Statistical Inference for Nonlinear State Space Models: An Application to the Analysis of Forest Fire Counts in Canada

by

©Zhian Tang

A thesis submitted to the School of Graduate Studies

in partial fulfillment of the requirement for the Degree of

Master of Science

Department of Mathematics and Statistics

Memorial University of Newfoundland

St. John's, Newfoundland and Labrador, Canada   March 2017

Abstract

Nonlinear state space models occupy a predominant position in statistical studies. They are widely used in various fields such as economics, finance, ecology and epidemiology. However, such models may be problematic when it comes to statistical inference, due to the fact that they could be quite sensitive to small variations in system states and parameters. In this dissertation, we present three estimation procedures and their respective algorithms for the statistical inference of such nonlinear, non-Gaussian state space models. Also, simulation studies are carried out to evaluate the performance of these methods. At the end, we analyze the time series of forest fire counts that annually occurred in Canada using the proposed methodologies.

Keywords: Nonlinear, State Space Models, Particle Filter, Iterated Filtering, Approximate Bayesian Computation, Particle Markov Chain Monte Carlo, Forest Fires

Acknowledgements

I'm so grateful to have the opportunity to express my sincere and genuine appreciation to all of you who have provided various kinds of assistance to me in my life studying at Memorial University of Newfoundland.

First and foremost, I would like to say thank you to my supervisor Dr. J Concepción Loredo-Osti. Thank you for your kindness and professionalism that show me how to become a truth seeker, not to mention your generous patience, inspiring encouragement, and unselfish support to my choice during the master's study. Also, thank you for your guidance through the learning and writing process of this thesis paper.

Secondly, I would like to thank all the faculty and staff at the Department of Mathematics & Statistics, especially Dr. J C. Loredo-Osti, Dr. Zhaozhi Fan, Dr. Alwell Oyet, and Dr. Hong Wang. Thank you for providing me excellent courses. The experience of fighting for your tough assignments, preparing for those challenging

exams will always stay in my heart. Thank you for your rigorous academic training.

Finally, thank you all my dear friends. Because of you, my life has become more colorful and wonderful.

Dedication

This dissertation work is specially dedicated to my parents and my fiancee who

always love me and support me unconditionally.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Since the breakthrough article of Kalman (1960) and the early development in engineering, state space models have become an increasingly significant tool for research in a wide range of areas in recent years. Some specific examples are biology (Wilkinson, 2011), control (Ljung, 1999), epidemiology (Keeling and Rohani, 2008) and finance (Tsay, 2005; Hull, 2009). Formally, state space models are also known as partially observed Markov process models, or hidden Markov models. Their constructions usually are intended to reflect the real world phenomena based on certain physical, chemical, or economic principles. As in most cases, only noisy or incomplete observations can

be observed, while the latent system states or the parameter spaces generally remain unknown.

It can be said that, nonlinear state space models are a practical and flexible option to describe different type of systems. Stochastic dynamical models like nonlinear state space models can satisfactorily be used under a wide range of potential causal mechanisms. However, this kind of models lead to many additional complications in terms of their statistical inference. It may happen that the nonlinear property invalidates the use of conventional statistical methods. Therefore, in the past decade, the development of computational methodologies and algorithms has increased substantially.

1.2   Nonlinear State Space Models

In general, state space models consist of an unobserved stochastic state process, and an observation process. The implicit state process connects to the observed data via an explicit, potentially unknown, measurement model.

Now consider a time series of observations $\{Y_{1:n} : Y_1, ..., Y_n\}$, consisting of n observations made at times $t_1, ..., t_n$, and let $\{X_{1:n} : X_1, ..., X_n\}$ denote the state process. The status of the state at a given point in time is latent and unobservable, and the

statistical behavior of this hidden process is determined by the density $h_{X_n|X_{n-1}}$ and the initial density h_{X_0}. On the other hand, the measurement process is modeled by the density $h_{Y_n|X_n}$.
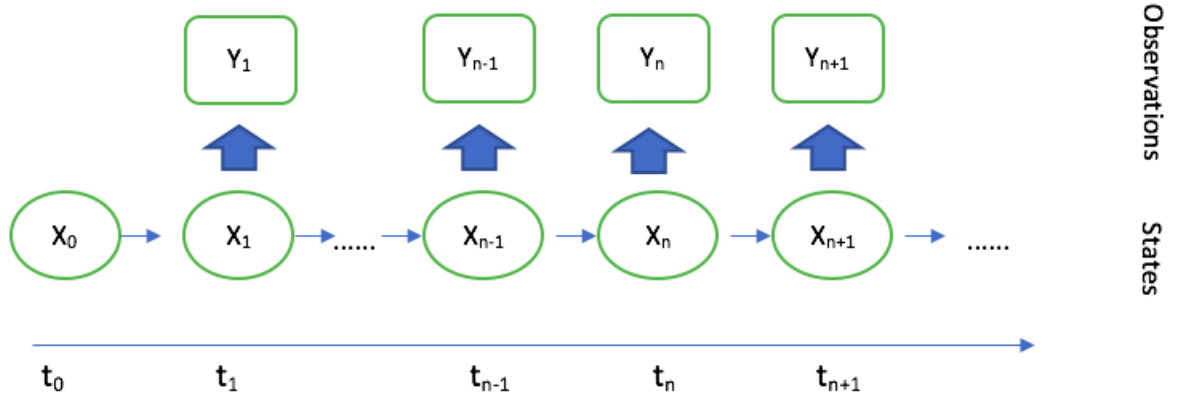
Figure 1.1: State Space Models Schematic

The whole processes can be depicted as the above Figure 1.1, which shows the dependence among model variables. Under the Markovian assumption, the model can be simply expressed as the follows, for all n,

$$X_n|X_{n-1} \sim h_{X_n|X_{n-1}},$$

$$Y_n|X_n \sim h_{Y_n|X_n}.$$

Because of the Markovian property of the process and the relationship between $X_{1:n}$ and $Y_{1:n}$, we know that for the state process, $h_{X_n|X_{0:n-1},Y_{1:n-1}} = h_{X_n|X_{n-1}}$. Moreover, the measurements Y_n depend only on the state at that time, X_n, which is

$h_{Y_n|X_{0:n},Y_{1:n-1}} = h_{Y_n|X_n}$. Let θ be a $p-$dimensional real-valued parameter, $\theta \in \mathbb{R}^p$. The state space structure implies that the joint density is determined by the initial density, $h_{X_0}(x_0;\theta)$, together with the conditional transition probability density, $h_{X_i|X_{i-1}}(x_i|x_{i-1};\theta)$, and the measurement density, $h_{Y_i|X_i}(y_i|x_i;\theta)$, for $i = 1, \cdots, n$. In particular, we have

$$h_{X_{0:n},Y_{1:n}}(x_{0:n}, y_{1:n}; \theta) = h_{X_0}(x_0;\theta) \prod_{i=1}^{n} h_{X_i|X_{i-1}}(x_i|x_{i-1};\theta) h_{Y_i|X_i}(y_i|x_i;\theta)$$

This kind of nonlinear stochastic dynamical systems is widely used to model real data. In this case, the well known Kalman filter method may not be suitable for the statistical inference because of its assumptions. Instead, alternative mathematical procedures to analyze these nonlinear state space models have been proposed. In this context, many procedures have been proposed in recent years, leading to a prosperous atmosphere in the research of nonlinear state space models. In this thesis, we focus on the inference methods for nonlinear state space models. We present the estimation procedures and their algorithms including, iterated filtering (IF), approximate Bayesian computation (ABC) and particle Markov chain Monte Carlo (PMCMC).

1.3   Some Review of Literature

Over the past decade, various researchers have made significant contributions towards the development of methodologies for statistical inference for state space models (Shumway and Stoffer 2006), i.e, partially observed Markov process models (Ionides et al. 2006; Bretó et al. 2009). Sequential Monte Carlo, also known as particle filter (Doucet et al. 2001; Arulampalam et al. 2002; Cappé et al. 2007) provides a standard method to obtain the log likelihood for this kind of stochastic dynamic models. Bretó et al. (2009), He et al. (2010) introduced procedures whose main feature is that the full density does not need to be explicitly evaluated and only a simulator is required for the state space model.

Generally speaking, approaches that work with the full likelihood function are called full-information methods. On the other hand, approaches not based on the full likelihood are called feature-based procedures. Each method may be categorized as full-information or feature-based, Bayesian or frequentist. Both Bayesian (Liu and West 2001; Toni et al. 2009) and frequentist (Ionides et al. 2006; Poyiadjis et al. 2006) approaches to simulation likelihood-based inference via sequential Monte Carlo have been proposed. The maximum likelihood approach of Ionides et al. (2006, 2010) offers a possibility to carry on inference for general nonlinear stochastic state space

models.

1.4 Organization of Thesis

In this thesis, we focus on the inferential procedures and algorithms for nonlinear dynamic state space models. We evaluate the performance of these methods through simulation studies. Also, we conduct a case study applying nonlinear state space models to the analysis of time series data of annual forest fire counts in Canada.

The structure of this thesis is organized as follows. In Chapter 2, we present four algorithms for the statistical inference of nonlinear state space models. The first algorithm is intended to evaluate the likelihood, the other three are parameter estimation procedures. Chapter 3 constructs a nonlinear state space model and explores the performance of the discussed methods through simulation studies. Chapter 4 illustrates the implementation of the model by analyzing the annual numbers of forest fires through Canada as an application. Finally, summary of the entire research and some future work is discussed in Chapter 5.

Chapter 2

Methodologies and Algorithms for Nonlinear State Space Models

Statistical inference for state space models has been an active area of research. Yet, substantial restrictions or strict hypotheses upon the form of models have to be placed in advance when it comes to most existing inference methods. In nonlinear, non-Gaussian situations, some methods such as the Extended Kalman filter or the Gaussian sum filter are proposed to approximate the estimation for filtering and smoothing problems. However, the accuracy of these approximation, in most situations, may be an issue. Sequential Monte Carlo (SMC) methods have emerged as the most popular and successful alternative to the Kalman filter extensions. Many

variations and elaborations to SMC have been proposed. Below, we will discuss four procedures and their respective algorithms that can be used for estimation, prediction and forecasting. These procedures are sequential Monte Carlo (also known as particle filter), iterated filtering (IF), particle Markov chain Monte Carlo (PMCMC), and approximate Bayesian computation (ABC).

2.1 Sequential Monte Carlo (Particle Filter)

Sequential Monte Carlo (SMC) methods, also known as particle filter, have far-reaching and powerful applications in modern time series analysis problems involving state space models. Particularly, they are able to handle those nonlinear, non-Gaussian state space models, since particle filters algorithms do not rely on local linearization techniques or functional approximations. Instead, they are based on a set of simulations, which provides a convenient and attractive approach to computing the posterior distributions.

In the model we discussed in Section 1.2, we have that $X_n|X_{n-1} \sim h_{X_n|X_{n-1}}$ and $Y_n|X_n \sim h_{Y_n|X_n}$. Consider the parameter value $\theta \in \Theta$ and define

$$X_n|(X_{n-1} = x) \sim h_{X_n|X_{n-1}} \equiv f_\theta(\cdot|x)$$

$$Y_n|(X_n = x) \sim h_{Y_n|X_n} \equiv g_\theta(\cdot|x)$$

where $f_\theta(\cdot|x)$ is the transition probability density for some static parameter θ and $g_\theta(\cdot|x)$ is the marginal density probability. For the state process $\{X_n; n \geq 1\}$, the initial density is $X_1 \sim \pi_\theta(\cdot)$.

The goal is to perform Bayesian inference conditional on the observations $\{y_{1:N} = (y_1, \cdots, y_N)\}$ for $N \geq 1$. When $\theta \in \Theta$ is a known parameter, the posterior density $\pi_\theta(x_{1:N}|y_{1:N})$ is proportional to $\pi_\theta(x_{1:N}, y_{1:N})$. That is $\pi_\theta(x_{1:N}|y_{1:N}) \propto \pi_\theta(x_{1:N}, y_{1:N})$ where

$$\pi_\theta(x_{1:N}, y_{1:N}) = \pi_\theta(x_1)g_\theta(y_1|x_1) \prod_{n=2}^{N} f_\theta(x_n|x_{n-1})g_\theta(y_n|x_n).$$

If $\theta \in \Theta$ is unknown, we denote $\pi(\theta)$ as the prior density of θ. Then, the posterior density is proportional to the joint density

$$\pi(\theta, x_{1:N}|y_{1:N}) \propto \pi_\theta(x_{1:N}, y_{1:N})\pi(\theta).$$

The difficulty of statistical inference for nonlinear, non-Gaussian state space models is that the densities $\pi_\theta(x_{1:N}, y_{1:N})$ and $\pi(\theta, x_{1:N}|y_{1:N})$ generally do not have a closed form expressions. Therefore, these densities need to be approximated or evaluated numerically.

Now we can factorize the likelihood in the following way:

$$L(\theta) = \pi_\theta(y_{1:N})$$

$$= \prod_{n=1}^{N} \pi_\theta(y_n|y_{1:n-1}) \tag{2.1}$$

Noted that

$$\pi_\theta(y_n|y_{1:n-1}) = \int_{x_n} \pi_\theta(y_n, x_n|y_{1:n-1})dx_n$$

$$= \int_{x_n} \pi_\theta(x_n|y_{1:n-1})\pi_\theta(y_n|x_n, y_{1:n-1})dx_n$$

$$= \int_{x_n} \pi_\theta(x_n|y_{1:n-1})\pi_\theta(y_n|x_n)dx_n, \tag{2.2}$$

we have

$$L(\theta) = \prod_{n=1}^{N} \int \pi_\theta(x_n|y_{1:n-1})\pi_\theta(y_n|x_n)dx_n.$$

Moreover, we can obtain the prediction formula using the Markovian property as

$$\pi_\theta(x_n|y_{1:n-1}) = \int_{x_{n-1}} \pi_\theta(x_n, x_{n-1}|y_{1:n-1})dx_{n-1}$$

$$= \int_{x_{n-1}} \pi_\theta(x_{n-1}|y_{1:n-1})\pi_\theta(x_n|x_{n-1}, y_{1:n-1})dx_{n-1}$$

$$= \int_{x_{n-1}} \pi_\theta(x_{n-1}|y_{1:n-1})\pi_\theta(x_n|x_{n-1})dx_{n-1} \tag{2.3}$$

The filtering formula can be obtained by using the Bayes' theorem as

$$\pi_\theta(x_n|y_{1:n}) = \pi_\theta(x_n|y_n, y_{1:n-1})$$

$$= \frac{\pi_\theta(y_n|x_n, y_{1:n-1})\pi_\theta(x_n|y_{1:n-1})}{\pi_\theta(y_n|y_{1:n-1})}$$

$$= \frac{\pi_\theta(x_n|y_{1:n-1})\pi_\theta(y_n|x_n)}{\int_{x_n} \pi_\theta(x_n|y_{1:n-1})\pi_\theta(y_n|x_n)dx_n}. \tag{2.4}$$

Overall, the prediction and filtering formulas give us a recursion. Specifically, the prediction formula gives the prediction distribution at time n using the filtering

distribution at time $n - 1$, and the filtering formula gives the filtering distribution at time n using the prediction distribution at time n, for all $n = 1, 2, \cdots, N$.

