# Analysing and Applying the Metropolis-Hastings Algorithm for Multi-Modal Distributions

Jordan Pears

May 8, 2018

**Abstract**

In this project, I will explain how to formulate, apply, and analyze the Metropolis-Hastings algorithm. This will be achieved by first setting out the fundamentals of the Metropolis-Hastings algorithm including the theory behind the algorithm and its formulation. I will then utilize the algorithm to complete simplistic examples such as attempting to model a known target distribution to gain a fundamental understanding. After establishing the intuition behind the algorithm I will apply it to advanced examples on IBM employee data and Titanic survival data to establish the potential of the algorithm for parameter estimation. The potential pitfalls one can encounter when applying the Metropolis-Hastings algorithm to multi-modal distributions will be explored using examples from bimodal distributions to unknown multi-modal distributions. I will also cover how to generate convergence diagnostics for a range of different cases using multiple diagnostic methods spanning graphical and analytical methods.

# Contents

# 1 Introduction

## 1.1 Marvok Chain Monte Carlo

Suppose an ordered sequence of random numbers is generated, such as:

$$y = (x_0, x_1, x_2, x_3, ...x_n).$$

Here $x_n$ represents the $n$th number/vector within the chain. At each step $n \geq 0$ a new number is added to the list at the end. This new number is determined by the previous value in the chain given by the conditional probability $Q(x_{n+1}|x_n)$. This makes the chain history independent as $x_{n+1}$ given $x_n$ does not depend on $\{x_0, x_1, x_2, ..., x_{n-1}\}$. This idea of a time-independent chain with new samples being formed from a distribution conditional on the previous value is called a *Markov Chain*. The probability $Q(x_{n+1}|x_n)$ is known as the *Jumping Distribution*[1], it is important that this probability has no time dependence to ensure the chain is not biased within any direction as the chain is generated. This allows the chain to be sampled at any period along the chain and if it is sufficiently long it will represent the entire chain.

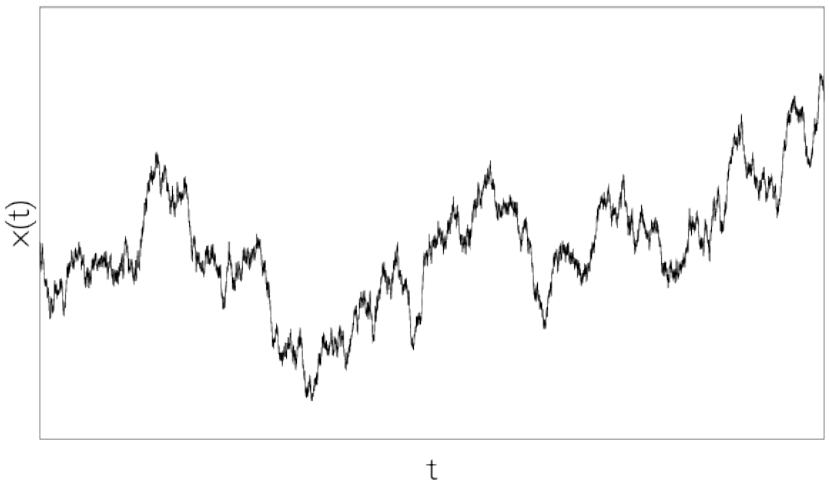


Figure 1: An example of a discrete random walk process over a time $t$ with a very small jumping distribution[2].

## 1.2 Metropolis-Hastings

The basics of the Metropolis-Hastings algorithm were developed in 1953 by Nicholas Metropolis. In 1970 W. K. Hastings went on to extend it into its modern-day formulation[3]. The algorithm is used to generate a sequence of random samples from a chosen probability distribution, $P(x)$. As the algorithm proceeds the chain will begin to mimic the distribution of the target distribution until, as $N \to \infty$ the chain distribution, $\pi(x)$ will be equal to $P(x)$. For this to occur the following conditions must be met[4]:

- Stationary distribution existence: A stationary distribution $\pi(x)$ must exist. This can be shown to be the case if each transition from $x_n \to x_{n+1}$ is reversible such that the probability to make that move is equal to the probability of reverse, $x_{n+1} \to x_n$. Thereby giving:

$$\pi\left(x_n\right) Q\left(x_{n+1} | x_n\right) = \pi(x_{n+1}) Q(x_n | x_{n+1})$$

- Stationary distribution uniqueness: $\pi(x)$ must be a unique distribution. This is proven to be the case by the ergodicity of Markov Chains, that is to say each state must be aperiodic and positive recurrent - the system returns to the same state within a finite number of steps and doesn't return to the same state in fixed intervals.

## 1.3 Algorithm

The Metropolis-Hastings algorithm is detailed below[3];

1. Pick an initial state for the chain, $x_0$.
2. Generate a random candidate value, $x_{n+1}$ using the jumping distribution, $Q(x_{n+1}|x_n)$.
3. Determine the acceptance probability,

$$A = \min\left(1, \frac{P(x_{n+1})}{P(x_n)}\right).$$

4. Generate a uniformly distributed random number $r \in [0, 1]$.
5. If $r \leq A$ accept the state transition and continue using the proposed $x_{n+1}$ value.
6. If $r > A$ reject the state transition and set $x_{n+1} = x_n$.
7. Repeat from 2 until a desired chain length has been obtained.

## 1.4 Jumping Distribution

The Jumping Distribution is used to obtain the size of the following step in the chain by sampling a random number from it. It can be negative or positive allowing the front of the chain (commonly referred to as a random walker) to explore all of the possible spaces available. The jumping distribution in the Metropolis-Hastings algorithm is essential to the efficient convergence of the chain. Too wide a jumping distribution and the nuances of the distribution will be lost, such as skipping small tight peaks. Too thin and the chain will take extremely long to converge requiring an excessive runtime to generate the target distribution.

This distribution must also be symmetric for the above algorithm, there is an alternative formulation that permits an asymmetric distribution but we will not look at this within this project. Because of this requirement, a popular candidate is the

Gaussian distribution which allows the typical jump to be appropriately sized for exploration of the target distribution whilst allowing for the low probability of particularly large jumps which are essential for crossing regions of low/zero probability within the target distribution.

A Gaussian distribution is defined by its mean and standard deviation. For the Metropolis-Hastings algorithm, the mean is the previous value within the chain and the standard deviation is important to tune for each individual chain.

### 1.5 Burn-In

The purpose of a burn-in is to ensure that the converged chain is independent of the initial conditions of the chain by discarding the first $n$ elements of the chain. This prevents a poorly chosen starting point from skewing the data in the final distribution, as areas of low probability may be oversampled in these first $n$ samples.

## 2 Background Literature

### 2.1 Bayes Theorem

It is important to acknowledge to fundamentals when conducting these investigations. The first reference to Bayes Theorem appeared in a essay written in 1763 by Thomas Bayes[5], it discusses the logic behind the theorem and how it can be understood through simplistic examples. This is important to acknowledge because it is the basis for all of my analysis within this report as I am operating within a Bayesian framework which means that I consider my knowledge to be static and my beliefs to be subject to change based on uncertainty, which is a contrast to the Frequentist framework which considers the beliefs to be static and based only on knowledge.

### 2.2 Markov Chain Monte Carlo (MCMC)

The method for using computers to complete Monte Carlo simulations was outlined within a 1953 paper by Nicholas Metropolis et al[6]. This paper discusses how "fast computing machines" can be used to create a Markov Chain that explores a potential parameter space and eventually can result in converging to a histogram that represents the distribution of that parameter space. This is the base of sampling using Markov chains and this paper went on to influence a huge amount of research on the applications of Markov chains such as the Metropolis-Hastings algorithm. Within this report, I will use the method outlined in this paper to conduct analysis on unknown target distributions with the desire to obtain a shape that represents the distribution.

## 2.3 Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm was formulated within the 1970 paper by Wilfred Hastings[3]. It extends upon the previously discussed paper introducing MCMC methods to apply it to a general case called the Metropolis-Hastings algorithm. It gives examples of how this algorithm can be applied to simplify challenging numerical problems. I will extensively use this algorithm within my report and will extend upon the challenges that this paper discusses. I will also discuss multi-dimensional Metropolis-Hastings executions as referenced within the paper in the later parts of my report and how these multi-dimensional chains can be used to conduct parameter estimation within a Bayesian framework.

## 2.4 Analysing Convergence of Markov Chains

The documentation on the coda software[7] highlights a range of tools that are available to analyse Markov chains for successful convergence diagnostics. I will extend the tools available in this paper to analyse my chains and I will give examples of how to utilize these tools to determine the convergence of a chain. These tools are to be used with the programming language R, I will be using this throughout my report.

# 3 Generating Probability Distribution Functions

## 3.1 Method

I chose to use the R programming language instead of the popularly chosen Python. This is because of its great support for statistical methods and large use academically. To begin analyzing MCMC methods a method of generating arbitrary probability distributions was required, this was achieved using the following method;

1. Choose a smooth bounded curve $f(x)$ to convert into a probability distribution function.
2. Select the boundaries of the function $x_i$ and $x_f$ such that $-\infty < x_i < x_f < \infty$.
3. Computationally determine the integral of this curve between the boundaries and set this value as the normalization constant, $k$.
4. Define the probability density function as $f'(x) = \frac{f(x)}{k}$ and use this going forward.

## 3.2 Example

We can apply the method above to a function which is defined by:

$$f(x) = \begin{cases} 1 & x \in [0, 4] \\ 0 & \text{otherwise} \end{cases}$$

It is clear that this function will not be a suitable candidate for a probability density function as

$$\int_0^4 f(x)dx = 4.$$

For f(x) to be suitable as a probability density function the integral above must evaluate to 1. It is therefore required to scale the function by a multiple of $\frac{1}{4}$.

# 4 Applying Metropolis Hastings to A Distribution

Now that the Metropolis-Hastings algorithm has been defined it can be applied to some distributions to show its potential.

## 4.1 Example 1

Given a distribution such as the one defined in Section 3.2. The Metropolis-Hastings algorithm can be applied to this and will follow the path of the random walker as it explores different areas within the probability distribution. Using the standard Gaussian jumping distribution, a chain of length 100,000 can be generated and plotted. For this example it is not necessary to conduct a burn-in phase as the shape is uniform and the initial conditions do not have any effect - any potential skewing to be removed from a burn-in period would simply be shifted to another position within the histogram.

