

# Parameter Estimation Problem Of A Population Dynamics Model Using Monte Carlo Markov Chain Method

A thesis submitted to  
Indian Institute of Science Education and Research Pune  
in partial fulfillment of the requirements for the  
BS-MS Dual Degree Programme

Thesis Supervisor: Dr. Amit Apte

by  
Purvi Tiwari  
April, 2014



Indian Institute of Science Education and Research Pune  
Sai Trinity Building, Pashan, Pune India 411021



This is to certify that this thesis entitled “Parameter Estimation Problem Of A Population Dynamics Model Using Monte Carlo Markov Chain Method” submitted towards the partial fulfillment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research Pune, represents work carried out by Purvi Tiwari under the supervision of Dr. Amit Apte.

ADVISOR

Dr. Amit Apte

LOCAL COORDINATOR

Dr. Pranay Goel





# Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor Dr. Amit Apte for the continuous support of my MS thesis, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of my project and writing this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Pranay Goel for his encouragement, insightful comments, and hard questions.

I thank my fellow lab-mates in IISER, TIFR and ICTS Group: Roshni Patil, Ankita Sharma, Aditi Jakhar, Deepika Ananda, Shubham Pandey, Ashok Choudhary, Raman Gupta for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we had in the last year.

Last but not the least, I would like to thank my parents and my siblings for supporting me throughout the whole year.

# Abstract

We will consider a model for the temporal dynamics of population size of organisms with multiple life stages (egg, larvae, pupa, and adult stages in insects) which are influenced by the nonlinear, density-dependent feedback mechanisms operating at different stages and the interactions of these mechanisms. The model involves several independent interacting parameters. The main aim of the project will be to devise Markov Chain Monte Carlo based methods in order to estimate parameters of this model, based on the data available. We will mainly focus on using synthetic data to study the performance of these MCMC methods.



# Contents

<b>1</b>	<b>Introduction</b>	<b>10</b>
1.1	Overview . . . . .	10
1.2	Objective . . . . .	10
1.3	Thesis Organization . . . . .	10
<b>2</b>	<b>The General Discrete Inverse Problem</b>	<b>12</b>
2.1	Overview . . . . .	12
2.2	Model Space . . . . .	13
2.3	Data Space . . . . .	13
2.4	Forward Problem . . . . .	14
2.5	Measurements and A Priori Information . . . . .	14
2.5.1	Results of the Measurements . . . . .	14
2.5.2	A Priori Information on Model Parameters . . . . .	14
2.6	Defining a solution of the Inverse Problem . . . . .	15
2.6.1	Combination of Experimental, A Priori, and theoretical Information . . . . .	15
2.7	Using the Solution of the Inverse Problem . . . . .	15
2.7.1	Computing Probabilities . . . . .	15
<b>3</b>	<b>Monte Carlo Method</b>	<b>21</b>
3.1	Overview . . . . .	21
3.2	Markov Chain Theory . . . . .	21
3.3	Markov Chain Monte Carlo Idea . . . . .	22
3.4	Metropolis Hastings Algorithm . . . . .	22
3.5	The Practice of MCMC . . . . .	30
3.5.1	Black Box MCMC . . . . .	30
3.5.2	Pseudo Convergence . . . . .	31
3.5.3	Burn-In . . . . .	31
3.5.4	Diagnostics . . . . .	31
3.6	Kalman-Filter Equations . . . . .	32
<b>4</b>	<b>Parameter Estimation of a Population Dynamics Model</b>	<b>34</b>
4.1	Overview . . . . .	34
4.2	Introduction . . . . .	34

4.3	The Model Development . . . . .	35
4.3.1	The full two-variable model for Drosophila population dynamics . . .	35
4.4	Model Explanation . . . . .	36
4.5	Bifurcation Theory . . . . .	36
4.6	MCMC Scheme . . . . .	42
4.6.1	Model Space . . . . .	43
4.6.2	Data Space . . . . .	43
4.6.3	Model Function . . . . .	43
4.6.4	Prior . . . . .	43
4.6.5	Likelihood . . . . .	44
4.6.6	Posterior . . . . .	44
4.6.7	Proposal . . . . .	44
4.7	MCMC Algorithm . . . . .	44
4.8	Conclusion . . . . .	49
4.9	Application potential . . . . .	50



# Chapter 1

## Introduction

### 1.1 Overview

An important part of life of any scientist is comparing theoretical models to data. For that we are trying to study a approach to estimate the most probable set of parameter estimation method for nonlinear models. Hence in this thesis we explain a very efficient tool to estimate the parameters for high-dimensional model. The method is known as Markov Chain Monte Carlo (MCMC). MCMC was first introduced in the early 1950s by statistical physicists (N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller) as a method for the simulation of simple fluids. In this thesis we will focus on its use in evaluating the multi-dimensional integrals required in a Bayesian analysis of the models with some or many parameters.

### 1.2 Objective

The objective of this research is to study the concepts of Monte Carlo Markov Chain Method for solving the parameter estimation problem of a population dynamics model. The main aim of the project will be to device Markov Chain Monte Carlo based methods in order to estimate parameters of the model for the temporal dynamics of population size of organisms with multiple life stages,(egg, larvae, pupa, and adult stages in insects) which are influenced by the nonlinear, density-dependent feedback mechanisms operating at different stages and the interactions of these mechanisms. We will mainly focus on using synthetic data to study the performance of these MCMC methods.

### 1.3 Thesis Organization

Chapter 2 describes a general discrete inverse problem. In that we will explain how to define a solution of the inverse problem.

The Monte Carlo method is described in Chapter 3. Markov Chain theory is explained which leads to the Monte Carlo Idea and in that section Metropolis algorithm is explained

followed by some of the characteristics of MCMC. An analytic solution is also described using the Kalman Filter Equations.

Chapter 4 explains the population dynamics model of *Drosophila Melanogaster*. In that we will explain how the model was developed and the description of model parameters which needs to be estimated. In the next section Bifurcation theory is explained to see how the parameters behave with the population dynamics model followed by the MCMC algorithm. Later, the conclusions made from the results and the application potential of MCMC method are described.



# Chapter 2

## The General Discrete Inverse Problem

### 2.1 Overview

In general, we deal with two types of problems: *Forward Problems* and *Inverse Problems*. An **inverse problem** is a general framework that is used to convert observed measurements into information about a physical object or system that we are interested in. The most general theory, that we learnt, is obtained when using a probabilistic point of view, where the priori information on the model parameters is represented by a probability distribution over the 'model space'. So, for that we have started by understanding the concepts of inverse modeling including model space, data space, joint manifolds followed by conditional probability distributions and that leads to the better understanding of formulation of Monte Carlo Markov Chain methods for sampling any given target density.

Given a complete description of a physical system, we can predict the outcome of some measurements. This problem of predicting the result of measurements is called *modelization problem* or *forward problem*. The *inverse problem* consists of using the actual result of some measurements to infer the values of parameters that characterize the system. Unlike the *forward problem*, *inverse problem* does not have a unique solution. As an example, consider measurements of gravity field around a planet: given the distribution of mass inside the planet, we can uniquely predict the values of the gravity field around the planet (forward problem), but there can be many different distributions of mass that give *exactly* the same gravity field around the planet. Therefore, the inverse problem-inferring the mass distribution from the observations of the gravity field-has multiple solutions. Because of this, in *inverse problem*, one needs to make explicit any available a priori information on the model parameters. One also needs to be careful with the data uncertainties. It follows that the results of the measurements of the observable parameters (data), the a priori information on model parameters, and the information on a physical correlations between observable parameters and model parameters can all be described using probability densities. The general

inverse problem can then be set as a problem of 'combining' all of this information. Hence, Monte Carlo techniques are examined for the same. For the study of a physical system we divided the scientific procedure into the following three steps:

- **Parametrization of the system** : Discovery of a minimal set of a model parameters whose values completely characterize the system.
- **Forward modeling**: For given values of the model parameters, to make predictions on the results of measurements on some observable parameters.
- **Inverse modeling**: Use of the actual results of some measurements of the observable parameters to infer the actual values of the model parameters.

Here is the description of some of the terms that we have used in the text.

## 2.2 Model Space

Independently of any particular parametrization, it is possible to introduce an abstract space of points, a manifold, each point of which represents a conceivable model of the system. This manifold is named 'model space' and is denoted  $\mathbf{M}$ . Individual models are points of the model space manifold and could be denoted  $M_1, M_2, M_3, \dots$ . With each point  $M$  of the model space  $\mathbf{M}$  a set of numerical values  $m_1, m_2, m_3, \dots$ . This corresponds to the definition of a system of coordinates over the model manifold  $\mathbf{M}$ . Each point  $M$  of  $\mathbf{M}$  is named a *model*. The number of model parameters needed to completely describe a system may be either finite or infinite.

## 2.3 Data Space

To obtain information on model parameters, we have to perform some observations during a physical experiment, i.e., we have to perform a measurement of some observable parameters. Data space can be defined as the space of all conceivable instrumental responses. This corresponds to the other manifold, the data manifold, which may be represented by  $\mathbf{D}$ . Any conceivable result of the measurements then corresponds to a particular point  $D$  on the manifold  $\mathbf{D}$ .

## 2.4 Forward Problem

Experiments suggest physical theories, and physical theories predict the outcome of experiments. The comparison of the predicted outcome and the observed outcome allows us to ameliorate the theory. To solve a 'forward problem' means to predict the error-free values of the observable parameters  $\mathbf{d}$  that would correspond to a given model  $\mathbf{m}$ . This theoretical prediction can be denoted  $\mathbf{m} \mapsto \mathbf{d} = \mathbf{g}(\mathbf{m})$  where  $\mathbf{d} = \mathbf{g}(\mathbf{m})$  is a short notation for the set of equations  $d^i = g^i(m^1, m^2, m^3, \dots)$  ( $i = 1, 2, \dots$ ).

Let  $\mathbf{M}$  be the model space manifold, with some coordinates  $m = m^\alpha = m^1, m^2, m^3, \dots$  and with the homogeneous probability density  $\mu_M(\mathbf{m})$ , and let  $\mathbf{D}$  be the data space manifold, with some coordinates  $\mathbf{d} = d^i = d^1, d^2, d^3, \dots$  and with the homogeneous probability density  $\mu_D(\mathbf{d})$ . Let  $\mathbf{X}$  be the joint manifold built as in the cartesian product of the two manifolds,  $\mathbf{D} * \mathbf{M}$ , with coordinates  $\mathbf{x} = \mathbf{d}, \mathbf{m} = d^1, d^2, \dots, m^1, m^2, \dots$  and with the homogeneous probability density that, by definition, is  $\mu(\mathbf{x}) = \mu(\mathbf{d}, \mathbf{m}) = \mu_D(\mathbf{d})\mu_M(\mathbf{m})$ .