Now denote $X_{n-1,j}^F$, $j = 1, \ldots, J$ as a set of J points drawn from the filtering distribution $\pi_\theta(x_{n-1}|y_{1:n-1})$ at time $n-1$ and $X_{n,j}^P$ as points drawn from the prediction distribution $\pi_\theta(x_n|y_{1:n-1})$ at time n by simply simulating the process model: $X_{n,j}^P \sim \pi_\theta\left(x_n|X_{n-1,j}^F\right)$ for j in $1 : J$.

Having obtained $X_{n,j}^P$, we can get a sample of points from the filtering distribution $\pi_\theta(x_n|y_{1:n})$ at time n by resampling from $\left\{X_{n,j}^P, j \in 1 : J\right\}$ with weights $w_{n,j} = \pi_\theta\left(y_n|X_{n,j}^P\right)$. In addition, the Monte Carlo methods provide us an approximation to the conditional likelihood that we obtained from the above. That is

$$
\begin{aligned}
L_n(\theta) &= \pi_\theta(y_n|y_{1:n-1}) \\
&= \int_{x_n} \pi_\theta(x_n|y_{1:n-1})\pi_\theta(y_n|x_n)dx_n,
\end{aligned}
\tag{2.5}
$$

can be estimated by

$$
\hat{L}_n(\theta) \approx \tfrac{1}{N} \sum_j \pi_\theta(y_n|X_{n,j}^P),
$$

wherever $X_{n,j}^P$ is random sample drawn from $\pi_\theta(x_n|y_{1:n-1})$.

Now we can iterate this procedure through the data, one step at a time, alternately simulating and resampling, until we reach $n = N$. Then the full log likelihood has

approximation:

$$\ell(\theta) = \log L(\theta)$$

$$= \sum_n \log L_n(\theta)$$

$$\approx \sum_n \log \hat{L}_n(\theta). \tag{2.6}$$

In general, there is a more generic way to express the whole calculation procedure of particle filter (Andrieu et al. 2010). We aim to yield the estimate, $\hat{\pi}$. The procedure can be summarized in the following steps.

Particle Filter Algorithm

Step 1. At time $t = 1$, define an importance density $v(\cdot)$ for importance sampling; we aim to approximate $\pi_\theta(x_1|y_1)$.

   a. Draw a sample of J particles $X_1^k = (X_1^1, \cdots, X_1^J)$ from $v_\theta(x_1|y_1)$.

   b. Calculate the normalized importance weights and denote them as $W_1^k = (W_1^1, \cdots, W_1^J)$, where

$$w_1(X_1^k) = \frac{\pi_\theta(X_1^k, y_1)}{v_\theta(X_1^k|y_1)}$$

$$W_1^k = \frac{w_1(X_1^k)}{\sum_{m=1}^J w_1(X_1^m)}.$$

c. The estimate of $\pi_\theta(x_1|y_1)$ can be calculated by

$$\hat{\pi}_\theta(x_1|y_1) = \sum_{k=1}^{J} W_1^k \delta_{X_1^k}(x_1)$$

where $\delta_x(\cdot)$ is the Dirac Delta function.

d. Use these particles and weights to resample J new particles from the approximation, $\hat{\pi}_\theta(x_1|y_1)$.

Step 2. Iteration. At time $t = 2, 3, \cdots, N - 1$, we again use importance sampling to approximate $\pi_\theta(x_{1:t}|y_{1:t})$.

a. Denote A_{t-1}^k as the index of the parent of particles $X_{1:t}^k$ at time $t - 1$; draw a sample $A_{t-1}^k \sim \mathcal{M}(\cdot|\mathbf{W}_{t-1})$, where $\mathbf{W}_t = (W_t^1, \cdots, W_t^J)$ and $\mathcal{M}(\cdot|\mathbf{p})$ is the multinomial distribution with parameter $\mathbf{p}$.

b. Draw a sample $X_t^k \sim v\left(\cdot|y_t, X_{t-1}^{A_{t-1}^k}\right)$, and set $X_{1:t}^k = \left(X_{1:(t-1)}^{A_{t-1}^k}, X_t^k\right)$.

c. Calculate the normalized importance weights:

$$w_t(X_{1:t}^k) = \frac{\pi_\theta(X_{1:t}^k, y_{1:t})}{\pi_\theta\left(X_{1:(t-1)}^{A_{t-1}^k}, y_{1:(t-1)}\right) v_\theta\left(X_t^k|y_t, X_{t-1}^{A_{t-1}^k}\right)}$$

$$W_t^k = \frac{w_t(X_{1:t}^k)}{\sum_{m=1}^{J} w_t(X_{1:t}^m)}.$$

d. The procedure yields the approximation of the posterior density $\pi_\theta(x_{1:t}|y_{1:t})$

given by

$$\hat{\pi}_\theta(x_{1:t}|y_{1:t}) = \sum_{k=1}^{J} W_t^k \delta_{X_{1:t}^k}(x_{1:t})$$

where $\delta_x(\cdot)$ is the Dirac Delta function.

Step 3. At time $t = N$, the procedure yields the approximation of the posterior density

$\pi_\theta(x_{1:N}|y_{1:N})$ given by

$$\hat{\pi}_\theta(x_{1:N}|y_{1:N}) = \sum_{k=1}^{J} W_N^k \delta_{X_{1:N}^k}(x_{1:N}).$$

At the end, the estimate of the likelihood $\pi_\theta(y_{1:N})$ is

$$\hat{\pi}_\theta(y_{1:N}) = \hat{\pi}_\theta(y_1) \prod_{t=2}^{N} \hat{\pi}_\theta(y_t|y_{1:(t-1)})$$

where

$$\hat{\pi}_\theta(y_t|y_{1:(t-1)}) = \frac{1}{N} \sum_{k=1}^{J} w_t(X_{1:t}^k).$$

2.2   Iterated Filtering

Iterated filtering technique is a significant inference method that can maximize the likelihood obtained by SMC (Ionides et al. 2006, 2011). It's specially practical to the state space models. The idea of iterated filtering is that an optimization can be obtained by taking parameter perturbations into consideration when iteratively reconstructing the latent states. In terms of the unknown parameter space, stochastic perturbations are introduced into the method, which can be dynamically used to search for a suitable parameter estimate.

The main goal of the algorithm is to find the maximum likelihood estimates of the unknown parameters. As long as a proper procedure iterates with successively diminished perturbations, the estimating result will converge to the maximum likelihood estimate. Ionides et al (2015) improve the iterated filtering algorithm based on the convergence of an iterated Bayes map, and name the algorithm as IF2. In general, the IF2 algorithm can be summarized as the following procedure.

Iterated Filtering Algorithm

Step 1. Initialization. Arbitrary starting parameter $[\theta_j]_0$, where $j = 1, 2, \cdots J$ with J as the number of particles; set $\gamma_0(\cdot|[\theta_j]_0; \sigma_m)$ as the initial perturbation density,

where $m = 1, 2, \cdots, M$ with M as the number of operation. Let the superscript

F represent filtering recursion and P represent prediction recursion.

Step 2. Iteration. For $m = 1, 2, \cdots, M$,

1. Draw a random sample of parameter, $[\theta_{0,j}^F]_m \sim \gamma_0(\theta|[\theta_j]_{m-1}; \sigma_m)$.

2. Draw a random sample of states $[X_{0,j}^F]_m \sim f_{X_0}(x_0; [\theta_{0,j}^F]_m)$.

3. Iteration. For $n = 1, 2, \cdots, N$,

   a. Draw a random sample $[\theta_{n,j}^P]_m \sim \gamma_n(\theta|[\theta_{n-1,j}^F]_m; \sigma_m)$.

   b. Draw a random sample $[X_{n,j}^P]_m \sim \pi(x_n|[X_{n-1,j}^F]_m; [\theta_j^P]_m)$.

   c. Calculate weights: $[w_{n,j}]_m = \pi(y_n|[X_{n,j}^P]_m; [\theta_{n,j}^P]_m)$.

   d. Draw indices $k_{1:J}$ with $P\{k_j = s\} = [w_{n,s}]_m / \sum_{u=1}^{J} [w_{n,u}]_m$.

   e. Let $[\theta_{n,j}^F]_m = [\theta_{n,k_j}^P]_m$ and $[X_{n,j}^F]_m = [X_{n,k_j}^P]_m$.

4. Set $[\theta_j]_m = [\theta_{N,j}^F]_m$.

Step 3. After M times iterations, we can obtain the Maximum Likelihood Estimate $\hat{\theta}_M$.

2.3   Particle Markov Chain Monte Carlo (PMCMC)

Particle Markov chain Monte Carlo (PMCMC) is a Bayesian inference method using full information. It's proposed by Andrieu et al. (2010) to perform inference on the unknown parameter vector θ. It targets the full joint posterior distribution $\pi(\theta, x_{1:N}|y_{1:N})$. PMCMC methods combine likelihood evaluation via particle filter with MCMC moves in the parameter space. It works well in nonlinear non-Gaussian scenarios while the traditional MCMC methods can fail in this specific situation. One common used PMCMC algorithm is termed as particle marginal Metropolis-Hastings (PMMH). It plugs the unbiased likelihood estimate obtained by particle filter into the Metropolis-Hastings update procedure to get the desired posterior distribution for the parameters (Andrieu and Roberts 2009).

First we take a brief review of the Metropolis-Hastings algorithm, which is one of the most common MCMC algorithms. It can generate correlated variables from a Markov chain. Given the target density $\pi(x_{1:N}, y_{1:N})$, it is associated with a proposed density $v(\cdot|x)$.

So the MH algorithm can be summarized as below. Given a target density $\pi(\mathbf{x})$ and a proposal density $v(\cdot|\mathbf{x})$, a new Markov chain $\{X_m^*\}$ whose stationary distribution is $\pi(x)$ can be generated by the following algorithm:

Metropolis-Hastings Algorithm

Step 1. Start with an arbitrary $\mathbf{x}_0^*$, generate $\mathbf{x}'$ from $v(\cdot|\mathbf{x}_{m-1}^*)$.

Step 2. Compute the probability $\rho(\mathbf{x}_m^*, \mathbf{x}') = \min\left\{1, \frac{\pi(\mathbf{x}')v(\mathbf{x}_m^*|\mathbf{x}')}{\pi(\mathbf{x}_m^*)v(\mathbf{x}'|\mathbf{x}_m^*)}\right\}$.

Step 3. Set $\mathbf{x}_{m+1}^* = \mathbf{x}'$ with probability $\rho(\mathbf{x}_m^*, \mathbf{x}')$; set $\mathbf{x}_{m+1}^* = \mathbf{x}_m^*$ with probability

$1 - \rho(\mathbf{x}_m^*, \mathbf{x}')$.

In our state space models, when the parameter θ is unknown, we are interested in sampling from $\pi(\theta, x_{1:N}|y_{1:N})$. The PMMH algorithm will focus on jointly updating θ and $x_{1:N}$. Given that $\pi(\theta, x_{1:N}|y_{1:N}) = \pi(\theta|y_{1:N})\pi_\theta(x_{1:N}|y_{1:N})$, a natural choice of proposal density for an MH update is

$$v(\theta^*, x_{1:N}^*|\theta, x_{1:N}) = v(\theta^*|\theta)\pi_{\theta^*}(x_{1:N}^*|y_{1:N}). \tag{2.7}$$

So the resulting MH acceptance ratio is given by

$$A = \frac{\pi(\theta^*, x_{1:N}^*|y_{1:N})v(\theta, x_{1:N}|\theta^*, x_{1:N}^*)}{\pi(\theta, x_{1:N}|y_{1:N})v(\theta^*, x_{1:N}^*|\theta, x_{1:N})}$$

$$= \frac{\pi_{\theta^*}(y_{1:N})\pi(\theta^*)v(\theta|\theta^*)}{\pi_\theta(y_{1:N})\pi(\theta)v(\theta^*|\theta)}. \tag{2.8}$$

The PMMH algorithm can be summarized as the following procedure.

PMMH Algorithm

Step 1. Initialization. Arbitrary starting parameter θ_0; run the SMC algorithm with J particles targeting $\pi_{\theta_0}(x_{1:N}|y_{1:N})$ to obtain the estimates $\hat{\pi}_{\theta_0}(\cdot|y_{1:N})$ and the marginal likelihood estimate $\hat{\pi}_{\theta_0}(y_{1:N})$; draw a random sample $X_{1:N}^0 \sim \hat{\pi}_{\theta_0}(\cdot|y_{1:N})$.

Step 2. Iteration. Denote M as the number of operation. For $m = 1, 2, \cdots, M$,

1. Draw a parameter θ^P from the proposal distribution $\theta^P \sim v(\cdot|\theta_{m-1})$.

2. Run the SMC algorithm with J particles targeting $\pi_{\theta^P}(x_{1:N}|y_{1:N})$ to obtain the density estimate $\hat{\pi}_{\theta^P}(\cdot|y_{1:N})$ and the marginal likelihood estimate $\hat{\pi}_{\theta^P}(y_{1:N})$; draw a random sample $X_{1:N}^P \sim \hat{\pi}_{\theta^P}(\cdot|y_{1:N})$.

