Metropolis, Metropolis-Hastings and Gibbs Sampling Algorithms

by Nicholas Burke

A project submitted to the Department of Mathematical Sciences in
conformity with the requirements for Math 4301 (Honours Seminar)

Lakehead University Thunder Bay, Ontario

Abstract

This honours paper will discuss the Metropolis, the Metropolis-Hastings and the Gibbs sampling algorithms. These algorithms are classified as Markov Chain Monte Carlo methods. The purpose of these algorithms are to generate a sequence of random samples from a probability distribution to approximate a density of interest when direct sampling is difficult. This paper will include definitions, theorems, densities as well as R programming codes in relation to these various algorithms.

Acknowledgements

I would like to thank my supervisor Dr. Yin Chen and Dr. Razvan Ansica for their continual support and reassuring feedback during this process. I also want to thank everyone who has supported me throughout my undergraduate journey.

Contents

1 Introduction

The goal of this project is to depict the Metropolis, the Metropolis-Hastings and the Gibbs sampling algorithms functionality and applications in the field of mathematics.

A brief history of these algorithms, in 1953, the Metropolis algorithm was created by Greek-American physict Nicholas Metropolis along side Arianna W. Rosenbluth, Augusta H. Teller, and Edward Teller using only symmetrical proposal distributions. Seventeen years later, in 1970, W.K Hastings coined the Metropolis-Hastings algorithm, which extended to non-symmetrical proposal distributions. In 1984, the Gibbs sampling algorithm was created by Stuart and Donald Geman, this is a special case of the Metropolis-Hastings algorithm, which uses conditional distributions as the proposal distribution.

These three algorithms fall under the umbrella of Markov Chain Monte Carlo methods, more specifically Random Walk Monte Carlo methods. Markov Chain Monte Carlo methods are used to sample from a probability distribution by constructing a Markov Chain that has the same desired distribution as its equilibrium distribution. As the number of steps in the Markov chain increases the more the distribution of samples will resemble the actual desired distribution. The Metropolis, the Metropolis-Hastings and the Gibbs sampling algorithms are all Random Walk Monte Carlo methods. Each algorithm generates a random walk using a proposal distribution and an accept-reject criteria for particular steps. Typically these algorithms are used for sampling from a distribution when other sampling methods are not applicable, since the only requirement being the evaluation of the density function.

This paper is constructed as follows; chapter 2, will introduce some important basic definitions and various distributions in relation to probability theory. This will include plots of important density functions to help build a basis for this paper. Also, two very important concepts will be addressed the Central Limit Theorem and the Law of Large Numbers. This will help establish the reasoning on how the Metropolis-Hastings Algorithm is able to generate representative sampling of the function. Chapter 3 will divide into the mechanism of a Markov Chain Monte Carlo method, by defining what a Markov Chain is, the use of the Ergodic Theorem, and the general procedure of a MCMC method. Chapter 4 will conclude with the Metropolis, the Metropolis-Hastings, the Gibbs sampling algorithms being outline and used in various examples.

2 Preliminaries

In order to understand Markov Chain Monte Carlo methods, I will introduce some basic probability theory by defining important terms and concepts. This chapter will include various probability density functions that will reappear in this paper. To grasp the content of this project, one should have a solid understanding on certain concepts such as random variables, samples, mean, variance, probability distributions, densities and various other topics.

2.1 Definitions

Definition 2.1. *A population is a group of objects about which inferences can be made.*

Definition 2.2. *A sample is a subset of the population*

Definition 2.3. *The sample space of the experiment is the set of all possible outcomes.*

Definition 2.4. *A random variable X is an expression whose value is the outcome of a particular experiment. A random variable can be discrete by having a specific finite value, or continuous by taking any numerical value in an interval.*

Example 2.5. *In the game of basketball a player can take a shot at the basket. The shot is the discrete variable, a made shot can be denoted at 1, and a missed shot can be denoted as 0. A make or a miss are the only two possible outcomes.*

Definition 2.6. *A probability distribution is a function that describes all the possible values and likelihoods that a random variable can take within a given range.*

Example 2.7. *A basketball player has a probability of making a free throw say $p = 0.7$ and a probability of missing a free throw say $q = 1 - p = 0.3$. The probability distribution of the free throws is binomial*

$$b(n, p, j) = \binom{n}{k} p^j q^{n-j}$$

where n is the total number of free throws taken, j is the number of made free throws and $n - j$ is the number of free throws missed.

Definition 2.8. *The conditional probability of F given E is the new probability for an event F given that an event E has already occurred, denoted as $P(F|E)$.*

Example 2.9. *A basketball player shoots 80% from 2-point range and 30% from 3-point range. If he makes a 2-point shot k times and then makes a 3-point shot as many times as the successful shots from 2-point range.*
Let X be a made 2-point shot $\sim Bin(k, 0.8)$
Let Y be a made 3-point shot $\sim Bin(m, 0.3)$
Let N be the number of total shots taken
Then the conditional probability of making a three point shot given a made 2-point shot:

$$\sum_{k=m}^{N} P(Y=m|X=k) = \sum_{k=m}^{N} \binom{N}{k}(0.8)^k(1{-}0.8)^{N-k}\binom{k}{m}(0.3)^m(1{-}0.3)^{k-m}$$

Definition 2.10. *Let X be a discrete random variable with sample space Ω and distribution function m(x). The expected value E(X) is defined by*

$$E(X) = \sum_{x\in\Omega} xm(x)\ ,$$

denoted mean or μ

Definition 2.11. *Let $X_1, X_2, X_3, ..., X_n$ be a random sample from the distribution of X. Then the sample mean is denoted as*

$$\bar{X_n} = \sum_{i=1}^{n} \tfrac{X_i}{n}$$

Example 2.12. *The heights, in inches, of 12 players on a basketball team are 5' 9", 5' 9", 5' 6", 5' 8", 5' 11", 5' 5", 5' 7", 5' 6", 5' 6", 5' 7", 5' 10", and 6' 0". To calculate the average height (in inches) as follows:*

$$\tfrac{69+69+66+68+71+65+67+66+66+67+70+72}{12} = 67.9$$

This number can be interpreted as the expected value of a random variable. To see this, let an experiment consist of choosing one of the players at random, and let X denote their height. Then the expected value of X equals 67.9. It can also interpret this value as a sample mean, as this one team can be a sample from their respective league of teams.

Definition 2.13. *Let X be a real-valued random variable with density function f(x). The expected value μ=E(X) is defined*

$$\mu = E(X) = \int_{-\infty}^{\infty} xf(x)dx\ ,$$

Definition 2.14. *If F is any event and X is a random variable with sample space $\Omega = \{x_1, x_2, ...\}$, then the conditional expectation given F is by*

$$E(X|F) = \sum_j x_j \; P(X = x_j|F)$$

Definition 2.15. *Let X be a numerically valued random variable with expected value μ = E(X). Then the variance of X, denoted by V(X),*

$$V(X) = E((X)^2)$$

also denoted as σ^2.

Definition 2.16. *The standard deviation of X, denoted by D(X)*

$$D(X) = \sqrt{(V(X)}$$

also denoted as σ.