From now on we have used the notation  $\Theta(\mathbf{d}, \mathbf{m})$  for the joint probability density describing the correlations that correspond to our physical theory, together with the inherent uncertainties of the theory (due to an imperfect parameterization or to some more fundamental lack of knowledge).

## 2.5 Measurements and A Priori Information

### 2.5.1 Results of the Measurements

All physical measurements are subjected to uncertainties. Therefore, the results of a measurement act is not simply an 'observed value' (or a set of observed values') but a 'state of information' acquired on some observable parameter. If  $\mathbf{d}$  represents the set of observable parameters, the result of the measurement act can be represented by a probability density  $\rho_D(\mathbf{d})$  defined over the data space  $D$ .

### 2.5.2 A Priori Information on Model Parameters

By a priori information (or prior information) we shall mean information that is obtained independently of the results of measurements. The probability density representing this a priori information will be denoted by  $\rho_M(\mathbf{m})$

## 2.6 Defining a solution of the Inverse Problem

### Joint Prior Information

The a priori information on model parameters is independent of observations. The information that we have in both model parameters and observable parameters can then be described in the Manifold  $\mathbf{D} * \mathbf{M}$  by the joint probability density

$$\rho(\mathbf{d}, \mathbf{m}) = \rho_D(\mathbf{d}) * \rho_M(\mathbf{m})$$

### 2.6.1 Combination of Experimental, A Priori, and theoretical Information

Previously, we have defined that the priori probability density  $\rho(\mathbf{d}, \mathbf{m})$ , defined in the space  $D * M$ , represents both information obtained on the observable parameters (data)  $\mathbf{d}$  and a priori information on the model parameters  $\mathbf{m}$ . We have also seen that the theoretical probability density  $\Theta(\mathbf{d}, \mathbf{m})$  represents the information on the physical correlations between  $\mathbf{d}$ , and  $\mathbf{m}$ , as obtained from a physical law, for instance.

These two states of information combine to produce the a posteriori state of information. The method of the previous sections to introduce the a priori and the method of the previous sections to introduce the a priori and the theoretical states of information is such that the a posteriori state of information is given by the conjunction of these two states of information. we can represent the a posteriori information is then

$$\sigma(\mathbf{d}, \mathbf{m}) = k * \frac{\rho(\mathbf{d}, \mathbf{m}) * \Theta(\mathbf{d}, \mathbf{m})}{\mu(\mathbf{d}, \mathbf{m})}$$

where  $\mu(\mathbf{d}, \mathbf{m})$  represents the homogeneous state of information and where  $k$  is a normalization constant.

## 2.7 Using the Solution of the Inverse Problem

### 2.7.1 Computing Probabilities

The solution of the inverse problem is the posterior probability distribution over the model space manifold, represented by the probability density  $\sigma_M(\mathbf{m})$ .

We solve the problem in Monte Carlo way, so the solution to the problem is large set of possible solutions, samples of the posterior probability density in the model space,  $\sigma_M(\mathbf{m})$ . We plot histograms to represent the probability density of what the final density can be.

From now on we have used the notation  $\Theta(\mathbf{d}, \mathbf{m})$  for the joint probability density describing the correlations that correspond to our physical theory, together with the inherent uncertainties of the theory (due to an imperfect parameterization or to some more fundamental lack of knowledge).

In many situations, we have written a joint probability density as the product of a conditional and a marginal. Taking for the marginal for the model parameters the homogeneous probability density then gives

$$\Theta(\mathbf{d}, \mathbf{m}) = \theta(\mathbf{d}|\mathbf{m}) * \mu_M(\mathbf{m})$$

Examples: refer to page 253-266 of [1]

Define the model space, data space, prior, posterior in the following problems:

**Estimation of the Epicentral Coordinates of a Seismic Event**

A seismic source was activated at time  $T = 0$  in an unknown location at the surface of Earth. The seismic waves produced by the explosion have been recorded at a network of six seismic stations whose coordinates in a rectangular system are

$$\begin{aligned} (x^1, y^1) &= (3km, 15km), (x^2, y^2) = (3km, 16km), \\ (x^3, y^3) &= (4km, 15km), (x^4, y^4) = (4km, 16km) \\ (x^5, y^5) &= (5km, 15km), (x^6, y^6) = (5km, 16km) \end{aligned}$$

The observed arrival times of the seismic waves at these stations are

$$\begin{aligned} t_{obs}^1 &= 3.12s \pm \sigma, t_{obs}^2 = 3.26s \pm \sigma, \\ t_{obs}^3 &= 2.98s \pm \sigma, t_{obs}^4 = 3.12s \pm \sigma, \\ t_{obs}^5 &= 2.84s \pm \sigma, t_{obs}^6 = 2.98s \pm \sigma, \end{aligned}$$

where  $\sigma = 0.10s$ , the symbol  $\pm\sigma$  being a short notation indicating that experimental uncertainties are independent and can be modeled using a Gaussian probability density with a standard deviation equals to  $\sigma$ .

Solution:

The model parameters are the coordinates of the epicenter of the explosion,

$$\mathbf{m} = (X, Y)$$

and the data parameters are the arrival times at the seismic network,

$$\mathbf{d} = (t^1, t^2, t^3, t^4, t^5, t^6)$$

while the coordinates of the seismic stations and the velocity of the seismic waves are assumed perfectly known (i.e., known with uncertainties that are negligible with respect to the uncertainties in the observed arrival times).

For a given  $(X, Y)$ , the arrival times of the seismic wave at the seismic stations can be computed using the (exact) equation

$$t^i = g^i(X, Y) = \frac{\sqrt{(x^i - X)^2 + (y^i - Y)^2}}{v} \text{ where } (i = 1, 2, 3, 4, 5, 6)$$

which solves the forward problem  $\mathbf{d} = \mathbf{g}(\mathbf{m})$

As we are not given any a priori information on the epicentral coordinates, we take a uniform a priori probability density, i.e., because we are using Cartesian coordinates,

$$\rho_M(X, Y) = \text{const.},$$

assigning equal a priori probabilities to equal volumes.

As data uncertainties are Gaussian and independent, the probability density representing the information we have on the true values of the arrival times is

$$\rho_D(t^1, t^2, t^3, t^4, t^5, t^6) = \text{const.} \exp\left(-\frac{1}{2} \sum_{i=1}^6 \frac{(t^i - t_{obs}^i)^2}{\sigma^2}\right)$$

with the three pieces of information, we can directly pass to the resolution of the inverse problem. The posterior probability density in the model space combining the three pieces of information, is  $\sigma_M(\mathbf{m}) = k \rho_M(\mathbf{m}) \rho_D(\mathbf{g}(\mathbf{m}))$  i.e., particularization the notation to the present problem.

$$\sigma_M(X, Y) = k \rho_M(X, Y) * \rho_D(\mathbf{g}(\mathbf{X}, \mathbf{Y})),$$

where k is a normalization constant, Explicitly,

$$\sigma_M(m) = k' \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^6 \frac{(t_{cal}^i(X, Y) - t_{obs}^i)^2}{\sigma^2}\right)$$

where k is a new normalization constant and

$$t_{cal}^i(X, Y) = \frac{\sqrt{(x^i - X)^2 + (y^i - Y)^2}}{v}$$

The probability density  $\sigma_M(X, Y)$  describes all the a posterior information we have on the epicentral coordinates. As we only have two parameters, the simplest (and most general) way of studying this information is to plot the values of  $\sigma_M(X, Y)$  directly in the region of the plane where it takes significant values.

**Measuring the Acceleration of gravity** An absolute gravimeter uses the free fall of a mass in vacuo to measure the value of the acceleration g due to gravity. A mass is sent upward with some initial velocity  $v_0$ , and the positions  $z^1, z^2, \dots$  of the mass are (very precisely) measured at different instants  $t^1, t^2, \dots$  In vacuo (orienting the z axis upward),

$$z(t) = v_0 t - \frac{1}{2} g t^2$$

The measured values of the  $z_i$  and the  $t_i$  can be used to infer the values of  $v_0$  and g. The measurements made during a free-fall experiment have provided the values

$$t_1 = 0.20s \pm 0.01s, z_1 = 0.62m \pm 0.02m,$$

$$t_2 = 0.40s \pm 0.01s, z_2 = 0.88m \pm 0.02m,$$

$$t_3 = 0.60s \pm 0.01s, z_3 = 0.70m \pm 0.02m,$$

$$t_4 = 0.80s \pm 0.01s, z_4 = 0.15m \pm 0.02m,$$

where uncertainties in the times are of the boxcar type and the uncertainties in the positions are of the double exponential type (the mean deviations having the value 0.02m). Assuming that there is no particular a priori information on these two values.

Solution: Let us introduce the vector  $\mathbf{m}$  with components

$$\mathbf{m} = v_0, g, t_1, t_2, t_3, t_4$$

and the vector  $\mathbf{d}$  with components

$$\mathbf{d} = z_1, z_2, z_3, z_4$$

we can write the theoretical relations

$$z_1 = v_0 t_1 - \frac{1}{2} g t_1^2,$$

$$z_2 = v_0 t_2 - \frac{1}{2} g t_2^2,$$

$$z_3 = v_0 t_3 - \frac{1}{2} g t_3^2,$$

$$z_4 = v_0 t_4 - \frac{1}{2} g t_4^2$$

that correspond to the usual relation  $\mathbf{d} = \mathbf{g}(\mathbf{m})$ . Although we may call  $\mathbf{m}$  the model parameters and  $\mathbf{d}$  the data parameters, we see that there are observed quantities both in  $\mathbf{d}$  and in  $\mathbf{m}$

The prior information we have on  $\mathbf{m}$  is to be represented by a probability density

$$\rho_M(m) = \rho_M(v_0, g)$$

As we do not wish to include any special a priori information on the parameters  $v_0, g$ , we shall take a probability density that is constant on these parameters. The prior information we have on the other four parameters,  $t_1, t_2, t_3, t_4$

Prior distribution on model space

$$\rho_M(v_0, g) = \frac{1}{\sqrt{2\pi\sigma_v^2\sigma_g^2}} \exp\left(-\left(\frac{v_0 - m_v}{\sigma_v}\right)^2 + \left(\frac{g_0 - m_g}{\sigma_g}\right)^2\right)$$

