\phi(X_1)$, that is,

$$\begin{aligned} k &= \int \lambda(x)\phi(x)f_1(x)dx \\ &= \int \frac{\phi(x)f_0(x)}{f_1(x)} f_1(x)dx \\ &= \int \phi(x)f_0(x)dx = \mu. \end{aligned}$$

Therefore, the optimal importance sampling density satisfies $\lambda(x)\phi(x) = \mu$ hence,

$$f_1(x) = \frac{\phi(x)f_0(x)}{\mu}.$$

□

This is not usable in practice, because it involves μ,which is precisely the unknown number we want to approximate. However, the theoretically optimal solution suggests that the importance sampling density should follow key properties of the unnormalized function $\phi(x)f_0(x)$. For example, f_1 should have the same shape and tail behavior as $\phi(x)f_0(x)$.

We have seen this phenomena in the Example 3.3. Because Graph of $\phi(x)h_0(x)$ and $h_1(x)$ in Figure 3.8, they both have the same key properties.

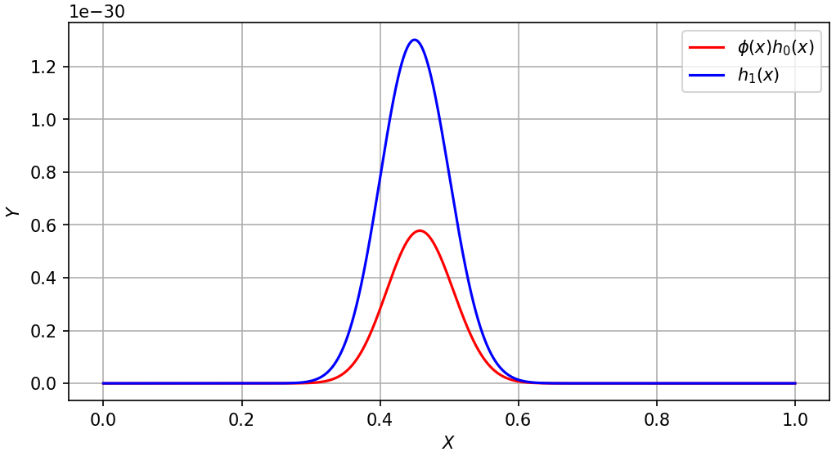

Figure 3.8: Graph of $\phi(x)h_0(x)$ and $h_1(x)$

Chapter 4

Markov Chain Monte Carlo Methods

Markov Chain Monte Carlos(MCMC) are sophisticated collection of techniques that allows us to simulate complex distributions with Markov chains. When the target distributions is an unconventional one, or it is known only up to a normalizing constant that is $f(x) = \frac{h(x)}{c}$ for some explicit function h but only an implicit normalizing constant c, because c can not be computed exactly, the standard simulation techniques are difficult to apply or even not applicable in that case *Markov Chain Monte Carlo*(MCMC) comes in to play. The basic idea for MCMC is to construct a *Markov Chain* whose stationary distribution is the distribution of interest.

MCMC is widely used algorithms it is primarily used for calculating numerical approximations of multi-dimensional integration for example is Bayesian statistic, computational physics, computational biology. In Bayesian statistic, Markov Chain Monte Carlo method are typically used to calculate moments and posterior distribution.

To understand *Markov Chain Monte Carlo*(MCMC) we have to understand about *Markov Chains.*

4.1 Markov Chains

Definition 4.1.1 (Markov Chain)**.** *A discrete time stochastic process* $\{X_n, n = 1, 2, 3, \ldots\}$ *is defined to be Discrete Time Markov Chain or simply Markov Chain if it takes value the state space* $\mathbf{S}$*, and for every* $n \geq 0$ *it satisfies the property*

$$\mathbf{P}(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = \mathbf{P}(X_{n+1} = j | X_n = i) \tag{4.1}$$

Unless otherwise mentioned we take the state space $\mathbf{S}$ to be $\{0, 1, 2, 3, \ldots\}$. If $X_n = i$ we say that the process is in ith state at time n. In the definition Equation (4.1) may be interpreted as for Markov Chain, the conditional distribution of any future state X_{n+1}, given the past states $X_0, X_1, \ldots, X_{n-1}$ and the present state X_n, is independent of the past and only depend on the present state. This property is called *Markovian Property.* In other word for Markov chain predicting the future we only need information about the present state.

Definition 4.1.2 (Homogeneous Markov Chain)**.** *We say a Markov chain* $\{X_n, n \geq 0\}$ *is homogeneous if* $\mathbf{P}(X_{n+1} = j | X_n = i) = \mathbf{P}(X_2 = j | X_1 = i)$ $\forall n > 0$.

The quantity $\mathbf{P}(X_{n+1} = j | X_n = i)$ is called the *transition probability* from state i to state j. For homogeneous Markov Chain we can specify the transition probabilities $\mathbf{P}(X_{n+1} = j | X_n = i)$ by a sequence of value $p_{ij} = \mathbf{P}(X_{n+1} = j | X_n = i)$.

Then the transition probabilities are p_{ij}, $1 \leq i, j \leq \infty$ for transition from state i to state j. The matrix $P = (p_{ij})$ is called The *Transition Matrix* of chain. Since probabilities are non-negative and since the process must make a transition into some state, we have

$$p_{ij} \geq 0, \quad i, j \geq 0, \text{ and } \sum_{j=0}^{\infty} p_{ij} = 1, \quad \forall\, i = 0, 1, 2, \ldots.$$

For, infinite state Markov chain the probability transition matrix will be infinite order. Then,

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdots \\ p_{10} & P_{11} & p_{12} & \cdots \\ \vdots & \vdots & \vdots & \ldots \\ p_{i0} & \cdots & p_{ij} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

We have already defined the one step transition probabilities p_{ij}. We now define the n-step transition probabilities p_{ij}^n to be the probability that a process in state i will be in state j after n additional transitions. i.e.

$$p_{ij}^n = \mathbf{P}(X_{n+m} = j | X_m = i), \ n \geq 0, \ i, j \geq 0.$$

By definition of Markov chain we get,

$$\begin{aligned} p_{ij}^{n+m} &= \mathbf{P}(X_{n+m} = j | X_0 = i) \\ &= \sum_{k=0}^{\infty} \mathbf{P}(X_{n+m} = j, X_n = k | X_0 = i) \text{ (By theorem of total probability)} \\ &= \sum_{k=0}^{\infty} \mathbf{P}(X_{n+m} = j | X_n = k, X_0 = i) \mathbf{P}(X_n = k | X_0 = i) \\ &= \sum_{k=0}^{\infty} \mathbf{P}(X_{n+m} = j | X_n = k) \mathbf{P}(X_n = k | X_0 = i) \end{aligned}$$

Then,

$$p_{ij}^{n+m} = \sum_{k=0}^{\infty} p_{kj}^m p_{ik}^n. \tag{4.2}$$

Equation (4.2) is known as **Chapman-Kolmogorov equation**. If we take $n = m = 1$. Then

$$p_{ij}^2 = \sum_{k=0}^{\infty} p_{kj} p_{ik} \tag{4.3}$$

the above expression is (i, j) element of P^2 matrix then we see Equation (4.3) in matrix form,

$$P^2 = P \cdot P$$

Hence, Equation (4.2) can also be written in matrix form,

$$P^{n+m} = P^n \cdot P^m$$

Where P^n and P^m are the n-step and m-step transition matrix respectively.

Proposition 4.1.1 (Marginal Distribution of X_n)**.** *Define* $\mathbf{t} = (t_1, t_2, \ldots)$ *by* $t_i = \mathbf{P}(X_0 = i)$*, and view* $\mathbf{t}$ *as a row vector. Then the marginal distribution of* X_n *is given by the vector* $\mathbf{t}P^n$*. That is the* j*-th component of* $\mathbf{t}P^n$ *is* $\mathbf{P}(X_n = j)$*.*

Definition 4.1.3. *We say that state* j *is accessible from state* i*, written as* $i \to j$*, If* $p^n_{ij} > 0$ *for some* $n \geq 0$*.*
We assume every state is accessible from itself since

$$p^0_{ii} = \mathbf{P}(X_0 = i | X_0 = i) = 1.$$

Definition 4.1.4. *Two states* i *and* j *are said to communicate, written as* $i \longleftrightarrow j$*, if they are accessible from each other.*
i.e.

$$i \longleftrightarrow j \implies i \to j \quad \& \quad j \to i$$

Communication is an equivalence relation.

Example 4.1. Consider the Markov chain define in the picture Figure 4.1.

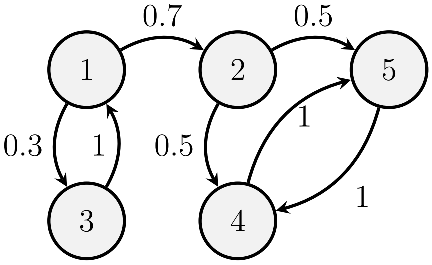

Figure 4.1: Communication classes

here the classes are $\{1,3\}, \{2\}, \{4,5\}$

Definition 4.1.5 (Irreducible Markov chain)**.** *A Markov chain is said to be irreducible if it has only one communicating class. That is, every states communicate with each other. That is, for any states* i*,* j *there is some positive integer* n *such that the* (i,j) *entry of* P^n *is positive.*

A Markov chain that is not irreducible called reducible.

For any state i and j define f^n_{ij} to be the probability that, starting from i, the first transition into j occurs at n time.
i.e.

$$f^n_{ij} = \mathbf{P}(X_n, X_k \neq j, k = 1, 2, \ldots n-1 | X_0 = i).$$

Let

$$f^*_{ij} = \sum_{n=0}^{\infty} f^n_{ij}$$

Then, f^*_{ij} denote the probability of ever making a transition into step j when start from state i. If j is not accessible from i then f^*_{ij} will be zero.

Definition 4.1.6 (Recurrent and Transient state). *A state j of a Markov chain is said to be recurrent $f_{ii}^* = 1$ and transient if $f_{ii}^* < 1$.*

In other word, if a Markov chain start in a recurrent state, there is a guarantee that it will visit that state again in the future (eventually return to that state with probability 1).

In contrast, a transient state in a Markov chain is a state where, once you reach it, there is a positive probability that you will never return to that state. i.e. if you begin in a transient state, there's a chance you won't return there.

Proposition 4.1.2. *In an irreducible Markov chain with a finite state space, all states are recurrent.*

Definition 4.1.7 (Absorbing State). *A state i of Markov chain is called absorbing it $p_{ii} = 1$ that is, it is impossible to leave the state.*

Definition 4.1.8 (Stationary Distribution). *A probability distribution $\{p_j, j \geq 0\}$ is called stationary for the Markov chain if*

$$p_j = \sum_{i=0}^{\infty} p_i p_{ij}, \;\; j \geq 0 \tag{4.4}$$

i.e. If $\mathbf{t} = (p_1, p_2, \ldots, p_j, \ldots)$ is a stationary distribution vector and P is transition matrix, Then

$$\mathbf{t} = \mathbf{t}P. \tag{4.5}$$

From Equation (4.5) we see that 1 is a eigenvalue of transition matrix P and $\mathbf{t}$ is eigenvector corresponding to 1. Since in transition matrix such that $\sum_{j=0}^{\infty} p_{ij} = 1 \; \forall i$ i.e. sum of all elements of row is 1, 1 must be an eigenvalue.

For an irreducible Markov chain where all states are positive recurrent stationary distribution always exists and it is unique.

Definition 4.1.9 (Time Reversible Markov Chain). *If for any Markov chain $p_{ij}^* = p_{ij}, \; \forall \; i, \; j$ then the Markov chain is called time Reversible.*

Hence, the condition for time Reversibility is

$$\pi_i p_{ij} = \pi_j p_{ji} \;\; \forall \; i, \; j$$

Proposition 4.1.3 (Reversible implies stationary). *Suppose that $P = (qij)$ is a transition matrix of a Markov chain that is reversible with respect to a non-negative vector $\mathbf{s} = (s_1, ..., s_M)$ whose components sum to 1. Then s is a stationary distribution of the chain.*

Proof. We have

$$\sum_{i=0}^{\infty} s_i p_{ij} = \sum_{i=0}^{\infty} s_j p_{ji} = s_j \sum_{i=0}^{\infty} p_{ji} = s_j,$$

So, $\mathbf{s}$ is stationary. □

4.2 The Metropolis-Hastings Algorithm

Metropolis-Hastings algorithm is one of the most used *Markov Chain Monte Carlo*(MCMC) algorithm. The *Metropolis algorithm* was first introduced by Nicholas Metropolis in 1953 in his paper entitled *"Equation of State Calculations by Fast Computing Machines"*, with Arianna W. Rosenbluth, Marshall Rosenbluth, Augusta H. Teller and Edward Teller. Arianna Rosenbluth wrote the first full implementation of Metropolis Algorithm for *Mathematical Analyzer Numerical Integrator and Automatic Computer Model I*(MANIAC 1) which was an early computer built under the direction of Nicholas Metropolis at the Los Alamos Scientific Laboratory.

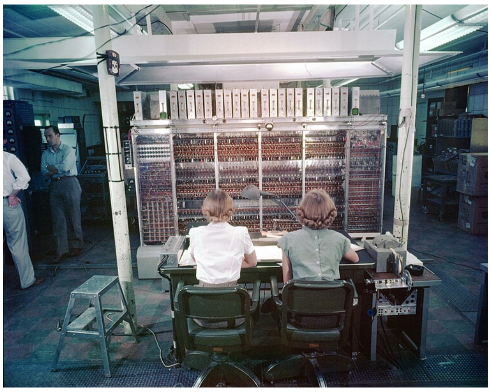

Figure 4.2: MANIAC 1 one of the earliest computer.

For many years this algorithm was simply known as *Metropolis Algorithm*, later in 1970 W.K. Hastings introduce more general version of this algorithm in his paper *"Monte Carlo Sampling Methods Using Markov Chains and Their Applications"*. This generalized Metropolis algorithm is known as *Metropolis-Hastings algorithm*(MH algorithm).

Suppose we want to simulate a random variable or sequence of random variables with probability mass function

$$\pi(\theta) = \frac{f(\theta)}{K} \tag{4.6}$$

where K is normalizing constant which is unknown or difficult to compute.