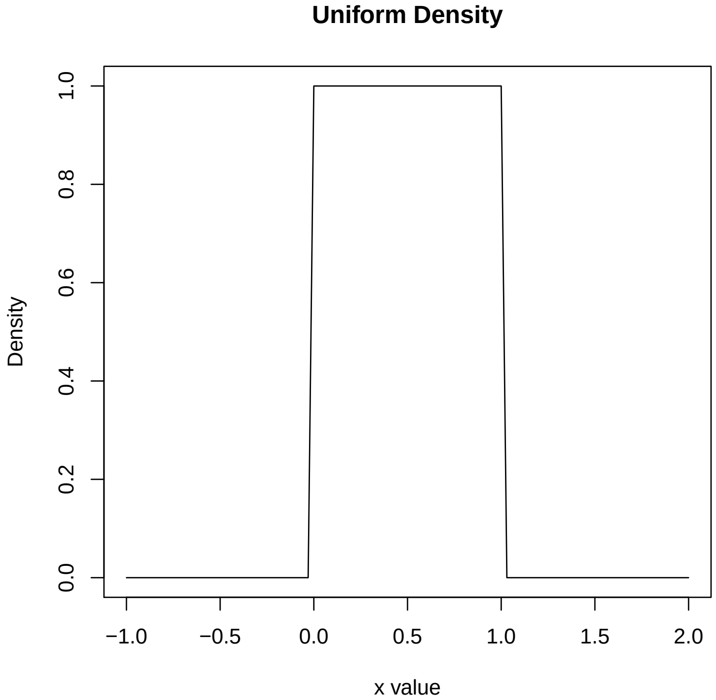
2.2 Densities

We will now discuss various probability distributions, along with their respective density function plots to help visualize these concepts.These densities can be used as proposal distributions for the Metropolis and the Metropolis-Hastings algorithms, they can also be generated using these same algorithms.

Continuous Uniform Density

The continuous uniform density function is the simplest density function. This density corresponds to an experiment in which all events of equal size are likely to occur. The continuous uniform density is important in generating a random value for the Metropolis and the Metropolis-Hastings algorithms. The following is the the equation for the uniform density function

$$f(\omega)= \frac{1}{b-a} \text{ if a} \leq \omega \leq b, \text{ 0 otherwise}$$



Uniform Distribution on the interval [0,1] with mean 1 and variance 0

Gamma Density

The gamma density has parameters λ and n, in the general case n can be any positive real number. It is often used as a conjugate prior distribution for various types of inverse scale parameters. The inverse gamma distribution can be used as the conjugate prior for the variance of a normal distribution. The expected value for the gamma density is

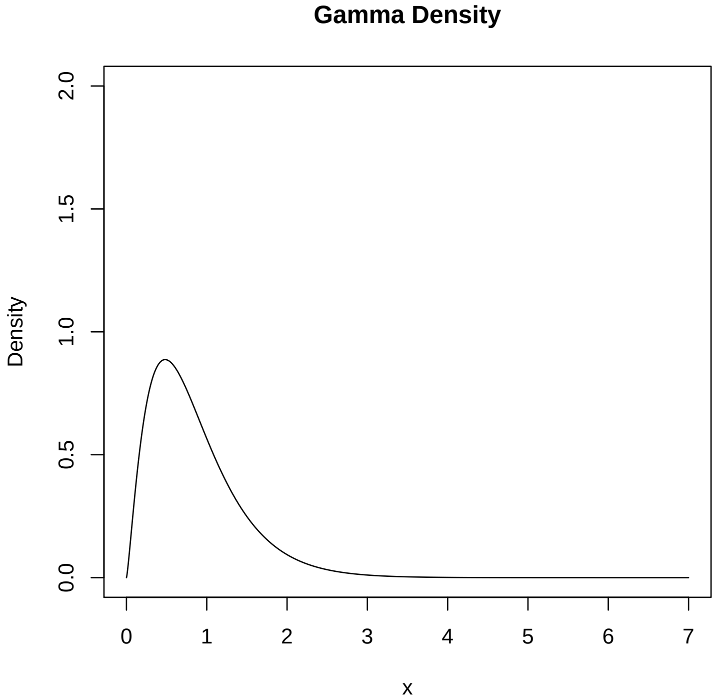
$$E(X) = \frac{\alpha}{\beta}$$

and the variance is

$$Var(X) = \frac{\alpha}{\beta^2}$$

The following is the the equation for the gamma density function:

$$g_n(x) = \lambda \frac{(\lambda x)^{n-1}}{(n-1)! e^{-\lambda x}} \text{ if } x > 0, \text{ 0 otherwise}$$



Gamma Distribution with Alpha=2.3 and Beta=2.7

Beta Density

The beta density is the conjugate prior probability density for the Bernoulli, binomial, negative binomial and geometric densities. It has both discrete and continuous coordinates, with two parameters α ,β defined by

$$B(\alpha,\beta,x)=\tfrac{1}{B(\alpha,\beta)}\ x^{\alpha-1}(1-x)^{\beta-1}, \text{ if } 0 \leq x \leq 1 \ , \\ 0 \text{ otherwise}$$

The expected value for the beta density is

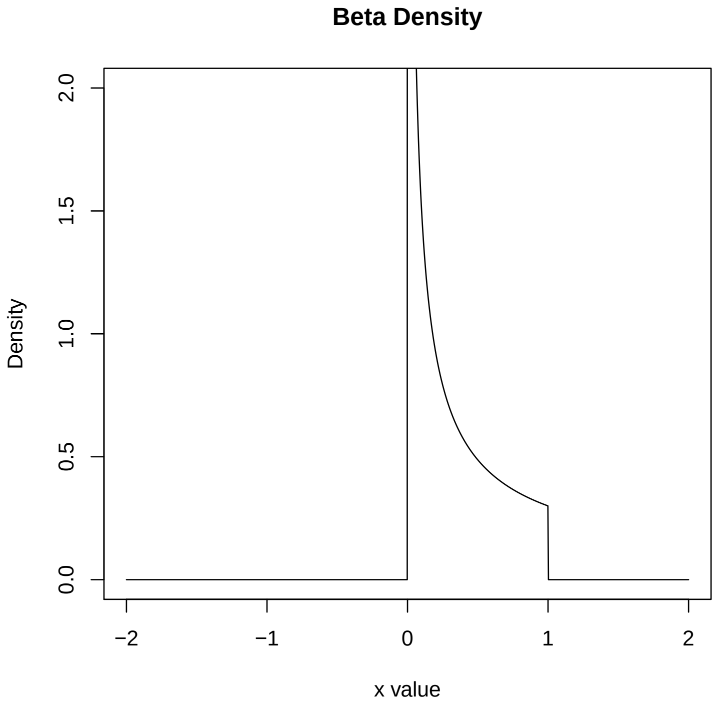
$$E(X) = \tfrac{\alpha}{\alpha+\beta}$$

and the variance is

$$Var(X) = \tfrac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Here α and β are any positive numbers, and the beta function $B(\alpha,\beta)$ is given by the area under the graph of $x^{\alpha-1}(1-x)^{\beta-1}$ between 0 and 1:

$$B(\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx$$



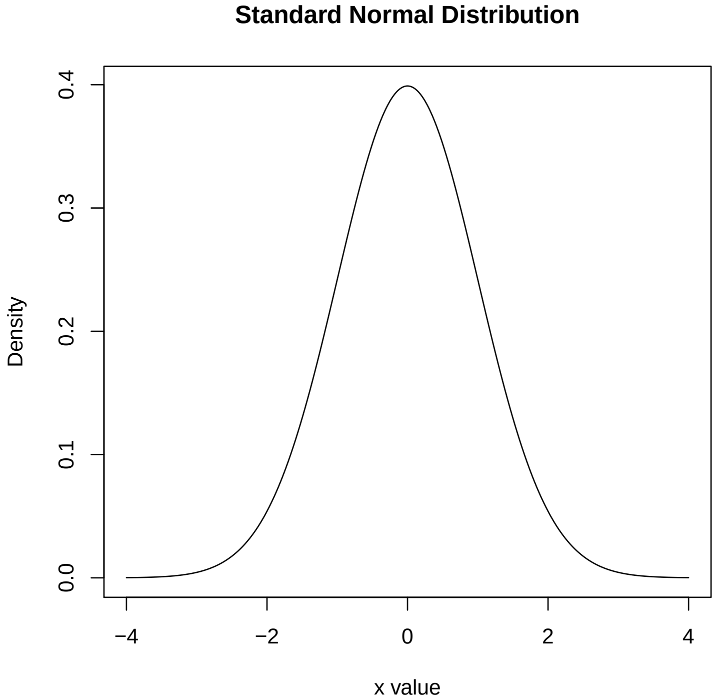
Beta Density with alpha=0.3, beta=0.7

Normal Density

The normal density is one of the more important densities in probability, its properties can relate to various theorems and processes. The normal distribution is symmetric which makes it an ideal proposal density for the Metropolis Algorithm.
The equation for the normal density function is as follows

$$f_x(x) = \frac{1}{2\pi\sigma} e^{\frac{-x-\mu^2}{2\sigma^2}}$$

where μ represents the ”center” of the density and σ represents the ”spread” of the density.



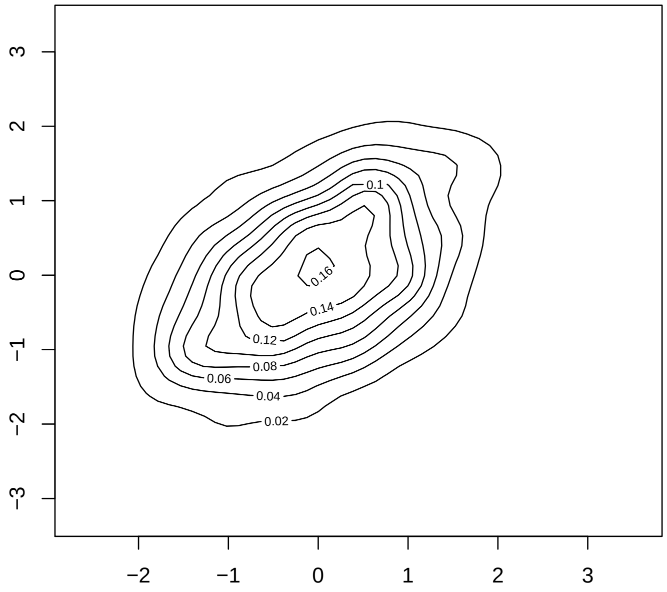
Standard Normal Density with mean=0 and variance=1

Bi-variate Normal density

Two random variables X and Y are said to be bi-variate normal, if $aX + bY$ has a normal distribution for all a,b $\in \Re$.
The equation for the bi-variate normal distribution function is as follows:

$$f_{XY}(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}}$$
$$\exp\{-\frac{1}{2(1-\rho^2)}[(\frac{x-\mu_X}{\sigma_X})^2 + (\frac{y-\mu_Y}{\sigma_Y})^2 - 2\rho\frac{(x-\mu_x)(y-\mu_Y)}{\sigma_X\sigma_Y}]\}$$

where $\mu_X, \mu_Y \in \Re, \sigma_X, \sigma_Y > 0$ and $\rho \in (-1, 1)$ are all constants



The bi-variate normal distribution can be used to demonstrate how the Gibbs sampling algorithm works, since it's multi-variate with a correlation between each variable.

Now to introduce two very important facts to help solidify the foundation for this project. The Law of Large Numbers is a fundamental theorem of probability which is also referred to as the Law of Averages, is consistent for both continuous and discrete random variables. The Central Limit Theorem is another fundamental theorem of probability that tells us what happens when we have the sum of a large number of independent random variables each of which contributes a small amount to the total.

2.3 Law of Large Numbers

The Law of Large Numbers implies that as a sample size grows the sample mean gets closer to the average of the whole population.

The Law of Large Numbers for discrete random variables

Theorem 2.17. *Let $X_1, X_2 ... X_n$, be an independent trails process with finite expected value μ and finite variance σ^2.*
Let $S_n = X_1 + X_2 + ... + X_n$. Then for any $\epsilon > 0$,

$$P(|\frac{S_n}{n} - \mu| \geq \epsilon) \to 0 \text{ as } n \to \infty$$

$$P(|\frac{S_n}{n} - \mu| < \epsilon) \to 1 \text{ as } n \to \infty$$

where $\frac{S_n}{n}$ denotes the average of individual outcomes.

The Law of Large Numbers has a natural analogue for continuous random distributions.

The Law of Large Numbers for continuous random variables,

Theorem 2.18. *Let $X_1, X_2 ... X_n$, be an independent trails process with continuous density function f, finite expected value μ and finite variance σ^2*
Let $S_n = X_1 + X_2 + ... + X_n$ then for an $\epsilon > 0$,

$$\lim\ P(|\frac{S_n}{n} - \mu| \geq \epsilon) = 0 \text{ as } n \to \infty$$

$$\lim P(|\frac{S_n}{n} - \mu| < \epsilon) = 1 \text{ as } n \to \infty,$$

where $\frac{S_n}{n}$ denotes the average of individual outcomes.

Example 2.19. *Consider the career free throw percentage of the basketball player to be 75% . Suppose that the player makes only 33 of their next 50 throws resulting in a relative frequency of 0.66, the more free throws that he attempts, the closer his relative frequency will get to 0.75. Suppose he were to make exactly 75 of his next 100 free throws that he takes. This probability one would predict to occur, then his relative frequency would be 108/150, or 0.72 making it much closer to 0.75. If he makes 75 of the next 100 free throws, then his relative frequency becomes 183/250, or 0.732, and it will keep getting closer from there. The Law of Large Numbers holds that with an increasing number of trials, the cumulative relative frequency of an event will tend closer and closer to the actual probability of the event.*

This can be categorized as a Bernoulli experiment, and we can expect that in a large number of repetitions the proportion of times that the event will occur to be near p.

2.4 Central Limit Theorem

The Central Limit states under very general conditions, if we sum a large number of mutually independent random variables then the distribution of the sum can be closely approximated by the normal density.

