Particle Methods in Nonlinear State Space Models

CHENG, Jing

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy
in
Statistics

The Chinese University of Hong Kong
July 2015

Thesis Assessment Committee

Professor FAN Xiaodan (Chair)

Professor CHAN Ngai Hang (Thesis Supervisor)

Professor YAU Chun Yip (Committee Member)

Professor Stoffer David S. (External Examiner)

Abstract

Particle methods, also known as Sequential Monte Carlo methods, represent a general class of algorithms that approximate a sequence of distributions of interest by a number of weighted particles. Particle methods are applied to study nonlinear state space (NLSS) models for classical and Bayesian frameworks. More specifically, the particle Monte Carlo Expectation Maximization (MCEM) algorithm is introduced to conduct inference for NLSS models. The particle MCEM algorithm, which constitutes the surrogate to approximate the expectation in expectation step with weighted particles, is a special variant of the MCEM algorithm and is tailored to inference for NLSS models.

In the implementation of the particle MCEM algorithm, choosing a precise sample size at each iteration is an important issue. Renewal theory and the central limit theorems of particle filters and smoothers are introduced to construct confidence regions of the corresponding variables to deal with this issue. Based on the confidence regions, an automated selection criterion is proposed to boost the sample size as the estimation at any iteration is approaching the one at the previous iteration.

To illustrate, the proposed methodology is applied to handle the stochastic volatility model, which is a special case of NLSS models. Finally, the simulation study and empirical analysis using this new approach are also presented.

摘要

粒子方法，也称为序贯蒙特卡洛方法，是一类算法的通称。这类算法利用大量加权粒子来近似模拟研究者所关心的一系列分布。在经典统计以及贝叶斯统计框架下，本文分别分析了将粒子方法引入到非线性状态空间模型的过程。更具体地说，本文利用粒子蒙特卡洛期望最大化算法对非线性状态空间模型进行统计推断。粒子蒙特卡洛期望最大化算法是蒙特卡洛期望最大化算法的一个变体。这个方法利用加权粒子所计算得到的统计量来近似期望最大化算法中的期望值，因而十分适用于对非线性状态空间模型的推断分析。在应用粒子蒙特卡洛期望最大化算法时，如何在每一步迭代中更加精准地选取样本容量是一个很重要的问题。为了解决这一问题，本文利用粒子滤波以及粒子平滑的中心极限定理和更新理论，构建出相关变量的置信区域，并在此基础上提出了自动判定准则：当某两步迭代的估计值十分靠近时，则在下一步迭代时增加样本量。 由于随机波动模型是非线性状态空间的一个特例，本文将所提出的理论和方法运用到对随机波动模型的推断分析中。此外，本文也对这个新方法进行了模拟研究以及实证分析研究。

Acknowledgments

This template is supported by The Teaching Development Unit of The Chinese University of Hong Kong.

Sincere and special thanks are given to my supervisor Professor Chan, who gives me great assistance in my research and study. Here, I also thank my family for their understanding and supporting, especially my husband who encouraged and helped me whenever I met with dilemma in these four years pursuing my PhD's degree. Finally, I am particularly grateful to all the people I met and things happened in the Chinese University of Hong Kong.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

This thesis focuses on the inference problem of nonlinear state space (NLSS) models. Due to the non-Gaussian and nonlinear properties of these models, the Kalman filter may be difficult to handle them. Particle methods have been developed to solve this issue.

In this thesis, both maximum likelihood estimation and Bayesian analysis for NLSS models are studied. For maximum likelihood estimation, the Monte Carlo Expectation Maximization (MCEM) algorithm is used to get the estimators, and the corresponding Monte Carlo sample is generated using a newly-developed particle method (i.e., the particle Monte Carlo Marko chain method). For Bayesian analysis, type II maximum likelihood prior is introduced to obtain robust Bayesian estimators, and the MCEM algorithm is also used to obtain the estimators. The particle Monte Carlo Marko chain method and the resample-move algorithm are used to generate the Monte Carlo sample in this framework.

Furthermore, several issues in the implementation of MCEM algorithm are worth investigating. Choosing an appropriate sample size for each iteration is a major concern because this may determine the balance between estimation accuracy and computational cost. An attempt is made in this thesis to find a solution to this problem.

1.2 State Space Models

The state space models are widely used for statistical applications, such as in time series modeling and signal processing. Consider a sequence of Markov chain $\{X_t; t \in \mathbb{N}\}$ with initial distribution $X_0 \sim \xi(\cdot)$ and transition kernel $(X_t \mid X_{t-1} = x_{t-1}) \sim f(\cdot \mid x_{t-1})$, which is called a state process. The observed values $\{Y_t; t \in \mathbb{N}\}$ in the observation process are independently conditional on the hidden states $\{X_t; t \in \mathbb{N}\}$ and the conditional distribution of Y_t only depends on the corresponding state X_t. In other words, $(Y_t \mid X_0 = x_0, \ldots, X_t = x_t) \sim g(\cdot \mid x_t)$. These two processes compose a NLSS model or a Hidden Markov model. For simplicity, denote $x_{0:t} \triangleq \{x_0, \ldots, x_t\}$ and $y_{0:t} \triangleq \{y_0, \ldots, y_t\}$.

A special class of the state space models is the linear Gaussian state space model. This type of model has a hierarchical structure with two equations: the state equation

$$X_t = \Phi X_{t-1} + AU_t + Z_t, \tag{1.1}$$

and the observation equation

$$Y_t = \Theta X_t + BU_t + V_t, \tag{1.2}$$

where X_t is a sequence of $p \times 1$ random vectors for $t = 0, \ldots, T$ with initial state X_0 following a Gaussian distribution, Φ is a $p \times p$ unknown matrix, U_t is a $r \times 1$ matrix of exogenous variables, Y_t is a collection of $q \times 1$ random vectors of observed variables for $t = 0, \ldots, T$, Θ is a $q \times p$ unknown measurement matrix, A and B are $p \times r$ and $q \times r$ unknown regression matrices, and Z_t and V_t are $p \times 1$ and $q \times 1$ independent and identically distributed normal vectors with zero mean and covariance matrices Q and R respectively.

When the assumptions of linearity and Gaussianity are removed, the model becomes a NLSS model, which is the model discussed in this thesis. Generally, inference for state space models entails the following posterior distributions.

Definition 1.1 For a collection of observations $y_{0:T}$ and state variables $x_{0:T}$,
joint smoothing distribution: $\phi_{0:t|t}(x_{0:t}) \triangleq p(x_{0:t} \mid y_{0:t})$ for $0 \leq t \leq T$;
marginal smoothing distribution: $\phi_{s|t}(x_s) \triangleq p(x_s \mid y_{0:t})$ for $0 \leq t \leq T$ and $s < t$;
filtering distribution: $\phi_{t|t}(x_t) \triangleq p(x_t \mid y_{0:t})$ for $0 \leq t \leq T$;
prediction distribution: $\phi_{t+p|t}(x_{t+p}) \triangleq p(x_{t+p} \mid y_{0:t})$ for $0 \leq t \leq T$ and $p > 0$.

The joint smoothing distribution can be written as

$$\phi_{0:t|t}(x_{0:t}) \propto \xi(x_0)g(y_0 \mid x_0)\prod_{i=1}^{t} f(x_i \mid x_{i-1})g(y_i \mid x_i), \tag{1.3}$$

and the recursion is obtained using Bayes' theorem

$$\phi_{0:t+1|t+1}(x_{0:t+1}) = \phi_{0:t|t}(x_{0:t})\frac{f(x_{t+1} \mid x_t)g(y_{t+1} \mid x_{t+1})}{p(y_{t+1} \mid y_{0:t})}, \tag{1.4}$$

where

$$p(y_{t+1} \mid y_{0:t}) = \int \phi_{0:t|t}(x_{0:t})f(x_{t+1} \mid x_t)g(y_{t+1} \mid x_{t+1})\,\mathrm{d}x_{0:t+1}.$$

It can be checked that the filtering distribution and the prediction distribution are recursively related by integrating out $x_{0:t}$, and the relationship is formulated as follows

$$\phi_{t+1|t+1}(x_{t+1}) = \frac{g(y_{t+1} \mid x_{t+1})\phi_{t+1|t}(x_{t+1})}{p(y_{t+1} \mid y_{0:t})}, \tag{1.5}$$

and

$$\phi_{t+1|t}(x_{t+1}) = \int \phi_{t|t}(x_t)f(x_{t+1} \mid x_t)\,\mathrm{d}x_t. \tag{1.6}$$

The Kalman filter handles the inference for linear Gaussian state space models successfully, but not typically for the NLSS models. The above formulas show that all these distributions the common element $p(y_{t+1} \mid y_{0:t})$. However this density function does not have a closed form in NLSS models because it involves the calculation of complex and high-dimensional integrals. Therefore, the approximations of these distributions are required to conduct inference for NLSS models, and an effective way to make approximation is to use particle methods, which is the primary topic of this thesis.

1.3 Thesis Structure

This thesis is organized as follows. Chapter 2 reviews the expetation maximization (EM) algorithm, particle filters and particle smoothers, which are the primary statistical tools used in this thesis. In Chapter 3, inference for NLSS models according to maximum likelihood estimation and Bayesian methods are studied. To obtain the maximum likelihood estimation, the MCEM algorithm is used, where the particle Markov chain Monte Carlo (PMCMC) method is used to generate the Monte Carlo sample. For the Bayesian framework, type II maximum likelihood prior is introduced and the MCEM algorithm is used to deal with the corresponding inference. In Chapter 4, the method developed in Chapter 3 is applied to the stochastic volatility model, a special case of the NLSS model, and the corresponding simulation and empirical study is presented in Chapter 5. Finally, the results of this thesis and the future researches are presented in Chapter 6.

1.4 Contributions

NLSS models have been popular with the researchers in the previous two decades, and there are numerous methods have been developed to infer these models. The EM algorithm with particle methods is an effective tool when performing maximum likelihood estimation. Although the EM algorithm with forward filtering backward simulation (FFBSi), which is one of types of the particle smoothers, is discussed in Kim and Stoffer (2008), FFBSi experiences particle degeneracy as time increases, resulting in a poor estimator based on a small effective sample size of particles. Therefore, the EM algorithm is combined with the particle Monte Carlo Markov chain method, which is developed for particle smoothers to manage maximum likelihood estimation more effectively.

A Bayesian estimator is also considered in this thesis. To obtain robust results, type II maximum likelihood prior which is an empirical Bayes approach, is introduced. The EM algorithm along with particle methods are utilized to manage this problem, and the resample-move algorithm and the PMCMC method are chosen as the corresponding particle methods, respectively.

Another critical problem addressed in this thesis is the design of an automated rule to determine whether the Monte Carlo sample size should be increased at each iteration. Although Booth and Robert (1999) and Levine and Casella (2001) considered similar rules in the implementation of the MCEM algorithm, no discussion was presented regarding the particle MCEM algorithm, which is different from the traditional MCEM algorithm. The central limit theorem in Chopin (2004) causes the rule for sample size in Booth and Robert (1999)to be effective when the MCEM with resample-move algorithm is applied. For the MCEM with PMCMC algorithm, renewal theory is used to construct confidence intervals of relative likelihood, and thus produce the automated rule for determining whether the sample size should be increased at each iteration.