One way to simulate $\pi(\theta)$ is to constant a Markov Chain that is easy to simulate and whose limiting distribution is $\pi(\theta)$. The *Metropolis-Hastings algorithm* do exactly this. MH algorithm constant a time-reversible Markov Chain with desired limiting probabilities.

4.2.1 Algorithm for Metropolis-Hastings

1. Start with any **initial state** θ_0 satisfying $f(\theta) > 0$.

2. Using a **current state** θ, sample **candidate state** θ' from some **jumping distribution** $q(\theta, \theta') = q(\theta'|\theta)$, which is the probability of jumping to θ' provided the current state in θ.

3. For a given the candidate state θ' calculate the **acceptance probability** $\alpha(\theta, \theta')$ by,

$$\alpha(\theta, \theta') = \min\left(\frac{\pi(\theta')q(\theta', \theta)}{\pi(\theta)q(\theta, \theta')}, 1\right) = \min\left(\frac{f(\theta')q(\theta', \theta)}{f(\theta)q(\theta, \theta')}, 1\right)$$

4. Accept the candidate point with probability α.

We can summarize the Metropolis-Hastings Algorithm as first computing,

$$\alpha(\theta_t, \theta_{t+1}) = \min\left(\frac{f(\theta_{t+1})q(\theta_{t+1}, \theta_t)}{f(\theta_t)q(\theta_t, \theta_{t+1})}, 1\right)$$

and then accepting the candidate point θ_{t+1} with probability α. This generates a Markov Chain $(\theta_0, \theta_1, \ldots, \theta_t, \ldots)$, as the transition probabilities from θ_t to θ_{t+1} depends only on θ_t and not on $(\theta_0, \theta_1, \ldots, \theta_{t-1})$.

4.2.2 Metropolis-Hastings Algorithm as a Markov Chain

To determine Metropolis-Hastings Sampling generates a Markov Chain whose stationary distribution is candidate distribution $\pi(\theta)$ if the Metropolis-Hastings transition kernel,

$$P(\theta_1 \to \theta_2) = P(\theta_1, \theta_2) = q(\theta_1, \theta_2)\alpha(\theta_1, \theta_2) = q(\theta_1, \theta_2) \times \min\left(\frac{f(\theta')q(\theta', \theta)}{f(\theta)q(\theta, \theta')}, 1\right) \tag{4.7}$$

is time-reversible and satisfies

$$P(\theta_1, \theta_2)\pi(\theta_1) = P(\theta_2, \theta_1)\pi(\theta_2)$$

or

$$q(\theta_1, \theta_2)\alpha(\theta_1, \theta_2)\pi(\theta_1) = q(\theta_2, \theta_1)\alpha(\theta_2, \theta_1)\pi(\theta_2)\ \forall\theta_1, \theta_2 \tag{4.8}$$

For time-reversibility we choose jumping distribution $q(\theta_1, \theta_2)$ to be irreducible and $q(\theta_1, \theta_2) = q(\theta_2, \theta_1)$ and for Equation (4.8) we consider the cases.

Case 1: $q(\theta_1, \theta_2)\pi(\theta_1) = q(\theta_2, \theta_1)\pi(\theta_2)$ Hence,

$$\alpha(\theta_1, \theta_2) = \alpha(\theta_1, \theta_2) = 1$$

In this case Equation (4.8) will easily hold.

Case 2: $q(\theta_1, \theta_2)\pi(\theta_1) > q(\theta_2, \theta_1)\pi(\theta_2)$. Hence,

$$\alpha(\theta_1, \theta_2) = \frac{\pi(\theta_2)q(\theta_2, \theta_1)}{\pi(\theta_1)q(\theta_1, \theta_2)} \text{ and } \alpha(\theta_2, \theta_1) = 1$$

Then,

$$\begin{aligned}
P(\theta_1, \theta_2)\pi(\theta_1) &= q(\theta_1, \theta_2)\alpha(\theta_1, \theta_2)\pi(\theta_1) \\
&= q(\theta_1, \theta_2)\frac{\pi(\theta_2)q(\theta_2, \theta_1)}{\pi(\theta_1)q(\theta_1, \theta_2)}\pi(\theta_1) \\
&= q(\theta_2, \theta_1)\pi(\theta_1) = q(\theta_2, \theta_1)\alpha(\theta_2, \theta_1)\pi(\theta_1) \\
&= P(\theta_2, \theta_1)\pi(\theta_2)
\end{aligned}$$

Hence this case satisfies Equation (4.8).

Case 3: $q(\theta_1, \theta_2)\pi(\theta_1) < q(\theta_2, \theta_1)\pi(\theta_2)$ Hence,

$$\alpha(\theta_1, \theta_2) = 1 \text{ and } \alpha(\theta_2, \theta_1) = \frac{\pi(\theta_1)q(\theta_1, \theta_2)}{\pi(\theta_2)q(\theta_2, \theta_1)}$$

Then,

$$\begin{aligned}P(\theta_2,\theta_1)\pi(\theta_2) &= q(\theta_2,\theta_1)\alpha(\theta_2,\theta_1)\pi(\theta_2)\\ &= q(\theta_2,\theta_1)\frac{\pi(\theta_1)q(\theta_1,\theta_2)}{\pi(\theta_2)q(\theta_2,\theta_1)}\pi(\theta_2)\\ &= q(\theta_1,\theta_2)\pi(\theta_1) = q(\theta_1,\theta_2)\alpha(\theta_1,\theta_2)\pi(\theta_1)\\ &= P(\theta_1,\theta_2)\pi(\theta_1)\end{aligned}$$

Hence also for this case Equation (4.8) is satisfied.

4.2.3 Burn-In period

A main problem with the successful implementation of Metropolis-Hastings Algorithm infect any MCMC Methods is number of steps until the chain approaches stationarity. Typically, the first 25% samples are thrown out. These are called burn-in of a sample.

The name "burn-in" comes form electronics. Many electronics components fail quickly, those that don't are more reliable subset. So a burn-in is done at the factory to eliminate the worst.

There is no rule how many samples are chosen as burn-in, this is a very difficult problem to answer. A poor choice of initial values and/or jumping distribution can greatly increase the requirement of burn-in time, this is a hot research topic how do we choose an optimal starting point and jumping distribution. For simplicity, we choose starting value to be as close as center of the candidate distribution.

"Burn-in is only one method, and not a particularly good method, of finding a good starting point"

A chain is said to be **poorly mixing** if it says in small regions of the parameter space for long periods of time, as opposed to a well **mixing chain** that seems to happily explore the space. A poorly mixing chain can arise because the target distribution is multimodal and our choice of starting values traps us near one of the modes. To avoid this issue we can use multiple highly dispersed initial values to start several different chains.

4.2.4 Choosing Jumping Distribution

Now the question arise how to choose the best jumping distribution that works? There are two approaches. First and must common one is the new value θ_{t+1} equals the current value θ_t plus a random noise z. That is,

$$\theta_{t+1} = \theta_t + z$$

In this case, $q(\theta_t,\theta_{t+1}) = g(\theta_{t+1}-\theta_t) = g(z)$, the density associated with the random noise z. If $g(z) = g(-z)$, i.e., the density for the random variable z is symmetric.

Typically, we take, z to be from normal or multivariate normal distribution with mean zero. Then, θ_{t+1} is form normal or multivariate normal distribution with mean θ_t.

Then, we can use Metropolis-Hastings sampling as,

$$\frac{q(\theta_t,\theta_{t+1})\pi(\theta_t)}{q(\theta_{t+1},\theta_t)\pi(\theta_{t+1})} = \frac{g(z)\pi(\theta_t)}{g(-z)\pi(\theta_{t+1})} = \frac{\pi(\theta_t)}{\pi(\theta_{t+1})}$$

We can adjust the variance of jumping distribution to get better mixing.

Second one is, we use an independent chain. The probability of jumping to a point θ_{t+1} is independent of current position θ_t of the chain, i.e. $q(\theta_t, \theta_{t+1}) = g(y)$. Thus the current value is simply drawn from a distribution of interest, independent of current position.

4.2.5 Convergence Diagnostics

Now we have to ensure that Markov Chains have reached stationarity and only use those samples that have been generated after stationarity has been reached. But it is impossible to ensure when those two conditions are satisfied since the Markov Chain does not begin with stationary distribution. Instead, we can use various methods to assess whether or not stationarity appears to have reached. Most common one is:

Visual inspection where we plot variable of interest vs iteration number, plot running means of variables of interest etc or run various iteration of samples with different initial states and different jumping distribution and compare them. This method is manual and need lots of works.

Another one is **Geweke test**, splits sample (after burn-in period) into two parts. Say the first 10% and last 50%. If the chain is at stationarity, the means of two samples should be equal. A modified z-test can be used to compare the two subsamples, and the resulting test statistic is often referred to as a **Geweke z-score**. A value larger than 2 indicates that the mean of the series is still drifting, and a longer burn-in is required before monitoring the chain (to extract a sampler) can begin. Formula for Geweke z-score is given by,

$$z = \frac{\bar{X_1} - \bar{X_2}}{\sqrt{\frac{1}{n_1} S_0^{X_1} + \frac{1}{n_2} S_0^{X_2}}}$$

Where, X_1 is the first 10% subsamples, X_2 last 50% subsamples, S_0 is the spectral density at zero frequency and n_i are the number of samples in X_i for $i = 1, 2$.

4.2.6 Examples

Now we see some examples how we can use Metropolis-Hastings Algorithm.

Example 4.2 (Simulating from an unknown distribution). The besis problem Metropolis-Hastings algorithm solves is to provide a method for sampling from some arbitrary probability distribution. In this example we see how it is works,

Suppose, we have

$$p(x) = \frac{e^{(-x^2)} \left(2 + \sin(5x) + \sin(2x)\right)}{\int_{-\infty}^{\infty} e^{(-u^2)} \left(2 + \sin(5u) + \sin(2u)\right) du}$$

Now we want to generate a random variables form $p(x)$. It may be very hard to calculate the integration in the denominator, or we don't want to calculate it. i.e., we know the probability distribution up to normalizing constant. So we have,

$$p(x) \propto e^{(-x^2)} \left(2 + \sin(5x) + \sin(2x)\right)$$

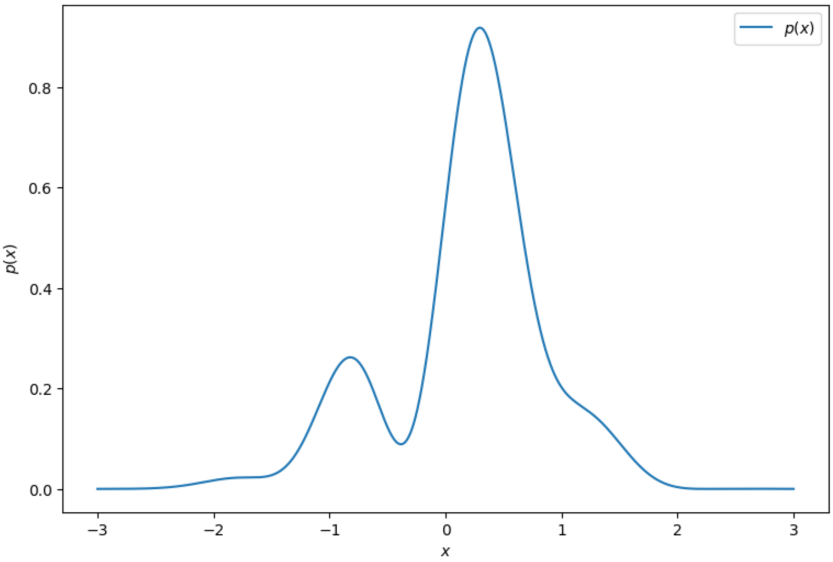

Figure 4.3: Plot of original $p(x)$

Here, we choose,

$$q(\theta_{t-1}, \theta_t) = q(\theta_t|\theta_{t-1}) \sim \mathrm{N}(\theta_{t-1}, \sigma^2)$$

for some stander deviation σ that we have to select. Then,

$$q(\theta_t|\theta_{t-1}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\theta_t - \theta_{t-1})^2\right)$$

and,

$$q(\theta_{t-1}|\theta_t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(\theta_{t-1} - \theta_t)^2\right)$$

Then, it is clear that $q(\theta_{t-1}, \theta_t) = q(\theta_t.\theta_{t-1})$

Hence, the acceptance probability becomes,

$$\begin{aligned}\alpha(\theta_{t-1}, \theta_t) &= \min\left(\frac{p(\theta_t)q(\theta_t.\theta_{t-1})}{p(\theta_{t-1})q(\theta_{t-1}, \theta_t)}, 1\right) \\ &= \min\left(\frac{p(\theta_t)}{p(\theta_{t-1})}, 1\right) \\ &= \min\left(\frac{e^{(-\theta_t^2)}(2 + \sin(5\theta_t) + \sin(2\theta_t))}{e^{(-\theta_{t-1}^2)}(2 + \sin(5\theta_{t-1}) + \sin(2\theta_{t-1}))}, 1\right)\end{aligned}$$

Now, using various initial state and different σ (stander deviation of jumping distribution) simulate samples and study them.

Case 1: First we choose 0 as initial state and $\sigma = 2$. Then after 100000 iteration we get mean of the samples to be 0.1866 and variance of the samples to be 0.4754. From Figure 4.4 we can see that as we are starting from middle of our target distribution the simulated values are very good estimation of desire distribution $p(x)$. Here we see we don't need burn-in time since from beginning the means of the sample are quite same. For the sake of tradition if we throw 25% of samples and get 0.1900 as mean, 0.4751 as variance of the new samples and 0.3041 Geweke Z-score.