Theorem 2.20. *Let $X_1, X_2, ..., X_n$ be a sequence of independent random variables. Let $S_n = X_1 + X_2 + ... + X_n$ for each n, the mean is μ_n and the variance σ_n^2. The mean and the variance of S_n to be m_n and s_n^2 as $s_n \to \infty$.If there exists a constant A, such that $|X_n| \geq$A for all n,then for a<b*

$$\lim P(a < \tfrac{S_n - m_n}{s_n} < b) = \tfrac{1}{\sqrt{2\pi}} \int_a^b e^{\frac{-x^2}{2}} dx \ \textit{as } n \to \infty$$

Example 2.21. *A basketball player shoots 80 percent from the free throw line on average. We can use the Central Limit Theorem to determine the probability that he will shoot 85 percent from the free throw line on 100 attempts. We will assume that each free throw is an independent event. $n = 100$ and each X_i is a Bernoulli random variable with $\mu = 0.80$ and $\sigma = 0.16$ Applying the Central Limit Theorem*

$$P(\sum_{i=1}^{n} X_i > 85) = P(\sum_{i=1}^{n} X_i > 85.5) = P(\tfrac{\sum_{i=1}^{n} X_i - 80}{4} \geq \tfrac{85.5-80}{4}) \approx 1 - \Phi(1.375) \approx 0.085$$

The Law of Large Numbers and the Central Limit Theorem are imperative to Markov Chain Monte Carlo methods. These principles explain how samples converge and can be manipulate to densities that are desirable. The Law of Large Numbers is essential to the Ergodic theorem, the main concept on why Markov Chain Monte Carlo works.

3 MCMC

Markov Chain Monte Carlo methods are iterative sampling methods that allow sampling from a posterior density. The main goal is to estimate the mean

$$\mu = E_\pi f(X).$$

By constructing a Markov chain that has a stationary distribution $\pi = \pi P$. Using the Law of Large Numbers it can generate a sample that would be representative of the population mean that we are trying to find. In order to grasp how Markov Chain Monte Carlo methods work, I will define what a Markov Chain is and how the samples are then generated.

3.1 Markov Chains

Definition 3.1. *A process in which the outcome of a given experiment can affect the outcome of the next experiment is called a Markov Chain.*

Definition 3.2. *Let $S = (s_1, s_2, ..., s_r)$ be a set of states. If the chain is in state s_i, then moves to state s_j at the next step with probability p_{ij} which is only dependent upon the current state alone.*

Definition 3.3. *At any time n, the next state X_{n+1} is conditionally independent of the past $X_0, ..., X_{n-1}$ given the present state X_n. This is called the Markov Property.*

Definition 3.4. *A stochastic process is a sequence of events in which the outcome at any stage depends on some probability.*

Definition 3.5. *A Markov process is a stochastic process with:*

1. *The number of possible outcomes or states is finite.*
2. *The outcomes at any stage depends only on the outcome of the previous stage.*
3. *The probabilities are constant over time.*

Transition Matrix

The matrix of transition probabilities or the transition matrix is a square array of p_{ij} probabilities denoting the probability of each step occurring. If a Markov chain has r states, then

$$p_{ij} = \sum_{k=1}^{r} p_{ik} p_{kj}.$$

$$P = \begin{pmatrix} p_{11} & \cdots & p_{1k} \\ p_{21} & \cdots & p_{2k} \\ \vdots & \ddots & \vdots \\ p_{k1} & \cdots & p_{kk} \end{pmatrix}$$

Definition 3.6. *A transition matrix is irreducible or ergodic if it is possible to go from every state to every state.*

Definition 3.7. *An irreducible Markov chain to be aperiodic is that there exist a state i such that $P(i,i) > 0$.*

Definition 3.8. *A state i is recurrent if, when the Markov chain is started out in state i, the chain is certain to return to i at some finite future time.*

Definition 3.9. *A stationary distribution of a Markov chain is a probability distribution that remains unchanged in the Markov chain as time progresses. Typically, it is represented as a row vector π whose entries are probabilities summing to 1 , and given transition matrix P ,it satisfies.*

Definition 3.10. *Regardless of the starting value i in the chain, the long run proportion of time the chain i, the long run proportion of time the chain spends in state j equals $\pi(j)$, for every possible state j. Then π is called a limiting distribution.*

Example 3.11. *A basketball player has a particular free throw percentages in the following scenarios:*

1/2 if he misses the previous two free throws

2/3 if made one of his last two free throws

3/4 if he has made both his last free throws

These probabilities and event can be modeled as a Markov Chain, and the limiting fraction if the time he makes a free throw can be calculated.
Let X_n represent a made free throw at time n
Let $Y_n = (X_{n-1}, X_n)$ represent the last two free throws taken.
There are 4 different states:

(0,0)- two missed free throws

(0,1)- a missed free throw followed by a made free throw

(1,0)- a made free throw followed by a missed free throw

(1,1,)- two consecutive made free throws

The transition matrix is the following

$$P = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 \\ 1/3 & 0 & 2/3 & 0 \\ 0 & 1/3 & 0 & 2/3 \\ 0 & 1/4 & 0 & 3/4 \end{pmatrix}$$

This is an irreducible, positive recurrent, aperiodic chain. The stationary distribution:

$$\pi(0,0) = 1/8,\ \pi(0,1) = \pi(1,0) = 3/16,\ \pi(1,1) = 1/2.$$

The proportion of the time that he makes one free throw is equal to the limit of the probability that he makes a free throw at a given time is equal to

$$\lim_{n\to\infty} P(X_n = 1)\ = \pi(1,0) + \pi(1,1) = 11/16.$$

All Markov chains used for Markov Chain Monte Carlo purposes need to have a unique stationary and limiting distribution.

Ergodic Theorem

The Ergodic Theorem is very important in Markov Chain Monte Carlo, this theorem is the Law of Large of Number in reference to Markov Chains. This Here is a formal definition of the theorem.

Theorem 3.12. *If $(X_0, X_1, ..., X_n)$ is an irreducible discrete Markov Chain with stationary distribution π, then*

$$\tfrac{1}{n}\sum_{i=1}^{n} f(X_i) \to Ef(X) \text{ as } n\to \infty \text{ where } X \sim \pi,$$

for any bounded function $f : X \to \Re$.

3.2 Importance Sampling

Importance sampling is used to estimate the properties of a specific distribution using only samples that are generated from a different distribution than the distribution of interest. In order to find an estimation of the expectation in regards to the distribution of interest let's say f we will need to compute the weighted average of these draws. Want to find

$$\mu = E(f(x)) = \textstyle\int f(x)p(x)dx = \displaystyle\int \frac{f(x)p(x)}{q(x)} \approx \frac{1}{n}\sum_{i=1}^{n} \frac{f(X_i)p(x_i)}{q(x_i)}$$
$$\forall q \text{ such that } q(x) = 0 \Rightarrow p(x) = 0$$

where p is the nominal distribution and q is the importance distribution The distribution p is hard to sample from where as the distribution q is easier to sample from. The likelihood ratio is $\frac{p(x)}{q(x)}$.
General Strategy:

- Sample n variables $x_1, x_2, .., x_n \sim q$ from $q(x)$
- For each variable define $w_i = \frac{p(x)}{q(x)}$
- The importance weighted estimator is $\hat{\mu} = \frac{\sum_{i=1}^{n} w_i(x_i)}{\sum_{i=1}^{n} w_i}$

Importance sampling does not work very well in higher dimensions.