Chapter 2

Literature Review

In this chapter, several critical methods and papers regarding the EM algorithm and particle methods are reviewed. In Section 2.1, the EM algorithm is introduced, in which the MCEM algorithm is a modification of the EM algorithm. Several key algorithms of particle filters and particle smoothers are presented in Section 2.2 and Section 2.3 respectively.

2.1 The Expectation Maximization Algorithm

The EM algorithm was first proposed in Dempster et al. (1977). It is a widely applicable tool for iteratively computing the maximum likelihood estimators , particularly for incomplete data. The EM algorithm replaces the intractable likelihood maximization problem of incomplete data with a sequence of maximization problems of complete data that are easier to handle. Each element in the sequence represents an iteration in the EM algorithm, and the Expectation step (E-step) and the maximization step (M-step) comprise each iteration.

Consider a model with observed variables $\mathbf{y} = (y_1, \ldots, y_n)$, hidden variables (or missing data) $\mathbf{x} = (x_1, \ldots, x_n)$ and parameter θ. The densities $g(\cdot \mid \theta)$ of $\mathbf{y}$ and $f(\cdot \mid \theta)$ of $(\mathbf{y}, \mathbf{x})$ have the relationship

$$g(\mathbf{y} \mid \theta) = \int f(\mathbf{y}, \mathbf{x} \mid \theta)\, \mathrm{d}x. \tag{2.1}$$

Therefore, the logarithm of this likelihood function is

$$L(\theta \mid \mathbf{y}) = \log g(\mathbf{y} \mid \theta) = \log \int f(\mathbf{y},\ \ \mathbf{x} \mid \theta)\, \mathrm{d}x. \tag{2.2}$$

The target is to obtain the Maximum Likelihood Estimator (MLE) of the parameter, defined as $\theta_{ML} \equiv \arg\max_{\theta} L(\theta \mid \mathbf{y})$. However, $g(\mathbf{y} \mid \theta)$ has no closed form in many cases, and the EM algorithm provides a useful approach to solve this type of problem. Assume that a sequence $\{\theta^{(r)}\}$ has been created from an initial value $\theta^{(0)}$, then $\theta^{(r+1)}$ can be acquired according to the following two steps:

E-Step. Calculate the expected log-likelihood function $Q(\theta, \theta^{(r+1)})$, where

$$Q(\theta, \theta^{(r)}) = E[\log f(\mathbf{y},\ \mathbf{x} \mid \theta) \mid \theta^{(r)}, \mathbf{y}]; \tag{2.3}$$

M-Step. Find the value $\theta^{(r+1)}$ that maximizes $Q(\theta, \theta^{(r)})$, meaning

$$\theta^{(r+1)} = \arg\max_{\theta} E[\log f(\mathbf{y},\ \mathbf{x} \mid \theta) \mid \theta^{(r)}, \mathbf{y}]. \tag{2.4}$$

Dempster et al. (1977) proves the properties of EM algorithm, and the primary property is the monotonicity

$$L(\theta^{(r+1)} \mid \mathbf{y}) \geq L(\theta^{(r)} \mid \mathbf{y}). \tag{2.5}$$

In Wu (1983), it is shown that the sequence $(\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(r)})$ converges to the MLE θ_{ML} as $r \to \infty$ under certain regularity conditions.

2.1.1 Monte Carlo Expectation Maximization Algorithm

When the EM algorithm is applied to complicated models, a dilemma arises. The E-step typically cannot be implemented because it needs to compute a complex or high-dimensional expectation. Therefore, the Monte Carlo EM (MCEM) algorithm was developed to solve this problem. The MCEM algorithm, as a modification of EM algorithm, was first introduced in Wei and Tanner (1990). In this algorithm, a Monte Carlo sampling scheme is used to estimate the expectation in E-step. Similar to an EM algorithm that has a Monte Carlo E-step (MCE-step) instead of an E-step, the MCEM algorithm also has two steps in each iteration. Assume that $\theta^{(r)}$ has been given at the rth iteration, then at the $(r+1)$th iteration, execute the following steps:

MCE-Step. Estimate the expected log-likelihood function $Q_M(\theta, \theta^{(r)})$,

$$Q_M(\theta, \theta^{(r)}) = \frac{1}{M} \sum_{j=1}^{M} \log f(\mathbf{y}, \mathbf{x}_j^* \mid \theta), \tag{2.6}$$

where $\mathbf{x}_j^*$ is a random sample from $f(\mathbf{x} \mid \theta^{(r)}, \mathbf{y})$;
M-Step. Find the value $\theta^{(r+1)}$ that maximizes $Q_M(\theta, \theta^{(r)})$, meaning

$$\theta^{(r+1)} = \arg\max_{\theta} \frac{1}{M} \sum_{j=1}^{M} \log f(\mathbf{y}, \mathbf{x}_j^* \mid \theta). \tag{2.7}$$

Because $Q_M(\theta, \theta^{(r)}) \to Q(\theta, \theta^{(r)})$ a.s. as $M \to \infty$, the larger Monte Carlo sample size is, the more accurate the estimator $Q_M(\theta, \theta^{(r)})$ is. However, the computation time increases when the sample size increases. In the MCEM algorithm, choosing the right sample size is an critical issue because it strikes a balance between the estimation accuracy and computational cost. In Booth and Robert (1999), an automated rule is designed to increase the sample size of independent sample for a generalized linear mixed model if the true EM step is covered by Monte Carlo error, where the central limit theorem and Taylor expansion are combined to access the Monte Carlo error at each iteration. With the rapid development of the MCMC method in the MCEM algorithm (see Chan and Ledolter (1995), McCulloch (1994), McCulloch (1997)) and Levine and Casella (2001) developed an automated scheme to increase the number of sample size if the Monte Carlo error exceeds the EM estimator at any iteration and this rule is tailored to the sample from the MCMC method.

Although the MCEM algorithm is more convenient to use than the original EM algorithm, it does not have the property of monotonicity, one of the key properties of the EM algorithm. Booth and Robert (1999) and Chan and Ledolter (1995) have shown that the sequence produced by the MCEM algorithm approaches the maximizer with a high probability in certain cases.

2.2 Particle Filters

Particle filters, also called Sequential Monte Carlo (SMC) algorithms, refer to a class of algorithms that use weighted particles to approximate a sequence of filtering distributions of interest. Although the SMC algorithm started to capture attention from the statisticians in the 1990s, its basic idea was developed from the sequential importance sampling (SIS), which is proposed in Handschin and Mayne (1969). In this section, a brief description of particle filters is presented.

2.2.1 Sequential Importance Sampling

SIS is an extension of importance sampling (IS). The basic idea of IS (Hammersley and Handscomb (1964)) is shown in the following equations

$$\mathbb{E}_p\left(f(X)\right) = \int f(x)p(x)\,\mathrm{d}x = \int \frac{f(x)p(x)}{q(x)}q(x)\,\mathrm{d}x = \mathbb{E}_q\left(\frac{f(X)p(X)}{q(X)}\right), \tag{2.8}$$

where $\mathbb{E}_p$ and $\mathbb{E}_q$ denote expectation for $X \sim p$ and $X \sim q$ respectively. Set the weights as

$$w(X) = \frac{p(X)}{q(X)}, \tag{2.9}$$

then the importance estimator of $\mathbb{E}_p\left(f(X)\right)$ is

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i)w(x_i), \tag{2.10}$$

where $\{x_i\}_{i=1}^N$ is an i.i.d. sample from q. According to the strong law of large numbers, this estimator converges to $\mathbb{E}_p\left(f(X)\right)$ almost surely.

SIS extends IS to a sequence of target distributions. Here SIS is introduced within the framework of NLSS models. Suppose that the target is to attain a sample from a sequence of filtering distributions $\phi_{t|t}(x_t), t = 1, \ldots, T$, then SIS consists of the procedures listed in Algorithm 1.

Algorithm 1 Sequential importance sampling

```
Ensure: $\{x_t^i, \omega_t^i\}_{i=1}^N$ approximates $\phi_{t|t}(x_t)$.
for t=1, ..., T do
    simulate $x_{t+1}^i \sim q(\cdot \mid x_t^i)$, $i = 1, \ldots, N$;
    compute the incremental weights
        $w_{t+1}(x_t^i, x_{t+1}^i) = \frac{f(x_{t+1}^i \mid x_t^i) g(y_{t+1} \mid x_{t+1}^i)}{q(x_{t+1}^i \mid x_t^i)}, i = 1, \ldots, N;$
    update the weights $\omega_{t+1}^i = \omega_t^i w_{t+1}(x_t^i, x_{t+1}^i)$, $i = 1, \ldots, N$.
end for
Take $\{x_{t+1}^i, \omega_{t+1}^i\}_{i=1}^N$ as an approximation for $\phi_{t+1|t+1}(x_{t+1})$.
```

2.2.2 Sampling Importance Resampling

Another indispensable element of particle filters is the sampling importance resampling introduced in Rubin (1987). Sampling importance resampling typically adds a resampling step into IS. Assume that the target is to draw a sample from a distribution proportional to the function $p(x)$, but it is difficult to simulate from $p(x)$ directly. The sampling importance resampling can be used to address this problem, and its procedures are summarized in Algorithm 2.

2.2.3 Sequential Importance Sampling and Resampling

SIS is designed to manage a sequence of target distributions. However, as the length of a sequence increases, this algorithm suffers from the degeneracy

Algorithm 2 Sampling importance resampling

```
Simulate $x_1, \dots, x_N$ from the proposal distribution $q(x)$;
compute the unnormalized weights $w(x_i)$ as Eq.(2.9), $i = 1, \dots, N$;
calculate the normalized weights
    $\omega_i = \frac{w(x_i)}{\sum_{j=1}^{N} w(x_j)}, i = 1, \dots, N;$
draw $\{x_i'\}_{i=1}^{N}$ independently from the set $\{x_i\}_{i=1}^{N}$ with the probabilities $\{\omega_i\}_{i=1}^{N}$.
Then $\{x_i'\}_{i=1}^{N}$ is a sample from the distribution proportional to $p(x)$.
```

of the importance weights: most of the importance weights tend to 0, which causes the effective sample size to be small and the strong law of large numbers to no longer apply. To solve this problem, Gordon et al. (1993) introduces the sampling importance resampling scheme to the framework of SIS, which is regarded as the first work performed on particle filters (also known as bootstrap particle filter). The sequential importance sampling and resampling (SISR) scheme, which combines SIS and sampling importance resampling, is presented in Algorithm 3.