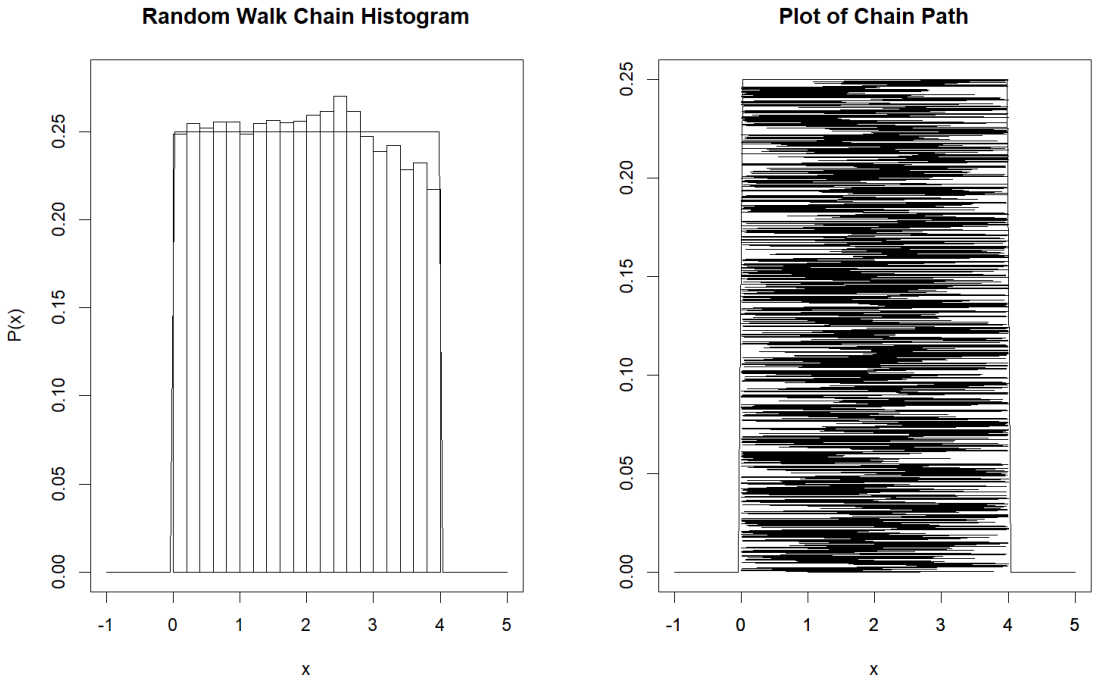


Figure 2: Graphs showing the results of the Metropolis-Hastings algorithm on the distribution above. The left graph shows a histogram of the converged chain above the target distribution. The right graph shows the chain's path as it explored the target distribution overlaid atop the target distribution.

The plot on the left of Figure 2 shows a histogram of the final chain after 100,000 iterations and plotted around it is the target probability distribution that is being modeled. This is not a perfect representation of the target distribution and could have been improved by changing the variance of the jumping distribution or increasing the chain length. The plot on the left shows a histogram of the chain plotted above the target distribution. This plot is particularly illustrative as it shows that the chain has explored all of the available spaces, this will become more important when multi-modal distributions are explored.

## 4.2 Example 2

The example shown above is good for illustrating the potential of the Metropolis-Hastings algorithm but does little in the way of a real-world application. A popular application of the Metropolis-Hastings algorithm is to determine unknown parameters in a Bayesian framework for a given model.

### 4.2.1 Bayesian Parameter Estimation

Bayesian parameter estimation is the process of determining values for unknown parameters within a model. It is derived from Bayes rule which states that if we have

random variables $A$ and $B$ that can take on specific values of $a$ and $b$ respectively we have[5]:

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)}.$$

This means that we can estimate the conditional probability of a random variable $A$ given $B$ from the conditional probability of $B$ given $A$. In this framework, our prior knowledge is represented by $P(a)$. We already know this value. Our posterior knowledge is $P(a|b)$ after we have had some observations of $B$ taking on some values of $b$.

We can therefore use this to determine a parameter $x$ given some data $d$ that is related to $x$. The probability of the data given the parameter is called the likelihood therefore we can calculate the probability of a parameter given the likelihood of some data. Through many observations we can update our parameter value using the likelihood function and the prior distribution to obtain a final posterior distribution, these observations come from the data and the updates of the parameter value come from the Markov Chain process.

### 4.2.2 Likelihood Function

There needs to be a function that gives the likelihood for a parameter value given some data that suggests a possible value for that parameter.

$$L(\theta|x)$$

Here $\theta$ is the parameter value and $x$ is the data that is accessible to us for estimating $\theta$. Therefore $L(\theta|x)$ is higher when values of $\theta$ support our data, $x$. In most practical cases it is more appropriate to use the *log likelihood*.

$$l(\theta|x) = \log(L(\theta|x)).$$

This is useful because the likelihood function can become very small quickly and using logarithms enhances the differences at small values of $L$. From this point onwards the log-likelihood will be referred to as the likelihood.

### 4.2.3 Prior Probability

The prior probability is a probability distribution that represents the initial knowledge regarding the desired parameters. It can be objective if it has some basis in previous knowledge, or non-informative if it is based on no prior knowledge. For example, an objective prior for the number of people that wear glasses could be based on the prior knowledge of the number of people with vision problems. A non-informative prior could simple be the uniform random distribution, $Beta(1, 1)$.

### 4.2.4 Posterior Probability

The posterior probability is the desired output of the Metropolis-Hastings algorithm, it is the prior probability having taken into account the data available with the likelihood function. It is proportional to the product of the Prior and Likelihood functions, but will become proportional to the sum as logarithms are being used.

### 4.2.5 Setting up Data

For this analysis a linear model with 3 unknown parameters, $\alpha, \beta,$ and,$\gamma$ will be used. The model takes the following form:

$$y = \alpha x + \beta + \mathcal{N}(0, \gamma)$$

Where $\mathcal{N}(0, \gamma)$ denotes a random normal distributed variable with mean 0 and standard deviation $\gamma$, this represents the error within the model.

Data now needs to be generated for the analysis to continue, this is generated by choosing values for the parameters that are desired to be extracted from the Metropolis-Hastings algorithm.

$$\alpha = 4, \beta = 2, \gamma = 8$$

From this 100 samples are generated that shall constitute the available data.

### 4.2.6 Prior Probability

The prior distributions for these parameters will be normally distributed with a standard deviation of 1.0 to represent the uncertainty in these guesses and the mean of these distributions will represent our guesses about the possible value of the parameter;

$$\alpha = 3.8, \beta = 1.2, \gamma = 8.3.$$

### 4.2.7 Jumping Distribution

The jumping distribution to be used with this example will be a standard Gaussian distribution with a mean of 0 and a standard deviation of 0.3. This has been chosen as it will ensure that the full range of potential parameter values are considered.

### 4.2.8 Results

After running the chain for 100,000 iterations a good approximation is obtained which is close to the correct values of the previously defined parameters.

After running this implementation of the Metropolis-Hastings algorithm on this chain one obtains the following mean value for the parameters in the normal distribution function;

$$\alpha = 3.962, \beta = 1.904, \gamma = 7.978.$$

This is clearly going in the correct direction and shows the potential of the Metropolis-Hastings algorithm for parameter estimation.

The below figure shows the diagnostics for each parameter. These diagnostics were obtained by using the "coda" package on R - coda is an MCMC convergence diagnostic tool that allows for easier handling of long chain objects through plotting and statistical overview[7].

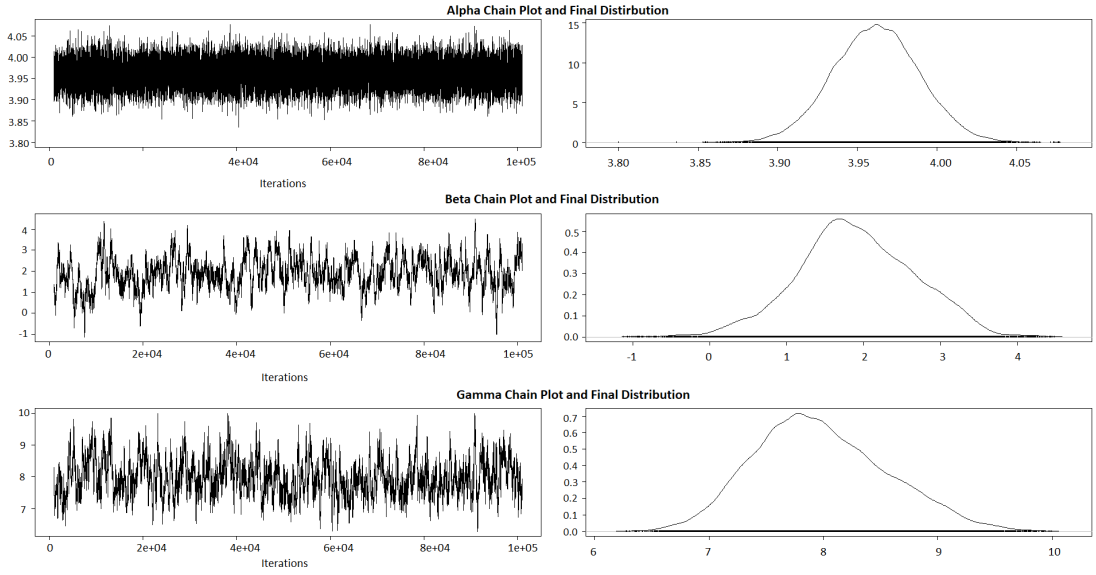


Figure 3: Final distributions of parameters $\alpha$, $\beta$, and $\gamma$ after 100,000 Metropolis-Hastings iterations

## 4.3 Intuition Behind Metropolis-Hastings

The Metropolis-Hastings algorithm is an elegant way to draw samples from a probability distribution as shown in Section 4.1. The benefit of this is that we do not need to know the height of the distribution at any point, only the relative difference to the previous point. This effectively results in the chain randomly moving about the distribution as shown in Figure 2 on the right side. This *random walking* process results in proportional amounts of iterations of the chain spent in certain regions. This proportional amount of iterations can eventually result in the modeling of the probability distribution being sampled from. This is typically achieved by viewing the chain as a histogram of discrete values, as is visible on the right side of Figure 2.

# 5 Analysing Convergence

Assessing whether a chain has converged to the stationary distribution is a topic of active research and many people have proposed methods to determine convergence in MCMC methods[8]. In the example shown in Figure 2 it was trivial to determine convergence as the target distribution could be directly observed and outlined on the chain. However, this is not a realistic example. Some potential qualitative ways to assess convergence are:

- Are multiple different iterations of the chain similar?
- Are any sufficiently long segments of the chain identical?
- Do different initial conditions yield similar chains?

If the answer to all of the above is yes then the likelihood of having a converged chain is increased. This leads to two potential methods to ensure convergence; make extremely long chains[9] which ensure all of parameter space is explored or make multiple simultaneous chains with different initial conditions which through comparison may reveal a lack of convergence [10].As shown in D. B. Gelman[11] chains can have a particularly difficult time revealing convergence when there are multi-modal distributions. It is now considered standard practice to run 3-7 long simultaneous chains which is a combination of the above methods[12].

## 5.1 Graphical Methods

### 5.1.1 Time-Series Chain Plots

A time-series plot is a plot which shows the chains progression through each step, this is the method is utilized on the right side of Figure 2 and the left side of Figure 3. This method is one of the most commonly used ways to observe convergence.

### 5.1.2 Parameter Dispersion Plots

Another type of graphical method is shown on the right side of Figure 3. This plot shows the probability of the parameter being that given value given the data. As you can see from the parameter plot in figure 3 for $\alpha$, this line has a low width giving the parameter a lower standard deviation, this corresponds to less uncertainty in the parameter. The same cannot be said for $\beta$ and $\gamma$, the standard deviations for these chains are large ($\approx 2$) making the prediction far less stable and likely to differ for independent runs. This can be improved by increased the amount of data available, as the width of this is determined by the data.