R Output

This is an example of importance sampling used to estimate

$$\int_1^0 x^2$$

With beta being the importance distribution and the weighted estimator being the uniform density over the beta density. Using with 1,000,000 samples generated by R.

```
> w <- function(x) dunif(x,0,1)/dbeta(x,0.3,1)
> f <- function(x) x^(2)
> X <- rbeta(1000000,0.3,1)
> Y <- w(X)*f(X)
> c(mean(Y),var(Y))
```

```
[1] 0.3334128 0.4735536
```

Based on this output it demonstrates that importance sampling is a good approximation for the integral. The real value of the integral is

$$\int_1^0 x^2 = \frac{1}{3}$$

Monte Carlo

Monte Carlo is sampling from a distribution, to estimate quantities whose exact values are difficult to calculate exactly. It uses random numbers to simulate complicated sequences. The Monte Carlo estimator for μ is:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} f(X_i) \text{ where } (X_i) \overset{iid}{\sim} \pi, i = 1, ..., n$$

where $\pi(x)$ is the uniform distribution on the unit square

3.3 Markov Chain Monte Carlo

The goal of a March Chain Monte Carlo is to sample from a distribution p or approximate

$$E(f(x)) \approx \frac{1}{n}\sum_{i=1}^{n} f(X_i)\ ,\ X_i \overset{iid}{\sim} p$$

where p is a complicated distribution.
Markov Chain Monte Carlo is sampling using local information that is available. Markov chain Monte Carlo methods are primarily used for calculating numerical approximations of multi-dimensional integrals. These methods sample from a probability distribution based on the construction a Markov chain that has the desired distribution as its equilibrium distribution. The state of the chain after a number of steps is then used as a sample of the desired distribution. The quality of the sample improves as a function of the number of steps that occur.
General Strategy:

- Current position $X_t = x$
- Propose a candidate $y \sim q(x, .)$
- with probability $\alpha(x, y)$ accept y: $X_{t+1} = y$
- otherwise stay where you are:$X_{t+1} = x$

The Markov chain becomes a random walk if we have a symmetric proposal density q, this is more evident in the Metropolis Algorithm.

Definition 3.13. *Let $\{X_k\}_{k=1}^{\infty}$ be a sequence of independent, identically distributed discrete random variables. For each positive integer n, we let S_n denote the sum $X_1 + X_2 + ... + X_n$. The sequence $\{S_n\}_{n=1}^{\infty}$ is called a random walk.*

4 Metropolis, Metropolis-Hastings and Gibbs

4.1 Metropolis Algorithm

The Metropolis Algorithm is considered to be a top 10 sampling algorithm in the statistical field today. The goal of this algorithm is to sample from a distribution π to approximate

$$E_\pi f(X), \text{ where } X \overset{iid}{\sim} \pi$$

Construct a Markov Chain with a stationary distribution, so its easy to sample from then, apply the Ergodic Theorem. Run the Markov chain until stationary then all subsequent samples are from the stationary distribution.
General Strategy:

- Choose symmetric proposal distribution Q ($Q_{ab} = Q_{ba}$)
- Initialize at $x_0 \in X$
- for $i = 0, 1, 2, ..., n-1$
- sample x from $Q(x_i, x)$
- sample μ from uniform (0,1)
- accept or reject proposal based on:if $\mu < \frac{\pi(x)}{\pi(x_i)}$ then $x_{i+1} = x$, else $x_{i+1} = x_i$
  Output $x_0, x_1, ..., x_n$ sample after run n times

R Output

This is an example of using the Metropolis algorithm with 10,000 samples from the uniform proposal distribution $U(-1, 1)$ to generate the

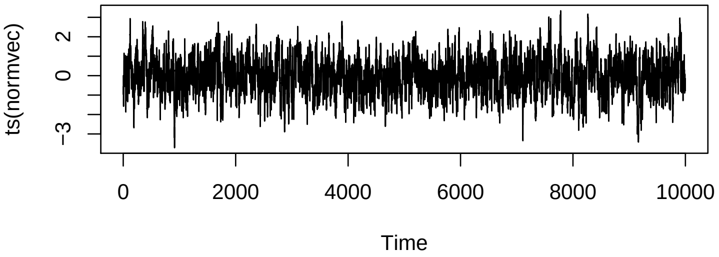
Normal distribution $N(0, 1)$

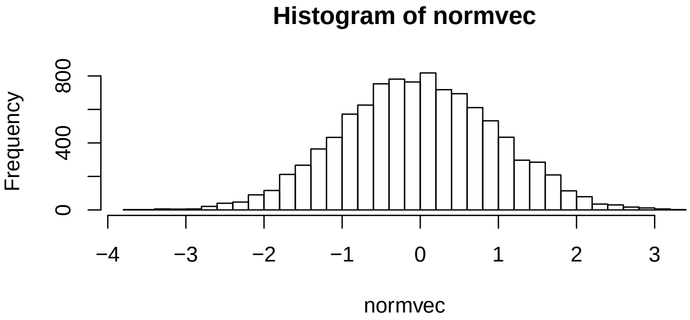
The plots below show that the Markov chain is well mixing and that there is independence among the samples generated. Also a histogram of the samples resembling the normal density curve.
R Code:

```
> norm<-function (n, alpha)
+ {
+   vec <- vector("numeric", n)
+   x <- 0
+   vec[1] <- x
+   for (i in 2:n) {
+     rv <- runif(1,-alpha,alpha)
+     candidate <- x + rv
+     accept <- min(1, dnorm(candidate)/dnorm(x))
+     u <- runif(1)
+     if (u < accept)
+       x <- candidate
+     vec[i] <- x
+   }
+   vec
+ }
> normvec<-norm(10000,1)
> c(mean(normvec),var(normvec))

[1] -0.02567952  0.97362265

> par(mfrow=c(2,1))
> plot(ts(normvec))
> hist(normvec,30)
> par(mfrow=c(1,1))
>
```


ts(normvec)
2
0
−3
0
2000
4000
6000
8000
10000
Time


Histogram of normvec
Frequency
800
400
0
−4
−3
−2
−1
0
1
2
3
normvec

4.2 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm is used to obtain a sequence of random samples from a probability distribution for which direct sampling is hard. The goal of this algorithm is still the same, to sample from a distribution π to approximate

$$E_\pi f(X), \text{ where } X \overset{iid}{\sim} \pi$$

This sequence can be used to approximate the distribution or to compute an integral. This algorithm generates a random walk using a proposal density and a method for rejecting some of the proposed moves. The Metropolis-Hastings algorithm is a more generalized version of the Metropolis algorithm. This algorithm can draw samples from any probability distribution that necessarily symmetric provided that the value of the function $f(x)$ can be computed. This proposal distribution must be proportional to the density of the probability distribution of interest.
General Strategy:

- Initialize at x_0, set $i = 0$
- Sample $x \sim q(x|x_i)$ from the proposal distribution
- Calculate acceptance probability $P(x|x_i)$=$min = \{1, \frac{\pi(x)q(x_i|x)}{\pi(x_i)q(x|x_i)}\}$ (moving from current state x_i to proposed state x)
- Sample μ from uniform [0,1]
- if $\mu \leq P(x|x_i)$ accept new state $x_{i+1} = x$
- if $\mu > P(x|x_i)$ reject new state and copy old state forward $x_{i+1} = x_i$
- Output $x_0, x_1, ..., x_n$ sample after run n times

R output

This is an example of the Metropolis Hastings Algorithm simulating the

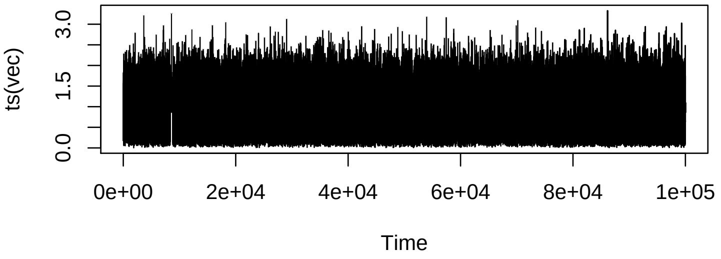
$$\text{Gamma distribution } Gam(2.3, 2.7)$$