Several schemes exist to operate the resampling step. The intuitive method is multinomial resampling proposed in Gordon et al. (1993). This scheme is to draw N new particles $\{x_{t+1}^i\}_{i=1}^{N}$ with replacement from the original pool $\{\tilde{x}_{t+1}^i\}_{i=1}^{N}$ with the corresponding importance weights $\{\tilde{\omega}_{t+1}^i\}_{i=1}^{N}$, which is equivalent to drawing a sample from a multinomial distribution

$$M \mid \{\tilde{x}_{t+1}^i, \tilde{\omega}_{t+1}^i\}_{i=1}^{N} \sim Multinomial(N, \{\tilde{\omega}_{t+1}^i\}_{i=1}^{N}), \qquad (2.11)$$

where M is an N-dimensional random vector in which the ith element represents the number of $\tilde{x}_{t+1}^i$ in the new particles.

Another scheme is residual resampling introduced in Liu and Chen (1995). Unlike multinomial resampling, the number of each original particle in this resampling method is the sum of two parts: a deterministic part and a random part. In the deterministic part, the number of $\tilde{x}_{t+1}^i$ in the new set is $\lfloor N\tilde{\omega}_{t+1}^i \rfloor$,

Algorithm 3 Sequential importance sampling and resampling

```
Ensure: $\{x_t^i, \omega_t^i\}_{i=1}^N$ approximates $\phi_{t|t}(x_t)$.
(Sampling:)
for i=1, ..., N do
    simulate $\tilde{x}_{t+1}^i \sim q(\cdot \mid x_t^i)$;
    compute the incremental weights
        $$w_{t+1}(x_t^i, \tilde{x}_{t+1}^i) = \frac{f(\tilde{x}_{t+1}^i \mid x_t^i) g(y_{t+1} \mid \tilde{x}_{t+1}^i)}{q(\tilde{x}_{t+1}^i \mid x_t^i)};$$
    update the weights $\tilde{\omega}_{t+1}^i = \omega_t^i w_{t+1}(x_t^i, \tilde{x}_{t+1}^i)$.
end for
(Resamping:)
draw index $I_{t+1}^1, \ldots, I_{t+1}^N$ independently with the probability
    $$P(I_{t+1} = j) = \tilde{\omega}_{t+1}^j, j = 1, \ldots, N;$$
set $x_{t+1}^i = \tilde{x}_{t+1}^{I_{t+1}^i}$ and $\omega_{t+1}^i = 1$ for $i = 1, \ldots, N$.
Take $\{x_{t+1}^i, \omega_{t+1}^i\}_{i=1}^N$ as an approximation for $\phi_{t+1|t+1}(x_{t+1})$.
```

where $\lfloor x \rfloor$ represents the largest integer that is smaller than x. A total of $\tilde{N} = N - \sum_{i=1}^N \lfloor N\tilde{\omega}_{t+1}^i \rfloor$ particles are left in the random part, and these particles are drawn from a multinomial distribution. Hence, the total number of particles in the new set becomes

$$M = \left(\lfloor N\tilde{\omega}_{t+1}^1 \rfloor, \ldots, \lfloor N\tilde{\omega}_{t+1}^N \rfloor \rfloor\right) + \tilde{M}, \tag{2.12}$$

where $\tilde{M} \sim Multinomial(\tilde{N}, \{\tilde{\omega}_{t+1}^i\}_{i=1}^N)$.

Compared with multinomial resampling, residual resampling reduces the variance of the estimators. Another method is systematic resampling (Carpenter et al. (1999)), which also serves the same purpose. This method designs an algorithm that satisfies the difference between the number of $\tilde{x}_{t+1}^i$ in the new set and $N\tilde{\omega}_{t+1}^i$ is no larger than 1.

2.2.4 Auxiliary Particle Filters

SISR has several drawbacks. The importance weights are extremely uneven when the observations contain an outlier, which requires an exceedingly large number of particles to approximate the target distribution. To solve this problem, a modified particle filters called an auxiliary particle filter (APF) is proposed in Pitt and Shephard (1999). An APF combines the information from the new coming observation and the importance weights generated in the previous iteration to calculate the new importance weights and then performs the resampling step at the beginning of each iteration. To implement this idea, Pitt and Shephard (1999) introduces an auxiliary variable to particle filters.

Together with the auxiliary variable k, the target distribution becomes the joint distribution

$$p(x_{t+1}, k \mid y_{0:t+1}) \propto g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t^k) \omega_t^k, \qquad k = 1, \ldots, N. \tag{2.13}$$

Since $\{x_t^k, \omega_t^k\}_{k=1}^N$ also approximates $\phi_{t|t}(x_t)$, the joint distribution is written as a mixture distribution related to $\{x_t^k, \omega_t^k\}_{k=1}^N$. Based on the idea of the SIR algorithm, a proposal distribution $q(x_{t+1}, k \mid y_{0:t+1})$ is used and then the importance weight is proportional to

$$\frac{g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t^k) \omega_t^k}{q(x_{t+1}, k \mid y_{0:t+1})}. \tag{2.14}$$

In the APF framework, the proposal distribution is

$$q(x_{t+1}, k \mid y_{0:t+1}) \propto q_1(y_{t+1} \mid \mu_{t+1}^k) q_2(x_{t+1} \mid x_t^k) \omega_t^k, \qquad k = 1, \ldots, N, \tag{2.15}$$

where μ_{t+1}^k can be some key value related to $x_{t+1} \mid x_t^k$, for example, the mean or the mode. According to Eq.(2.14), the importance weight is

$$\tilde{\omega}_{t+1} \propto \frac{g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t^k)}{q_1(y_{t+1} \mid \mu_{t+1}^k) q_2(x_{t+1} \mid x_t^k)}. \tag{2.16}$$

In this setting, first draw the index $k \sim q(k \mid y_{0:t+1}) \propto q_1(y_{t+1} \mid \mu_{t+1}^k) \omega_t^k$, and then simulate the state variable $x_{t+1} \sim q_2(x_{t+1} \mid x_t^k)$. The sample $\{x_{t+1}^i, k^i\}_{i=1}^N$

is obtained with the corresponding weights defined in Eq.(2.16), which approximates to the joint distribution $p(x_{t+1}, k \mid y_{0:t+1})$. Therefore, the weighted sample $\{x^i_{t+1}, \tilde{\omega}^i_{t+1}\}^N_{i=1}$ approximates the marginal distribution $p(x_{t+1} \mid y_{0:t+1})$. The entire procedure of APF is presented in Algorithm 4.

Algorithm 4 Auxiliary particle filters

```
Ensure: $\{x^i_t, \omega^i_t\}^N_{i=1}$ approximates $\phi_{t|t}(x_t)$.
compute the first-stage unnormalized weights $\tilde{\lambda}^k_{t+1} = q_1(y_{t+1} \mid \mu^k_{t+1})\omega^k_t, k = 1, \dots, N$.
for i=1, ..., N do
    draw index $k^i$ from the set $\{1, \dots, N\}$ with respect to the probability
        $P(k^i = j) = \frac{\tilde{\lambda}^j_{t+1}}{\sum_{m=1}^N \tilde{\lambda}^m_{t+1}}, j = 1, \dots, N;$
    sample $\tilde{x}^i_{t+1} \sim q_2(\cdot \mid x^{k^i}_t)$;
    calculate the second-stage unnormalized weights
        $\bar{\omega}^i_{t+1} = \frac{g(y_{t+1} \mid \tilde{x}^i_{t+1}) f(\tilde{x}^i_{t+1} \mid x^{k^i}_t)}{q_1(y_{t+1} \mid \mu^{k^i}_{t+1}) q_2(\tilde{x}^i_{t+1} \mid x^{k^i}_t)}.$
end for
Compute the second-stage normalized weights
    $\tilde{\omega}^j_{t+1} = \frac{\bar{\omega}^j_{t+1}}{\sum_{m=1}^N \bar{\omega}^m_{t+1}}, j = 1, \dots, N;$
draw index $I^1_{t+1}, \dots, I^N_{t+1}$ independently with the probability
    $P(I_{t+1} = j) = \tilde{\omega}_{t+1}, j = 1, \dots, N;$
set $x^i_{t+1} = \tilde{x}^{I^i_{t+1}}_{t+1}$ and $\omega^i_{t+1} = 1$ for $i = 1, \dots, N$.
Take $\{x^i_{t+1}, \omega^i_{t+1}\}^N_{i=1}$ as an approximation for $\phi_{t+1|t+1}(x_{t+1})$.
```

2.2.5 Resample-move Algorithm

Although sampling importance resampling is introduced to SIS, SISR continues to cause the impoverishment of particles when static unknown parameters are involved. Because no new-coming particle exists, less and less particles remain; therefore, the strong law of large numbers becomes invalid in this situation. In Gilks and Berzuini (2001), a move step is added after the resampling step. In this step, the particles are renewed using a transition kernel with a stationary distribution. The Markov Chain Monte Carlo (MCMC) algorithm is typically applied to construct this kernel. Because the resampled particles are the sample from the target distribution, only one iteration of the MCMC move is required to rejuvenate the particles. Assume that the sequence of target distributions are $\pi_t(\theta)$, the resample-move algorithm is summarized in Algorithm 5.

Algorithm 5 Resample-move algorithm

```
Ensure: $\{\theta_t^i, \omega_t^i\}_{i=1}^N$ approximates $\pi_t(\theta)$.
(Resampling step:)
for i=1, ..., N do
    compute the importance weights
        $$\tilde{\omega}_{t+1}^i \propto \omega_t^i \frac{\pi_{t+1}(\theta_t^i)}{\pi_t(\theta_t^i)}.$$
end for
Draw index $I_{t+1}^1, \dots, I_{t+1}^N$ independently with the probability
    $$P(I_{t+1} = j) = \tilde{\omega}_{t+1}^j, j = 1, \dots, N;$$
set $\tilde{\theta}_{t+1}^i = \theta_t^{I_{t+1}^i}$ and $\omega_{t+1}^i = 1$ for $i = 1, \dots, N$.
(Move step:)
renew the particles by sampling
    $$\theta_{t+1}^i \sim q_{t+1}(\cdot, \tilde{\theta}_{t+1}^i),$$
where $q_{t+1}$ is a transition kernel of stationary distribution $\pi_{t+1}$.
Take $\{\theta_{t+1}^i, \omega_{t+1}^i\}_{i=1}^N$ as an approximation for $\pi_{t+1}(\theta)$.
```

Although Chopin (2002) classifies this algorithm into particle filters, it is typically used to do parameter learning, which involves the sequential and joint learning of state variables and parameters. Its application to the NLSS models is discussed in Section 3.2.1.

2.3 Particle Smoothers

In the previous section, particle filters that produce the sample approximating the filtering distributions have been discussed. In this section, algorithms called particle smoothers, which are related to but more complicated than particle filters, are presented. The target distributions becoming smoothing distributions cause additional challenges because of the increasing dimension of variables as the process evolves.

In Kitagawa (1996), an algorithm is proposed with the same basic idea as Gordon et al. (1993) to manage both filtering distributions and smoothing distributions. Hürzeler and Künsch (1998) proposes the basic idea of the forward filtering backward smoothing (FFBSm) algorithm, and Godsill et al. (2001) introduces this algorithm to deal with maximum a posteriori sequence estimation in nonlinear and non-Gaussian dynamic models. The marginal smoothing distributions are the distributions of interest. Subsequently, Godsill et al. (2004) introduces the FFBSi algorithm to approximate joint smoothing distributions. Andrieu et al. (2010) proposes the PMCMC algorithm, which combines the idea of particle methods and the MCMC algorithm. As a special case of PMCMC, the particle independent Metropolis-Hastings (PIMH) algorithm can be used to obtain particles from the joint smoothing distribution. In the remainder of this section, the FFBSi algorithm and the PIMH algorithm are introduced.