Likelihood will be

$$\rho(z_1^{obs}, \dots, z_n^{obs} | v_0, g) = \prod_{i=1}^4 \frac{1}{\sqrt{2\pi\gamma^2}} \exp\left(-\frac{1}{2} \frac{(z_i - z_i^{obs})^2}{\gamma^2}\right)$$

where

$$z_i^{obs} = v_0 t_i - \frac{1}{2} g t_i^2 + \eta$$

where  $\eta$  is the random variable for the noise in data. and

$$z_i = v_0 t_i - \frac{1}{2} g t_i^2$$

Posterior is the conditional probability density of model space given data. It will be defined as the product of prior and likelihood.

$$Posterior = \sigma_m(v_0, g \mid z_1, z_2, z_3, z_4)$$

i.e.,

$$S(v_0, g \mid z_1, z_2, z_3, z_4) = \sum_{i=1}^4 \frac{|(v_0 t_i - \frac{1}{2} g t_i^2) - z_i^{obs}|}{\sigma_i}$$

where  $S = -2\ln(\sigma_m)$  and  $|(v_0 t_i - \frac{1}{2} g t_i^2) - z_i^{obs}|$  stands for the norm of the function

**Linear Regression with rounding errors** A physical quantity  $d$  is related to the physical quantity  $x$  through the equation

$$d = m^1 + m^2 x,$$

where  $m^1, m^2$  are unknown parameters. The above equation represents the equation of a line in a plane ( $d, x$ ). In order to estimate  $m^1$  and  $m^2$ , the parameter  $d$  has been experimentally measured for some selected values of  $x$ , and the following results have been obtained

$$x^1 = 03.500, d_{obs}^1 = 2.0 \pm 0.5$$

$$x^2 = 05.000, d_{obs}^2 = 2.0 \pm 0.5$$

$$x^3 = 07.000, d_{obs}^3 = 3.0 \pm 0.5$$

$$x^4 = 07.500, d_{obs}^4 = 3.0 \pm 0.5$$

$$x^5 = 10.000, d_{obs}^5 = 4.0 \pm 0.5$$

where  $\pm 0.5$  denotes rounding errors.

Solutions:

Lets an arbitrary set  $(d^1, d^2, d^3, d^4, d^5)$  be called a data vector and be denoted by  $\mathbf{d}$  and let an arbitrary set  $(m^1, m^2)$  be called a parameter vector and be denoted by  $\mathbf{m}$ . Let

$$\mathbf{d} = \mathbf{g}(\mathbf{m})$$

denote the relationship.

$$d^i = m^1 + m^2 x^i$$

$i=(1, \dots, 5)$  So now,

$$\rho_M(\mathbf{m}) = \rho_M(m^1, m^2)$$

be the probability density representing the prior information on model parameters.

$$\rho_D(\mathbf{d}) = \rho_D(d^1, d^2, d^3, d^4, d^5)$$

be the probability density describing the experimental uncertainties. As rounding uncertainties are mutually independent,

$$\rho_D(\mathbf{d}) = \rho_D(d^1, d^2, d^3, d^4, d^5) = \rho_D^1(d^1) \rho_D^2(d^2) \rho_D^3(d^3) \rho_D^4(d^4) \rho_D^5(d^5)$$



where  $\rho_D^i(d^i)$  denotes the probability density describing the experimental uncertainty for the observed data  $d^i$ . As the uncertainties are only rounding uncertainties, they can be conveniently modeled using uniform probability density functions:

$$\rho_D^i(d^i) = \text{const. if } d_{obs}^i - 0.5 < d^i < d_{obs}^i + 0.5$$

$$\rho_D(\mathbf{d}) = 0 \text{ otherwise}$$

For prior

$$\rho_M(\mathbf{m}) = \rho_M(m^1, m^2) = \text{const.},$$

then we obtain the posterior probability density in the model space has been defined.

$$\sigma_M(\mathbf{m}) = \text{const.} \rho_M(\mathbf{m}) \rho_D(\mathbf{g}(\mathbf{m}))$$

$$\sigma_M(\mathbf{m}) = \text{const.} \quad \text{if } d_{obs}^i - 0.5 < m^1 + m^2 x^i < d_{obs}^i + 0.5$$

$$\sigma_M(\mathbf{m}) = 0 \quad \text{otherwise}$$

# Chapter 3

## Monte Carlo Method

### 3.1 Overview

In the last chapter we have seen that the most general solution of an inverse problem provides a probability distribution over the model space. It is only when the probability distribution in the model space is very simple (for instance, when it has only one maximum).

For more general probability distributions, one needs to perform an extensive exploration of the model space. This exploration can not be systematic (as the number of required points grows too rapidly with the dimension of the space). Well designed random exploration can solve many complex problems. These random methods were called *Monte Carlo methods*.

One domain where Monte Carlo computations are usual is for the numerical evaluation of integrals in large-dimensional spaces. A Monte Carlo sampling of the function can provide an estimation of the result, together with an estimation of the error. When the model space has a large number of dimensions, representing a probability density is impossible, but we can, at least in principle, do something which is largely equivalent. So we can generate (independent) points that are samples of the probability density, i.e., such that the probability of any of the points being inside any domain equals the probability of the domain, which we call *Sampling* method.

### 3.2 Markov Chain Theory

Consider a stochastic process  $X_n, n = 0, 1, 2, \dots$  that takes on a finite or countable number of possible values. Unless otherwise mentioned, this set of possible values of the process will be denoted by the set of non-negative integers  $(0, 1, 2, \dots)$ . If  $X_n = i$ , then the process is said to be in state  $i$  at time  $n$ . We suppose that whenever the process is in state  $i$ , there is a fixed probability  $P_{ij}$  that it will next be in state  $j$ . That is, we suppose that

$$P(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0) = P_{ij}$$

for all the states  $i_0, i_1, \dots, i_{n-1}, i, j$  and all  $n > 0$ , such a stochastic process is known as a **Markov chain**. The above equation may be interpreted as stating that, for a Markov chain, the conditional distribution of any future state  $X_{n+1}$  given the past states  $X_0, X_1, \dots, X_{n-1}$  and the present state  $X_n$ , is independent of the past states and depends only on the present state.

The value  $P_{ij}$  represents the probability that the process will, when in state  $i$ , next make a transition into state  $j$ . Since probabilities are non-negative and since the process must make a transition into some state, we have that

$$P_{ij} \geq 0, i, j \geq 0; \sum_{j=0}^{\infty} P_{ij} = 1, i = 0, 1, \dots$$

**Example** (A Random Walk Model): A Markov chain whose state space is given by the integers  $i = 0, \pm 1, \pm 2, \pm 3, \dots$  is said to be a random walk if, for some number  $0 < p < 1$ ,

$$P^{i,i+1} = p = 1 - P^{i,i-1}, i = 0, \pm 1, \dots$$

The preceding Markov chain is called a random walk for we may think of it as being a model for an individual walking on a straight line who at each point of time either takes one step to the right with probability  $p$  or one step to the left with probability  $p-1$ .

### 3.3 Markov Chain Monte Carlo Idea

Markov chain Monte Carlo methods are often applied to solve integration and optimization problems in large dimensional spaces. The idea of Monte Carlo is as follows: We want to generate random draws from a target distribution. We then identify a way to construct a 'nice' Markov chain such that its equilibrium probability distribution is our target distribution. If we can construct such a chain then we arbitrarily start from some point and iterate the Markov chain many times. Eventually, the draws we generate would appear as if they are coming from our target distribution. We then approximate the quantities of interest by taking the sample average of the draws after discarding a few initial draws which is the Monte Carlo component.

To construct 'nice' Markov chains we have studied algorithms like Metropolis-Hastings algorithm, Gibbs sampler.

### 3.4 Metropolis Hastings Algorithm

Metropolis-Hastings algorithm is a Markov chain Monte Carlo method for obtaining a sequence of random samples from a probability distribution for which direct sampling is dif-

ficult. This sequence can be used to approximate the distribution (i.e. to generate a histogram). It is Markov chain Monte Carlo (MCMC) method, i.e, it is random (Monte Carlo) and has no memory, in the sense that each step depends only on the previous step (Markov Chain).

The basic idea is to perform a random walk, a sort of Brownian motion, that, if unmodified, would sample some initial probability distribution, then, using a probabilistic rule, to modify the walk (some proposed moves are accepted. some are rejected) in such a way that the modified random walk samples that target distribution. With such a condition, the Metropolis rule is the most efficient to sample the target distribution.

*Aim:* Construct Markov chain for a given high dimensional complex distribution which is termed as 'Target distribution'.

### Algorithm

- Let  $f(x)$  be a function that is propotional to the desired probability distribution  $p(x)$

$$f(x) \propto p(x)$$

where  $f(x)$  be the non-normalized function of the desired probability function.

- **Initialization**

Choose any arbitrary point  $x_0$  (i.e. first sample)

Choose an arbitrary probability density  $Q(x | y)$  where Q stands for proposal density or jumping distribution, and  $Q(x | y)$  is a conditional proposal density which suggests a candidate for the next sample value  $x$ , given the previous sample value  $y$ .

- For each iteration 't'
- – Generate a candidate  $x^*$  for the next sample by picking from the distribution  $Q(x^* | x_t)$
- Calculate the acceptance ratio  $\alpha$  where  $\alpha = \frac{f(x^*)}{f(x_t)}$  which will be used to decide whether to accept or reject the candidate. As  $f \propto p$ , we have

$$\alpha = \frac{f(x^*)}{f(x_t)} = \frac{p(x^*)}{p(x_t)}$$

- If  $\alpha \geq 1$ , then the candidate is more likely than  $x_t$ .  
Automatically accept the candidate by setting  $x_{t+1} = x^*$
- Otherwise  
accept the candidate with probability  $\alpha$
- If candidate is rejected,  
put  $x_{t+1} = x_t$  instead

This algorithm proceeds by randomly attempting to move about the sample space, sometimes accepting the moves and sometimes remaining in place.

- acceptance ratio ' $\alpha$ ' indicates that how probable the new proposed sample is w.r.t the new current sample, according to the distribution  $p(x)$ .

If we attempt to move to a point that is more probable than the existing point i.e.  $\alpha \geq 1$

However, if we attempt to move to a less probable point i.e.  $\alpha < 1$

Sometimes, we reject the move and the more the relative drop in probability the more likely we are to reject the new point.

Thus, we will tend to stay in high-density regions of  $p(x)$

### Steps

- $t = 1$
- generate an initial value  $x_0$  and set  $x^t = x^0$
- update  $t=t+1$   
generate a proposal  $x^*$  from  $q(x^* | x_t)$
- evaluate the acceptance probability

$$\alpha = \min\left(1, \frac{p(x^*)q(x_t | x^*)}{p(x_t)q(x^* | x_t)}\right)$$

- generate  $r$  from a uniform (0,1) distribution
  - if  $r \leq \alpha$  accept the proposal and set  $x^t = x^*$
  - else set  $x^t = x^{t-1}$
  - until  $t=T$

Many basic scientific problems are now routinely solved by simulation. Here we have studied an example drawn from course work of Stanford students.