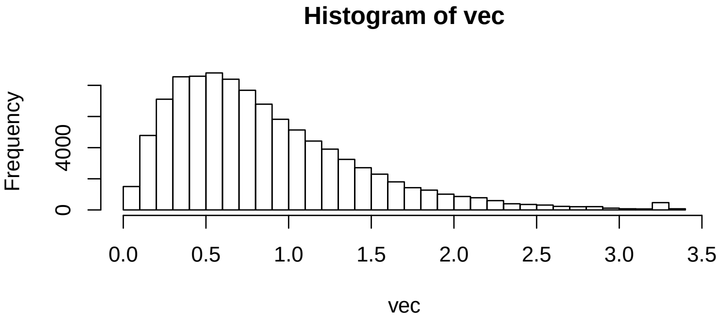
Using 100,000 samples from normal proposal distribution $N(0.85, 0.30)$ .The plots below show that the chain is well mixing and that there is independence among the samples generated. Also a histogram of the samples generated resembling the gamma density curve.
R code:

```
> gamm<-function (n, a, b)
+ {
+   mu <- a/b
+   sig <- sqrt(a/(b * b))
+   vec <- vector("numeric", n)
+   x <- a/b
+   vec[1] <- x
+   for (i in 2:n) {
+     candidate <- rnorm(1, mu, sig)
+     accept <- min(1, (dgamma(candidate,a, b)/dgamma(x, a, b))/(dnorm(candidate,mu,
+     u <- runif(1)
+     if (u < accept)
+       x <- candidate
+     vec[i] <- x
+   }
+   vec
+ }
> vec<-gamm(100000,2.3,2.7)
> c(mean(vec),var(vec))

[1] 0.8553529 0.3268857

> par(mfrow=c(2,1))
> plot(ts(vec))
> hist(vec,30)
> par(mfrow=c(1,1))
>
```


ts(vec)
0.0
1.5
3.0
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
Time


Histogram of vec
Frequency
0
4000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
vec

4.3 Gibbs Sampling

Gibbs sampling is a special case of the Metropolis-Hastings Algorithm. Gibbs sampling can be applied when the joint distribution is unknown or hard to sample from directly, but the conditional distribution of each variable is known and is easy to sample from. The Gibbs sampling algorithm generates a new sample from the distribution of each variable based upon the conditional distribution among the current values of the other variable.
The goal of Gibbs Sampling is to sample from a distribution π to approximate

$$E_\pi f(x_1, x_2), \text{ where } (x_1, x_2) \overset{iid}{\sim} \pi$$

By obtaining a sequence of observations which are approximated form a specific multivariate probability distribution. This sequence can be used to approximate the joint distribution to approximate the marginal distribution of one of the variables, or some subset of variables to compute an integral. It requires all the conditional distributions of the target distribution to be sampled exactly. Based on full conditional distributions $p(x_1|x_2)$ and $p(x_2|x_1)$ with the target distribution $\pi(x_1, x_2)$
General Strategy:

- Alternate between x_1 and x_2
- Sample from $x_1 \sim P(x_1|x_2)$
- Sample from $x_2 \sim P(x_2|x_1)$

Using the procedure does not require an acceptance ratio to be calculate since it will always be 1.

Bivariate Normal

The bivariate normal distribution can be generated using the Gibbs Sampler. Let $Z = (x, y)$ be a bivariate normal sample(of size 1), with mean $\mu = (\mu_x, \mu_y)^T$ is unknown. The variances are known and equal to 1, and the correlation ρ is known. Let's take a flat prior for μ, then the posterior is also bivariate normal, with the same correlation matrix but with mean Z.
Gibbs Sampler:

1. Start with $\mu_x^{(0)} = Z_1$
2. For m=1,...,M do

$$\mu_y^{(m)}|(\mu_x^{(m-1)}, Z) \sim N(Y_2 + \rho(\mu_x^{(m-1)} - Z_1), 1 - \rho^2)$$
$$\mu_x^{(m)}|(\mu_y^{(m)}, Y) \sim N(Z_1 + \rho(\mu_y^{(m)} - Z_2), 1 - \rho^2)$$

The first few iterations can be disregarded as a burn-in. For a sufficiently large M, the sample

$$\{\mu^{(m)} = (\mu_x^{(m}, \mu_y^{(m)})^T : m = 1, ..M\}$$

should be close to a sample from the bivariate normal posterior.

R Output

This is an example of using Gibbs sampling to generate the

$$\text{Bivariate normal distribution } N(0, 0, 1, 1, 0.97).$$

Using 10,000 samples successively sampling from the conditional distributions. The plots below show that the chain is well mixing among the samples generated for both the x and y variables. Also a scatter plot of the data displaying their correlation as well as a histograms of the data resembling the normal density curve for both variables.
R code:

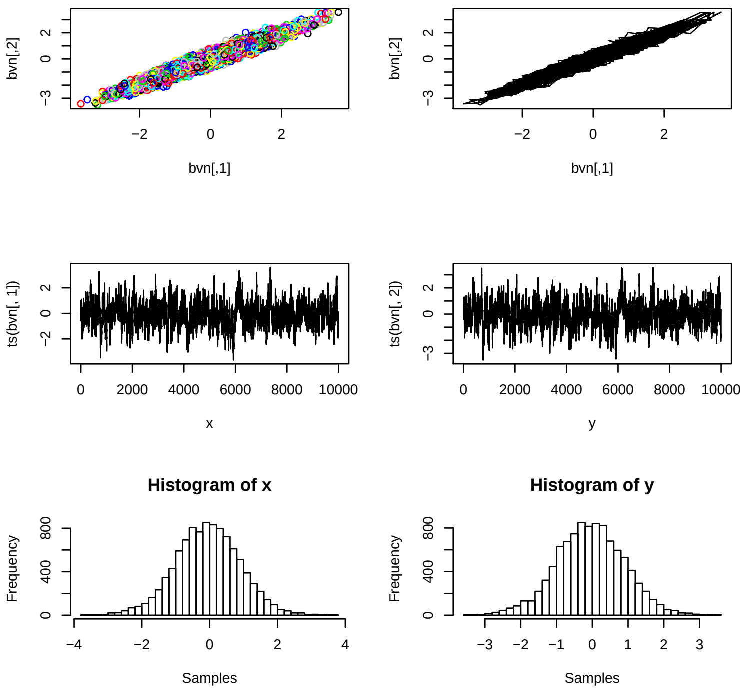
```
> gibbs<-function (n, rho)
+ {
+   mat <- matrix(ncol = 2, nrow = n)
+   x <- 0
+   y <- 0
+   mat[1, ] <- c(x, y)
+   for (i in 2:n) {
+     x <- rnorm(1, rho * y, sqrt(1 - rho^2))
+     y <- rnorm(1, rho * x, sqrt(1 - rho^2))
+     mat[i, ] <- c(x, y)
+   }
+   mat
+ }
> bvn<-gibbs(10000,0.97)
> mean(bvn)

[1] -0.05310733

> var(bvn)

          [,1]      [,2]
[1,] 0.9076485 0.8755069
[2,] 0.8755069 0.9049626

> par(mfrow=c(3,2))
> plot(bvn,col=1:10000)
> plot(bvn,type="l")
> plot(ts(bvn[,1]), xlab='x')
> plot(ts(bvn[,2]), xlab='y')
> hist(bvn[,1],40, xlab='Samples', main= 'Histogram of x')
> hist(bvn[,2],40, xlab= 'Samples', main= 'Histogram of y')
> par(mfrow=c(1,1))
```


bvn[,2]
−3
0
2
−2
0
2
bvn[,1]
bvn[,2]
−3
0
2
−2
0
2
bvn[,1]
ts(bvn[, 1])
−2
0
2
0
2000
4000
6000
8000
10000
x
ts(bvn[, 2])
−3
0
2
0
2000
4000
6000
8000
10000
y
Histogram of x
Frequency
0
400
800
−4
−2
0
2
4
Samples
Histogram of y
Frequency
0
400
800
−3
−2
−1
0
1
2
3
Samples