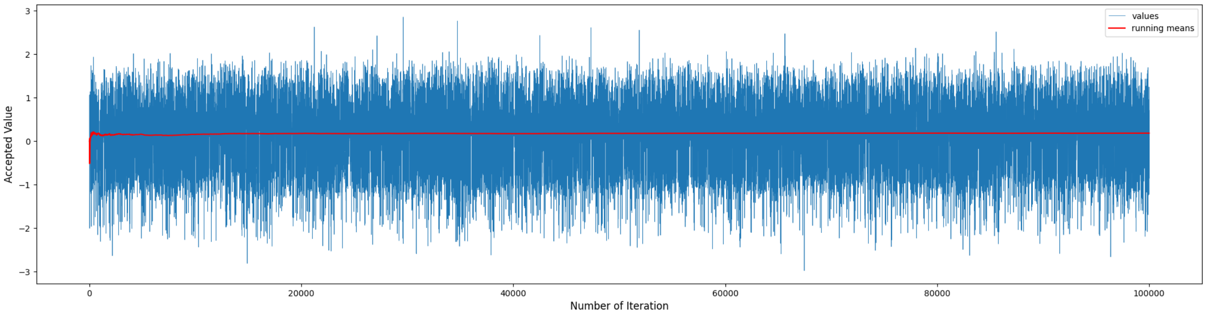

Figure 4.4: Accepted values and running means for case 1 (initial state 0, $\sigma = 2$) before removing burn-ins

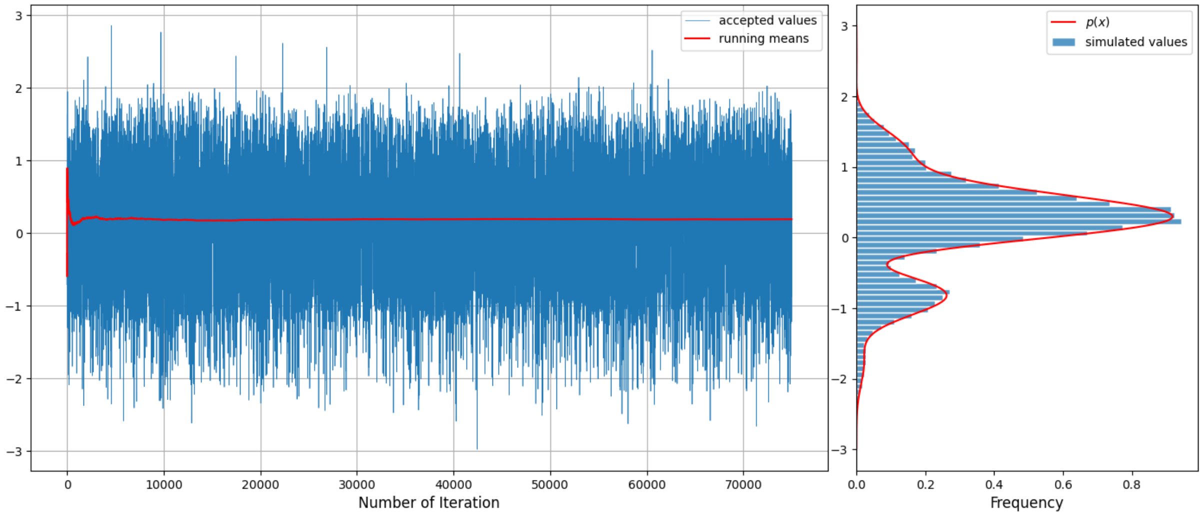

Figure 4.5: Samples of case 1 after removing Burn-Ins

Case 2: Now we take -1 as initial state and σ to be 2, then we get mean and variance of the samples 0.1832 and 0.4641 respectively. After removing the Burn-Ins we get mean 0.17842, variance 0.47186 and Geweke Z-score 0.5433.

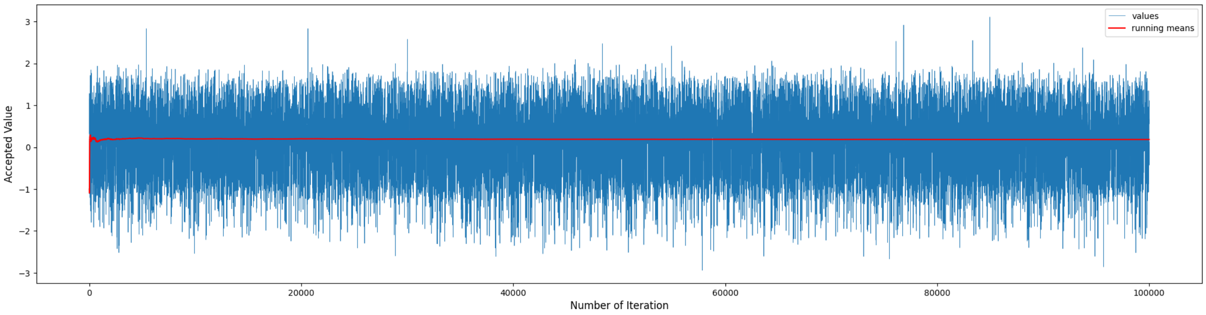

Figure 4.6: Accepted values and running means for case 2 (initial state -1, $\sigma = 2$) before burn-in removed

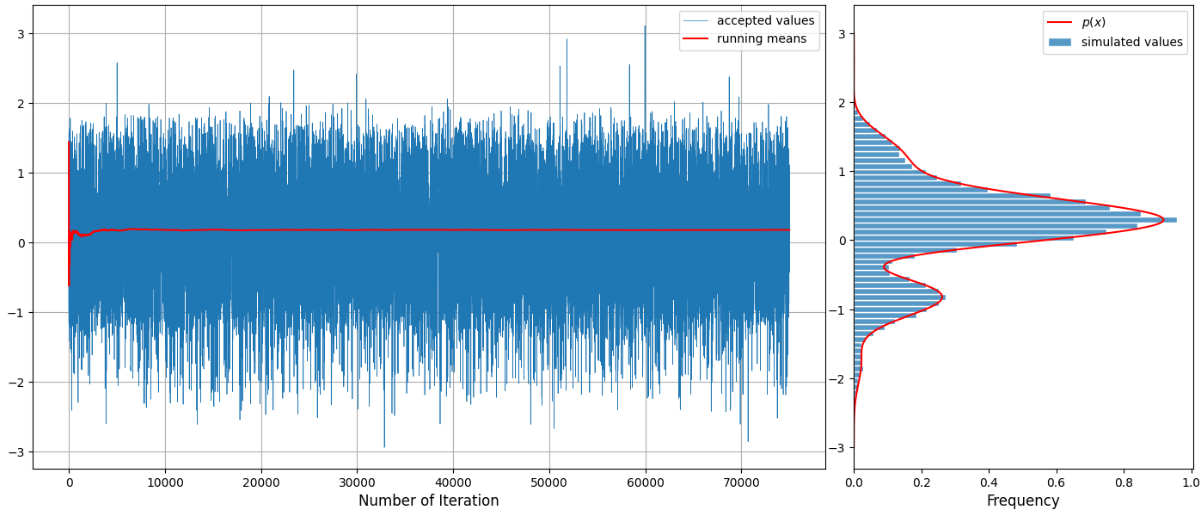

Figure 4.7: Samples of case 2 after removing Burn-Ins

Case 3: As case 3 we take initial state to be -4 and $\sigma = 1$, then we have mean = 0.1906, variance = 0.4686. From Figure 4.8 we can see for first few samples we get means to fluctuate heavily after that it is stable. So, here it is necessary to remove first few terms (Burn-In). After removing first 25% of term we get, mean = 0.1695, variance = 0.4686 and Geweke Z-score = -0.348. Hence seeing the Geweke z-score we see after Burn-Ins are removed we get good estimation of $p(x)$ (Figure 4.9).

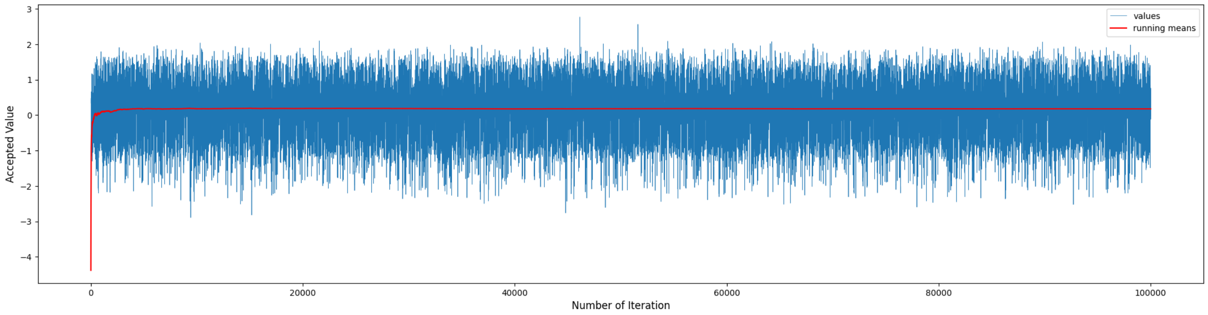

Figure 4.8: Accepted values and running means for case 3 (initial state -4, $\sigma = 1$) before burn-in removed

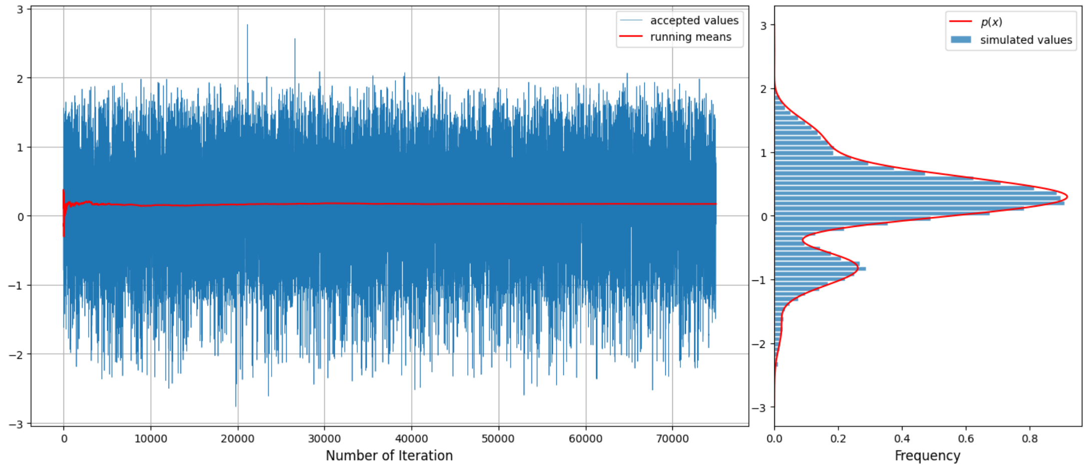

Figure 4.9: Samples of case 3 after removing Burn-Ins

Case 4: In this scenario, we start with an initial value far outside the desired distribution, specifically at 10, with a standard deviation of $\sigma = 2$. Observing Figure 4.10, it becomes evident that removing some of the initial samples (known as Burn-Ins) is necessary. After discarding the first 25% of the samples, we achieve a much better approximation of the target distribution. The results after this adjustment show a mean of 0.1695, a variance of 0.4686, and a Geweke Z-score of -0.4693.

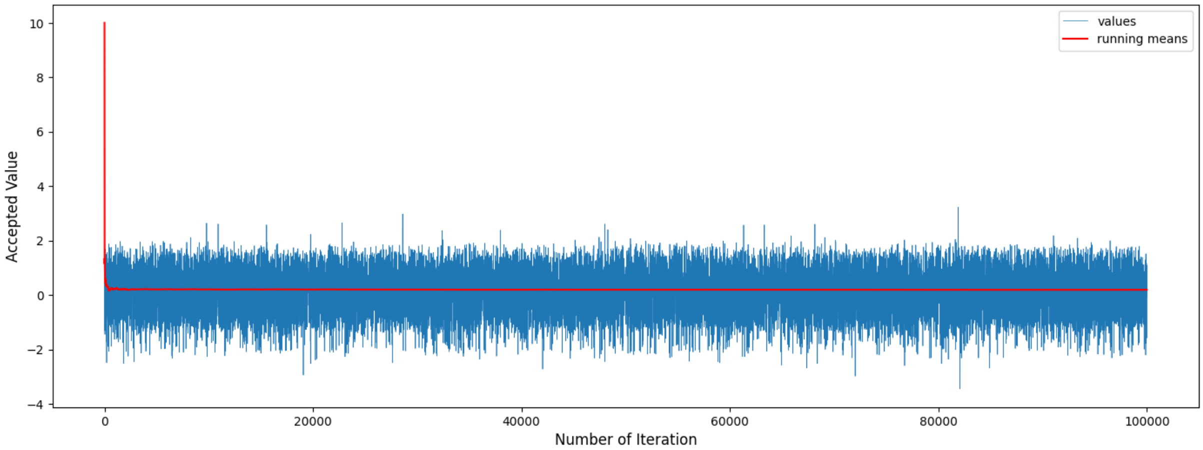

Figure 4.10: Accepted values and running means for case 4 (initial state 10, $\sigma = 2$) before burn-in removed

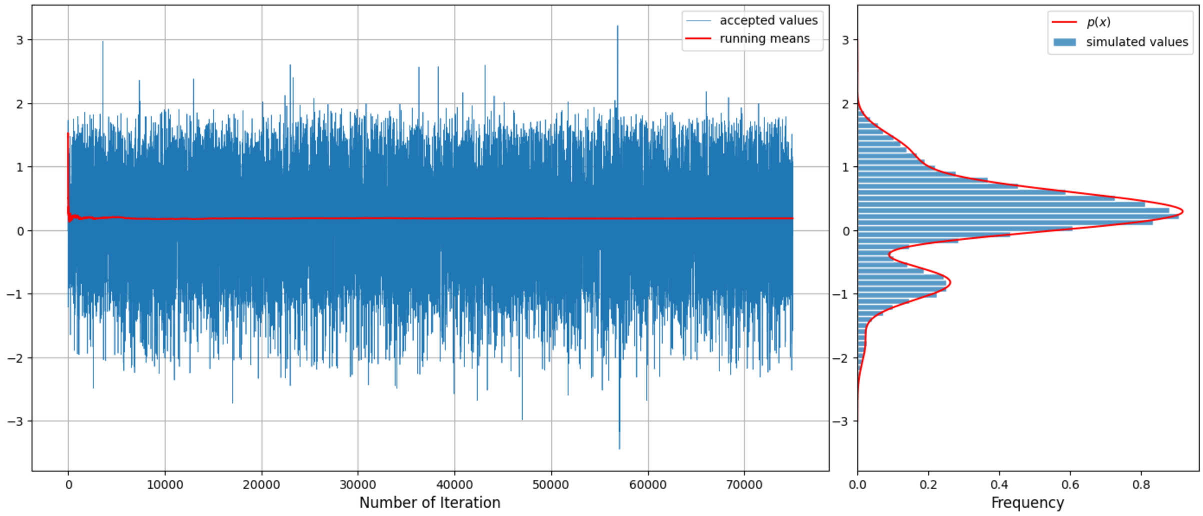

Figure 4.11: Samples of case 4 after removing Burn-Ins

Case 5: Now, let's examine some poor examples of the jumping distribution, using $\sigma = 0.025$ and an initial state of 10. Initially, the mean is 0.2626 and the variance of the sample is 1.4632. Although the mean is quite accurate, the high variance and triable Geweke Z-score (1.152) indicates a poor estimation, as confirmed by Figure 4.12. It is crucial to remove the first 25% of the samples. After discarding these initial samples, we obtain a mean of 0.0726 and a variance of 0.3740. With the reduced variance, this can be considered a good estimator.