2.3.1 Forward Filtering Backward Simulation

The FFBSi algorithm consists of two segments. The first segment is the forward filtering part, and the methods introduced in Section 2.2 can be used to obtain weighted samples approximating the filtering distribution. The second segment is the backward process, which updates the previous particles to approximate the joint smoothing distribution.

The joint smoothing distribution can be written as

$$\begin{aligned}\phi_{0:t|t}(x_{0:t}) &= p(x_t \mid y_{1:t}) \prod_{i=0}^{t-1} p(x_i \mid x_{i+1:t}, y_{1:t}) \\ &= \phi_{t|t}(x_t) \prod_{i=0}^{t-1} p(x_i \mid x_{i+1}, y_{1:i}) \\ &\propto \phi_{t|t}(x_t) \prod_{i=0}^{t-1} \phi_{i|i}(x_i) f(x_{i+1} \mid x_i). \qquad (2.17)\end{aligned}$$

In the forward filtering step, the weighted particles $\{x_i^j, \omega_i^j\}_{j=1}^N$ are obtained to approximate the filtering distribution $\phi_{i|i}(x_i)$ for $i = 0, \ldots, t$. According to Eq.(2.17), the particles with respect to the smoothing distribution can be obtained using the following approximation for $i = t-1, \ldots, 0$ recursively

$$p(x_i \mid x_{i+1:t}, y_{1:t}) = p(x_i \mid x_{i+1}, y_{1:i}) \approx \sum_{j=1}^{N} \omega_{i|i+1}^{(j)} \delta_{x_i^j}(x_i), \qquad (2.18)$$

where the importance weight is

$$\omega_{i|i+1}^{(j)} = \frac{\omega_i^j f(x_{i+1} \mid x_i^j)}{\sum_{k=1}^{N} \omega_i^k f(x_{i+1} \mid x_i^k)}. \qquad (2.19)$$

The FFBSi is summarized in Algorithm 6.

2.3.2 Particle Independent Metropolis-Hastings

It is intuitive to use the MCMC method to manage the joint smoothing distribution. However, it is difficult to find an efficient high-dimensional proposal distribution in an original way. Andrieu et al. (2010) combines the MCM-C algorithms and SMC method to propose the PMCMC method, where the SMC method is used to provide suitable proposal distributions for MCMC algorithms.

Algorithm 6 Forward filtering backward simulation

```
Ensure: $\{x_i^j, \omega_i^j\}_{j=1}^N$ approximates $\phi_{i|i}(x_i)$ for $i = 0, \ldots, t$.
for l=1, ..., N do
    Draw index $I_t^l$ with the probability
        $P(I_t = j) = \omega_t^j, j = 1, \ldots, N;$
    set $\tilde{x}_t^l = x_t^{I_t^l}$.
    for i=t-1, ..., 0 do
        compute the importance weights
            $\omega_{i|i+1}^{(j)} = \frac{\omega_i^j f(\tilde{x}_{i+1}^l \mid x_i^j)}{\sum_{k=1}^N \omega_i^k f(\tilde{x}_{i+1}^l \mid x_i^k)}, j = 1, \ldots, N;$
        draw index $I_i^l$ with the probability
            $P(I_i = j) = \omega_{i|i+1}^{(j)}, j = 1, \ldots, N;$
        set $\tilde{x}_i^l = x_i^{I_i^l}$.
    end for
end for
Take $\{\tilde{x}_{0:t}^j\}_{j=1}^N$ as an approximation for $\phi_{0:t|t}(x_{0:t})$.
```

One of the key elements of the PMCMC method is to find an unbiased estimator for the marginal distribution $p(y_{1:t} \mid \theta)$, which is denoted by $p_\theta(y_{1:t})$. For simplicity, the distributions related to θ are denoted using the similar forms for the remainder of this thesis. Because

$$p_\theta(y_{1:t}) = p_\theta(y_1) \prod_{i=2}^{t} p_\theta(y_i \mid y_{1:i-1}), \tag{2.20}$$

where

$$\begin{aligned} p_\theta(y_i \mid y_{1:i-1}) &= \int \frac{p_\theta(x_{1:i}, y_{1:i})}{p_\theta(y_{1:i-1})} \, \mathrm{d}x_{1:i} \\ &= \int \frac{p_\theta(x_{1:i}, y_{1:i}) p_\theta(x_{1:i-1} \mid y_{1:i-1})}{p_\theta(x_{1:i-1}, y_{1:i-1})} \, \mathrm{d}x_{1:i} \\ &= \int w_i(x_{1:i}) q_\theta(x_i \mid x_{i-1}, y_i) p_\theta(x_{1:i-1} \mid y_{1:i-1}) \, \mathrm{d}x_{1:i}, \end{aligned} \tag{2.21}$$

and

$$w_i(x_{1:i}) = \frac{p_\theta(x_{1:i}, y_{1:i})}{p_\theta(x_{1:i-1}, y_{1:i-1}) q_\theta(x_i \mid x_{i-1}, y_i)}, \tag{2.22}$$

which is the unnormalized importance weight at the ith iteration of the SMC methods, the following estimator can be used

$$\hat{p}_\theta(y_{1:t}) = \hat{p}_\theta(y_1) \prod_{i=2}^{t} \hat{p}_\theta(y_i \mid y_{1:i-1}), \tag{2.23}$$

where

$$\hat{p}_\theta(y_i \mid y_{1:i-1}) = \frac{1}{N} \sum_{k=1}^{N} \hat{w}_i(x_{1:i}^k). \tag{2.24}$$

The unbiasedness of the estimator in Eq.(2.23) is proved in Pitt and Silva (2012). Based on the above estimator, the PIMH algorithm is presented in Algorithm 7. More details about the rationale of PIMH are provided in Douc et al. (2014).

Algorithm 7 Particle independent Metropolis-Hastings

```
for k=1 do
    run a SMC algorithm targeting $\phi_{0:t|t}(x_{0:t})$, obtain the corresponding weighted particles $\{x^j_{0:t}, \omega^j_t\}^N_{j=1}$, and compute $\hat{p}_\theta(y_{1:t})(1)$ according Eq.(2.23);
    draw index $I$ with the probability $P(I = j) = \tilde{\omega}^j_t, j = 1, \ldots, N$, where $\tilde{\omega}_t$ is the normalized weight;
    set $x_{0:t}(1) = x^I_{0:t}$.
end for
for  k=2, ..., M do
    run a SMC algorithm targeting $\phi_{0:t|t}(x_{0:t})$, obtain the corresponding weighted particles $\{x^j_{0:t}, \omega^j_t\}^N_{j=1}$, and compute $\hat{p}_\theta(y_{1:t})^*$;
    calculate the acceptance rate
        $$r = \min\left\{1, \frac{\hat{p}_\theta(y_{1:t})^*}{\hat{p}_\theta(y_{1:t})(k-1)}\right\};$$
    draw $U$ uniformly on $[0, 1]$;
    if $U \leq r$, draw index $I$ with the probability $P(I = j) = \tilde{\omega}^j_t, j = 1, \ldots, N$, set $x_{0:t}(k) = x^I_{0:t}$ and $\hat{p}_\theta(y_{1:t})(k) = \hat{p}_\theta(y_{1:t})^*$; otherwise set $x_{0:t}(k) = x_{0:t}(k-1)$ and $\hat{p}_\theta(y_{1:t})(k) = \hat{p}_\theta(y_{1:t})(k-1)$.
end for
Burn the first $\lfloor M/c \rfloor$ steps, take $\{x_{0:t}(k)\}^M_{k=\lfloor M/c \rfloor+1}$ as an approximation for $\phi_{0:t|t}(x_{0:t})$, where $c$ is a positive constant.
```

Chapter 3

Inference for Nonlinear State Space Models

In this chapter, the maximum likelihood estimation and Bayesian analysis of inference for NLSS models are studied. Because the densities involved in NLSS models are high-dimensional with no closed forms, particle methods are used to approximate the corresponding densities. Section 3.1 introduces the particle-based maximum likelihood estimation for NLSS models, and Bayesian analysis for these models is presented in Section 3.2.

3.1 Maximum Likelihood Estimation for Nonlinear State Space Models

In the NLSS models, the likelihood function of observations $y_{0:T}$ is

$$\begin{aligned} L(\boldsymbol{\theta} \mid y_{0:T}) &= p_{\boldsymbol{\theta}}(y_{0:T}) \\ &= \int \cdots \int \xi(x_0) g_{\boldsymbol{\theta}}(y_0 \mid x_0) \prod_{i=1}^{T} f_{\boldsymbol{\theta}}(x_i \mid x_{i-1}) g_{\boldsymbol{\theta}}(y_i \mid x_i) \, \mathrm{d}x_{0:T} \\ &= p_{\boldsymbol{\theta}}(y_0) \prod_{i=1}^{T} p_{\boldsymbol{\theta}}(y_i \mid y_{1:i-1}), \end{aligned} \tag{3.1}$$

where

$$p_{\boldsymbol{\theta}}(y_i \mid y_{1:i-1}) = \int\int \phi^{\boldsymbol{\theta}}_{i-1|i-1}(x_{i-1}) f_{\boldsymbol{\theta}}(x_i \mid x_{i-1}) g_{\boldsymbol{\theta}}(y_i \mid x_i)\, \mathrm{d}x_{i-1:i}. \tag{3.2}$$

Because the latent variables $x_{0:T}$ are involved in this model, the EM algorithm is tailored to obtain the MLE. To obtain

$$\arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta} \mid y_{0:T}), \tag{3.3}$$

the latent variables $x_{0:T}$ are augmented to form the complete likelihood

$$L(\boldsymbol{\theta} \mid x_{0:T}, y_{0:T}) = p_{\boldsymbol{\theta}}(x_{0:T}, y_{0:T}). \tag{3.4}$$

In the EM algorithm, a sequence $\{\boldsymbol{\theta}^{(r)}\}$ is obtained from a starting point $\boldsymbol{\theta}^{(0)}$ by

$$\begin{aligned} \boldsymbol{\theta}^{(r+1)} &= \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)}) \\ &= \arg\max_{\boldsymbol{\theta}} E[\log p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta}) \mid \boldsymbol{\theta}^{(r)}, y_{0:T}]. \end{aligned} \tag{3.5}$$

The calculation of expectation in Eq.(3.5) is difficult and time-consuming because of the high-dimensional variables involved. Therefore, the MCEM algorithm is used to obtain the sequence

$$\begin{aligned} \boldsymbol{\theta}^{(r+1)} &= \arg\max_{\boldsymbol{\theta}} Q_M(\boldsymbol{\theta}, \boldsymbol{\theta}^{(r)}) \\ &= \arg\max_{\boldsymbol{\theta}} \frac{1}{M} \sum_{j=1}^{M} \log p(x_{0:T}^{(j,r)}, y_{0:T} \mid \boldsymbol{\theta}), \end{aligned} \tag{3.6}$$

where $\{x_{0:T}^{(j,r)}\}_{j=1}^{M}$ are generated from the joint smoothing distribution $p(x_{0:T} \mid y_{0:T}, \boldsymbol{\theta}^{(r)})$. Hence, the methods introduced in Section 2.3 can be used to generate this sample. In Kim (2005), the MCEM algorithm with the FFBSi algorithm is discussed, and only the application of the PIMH algorithm is presented here. Algorithm 7 is used to obtain the Monte Carlo sample at each iteration, and obtain the sequence $\{\boldsymbol{\theta}^{(r)}\}$. Two critical issues in the MCEM algorithm with the PIMH algorithm in the NLSS models are discussed as follow.