3. Calculate the probability $\rho_m = \min\left\{1, \frac{\pi_{\theta^P}(y_{1:N})\pi(\theta^P)v(\theta_{m-1}|\theta^P)}{\pi_{\theta_{m-1}}(y_{1:N})\pi(\theta_{m-1})v(\theta^P|\theta_{m-1})}\right\}$.

4. With probability ρ_m, set $\theta_m = \theta^P$, $X_{1:N}^m = X_{1:N}^P$, $\hat{\pi}_{\theta_m}(y_{1:N}) = \hat{\pi}_{\theta^P}(y_{1:N})$; otherwise, set $\theta_m = \theta_{m-1}$, $X_{1:N}^m = X_{1:N}^{m-1}$, $\hat{\pi}_{\theta_m}(y_{1:N}) = \hat{\pi}_{\theta_{m-1}}(y_{1:N})$.

Step 3. After M times iterations, we will have the samples $\theta_{1:M}$ where the posterior distribution $\pi(\theta|y_{1:N})$ can be obtained as well as the set of particles $X_{1:M}^m$.

2.4   Approximate Bayesian Computation (ABC)

ABC algorithms are Bayesian feature-based methods to evaluate the posterior distributions through simulations instead of calculations of likelihood functions. They compare the distance between the observed and simulated data (Pritchard et al. 1999; Marjoram et al. 2003; Sisson et al. 2007).

Let θ as the parameter vector we are about to estimate in our models. Denote $\pi(\theta)$ as its prior distribution and x_0 as the observed data. Our goal is to obtain an approximation of the posterior distribution, $\pi(\theta|x_0)$. According to Bayes' Theorem, we have $\pi(\theta|x_0) \propto f(x_0|\theta)\pi(\theta)$, where $f(x_0|\theta)$ is the likelihood of θ given the observed data x_0.

A simple ABC algorithm is called the rejection sampler (Pritchard et al. 1999). It includes the following steps:

Rejection Sampler Algorithm

Step 1. Sample a candidate parameter θ^* from a the prior distribution $\pi(\theta)$.

Step 2. Simulate a dataset x^* from $f(x|\theta^*)$.

Step 3. Compare the distance between the simulated data x^* and the observed data x_0

using a distance function D; and a tolerance ϵ, if $D(x_0, x^*) \leq \epsilon$, then accept θ^*, otherwise reject.

Step 4. Repeat the above steps until one has a sample $\{\theta_k^*\}$ of size M where inference about $\pi(\theta \mid x)$ can be done by taking uniform weights $W^k = \frac{1}{M}$ for $k = 1, 2, \ldots, M$.

Often times, instead of using the full data set to obtain the posterior distribution, either because we only have summary statistics or because of the dimension of the data, we would like to obtain the distribution of the parameters given a set of summary statistics. For this situation, we define a distance function based on the summary statistics. These summary statistics are the features of the full dataset. The features, also called probes (Kendall et al. 1999), are denoted by a collection of functions, $\mathbb{S} = (\mathbb{S}_1, \cdots, \mathbb{S}_d)$, where each $\mathbb{S}_i$ maps an observed time series to a real number. We write $S = (S_1, \cdots, S_d)$ for the vector-valued random variable with $S = \mathbb{S}(Y_{1:N})$, with $h_S(s; \theta)$ being the corresponding joint density. Also, the observed feature vector is s^0 where $s_i^0 = \mathbb{S}_i(y_{1:N})$. The goal of ABC is to estimate the posterior distribution of the unknown parameters given $S = s^0$. Denote the distance function as ρ and s as the summary statistics of the simulated data, then we will accept θ if $\rho(s^0, s) \leq \epsilon$.

But the acceptance rate of this ABC rejection sampler will be quite low if the prior distribution vastly differs from the posterior distribution. To solve the problem, Marjoram et al. (2003) proposed an ABC method based on Markov chain Monte Carlo. The procedure can be summarized as below:

ABC Algorithm

Step 1. Initialization. Arbitrary starting parameter θ_0.

Step 2. Iteration. Denote M as the number of operation. For $m = 1, 2, \cdots, M$,

    1. Draw a proposed parameter θ^P from a proposal distribution $\theta^P \sim v(\cdot | \theta_{m-1})$.

    2. Sample a dataset x^P from $f(x | \theta^P)$.

    3. Compute observed probes s_0 and the simulated probes s^P.

    4. Calculate the probability $p_m = \min\left\{ 1, \frac{\pi(\theta^P) v(\theta_{m-1} | \theta^P)}{\pi(\theta_{m-1}) v(\theta^P | \theta_{m-1})} \mathbb{I}_{[\rho(s_0, s^P) \leq \epsilon]} \right\}$.

    4. With probability p_m, set $\theta_m = \theta^P$; otherwise, set $\theta_m = \theta_{m-1}$.

Step 3. After M times iterations, we obtain the samples $\theta_{1:M}$, as well as the posterior distribution of parameters.

2.5   The R Package: POMP

The R package POMP (King et al. 2016) provides a suite of tools for analysis of time series data based on state space models. It provides a very flexible framework for statistical inference using nonlinear, non-Gaussian state space models. Many modern statistical methods have been implemented in this framework including sequential Monte Carlo, iterated filtering, particle Markov chain Monte Carlo, approximate Bayesian computation, maximum synthetic likelihood estimation, etc.

POMP is fully object-oriented. A partially observed Markov process model is represented by an object of class 'pomp'. Methods for the class 'pomp' use various components to carry out computations on the model. A brief summary of the mathematical notations corresponding to the elementary methods is shown as below:

Table 2.1: A Brief Summary of notations for POMP Models

Method	Mathematical terminology
rprocess	Simulate from $h_{X_n|X_{n-1}}(x_n|x_{n-1};\theta)$
dprocess	Evaluate $h_{X_n|X_{n-1}}(x_n|x_{n-1};\theta)$
rmeasure	Simulate from $h_{Y_n|X_n}(y_n|x_n;\theta)$
dmeasure	Evaluate $h_{Y_n|X_n}(y_n|x_n;\theta)$
rprior	Simulate from the prior distribution $\pi(\theta)$
dprior	Evaluate the prior density $\pi(\theta)$
init.state	Simulate from $h_{X_0}(x_0;\theta)$
timezero	Initial time t_0
time	Times $t_{1:N}$
obs	Data $y_{1:N}$
states	States $x_{0:N}$
coef	Parameters θ

There are many examples illustrated in the package, as well as a large volume of corresponding R and C codes that are provided with the package. Further documentation and an introductory tutorial can be found on the POMP website, http://kingaa.github.io/pomp.

Chapter 3

Simulation Study

Nonlinear state space models can be used to analyze the time series of count data. One of the applications is to model the number of major (with magnitude 7 or higher on the Richter scale) earthquakes each year. We use the model from Langrock (2011) and Zeger (1988), which assumes that the number of earthquakes y_t is a conditional Poisson distributed variable with mean λ_t. Also, we assume that the mean of λ_t follows an $AR(1)$ process,

$$log(\lambda_t) - \mu = \phi(log\lambda_{t-1} - \mu) + \sigma v_t$$

where v_t denotes a standard Gaussian random variable. By introducing $x_t = log(\lambda_t) - \mu$ and $\beta = exp(\mu)$, we obtain the state space model as below:

$$x_{t+1}|x_t \sim \mathcal{N}(\phi x_t, \sigma^2)$$

$$y_t|x_t \sim \mathcal{P}(\beta e^{x_t})$$

where the parameter vector is $\theta = \{\beta, \sigma, \phi\}$ with the constraints $\phi \in (-1, 1) \subset \mathbb{R}$ and $\{\sigma, \beta\} \in \mathbb{R}_+^2$. Here, $\mathcal{P}(\lambda)$ denotes a Poisson distributed variable with mean λ. That is, the probability of $k \in \mathbb{N}$ earthquakes during year t is given by the probability mass function (PMF),

$$P\{Y_t = k\} = e^{-\lambda}\frac{\lambda^k}{k!}$$

In this chapter we set up a nonlinear state space model and present the simulation study of a time series of count data to compare the four methodologies we have studied: SMC, IF, PMCMC, and ABC.

3.1   Simulation Setup

We use R software to generate the random numbers. Set the parameter vector θ as $\theta = \{\beta = 1, \sigma = 0.1, \phi = 0.2\}$ and the initial state as $x_0 = 1$. It leads to the state process,

$$x_{t+1}|x_t \sim \mathcal{N}(0.2x_t, 0.1^2)$$

and for the measurement process,

$$y_t | x_t \sim \mathcal{P}(e^{x_t}).$$

We simulate a time period from 1 to 100, and the nonlinear state space model generate a series of simulated counts. The plot of simulated counts is shown in Figure 3.1. Since the true value of parameters we set up are small, the largest simulated outcomes are just 4 and there are lots of zero in this simulation.
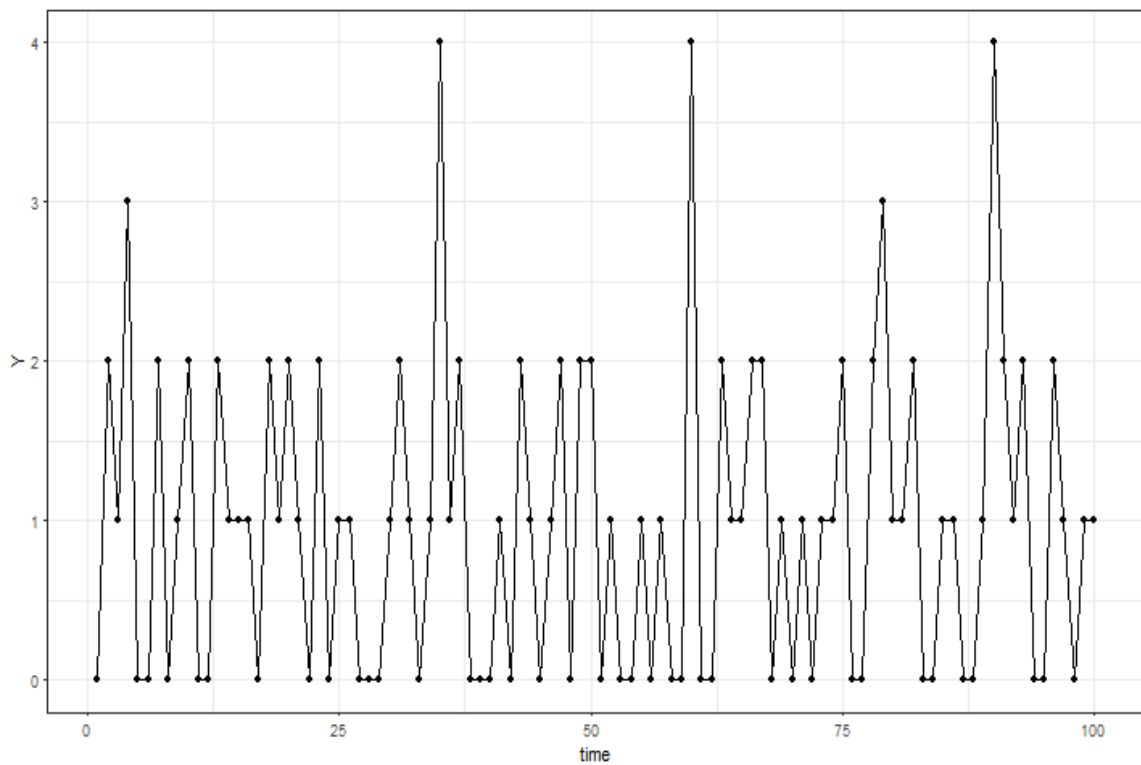

Figure 3.1: The Simulated Counts

3.2 Simulation Analysis

Based on the series of count data we simulated, we can use the iterated filtering approach to obtain the maximum likelihood estimate of parameters. In R, it can be used the function named `mif` in POMP package to obtain the results. Since the parameters in our model are constrained to be positive, we prefer to transform them into an unconstrained scale to estimate instead. So we designate the logarithm transformation to the parameters when estimating. In order to improve the computation efficiency, the foreach R package (Revolution Analytics and Weston 2014) will be used to parallelize the computations.

We run 10 trajectories, and for each run, the number of iterations is 100, the number of particles is 2000. The calculation result is shown in Figure 3.2. It shows that each trajectory converges to an estimate. For the parameter $\log \sigma$ and $\log \phi$, most of runs have different estimations after 100 iterations. But for $\log \beta$ and the log-likelihood $\log L$, nine of ten runs converges to a really close estimate. Usually, we will focus on the estimate with the highest estimated log-likelihood.