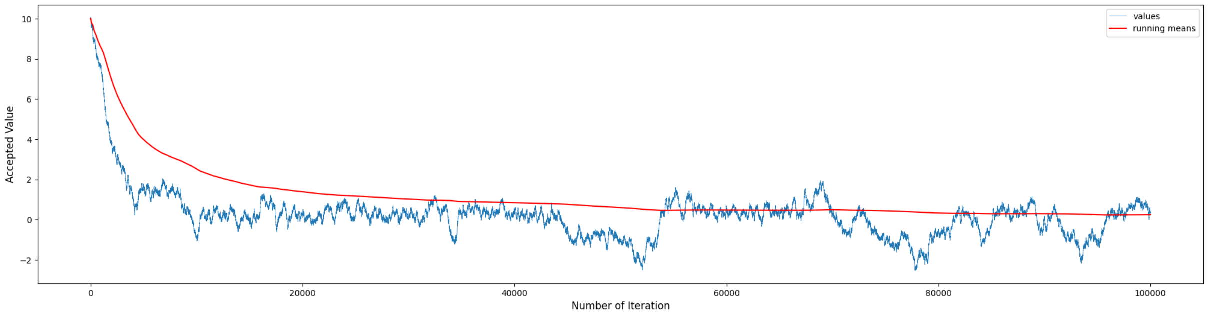

Figure 4.12: Accepted values and running means for case 5 (initial state 10, $\sigma = 0.025$) before burn-in removed

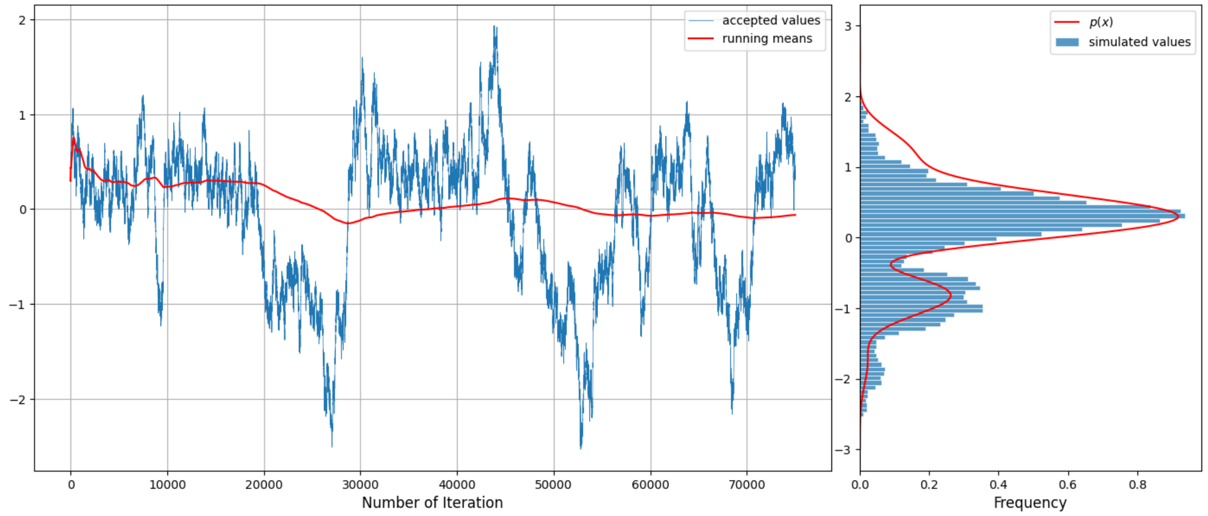

Figure 4.13: Samples of case 5 after removing Burn-Ins

Case 6: Now we take σ to be 20 and initial state 15. After removing Burn-Ins we obtain mean of 0.1899, variance of 0.4423 and Geweke Z-score 3.301. Here we can see not much candidates are getting accepted and jugging from Geweke Z-score we can't accept these samples.

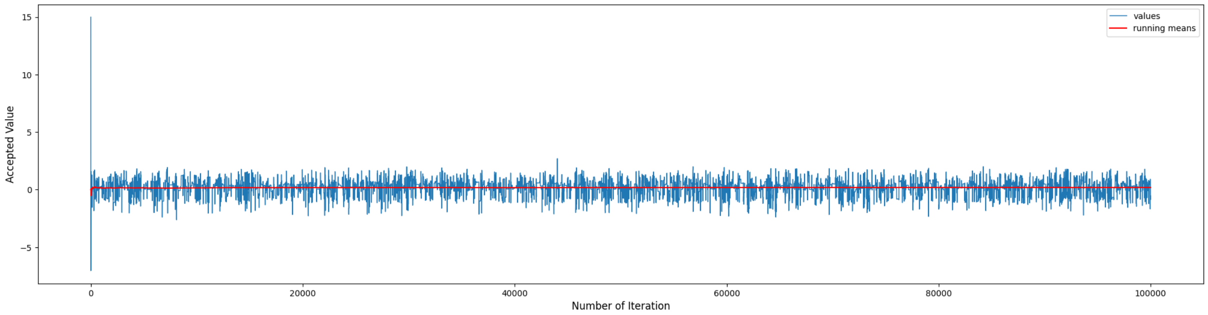

Figure 4.14: Accepted values and running means for case 6 (initial state 15, $\sigma = 20$) before burn-in removed

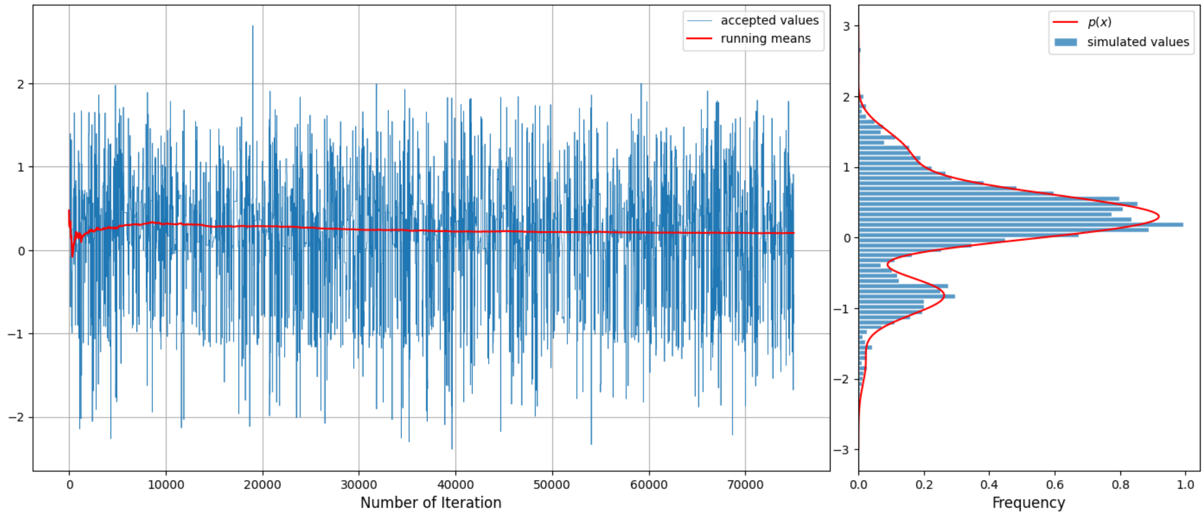

Figure 4.15: Samples of case 6 after removing Burn-Ins

Case	Initial State	σ	Mean	Variance	Mean after Burn-In	Variance after Burn-In	Geweke z-score
1	0	2	0.1866	0.4754	0.1900	0.4751	0.3041
2	-1	2	0.1832	0.4641	0.17842	0.4718	0.5433
3	-4	1	0.1906	0.4686	0.1695	0.4686	-0.348
4	10	2	0.1908	0.4707	0.1695	0.4686	-0.4693
5	10	0.025	0.2626	1.4632	0.0726	0.3740	1.512
6	15	20	0.1899	0.4423	0.1899	0.4423	3.301

Table 4.1: Summary of Cases with Initial States, σ, Means, Variances, and Geweke z-test

Example 4.3 (Finding distribution of some observed data). Another use case of Metropolis-Hastings Algorithm to find distribution of some observed data using **Bayesian Method**. Suppose we have 1000 observed samples. Now we want to find the distribution of the samples, i.e. from which distribution the samples is taken so we can generate more sample data from that distribution. Our observed data looks like Figure 4.16.

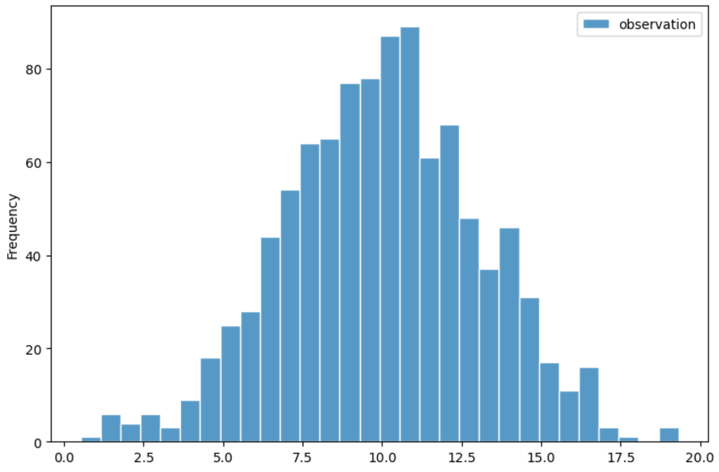

Figure 4.16: Observed Samples

From Figure 4.16 its looks like samples are form Normal Distribution with mean (μ_{obs}) 10, but we don't know the variance (σ^2). Using MH algorithm with the help of Bayesian method we try to find the variance. For that, at first we discuss about **Bayesian Method**.

$$\begin{aligned} P(\Theta = \theta | X = x) &= \frac{P(X = x, \Theta = \theta)}{P(X = x)} \\ &= \frac{P(X = x | \Theta = \theta) P(\Theta = \theta)}{\sum_\theta P(X = x | \Theta = \theta) P(\Theta = \theta)} \end{aligned} \tag{4.9}$$

Equation (4.9) we know as **Bayes' theorem** for continuous case we may write Equation (4.9) as

$$f(\theta|x) = \frac{f(x|\theta)f(\theta)}{\int f(x|\theta)f(\theta)d\theta} \tag{4.10}$$

here, the probability density $f(\theta)$ is called **prior distribution**, $f(\theta|x)$ is known as **posterior distribution**

If we have n IID observations $X_1, X_2, X_3, \ldots, X_n$, we replace $f(x|\theta)$ with

$$f(x_1, x_2, \ldots, x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta) = \mathcal{L}_n(\theta)$$

$\mathcal{L}_n(\theta)$ is known as **likelihood function**. Now, we denote $X^n = (X_1, X_2, \ldots, X_n)$ and $x_n = (x_1, x_2, \ldots, x_n)$.

Now,

$$f(\theta|x^n) = \frac{f(x^n|\theta)f(\theta)}{\int f(x^n|\theta)f(\theta)d\theta} = \frac{\mathcal{L}_n(\theta)f(\theta)}{\int \mathcal{L}_n(\theta)f(\theta)d\theta} \propto \mathcal{L}_n(\theta)f(\theta) \tag{4.11}$$

Since, $\int \mathcal{L}_n(\theta)f(\theta)d\theta$ does not depend on θ

So, we can summarize Bayesian Method to be

$$f(\theta|x^n) \propto \mathcal{L}_n(\theta)f(\theta) \tag{4.12}$$

choosing some prior beliefs using some observed data we upgrade our beliefs and calculate the posterior.

Now coming to our example write the Equation (4.12) as,

$$f(\sigma|D, \theta) \propto \mathcal{L}_n(\sigma)f(\sigma)$$

Here, we are taking the prior to be $f(\sigma)$ as μ is constant and

$$f(d_i|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(d_i - \mu)^2}{2\sigma^2}\right)$$

then, likelihood function is given by,

$$\mathcal{L}_n(\sigma) = \prod_{i=1}^{n} f(d_i|\mu, \sigma) = \prod_{i=1}^{n} \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(d_i - \mu)^2}{2\sigma^2}\right)\right)$$

Now the question is that how we choose prior? If we know σ can't be negative as $\sigma = \sqrt{\frac{1}{n}\sum_i^n (d_i - \mu)^2}$. So we take,

$$f(\sigma) = \begin{cases} 0, & \sigma \le 0 \\ 1, & \sigma > 0 \end{cases} \tag{4.13}$$

We take **jumping distribution** to be,

$$q(\sigma_{t+1}|\sigma_t) = \mathrm{N}(\sigma_t, 1) \tag{4.14}$$

Now, for acceptance probability, we accept σ_{t+1} if,

$$\frac{\mathcal{L}_n(\sigma_{t+1})f(\sigma_{t+1})}{\mathcal{L}_n(\sigma_t)f(\sigma_t)} \ge 1 \tag{4.15}$$

If the ratio is less than 1, then we compare it to a uniform random number from $[0, 1]$. If the ratio is larger or equal to the random number we accept σ_{t+1} else we reject it.