**Example** (Cryptography). Stanford's statistics department has a drop-in consulting service. One day, a psychologist from the state prison system showed up with a collection of coded messages. The problem was to decode these messages. It was guessed that the code was a simple substitution cipher, each symbol standing for a letter, number, punctuation mark or space. thus, there is an unknown function  $f$

$$f : \text{codespace} \mapsto \text{usualalphabet}.$$

One standard approach to decrypting is to use the statistics of written English to guess at probable choices for  $f$ ,

$$Pl(f) = \prod_i M(f(s_i), f(s_{i+1})),$$

where  $s_i$  runs over consecutive symbols in the coded message. Functions  $f$  which have high values of  $Pl(f)$  are good candidates for decryption. Maximizing  $f$ 's were searched for by running the following Markov chain Monte Carlo algorithm:

1. Start with a preliminary guess, say  $f$ .
2. Compute  $Pl(f)$
3. Change to  $f^*$  by making a random transposition of the values  $f$  assigns to two symbols.
4. Compute  $Pl(f^*)$ ; if this is larger than  $Pl(f)$ , accept  $f^*$ .
5. If the coin toss comes up tails, stay at  $f$ .

The algorithm continues, trying to improve the current  $f$  by making random transpositions. The coin tosses allow it to go to less plausible  $f$ 's, and keep it from getting stuck in local maxima. So when the Monte Carlo algorithm was run. After 100 steps, the message was a mess. After 2000 steps, the decrypted message made sense. It stayed essentially the same as further steps are tried. It is remarkable that a few thousand of this simple optimization procedure work so well.

After that we have generated some Markov chains for different different distributions over same proposal density i.e. Gaussian distribution. The algorithms of the examples followed by the results (plotted histograms) are following:

### Example 1:

**Target density:**  $p_{normal} = (x, variance) = (\frac{x^2}{variance})$

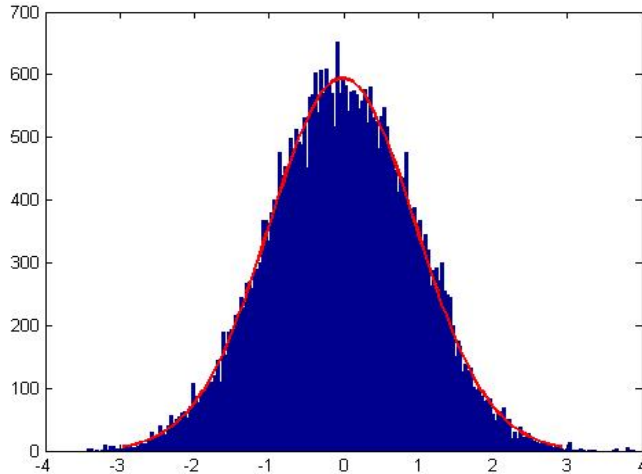
**Proposal density:**  $q(x_1, x_2, variance) = (\frac{(x_1 - x_2)^2}{variance})$

**Algorithm:**

- $t = 1$
- generate an initial value  $x_0$  and set  $x^t = x_0$
- update  $t=t+1$   
generate a proposal  $x^*$  from  $q(x^* | x_t)$
- evaluate the acceptance probability

$$\alpha = \exp * (-\frac{1}{2}) * (p_{normal}(x^*) + q(x_t | x^*) - p_{normal}(x_t) - q(x^* | x_t))$$

- generate  $r$  from a uniform  $(0,1)$  distribution
  - if  $r \leq \alpha$  accept the proposal and set  $x^t = x^*$
  - else set  $x^t = x^{t-1}$
  - until  $t=T$



Figure“3.1: histogram of  $mc_1$

**Example 2:**

**Target density:**  $p_\theta(\theta) = \frac{1}{\pi(1+\theta^2)}$

**Proposal density:**  $q(x_1, x_2, variance) = \left(\frac{(x_1-x_2)^2}{variance}\right)$

**Algorithm:**

- $t = 1$
- generate an initial value  $x_0$  and set  $x^t = x^0$
- update  $t=t+1$ 
  - generate a proposal  $x^*$  from  $q(x^* | x_t)$
- evaluate the acceptance probability
 
$$\alpha = \left(\exp * \left(-\frac{1}{2}\right) * (q(x_t | x^*) - q(x^* | x_t))\right) * \left(\frac{p_\theta(x^*)}{p_\theta(x_t)}\right)$$
- generate  $r$  from a uniform  $(0,1)$  distribution
  - if  $r \leq \alpha$  accept the proposal and set  $x^t = x^*$
  - else set  $x^t = x^{t-1}$
  - until  $t=T$

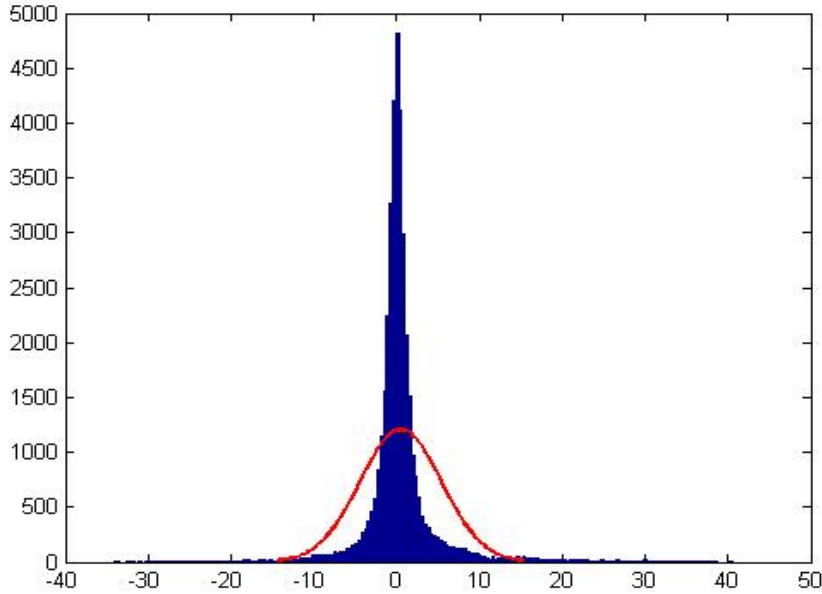


Figure 3.2: histogram of  $mc_new$

**Example 3:**

**Target density:**  $posterior = \sum_{i=1}^4 (v_0 t_i - \frac{1}{2} g t^2 - z_1^{obs})^2 / 0.0004 + (\frac{g - g_{mean}}{g_{variance}})^2$

**Proposal density:**  $q(x_1, x_2, variance) = (\frac{(x_1 - x_2)^2}{variance})$

**Data given:**

$$t_1 = 0.20; t_2 = 0.40; t_3 = 0.60; t_4 = 0.80$$

$$z_1^{obs} = 0.62; z_2^{obs} = 0.88; z_3^{obs} = 0.70; z_4^{obs} = 0.15$$

$$v_0 = 4.12$$

$$variance = 0.01 \quad g_{variance} = 0.01 \quad g_{mean} = 9.8$$

**Algorithm:**

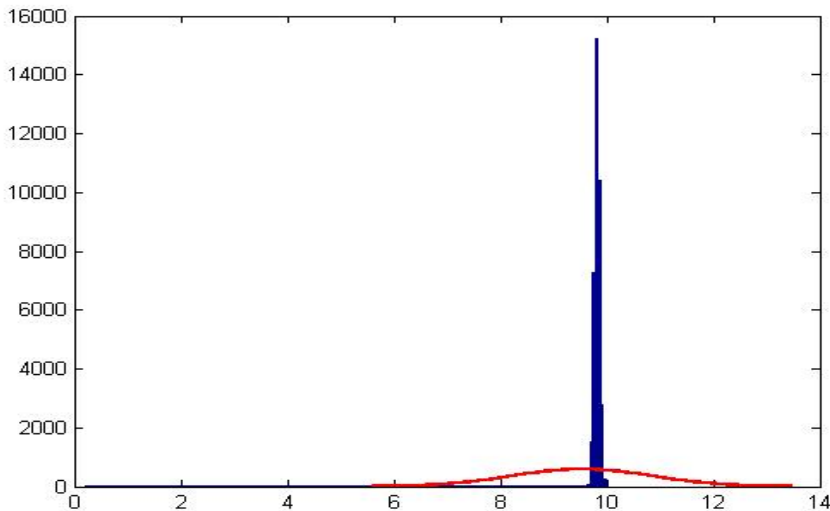
- $t = 1$
- generate an initial value  $g_0$  and set  $g^t = g^0$
- update  $t=t+1$   
generate a proposal  $g^*$  from  $q(g^* | g_t)$



- evaluate the acceptance probability

$$\alpha = \exp\left(-\frac{1}{2}\right) * (\text{posterior}(g^*) + q(g_t | g^*) - \text{posterior}(g_t) - q(g^* | g_t))$$

- generate r from a uniform (0,1) distribution
  - if  $r \leq \alpha$  accept the proposal and set  $g^t = g^*$
  - else set  $g^t = g^{t-1}$
  - until  $t=T$



Figure“3.3: histogram of  $g_{acceleration}$

The mean of the posterior is  $9.8m/s^2$  as we can analyze from the histogram.