## 5.2 Analytical Methods

### 5.2.1 Gelman Rubin Convergence Measure

As previously mentioned, a potential way to measure the convergence of the chain is to run multiple chains and compare them. This method is outlined by Gelman and Rubin[10]. The intention of this method is to remove the initial guess dependence and also obtain a so-called *Potential Scale Reduction Factor*, this is a factor of how accurate the estimations may become given infinite run time. This allows fine-tuning of the running length of the chains.

The method is outlined below:

1. A widely spread estimation of the target distribution is generated using any methods available, typically this is achieved by obtaining maxima and minima and creating a widely spread assortment of guesses close to these values.
2. Generate a number of chains for these initial guesses, $N$, typically consisting of 4-9 chains. With one chain per initial parameter guess.
3. Each parameter can now have a *Potential Scale Reduction Factor* (PSRF) generated for it,

$$PSRF = \sqrt{\frac{n-1}{n} + \frac{B}{nW}}.$$

   In which $W$ is the average value of the variances within the chains for $N$ chains generated. $B$ is the variance between the $N$ chain means. $n$ is the number of iterations within each chain - ignoring any burn-in.

As the value of the $PSRF$ converges to 1 the chain will also have converged and will no longer depend on the initial parameters given. If this occurs as $N$ becomes larger one can consider the chain to have explored all potential parameter values and to be accurate. However, this method can fall short for multi-modal distributions[10]. This problem comes about due to the large amount of time that the random walking process may take to pass from one node to another. Although an issue for this measure, the presence of a multi-modal distribution will be revealed by the $PSRF$ not converging to 1 when simultaneous chains reside within independent nodal neighborhoods.

### 5.2.2 Example

Looking back to Section 4.1, this method can be applied to analyse the convergence of the chains. Using 5 chains with starting values of $0.7, 1.4, 2.1, 2.8$, and $3.5$. The chains can now be generated using $n = 1,000$ iterations each and then the PSRF can be calculated from these chains giving a value of 1.48. This is above the threshold for acceptable convergence and therefore shows that the chains have not suitably explored the parameter space, this can also be shown graphically.

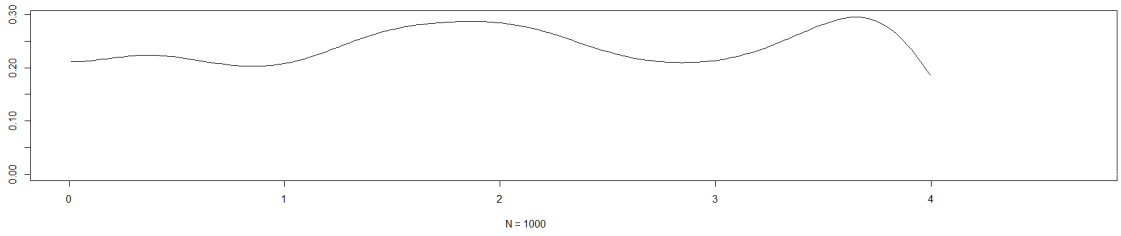


Figure 4: The resulting smoothed histogram after 1,000 iterations on the 5 chains.

Now the iterations can be increased, such that $n$ is equal to $100,000$ and the process can be repeated but this time the results are much more accurate due to the increased iterations within the chain. This results in a PSRF value of 1.02 which suggests the chains have converged, and end up with the below results for the chains, which are correct as they resemble Figure 2.

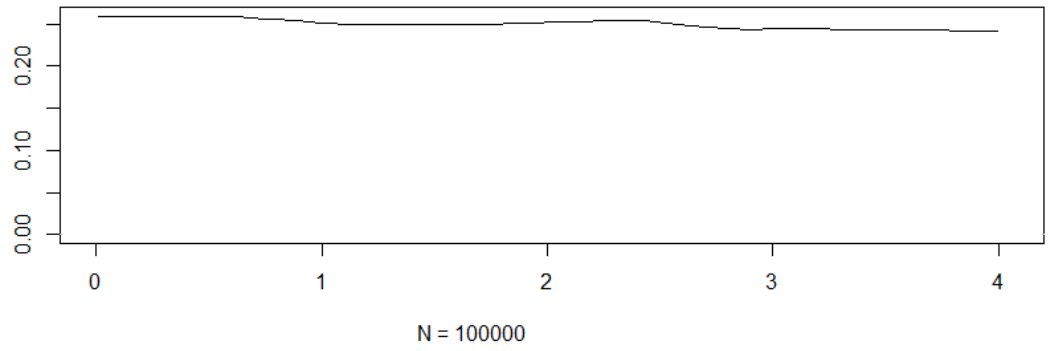


Figure 5: The resulting smoothed histogram after 100,000 iterations on the 5 chains.

The PSRF will go down as the number of iterations decrease which allows one to view when the chains have converged and therefore run trial chains on the generated data from chapter 4.2.5 before running the ones that are unknown as to preserve computation power. Below is a graph of the PSRF with respect to the number of iterations, it shows the optimum number of iterations would be $\approx 5000$.

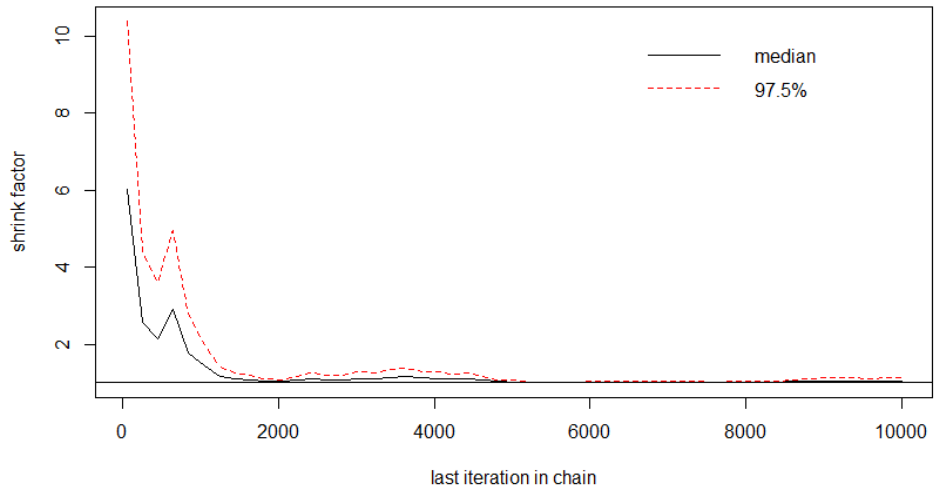


Figure 6: A graph of the PSRF with respect to the number of iterations within the chain.

### 5.2.3 Known Target Distribution Method

I propose a method to analyse complete chains given the prior knowledge of the target distribution desired, this method utilizes the converged chain histogram of 100 values and the target distribution to compare the two to determine a quantitative measure of accuracy for the converged distribution. The method starts by splitting the target distribution into $I$ discrete points within the range of the parameters so that you have vector, $V_{target}$.

$$\vec{V}_{target} = (P_{target}(x_1), P_{target}(x_2), ..., P_{target}(x_I).$$

Where $x_i$ represents an $x$ value discretized into $I$ unique, equally spaced values for the whole range of the target distribution, such that if the range of the target distribution is 1 to 100, and $I = 100$ then the vector $x_i$ will span the values $[1, ..., 100]$.

To determine the value for $\vec{V}_{chain}$, the discrete density of the chain needs to be determined. This is achieved by utilizing the discrete values of $x_i$ and summing the number of chain locations within each "bucket" of the x-values then determining the density of each one afterwards. This first step follows the below method;

1. Given chain $\vec{\pi}$, determine which bucket the first value belongs to within $x_i$.
2. Count this occurrence within a new vector $\vec{\pi_{x_i}}$.
3. Repeat this process until each value in $\vec{\pi}$ has been accounted for.

Now there exists a vector of discrete values of the chain, $\vec{\pi_x}$. This now needs to be used to determine the density of each bucket. To do this each bucket value must be multiplied by the total number in all buckets, the chain length $N$, such that,

$$\vec{V}_{chain} = \frac{1}{N} (\vec{\pi}_{x_1}, \vec{\pi}_{x_2}, ..., \vec{\pi}_{x_I}) .$$

Now there are values for the $\vec{V}_{chain}$ and the $\vec{V}_{target}$ which represent a discretized value for a precision of 100 the absolute error $E$, can be determined for each point along the chain on a known target distribution using,

$$\vec{E} = \left( |\vec{V}_{chain_i} - \vec{V}_{target_i}| \right).$$

This vector can now be used to see a detailed analysis of the chains by plotting the error at particular parts of the curve or by summing the error to determine a quantitative value for the accuracy of the chain.

# 6 Multi-Modal Target Distributions

A multi-modal target distribution is one which has multiple nodes with areas of lower probability between them, these often take the form of a linear normalized combination of normal distributions, forming distinct peaks and troughs of probability.

## 6.1 Bimodal Distribution

A simple two-mode distribution may take the form shown below,

$$f(x) = \begin{cases} \frac{|\sin(x)|}{k} & x \in [0, \pi] \\ \frac{|\sin(x-0.4)|}{k} & x \in [\pi + 0.4, 2\pi + 0.4] \\ 0 & \text{otherwise} \end{cases}$$

Where k is the constant that is required to normalise the probability density function and is calculated from,

$$k = \int_0^{2\pi+0.4} f(x) \approx 3.95.$$

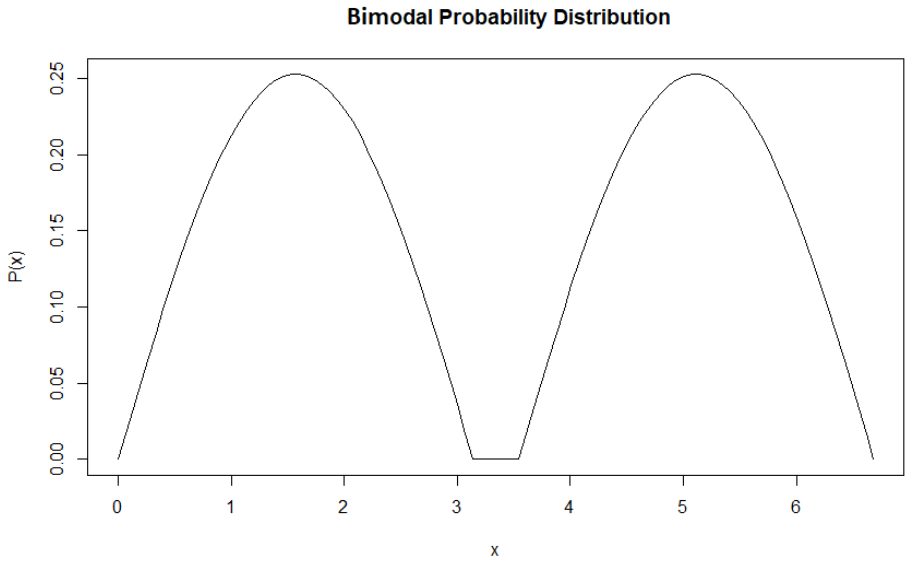


Figure 7: The plot of the probability distribution defined above.