Specifically, we can take a comparison between the parameter estimates and their true values. From Table 3.1, we obtain the MLE of parameters as $\hat{\theta}_{MLE} = \{\beta = 0.949, \sigma = 0.213, \phi = 0.0529\}$. The log likelihood $\hat{\ell}$ estimated by particle filter at
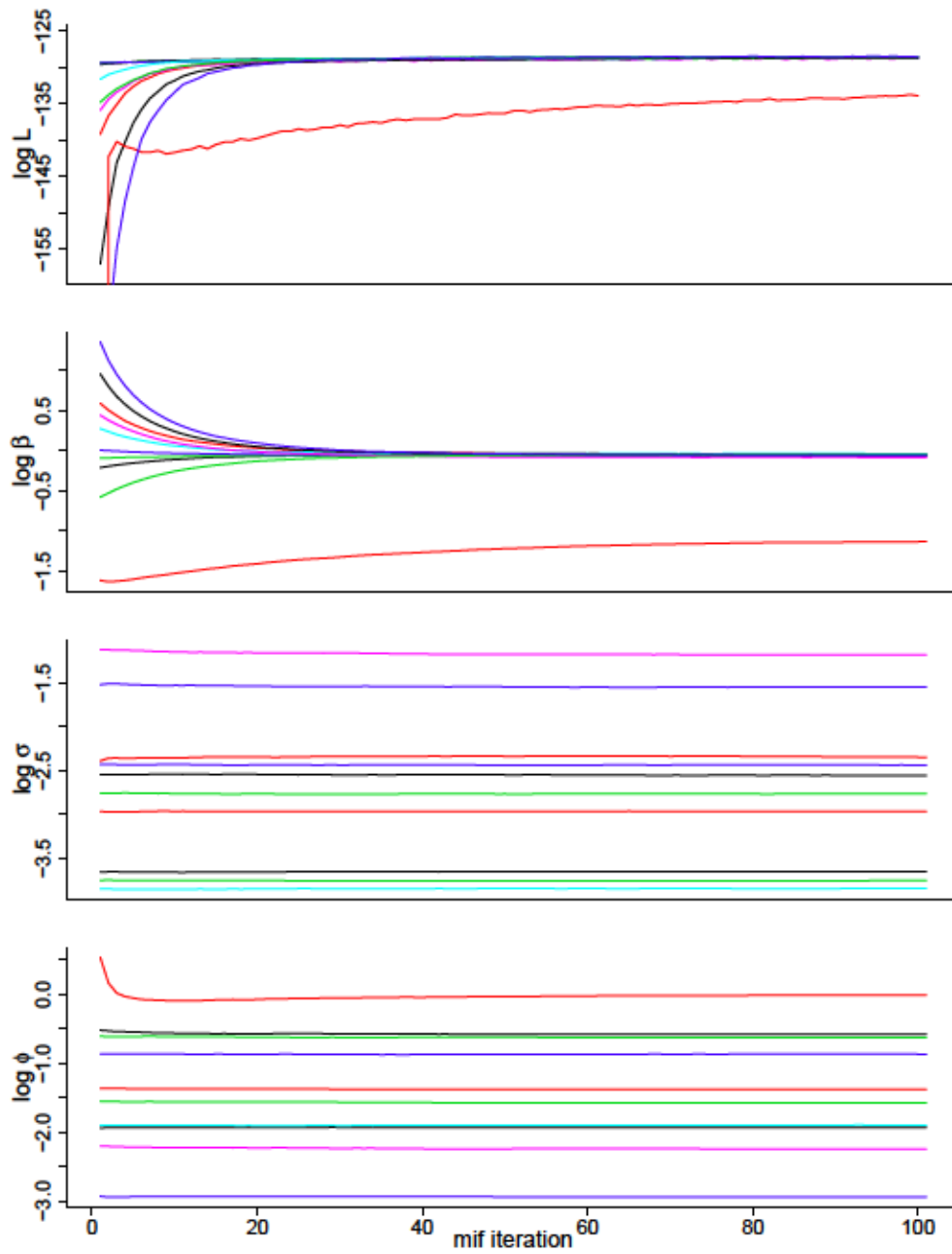
Figure 3.2: Iterated Filter Iteration

MLE is -128.39, and its standard error is 0.01.

Table 3.1: Results of Estimating Parameters Using IF2 Algorithm

	β	ϕ	σ	$\hat{\ell}$	s.e.
$Truth$	1.0000	0.2000	0.1000	-128.58	0.01
mif MLE	0.9490	0.0529	0.2130	-128.39	0.01

For the iterated filtering method, we can consider two more cases with other true values as comparison. In Case Two, we set $\theta_2 = \{\beta = 5, \sigma = 0.3, \phi = 0.5\}$, and the MLE obtained by the iterated filtering is $\hat{\theta}_{2MLE} = \{\beta = 4.97, \sigma = 0.372, \phi = 0.47\}$ with the log-likelihood -253.91. The result is shown in Table 3.2.

Table 3.2: Results of Estimating Parameters Using IF2 Algorithm in Case Two

	β	ϕ	σ	$\hat{\ell}$	s.e.
$Truth$	5.0000	0.5000	0.3000	-254.61	0.05
mif MLE	4.9700	0.4700	0.3720	-253.91	0.13

Likewise, we set the true values of θ in Case Three as $\theta_3 = \{\beta = 10, \sigma = 0.5, \phi = 0.7\}$. Correspondingly, we can obtain the MLE as $\hat{\theta}_{3MLE} = \{\beta = 6, \sigma = 0.488, \phi = 0.61\}$. The detailed result is shown in Table 3.3.

Table 3.3: Results of Estimating Parameters Using IF2 Algorithm in Case Three

	β	ϕ	σ	$\hat{\ell}$	s.e.
$Truth$	10.0000	0.7000	0.5000	-291.38	0.09
mif MLE	6.0000	0.6100	0.4880	-288.00	0.19

The above shows that the `mif` procedure can successfully maximize the likelihood and propose a reasonable parameter estimate.

In the following methods, we just consider the original values of θ, which is $\theta = \{\beta = 1, \sigma = 0.1, \phi = 0.2\}$. Now the statistical estimation for the unknown parameters can be carried out by using PMCMC algorithm that we've discussed in Section 2.3. PMCMC is a full-information Bayesian method that it pays large price to run the SMC algorithm to finally obtain the Metropolis-Hastings acceptance probability. Using the R function pmcmc in POMP package, we specify a uniform prior distribution on unknown parameters and set the particles number as 100. We run 5 independent MCMC chains with 30,000 iterations for each chain. After a mass of calculation, we can obtain a swam of the posterior parameter estimates.

Table 3.4: PMCMC Quantiles for Each Parameter

	2.50%	25%	50%	75%	97.50%
β	0.75109	0.87227	0.9438	1.0096	1.1577
σ	0.01763	0.07494	0.1443	0.2401	0.4551
ϕ	0.03149	0.14002	0.2813	0.4834	0.8134

Table 3.4 shows the PMCMC quantiles for each parameters, and we use the 50% quantile as the estimates of unknown parameters. So we can obtain $\hat{\theta}_{PMCMC} = \{\beta = 0.9438, \sigma = 0.1443, \phi = 0.2813\}$.

Besides, we can calculate the mean and the standard deviation of paramters β, σ, and ϕ. As the Bayesian inference, we can also calculate their naive standard errors and the time-series standard error from the big volume of posterior parameter sample

set.

Table 3.5: PMCMC Empirical Mean and Standard Deviation for Each Parameter, Plus Standard Error of the Mean

	Mean	SD	Naive SE	Time-series SE
β	0.9432	0.1031	0.002061	0.007088
σ	0.1679	0.118	0.002358	0.009738
ϕ	0.3248	0.2244	0.004483	0.034891

Table 3.5 reflects the PMCMC empirical mean and standard deviation for each parameter, as well as the standard error of the mean. As we know, the true value of β is 1, while the mean of the posterior distribution of β is 0.9432 and its standard deviation is 0.1031.

The diagnostic plots for the PMCMC algorithm is shown in Figure 3.3. The trace plots in the left column show the evolution of 5 independent MCMC chains from iterations 10001 to 30001. We can see that for β and σ, the five traces are relatively close, while the traces of ϕ are very different. In addition, based on the big swam of posterior estimates, we can also plot the kernel density estimates of the marginal posterior distributions, which are shown at right. Specifically, the solid vertical line is the true parameters and the red dashed line is the PMCMC estimates of parameters.
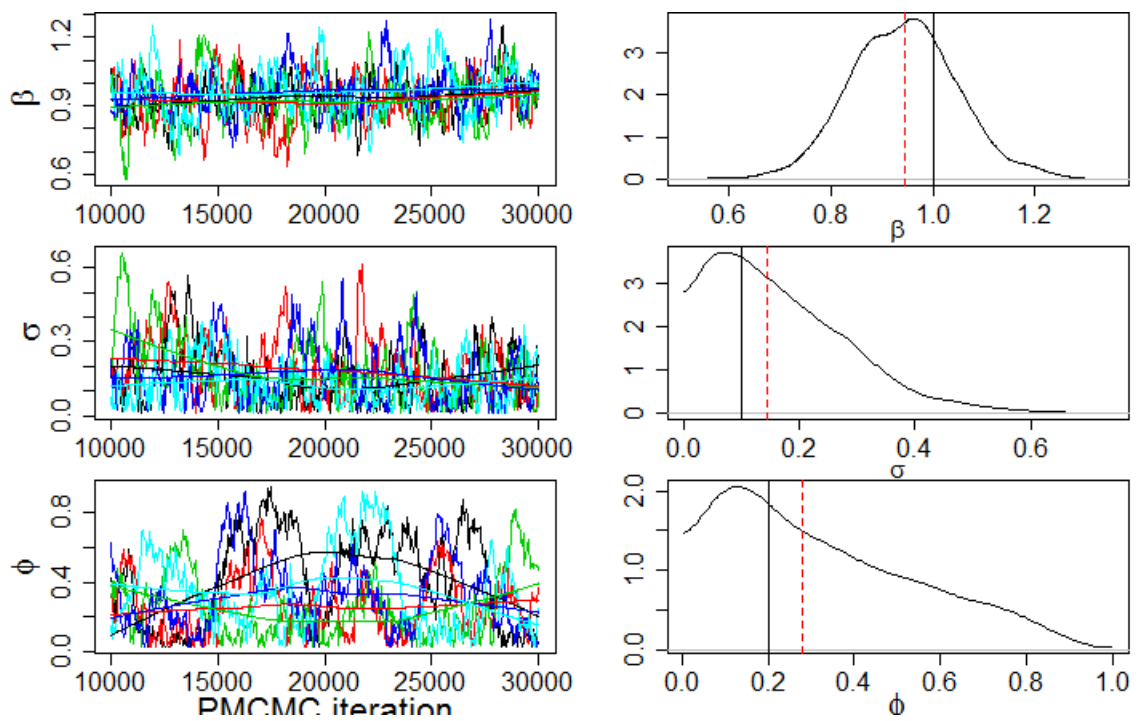
Figure 3.3: The Diagnostic Plots for the PMCMC Algorithm

As comparison, we apply the ABC algorithm to obtain the evaluation of unknown parameters. Since the ABC algorithm use partial features of the data, we need to designate the probes first. We set up the mean and their autocorrelation function as their probes. In Section 2.4, we've discussed the procedures. Now using the R function `abc` in POMP package, we can obtain the posterior parameter estimates based on partial feature of data. Also, we run 5 independent chains and for each one the iteration step is 30,000.

Table 3.6: ABC Quantiles for Each Parameter

	2.50%	25%	50%	75%	97.50%
β	0.69828	0.82642	0.9063	0.9838	1.1181
σ	0.01692	0.11229	0.2172	0.2828	0.5765
ϕ	0.02464	0.07618	0.1918	0.5072	0.736

Table 3.6 shows the ABC quantiles for each parameters, and we use the 50% quantile as the estimates of unknown parameters. So we can obtain $\hat{\theta}_{ABC} = \{\beta = 0.9063, \sigma = 0.2172, \phi = 0.1918\}$. Table 3.7 reflects the ABC empirical mean and standard deviation for each parameter, as well as the standard error of the mean.

Table 3.7: ABC Empirical Mean and Standard Deviation for Each Parameter, Plus Standard Error of the Mean

	Mean	SD	Naive SE	Time-series SE
β	0.9062	0.1107	0.002212	0.01668
σ	0.2311	0.1522	0.003041	0.01853
ϕ	0.2972	0.2448	0.004892	0.02956

In addition, the diagnostic plots for the ABC algorithm is shown in Figure 3.4. We can see that the five chains diverge in different directions, which means the model is quite sensitive to the parameters. Kernel density estimates of the marginal posterior distributions are shown at right. These posterior distribution are plot by the posterior parameter estimates that obtained by ABC algorithm using R. The solid vertical line is the true parameters and the red dashed line is the ABC estimates of parameters.
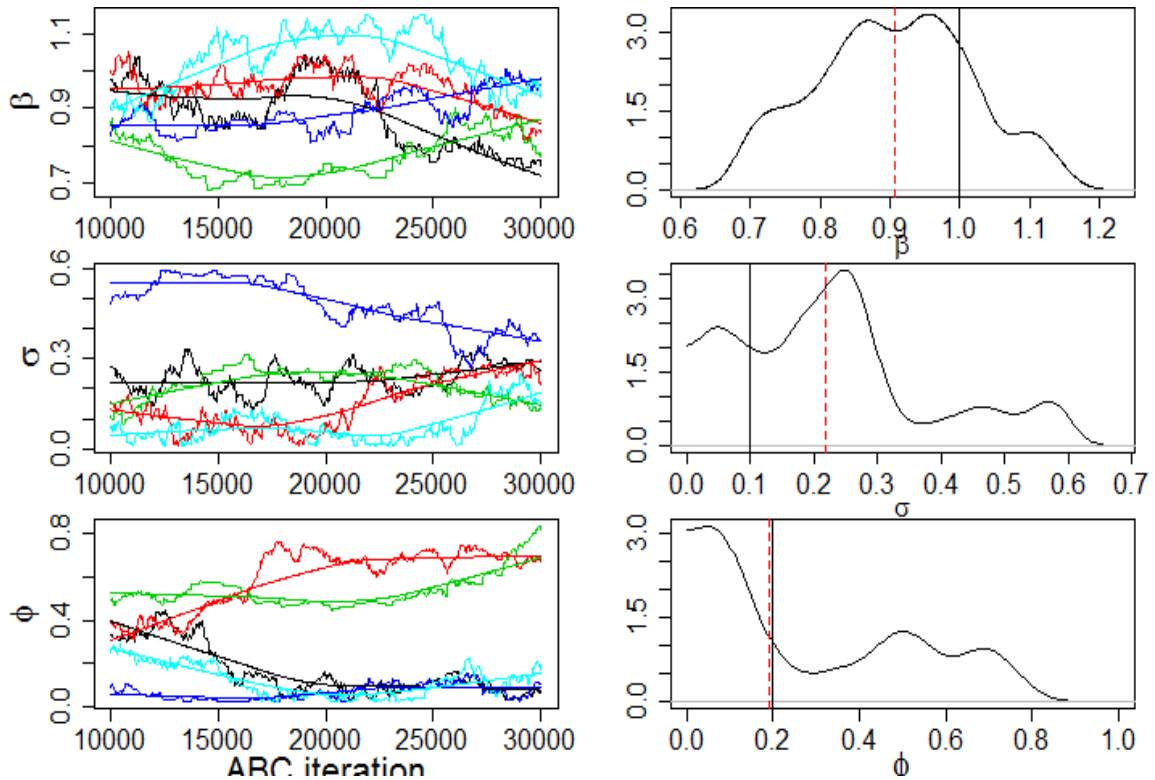