Gibbs Sampler for Full Conditional Distributions

Consider the set of samples $(x_1, ..., x_n)$

$$x_i \overset{iid}{\sim} N(\mu, \hat{\sigma}^2)\text{for } i = 1, ...n$$

where μ and σ^2 are unknown.
We are trying to estimate μ and σ^2 based on their prior distributions:

$$\mu \sim N(m, s^2)$$
$$\sigma^{-2} = \phi \sim Gam(a, b)$$

where

$$m, s^2 \text{ and } b \text{ are known hyperparameters.}$$

The joint posterior distribution $p(\mu, \sigma^2|x)$ does not have closed form. To estimate μ and σ^2 we need to sample (μ, σ^2). Using the full conditional distribution for μ $p(\mu, \sigma^2|x)$ and the full conditional distribution $p(\sigma^2|\mu, x)$ for σ^2.

Gibbs Sampling Algorithm:

- Sample μ from $p(\mu|\sigma, x)$
- Sample σ^2 from $p(\sigma^2|\mu, x)$

The full conditional distribution for μ is normal and the full conditional for σ^2 is inverse gamma. In Bayes model

$$(\mu|\sigma^2, x) \sim N(m^*, s_*^2)$$
$$\sigma^{-2}|\mu, x = \phi|\mu, x \sim Gam(a^*, b^*)$$

where

$$m^* = \frac{\frac{1}{s^2}m + \frac{n}{\sigma^2}\bar{x}}{\frac{1}{s^2} + \frac{n}{\sigma^2}}$$
$$s_*^2 = \frac{1}{\frac{1}{s^2} + \frac{n}{\sigma^2}}$$
$$a^* = a + \frac{n}{2}$$
$$b^* = \frac{\sum_{i=1}^{n}(y_i - \hat{\mu})^2}{2} + b$$

Gibbs Sampler:

- Sample μ from $N(\mu^*, s_*^2)$
- Sample σ^2 by sampling ϕ from $Gam(a^*, b^*)$ and setting $\sigma^2 = \frac{1}{\phi}$

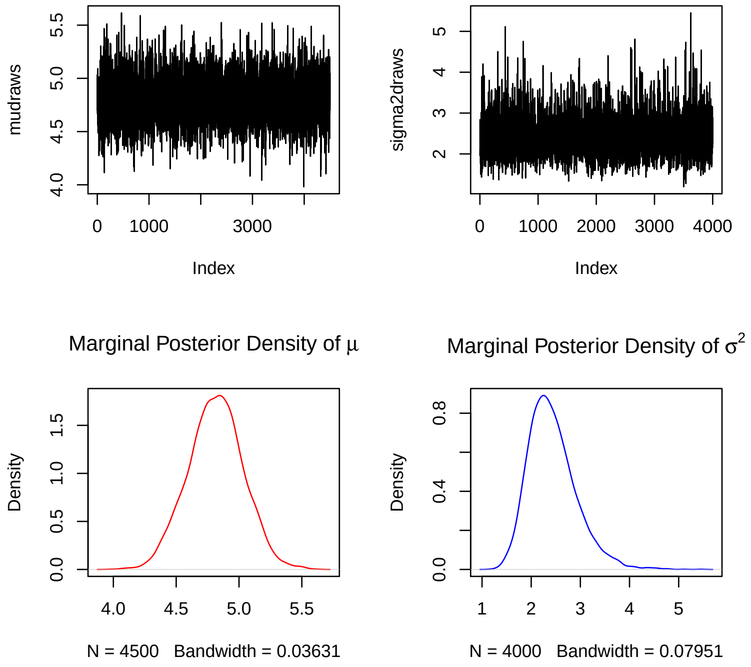
R Output

This is an example of using the Gibbs Sampler for μ and ϕ with 53 samples from the Normal Distribution with mean $\mu = 5$ and variance σ^2 from its inverse gamma marginal distribution and then, conditional on each value for $\sigma^2 = 2$, samples μ from the approximate

$$\text{Normal Distributions for } \mu = 5 \text{ and } \sigma = 2$$

The plots below show that the chain is well mixing among both μ and σ^2 and independence among the samples generated. Also the plots of the marginal posterior density of μ and σ^2 of resembling the Normal density curve centered at 5 for μ and 2 for σ^2.
R code:

```
> mu <- 5
> sigma2 <- 2
> phi <- 1/sigma2
> #Draw a Sample of size 53 From a N(mu,sigma2)
> x <- rnorm(53,mu,sqrt(sigma2))
> #Write the data to a data file
> write(x,file="./sampledata.dat",sep=",")
> #Read the data file and save it as an object
> mydata <- scan("./sampledata.dat",sep=",")
> #Prior Values
> m <- 0
> s2 <- 10
> a <- 2
> b <- 3
> n <- length(mydata)
> #Allocate Space to save Gibbs sampling draws
> totdraws <- 5000
> mudraws <- rep(0,5000) #Initial Value of mu is 0
> sigma2draws <- rep(1,5000) #Initial Value of s2 is 1
> #Begin the Gibbs sampler
> for(i in 2:totdraws){
+
+   #Generate a Draw for mu
+   mstar <- ((1/s2)*m + (n/sigma2draws[i-1])*mean(mydata))/((1/s2)+(n/sigma2draws[i
+   s2star <- 1/((1/s2)+(n/sigma2draws[i-1]))
+   mudraws[i] <- rnorm(1,mstar,sqrt(s2star))
+
+   #Generate a Draw for Sigma2
+   astar <- a + (n/2)
+   bstar <- (sum((mydata-mudraws[i])^2)/2) + b
+   phitemp <- rgamma(1,astar,bstar)
+   sigma2draws[i] <- 1/phitemp
+
+ }
> #Discard the first 500 Draws as Burn-in
> mudraws <- mudraws[501:length(mudraws)]
> sigma2draws <- sigma2draws[501:length(mudraws)]
> #Partition plot into 4 subplots
> par(mfrow=c(2,2))
> #Plot Trace Plots
> plot(mudraws,type="l")
> plot(sigma2draws,type="l")
> #Plot Marginal Posterior Densities
> plot(density(mudraws),col="red",main=expression(paste("Marginal Posterior Density
> plot(density(sigma2draws),col="blue",main=expression(paste("Marginal Posterior Den
>
>
```