#### Example 4:

##### Target density:

$$\text{posterior}_{bivar} = \sum_{i=1}^4 (v_0 t_i - \frac{1}{2} g t^2 - z_1^{obs})^2 / 0.0004 + \left( \frac{(g - g_{mean})^2}{g_{variance}} \right) + \left( \frac{(v - v_{mean})^2}{v_{variance}} \right)$$

##### Proposal density:

$$\text{proposal}_{bivar} = (x - \mu)' * \Sigma^{-1} * (x - \mu)$$

where x and  $\mu$  is a  $2 * 1$  matrix and  $\Sigma^{-1}$  is inverse of covariance matrix

##### Data given:

$$t1 = 0.20; t2 = 0.40; t3 = 0.60; t4 = 0.80$$

$$z_1^{obs} = 0.62; z_2^{obs} = 0.88; z_3^{obs} = 0.70; z_4^{obs} = 0.15$$

$$v_0 = 4.12$$

$$variance = 0.01$$

$$gvariance = 0.01$$

$$gmean = 9.8$$

**Algorithm:**

- $t = 1$

- generate an initial value  $x_0$  and set  $x^t = x^0$

- update  $t=t+1$

generate a proposal  $x^*$  from  $proposal_{bivar}(x^* | x_t)$

- evaluate the acceptance probability

$$\alpha = \exp\left(-\frac{1}{2}\right) * (posterior_{bivar}(x^*) + proposal_{bivar}(x_t | x^*) - posterior_{bivar}(x_t) - proposal_{bivar}(x^* | x_t))$$

- generate  $r$  from a uniform (0,1) distribution

– if  $r \leq \alpha$  accept the proposal and set  $x^t = x^*$

– else set  $x^t = x^{t-1}$

– until  $t=T$

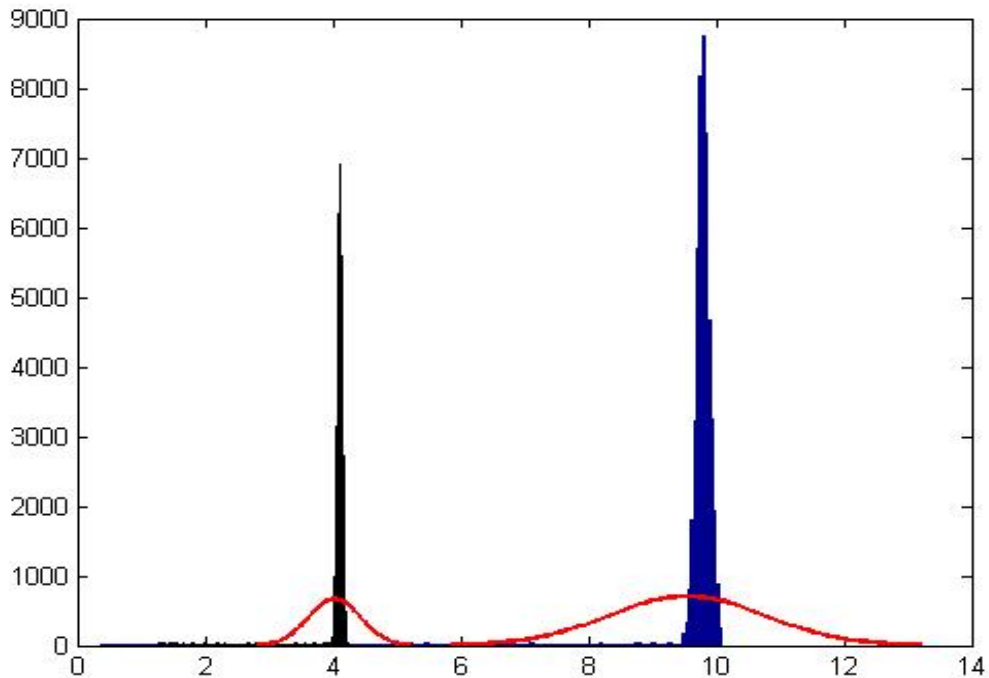


Figure 3.4: histogram of  $code_{bivar}$

Hence the mean of the bivariate posterior for  $g$  is  $9.8m/s^2$  and that is for the  $v$  is approximately  $4.12m/s$

Now, to analyze the histograms that we have got from MCMC we need to confirm about convergence of Markov Chain. Because we may know a lot about the particular Markov Chain being used, but if whatever we know is of no help in determining any convergence information about the Markov Chain, then whatever information we have about the Markov Chain is not reliable, it's useless.

## 3.5 The Practice of MCMC

The practice of MCMC is simple. Set up of a Markov chain having the required invariant distribution, and run it on a computer. the folklore of simulation makes this seem more complicated than it really is. None of this folklore is justified by theory and none of it actually helps user do good simulations, but, like other kinds of folklore, it persists despite its lack of validity.

### 3.5.1 Black Box MCMC

There is great deal of theory about convergence of Markov chains. Unfortunately, none of it can be applied to get useful convergence information for most MCMC applications. Thus most of the time we happen to be in the following situations called *black box MCMC*:

- We have a Markov chain having the required invariant distribution.

- We know nothing other than that. the Markov chain is a “Black Box” that we cannot see inside. When run, it produces output. That is all we know. We know nothing about the transition probabilities of the Markov chain, nor anything else about its dynamics.
- We know nothing about the invariant distribution except what we may learn from running the Markov chain.

### 3.5.2 Pseudo Convergence

Sometimes a Markov Chain can appear to have converged to its equilibrium distribution when it has not. This happens when parts of the state space are poorly connected by the Markov Chain dynamics. It takes many iterations to get from one part to another. When the time it takes to transition between these parts is much longer than the length of simulated Markov Chain, then the Markov Chain can appear to have converged but the distribution it appears to have converged to is the equilibrium distribution conditioned on the part in which the chain was started. We call this phenomenon *Pseudo Convergence*.

Hence to confirm whether the convergence that we have got isn't *pseudo convergence* we checked whether the multiple runs appear to converge to the same distribution, and all of the codes did converge to the same distribution and hence we confirmed that the Markov Chain is converging. But still this method is not convincing enough to prove that it's not pseudo convergence because in some complicated MCMC codes this has been observed that the samplers are appeared to be converged after weeks of running, they discovered a new part of state space and the distribution changed radically. So, to check that we made an overnight run. This may detect a pseudo-convergence.

### 3.5.3 Burn-In

To make it more precise we have put a burn in period as well in code. Burn-in is a colloquial term that describes the practice of throwing away some iterations at the beginning of an MCMC run. So if we start somewhere, say at  $x$ , then we run the Markov Chain for  $n$  steps during which we throw away all the data.

### 3.5.4 Diagnostics

If we see look upto diagnostics, there have been many MCMC diagnostics that are there in the literature. Some work with one run of a Markov chain, while others with multiple runs of a Markov chain started at different points, what we call multistart heuristic.

There is only one perfect MCMC diagnostic: perfect sampling. This is basically a method of Markov-chain-assisted i.i.d. sampling. Since it produces an i.i.d. sample from the equilibrium distribution of the Markov chain, we will get a sufficiently large sample to not miss any parts of the state space. Perfect sampling does not work on black box MCMC, because it requires complicated theoretical conditions on the Markov chain dynamics. If we

know nothing about the Markov chain dynamics or equilibrium distribution except what we learn from output of the sampler, we can always be fooled by pseudo-convergence.

So, we have learnt that the diagnostics can find the known unknowns. They can not find the unknown unknowns. They can not find out what a black box MCMC sampler will do eventually. Only sufficiently long runs can do that.

Hence the histograms we have got are the histograms that we got after running the MCMC code for a sufficiently long time i.e. in this case overnight.

### 3.6 Kalman-Filter Equations

Now, for the bivariate code i.e. finding acceleration of gravity and the initial velocity of a mass sent upward. We have tried finding its analytical solution as well so that we can compare the difference between analytical solution and the solution that we have got from MCMC method.

To find the analytical solution of the problem we have used Kalman-Filter equations.

Let

$$Prior : N(m_0, C_0)$$

$$Measurementmap : d = Gm + \eta$$

where  $\eta$  is the noise in data space.

$$Likelihood : p(d | m) \propto \exp\left(-\frac{1}{2}(d - Gm)'R^{-1}(d - Gm)\right)$$

As we know that the Posterior is the product of Prior and Likelihood and also normally distributed. So, the expected posterior would be like

$$m \sim N(m_1, C_1)$$

So, by Kalman-Filter equations

$$m_1 = m_0 + K(d^{obs} - Gm_0)$$

$$C_1 = (I - KG)C_0$$

or

$$C_1^{-1} = C_0^{-1} + (G'RG)^{-1}$$

where,

$$K = C_0G'(GC_0G' + R)^{-1}$$

Using the above Kalman-Filter equations we have calculated the mean of posterior and those are the following:

$$g - \textit{Posterior} - \textit{Mean} = 9.5788$$

$$v - \textit{Posterior} - \textit{mean} = 4.0749$$

when we used the data that was given in the problem and the variance of the prior i.e. variance of  $g$  as well as variance of  $v$  is taken to be 0.00004 units.

And the mean we got from the MCMC code is

$$g - \textit{Posterior} - \textit{Mean} = 9.4824$$

$$v - \textit{Posterior} - \textit{Mean} = 4.0059$$

# Chapter 4

## Parameter Estimation of a Population Dynamics Model

### 4.1 Overview

*Drosophila* population in laboratory exhibits a variety of dynamics over changing the stage-specific food regime, which underlie the intensity of different density dependent feedback.

We will consider a model for the temporal dynamics of population size of *Drosophila Melanogaster* with multiple life stages (egg, larva, pupa, and adult stages in insects) which are influenced by the nonlinear density- dependent feedback mechanisms operating at different stages and the interactions of these mechanisms. The model involves seven independent and interacting parameters. So, Here we will try to devise Markov Chain Monte Carlo based methods in order to estimate parameters of this model, based on the laboratory data available.

### 4.2 Introduction

There are two active (feed and move) life stages - Larvae and Adult- of *Drosophila* life cycle in the laboratory. The crowding at these two stages induces major effect on their survival and hence the final adult population abundance, by regulating different life history factors of organisms, like growth rate, body weight, fecundity or development time etc.

In insect species, the multiple life cycle stages experience different types of developmental and environmental processes to regulate their life history traits (e.g. body size, fecundity etc) and numbers. The regulation is highly nonlinear and interacts across different stages. On the basis of a detailed literature survey on *Drosophila* life cycle traits and its population dynamics, along with the analysis of data provided by laboratory experiments, it was understood that there are three major density-dependent mechanisms that govern the population dynamics of *Drosophila*. they are- (1) Larval crowding on pre-adult survivorship; (2) larval crowding on female fecundity; and (3) Effect of adult-density on female fecundity. These density-dependence mechanisms interact in a complex manner and at different combinations

in different larval and adult food regimes.

Crowding is introduced in experimental populations in the laboratory by manipulating the amount and quality of food given at these two life stages. There are four combinations of food regimes LH, HL, HH, LL, which are maintained in laboratory, where L- stands for low and H for high food level- with the first position for Larvae and the second position for Adult. These four food regimes exhibit different dynamics of *Drosophila* population in laboratory due to differential effect on the feedback processes.

So, LH- stands for Low food for Larvae and High food for Adult

HL- stands for High food for Larvae and Low food for Adult

LL- Low food for both Larvae and Adult

HH- High food for both Larvae and Adult

### 4.3 The Model Development

The model is described below along with the four functional forms used to frame the two variable model with adult number and mean weight over generation as variables. the model consists of several parameters that quantify the effects of the density-dependent feedback.