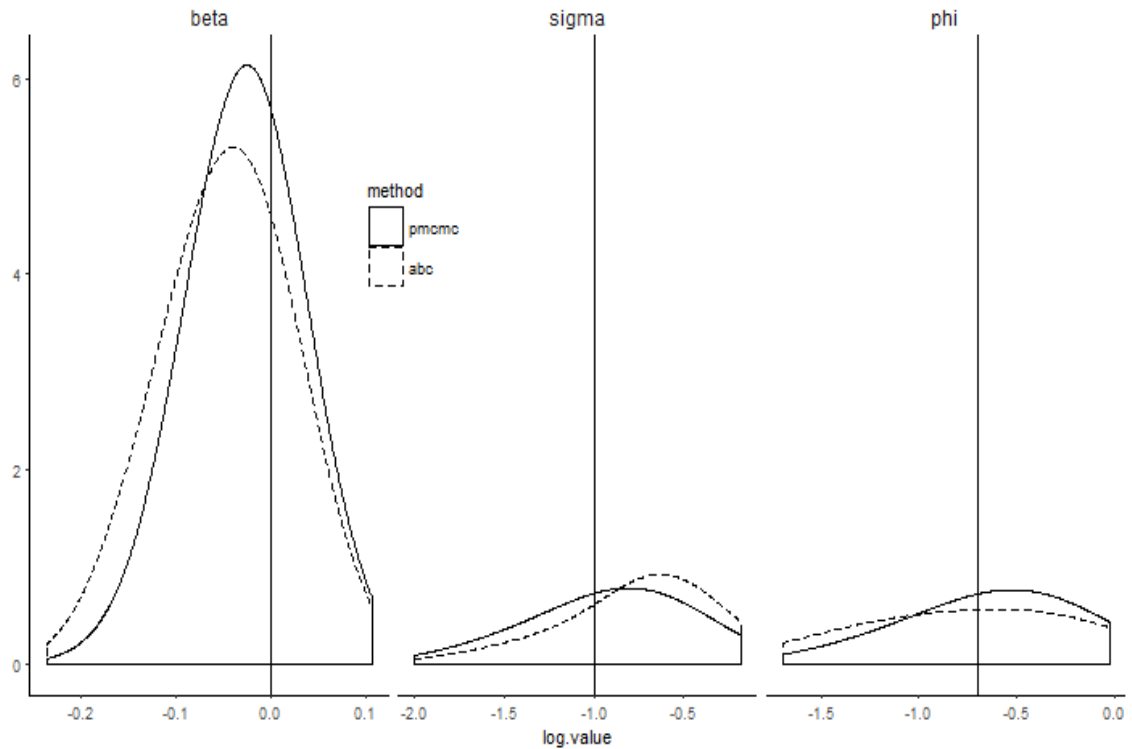Figure 3.4: The Diagnostic Plots for the ABC Algorithm

Figure 3.5: The Marginal Posterior Distributions of $\log_{10}$ Value of Parameters

In the end, we compare the statistical efficiency between ABC and PMCMC. We take the $\log_{10}$ value for all the posterior parameter estimates obtained by PMCMC and ABC. Then We plot their density functions into the same figure.

Figure 3.5 shows the marginal posterior distributions using full information via PMCMC (solid line) and partial information via ABC (dashed line). Kernel density estimates are illustrated for the posterior marginal densities of $\log_{10}(\beta)$, $\log_{10}(\sigma)$, and $\log_{10}(\phi)$, respectively. It reflects that ABC leads to somewhat broader posterior

distributions than the posteriors from PMCMC. In some ways, the reason may be straightforward. Since PMCMC use more information of the data than that of ABC, PMCMC then should have a more narrow and precise estimate than ABC.

Chapter 4

Application: The Analysis of Annual Forest Fire Counts in Canada

4.1 Background

Forest fire is a major environmental problem in Canada. It can cause catastrophic damages on natural resources and bring serious economic and social impacts. Each year, there are over thousands of forest fires around Canada, causing the destruction of large volumes of forest land. Nevertheless, forest fires have also benefits for the

health of the flora. For example, forest fires have been associated to the control of spread of beatle trees. According to the statistics from the Natural Resources Canada, about 7,588 forest fires have occurred each year over the last 25 years. Therefore, it is necessary to maintain an efficient wildland fire management in order to predict and manage risks and benefits.

A comprehensive study of forest fire activity would require the analysis of annual number of forest fires. Often a Poisson model has been employed for the number of fires, see Dayananda (1977), Mandallaz and Ye (1997). In this work, we consider the historical recorded data of forest fires as a time series of count data. We propose a nonlinear state space model to analyze the annual number of forest fires in Canada. And the statistical inference and estimation for the proposed model is then processed by the four methodologies that we have discussed in Chapter 2.

4.2   Data Description

The data we analyze consist of yearly total number of forest fires occurred in Canada from 1970 to 2014. This time series of forest fire counts is collected from the National Forestry Database (http://nfdp.ccfm.org/dynamic_report/dynamic_report_ui_e.php). We select the reporting agency as Canada, and the reporting item is the number of

fires in total. This dataset reflects the overall forest fire occurrences in Canada for each year. Table 4.1 shows the collected data.

Table 4.1: Total Number of Forest Fires

| Year	Total Fires	Year	Total Fires
1970	9250	1993	6043
1971	9167	1994	9763
1972	8232	1995	8486
1973	7593	1996	6349
1974	8129	1997	6148
1975	11178	1998	10723
1976	10236	1999	7633
1977	8945	2000	5349
1978	8028	2001	7753
1979	10051	2002	7861
1980	9138	2003	8230
1981	10095	2004	6680
1982	8942	2005	7542
1983	8935	2006	9820
1984	9220	2007	6917
1985	9354	2008	6278
1986	7320	2009	7210
1987	11301	2010	7291
1988	10741	2011	4743
1989	12185	2012	7956
1990	10111	2013	6264
1991	10327	2014	5152
1992	9068		
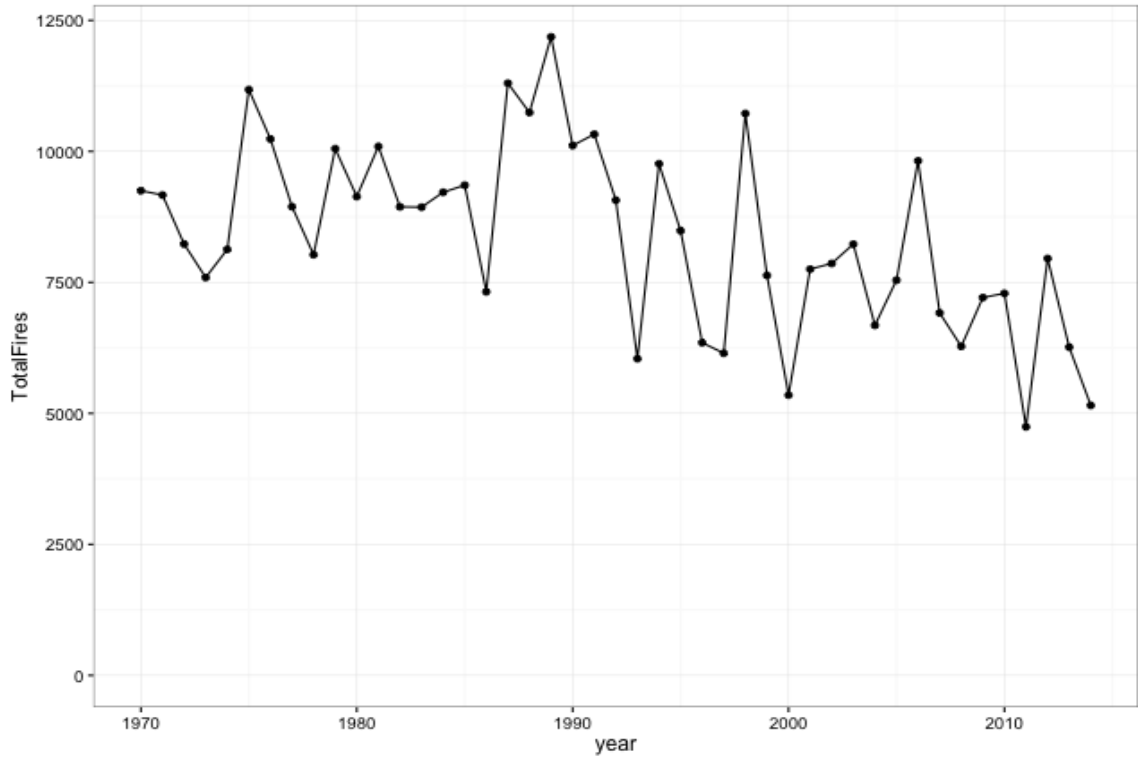
Figure 4.1: The Trend Plot of the Total Number of Forest Fires

The trend plot of this time series of forest fire counts can be illustrated in Figure 4.1 as above. We can see that in the recent 45 years, the minimum annual number of forest fires is 4,743, while the maximum number is 12,185. The general tendency of forest fires is declining though it oscillates each year.

Also, we can plot the histogram of forest fires counts, as shown in Figure 4.2. This figure shows that the distribution of the historical fires data is slightly left-skewed, with an average of 8,394 yearly occurrences.
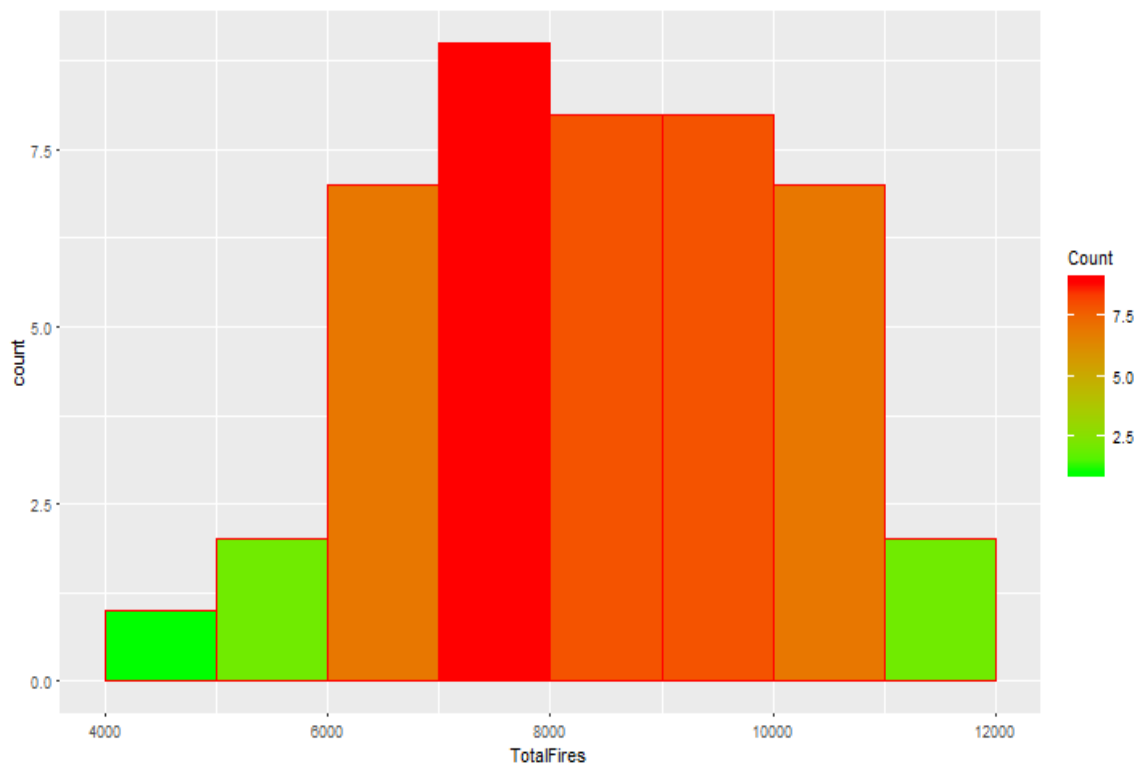
Figure 4.2: The Histogram of Forest Fire Counts

4.3 Model Setup

Now we propose a nonlinear state space model to practically simulate and replicate the real counts generating process of the forest fires. First of all, we consider the state process. Assuming it is a discrete time process model, let's denote $\{N_t, t = 0, 1, 2, \cdots\}$ as the state space. Then the state process can be set as below:

$$N_{t+1} = \frac{rN_t}{1 + \frac{N_t}{K}}\epsilon_t, \quad \epsilon_t \sim \mathcal{LN}(-\tfrac{1}{2}\sigma^2, \sigma^2),$$

where the unknown parameter vector is $\theta = \{r, K, \sigma\}$ and ϵ_t follow a log-normal distribution. This stochastic model is known as the Beverton-Holt model, which was introduced in the context of fisheries by Beverton & Holt (1957) . Despite it is a classic discrete time population model usually applied in ecology or epidemiology, it might still be able to depict the potential relationships between those transferable state spaces as a latent state system for the annual count of forest fires. Here r can be explained as the inherent growth rate, and K is assumed as a quasi-carrying capacity. We can define the state N_t as the true population size of forest fire counts at Year t.

For the measurement process, let us assume that the observation at time t follows a Poisson distribution with a parameter size N_t. That leads to the annual observed number of forest fires y_t follows:

$$y_t \sim \mathcal{P}(N_t).$$

4.4 Model Simulation

For those state spaces, we established the implied state-generating structure, but the parameter vector still remains unknown. The difficulties are how to estimate the latent unknown parameters. According to our observed numbers of forest fire, we can try a reasonable initial guess for the parameter vector $\theta = \{r, K, \sigma\}$, and plot the simulated states.

Let us assume the initial parameters are $\theta_0 = \{r = 1.4, K = 20000, \sigma = 0.15\}$, and the initial state value is $N_0 = 8000$. With the process model in place, we can simulate the state spaces and plot 10 state realizations with the initial parameters. It is shown in Figure 4.3. Since the measurement model we are considering follows a Poisson distribution with parameter N_t, it also reflects the simulated annual averages of forest fires.