Application

Shooting Percentage Using the Metropolis Algorithm

I will now use the Metropolis Algorithm to model the shooting percentages of basketball team as the normal distribution. A basketball team shooting percentage is the compilation of the each players 3-point and 2-point field goal makes divided by the total amount if shots taken by the team.
Now let's consider the shooting percentage of a basketball team's over n games:

$$y = (y_1, ..., y_n)$$

The percentages can be modeled using beta distribution,

$$y_i i \sim Beta(\theta, 2) \text{ for } \theta > 0$$
$$f(y_i|\theta) = \theta(1+\theta)y_i^{\theta-1}(1-y_i)$$

and use Gamma(a,b) as the prior distribution for θ,

$$p(\theta|y) \propto \theta^{n+a-1}(\theta+1)^n e^{-b\theta}(\prod_{i=1}^{n} y_i)^\theta := h(\theta)$$

Let's use the shooting percentages over the n=20 games with

$$\sum_{i=1}^{20} log y_i = -9.89$$

Choose $a = b = 1$ for the prior distribution
Previously approximated posterior using Bayesian Central Limit Theorem:

$$p(\theta|y) \approx N(3.24, 0.33)$$

Use the Metropolis Algorithm to sample from $p(\theta|y)$
Steps for Metropolis Algorithm

- Initial value $\theta^{(0)} = 3.24$
- Sample from $p(\cdot|\theta) = N(\theta^{t-1}, 0.33)$
- Sample μ from uniform(0,1)
- accept or reject proposal based on if $\mu < \frac{N(\theta^t,0.33)}{N(\theta^{t-1},0.33)}$ then $\theta^{i+1} = \theta$ else $\theta^{i+1} = \theta^i$

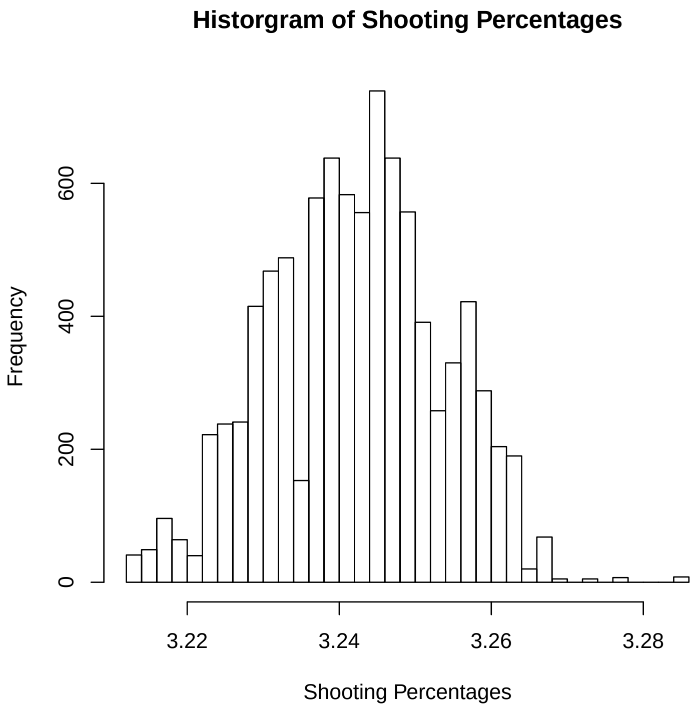
We will run this iteration 1000 times and treat the first 50 as burnins

R Output

R code:

```
> x = rnorm(1000,3.24,0.33);
> T <- 10000; B <- 1000; u.mu <- runif(T); u.sig <- runif(T)
> mu <- sig <- numeric(T); mu[1] <- 3; sig[1] <- 5
> REJmu <- 0; REJsig <- 0
> logpost = function(x,mu,sig){
+ loglike = sum(dnorm(x,mu,sig,log=TRUE))
+ return(loglike - log(sig))}
> for (t in 2:T) { mut <- mu[t-1]; sigt <- sig[t-1]
+ mucand <- mut + runif(1,-0.5,0.5)
+ sigcand <- abs(sigt + runif(1,-0.5,0.5))
+ log.alph.mu = logpost(x,mucand,sigt)-logpost(x,mut,sigt)
+ if (log(u.mu[t]) < log.alph.mu) mu[t] <- mucand
+ else { mu[t] <- mut; REJmu <- REJmu+1 }
+ log.alph.sig = logpost(x,mu[t],sigcand)-logpost(x,mu[t],sigt)
+ if (log(u.sig[t]) < log.alph.sig) sig[t] <- sigcand
+ else { sig[t] <- sigt; REJsig <- REJsig+1 }}
> REJratemu <- REJmu/T; REJratesig <- REJsig/T;
> samp <- 1001:10000; musamp <- mu[samp]; sigsamp <- sig[samp]
> hist(musamp,50, xlab='Shooting Percentages', main='Historgram of Shooting Percenta
> mu.den <- density(musamp)
> plot(mu.den)
> plot(musamp,sigsamp)
> 00
```

```
[1] 0
```


Historgram of Shooting Percentages
Frequency
0
200
400
600
3.22
3.24
3.26
3.28
Shooting Percentages

Further Discussion

In this paper, we examined how the Metropolis Algorithm, the Metropolis-Hastings Algorithm and the Gibbs Sampler operates and their respective usefulness in sampling. Through simulation these algorithms use probabilities of separate events merging them into a composite picture which gives an approximation that can then be used to solve a problem. All three of these processes can be categorized as a Markov Chain Monte Carlo Method. These methods can be used to complete various tasks such as numerical approximations of multi-dimensional integrals. The main algorithm, the Metropolis-Hastings Algorithm, specifically is primarily used for numerical integration. The Gibbs Sampler, a variation of the Metropolis-Hastings Algorithm, is primarily used for statistical inference more specifically Bayesian inference, determining the best value of a parameter.

R Studio was used in the making of this paper to create various R codes and different types of graphs. R Studio is a free and open-source integrated development environment for R, a programming language for statistical computing and graphics.There currently exists other software packages that can carry the Gibbs Sampler such as BUGS Bayesian inference Using Gibbs Sampling and JAGS is Just Another Gibbs Sampler. Both software packages simulate Bayesian hierarchical models using Markov chain Monte Carlo.

Markov Chain Monte Carlo methods are very prevalent today particularly in statistics, computational physics,biology, linguistics, econometric and computing science. Some real world applications include economical sector where option pricing in relation to the stock market, state space models epidemiology and meteorology. Mixture models can be created for cluster analysis pertaining to population studies as well as operational optimization for various disciplines. In order to complete this various application more complex Markov Chain Monte Carlo method are required to create estimation model. The applications of the Metropolis and Metropolis-Hastings Algorithm and Gibbs Sampling are not constrained to the parameters of this paper.

In the future, it would be interesting to continue the study of various different Markov Chain Monte Carlo Methods and hopefully apply them to basketball analytics. In conclusion, this paper was only a foundation for the Metropolis Algorithm, Metropolis-Hastings Algorithm and Gibbs Sampler. Many more applications and outcomes can found and studied using the incredible results we can achieve through these various algorithms.

5 Bibliography

1. Introduction to Probability by Charles M. Grinstead J. Lauire Snell
2. Introduction to Monte Carlo and MCMC Methods by Anotonietta Mira
3. An Introduction to MCMC Sampling Methods
4. The Conjugate Prior for the Normal Distribution by Michael I. Jordan
5. Gibbs Sampler Examples by Ryan Martin
6. Metropolis- Hastings Sampling by Darren Wilkinson
7. Metropolis- Hastings Sampling by Eric F. Lock