3.1.1 Stopping Rule

An critical issue in the MCEM algorithm is determining when to stop. The estimation may be inaccurate if the procedure is stopped too early; while it is time-consuming if too many iterations are implemented. Most of the stopping rules for EM algorithm are based on the idea that the EM procedure is stopped when the difference between the parameter values in the adjacent iterations is small. One of the common stopping rules for the EM algorithm is

$$\max_i \left(\frac{\mid \boldsymbol{\theta}_i^{(r+1)} - \boldsymbol{\theta}_i^{(r)} \mid}{\mid \boldsymbol{\theta}_i^{(r)} \mid + \delta_1} \right) < \delta_2, \tag{3.7}$$

where δ_1 and δ_2 are given constants. Searle et al. (1992) states that using $\delta_1 = 0.001$ and $\delta_2 = 0.0001$ are popular choices. However, the implementation of the MCEM algorithm costs too much time if 0.0001 is used as the predetermined value for δ_2. The possible reason is that the Monte Carlo sample is used to approximate the expectation, and the corresponding Monte Carlo error increases the variability of estimators. Booth and Robert (1999) suggests that a value of δ_2 between 0.002 and 0.005 can be chosen when implementing the MCEM algorithm. Because particle methods typically create Monte Carlo errors, $\delta_1 = 0.001$ and $\delta_2 = 0.005$ are used in this thesis. To reduce the risk of stopping prematurely, it is suggested in Booth and Robert (1999) that the iteration procedure is not stopped until Eq.(3.7) is satisfied for three adjacent iterations. This stopping scheme is also adopted.

3.1.2 Sample Size Issue

Another critical issue in the MCEM algorithm is how to choose the sample size at each iteration. The algorithm is inefficient if it starts with a large sample size because the values of the parameters are too divergent from the true values. As the value of the parameter estimate becomes close to the true value, a large sample size is required to achieve the estimation with a high level of reliability. Therefore, a good selection of sample size is to increase the sample size when the parameters converge, which maintains the balance between estimation accuracy and computational cost.

Levine and Casella (2001) proposes an automatic rule to determine whether increasing the sample size should be increased at each iteration in the MCEM algorithm with MCMC sampler. It would be difficult to obtain the variance estimator of the central limit theorem in the MCMC setting. Renewal theory and its corresponding central limit theory developed in Robert et al. (1999) are used to construct the confidence region of the first derivative of $Q_M(\theta, \theta^{(r)})$, denoted as $Q_M^{(1)}(\theta, \theta^{(r)})$. If $Q_M^{(1)}(\theta^{(r)}, \theta^{(r-1)})$ lies in this confidence region, increase the sample size in the next iteration. However, this criterion would not provide hints about the convergence of the estimators. Further, to construct rectangular confidence regions, the independence of parameters is imposed. To circumvent these difficulties, the relative likelihood function introduced in Chan and Ledolter (1995) is used to propose an automated rule to determine if the sample size should be increased at each iteration in the MCEM algorithm within MCMC setting. As in Levine and Casella (2001), renewal theory is used to construct the confidence interval. However, unlike the central limit theorem used in Levine and Casella (2001), the intermediate lemma derived in Robert et al. (1999) is applied. Andrieu et al. (2010) proves that the PIMH algorithm is a standard independent Metropolis-Hastings (IMH) algorithm, which is a special case of the MCMC method. Therefore, a new automated rule within the PIMH framework is introduced.

Renewal theory is a subsampling scheme that causes the dependence among subsamples to disappear asymptotically. Assume that the sample $\{x_{0:T}^i\}_{i=1}^M$ is given, renewal theory is used to obtain a subsample. First, a sequence of $\{u_k\}$ is generated independently from a Possion distribution with parameter λ_k, where $\lambda_k = \lambda k^d$ with $\lambda \geq 1$ and $d > 0$. The subsample index t_k is then calculated as $t_k = u_1 + \cdots + u_k + k$, and the sample size for the subsample is denoted by $N_M = \sup\{n : t_n \leq M\}$. Therefore the corresponding subsample is $\{x_{0:T}^{t_k}\}_{k=1}^{N_M}$. For a given function $h(x_{0:T})$, according to Robert et al. (1999), we have the following theorem.

Theorem 3.1 Suppose that the Markov chain $\{x_{0:T}^i\}$ is ergodic and strongly mixed with geometrically decaying mixing coefficients, and $\mathbb{E}_{\phi_{0:T|T}} \mid h(x_{0:T}) \mid^{2+\delta} < \infty$ for some $\delta > 0$, let

$$S_{N_M} = \frac{1}{\sqrt{N_M \hat{v}_M}} \sum_{k=1}^{N_M} [h(x_{0:T}^{t_k}) - \mathbb{E}_{\phi_{0:T|T}}(h(x_{0:T}))], \tag{3.8}$$

where

$$\hat{v}_M = \frac{1}{M} \sum_{i=1}^{M} [h(x_{0:T}^i)]^2 - \left(\frac{1}{M} \sum_{i=1}^{M} h(x_{0:T}^i) \right)^2, \tag{3.9}$$

then S_{N_M} converges weakly to a standard normal distribution.

The relative likelihood proposed in Chan and Ledolter (1995) is the ratio of likelihoods between the successive iterations. In this setting, the relative likelihood at the rth iteration is written as

$$RL^{(r)} = \frac{p(y_{0:T} \mid \boldsymbol{\theta}^{(r)})}{p(y_{0:T} \mid \boldsymbol{\theta}^{(r-1)})}. \tag{3.10}$$

The change of log-likelihoods can be expressed using the function of relative likelihood

$$\triangle l(\boldsymbol{\theta}^{(r)}, \boldsymbol{\theta}^{(r-1)}) = \log p(y_{0:T} \mid \boldsymbol{\theta}^{(r)}) - \log p(y_{0:T} \mid \boldsymbol{\theta}^{(r-1)}) = \log RL^{(r)}. \tag{3.11}$$

As the parameter values move closer together, the change of log-likelihoods becomes closer to zero, and hence the relative likelihood tends to 1. Based on this idea, confidence intervals of relative likelihood at each iteration are constructed to determine whether the approximations between the successive iterations are sufficiently close, and then to determine if the sample size should be increased at the subsequent iteration.

The relative likelihood at the rth iteration can be expressed as

$$\begin{aligned} RL^{(r)} &= \frac{p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta}^{(r)})}{p(x_{0:T} \mid y_{0:T}, \boldsymbol{\theta}^{(r)})} \cdot \frac{p(x_{0:T} \mid y_{0:T}, \boldsymbol{\theta}^{(r-1)})}{p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta}^{(r-1)})} \\ &= \mathbb{E}_{\boldsymbol{\theta}^{(r-1)}} \left[\frac{p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta}^{(r)})}{p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta}^{(r-1)})} \mid y_{0:T} \right], \end{aligned} \tag{3.12}$$

where the second equation is obtained by multiplying $p(x_{0:T} \mid y_{0:T}, \boldsymbol{\theta}^{(r)})$ and integrating out $x_{0:T}$. Define the function

$$h^{(r)}(x_{0:T}) = \frac{p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta}^{(r)})}{p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta}^{(r-1)})}, \tag{3.13}$$

and estimate $RL^{(r)}$ and its corresponding variance as

$$\hat{RL}^{(r)} = \frac{1}{N_M} \sum_{k=1}^{N_M} h^{(r)}(x_{0:T}^{t_k, r-1}) = \frac{1}{N_M} \sum_{k=1}^{N_M} \frac{f(x_{0:T}^{(t_k, r-1)}, y_{0:T} \mid \boldsymbol{\theta}^{(r)})}{f(x_{0:T}^{(t_k, r-1)}, y_{0:T} \mid \boldsymbol{\theta}^{(r-1)})} \tag{3.14}$$

and

$$v\hat{a}r[h(x_{0:T})]^{(r)} = \frac{1}{M} \sum_{j=1}^{M} [h(x_{0:T}^{j,r-1})]^2 - \left(\frac{1}{M} \sum_{j=1}^{M} h(x_{0:T}^{j,r-1}) \right)^2 \tag{3.15}$$

respectively, where $\{x_{0:T}^{(j,r-1)}\}_{j=1}^{M}$ is a sample from $p(x_{0:T} \mid y_{0:T}, \boldsymbol{\theta}^{(r-1)})$ and $\{x_{0:T}^{(t_k,r-1)}\}_{k=1}^{N_M}$ is the corresponding subsample generated using renewal theory. According to Theorem 3.1, an approximate $100(1-\alpha)\%$ confidence interval of relative likelihood at the rth iteration is constructed as

$$\left[\hat{RL}^{(r)} - z_{1-\alpha/2} \sqrt{\mathrm{v\hat{a}r}[h(\boldsymbol{z}_T)]^{(r)}/N_M}, \hat{RL}^{(r)} + z_{1-\alpha/2} \sqrt{\mathrm{v\hat{a}r}[h(\boldsymbol{z}_T)]^{(r)}/N_M} \right], \tag{3.16}$$

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$th quantile of a standard normal distribution. If this confidence interval contains 1, then the approximations at $(r-1)$th and rth are regarded as close. Increase the sample size at the $(r+1)$th iteration. The sample size is increased according to the formula $M^{(r+1)} = cM^{(r)}$, and in the simulation study and empirical analysis, $\alpha = 0.25$ and $c = 1.2$ are chosen as suggested in Booth and Robert (1999).

3.2 Bayesian Analysis for Nonlinear State Space Models

In this section, Bayesian analysis of type II maximum likelihood prior, which is one of the parametric empirical Bayesian methods, is discussed. Similar to other empirical Bayesian methods, robustness is an important advantage of Bayesian estimators based on type II maximum likelihood prior.

In a Bayesian setting, if the continuous random variable X follows the probability density $p(x \mid \theta)$, and the parameter θ has the prior density $\pi(\theta)$, then the marginal density of X is

$$m(x \mid \pi) = \int_{\theta} p(x \mid \theta)\pi(\theta)\,\mathrm{d}\theta. \tag{3.17}$$

From the empirical Bayes perspective, the unknown prior density π can be selected or estimated by observations. The marginal density $m(x \mid \pi)$ can be interpreted as the likelihood function of π. Similar to classical statistics, when the data x is observed and $m(x \mid \pi_1) > m(x \mid \pi_2)$, it is reasonable to conclude that π_1 is more plausible than π_2. This is the basic idea of type II maximum likelihood prior.