Figure 4.3: 10 Simulated State Realizations Based on the Process Model with the Initial Parameters
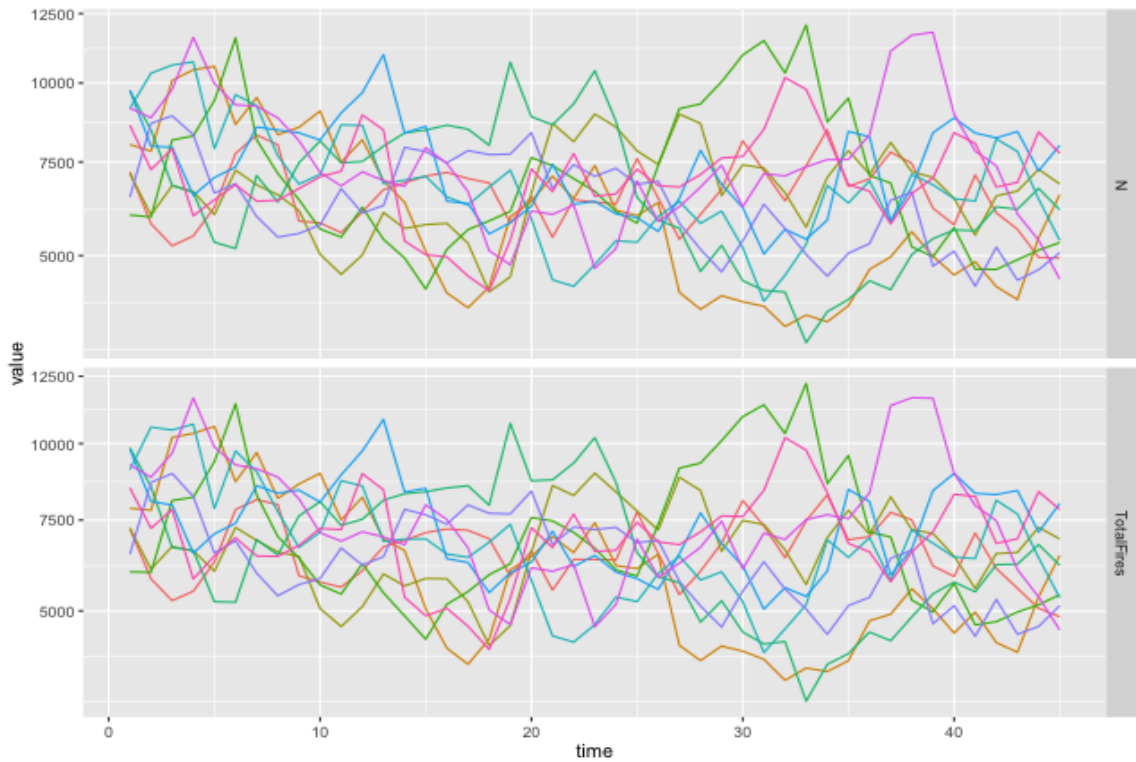
Figure 4.4: A Comparison Between Simulated State Realizations and Simulated Observation Realizations

With the measurement model, we can now simulate for the full nonlinear state space model. The result is shown in Figure 4.4. It is a comparison between the simulated states and the corresponding simulated observations. From the Figure 4.4, we find those trajectories are similar, which is quite straightforward. Another plot that shows the relationship between latent states and observed measurements is illustrated in Figure 4.5. All the points almost fall into a line, which reveals a property of the Poisson distribution in some ways.
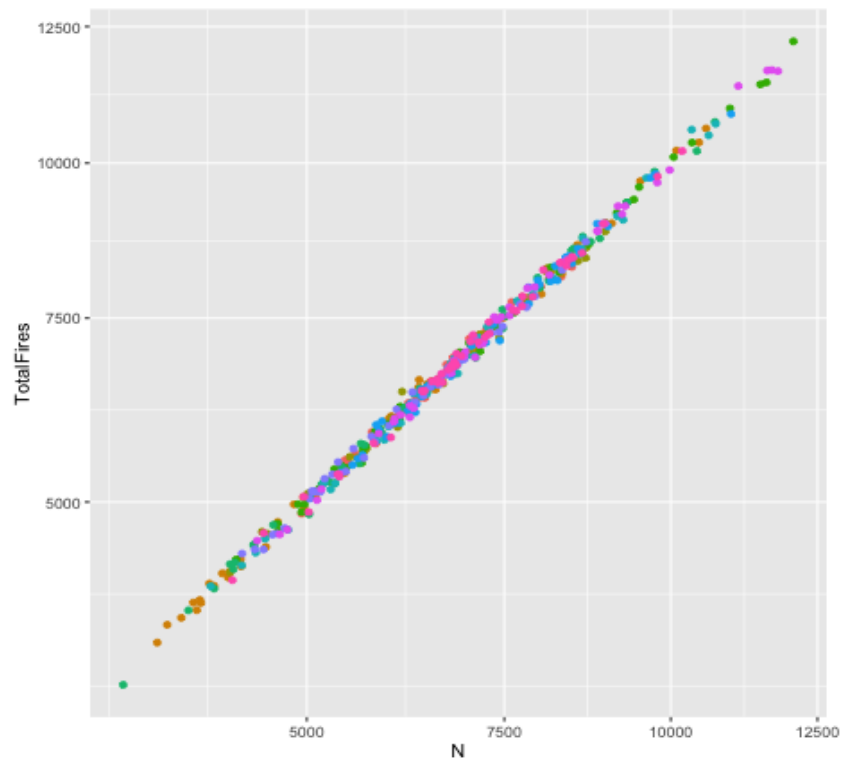
Figure 4.5: Simulated State Values N v.s. Simulated Observed Numbers of Fire
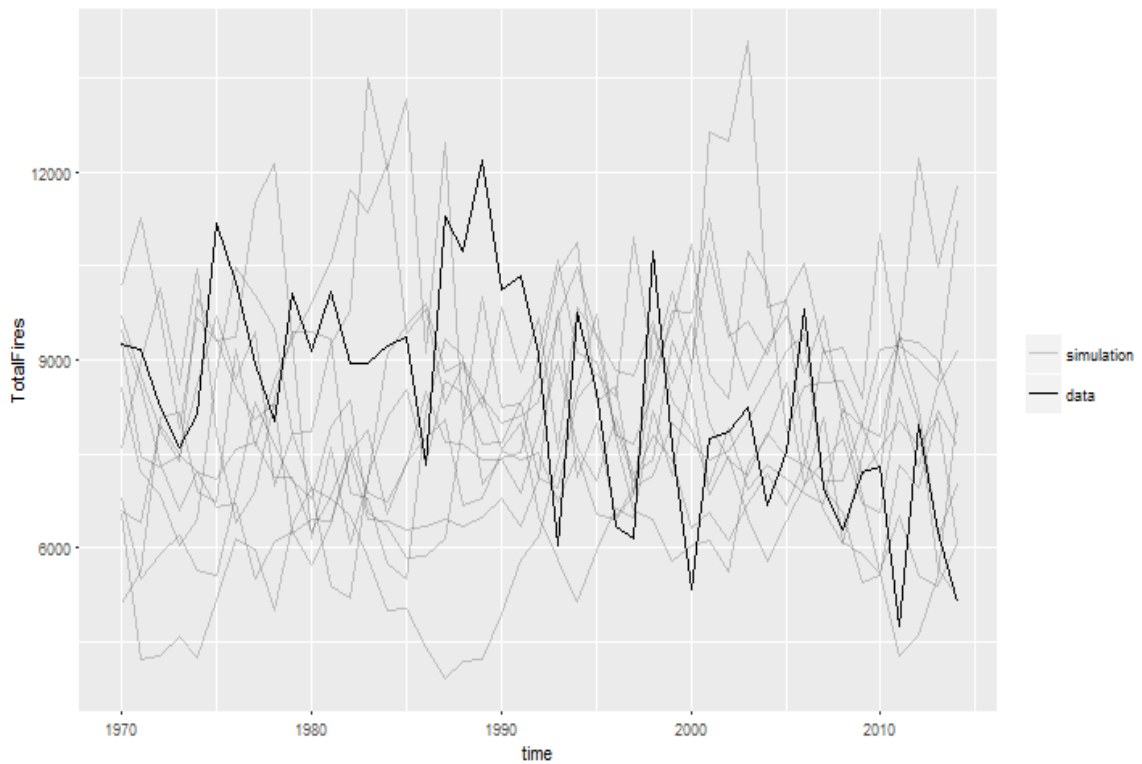
Figure 4.6: 10 Simulated Count Trajectories Based on Initial Parameters and the Actual Observed Number of Fires

To sum up, based on the nonlinear state space model that we have set up as well as the initial parameters that we have assigned, we can simulate the fitted model and compare it against the observed data - the total number of forest fires occurred in Canada. Figure 4.6 shows that our simulated trajectories cover the real data trajectory, and it exists certain similar shape. That means the initial parameters we assigned are plausible and the fitted model looks feasible and practical.

4.5   Model Estimation

Given the initial parameters, we can evaluate the likelihood of the forest fires data using the particle filter. Since the larger number of particles we use, the smaller Monte Carlo error but the greater computational burden we have, we decide to run 1000 particles to estimate the log likelihood at the initial guess of parameters $\theta_0 = \{r = 1.4, K = 20000, \sigma = 0.15\}$, and the initial state value is $N_0 = 8000$. We can obtain the log likelihood as -418.6927.

We will use the iterated filtering (Ionides et al. 2006, 2015) to obtain a maximum likelihood estimate for the real forest fires data. Since the parameters of the model $\theta = \{r, K, \sigma\}$ are constrained to be positive, we transform them to a scale on which they are unconstrained when estimating.

Table 4.2: Results of Estimating Parameters

	r	K	σ	$\hat{\ell}$	s.e.
Initial Guess	1.40	20000	0.150	-384.21	0.31
`mif` MLE	1.29	27900	0.172	-383.74	0.63

We replicate the iterated filtering search, and make a careful estimation of the log likelihood $\hat{\ell}$ as well as its standard error at each of the resulting point estimates. And then the parameter vector corresponding to the highest likelihood is chosen as the numerical approximation to the maximum likelihood estimates of parameters. The

resulting estimates are shown in Table 4.2. We can see the estimating parameters of the nonlinear state space model by maximum likelihood using iterated filtering are $\hat{\theta}_{MLE} = \{\hat{r} = 1.29, \hat{K} = 27900, \hat{\sigma} = 0.172\}$.
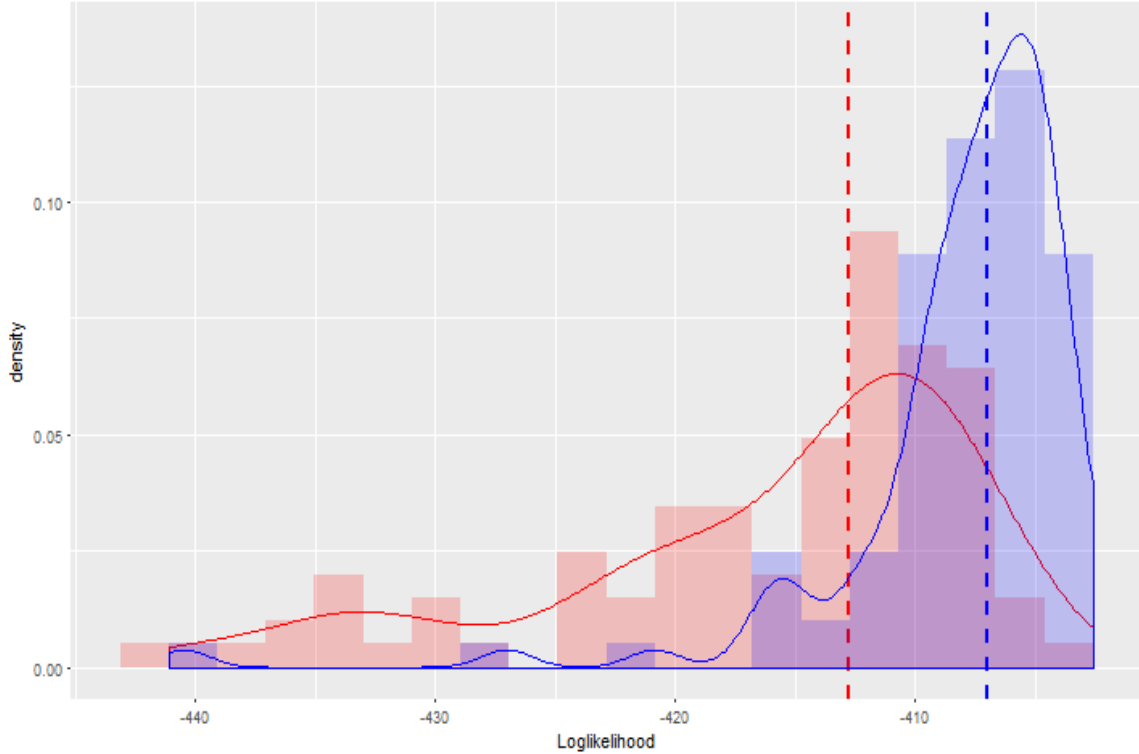

Figure 4.7: The Histogram of Log-likelihood at Different Parameter Values. Blue line represents the Maximum Likelihood Estimate, while pink line represents the initial guess.

We proceed to carry out 100 replicated particle filters at the initial guess and the MLE of parameters. A histogram shown as Figure 4.7 is plot to compare the calculated log likelihood at the two different points. We can see that the median

of the log likelihoods at the MLE is -406.9707, which is greater than that of log likelihoods at the initial guess, -412.7548. It means that the MLE is better.