Now we take log on the both side of Equation (4.15). Since log is increasing function, so sign will be same then we have,

$$\begin{aligned}
\log\left(\mathcal{L}_n(\sigma_{t+1})f(\sigma_{t+1})\right) - \log\left(\mathcal{L}_n(\sigma_t)f(\sigma_t)\right) &\ge 0 \\
\text{or, } \log(\mathcal{L}_n(\sigma_{t+1})) + \log(f(\sigma_{t+1})) &\ge \log(\mathcal{L}_n(\sigma_t)) + \log(f(\sigma_t)) \\
\text{or, } \sum_{i=1}^{n} \log(f(d_i|\mu, \sigma_{t+1})) + \log(f(\sigma_{t+1})) &\ge \sum_{i=1}^{n} \log(f(d_i|\mu, \sigma_t)) + \log(f(\sigma_t)) \\
\text{or, } \sum_{i=1}^{n} \left(\log(\sigma_{t+1}\sqrt{2\pi}) - \left(\frac{(d_i-\mu)^2}{2\sigma_{t+1}^2}\right)\right) + \log(f(\sigma_{t+1})) &\ge \\
\sum_{i=1}^{n} \left(\log(\sigma_t\sqrt{2\pi}) - \left(\frac{(d_i-\mu)^2}{2\sigma_t^2}\right)\right) + \log(f(\sigma_t))
\end{aligned} \tag{4.16}$$

Now, we accept σ_{t+1} if it satisfies Equation (4.16) or if not satisfies the Equation we calculate $\alpha(\sigma_{t+1}, \sigma_t)$ by,

$$\begin{aligned}
\alpha(\sigma_{t+1}, \sigma_t) = \exp\Bigg(&\sum_{i=1}^{n} \left(\log(\sigma_{t+1}\sqrt{2\pi}) - \frac{(d_i-\mu)^2}{2\sigma_{t+1}^2}\right) + \log(f(\sigma_{t+1})) \\
&- \sum_{i=1}^{n} \left(\log(\sigma_t\sqrt{2\pi}) - \frac{(d_i-\mu)^2}{2\sigma_t^2}\right) + \log(f(\sigma_t))\Bigg)
\end{aligned}$$

and accept σ_{t+1} with probability α

But why take log? Because, it helps with analytical precision, i.e. multiplying a thousand of small values may case an underflow in the system's memory (situation where a number is so close to zero that it cannot be represented accurately within the given precision of the system's floating point representation.). log is perfect solution because it transforms multiplication to additions and small positive numbers into non-small negative numbers.

Now, using Equation (4.13) as **prior**, Equation (4.14) as **jumping distribution**, Equation (4.16) as acceptance scam and 0.1 to be initial state we use Metropolis-Hastings Algorithm.

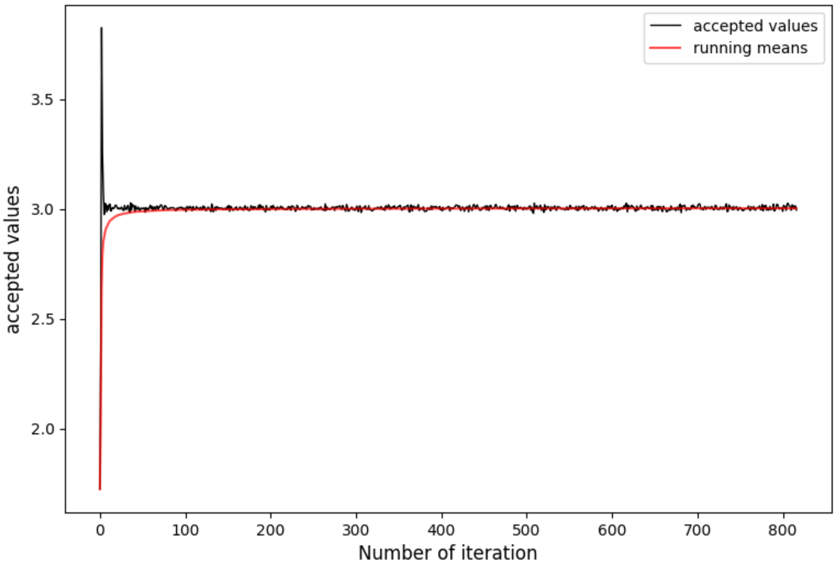

Figure 4.17: Accepted values from Metropolis-Hastings Algorithm

After running 100000 iteration we only accepted 817 values and values are looks like Figure 4.17. Here we can see we have to throw Burn-Ins (first 25% sample). After removing Burn-Ins we get our desire samples to estimate σ.

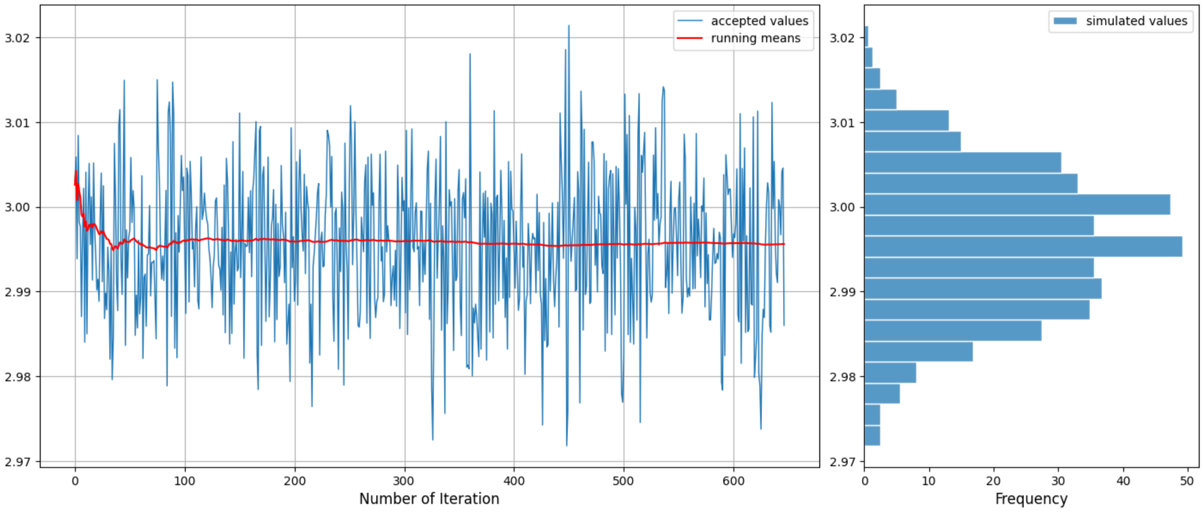

Figure 4.18: Accepted values after removing Burn-Ins

From this we get our desire values of σ to be 3.0032, variance of the samples to be 0.0001 and 0.05545 as Geweke Z-score. Because of low variance and Geweke Z-score less than 2 for simulated data we conform that it is very good estimate.

As a matter of fact I have taken observed values from N$(10, 9)$ distribution. So, we see the algorithm is very accurate to finding the means.

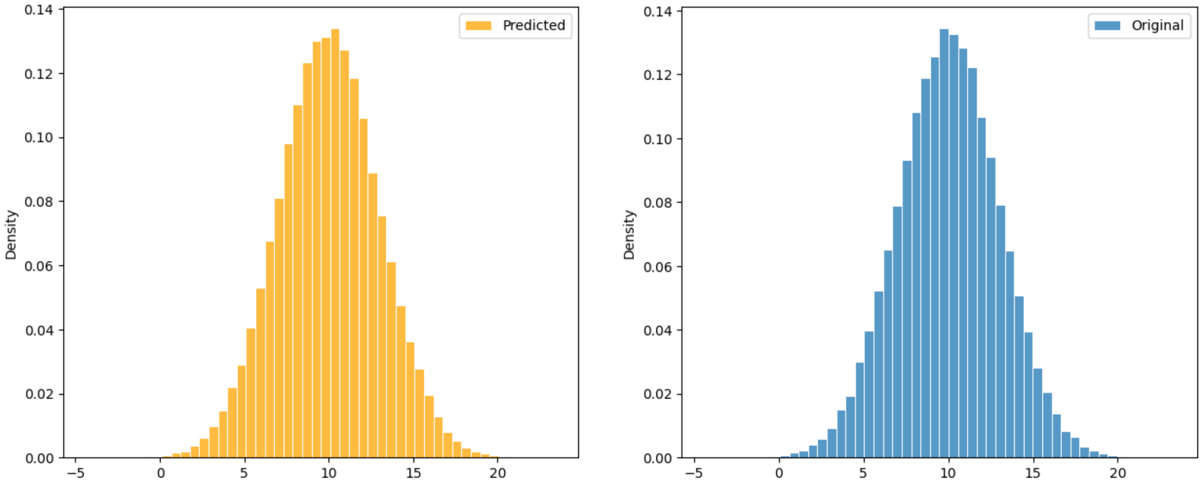

Figure 4.19: Original population with simulated samples

4.3 The Gibbs Sampler

The Metropolis-Hastings Algorithm of the previous section can be difficult to apply when the dimension of the state space is high. The generation of the chain becomes too much of a multidimensional problem and becomes at least unwieldy, and possibly undoable. Here Gibbs Sampler comes into play. The Gibbs Sampler is a spatial kind of Metropolis-Hastings Algorithm that very cleverly reduces the multidimensional problem into a sequence of *one-dimensional* problem.

Suppose a state $\mathbf{x}$ in the state specs S is a vector in some m-dimensional specs with $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_m)$. Suppose from current state $\mathbf{x}$ we want to jump to a new state $\mathbf{y} \in S$. According to Gibbs sampler we change coordinate one at a time, such as $(x_1, x_2, \ldots, x_m) \to (y_1, x_2, \ldots, x_m) \to (y_1, y_2, \ldots, x_m) \to \ldots \to (y_1, y_2, \ldots, y_m)$, and each coordinate change is made by using the conditional distribution of that coordinate given the rest of the coordinates. For example, the transition $(x_1, x_2, \ldots, x_m) \to (y_1, x_1, \ldots, x_m)$ is made by simulating from the distribution $f(x_1|x_2, \ldots, x_m)$. These conditional distribution of one coordinate gives all the rest are **full conditionals**. As, long as we can calculate and also simulate from all the full conditionals, a complicated multidimensional problem turns in to m one dimensional problems.

If current state $\mathbf{x} = (x_1, x_2, x_3, \ldots, x_m)$.Pick the coordinate to be changed at random from the m available coordinate. If the coordinate picked is i, then the state $\mathbf{y} = (x_1, x_2, \ldots, x_{j-1}, x, x_{j+1}, \ldots, x_m)$ work as a candidate state. Then Gibbs sampler uses the Metropolis-Hastings algorithm with

$$\begin{aligned} q(\mathbf{x}, \mathbf{y}) &= \frac{1}{m} P(X_i = x | X_j = x_j, i \neq j) \\ &= \frac{f((y))}{m P(X_j = x_j, i \neq j)} \end{aligned}$$

Now the acceptance probability

$$\begin{aligned} \alpha(\mathbf{x}, \mathbf{y}) &= \min\left(\frac{f(\mathbf{y}) q(\mathbf{y}, \mathbf{x})}{f(\mathbf{x}) q(\mathbf{x}, \mathbf{y})}, 1\right) \\ &= \min\left(\frac{f(\mathbf{y}) f(\mathbf{x})}{f(\mathbf{x}) f(\mathbf{y})}, 1\right) \\ &= 1 \end{aligned}$$

Hence, Gibbs sampler is a special Metropolis-Hastings algorithm whose acceptance probability is always 1.

4.3.1 Algorithm for Gibbs Sampler

Suppose $\mathbf{x} \in \mathbb{R}^m$. Then algorithm of The Gibbs sampler can be summarized as below:

1. Set initial values $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_m^{(0)})$.

2. Obtain a new value $\mathbf{x}^{(j)} = (x_1^{(j)}, x_2^{(j)}, \ldots, x_m^{(j)})$ form $x^{(j-1)}$ through *full conditional*

distributions

$$\begin{aligned}
x_1^{(j)} &\sim f(x_1|x_2^{(j-1)}, \dots, x_m^{(j-1)}),\\
x_2^{(j)} &\sim f(x_2|x_1^{(j)}, x_3^{(j-1)}, \dots, x_m^{(j-1)}),\\
&\vdots\\
x_m^{(j)} &\sim f(x_m|x_1^{(j)}, \dots, x_{m-1}^{(j)});
\end{aligned}$$

3. Change counter j to $j+1$ and return to step 2 until convergence is reached.

4.3.2 Examples

Now, we see some examples how to use Gibbs sampler.

Example 4.4 (Generating standard bivariate normal distribution). In this example we try to generate standard bivariate normal distribution with correlation coefficient ρ. Suppose,

$$(X_1, X_2) \sim \mathrm{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}\right)$$

Where, $\Sigma = \begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}$ is known as **covariance matrix**. Then, the probability density function of (X_1, X_2) would be,

$$f(x_1, x_2) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}(x^2 - 2\rho xy + y^2)\right)$$

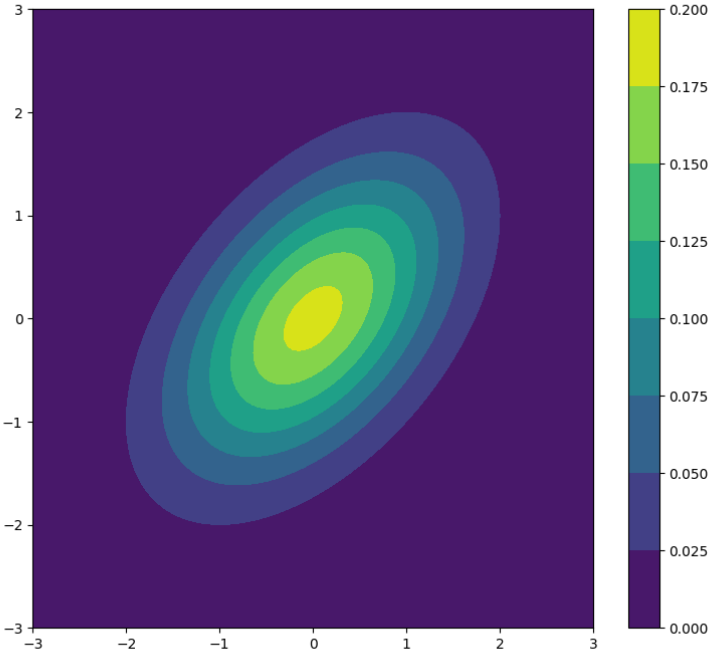

Figure 4.20: Contour Plot of Bivariate Normal Distribution when $\rho = 0.5$

For using Gibbs Sampler we have to calculate $f_{X_1|X_2}(x_1|x_2)$ and $f_{X_2|X_1}(x_2|x_1)$.

$$
\begin{aligned}
f_{X_1|X_2}(x_1|x_2) &= \frac{f(x_1,x_2)}{f_{X_2}(X_2)} \\
&= C_1 f(x_1,x_2) \\
&= C_2 \exp\left(-\frac{1}{2(1-\rho^2)}(x_1^2 - 2\rho x_1 x_2)\right) \\
&= C_3 \exp\left(-\frac{1}{2(1-\rho^2)}(x_1 - \rho x_2)^2\right)
\end{aligned}
$$

Recognizing this equation as a normal density, we can conclude that,

$$X_1|X_2 \sim \mathrm{N}(\rho x_2, (1-\rho^2))$$

Also,

$$
\begin{aligned}
f_{X_2|X_1}(x_2|x_1) &= \frac{f(x_2,x_1)}{f_{X_1}(X_1)} \\
&= C_4 f(x_2,x_1) \\
&= C_5 \exp\left(-\frac{1}{2(1-\rho^2)}(x_2^2 - 2\rho x_1 x_2)\right) \\
&= C_6 \exp\left(-\frac{1}{2(1-\rho^2)}(x_2 - \rho x_1)^2\right)
\end{aligned}
$$

Here also we can see that,

$$X_2|X_1 \sim \mathrm{N}\left(\rho x_1, (1-\rho^2)\right)$$

So, the Gibbs sampler algorithm for this case would be,

1. Set an initial state $\left(x_1^{(0)}, x_2^{(0)}\right)$
2. We obtain the next state $\left(x_1^{(t+1)}, x_2^{t+1}\right)$ through the full conditional distributions,

$$
\begin{aligned}
x_1^{(t+1)} &\sim \mathrm{N}\left(\rho x_2^{(t)}, (1-\rho^2)\right) \\
x_2^{(t+1)} &\sim \mathrm{N}\left(\rho x_1^{(t+1)}, (1-\rho^2)\right)
\end{aligned}
$$

Now choosing different initial state we study the outputs.