Definition 3.2 Suppose that Γ is a class of priors under consideration, and $\hat{\pi}$ satisfies (for the observed data x)

$$m(x \mid \hat{\pi}) = \sup_{\pi \in \Gamma} m(x \mid \pi), \tag{3.18}$$

then $\hat{\pi}$ is called the type II maximum likelihood prior, or ML-II prior.

Specifically, Γ is considered as a class of known density functions $\pi(\theta \mid \lambda)$ with unknown hyper-parameters λ. Therefore, the target becomes $\sup_{\lambda \in \Lambda} m(x \mid \pi(\theta \mid \lambda))$. Considering the inference for NLSS models in this framework, the target is to find the hyper-parameters $\boldsymbol{\lambda}$ that maximize the marginal distribution, which can also be written as the "incomplete likelihood function" of $\boldsymbol{\lambda}$

as

$$\arg\max_{\boldsymbol{\lambda}} L(\boldsymbol{\lambda} \mid y_{0:T}) = \arg\max_{\boldsymbol{\lambda}} p(y_{0:T} \mid \boldsymbol{\lambda}). \tag{3.19}$$

However, this likelihood involves complex and high-dimensional integration in the NLSS models, which cannot be dealt with directly. The EM algorithm becomes a reasonable choice.

When estimating the hyper-parameters, both the latent variables $x_{0:T}$ and parameters $\boldsymbol{\theta}$ can be regarded as missing values. Hence the complete likelihood function is

$$p(x_{0:T}, y_{0:T}, \boldsymbol{\theta} \mid \boldsymbol{\lambda}) = p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta})\pi(\boldsymbol{\theta} \mid \boldsymbol{\lambda}). \tag{3.20}$$

According to the EM algorithm, a sequence $\{\boldsymbol{\lambda}^{(r)}\}$ is obtained from a starting point $\boldsymbol{\lambda}^{(0)}$ by

$$\begin{aligned}
\boldsymbol{\lambda}^{(r+1)} &= \arg\max_{\boldsymbol{\lambda}} Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(r)}) \\
&= \arg\max_{\boldsymbol{\lambda}} E[\ln p(x_{0:T}, y_{0:T}, \boldsymbol{\theta} \mid \boldsymbol{\lambda}) \mid \boldsymbol{\lambda}^{(r)}, y_{0:T}] \\
&= \arg\max_{\boldsymbol{\lambda}} \{E[\ln p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta}) \mid \boldsymbol{\lambda}^{(r)}, y_{0:T}] + E[\ln \pi(\boldsymbol{\theta} \mid \boldsymbol{\lambda}) \mid \boldsymbol{\lambda}^{(r)}, y_{0:T}]\} \\
&= \arg\max_{\boldsymbol{\lambda}} E[\ln \pi(\boldsymbol{\theta} \mid \boldsymbol{\lambda}) \mid \boldsymbol{\theta}^{(r)}, y_{0:T}]. \qquad (3.21)
\end{aligned}$$

Because the expectation in Eq.(3.21) has no closed form, the EM algorithm is replaced by the MCEM algorithm and the corresponding sequence is generated by

$$\begin{aligned}
\boldsymbol{\lambda}^{(r+1)} &= \arg\max_{\boldsymbol{\lambda}} Q_M(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(r)}) \\
&= \arg\max_{\boldsymbol{\lambda}} \frac{1}{M} \sum_{j=1}^{M} \ln \pi(\boldsymbol{\theta}^{(j,r)} \mid \boldsymbol{\lambda}), \qquad (3.22)
\end{aligned}$$

where $\{\boldsymbol{\theta}^{(j,r)}\}_{j=1}^{M}$ are sampled from $p(\boldsymbol{\theta} \mid \boldsymbol{\lambda}^{(r)}, y_{0:T})$. If the sample $\{x_{0:T}^{(j,r)}, \boldsymbol{\theta}^{(j,r)}\}_{j=1}^{M}$ can be obtained from the joint distribution $p(x_{0:T}, \boldsymbol{\theta} \mid \boldsymbol{\lambda}^{(r)}, y_{0:T})$, then the corresponding value $\{\boldsymbol{\theta}^{(j,r)}\}_{j=1}^{M}$ is a sample from the marginal distribution $p(\boldsymbol{\theta} \mid \boldsymbol{\lambda}^{(r)}, y_{0:T})$. Hence, the problem is transformed into generating the sample from $p(x_{0:T}, \boldsymbol{\theta} \mid y_{0:T}, \boldsymbol{\lambda})$, which is a crux in Bayesian analysis. In the remainder of this section, applications of the resample-move algorithm and the particle marginal Metropolis-Hastings (PMMH) sampler are discussed.

3.2.1 Resample-move Algorithm

As in Gilks and Berzuini (2001), a vector involving all unknown variables at time t is constructed, denoted by

$$\boldsymbol{z}_t = (x_{0:t}, \boldsymbol{\theta}). \tag{3.23}$$

Therefore, the target distribution can be written as

$$\pi_T(\boldsymbol{z}_T) = p(\boldsymbol{z}_T \mid \boldsymbol{\lambda}, y_{0:T}). \tag{3.24}$$

In this case, the dimension of the vector $\boldsymbol{z}_t$ increases as the process evolves. To deal with this situation, Gilks and Berzuini (2001) adds an augmentation step at the beginning of each iteration in the resample-move algorithm. In this augmentation step, the sample for new-coming variable x_{t+1} is drawn from $p(x_{t+1} \mid x_t, \boldsymbol{\theta})$. For the resampling step, the importance weight is computed as

$$\begin{aligned} \tilde{\omega}_{t+1} &= \omega_t \frac{\pi_{t+1}(\boldsymbol{z}_{t+1})}{\pi_t(\boldsymbol{z}_t) p(x_{t+1} \mid x_t, \boldsymbol{\theta})} \\ &= \omega_t \frac{p(x_{0:t+1} \mid \boldsymbol{\theta}, y_{0:t+1}) \pi(\boldsymbol{\theta} \mid \boldsymbol{\lambda})}{p(x_{0:t} \mid \boldsymbol{\theta}, y_{0:t}) \pi(\boldsymbol{\theta} \mid \boldsymbol{\lambda}) p(x_{t+1} \mid x_t, \boldsymbol{\theta})} \\ &\propto \omega_t \frac{f(x_{t+1} \mid x_t, \boldsymbol{\theta}) g(y_{t+1} \mid x_{t+1}, \boldsymbol{\theta})}{p(x_{t+1} \mid x_t, \boldsymbol{\theta})}. \end{aligned} \tag{3.25}$$

Based on this modified algorithm, samples from $\{\pi_t(\boldsymbol{z}_t)\}_{t=0}^{T}$ are obtained sequentially. The entire procedure is presented in Algorithm 8. Using Algorithm 8, $Q_M(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(r)})$ and $\boldsymbol{\lambda}^{(r+1)}$ are obtained iteratively until the stopping rule

$$\max_i \left(\frac{\mid \boldsymbol{\lambda}_i^{(r+1)} - \boldsymbol{\lambda}_i^{(r)} \mid}{\mid \boldsymbol{\lambda}_i^{(r)} \mid + \delta_1} \right) < \delta_2 \tag{3.26}$$

is satisfied, where δ_1 and δ_2 are as given in Section 3.1.1.

For the issue of sample size, an automated rule proposed in Booth and Robert (1999) is used to determine whether the Monte Carlo sample size should

Algorithm 8 The Resample-move algorithm in NLSS models

```
Ensure: $\{\boldsymbol{z}_{\boldsymbol{t}}^i, \omega_t^i\}_{i=1}^N$ approximates $\pi_t(\boldsymbol{z}_t)$.
(Augmentation step:)
for i=1, ..., N do
    simulate $\bar{x}_{t+1}^i \sim p(\cdot \mid x_t^i)$,
    add this new component to the particles
        $$\bar{\boldsymbol{z}}_{t+1}^i = (\boldsymbol{z}_t^i, \bar{x}_{t+1}^i).$$
end for
(Resampling step:)
for i=1, ..., N do
    compute the unnormalized importance weights
        $$\bar{\omega}_{t+1}^i \propto \omega_t^i \frac{f(x_{t+1}^i \mid x_t^i, \boldsymbol{\theta}^i) g(y_{t+1} \mid x_{t+1}^i, \boldsymbol{\theta}^i)}{p(x_{t+1}^i \mid x_t^i, \boldsymbol{\theta}^i)};$$
end for
calculate the normalized weights
    $$\tilde{\omega}_{t+1}^j = \frac{\bar{\omega}_{t+1}^j}{\sum_{m=1}^N \bar{\omega}_{t+1}^m}, j = 1, \ldots, N;$$
draw index $I_{t+1}^1, \ldots, I_{t+1}^N$ independently with the probability
    $$P(I_{t+1} = j) = \tilde{\omega}_{t+1}^j, j = 1, \ldots, N;$$
set $\tilde{\boldsymbol{z}}_{t+1}^i = \bar{\boldsymbol{z}}_{t+1}^{I_{t+1}^i}$ and $\omega_{t+1}^i = 1$ for $i = 1, \ldots, N$.
(Move step:)
renew the particles by sampling
    $$\boldsymbol{z}_{t+1}^i \sim q_{t+1}(\cdot, \tilde{\boldsymbol{z}}_{t+1}^i),$$
where $q_{t+1}$ is a transition kernel of stationary distribution $\pi_{t+1}$.
Take $\{\boldsymbol{z}_{t+1}^i, \omega_{t+1}^i\}_{i=1}^N$ as an approximation for $\pi_{t+1}(\boldsymbol{z}_{t+1})$.
```

be increased at each iteration, since particles generated from the resample-move algorithm are independent. Define

$$Q^{(1)}(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = \frac{\partial}{\partial \boldsymbol{\lambda}} Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}') \tag{3.27}$$

$$Q^{(2)}(\boldsymbol{\lambda}, \boldsymbol{\lambda}') = \frac{\partial^2}{\partial \boldsymbol{\lambda} \partial \boldsymbol{\lambda}^T} Q(\boldsymbol{\lambda}, \boldsymbol{\lambda}'), \tag{3.28}$$

and define $Q_M^{(1)}(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$ and $Q_M^{(2)}(\boldsymbol{\lambda}, \boldsymbol{\lambda}')$ as their corresponding Monte Carlo estimators. Using Taylor expansion, it is shown in Booth and Robert (1999) that

$$\begin{aligned} 0 &= Q_M^{(1)}(\boldsymbol{\lambda}^{(r+1)}, \boldsymbol{\lambda}^{(r)}) \\ &\approx Q_M^{(1)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)}) + (\boldsymbol{\lambda}^{(r+1)} - \boldsymbol{\lambda}^{*(r+1)})^T Q_M^{(2)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)}), \end{aligned} \tag{3.29}$$

where $\boldsymbol{\lambda}^{*(r+1)}$ satisfies $Q^{(1)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)}) = 0$. From the central limit theorem of samples generated by rejection sampling and IS, the Monte Carlo estimation for the first derivative $Q_M^{(1)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)})$ is approximately normal. Hence,

$$\boldsymbol{\lambda}^{(r+1)} \mid \boldsymbol{\lambda}^{(r)} \overset{a.}{\sim} N(\boldsymbol{\lambda}^{*(r+1)}, \sigma^2), \tag{3.30}$$

where

$$\begin{aligned} \sigma^2 &= \mathrm{var}(\boldsymbol{\lambda}^{(r+1)} \mid \boldsymbol{\lambda}^{(r)}) \\ &\approx Q_M^{(2)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)})^{-1} \mathrm{var}\{\mathrm{Q}_{\mathrm{M}}^{(1)}(\boldsymbol{\lambda}^{*(\mathrm{r}+1)}, \boldsymbol{\lambda}^{(\mathrm{r})})\} Q_M^{(2)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)})^{-1}. \end{aligned}$$