This is a particularly simple example of a bimodal distribution but does have some basis in reality, this form of bimodal distribution is commonly found within the brain and is often used to model the potential energy of a dual-state neuron[13]. This is plain to see as the neuron can have two different states that are defined by the two nodes of the graph. The energy is likely to lie within the peak of the state but there is some variance in this energy hence developing the form seen in Figure 7.

### 6.1.1 Applying Metropolis Hastings to a Bimodal Distribution

The Metropolis-Hastings algorithm will be applied to this distribution using an arbitrary starting point 6, as the beginning of the chain and using $N = 100,000$ values within the chain. It is to be expected that the random walker will suitably sample the peak which has the value 6 within it on the right. Below is the histogram of the chain after $N$ iterations. Also included in this graph is the target distribution for comparison.

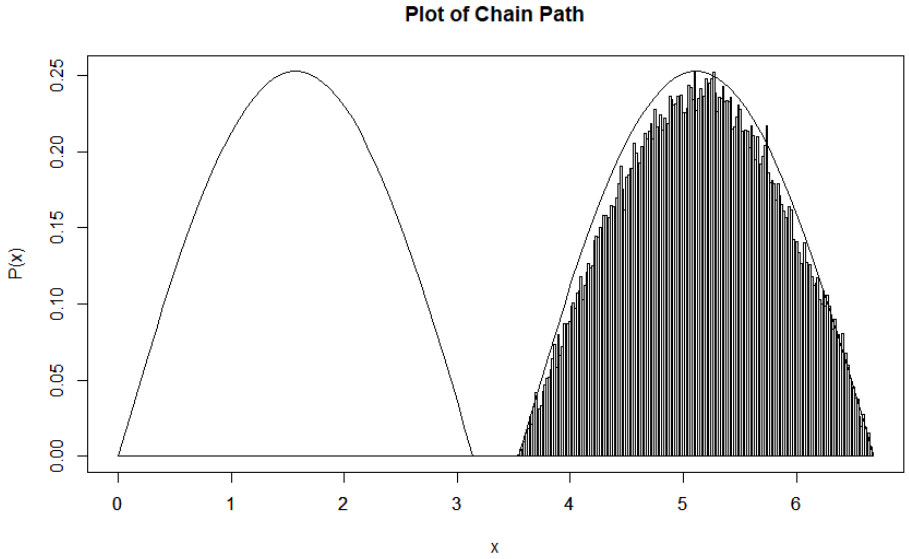


Figure 8: The chain after $N$ iterations on the probability distribution defined in figure 8.

For this execution, a jumping distribution with a standard deviation of 0.01 has been used to illustrate where this could cause a problem within a bimodal distribution.

This graph shows that the chain has approximated the right peak well, most areas within the right peak have been explored and the chain shows approximately the same shape. However, the left peak is totally untouched by the chain and therein lies the problem with multi-modal distributions, they can appear to have converged without actually doing so, causing the chain to not fully explore the parameter space.

This outlines the importance of selecting the appropriate jumping distributions to allow the chains to explore each element of the parameter space. If the standard deviation on the jumping distribution is too low the random walker will not be able to cross areas of low probability such as the gap between the peaks in Figure 8. If the standard deviation is too high there may be a loss of the finer details within the target distribution by continually jumping over them.

### 6.1.2 Selecting An Appropriate Jumping Distribution

I have generated multiple chains using different transition probabilities to compare each chain for convergence. This will be done in a similar method used to determine the PSRF by Gelman & Rubin [10] by selecting an overly large range of randomized standard deviations to pick from for the jumping distribution. This will illustrate if a Metropolis-Hastings run has converged or not via the comparison of results. Six different standard deviations can be selected for the chain to execute with, ranging from 0.01 to 0.31 inclusive. The results of each different run is shown below in a similar form to Figure 7. A Gaussian distribution is being used for the jumping distribution and the standard deviation of each distribution is being altered.

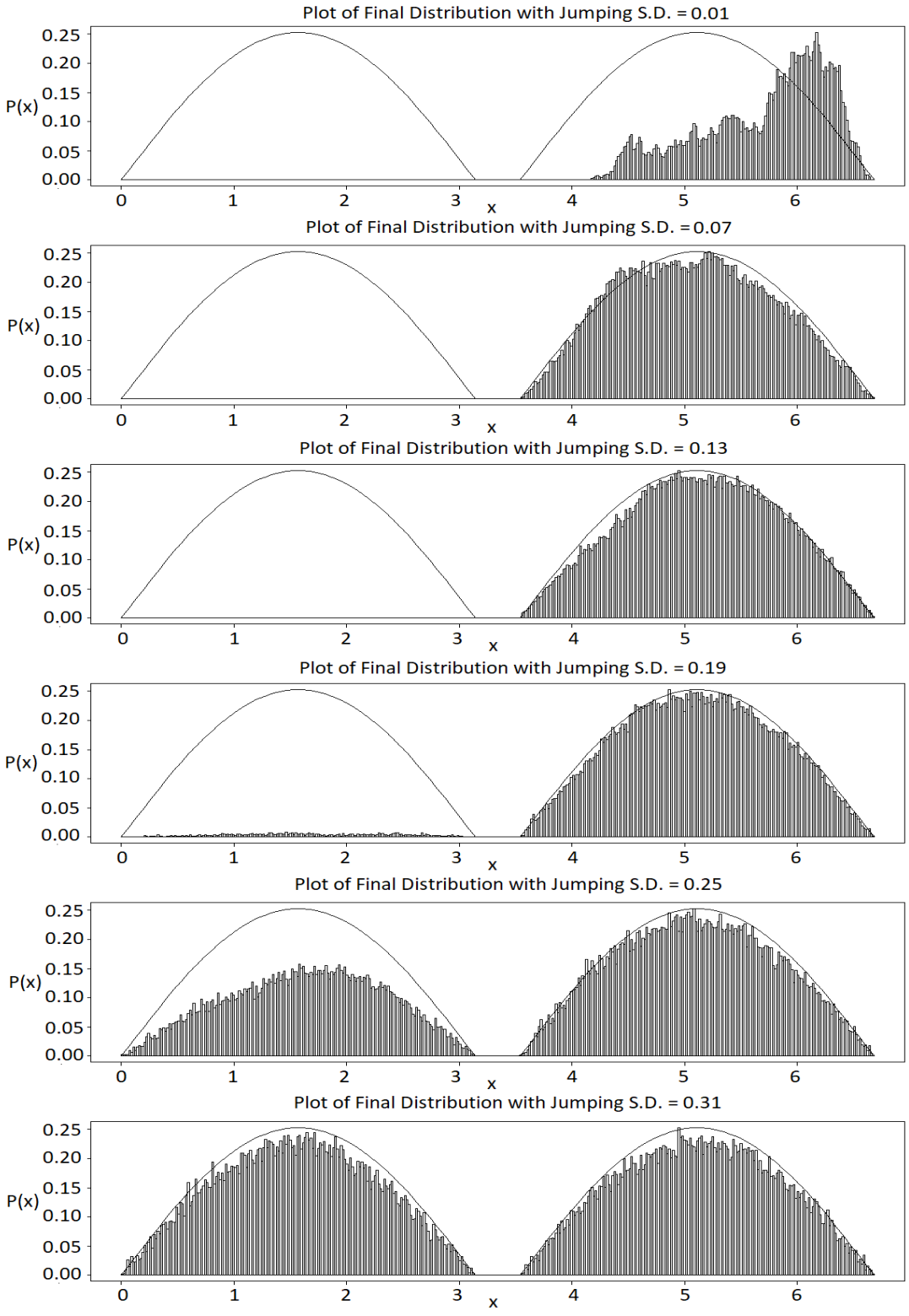


Figure 9: The chains after 100,000 iterations on the probability distribution defined in Figure 8 using different jumping probabilities.

From Figure 9 above it is clear that once the standard deviation of the jumping distribution goes above 0.19 the chain has a reasonable possibility of passing from one node to another within N iterations and therefore converging to the correct

target distribution. This can be shown to be the case by looking at the probability density function for the Gaussian distribution, $f$.

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

In which $\mu$ is the mean, $x$ is the value to determine the probability of observing, and $\sigma^2$ represents the standard deviation. To determine the probability of all values above the gap size, 0.4 it is required to integrate the Gaussian distribution density function to obtain the cumulative density function (CDF)[14].

$$CDF = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx,$$
$$CDF = \frac{1}{2}\left[1 + Erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right].$$

Where $Erf(x)$ is the error function,

$$Erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt.$$

Now using the values for the standard deviation, $\sigma^2 = 0.19$ and the mean, $\mu = 0$ one can work out the probability that any given step size will bridge the gap between the probability peaks. Entering these values and setting $x$ to be the gap size, 0.4 one obtains,

$$\frac{1}{2}\left[1 + Erf\left(\frac{0.4-0}{0.19\sqrt{2}}\right)\right] = 0.9823658.$$

This is the probability that there will be a step size lower than 0.4. Now the probability can be obtained by setting $P = 1 - 0.9823658 = 0.0176342$ Therefore we have approximately a 1.7% chance that the gap size will be sufficiently large to bridge the gap, however, this step size will not always be enough as the chain will rarely be at the edge to allow the gap to be bridged. One can directly determine how many times the chain bridged the gap by the following adjustment to the Metropolis-Hastings algorithm;

1. At each iteration of an accepted proposed change note the previous value.
2. If this new value has passed into the other side of the gap count the occurrence, $N_{cross}$, if not repeat 1.
3. Continue this process for the other side when the walker has successfully passed the gap.

Applying this method to the random walker process one can estimate the probability of transition from peak to peak,

$$P_{cross} = \frac{N_{cross}}{N}.$$

If the random walk process is now repeated with the jumping standard deviations in Figure 9. One can now obtain a value for the $P_{cross}$ in each variant, it is clear the difference this makes as the standard deviation increases.

| Standard Deviation | $P_{cross}$ |
|---|---|
| 0.01 | 0 |
| 0.07 | 0 |
| 0.13 | 0 |
| 0.19 | 0.00002 |
| 0.25 | 0.00025 |
| 0.31 | 0.00054 |

From looking at Figure 9 it is clear to see where these transitional probabilities become important, if the probability is too low as in the $S.D. = 0.19$ case it is likely to only transition once which means that if that one peak is not accurately explored before that transition it is likely to not be fully explored at all. This can be mitigated by having a longer run time but that is not always feasible in multi-dimensional models.

Also notable is the first case in which the $S.D. = 0.01$. This chain has not converged to the target distribution because as previously discussed the step size is too low to sufficiently converge given the number of iterations, in this case it is important to have a higher standard deviation.