Initial State 1: Here we are taking initial state to be $(-1,-1)$.

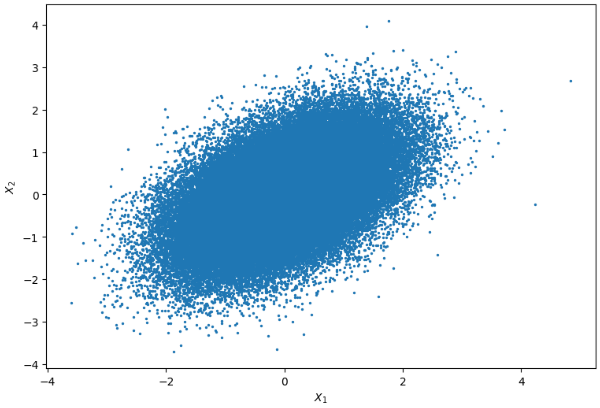

Figure 4.21: Samples when initial state $(-1,-1)$

Initial State 2: Now we consider $(0,0)$ (middle of the density) as an initial state.

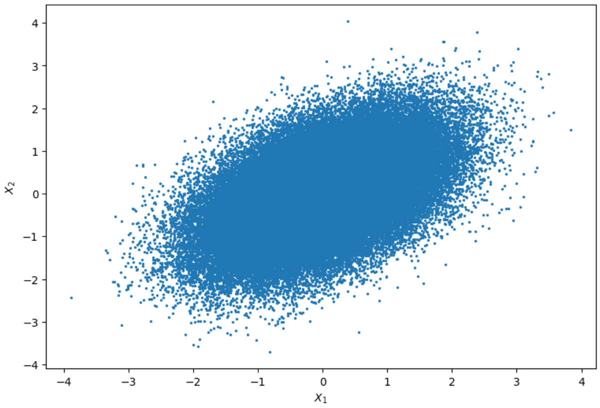

Figure 4.22: Samples when initial state $(0,0)$

Initial State 3: Here we take initial state way outside form middle of distribution consider $(-4,-4)$ as initial state. We can see from Figure 4.23, although we are taking initial state way outside, but the samples are come out to be same as above samples.

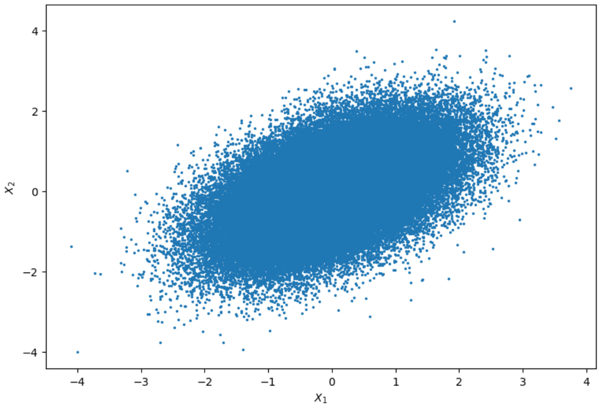

Figure 4.23: Samples when initial state $(-4,-4)$

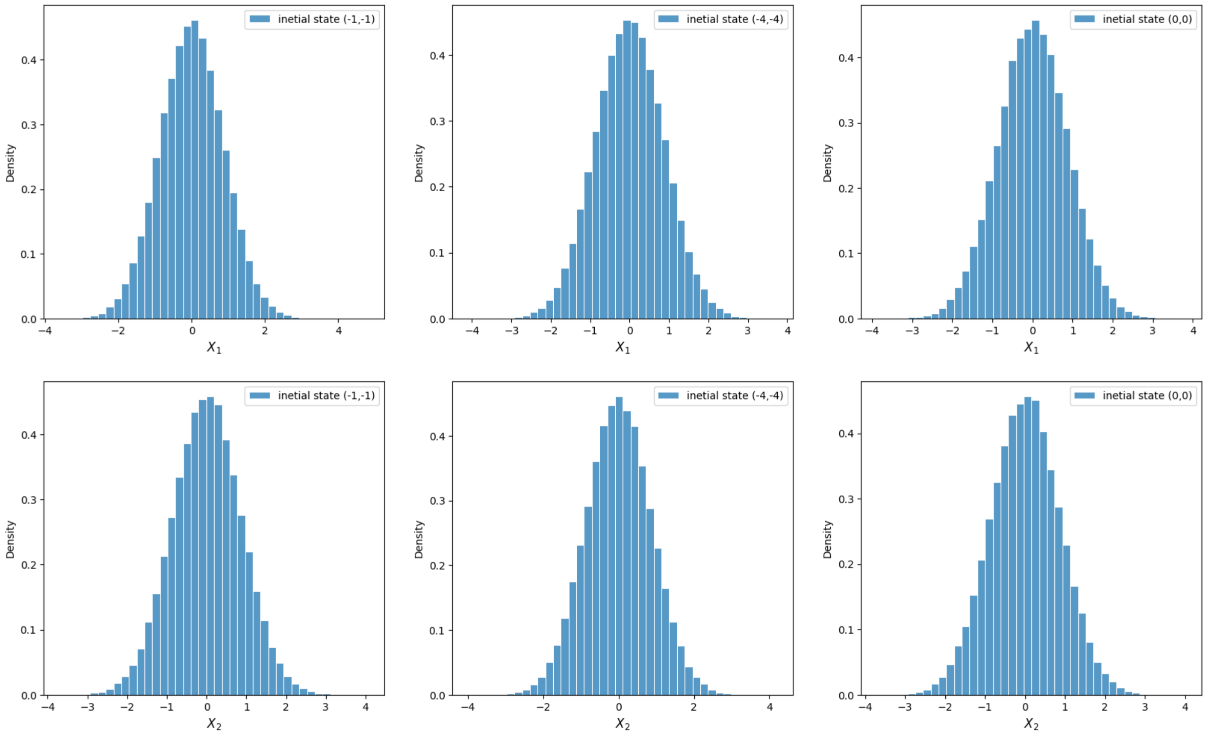

Figure 4.24: Marginal distribution for X_1 and X_2 for different initial states

Example 4.5 (Simulating Beta-Binomial Distribution). If $X|p \sim \text{Bin}(n, p)$ for some fix n, and $p \sim \text{Beta}(\alpha, \beta)$, then the marginal distribution of X would be called a **Beta-Binomial distribution**. Its probability density function,

$$f_X(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \binom{n}{x} \int_0^1 p^{x+\alpha-1}(1-p)^{n-x+\beta-1} dp \quad x = 0, 1, 2, \ldots, n$$

For simulating Beta-Binomial distribution using Gibbs sampler we have to simulate the pair (X, p) whose joint distribution is,

$$f_{(X,p)}(x, p) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \binom{n}{x} p^{x+\alpha-1}(1-p)^{n-x+\beta-1},\ x = 0, 1, 2, \ldots, n,\ 0 < p < 1.$$

We know form definition of Beta-Binomial distribution,

$$X|p \sim \text{Bin}(n, p)$$

Now,

$$\begin{aligned} f_{p|X}(p|x) &= \frac{f(x,p)}{f(x)} \\ &= \frac{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\binom{n}{x} p^{x+\alpha-1}(1-p)^{n-x+\beta-1}}{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\binom{n}{x}\int_0^1 p^{x+\alpha-1}(1-p)^{n-x+\beta-1}dp} \\ &= \frac{p^{x+\alpha-1}(1-p)^{n-x+\beta-1}}{\int_0^1 p^{x+\alpha-1}(1-p)^{n-x+\beta-1}dp} \\ &= \frac{1}{\text{Beta}(x+\alpha-1, n-x+\beta-1)} p^{x+\alpha-1}(1-p)^{n-x+\beta-1} \end{aligned}$$

Hence we have,

$$f_{p|X}(p|x) = \frac{\Gamma(\alpha+\beta+n)}{\Gamma(x+\alpha)\Gamma(n-x+\beta)} p^{x+\alpha-1}(1-p)^{n-x+\beta-1} \tag{4.17}$$

Form Equation (4.17) we conclude that $p|X \sim \text{Beta}(x+\alpha, n-x+\beta)$.
So, the Gibbs sampler algorithm for this example,

1. Choose an initial state $p^{(0)} \sim \text{Beta}(\alpha, \beta)$.
2. Obtain the next state $\left(x^{(t+1)}, p^{(t+1)}\right)$ through the full conditional distributions,

$$\begin{aligned} x^{(t+1)} &\sim \text{Bin}\left(n, p^{(t)}\right), \\ p^{(t+1)} &\sim \text{Beta}\left(x^{(t+1)}+\alpha, n-x^{(t+1)}+\beta\right) \end{aligned}$$

3. Replete step 2.

Now, taking $n = 10$, $\alpha = 7$, $\beta = 2$ we simulate beta-binomial distribution, and we get empirical mean 7.7797, variance 3.2788 which are close to theoretical mean(μ) and variance(σ^2).

$$\begin{aligned} \mu &= \frac{n\alpha}{\alpha+\beta} = \frac{10 \times 7}{7+2} = 7.7777 \\ \sigma^2 &= \frac{n\alpha\beta(\alpha+\beta+n)}{(\alpha+\beta)^2(\alpha+\beta+1)} = 3.2840 \end{aligned}$$

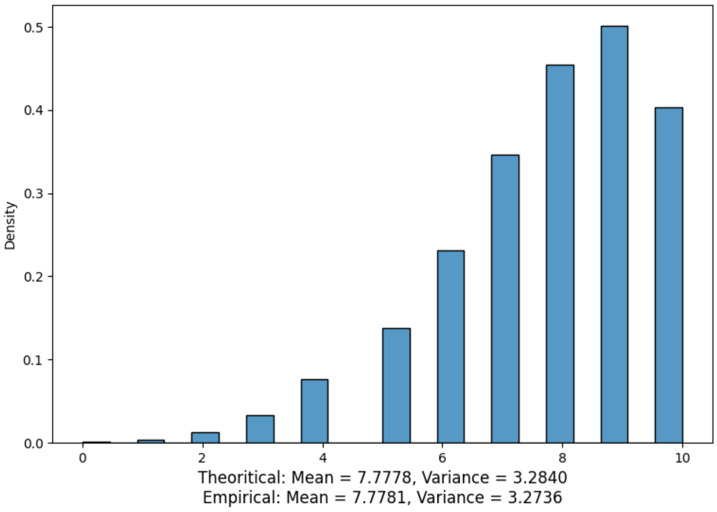

Figure 4.25: Simulation of Beta-Binomial(10,7,2)

Example 4.6 (Finding distribution of some observed data). In the Example 4.3 we have done a little cheating, here we did not find the mean(μ) in proper way, we assume the mean from the given observation, but this is not the proper way. In this example we try to fix our mistake.

Now, from **Bayesian Method** we have

$$f\left(\mu, \sigma^2 | d^n\right) \propto f\left(d^n | \mu, \sigma^2\right) f\left(\mu, \sigma^2\right) \tag{4.18}$$

The **likelihood function**,

$$
\begin{aligned}
f\left(d^n|\mu,\sigma^2\right) &= \prod_{i=1}^{n}\left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(d_i-\mu)^2}{2\sigma^2}\right)\right) \\
&= \frac{1}{\left(2\pi\sigma^2\right)^{\frac{n}{2}}}\exp\left(-\frac{\sum_{i=1}^{n}\left(d_i-\mu\right)^2}{2\sigma^2}\right) \qquad (4.19)
\end{aligned}
$$

Then Equation (4.18) becomes,

$$
f(\mu,\sigma^2|d^n) \propto \frac{1}{\left(2\pi\sigma^2\right)^{\frac{n}{2}}}\exp\left(-\frac{\sum_{i=1}^{n}\left(d_i-\mu\right)^2}{2\sigma^2}\right) f(\mu,\sigma^2) \qquad (4.20)
$$

Then, the question become how do we choose the **prior** in this case. We can consider the two possibilities,

1. We can define a joint prior distribution for μ and σ^2, i.e. we can define $f(\mu,\sigma^2)$, but it will very hard to accomplish.

2. Another way we can consider μ and σ^2 to be independent. i.e.,

   $$f(\mu,\sigma^2) = f_\mu(\mu)f_{\sigma^2}(\sigma^2)$$

   This is relatively easy to implement.

Here if we take conjugate prior[1] As our choice of prior it will be easy to sample from posterior distribution.

Now we have the conjugate prior for μ for normal likelihood function is Normal distribution.

$$
\begin{aligned}
f_\mu(\mu) &\propto \mathrm{N}\left(\theta,\sigma_\mu^2\right) \\
&\propto \frac{1}{\sqrt{2\pi\sigma_\mu^2}}\exp\left(-\frac{(\mu-\theta)^2}{2\sigma_\mu^2}\right)
\end{aligned}
$$

Where, θ and σ_μ^2 are **hyperparameters** of the prior distribution.