#### The functional forms of density dependence

Table1

$$1. N_{t+1} = Cn_{t+1} \exp(-cn_{t+1})$$

$$2. f_t = f_{max} \exp(-bn_t)$$

$$3. n_{t+1} = \frac{1}{2} f_t \frac{1}{(1 + dN_t)} N_t$$

$$4. W_t = \exp(A - an_t)$$

#### 4.3.1 The full two-variable model for *Drosophila* population dynamics

$$N_{t+1} = \left( \frac{Cf_{max}N_t}{2(1 + dN_t)} \right) * \exp \left[ \overset{table2}{-\frac{b(A - \log W_t)}{a} - \left( \frac{cf_{max}N_t}{2(1 + dN_t)} * \exp [-b(A - \log W_t)/a] \right)} \right]$$

$$W_{t+1} = \exp \left( A - \left\{ \frac{af_{max}N_t}{2(1 + dN_t)} * \exp [-b(A - \log W_t)/a] \right\} \right)$$

Where C, c,  $f_{max}$ , b, d, A, a are different life-history parameters of the organism.



## 4.4 Model Explanation

To arrive at a model, which may be adequate to describe these density-dependent processes acting at different stages in *Drosophila*, we elaborate the possible functional relationships between the different life stages in Table (1). As is clear from the four relationships, the adult number ( $N_t$ ) and egg number ( $n_t$ ), fecundity( $f$ ), and weight ( $W_t$ ) at time  $t$  interact in a complex manner and at different combinations. We coupled these four equations and arrived at a two variable model that involves both the adult size ( $N_t$ ) and adult weight ( $W_t$ ) as given in table (2). This is the first time that two life history parameters have been coupled to give both quantitative and qualitative assessments of their relationship in deciding the population dynamics.

The model contains seven life-history parameters indicating the intensity of different density dependent feedback. The problem is that a single parameter may embody the effect of more than one density dependent feedback process at different food regime, and their relative importance may be different food regime. The complexity increases with natural intrinsic noise in the experiment. So any standard and straight forward estimation of parameters is not adequate to sort out the problem. hence, The main objective of our study would be to the dynamics observed in different food regimes. Since the data is noisy, it naturally leads to use of a Bayesian approach to this parameter estimation problem. In the Bayesian formulation, the main object of interest is the conditional posterior distribution function on the parameters, conditioned on a given data set.

In this project we will develop and use the Markov Chain Monte Carlo methods, for sampling this posterior distribution function. In contrast with the deterministic parameter estimation methods, the stochastic sampling methods give us quantitative measure for the errors in the estimated parameters. In addition, different data sets would give us information about the parameter regions for the model that would be consistent with the different regimes described above. This will help us understand whether these regimes overlap in the parameter space and will lead to an improved understanding of the experimental observations. thus the probability density on the parameters will provide us both statistical and dynamical information.

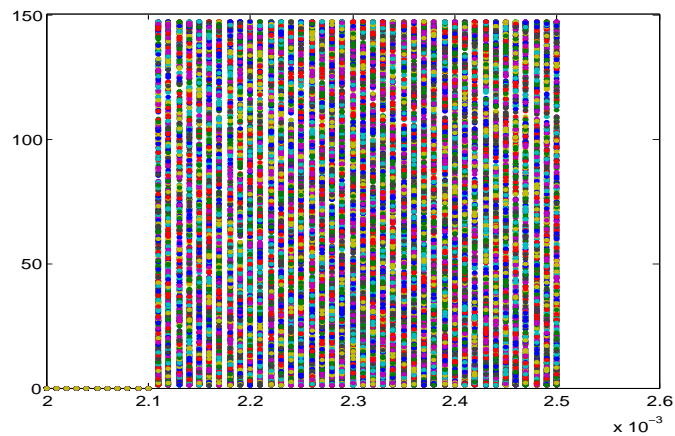
## 4.5 Bifurcation Theory

The main problem is that we do not have parameter values or even estimates and we need to estimate the parameters from the data. On top of that parameter values would also depend on the food regimes-as different food regimes may affect the feedback processes in nonlinear fashion. Hence, we can give biological arguments about how food may affect some of these parameters. So, for that we have plotted bifurcation diagrams of all the parameter values, so that we can have a rough idea about the chaotic behavior of the parameter values according

to the dynamics. Accordingly, from those bifurcation diagrams we can have an estimate of the mean and variance of the parameters, by analyzing the bifurcation diagrams. Below are the bifurcation diagrams:

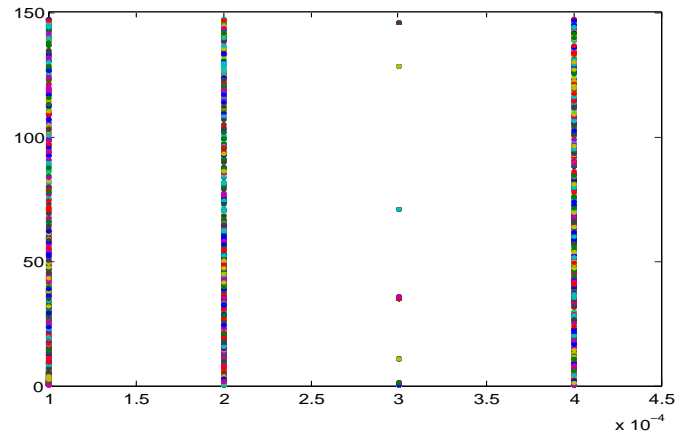
$$N_{t+1} = \left( \frac{Cf_{\max}N_t}{2(1 + dN_t)} \right) * \exp \left[ -\frac{b(A - \log W_t)}{a} - \left( \frac{cf_{\max}N_t}{2(1 + dN_t)} * \exp [-b(A - \log W_t)/a] \right) \right]$$

$$W_{t+1} = \exp \left( A - \left\{ \frac{af_{\max}N_t}{2(1 + dN_t)} * \exp [-b(A - \log W_t)/a] \right\} \right)$$



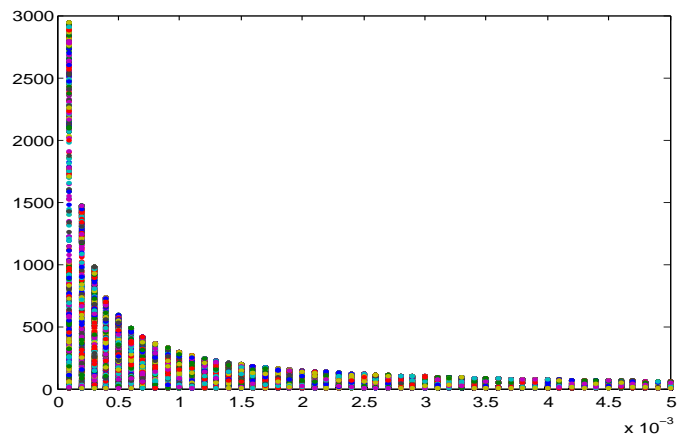
Figure“4.1: Bifurcation diagram of the model v/s the parameter (a)

In the above bifurcation diagram we have varied the parameter  $a$  in the window of 0.002 to 0.003 and we can see the one stable point upto 0.0021 and after that it is showing a complete chaotic behavior.



Figure“4.2: Bifurcation diagram of the model  $v/s$  parameter (b) *bifurcation<sub>b</sub>*

In the above bifurcation diagram we have varied the parameter  $\mathbf{b}$  in the window of 0.0001 to 0.00045. It is showing rather a strange behavior which we can not explain.



Figure“4.3: Bifurcation diagram of the model v/s the parameter (c)

In the above bifurcation diagram we have varied the parameter  $c$  in the window of 0 to 0.005. In the bifurcation diagram we can see that everything is apparently getting converged to 0.

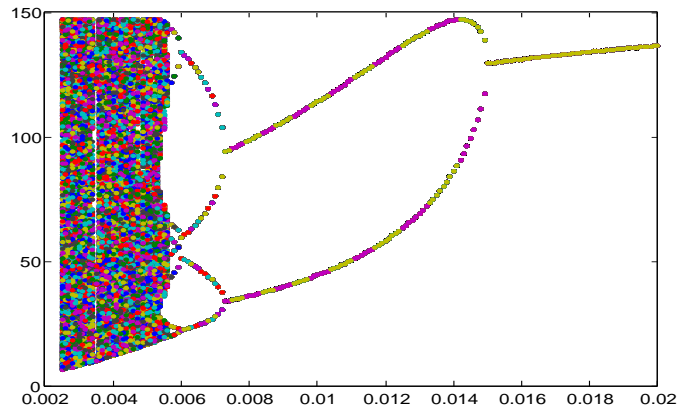
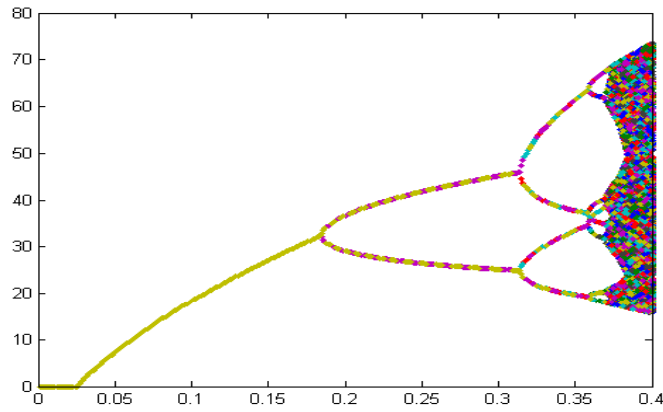


Figure 4.4: Bifurcation diagram of the model w.r.t the parameter ( $d$ )

In bifurcation diagram we have varied the parameter  $d$  in the window of 0.002 to 0.02. In the bifurcation diagram we can see that as we go on the left from 0.02 one fixed point is splitting to give a 2-period cycle at point 0.015. Further moving to the left at point 0.007 is again splitting to 4-periodic cycle and then 8-period cycle for a very short set of points which ultimately ended up showing a chaotic behavior.



Figure“4.5: Bifurcation diagram of the model v/s the parameter (C)

In bifurcation diagram we can see that we have varied the parameter  $C$  in the window of 0 to 0.4. There we can see that one fixed point is at 0.18 is splitting to give a 2-period cycle. Which later on at 0.325 is giving 4-period cycle and after that it is showing a complete chaotic behavior.