As discussed by Sherlock et al[15], for this case the optimal value for the acceptance rate is 23.4%. The Acceptance Rate is the ratio of accepted jumps to rejected ones. The acceptance rate can be decreased by increasing the standard deviation of the jumping distribution. Therefore, one is able to decrease the acceptance rate to $\approx 23\%$ consistently when the jumping distribution standard deviation is increased to 7.0. Through doing this it is possible to obtain $P_{cross} = 11.37\%$. This percentage represents a high amount of cross-over and negates the effects of the barrier allowing the chain to explore different regions freely, the cost of this is the finer details at the peak of each probability area is lost. This is visible within the graphic below comparing $S.D. = 1.0$ and $S.D. = 7.0$.

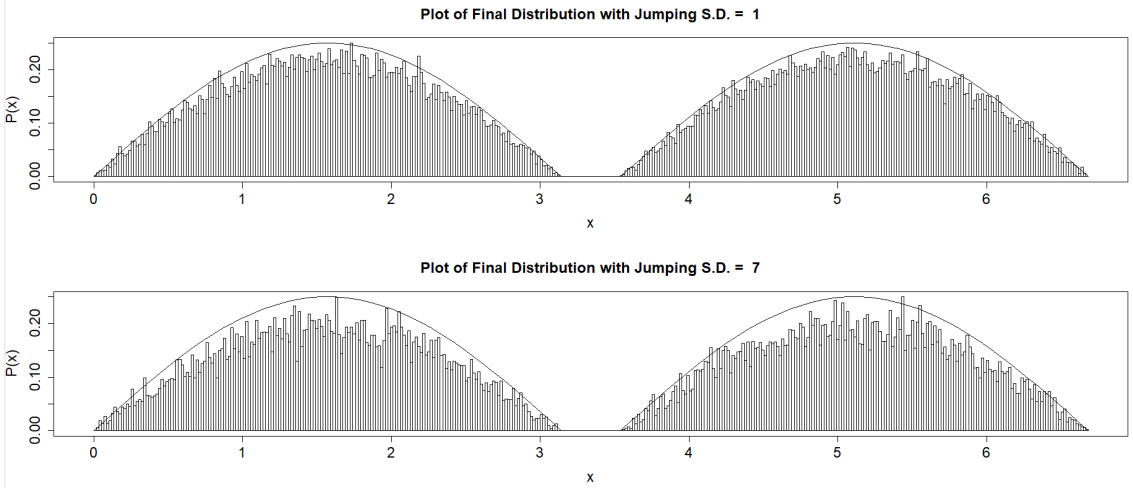


Figure 10: A comparison of both final distributions after 100,000 iterations with jumping distribution standard deviation = 1 and 7 respectively.

From these preliminary tests it is clear that the optimal jumping distribution standard deviation is within the range of 1 and 7, therefore the Metropolis-Hastings algorithm can be executed using this range.

### 6.1.3 Analysing the Result

Now two Metropolis-Hastings algorithm iterations will be executed with; $N = 1,000,000$, *initial value* $= 6$, and *jumping distribution S.D.* $= [1, 7]$. The results can be analysed to determine the accuracy of this execution by using the known target distribution method.

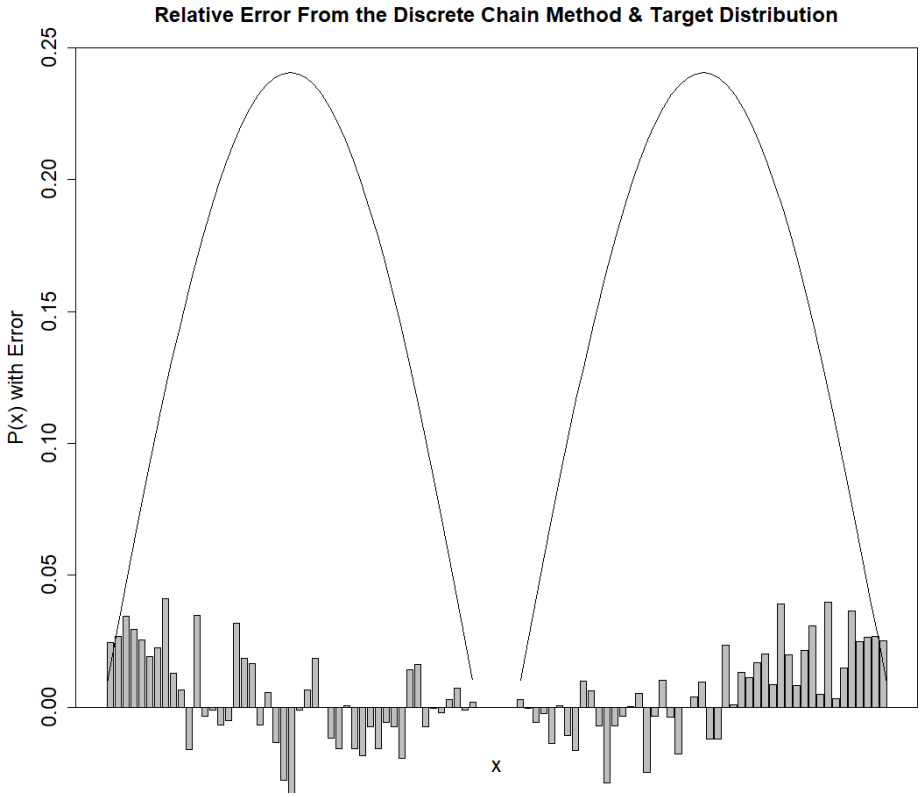


Figure 11: For S.D = 7.0, with the bars on the bottom representing the relative error at each part of the chain as it passes through the target distribution, the negative bars represent areas where the chain was lower than target distribution, and positive are higher than target distribution.

Figure 11 shows that the chain has areas with greater error than others, these areas exist because the jumping distribution standard deviation used does not allow the chain to explore areas with finer details than a smaller standard deviation. The bottom of each curve in Figure 11 are particularly susceptible to this problem due to the rapidly changing probability and lower areas of probability resulting in less accurate sampling. This can be mitigated with additional simultaneous runs.

## 6.2 Unknown Multi-Modal Distribution

Metropolis-Hastings algorithms are primarily utilized to profile an unknown target distribution, this example distribution is typical for the glucose levels relative to the baseline in a normal healthy person upon consumption of a variety of meals.[16]

It is required to consider the distribution to be sampled first. Start by having some assumptions about the distribution, it is known that it has multiple nodes and that it goes above and below 0 on the x-axis, this information can be used to run a few short trial chains to get an idea of the type of distribution that is behind the data which will inform a longer more detailed run afterwards. Arbitrarily select initial values $-6, -4, -2, 0, 2, 4, 6$ with a chain length of $N = 10,000$ for each, We shall now plot each of these final distributions upon this graph atop each other to see

where potential nodes may exist.

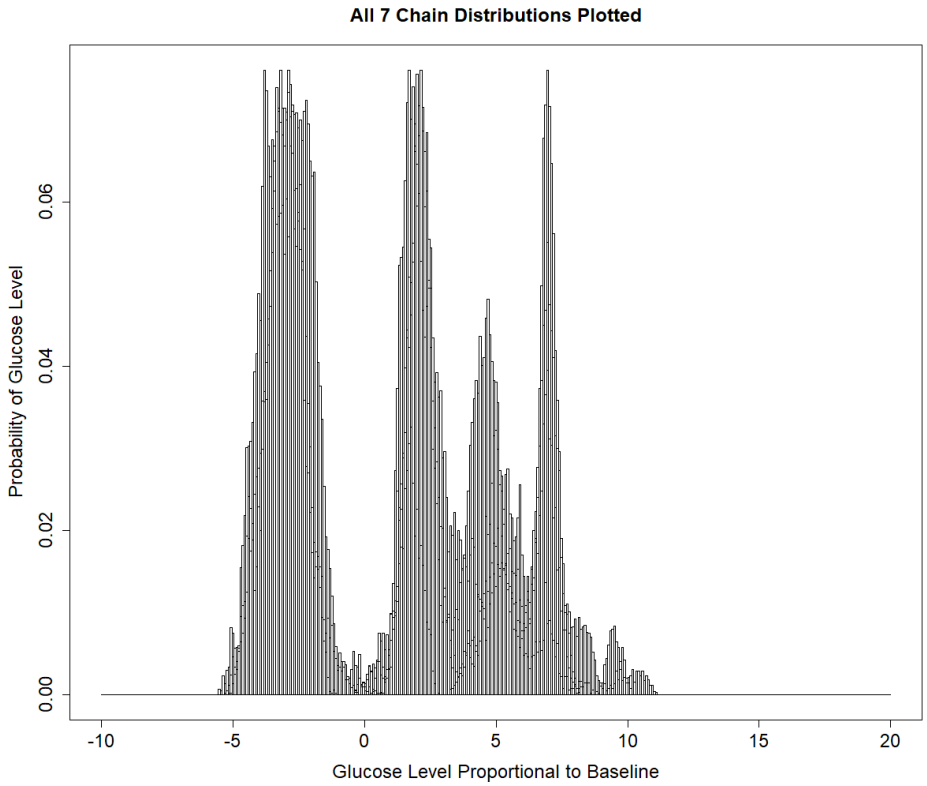


Figure 12: A combination of all 7 histograms for initial values $-6, -4, -2, 0, 2, 4, 6$ and with chain length, $N = 10,000$. This plot is unscaled and shows the peaks amongst the chains.

This chain was executed by using a Gaussian prior distribution centered around the parameter of choice with a standard deviation of 1.0 to represent uncertainty with these values. The jumping distribution standard deviation used was 0.10, as a trial to determine more information about our target distribution.
Now this initial testing phase has been completed it is possible to observe what the target distribution may look like, there are 3 obvious peaks at the points $x \approx -3$, $x \approx 2$, and $x \approx 7$. The main execution shall be focused on these points. It is also possible to see that the parameter value spans the range of $-5$ to 15, therefore, the jumping distribution standard deviation is required to be such that the walker is able to fully explore that range. I propose we use a jumping distribution S.D. of 0.2 as that will allow the walker to jump any barriers that may exist in the data as there appears to be one at $x = 0$ from Figure 12. Now the 3 chains can be run again with the initial conditions $x = -3, 2, 7$ and a larger number of iterations $N = 1,000,000$. The chains can be plotted on the same axis after each iteration to get an illustration of the target distribution from these chains.