The central limit theorem for particles generated using the resample-move algorithm is proven in Chopin (2004), and Eq.(3.30) holds in this case. Because$\boldsymbol{\lambda}^{*(r+1)}$ is unavailable, it is replaced by $\boldsymbol{\lambda}^{(r+1)}$ when estimating $\mathrm{var}(\boldsymbol{\lambda}^{(r+1)} \mid \boldsymbol{\lambda}^{(r)})$ and the corresponding estimator is

$$\hat{\mathrm{var}}(\boldsymbol{\lambda}^{(r+1)} \mid \boldsymbol{\lambda}^{(r)}) = Q_M^{(2)}(\boldsymbol{\lambda}^{(r+1)}, \boldsymbol{\lambda}^{(r)})^{-1} \hat{\mathrm{var}}\{Q_M^{(1)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)})\} Q_M^{(2)}(\boldsymbol{\lambda}^{(r+1)}, \boldsymbol{\lambda}^{(r)})^{-1},$$

where $\hat{\mathrm{var}}\{Q_M^{(1)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)})\}$ is estimated by

$$\frac{1}{M^2} \sum_{j=1}^{M} \left(\omega_T^{(j,r)} \frac{\partial}{\partial \boldsymbol{\lambda}} \ln p(x_{0:T}^{(j,r)}, y_{0:T}, \boldsymbol{\theta}^{(j,r)} \mid \boldsymbol{\lambda}^{(r+1)}) \right) \left(\omega_T^{(j,r)} \frac{\partial}{\partial \boldsymbol{\lambda}} \ln p(x_{0:T}^{(j,r)}, y_{0:T}, \boldsymbol{\theta}^{(j,r)} \mid \boldsymbol{\lambda}^{(r+1)}) \right)^T$$

and $\{(x_{0:T}^{(j,r)}, \boldsymbol{\theta}^{(j,r)}), \omega_T^{(j,r)}\}$ are the weighted samples obtained from $p(x_{0:T}, \boldsymbol{\theta} \mid \boldsymbol{\lambda}^{(r)}, y_{0:T})$ using the resample-move algorithm. Therefore, at the $(r+1)$th iteration, an approximate $100(1-\alpha)\%$ confidence region of $\boldsymbol{\lambda}^{*(r+1)}$ is constructed by

$$\{\boldsymbol{\lambda}^{*(r+1)} : (\boldsymbol{\lambda}^{(r+1)} - \boldsymbol{\lambda}^{*(r+1)})^T \mathrm{v\hat{a}r}(\boldsymbol{\lambda}^{(r+1)} \mid \boldsymbol{\lambda}^{(r)})^{-1}(\boldsymbol{\lambda}^{(r+1)} - \boldsymbol{\lambda}^{*(r+1)}) \le \chi^2_\alpha(p)\}, \tag{3.31}$$

where it is assumed that the dimension of the hyper-parameters $\boldsymbol{\lambda}$ is p, and $\chi^2_\alpha(p)$ is the $(1-\alpha)$th quantile of χ^2 with p degrees of freedom. If $\boldsymbol{\lambda}^{(r)}$ falls into this region, the difference between the values in the adjacent iterations can be covered by Monte Carlo error, and the sample size M in the next iteration is increased. Otherwise, the sample size remains unchanged.

3.2.2 Particle Marginal Metropolis-Hastings Sampler

A special case of the PMCMC method, namely, the PIMH algorithm was introduced in Section 2.3.2. However, the target distribution becomes $p(x_{0:T}, \boldsymbol{\theta} \mid y_{0:T}, \boldsymbol{\lambda})$ which depends on particle smoothing and parameter estimation. To deal with this situation, a more general algorithm is required. To this end, the PMMH sampler is a viable alternative. For the ordinary Metropolis-Hastings (MH) algorithm, if the proposal distribution is

$$q(\boldsymbol{\theta}^*, x^*_{0:T} \mid \boldsymbol{\theta}, x_{0:T}) = q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta})p(x^*_{0:T} \mid \boldsymbol{\theta}^*, y_{0:T}), \tag{3.32}$$

then the candidate point $(\boldsymbol{\theta}^*, x^*_{0:T})$ is accepted with the probability equaling to $\min\{1, R\}$, where

$$\begin{aligned} R &= \frac{p(x^*_{0:T}, \boldsymbol{\theta}^* \mid y_{0:T}, \boldsymbol{\lambda},)q(\boldsymbol{\theta}, x_{0:T} \mid \boldsymbol{\theta}^*, x^*_{0:T})}{p(x_{0:T}, \boldsymbol{\theta} \mid y_{0:T}, \boldsymbol{\lambda})q(\boldsymbol{\theta}^*, x^*_{0:T} \mid \boldsymbol{\theta}, y_{0:T})} \\ &= \frac{p(x^*_{0:T} \mid \boldsymbol{\theta}^*, y_{0:T}, \boldsymbol{\lambda})p(\boldsymbol{\theta}^* \mid \boldsymbol{\lambda}, y_{0:T})q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^*)p(x_{0:T} \mid \boldsymbol{\theta}, y_{0:T})}{p(x_{0:T} \mid \boldsymbol{\theta}, y_{0:T}, \boldsymbol{\lambda})p(\boldsymbol{\theta} \mid \boldsymbol{\lambda}, y_{0:T})q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta})p(x^*_{0:T} \mid \boldsymbol{\theta}^*, y_{0:T})} \\ &= \frac{p_{\boldsymbol{\theta}^*}(y_{0:T})\pi(\boldsymbol{\theta}^* \mid \boldsymbol{\lambda})q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^*)}{p_{\boldsymbol{\theta}}(y_{0:T})\pi(\boldsymbol{\theta} \mid \boldsymbol{\lambda})q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta})}. \end{aligned} \tag{3.33}$$

By the PCMCM method, an auxiliary variable $W_{\boldsymbol{z}_T}$ is introduced and defined by

$$W_{\boldsymbol{z}_T} = \frac{\hat{p}_{\boldsymbol{\theta}}(y_{0:T})}{p_{\boldsymbol{\theta}}(y_{0:T})}. \tag{3.34}$$

Let the distribution of $W_{\boldsymbol{z}_T}$ be $Q_{\boldsymbol{z}_T}(\cdot)$. Then the joint distribution of $\boldsymbol{z}_T$ and $W_{\boldsymbol{z}_T}$ is given by

$$p(\boldsymbol{z}_T, w_{\boldsymbol{z}_T}) = p(x_{0:T}, \boldsymbol{\theta} \mid y_{0:T}, \boldsymbol{\lambda}) Q_{\boldsymbol{z}_T}(w_{\boldsymbol{z}_T}) w_{\boldsymbol{z}_T}. \tag{3.35}$$

Because $E[W_{\boldsymbol{z}_T}] = 1$, the marginal distribution of $\boldsymbol{z}_T$ is the original target distribution $p(x_{0:T}, \boldsymbol{\theta} \mid y_{0:T}, \boldsymbol{\lambda})$. If a sample is drawn from the joint distribution in Eq.(3.35), by discarding the auxiliary variable $w_{\boldsymbol{z}_T}$, a sample from the target distribution $p(x_{0:T}, \boldsymbol{\theta} \mid y_{0:T}, \boldsymbol{\lambda})$ is generated.

When the MH algorithm is used to deal with the joint distribution Eq.(3.35), the proposal distribution is

$$q(\boldsymbol{z}_T^*, w_{\boldsymbol{z}_T}^* \mid \boldsymbol{z}_T, w_{\boldsymbol{z}_T}) = q(\boldsymbol{\theta}^*, x_{0:T}^* \mid \boldsymbol{\theta}, x_{0:T}) Q_{\boldsymbol{z}_T^*}(w_{\boldsymbol{z}_T}^*), \tag{3.36}$$

It is derived in Douc et al. (2014) that the corresponding ratio of this MH algorithm becomes $R\frac{w_{\boldsymbol{z}_T}^*}{w_{\boldsymbol{z}_T}}$. Substituting $\frac{\hat{p}_{\boldsymbol{\theta}}(y_{0:T})}{p_{\boldsymbol{\theta}}(y_{0:T})}$ for $w_{\boldsymbol{z}_T}$, a new ratio is obtained as

$$R' = R\frac{w_{\boldsymbol{z}_T}^*}{w_{\boldsymbol{z}_T}} = \frac{\hat{p}_{\boldsymbol{\theta}^*}(y_{0:T})\pi(\boldsymbol{\theta}^* \mid \boldsymbol{\lambda})q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^*)}{\hat{p}_{\boldsymbol{\theta}}(y_{0:T})\pi(\boldsymbol{\theta} \mid \boldsymbol{\lambda})q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta})}. \tag{3.37}$$

According to Eq.(3.37), it is not necessary to generate $w_{\boldsymbol{z}_T}$. Instead, it only needs to obtain an unbiased estimator $\hat{p}_{\boldsymbol{\theta}}(y_{0:T})$, which can be achieved using the SMC method introduced in Section 2.3.2. Algorithm 9 summarizes the application of the PMMH algorithm in this case.