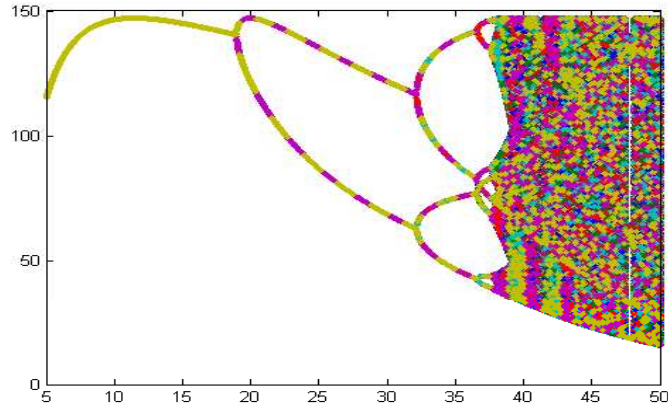


Figure 4.6: Bifurcation diagram of the model v/s the parameter (f)

In bifurcation diagram we have varied the parameter  $f$  in the window of 5 to 50. Here we can clearly see one fixed point is bifurcated to give first 2-period orbit between 15 to 20 leading to bifurcating between 30 to 35 to 4-period cycle. After that it leads to a chaotic behavior.

## 4.6 MCMC Scheme

Now that we have the bifurcation diagrams of all the parameter values. We can have an estimation of the mean value of parameter and the corresponding variance of parameter which we will need to write the MCMC code later. So we have an estimate of the parameter values so we can start of with writing the MCMC code. So, for that, we will define the Model Space, Data Space and the Model equation.

### 4.6.1 Model Space

Model space will consist of all the parameter values that are there in the model and the initial values of the data values which will be given to us.

$$M = a, b, c, d, A, C, f$$

### 4.6.2 Data Space

Data space will consist of all the data that we have from the laboratory experiments.

$$D = N1, N2, N3, \dots, N_k$$

### 4.6.3 Model Function

Model is a function that will take the model space to data space,

$$d = Gm$$

which we can be written as described below

$$N_{t+1} = \left( \frac{Cf_{\max}N_t}{2(1 + dN_t)} \right) * \exp \left[ -\frac{b(A - \log W_t)}{a} - \left( \frac{cf_{\max}N_t}{2(1 + dN_t)} * \exp [-b(A - \log W_t)/a] \right) \right]$$
$$W_{t+1} = \exp \left( A - \left\{ \frac{af_{\max}N_t}{2(1 + dN_t)} * \exp [-b(A - \log W_t)/a] \right\} \right)$$

Now, that we have defined model space, data space, model equation, we can start of with the algorithm for the MCMC code. First of all, we will have to define Prior, Likelihood, and Posterior probability functions to start with and then we can propose a proposal density accordingly.

### 4.6.4 Prior

Prior is the probability distribution function of model parameters. As from the experiments that are done in the laboratory and biological explanation we know that it can only take positive values all the time. So to keep it simple we need to find out such a probability distribution function which rejects all the negative values that are being proposed or that we are getting after the proposal as well. So, for that we have chosen the probability distribution function for the Prior to be a Chi-Squared distribution function. Keeping that in mind the Prior for the model parameters can written as

$$p_{prior}(m) = \prod_{i=1}^7 \left( \frac{1}{N} x_i^{\frac{m_i^2}{\sigma_i^2} - 1} \exp \frac{-x_i m_i}{\sigma_i^2} \right)$$



Where  $x$  is a  $7 \times 1$  matrix consists of all the parameters  
 $N$  is the normalization constant  
 $\sigma$  is the standards deviation of the parameters  
 $m$  is the mean of the parameter values

### 4.6.5 Likelihood

Likelihood is a conditional probability distribution function of data space given model space. We have chosen that to be the Gaussian conditional probability distribution function, which can be written as follows:

$$p_{likelihood}(d | m) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\gamma^2}} \exp\left(-\frac{1}{2} \frac{(N(i) - N^{obs}(i))^2}{\gamma^2}\right)$$

Where  $\gamma$  is the noise in experimental data

$N^{obs}(i)$  stands for all the observational values of  $N$

$N(i)$  stands for all the expected values of data that we have calculated with the given model equation.

### 4.6.6 Posterior

Posterior is a conditional probability distribution function of model space given data space. That, according to Baye's theorem, we can calculate by multiplying Prior and the Likelihood functions as follows:

$$P_{posterior}(m | d) = p_{prior}(m) * p_{likelihood}(d | m)$$

### 4.6.7 Proposal

Now, that we have probability distribution functions for prior, likelihood and posterior, we can propose a proposal density that we are gonna use in our MCMC code. We can simply take the proposal to be a multivariate Gaussian probability distribution function. That can be written as follows:

$$proposal_{multivariate} = (x - \mu)' * \Sigma^{-1} * (x - \mu)$$

where  $x$  and  $\mu$  is a  $7 \times 1$  matrix and  $\Sigma^{-1}$  is inverse of covariance matrix  
and  $\Sigma$  is a diagonal matrix of the covariances of the parameters

## 4.7 MCMC Algorithm

Now we may proceed to write an MCMC algorithm for the parameter estimation problem of the given population dynamics model:

**Algorithm:**

- $t = 1$
- generate an initial value  $x_0$  and set  $x^t = x^0$
- update  $t=t+1$   
generate a proposal  $x^*$  from  $proposal_{pop}(x^* | x_t)$
- evaluate the acceptance probability

$$\alpha = \frac{(posterior_{pop}(x^*) * proposal_{pop}(x_t | x^*))}{posterior_{pop}(x_t) * proposal_{pop}(x^* | x_t)}$$

- generate  $r$  from a uniform (0,1) distribution
  - if  $r \leq \alpha$  accept the proposal and set  $x^t = x^*$
  - else set  $x^t = x^{t-1}$
  - until  $t=T$

Now, using a certain set of all the seven parameters the data is generated which consists of number of adults of the population of *Drosophila* for about 1000 generations.

So, the model parameters that we have used to generate the data is as follows:

$$[p1, p2, p3, p4, p5, p6, p7] = [0.003, 0.001, 0.0019, 0.003, 0.000305, 0.79829, 32.429]$$

Now, by trying out the different sets of the standard deviations for the prior probability distribution, likelihood distribution and the proposal distribution we try to get a convergence that will give the best suited distribution for the parameters. So, after trying a lot of different sets of standard deviations below are the combination which gave the best suited distribution for most of the parameters.

*StandardDeviationofpriordensity*

$$p1_{SD} = 0.7$$

$$p2_{SD} = 0.3$$

$$p3_{SD} = 0.111$$

$$p4_{SD} = 0.09$$

$$p5_{SD} = 0.1$$

$$p6_{SD} = 0.79829$$

$$p7_{SD} = 8.0429$$

*Standard deviation of Proposal Density*

$$p1_{proposal}SD = 0.12124$$

$$p2_{proposal}SD = 0.16431$$

$$p3_{proposal}SD = 0.10819$$

$$p4_{proposal}SD = 0.08538$$

$$p5_{proposal}SD = 0.1$$

$$p6_{proposal}SD = 0.01383$$

$$p7_{proposal}SD = 0.01393$$

$$\text{Variance of Likelihood} = 7$$

*Initial Conditions*

$$N_{initial} = 200$$

$$W_{initial} = 0.459$$

The “Burn-in” steps that have been used in the MCMC code is 50,000.

The length of the Markov Chain is 8,000.

Hence, Using all the above sets and the initial conditions below are the distribution that we got for each of the parameter:

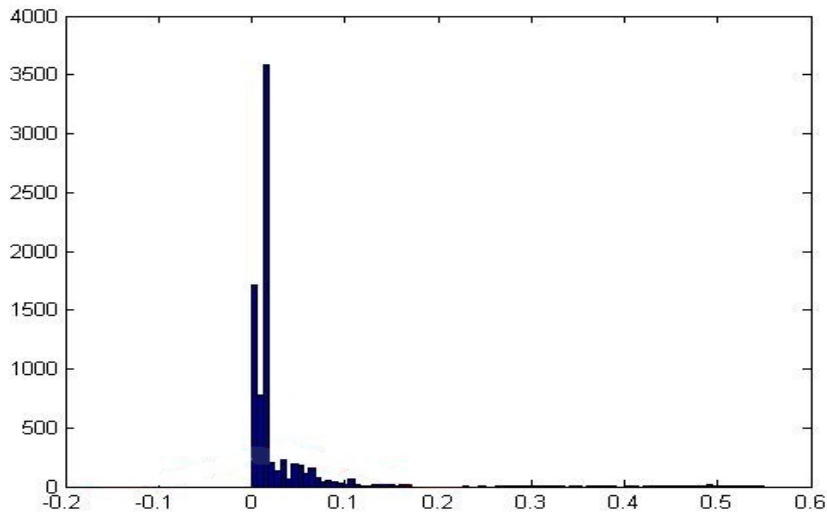
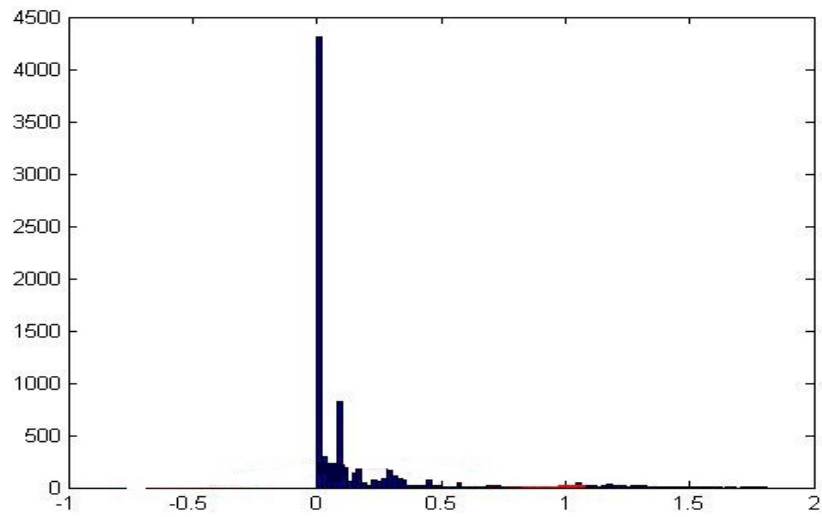
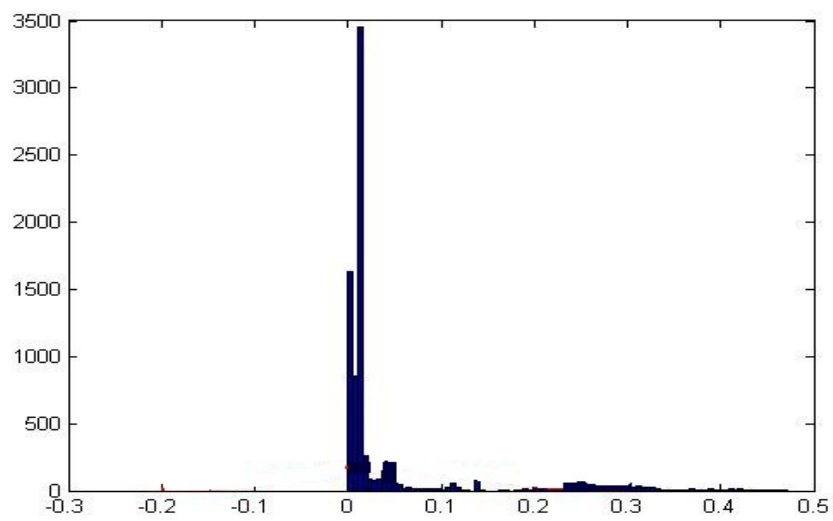