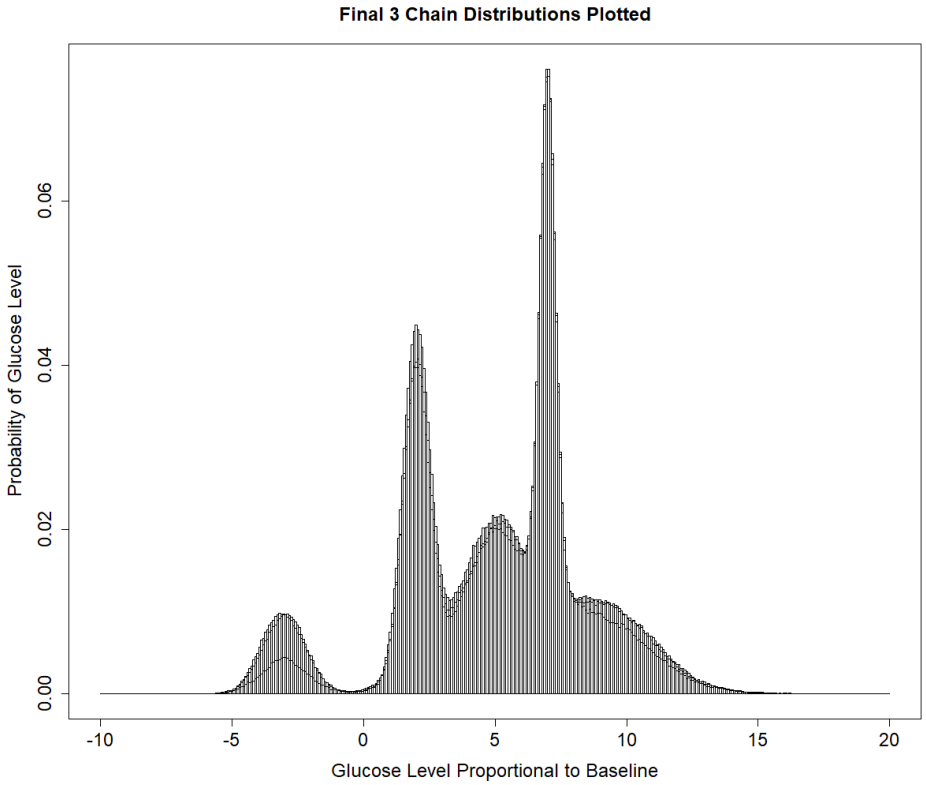


Figure 13: All 3 histograms for initial values $-3, 2, 7$ and with chain length, $N = 1,000,000$ plotted overlapping.

Now a much smoother picture of the target distribution is visible. The nodes that were initially guessed are fully sampled curves and there are visible overlapping Gaussian distributions, as would be typical of the expected results from the glucose experiment. For comparison, the original distribution is shown below.

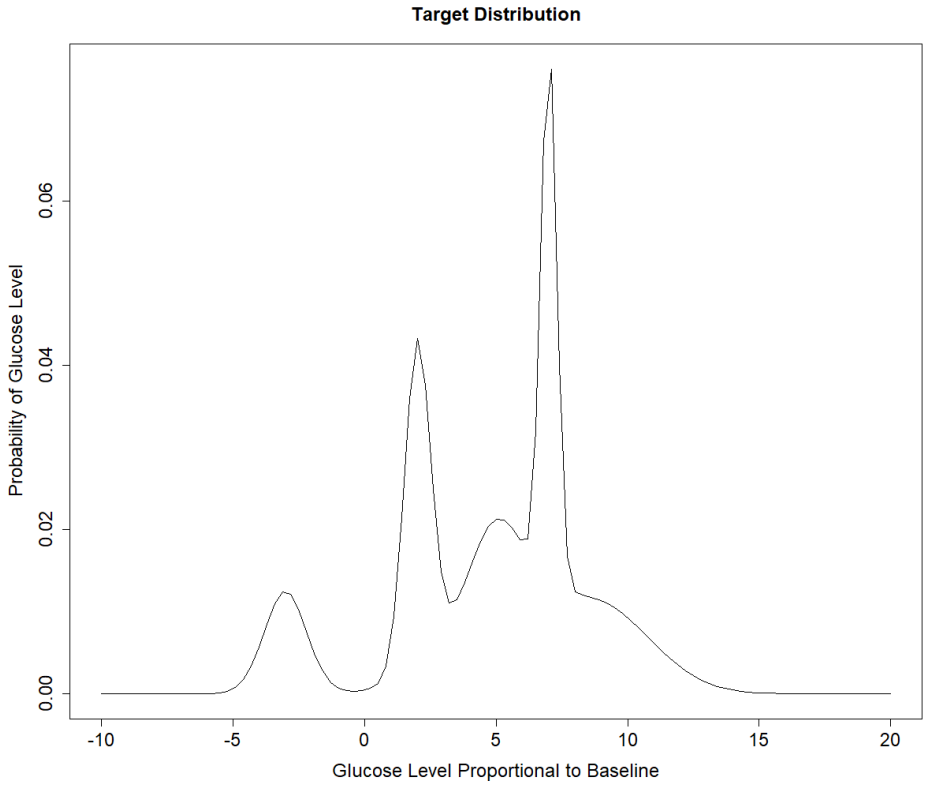


Figure 14: The intended target distribution from figure 13.

Each one of these 3 chains that have been generated is extremely similar to the intended target distribution and therefore it is clear that this has been a successful run that has effectively crossed any gaps such as the one at zero.

# 7 Parameter Estimation

A parameter estimation method was looked into in Chapter 4.2, but the Metropolis-Hastings algorithm shows its true potential when used to determine unknown parameters in a particular model given some data relevant to that parameter. In Chapter 4.2 it was a structured example but in the following sections, more realistic examples will be undertaken starting with a simple example before moving onto a more complex one.

## 7.1 Model Fitting Age to Years Spent Working

For this example data obtained from kaggle.com[17] will be used, this data is about the employees of IBM and has 29 variables, but for this initial example only the age of an employee will be considered, alongside the years that they have spent working during their life. This is highly likely to be positively correlated in a linear model.

The investigation begins by removing all undesired variables except employee age and years spent working.


Figure 15: Graph visualizing the data on individuals ages vs their years spent working.

In Figure 15 the line represents people that have started working at 18 and have continued to work until they drop off that line by taking any time off of work. This is why the graph has the shape displayed.

A parameter estimation using the Metropolis-Hastings algorithm can now be undertaken. This is begun by selecting a model to use. For this exercise it is clear that the linear model is desired as from the graph there is clearly a relationship in the form of

$$\text{years spent working} = \alpha * \text{age} + \beta.$$

Now the model has been chosen, the likelihood function needs to be defined. This function will be the normally distributed probability of obtaining the parameter value currently given the one that the data suggests. This likelihood function is

based on the data that is available and therefore encourages parameters to shift such that they fit that data.

There is no error parameter for this investigation as we would like to see how the two parameters interact.

The priors will be not based on much prior knowledge which allows one to have a mostly *data driven* investigation into these parameters - this means that the standard deviation of the prior distribution will be high to account for uncertainty.

The jumping distribution chosen to use is the Gaussian distribution from earlier with a standard deviation of 0.1 for each parameter.

Now the chain is set-up it can be executed using a value of $N = 10,000$, $\alpha_{initial} = 0.5$, and $\beta_{initial} = -10$.

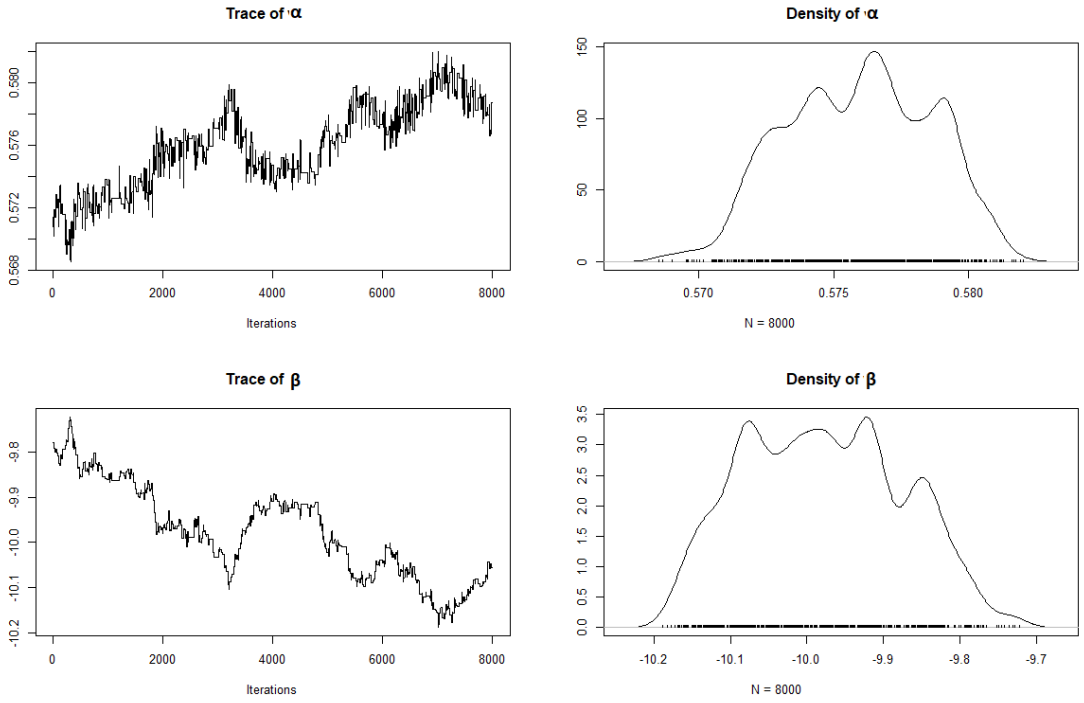


Figure 16: The state of the chain after 10,000 iterations and a 2,000 burn in phase, the left side depicts the chain path and the right side depicts a smoothed histogram of the chain values.

Figure 16 shows that the initial guesses were somewhat close to being correct and the chain has slowly converged to a stationary point, The uncertainty of these results is displayed in the graphs on the right through the variance on them, but this is low as shown by the tightness of the graphs and the standard deviations of the parameters below.

- $\alpha = 0.554, S.D = 0.002726$
- $\beta = -10.0292, S.D = 0.103469.$

The results can be shown to be correct by fitting a linear best-fit line into our data and comparing the results of that to the obtained results.

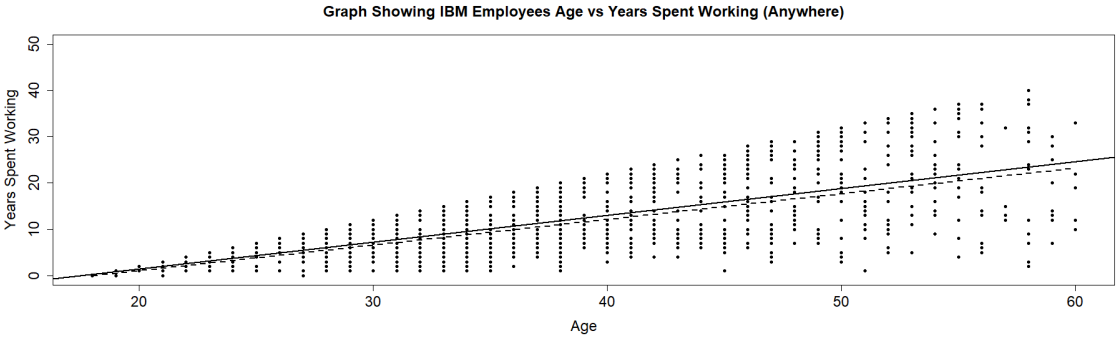


Figure 17: The obtained fitted model to the data alongside a linear best-fit line for this data, the line obtained by the chain is depicted by the dashed line with the best-fit line in bold.