And for σ^2 we have two choices for conjugate prior for normal likelihood function.

1. One is Inverse-gamma distribution.

   $$
   \begin{aligned}
   f_{\sigma^2}(\sigma^2) &\propto \text{Inv-Gamma}(\alpha,\beta) \\
   &\propto \frac{\beta^\alpha}{\Gamma(\alpha)}\left(\frac{1}{\sigma^2}\right)^{(\alpha+1)}\exp\left(-\frac{\beta}{\sigma^2}\right)
   \end{aligned}
   $$

   with **hyperparameters** $\alpha > 0$, shape[2] parameter of $f_{\sigma^2}(\sigma^2)$ and $\beta > 0$, scale [3] parameter of $f_{\sigma^2}(\sigma^2)$

[1] **conjugate prior:** In Bayesian Inference, for a given likelihood function $f(x|\theta)$, if the posterior distribution $f(\theta|x)$ is in the same probability density family as the prior $f(\theta)$, then the prior and posterior are called conjugate distributions with respect to that likelihood function and the prior is called a conjugate prior for the likelihood function $f(x|\theta)$.

[2] shape parameters are those parameter those effect the shape of the distribution

[3] In family of probability distribution there is such parameter s(with other parameter θ) for which the CDF satisfies $F(x;s,\theta) = F(x/s;1,\theta)$ known as scale parameter

2. Another one is Scaled-Inverse-χ^2 distribution.

$$\begin{aligned} f_{\sigma^2}(\sigma^2) &\propto \text{Scale-Inv-}\chi^2\left(\nu, \sigma_0^2\right) \\ &\propto \frac{\left(\sigma_0^2\nu/2\right)^{\nu/2}}{\Gamma(\nu/2)} \frac{\exp\left(-\frac{\nu\sigma_0^2}{2\sigma^2}\right)}{\left(\sigma^2\right)^{1+\nu/2}} \end{aligned}$$

With hyperparameters ν and σ_0^2.

Now, if we choose prior of σ^2 to be Inv-Gamma(α, β) and μ to be N(θ, σ_μ^2) Equation (4.20) becomes,

$$\begin{aligned} f(\mu, \sigma^2|d^n) \propto & \frac{1}{\left(2\pi\sigma^2\right)^{\frac{n}{2}}} \exp\left(-\frac{\sum_{i=1}^n \left(d_i - \mu\right)^2}{2\sigma^2}\right) \times \\ & \frac{1}{\sqrt{2\pi\sigma_\mu^2}} \exp\left(-\frac{(\mu-\theta)^2}{2\sigma_\mu^2}\right) \times \left(\frac{1}{\sigma^2}\right)^{(\alpha+1)} \exp\left(-\frac{\beta}{\sigma^2}\right) \end{aligned} \tag{4.21}$$

For simulating Equation (4.21) we will use Gibbs sampler. To use Gibbs sampler we have to find the full conditionals $f(\mu|d^n, \sigma^2)$ and $f(\sigma^2|d^n, \mu)$.

Now,

$$\begin{aligned} f(\mu|d^n, \sigma^2) &\propto \exp\left(-\frac{\sum_{i=1}^n (d_i-\mu)^2}{2\sigma^2}\right) \times \exp\left(-\frac{(\mu-\theta)^2}{2\sigma_\mu^2}\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{\sum_{i=1}^n (d_i-\mu)^2}{\sigma^2} + \frac{(\mu-\theta)^2}{\sigma_\mu^2}\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{\sum_{i=1}^n d_i^2 - 2\mu\sum_{i=1}^n d_i + n\mu^2}{\sigma^2} + \frac{\mu^2 - 2\mu\theta + \theta^2}{\sigma_\mu^2}\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{\sigma_\mu^2\sum_{i=1}^n d_i^2 - 2\sigma_\mu^2\mu\sum_{i=1}^n d_i + \sigma_\mu^2 n\mu^2 + \sigma^2\mu^2 - 2\sigma^2\mu\theta + \sigma^2\theta^2}{\sigma^2\sigma_\mu^2}\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\frac{\left(n\sigma_\mu^2 + \sigma^2\right)\mu^2 - 2\mu\left(\sigma_\mu^2\sum_{i=1}^n d_i + \sigma^2\theta\right)}{\sigma^2\sigma_\mu^2}\right) \\ &\propto \exp\left(-\frac{1}{2}\frac{\mu^2 - 2\mu\left(\frac{\sigma_\mu^2\sum_{i=1}^n d_i + \sigma^2\theta}{n\sigma_\mu^2+\sigma^2}\right)}{\frac{\sigma_\mu^2\sigma^2}{n\sigma_\mu^2+\sigma^2}}\right) \\ &\propto \exp\left(-\frac{\left(\mu - \frac{\sigma_\mu^2\sum_{i=1}^n d_i + \sigma^2\theta}{n\sigma_\mu^2+\sigma^2}\right)^2}{2\left(\frac{\sigma^2\sigma_\mu^2}{n\sigma_\mu^2+\sigma^2}\right)}\right) \\ &\propto \text{N}\left(\frac{\sigma_\mu^2\sum_{i=1}^n d_i + \sigma^2\theta}{n\sigma_\mu^2+\sigma^2}, \frac{\sigma_\mu^2\sigma^2}{n\sigma_\mu^2+\sigma^2}\right) \end{aligned}$$

So, posterior of μ is a normal distribution with,

$$\text{mean } = \frac{\sigma_\mu^2\sum_{i=1}^n d_i + \sigma^2\theta}{n\sigma_\mu^2+\sigma^2} \text{ and variance } = \frac{\sigma_\mu^2\sigma^2}{n\sigma_\mu^2+\sigma^2}$$

Now if we take Inv-Gamma as prior of σ^2 we get,

$$\begin{aligned}
f(\sigma^2|d^n\mu) &\propto \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}} \exp\left(-\frac{\sum_{i=1}^n (d_i-\mu)^2}{2\sigma^2}\right)\left(\frac{1}{\sigma^2}\right)^{\alpha+1} \exp\left(-\frac{\beta}{\sigma^2}\right) \\
&\propto \left(\frac{1}{\sigma^2}\right)^{\alpha+\frac{n}{2}+1} \exp\left(-\frac{2\beta+\sum_{i=1}^n (d_i-\mu)^2}{2\sigma^2}\right) \\
&\propto \left(\frac{1}{\sigma^2}\right)^{\alpha+\frac{n}{2}+1} \exp\left(-\frac{\beta+\frac{\sum_{i=1}^n (d_i-\mu)^2}{2}}{\sigma^2}\right) \\
&\propto \text{Inv-Gamma}\left(\alpha+\frac{n}{2}, \beta+\frac{\sum_{i=1}^n (d_i-\mu)^2}{2}\right)
\end{aligned}$$

And now if we take Scale-Inv-χ^2 as prior of σ^2 we the posterior to be,

$$\begin{aligned}
f(\sigma^2|d^n,\mu) &\propto \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \exp\left(-\frac{\sum_{i=1}^n (d_i-\mu)^2}{2\sigma^2}\right)\left(\frac{1}{\sigma^2}\right)^{1+\frac{\nu}{2}} \exp\left(-\frac{\nu\sigma_0^2}{2\sigma^2}\right) \\
&\propto \left(\frac{1}{\sigma^2}\right)^{1+\frac{\nu+n}{2}} \exp\left(-\frac{\sum_{i=1}^n (d_i-\mu)^2+\nu\sigma_0^2}{2\sigma^2}\right) \\
&\propto \left(\frac{1}{\sigma^2}\right)^{1+\frac{\nu+n}{2}} \exp\left(-\frac{(\nu+n)\frac{\sum_{i=1}^n (d_i-\mu)^2+\nu\sigma_0^2}{\nu+n}}{2\sigma^2}\right) \\
&\propto \text{Scale-Inv-}\chi^2\left(\nu+n, \frac{\sum_{i=1}^n (d_i-\mu)^2+\nu\sigma_0^2}{\nu+n}\right)
\end{aligned}$$

Then, the Gibbs sampler algorithm can be summarized as follows,
When we take Inv-Gamma(α,β) as prior of σ^2,

1. Set an initial state $\left(\mu^{(0)}, (\sigma^2)^{(0)}\right)$. Were,

$$\begin{aligned}
\mu^{(0)} &\sim \text{N}(\theta, \sigma_\mu^2) \\
\left(\sigma^2\right)^{(0)} &\sim \text{Inv-Gamma}(\alpha,\beta)
\end{aligned}$$

2. We obtain the next state $\left(\mu^{(t+1)}, (\sigma^2)^{(t+1)}\right)$ by the full conditionals,

$$\begin{aligned}
\mu^{(t+1)} &\sim \text{N}\left(\frac{\sigma_\mu^2\sum_{i=1}^n d_i+(\sigma^2)^{(t)}\theta}{n\sigma_\mu^2+(\sigma^2)^{(t)}}, \frac{\sigma_\mu^2(\sigma^2)^{(t)}}{n\sigma_\mu^2+(\sigma^2)^{(t)}}\right) \\
(\sigma^2)^{(t+1)} &\sim \text{Inv-Gamma}\left(\alpha+\frac{n}{2}, \beta+\frac{\sum_{i=1}^n (d_i-(\mu)^{(t+1)})^2}{2}\right)
\end{aligned}$$

3. Repeat step 2.

Now, we know if $X \sim$ Inv-Gamma(α,β) then $\frac{1}{X} \sim$ Gamma(α,β). Therefore, sampling from Inverse-gamma is not different at all.

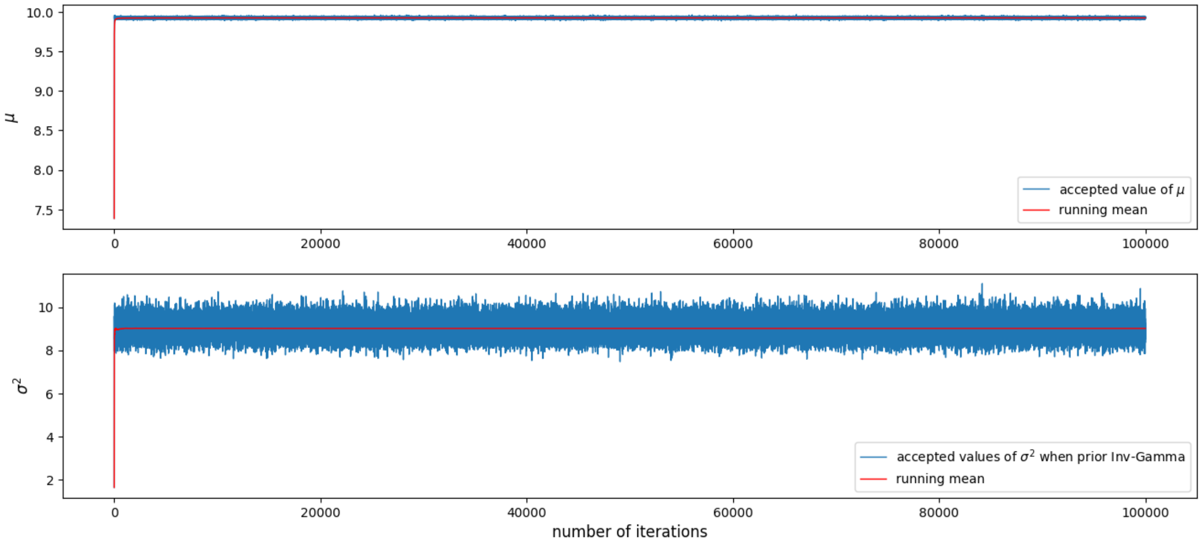

Figure 4.26: Accepted values of μ and σ^2 when prior of σ^2 is Inv-Gamma(α, β)

Now jugging form Figure 4.26 we have to remove first few samples as Burn-In. So after removing burn-in we get $\mu = 9.9230$ and $\sigma^2 = 9.0160$ which is close to original.

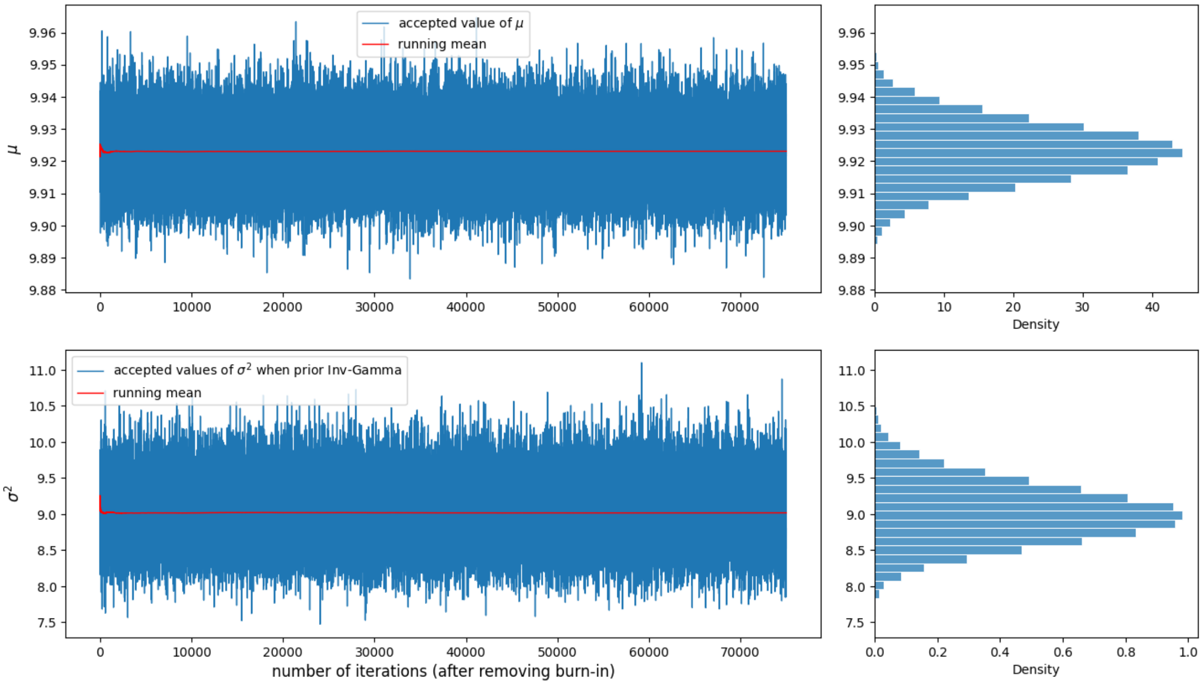

Figure 4.27: Accepted values of μ and σ^2 after removing Burn-In when prior of σ^2 is Inv-Gamma(α, β)