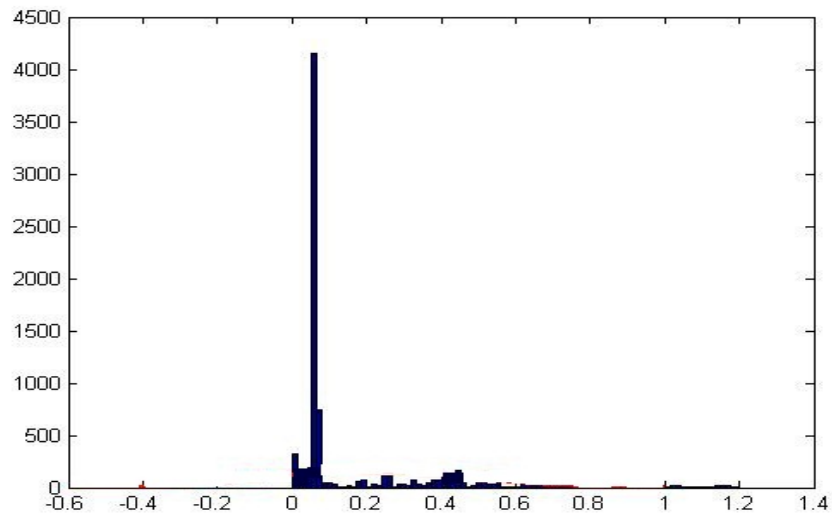
Figure “4.7: histogram of  $parameter1(p1)$



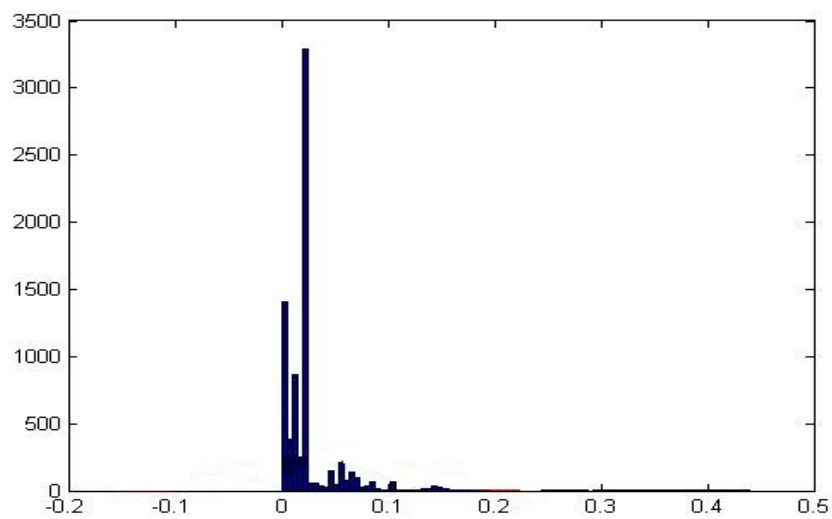
Figure“4.8: histogram of  $parameter2(p2)$



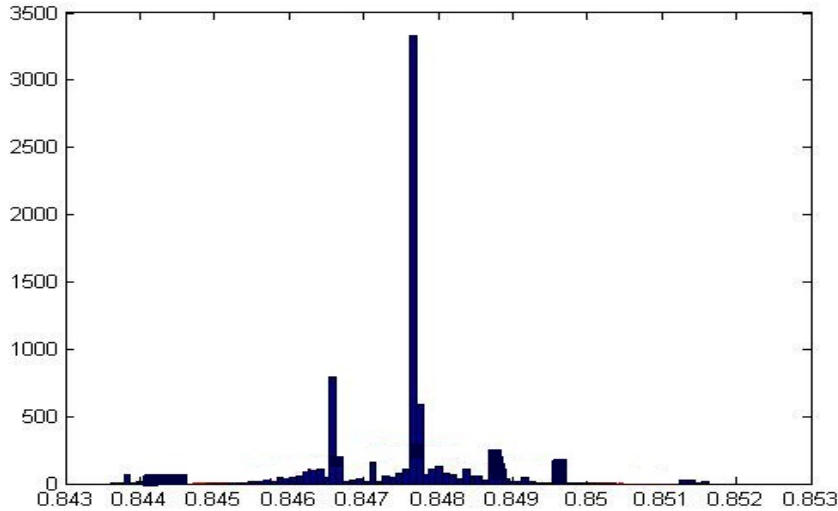
Figure“4.9: histogram of  $parameter3(p3)$



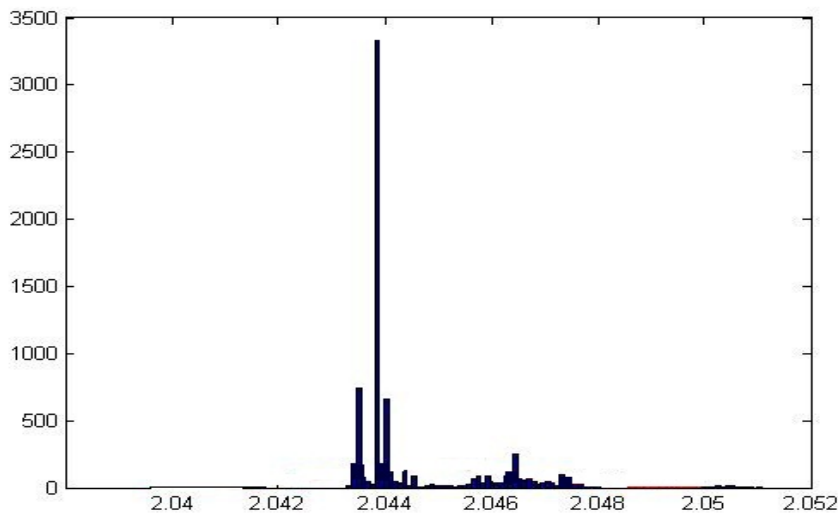
Figure“4.10: histogram of  $parameter4(p4)$



Figure“4.11: histogram of  $parameter5(p5)$



Figure“4.12: histogram of  $parameter6(p6)$



Figure“4.13: histogram of  $parameter7(p7)$

## 4.8 Conclusion

From the theory of Markov Chains, we expect our chains to eventually converge to the target distribution. We often can never surely claim that for how long we should iterate the markov chain to ensure convergence. So we use heuristically decide the convergence i.e., start the sampler from different starting points and monitor whether the trace plots of the draws eventually coincide. And in our problem this is not the case. Hence, we can not say about the convergence of the MCMC.

Based on the distribution curves that we got from the MCMC run we can say that we are getting the mean of the posterior distribution quite close to that of the mean of the prior

distribution, or in the other words we can say that, the mean of the posterior distribution is quite close to that of the “true-value” of the parameter values. For different random seeds values we are getting the similar distribution. But the posterior distribution is not independent of the prior distribution. It surely does depend on the value of the prior mean and the standard deviation of prior, as we have tried for some different set of the prior mean and prior standard deviation set of values, and accordingly we get the different posterior for the same. Hence, We can not comment on the values of the parameters from the results that we are getting.

## **4.9 Application potential**

These types of modeling studies are useful to predict population density of many insect pathogens that are harmful to economically important crops and live stock. Such mathematical models can give indications at which life stage pathogen control measures can induce maximal results.

# Bibliography

- [1] Albert Tarantola, SIAM 2005, Inverse Problem theory and Methods for Model Parameter Estimation.
- [2] Krishna B. Athreya and Soumendra N. Lahiri, 2006 edition, Measure Theory and Probability Theory
- [3] Jun. S. Lin, Department of Statistic, Harvard University, Monte Carlo Strategies in Scientific Computing
- [4] Phil Gregory, Cambridge, Bayesian logical analysis Data analysis, for the physical sciences.
- [5] George Casella, Stephen Fienberg, Ingram Olkin, Springer, Bayesian core: A practical approach to Computational Bayesian statistics
- [6] Robert C. and Casella G. (2004), Monte Carlo Statistical Methods, Second Edition, New York: Springer Verlag
- [7] T.W. Anderson, An introduction to multivariate statistical analysis
- [8] Hakon Tjelmeland, Lecture notes, Introduction to Markov chain Monte Carlo-with examples from Bayesian statistics
- [9] N. Shimkin, Fall 2009, Lecture notes, Derivations of the Discrete time Kalman Filter
- [10] Charles J. Geyer, Introduction to Markov Chain Monte Carlo
- [11] Mark Styvers, Computational statistics with MATLAB
- [12] Biometrika, Vol. 57, No. 1. (Apr., 1970), pp. 97-109, Monte Carlo Sampling Methods Using Markov Chains and Their Applications
- [13] Introduction to Random Signals and Applied Kalman Filtering, 2nd Edition, R.G. brown and P.Y.C. Hwang, John Wiley and Sons, Inc. New York, 1992
- [14] Applied Optimal Estimation. Edited by Arthur Gelb. M.I.T. Press 1986
- [15] Reinhard Furrer, Thomas Bengtsson, Estimation of High-Dimensional Prior and Posterior Covariance matrices in Kalman-Filter Variants.



- [16] R.V. Hogg, E. A. Tanis: Probability and Statistical inference, Mcmillan Publishing Company, 1993
- [17] J.W. Pratt , H. Raiffa, R. Schlaifer: Statistical Decision Theory , MIT Press, 1995
- [18] R.O. Duda, P.E.Hart: Pattern classification and scene analysis, John Wiley and sons, 1973
- [19] W.K Nicholson: Elementary Linear Algebra with Applications. PWS-KENT, 1990
- [20] C.G. Cullen Matrices and Linear Transformations, Dover Publications, 1990
- [21] S. K. Kachigan: Multivariate Statistical Analysis, Radius Press, 1991
- [22] H. Chernoff, L.E. Moses, Elementary Decision Theory, Dover Publications, 1986
- [23] E. Issacson, H.B. Keller: Analysis of Numerical Methods, Dover Publications, 1994
- [24] R.L. Burden, J.D. Fares, Numerical Analysis, PWS-KENT, 1993