Now we carry out a local search of the likelihood surface using the IF2 Algorithm. Given a model and a set of data, the likelihood surface is well defined. We set the starting point as $\theta_{start} = \{r = 1.1, K = 10000, \sigma = 0.05\}$ and a fixed initial state value 8000. Also, we set a perturbation size of 0.02 and the cooling type as geometric. Then running R codes, we can obtain the diagnostic plots shown as Figure 4.8.
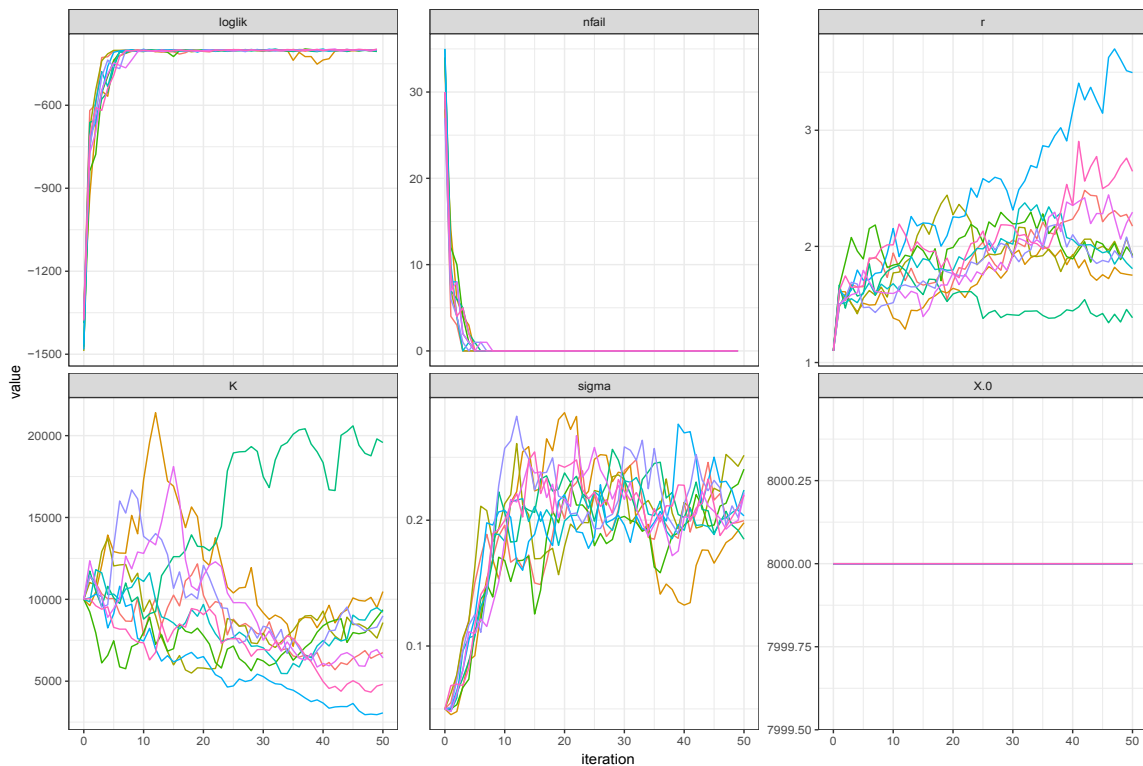

Figure 4.8: The Diagnostic Plots of the Local Search of the Likelihood Surface

In addition, the Figure 4.9 illustrates the geometry of the likelihood surface in a
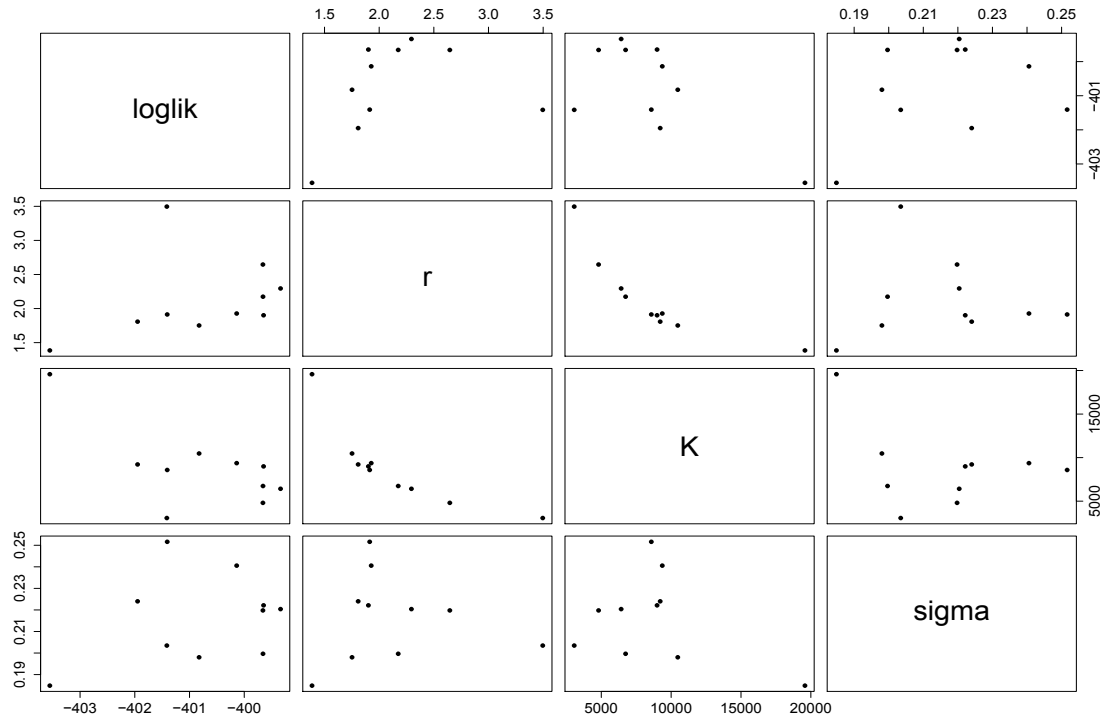neighborhood of the point estimate.

Figure 4.9: A Local Search of the Likelihood Surface

We can also carry out a global search by trying all remotely sensible parameter
vectors that are contained in a large box of parameter space. A starting values vector
is randomly drawn from the box. Then the result of the global search is shown in
Figure 4.10. The best result of this search had a log likelihood of -397.2959 with a
standard error of 0.6764758. A scatterplot is used to visualize the global geometry

of the likelihood surface. It is shown in Figure 4.11 that gray points are the starting

values and red points are the IF2 estimates. We conclude that optimization attempts

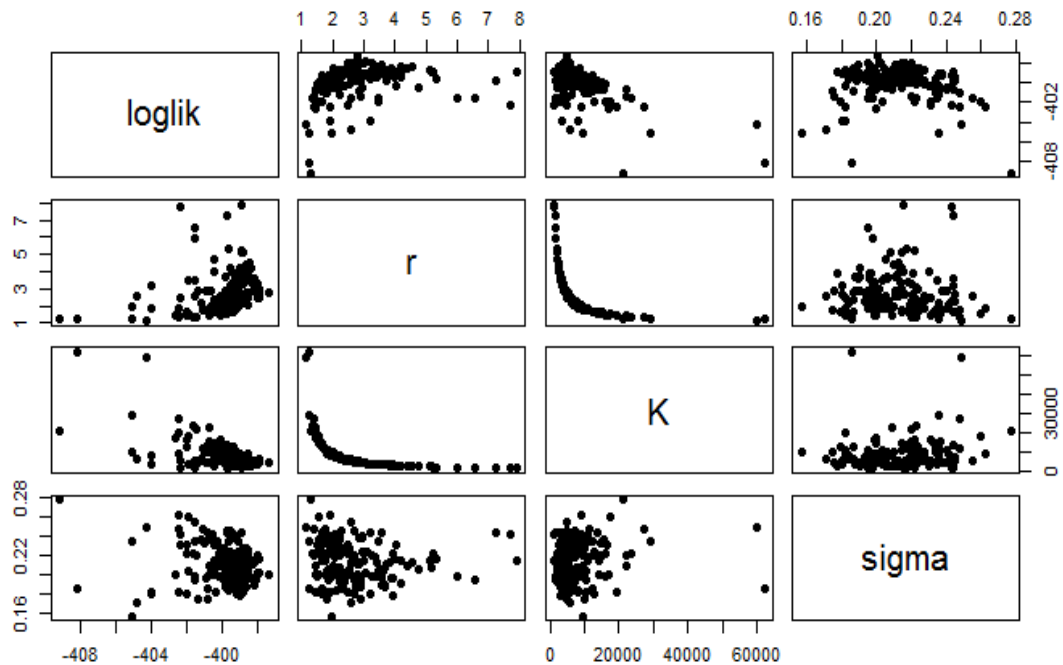from various starting points converge on a particular region in parameter space.

Figure 4.10: A Global Search of the Likelihood Surface

To get an idea of what the likelihood surface looks like corresponding to the

parameter r and K, we can evaluate the likelihood at a grid of points and visualize

the surface directly. In particular, all points with log likelihoods less than 50 units
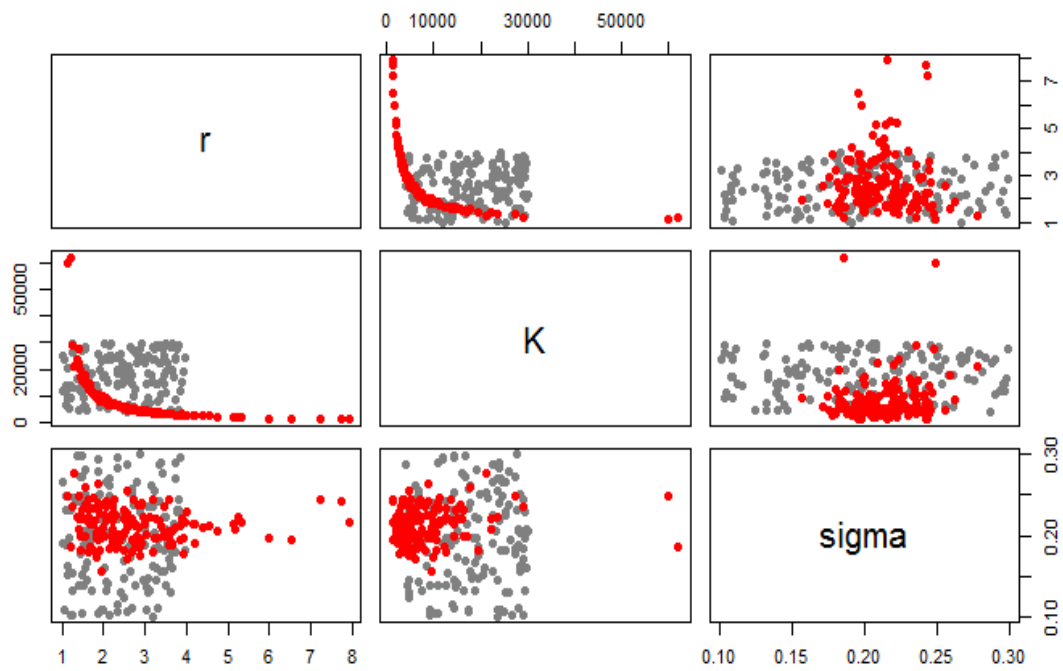
below the maximum are shown in gray in Figure 4.12.

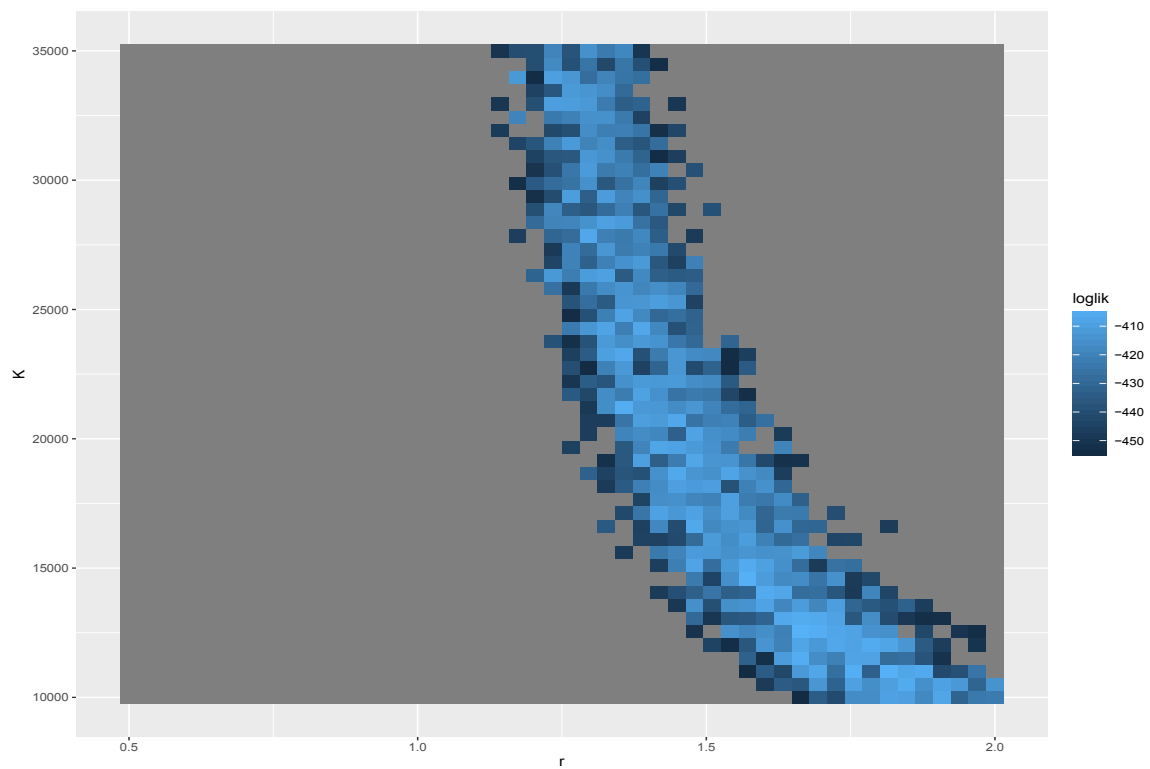Figure 4.11: A Scatterplot with Starting Values and IF2 Estimates

Figure 4.12: Log-likelihood Surface Corresponding to r and K

Now we specify a prior distribution on unknown parameters and carry out Bayesian inference. The PMCMC algorithm is applied to draw a sample from the posterior. Table 4.3 shows the PMCMC quantiles for each parameters, and we use the 50% quantile as the estimates of unknown parameters. So we can obtain $\hat{\theta}_{PMCMC} = \{r = 1.298, K = 27943, \sigma = 0.174\}$. The diagnostic plots for the PMCMC algorithm is shown in Figure 4.13. The trace plots in the left column show the evolution of 5 independent MCMC chains from iterations 10001 to 30001. Kernel density estimates of the marginal posterior distributions are shown at right. The solid vertical line is the initial guess of parameters and the red dashed line is the PMCMC estimates of parameters.