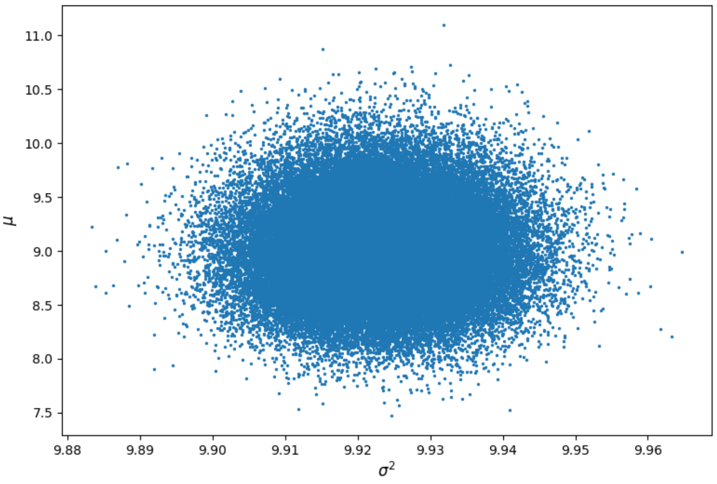

Figure 4.28: Scatter plot of μ and σ^2 together after removing Burn-In when prior of σ^2 is Inv-Gamma(α, β)

Now if we take Scale-Inv-$\chi^2(\nu, \sigma_0^2)$ as prior of σ^2, the algorithm become,

1. Set an initial state $\left(\mu^{(0)}, (\sigma^2)^{(0)}\right)$. Were,

$$\begin{aligned} \mu^{(0)} &\sim \mathrm{N}(\theta, \sigma_\mu^2), \\ \left(\sigma^2\right)^{(0)} &\sim \text{Scale-Inv-}\chi^2(\nu, \sigma_0^2) \end{aligned}$$

2. We obtain the next state $\left(\mu^{(t+1)}, (\sigma^2)^{(t+1)}\right)$ by the full conditionals,

$$\begin{aligned} \mu^{(t+1)} &\sim \mathrm{N}\left(\frac{\sigma_\mu^2 \sum_{i=1}^n d_i + (\sigma^2)^{(t)}\theta}{n\sigma_\mu^2 + (\sigma^2)^{(t)}}, \frac{\sigma_\mu^2 (\sigma^2)^{(t)}}{n\sigma_\mu^2 + (\sigma^2)^{(t)}}\right), \\ (\sigma^2)^{(t+1)} &\sim \text{Scale-Inv-}\chi^2\left(\nu + n, \frac{\sum_{i=1}^n (d_i - \mu^{(t+1)})^2 + \nu\sigma_0^2}{\nu + n}\right) \end{aligned}$$

3. Repeat step 2.

We know,

$$\begin{aligned} X &\sim \text{Scale-Inv-}\chi^2\left(\nu, \sigma_0^2\right) \\ \frac{X}{\nu\sigma_0^2} &\sim \text{Inv-}\chi^2(\nu) \\ \frac{\nu\sigma_0^2}{X} &\sim \chi^2(\nu) \\ \frac{1}{X} &\sim \frac{1}{\nu\sigma_0^2}\chi^2(\nu) \end{aligned}$$

So simulating Scale-Inv-χ^2 is also easy.

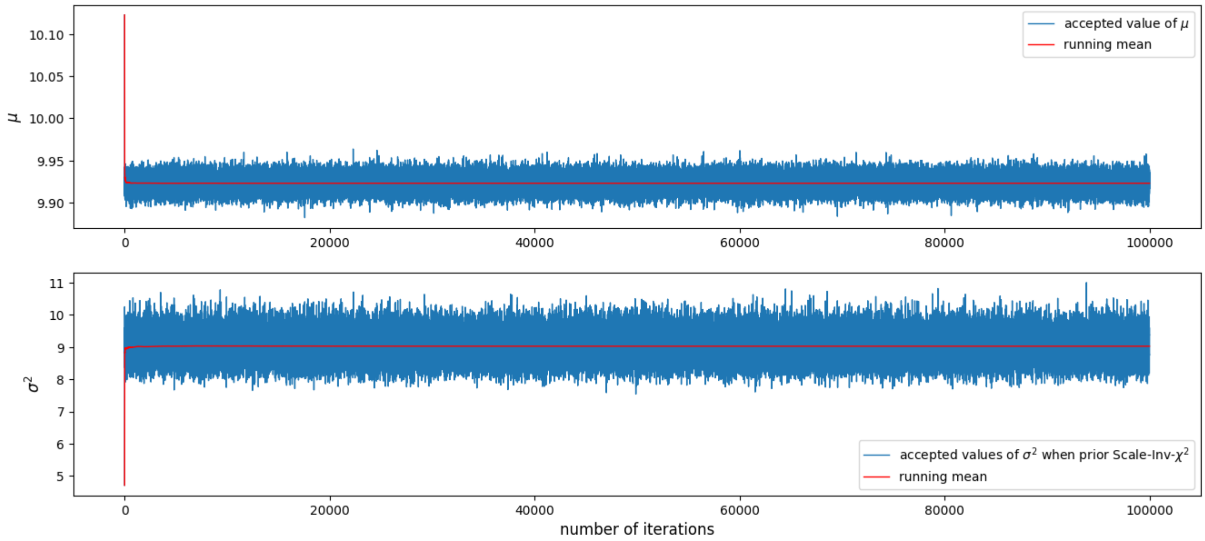

Figure 4.29: Accepted values of μ and σ^2 when prior of σ^2 is Scale-Inv-$\chi^2(\nu, \sigma_0^2)$

Here also we remove first 25% of term as Burn-In, and then we get μ to be 9.9230 and σ^2 to be 9.0258.

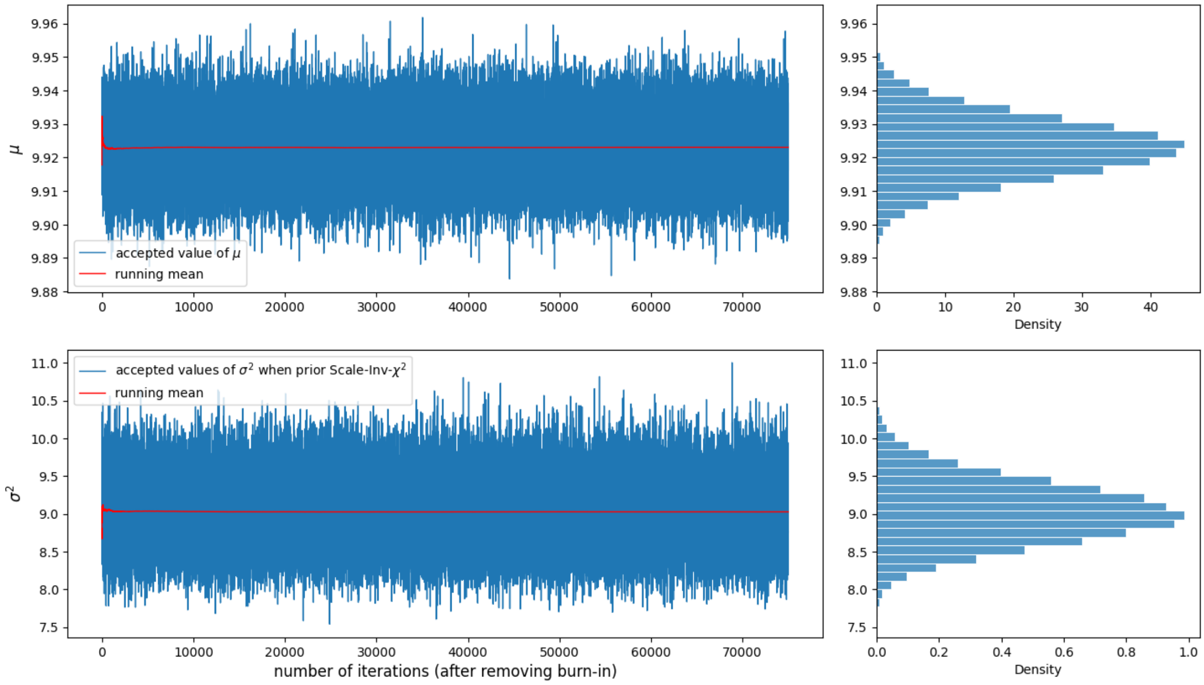

Figure 4.30: Accepted values of μ and σ^2 after removing Burn-In when prior of σ^2 is Scale-Inv-$\chi^2(\nu, \sigma_0^2)$

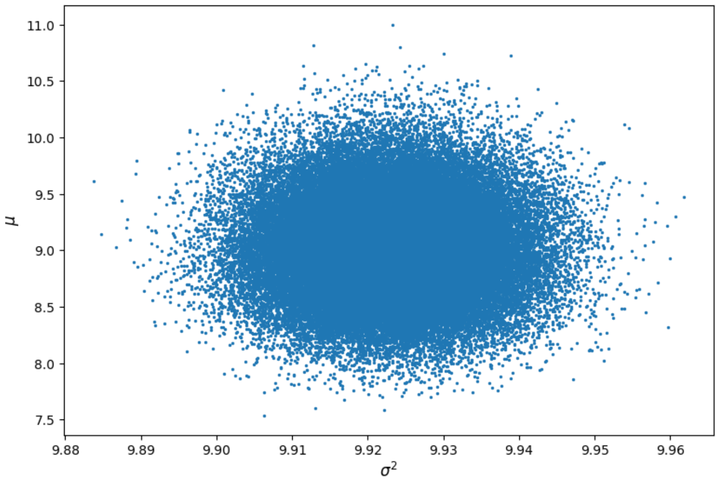

Figure 4.31: Scatter plot of μ and σ^2 together after removing Burn-In when prior of σ^2 is Scale-Inv-$\chi^2(\nu, \sigma_0^2)$

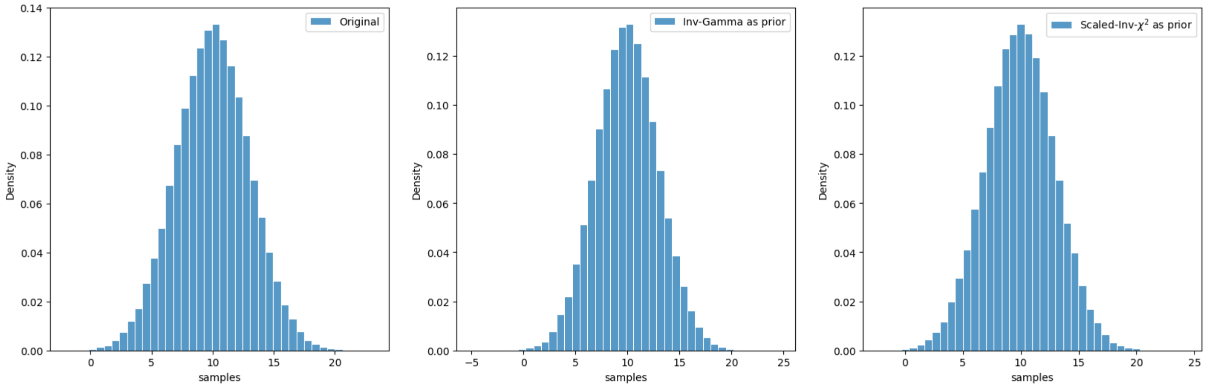

Figure 4.32: Comparison between original population and simulated population with different prior

Conclusion

In this project, I have discussed how we can simulate different kinds of random variables even from a unknown distribution function which are only partially known to us partially. Here, We use the inverse transform method for both discrete and continuous random variables, the accept-reject Method and the bivariate technique to simulate from known distribution function. Later, we use Importance Sampling for less known PDFs. We find out that although these methods are sufficient for generating random variables, but when it comes to generating random variables that are only known up to the normalizing constant, the previously mentioned methods truly struggle. So we introduce some new methods known as Markov Chain Monte Carlo (MCMC). The first MCMC method we have discussed here is the Metropolis-Hastings Algorithm (MH Algorithm), which help us to overcome the problem we have with general Monte Carlo Methods. Although, the HM Algorithm is a very sophisticated algorithm, but it is really difficult to sample multidimensional random variables. To tackle multi-variable PDFs, we introduce another algorithm named Gibbs Sampler. It is another version of Metropolis-Hastings algorithm, which is optimized for generating multidimensional random variables. With the help of MCMC methods and Bayesian Inference, we can solve a variety of difficult problems, like finding the parameters of a distribution for given observed samples.

Even if Gibbs Sampler looks like a very good algorithm but it has its own limitations, like we have to find the prior distributions, which is easily computable.

We can use MCMC method in the various domains of science like Bayesian statistics, computational physics, computational biology and computational linguistics

Bibliography

[1] S.M. Ross. *Simulation.* Elsevier Science, 2022.

[2] Anirban DasGupta. *Probability for statistics and machine learning: fundamentals and advanced topics.* Springer, 2011.

[3] D. Gamerman and H.F. Lopes. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition.* Chapman & Hall/CRC Texts in Statistical Science. CRC Press, 2006.

[4] Larry Wasserman. *All of Statistics: A Concise Course in Statistical Inference.* Springer Science & Business Media, December 2013. Google-Books-ID: qr-cuBAAAQBAJ.

[5] John Geweke and Forthcoming In. Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments. 4, November 1995.

[6] W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970. Publisher: [Oxford University Press, Biometrika Trust].

[7] Sheldon M Ross. A first course in probability. 2014.

[8] Joseph K. Blitzstein and Jessica Hwang. *Introduction to Probability, Second Edition.* Chapman and Hall/CRC, New York, 2 edition, February 2019.

[9] Burn-In is Unnecessary. `http://users.stat.umn.edu/~geyer/mcmc/burn.html#:~:text=Burn%2Din%20is%20only%20one,can%20be%20wrong%20with%20it%3F`.

[10] The Metropolis-Hastings algorithm. `https://blog.djnavarro.net/posts/2023-04-12_metropolis-hastings/`, April 2023.