## 7.2 Model Fitting Titanic Survival to Data

For this section data will be used from kaggle.com regarding Titanic passengers.[18] This data has been studied extensively in the past and is one of the most popular datasets on kaggle.com. There is a lot of intuitive understanding that can be obtained from the data that allow for informed guesses to the prior distributions of parameters to be made. The paper[19] discusses this intuition and from which we shall be deriving our prior distributions.

### 7.2.1 Parameters

For this investigation, we will be using 6 variables and 1 outcome. We desire to create a model that will accurately predict the outcome based on the weightings of the 6 variables. The variables, $x_i$, and the outcome variable $y$ are;

Table 1: Variables to be used within this model and their meanings

| Representation | Variable | Range, $x_i \in$ |
|---|---|---|
| Class | $x_1$ | {1,2,3} |
| Sex | $x_2$ | {"Female","Male"} |
| Age | $x_3$ | [1,80] |
| Number of Siblings/Spouse on board | $x_4$ | [0,5] |
| Number of Children/Parents on board | $x_5$ | [0,6] |
| Price paid to board | $x_6$ | [0,512.32] |
| Passenger Survived | $y$ | {0,1} |

First I shall transform the categorical Sex variable into a numerical representation, for simplicity the following change will be undertaken "$Male$" $\rightarrow -1$, and

"*Female*" $\rightarrow$ 1. The survived variable represents whether a passenger survived with 0 representing "No" and 1 representing "Yes".

### 7.2.2 Data Scaling

Now all of the data will be scaled around the mean of 0 so it is clear which variables are weighted higher than others and thus have more influence in the inference process[20]. This scaling process is undertaken by the following method;

$$x_i = \frac{x_i - \overline{x}}{\sqrt{\frac{\sum_{i=1}^{N}(x_i - \overline{x})^2}{N-1}}},$$

Where,

$$\overline{x} = \sum_{n=0}^{n=N} x_n.$$

This process will ensure that all of the data has a mean of 0 and a standard deviation of 1, which ensures that variables such as $x_6$ with a range of 512.32 will have an equal effect to the variable $x_2$ with a range of 2.

### 7.2.3 Prior Distributions

Now prior distributions for each parameter need to be generated which will go into calculating the total prior, this will do this using the knowledge that already exists on the Titanic.

### 7.2.4 Class, $x_1$

With class now being scaled it has 3 different possible values
$x_1 \in \{0.9105940, -0.2823677, -1.4753294\}$ which represent 3rd, 2nd, and 1st class respectively. Now prior knowledge can be utilized that those in the 1st class had a higher chance of survival due to their positioning higher in the ship than those in 3rd which were situated in the basement of the ship. Therefore, $x_1$ will increase the probability of survival as it goes down. The following prior will be assigned to $x_1$ with the parameters $\mu$ and $\sigma$ explained below,

$$f(x_1|\mu,\sigma) = \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right).$$

In which $\mu$ and $\sigma$ represent the mean and standard deviation respectively and for $x_1$ these will be -1, and 5 with the -1 representing the belief that there is a negative correlation with Class and Survival and the standard deviation representing our uncertainty with this belief. This equation is the log-normal density function, which will now be represented by $f(x|\mu,\sigma)$.

### 7.2.5 Sex, $x_2$

Now the sex of the passengers is represented by a binary option of either $x_2 = -0.7585196$, or $x_2 = 1.3165110$ representing female and male respectively. It is known that passengers were mistakenly ordered to allow "women and children first" which resulted in a proportionally lower number of male survivors. However, there were still many male survivors despite this order as it was not strictly followed, with this knowledge one can infer that as $x_2$ increases survivability decreases but it is a stronger correlation than class, therefore the following model will be utilized,

$$f(x_2| - 1.5, 5).$$

### 7.2.6 Age, $x_3$

Age has been scaled into a range of $X_3 \in [-1.475329, 0.910594]$ with higher numbers representing older passengers. This is likely to have also been effected by the statement of "women and children first". However particularly young passengers have a higher chance of survival due to the decreased space occupied on lifeboats, additionally, older passengers were more likely to be in 1st class and thus have increased survivability due to that. Therefore, the following model is a good expression of these prior beliefs, with the negative correlation being lower than that of $x_3$, but with similar large uncertainty,

$$f(x_3| - 0.7, 5).$$

### 7.2.7 Siblings/Spouse Number, $x_4$

This variable represents the number of siblings and/or spouse on board the boat at the time. It has been scaled into the form of $x_5 \in [0.5242027, 4.8262796]$ which represents 0 and above. This variable is relevant as passengers were less likely to board lifeboats without these passengers also, which brings up the survivability of the group as a whole. However, people were less likely to survive by refusing to board lifeboats without loved ones additionally. This variable is significantly less influential than others and it is uncertain whether it would be positive or negative, therefore the best option is a fully uninformed prior of

$$f(x_4|0, 5).$$

### 7.2.8 Children/Parent Number, $x_5$

This is similar to variable, $x_4$, but measures the number of children and/or parents on board ship, after scaling it is now $x_5 \in [-0.5055408, 6.5260715]$ with higher numbers representing more children and parents, and there is a value for 0 if a child is traveling with solely a nanny. There is a higher probability of an individual surviving if they have many children as they will have to travel with the children to board the lifeboat. Additionally, a child is more likely to survive with 2 parents than 1 as they have more care and are better able to navigate the chaos aboard

the boat. Therefore, it can be anticipated that the prior of this variable is strongly correlated positively with survival. Giving a Prior of,

$$f(x_5|1,5).$$

### 7.2.9 Fare, $x_6$

This variable represents the cost of a ticket that a passenger paid to board the Titanic. After scaling the data is within the range, $x_6 \in [-0.6556163, 9.0257813]$. This data will go up as the passenger has paid more to board the Titanic, and individuals paid more for a higher class ticket. It is possible to deduce that the passenger data within $x_6$ is highly correlated to the class of a passenger, $x_1$. However, as this parameter increases it will reduce the class of a passenger. It is therefore likely that as this variable increases the survivability will also increase giving a sensible prior of,

$$f(x_6|1,5).$$

### 7.2.10 Combining all Priors

Now one overarching prior is needed for the chain which considers all variables that we have. This prior will be obtained by a summation of all priors above as we are using a logarithmic prior and a logarithmic Metropolis-Hastings. So the logarithmic prior takes the form of,

$$\begin{aligned} P(x_1,x_2,x_3,x_4,x_5,x_6) = & \\ & f(x_1|-1,5) + f(x_2|-1.5,5) + f(x_3|-0.7,5) \\ & \qquad + f(x_4|0,5) + f(x_5|1,5) + f(x_6|1,5) \end{aligned}$$

The individual priors are shown below;

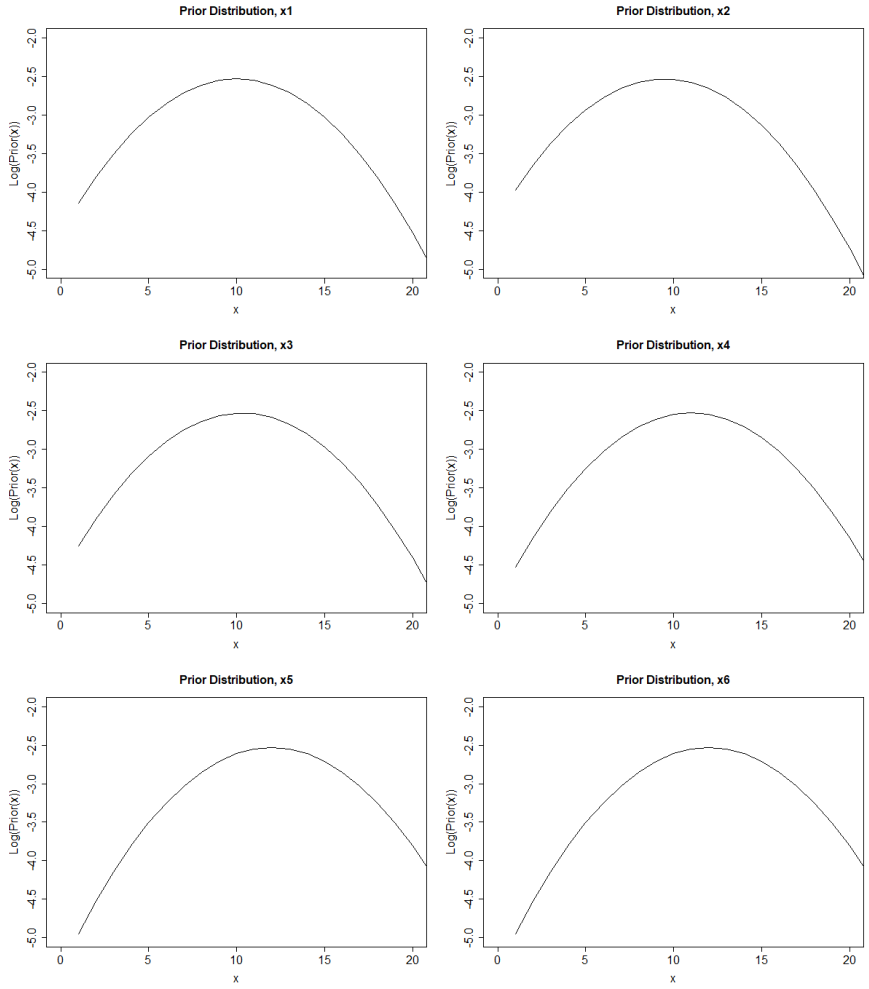


Figure 18: The log prior distributions for each parameter of the model.

Now each prior from figure 18 can be combined to give the final prior distribution to be used within the Metropolis-Hastings algorithm, with the below graphic representing the 2-D case in which each parameter is identical. A true representation would be 6-D.

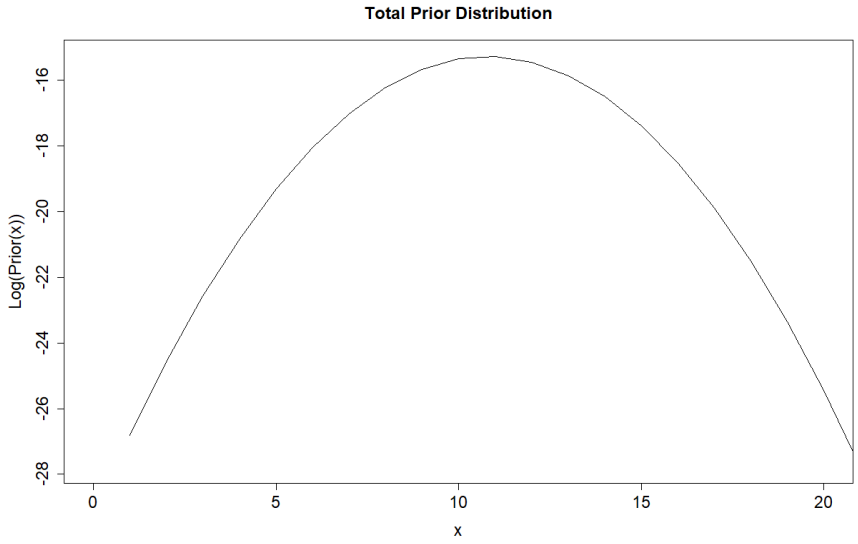


Figure 19: The total log prior distribution for all parameters of the chain.