Table 4.3: PMCMC Quantiles for Each Variable

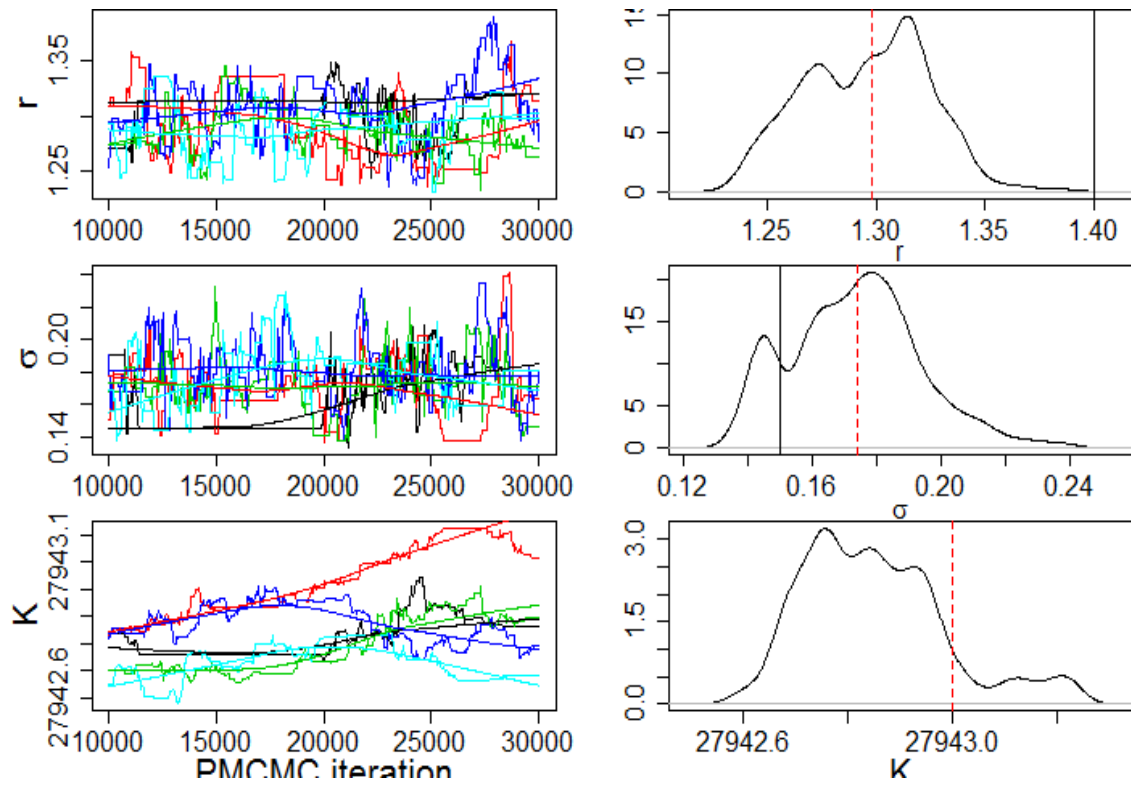|	2.50%	25%	50%	75%	97.50%
r	1.2430	1.2740	1.2980	1.3150	1.3460
σ	0.1386	0.1593	0.1740	0.1854	0.2162
K	27940.00	27940.00	27942.83	27940.00	27940.00

Figure 4.13: Diagnostic Plots for the PMCMC Algorithm

Also, we apply the ABC algorithm to obtain the evaluation of unknown parameters. Table 4.4 shows the ABC quantiles for each parameters, and we use the 50% quantile as the estimates of unknown parameters. So we can obtain $\hat{\theta}_{ABC} = \{r = 1.397, K = 19050, \sigma = 0.1582\}$. The diagnostic plots for the ABC algorithm is shown in Figure 4.14. Kernel density estimates of the marginal posterior distributions are shown at right. The solid vertical line is the initial guess of parameters and the red dashed line is the ABC estimates of parameters.

Table 4.4: ABC Quantiles for Each Variable

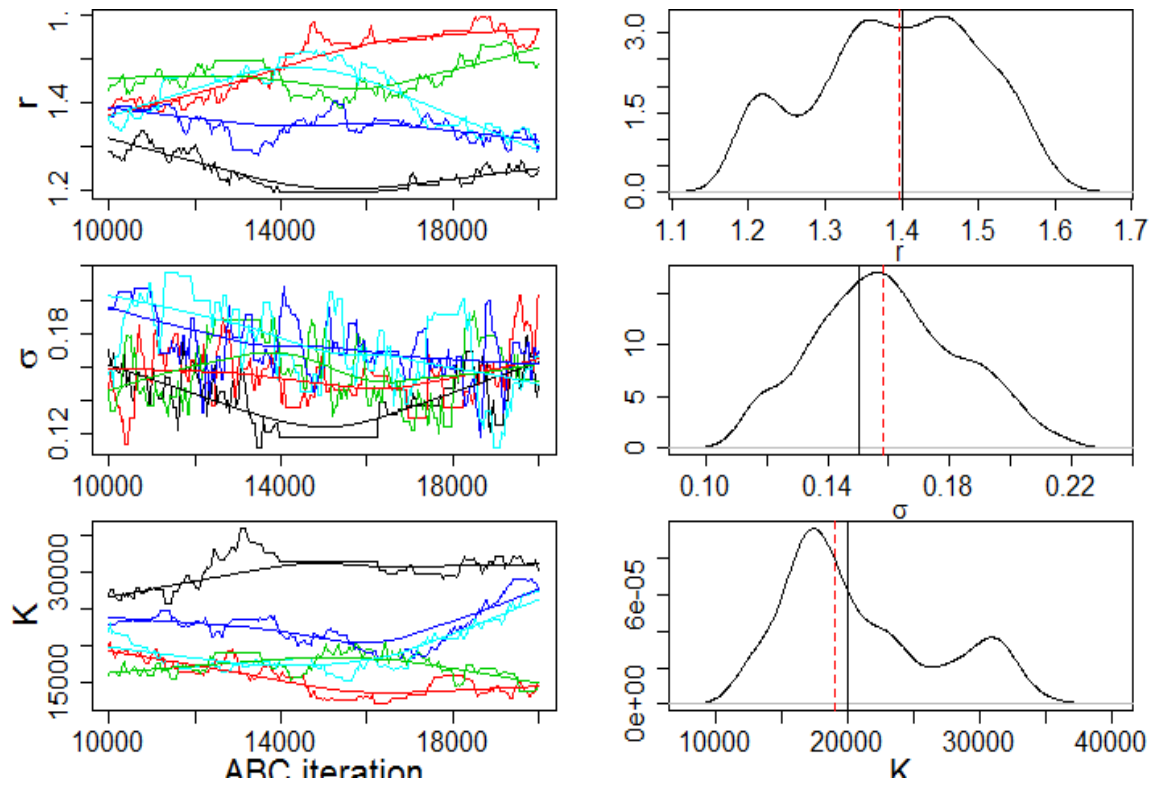|	2.50%	25%	50%	75%	97.50%
r	1.1980	1.3230	1.3970	1.4720	1.5660
σ	0.1175	0.1432	0.1582	0.1756	0.2046
K	12710.00	16740.00	19050.00	24240.00	32280.00

Figure 4.14: Diagnostic Plots for the ABC Algorithm

Chapter 5

Summary and Future Work

In this thesis, we mainly proposed a nonlinear state space model to analyze the annual forest fires occurred in Canada. This proposed model is different from the traditional Poisson regression models or Logistic models that are usually applied in the fires occurrence analysis. It provides a flexibility and reliability to the analysis of the real forest fires dataset. The difficult part of the proposed model is the statistical inference and estimation for the unknown parameters. To solve the problems, we used four numerical methodologies and algorithms, which have been proved to be feasible and practical in the literature. They are particle filter (also named sequential Monte Carlo), iterated filtering (IF), particle Markov chain Monte Carlo (PMCMC), and approximate Bayesian computation (ABC), respectively. All these methods have

plug-and-play property, which can be easily programmed and run in the R statistical software using the package POMP.

The paper first provides a brief introduction to the nonlinear state space model. Then, it takes a focus on the methodologies applied to fit the model and obtain the parameter estimates. Also, a simulation study is carried out to analyze a series of simulated counts using the methods. At the end, an analysis of the annual forest fires using the nonlinear state space model is presented. The unknown parameters of the model are estimated by using the proposed numerical methods with satisfactory outcomes.

For future work, we can add a parameter for the measurement process based on our original model. That means the new measurement process would be $\mathcal{P}(\phi N_t)$. Of course, an additional unknown parameter will bring larger computational costs to our analysis. Also, we may assume the measurement process follows a Negative Binomial distribution, considering the over-dispersion situation. Moreover, one may also be interested in the role of a vector-valued covariate process in explaining the forest fires data. Then modeling and inference conditional on covariates can be carried out in the further work as well.

Bibliography

[1] Andrieu C, Doucet A, Holenstein R (2010). Particle Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society B*, **72**(3), 269-342.

[2] Arulampalam MS, Maskell S, Gordon N & Clapp T (2002). A tutorial on particle filters for online nonlinear, non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process*, **50**, 174-188.

[3] Beaumont MA (2010). Approximate Bayesian Computation in Evolution and Ecology. *Annual Review of Ecology, Evolution, and Systematics*, **41**, 379-406.

[4] Bhadra A (2010). Discussion of Particle Markov Chain Monte Carlo Methods by C. Andrieu, A. Doucet and R. Holenstein. *Journal of the Royal Statistical Society B*, **72**, 314-315.

[5] Bretó C, He D, Ionides EL & King A A (2009). Time series analysis via mechanistic models. *Ann. Appl. Stat.*, **3**, 319-348.

[6] Cappé O, Godsill S, Moulines E (2007). An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. *Proceedings of the IEEE*, **95**(5), 899-924.

[7] Dayananda PWA (1977). Stochastic models for forest fires. *Ecological Modelling*, **3**:309-313.

[8] Doucet A, de Freita N & Gordon N, (2001). Sequential Monte Carlo methods in practice. New York, *NY: Springer*.

[9] He D, Ionides EL, King AA (2010). Plug-and-play Inference for Disease Dynamics: Measles in Large and Small Towns as a Case Study. *Journal of the Royal Society Interface*, **7**(43), 271-283.

[10] Hull J (2009). Options, Futures, and other Derivatives. *Pearson*, 7 edition.

[11] Ionides EL, Bhadra A, Atchadé Y, King AA (2011). Iterated Filtering. *The Annals of Statistics*, **39**(3), 1776-1802.

[12] Ionides EL, Bretó C, King AA (2006). Inference for Nonlinear Dynamical Systems. *Proceedings of the National Academy of Sciences of the USA*, **103**(49), 18438-18443.

[13] Ionides EL, Nguyen D, Atchadé Y, Stoev S, King AA. (2015). Inference for dynamic and latent variable models via iterated, perturbed Bayes maps. *Proc. Natl Acad. Sci. USA* **112**, 719-724.

[14] Kalman RE (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, **82**(1):35-45.

[15] Kantas N, Doucet A, Singh SS, Maciejowski JM (2015). On Particle Methods for Parameter Estimation in State-Space Models.

[16] Keeling MJ and Rohani P (2008). Modeling infectious diseases in humans and animals. *Princeton University Press*

[17] King AA, Ionides EL, Nguyen D (2016). Statistical inference for partially observed markov processes via the R package **pomp**. *Journal of Statistical Software*, **69**,1-43.

[18] Langrock R (2011). Some applications of nonlinear and non-Gaussian statespace mod- elling by means of hidden Markov models. *Journal of Applied Statistics*, **38**(12): 2955-2970.

[19] Liu J, West M (2001). Combining Parameter and State Estimation in Simulation-Based Filtering. In A Doucet, N de Freitas, NJ Gordon (eds.), *Sequential Monte Carlo Methods in Practice*, pp. 197-224. Springer-Verlag, New York.

[20] Ljung L (1999). System identification: theory for the user. *Prentice Hall*.

[21] Mandallaz D , Ye R (1997). Prediction of forest fires with Poisson models. *Canadian Journal of Forest Research*, **27**:1685-1694.

[22] Marin JM, Pudlo P, Robert CP, Ryder RJ (2012). Approximate Bayesian Computational Methods. *Statistics and Computing*, **22**(6), 1167-1180.

[23] Marjoram P, Molitor J, Plagnol V & Tavare S (2003). Markov chain Monte Carlo without likelihoods. *Proc. Natl Acad. Sci. USA* **100**, 15 324-15 328.

[24] Poyiadjis G, Singh S & Doucet A (2006). Gradient-free maximum likelihood parameter estimation with particle filters. *In Proc. Am. Control Conf.* pp. 3062 - 3067.

[25] Pritchard J, Seielstad MT, Perez-Lezaun A & Feldman MW (1999). Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Mol. Biol. Evol.* **16**: 1791-1798.

[26] Revolution Analytics, Weston S (2014). foreach: Foreach Looping Construct for R. R package version 1.4.2

[27] Shumway RH & Stoffer DS (2006). Time series analysis and its applications, 2nd edn, New York, *NY: Springer.*

[28] Sisson SA, Fan Y & Tanaka MM (2007). Sequential Monte Carlo without likelihoods. *Proc. Natl Acad. Sci. USA* **104**, 1760-1765.

[29] Toni T, Welch D, Strelkowa N, Ipsen A, Stumpf MP (2009). Approximate Bayesian Computation Scheme for Parameter Inference and Model Selection in Dynamical Systems. *Journal of the Royal Society Interface*, **6**(31), 187-202.

[30] Tsay RS (2005). Analysis of financial time series. *John Wiley & Sons*, 2 edition.

[31] Wan E, Van Der Merwe R (2000). The Unscented Kalman Filter for Nonlinear Estimation. *Adaptive Systems for Signal Processing, Communications, and Control*, pp. 153-158.

[32] Wood SN (2010). Statistical Inference for Noisy Nonlinear Ecological Dynamic Systems. *Nature*, **466**(7310), 1102-1104.

[33] Wilkinson DJ (2011). Stochastic modelling for systems biology. *CRC press*, 2 edition.

[34] Zeger SL (1988). A regression model for time series of counts. *Biometrika*, **75**(4): 621-629.