### 7.2.11 Jumping Distribution

For this Metropolis-Hastings execution, the standard Gaussian jumping distribution will be used, the standard deviation for each jump will be 1, as this allows full exploration of all parameter space whilst still being somewhat constrained within the reasonable values of the model.

### 7.2.12 Likelihood Function

The model most appropriately fitting for this type of analysis is a linear model with the form:

$$x_1 * Class + x_2 * Sex + x_3 * Age + x_4 * Sibling/Spouse\ Number \\ + x_5 * Parents/Children\ Number + x_6 * Fare = y.$$

The likelihood function was thereafter generated from the data available. This was achieved by determining the value for $y$ with the current variable values then working out the difference between this and the value that would be obtained using the data. Then the likelihood is returned in the form of a normal probability density with mean at the data $y$ and we obtain the probability of getting our obtained variables. The standard deviation used in this likelihood intended to represent the variability of our data is 0.01. That means that if we had variables that suggested our passenger may not survive with 0.1, but the data suggested survival would be 0.5 there would be a low likelihood for this case as 0.1 is a low probability of occurring in a normal probability function with a standard deviation of 0.01.

### 7.2.13 Execution of Algorithm

For this application of the Metropolis-Hastings algorithm, there will be, $N = 240,000$ iterations over 6 chains which allows a Gelman-Rubin convergence measure to be undertaken to asses the measure of convergence of the chains. My initial values will be dispersed over a range of -1 to 1 allowing for a full exploration of the parameter space by each chain. There will be a burn-in period of 2,000 iterations to ensure fair representation of the chain. The results are shown below of the initial chain execution.

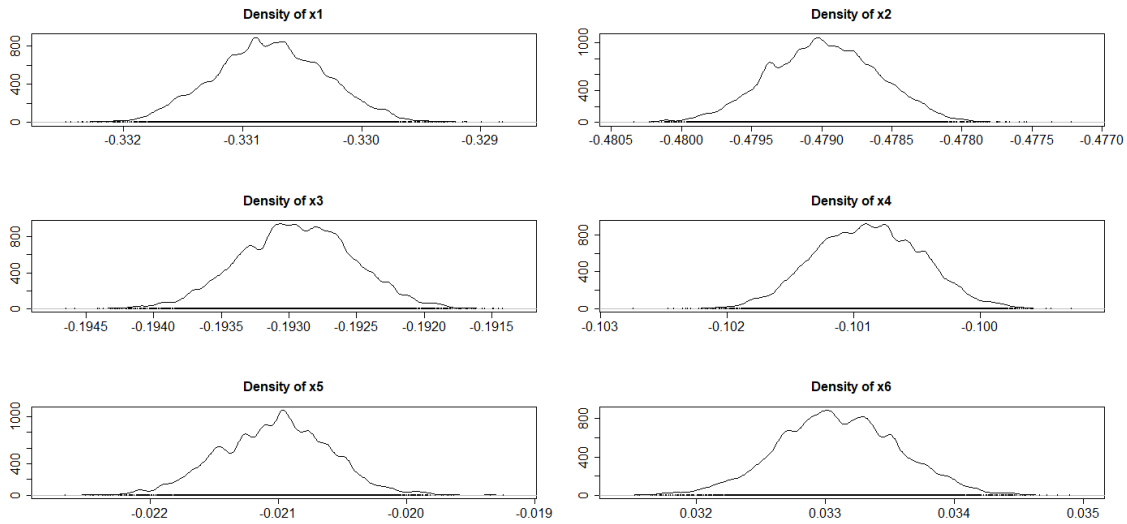


Figure 20: The distribution for each parameter for a run with all start values of 0.

The results of the Gelman-Rubin convergence measure for each parameter is $x_1 = 1.08$, $x_2 = 1.03$, $x_3 = 1.06$, $x_4 = 1.00$, $x_5 = 1.01$, and $x_6 = 1.09$. From this it can be deduced that all of the parameters have converged within an acceptable limit and now the effect of each parameter on the survival of a passenger on-board the Titanic can be analysed.

The mean value for each parameter is outlined below:

| Variable | Mean | Standard Deviation |
|---|---|---|
| $x_1$ | -0.33077 | 0.0004791 |
| $x_2$ | -0.47898 | 0.0003942 |
| $x_3$ | -0.19294 | 0.0004215 |
| $x_4$ | -0.10088 | 0.0004231 |
| $x_5$ | -0.02103 | 0.0004267 |
| $x_6$ | 0.03310 | 0.0004644 |

With this, we can make a model to predict the survival of a passenger by substituting the values we have above into the variables of the linear model previously defined.

These results also allow you to see how some characteristics of a passenger affect survival on the Titanic relative to others. Additionally, it is clear to see that Sex is the most important factor for a passengers survival, shortly followed by the class of a passenger.

We can verify the data-driven nature of these results by running another Metropolis-Hastings algorithm but this time we are starting from fully-uninformed priors of all Gaussian distributions with a mean of 0 and S.D. of 0.1. After 240,000 iterations the results are identical to the above results showing that the results are correct and unaffected by any incorrect priors.

This method has a large number of applications to the wider world for data analysis. It is used by corporations to determine characteristics of customer actions. For example, Capital One uses this linear parameter estimation method[21] to work out whether to take on a customer for a credit card. In this case, the available data is any that they have on the customer such as previous loans, age, location, etc. and the linear model will predict the probability of a customer defaulting on their loans, this probability is within the risk tolerance then the customer will be accepted.

# 8 Summary

Throughout this project, I have outlined how to use the Metropolis-Hastings algorithm to complete analysis on unknown probability distributions and how to estimate parameters within a Bayesian framework. I have begun by explaining the Metropolis-Hastings algorithm and what is required for it to function correctly. I then had a very simplistic example in Section 4.1, this was used to show how the chain samples the space available to it to result in a histogram that is shaped identically to the target distribution. I then introduced parameter estimation within a linear model using the Metropolis-Hastings algorithm in Section 4.2. After introducing the core concepts and potential of the Metropolis-Hastings algorithm I introduced multiple methods that can be utilized to analyse the convergence of the Metropolis-Hastings algorithm using the chains that it generates in Section 5. Included in these analytical methods is a method I created for easy visualization of error within a chains stationary distribution in Section 5.2.3. I then went on to discuss where the Metropolis-Hastings algorithm can fail such as in multi-modal target distributions resulting in nodes of low probability that the chain may be unable to cross. I discussed this to highlight the importance of tweaking the jumping distribution when running the algorithm in Section 6. I then showed how you can obtain an estimation to an unknown multi-modal target distribution by having investigatory beginning executions to determine where nodes are and then tailoring the final longer executions to account for these nodes. I finally went on to conclude my project with two very illustrative examples of parameter estimation in Section 7. I first worked on a linear fitting model for IBM employee data to highlight how parameter estimation will work, after that I analysed the Titanic passenger data to determine characteristic factors that contributed to a passengers survival and their respective weightings on whether a passenger will survive or not. The motivation behind this project was to give the reader an intuitive understanding of how to use the Metropolis-Hastings algorithm and how to analyse and utilize any results that it

yields. This project will act as a useful resource for anyone interested in applying the Metropolis-Hastings algorithm to complete parameter estimation or the modeling of unknown multi-modal distributions.

## 8.1 Code Listing

All of the code used to generate the graphics and the Metropolis-Hastings executions can be found stored on my GitHub at `https://github.com/jordpears/Maths-Project`

## 8.2 Future Research

This project could be further extended by altering the Metropolis-Hastings acceptance ratio to be asymmetric as this can reduce the execution time of the algorithm and increase the accuracy over multiple simultaneous executions as shown within this paper[22]. I could take these adjustments and apply them to my examples comparing the speed increase to determine a quantitative measure of the gain that is possible through using the asymmetric formulation.

## 8.3 Acknowledgements

I would like to thank Dr. Gustav Delius for his valuable insight and guidance during the development and research stages of my project.

# References


[1] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in practice*. CRC press, 1995.

[2] L. Lacasa and R. Flanagan, "Time reversibility from visibility graphs of non-stationary processes," *Physical Review E*, vol. 92, no. 2, 2015.

[3] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, 1970.

[4] C. Robert and G. Casella, "Monte Carlo statistical methods Springer-Verlag," *New York*, 2004.

[5] M. Bayes and M. Price, "An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfrs," *Philosophical Transactions (1683-1775)*, 1763.

[6] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[7] M. Plummer, N. Best, K. Cowles, and K. Vines, "Coda: convergence diagnosis and output analysis for MCMC," *R news*, vol. 6, no. 1, 2006.

[8] K. L. Mengersen, C. P. Robert, and C. Guihenneuc-Jouyaux, "Mcmc convergence diagnostics: a reviewww," *Bayesian statistics*, vol. 6, 1999.

[9] C. J. Geyer, "Practical Markov chain Monte Carlo," *Statistical science*, 1992.

[10] A. Gelman and D. B. Rubin, "Inference from iterative simulation using multiple sequences," *Statistical science*, 1992.

[11] D. B. Gelman, Andrew & Rubin, "Avoiding model selection in Bayesian social research," *Sociological methodology*, vol. 25, 1995.

[12] R. J. Patz and B. W. Junker, "A straightforward approach to Markov chain Monte Carlo methods for item response models," *Journal of educational and behavioral Statistics*, vol. 24, no. 2, 1999.

[13] J. Sui, R. Huster, Q. Yu, J. M. Segall, and V. D. Calhoun, "Function–structure associations of the brain: evidence from multimodal connectivity and covariance studies," *Neuroimage*, vol. 102, 2014.

[14] S. Kokoska and D. Zwillinger, *CRC standard probability and statistics tables and formulae*. CRC Press, 1999.

[15] C. Sherlock and G. Roberts, "Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets," *Bernoulli*, 2009.

[16] G. Freckmann, S. Hagenlocher, A. Baumstark, N. Jendrike, R. C. Gillen, K. Rössner, and C. Haug, "Continuous glucose profiles in healthy subjects under everyday life conditions and after different meals," *Journal of diabetes science and technology*, vol. 1, no. 5, 2007.

[17] IBM, "IBM HR analytics and attrition dataset." [Online]. Available: https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset

[18] "Titanic passenger data." [Online]. Available: https://www.kaggle.com/c/titanic/data

[19] W. Hall, "Social class and survival on the SS Titanic," *Social science & medicine*, vol. 22, no. 6, 1986.

[20] B. P. Carlin and T. A. Louis, *Bayesian methods for data analysis*. CRC Press, 2008.

[21] "How does Capital One differentiate itself in the card industry?" [Online]. Available: https://www.forbes.com/sites/sap/2018/05/01/workforce-inclusion-is-a-strength-for-any-company/#3a0e82004041

[22] C. Andrieu, A. Doucet, S. Yıldırım, and N. Chopin, "On the utility of Metropolis-Hastings with asymmetric acceptance ratio," *arXiv*, 2018.