We obtain $Q_M(\boldsymbol{\lambda}, \boldsymbol{\lambda}^{(r)})$ from particles generated by Algorithm 9, and $\boldsymbol{\lambda}^{(r+1)}$ is obtained iteratively. The same stopping rule in Eq.(3.26) is used. Because the sample drawn by the PMMH algorithm is a Markov chain, the same criterion provided in Section 3.1 is used to decide if the sample size should be increased at each iteration. The confidence interval of relative likelihood at

Algorithm 9 The PMMH algorithm in NLSS models

```
for k=1 do
    set initial value $\boldsymbol{\theta}(1)$ arbitrarily;
    run a SMC algorithm targeting $\phi^{\boldsymbol{\theta}(1)}_{0:T|T}(x_{0:T})$, obtain the corresponding weighted particles $\{x^j_{0:T}, \omega^j_T\}^N_{j=1}$, and compute $\hat{p}_{\boldsymbol{\theta}(1)}(y_{1:T})$ according Eq.(2.23);
    draw index $I$ with the probability $P(I = j) = \tilde{\omega}^j_T, j = 1, \ldots, N$, where $\tilde{\omega}_T$ is the normalized weight;
    set $x_{0:T}(1) = x^I_{0:T}$.
end for
for  k=2, . . . , M do
    simulate $\boldsymbol{\theta}^* \sim q(\cdot \mid \boldsymbol{\theta}(k-1))$
    run a SMC algorithm targeting $\phi^{\boldsymbol{\theta}^*}_{0:T|T}(x_{0:T})$, obtain the corresponding weighted particles $\{x^j_{0:T}, \omega^j_T\}^N_{j=1}$, and compute $\hat{p}_{\boldsymbol{\theta}^*}(y_{1:T})$;
    calculate the acceptance rate
        $$\tilde{R} = \min\left\{1, \frac{\hat{p}_{\boldsymbol{\theta}^*}(y_{0:T})\pi(\boldsymbol{\theta}^* \mid \boldsymbol{\lambda})q(\boldsymbol{\theta}(k-1) \mid \boldsymbol{\theta}^*)}{\hat{p}_{\boldsymbol{\theta}(k-1)}(y_{0:T})\pi(\boldsymbol{\theta}(k-1) \mid \boldsymbol{\lambda})q(\boldsymbol{\theta}^* \mid \boldsymbol{\theta}(k-1))}\right\};$$
    draw $U$ uniformly on $[0, 1]$;
    if $U \leq \tilde{R}$, draw index $I$ with the probability $P(I = j) = \tilde{\omega}^j_T, j = 1, \ldots, N$, set $x_{0:T}(k) = x^I_{0:T}$, $\boldsymbol{\theta}(k) = \boldsymbol{\theta}^*$ and $\hat{p}_{\boldsymbol{\theta}(k)}(y_{1:T}) = \hat{p}_{\boldsymbol{\theta}^*}(y_{1:T})$; otherwise set $x_{0:T}(k) = x_{0:T}(k-1)$, $\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k-1)$ and $\hat{p}_{\boldsymbol{\theta}(k)}(y_{1:T}) = \hat{p}_{\boldsymbol{\theta}(k-1)}(y_{1:T})$.
end for
Burn the first $\lfloor M/c \rfloor$ steps, take $\{x_{0:T}(k), \boldsymbol{\theta}(k)\}^M_{k=\lfloor M/c \rfloor+1}$ as an approximation for $p(x_{0:T}, \boldsymbol{\theta} \mid y_{0:T}, \boldsymbol{\lambda})$, where $c$ is a positive constant.
```

each iteration is the most critical element to devise this criterion. The relative likelihood at the rth iteration can be expressed as

$$RL^{(r)} = E_{\boldsymbol{\lambda}^{(r-1)}}\left[\frac{f(x_{0:T}, y_{0:T}, \boldsymbol{\theta} \mid \boldsymbol{\lambda}^{(r)})}{f(x_{0:T}, y_{0:T}, \boldsymbol{\theta} \mid \boldsymbol{\lambda}^{(r-1)})}\middle| y_{0:T}\right]. \tag{3.38}$$

Define the function

$$h^{(r)}(\boldsymbol{z}_T) = \frac{f(\boldsymbol{z}_T, y_{0:T} \mid \boldsymbol{\lambda}^{(r)})}{f(\boldsymbol{z}_T, y_{0:T}, \mid \boldsymbol{\lambda}^{(r-1)})}, \tag{3.39}$$

then the estimator of $\mathrm{var}_{\boldsymbol{\lambda}^{(r-1)}}[h(\boldsymbol{z}_T) \mid y_{0:T}]$ is

$$\hat{\mathrm{var}}[h(\boldsymbol{z}_T)]^{(r)} = \frac{1}{M}\sum_{j=1}^{M}[h(\boldsymbol{z}_T^{(j,r-1)})]^2 - \left[\frac{1}{M}\sum_{j=1}^{M} h(\boldsymbol{z}_T^{(j,r-1)})\right]^2, \tag{3.40}$$

where $\{\boldsymbol{z}_T^{(j,r-1)} = (x_{0:T}^{(j,r-1)}, \boldsymbol{\theta}^{(j,r-1)})\}_{j=1}^{M}$ is the sample drawn from $p(x_{0:T}, \boldsymbol{\theta} \mid \boldsymbol{\lambda}^{(r-1)}, y_{0:T})$ by the PMMH algorithm. The sequence $\{t_k\}_{k=1}^{N_M}$ is generated using the same method as in Section 3.1. The estimator of the relative likelihood at the rth iteration can be written as

$$\begin{aligned}
\hat{RL}^{(r)} &= \frac{1}{N_M}\sum_{k=1}^{N_M}\frac{f(x_{0:T}^{(t_k,r-1)}, y_{0:T}, \boldsymbol{\theta}^{(t_k,r-1)} \mid \boldsymbol{\lambda}^{(r)})}{f(x_{0:T}^{(t_k,r-1)}, y_{0:T}, \boldsymbol{\theta}^{(t_k,r-1)} \mid \boldsymbol{\lambda}^{(r-1)})}\\
&= \frac{1}{N_M}\sum_{k=1}^{N_M}\frac{f(x_{0:T}^{(t_k,r-1)}, y_{0:T} \mid \boldsymbol{\theta}^{(t_k,r-1)})\pi(\boldsymbol{\theta}^{(t_k,r-1)} \mid \boldsymbol{\lambda}^{(r)})}{f(x_{0:T}^{(t_k,r-1)}, y_{0:T} \mid \boldsymbol{\theta}^{(t_k,r-1)})\pi(\boldsymbol{\theta}^{(t_k,r-1)} \mid \boldsymbol{\lambda}^{(r-1)})}\\
&= \frac{1}{N_M}\sum_{k=1}^{N_M}\frac{\pi(\boldsymbol{\theta}^{(t_k,r-1)} \mid \boldsymbol{\lambda}^{(r)})}{\pi(\boldsymbol{\theta}^{(t_k,r-1)} \mid \boldsymbol{\lambda}^{(r-1)})}, \qquad (3.41)
\end{aligned}$$

where $\{(x_{0:T}^{(t_k,r-1)}, \boldsymbol{\theta}^{(t_k,r-1)})\}_{k=1}^{N_M}$ is a subsample of $\{(x_{0:T}^{(j,r-1)}, \boldsymbol{\theta}^{(j,r-1)})\}_{j=1}^{M}$. An approximate $100(1-\alpha)\%$ confidence interval of $RL^{(r)}$ then becomes

$$\left[\hat{RL}^{(r)} - z_{1-\alpha/2}\sqrt{\hat{\mathrm{var}}[h(\boldsymbol{z}_T)]^{(r)}/N_M}, \hat{RL}^{(r)} + z_{1-\alpha/2}\sqrt{\hat{\mathrm{var}}[h(\boldsymbol{z}_T)]^{(r)}/N_M}\right]. \tag{3.42}$$

If this confidence interval includes 1, then increase the sample size in the next iteration. Otherwise, the sample size remains unchanged.

Chapter 4

Special Case: Inference for Stochastic Volatility Model

The stochastic volatility (SV) model is a popular model used in time series analysis. In the canonical discrete-time SV model, the return Y_t depends on the corresponding unobserved volatility X_t, and the sequence of volatilities is modeled using an AR process. The entire model can be formulated as the following two equations:

$$\begin{aligned} X_t &= \rho X_{t-1} + \eta_t, && (4.1) \\ Y_t &= \beta \text{exp}(X_t/2)\epsilon_t, && (4.2) \end{aligned}$$

where $\epsilon_t \overset{i.i.d.}{\sim} N(0,1)$, $\eta_t \overset{i.i.d.}{\sim} N(0,\tau)$. Herein, we assume that the initial value x_0 be a constant for simplicity. In this model, the transition kernel is

$$(X_t \mid X_{t-1} = x_{t-1}) \sim N(\rho x_{t-1}, \tau) \qquad (4.3)$$

and the distribution of the observation process is

$$(Y_t \mid X_t = x_t) \sim N[0, \beta^2 \text{exp}(y_t)] \qquad (4.4)$$

In this chapter, the methodology proposed in Chapter 3 is used to conduct inference for the SV model.

4.1 Maximum Likelihood Estimation for Stochastic Volatility Model

In the SV model, the complete likelihood is

$$p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta}) = (2\pi\tau)^{-T/2}\left(2\pi\beta^2\right)^{-T/2}\exp\left\{-\sum_{t=1}^{T}\left[\frac{1}{2\tau}(x_t-\rho x_{t-1})^2+\frac{x_t}{2}+\frac{y_t^2}{2\beta^2\exp(x_t)}\right]\right\}, \tag{4.5}$$

where $\boldsymbol{\theta} = (\rho, \tau, \beta^2)^T$, and the joint smoothing distribution $p(x_{0:T} \mid y_{0:T}, \boldsymbol{\theta})$ is proportional to this likelihood function $p(x_{0:T}, y_{0:T} \mid \boldsymbol{\theta})$. If the sample $\{x_{0:T}^i\}_{i=1}^M$ is drawn from the joint distribution, then from a starting point $\boldsymbol{\theta}^{(0)}$, the sequence of parameter $\{\boldsymbol{\theta}^{(r)}\}$ can be obtained using Eq.(3.6). In the following sections, sample generation, log-likelihood maximization and several other issues with the setting of Monte Carlo MLE for the SV model will be discussed.

4.1.1 Sample Generation

We now show how to generate $\{x_{0:T}^i\}_{i=1}^M$ from $p(x_{0:T} \mid y_{0:T}, \boldsymbol{\theta})$ by the PIMH algorithm. Running a SMC algorithm that target $p(x_{0:T} \mid y_{0:T}, \boldsymbol{\theta})$ is an important step of implementing the PIMH algorithm. The APF introduced in Section 2.2.4 is a useful tool in the SMC methodology, thus we choose the APF in the SMC step. Implementation of the APF in the SV model is discussed in Pitt and Shephard (1999) in which the proposal distribution is introduced as

$$q_2(x_{t+1} \mid x_t^k) = N\left[\rho x_t^k + \frac{\tau}{2}\left\{\frac{y_{t+1}^2}{\beta^2}\exp(-\rho x_t^k) - 1\right\}, \tau\right]. \tag{4.6}$$

However, this proposal distribution becomes difficult to simulate when the observations $y_{0:t}$ consist of large value outliers. Because y_{t+1} has a quadratic effect in the mean term of the proposal distribution of the latent variable x_{t+1}, large values of y_{t+1} make the mean term have a large drift from ρx_t^k. As a result, the state variables $x_{0:T}$ are overestimated using this algorithm, which results in a bad estimator for τ. Hence, within the framework of the APF, another proposal distribution proposed in Douc et al. (2009) is used.

Assume that the particle weights at time t are all equal to 1. Then the first-stage unnormalized weights are

$$\tilde{\lambda}_{t+1}^{k} = \bar{\sigma}_t(x_t^k) g[y_{t+1} \mid \bar{m}_t(x_t^k)] f[\bar{m}_t(x_t^k) \mid x_t^k], \tag{4.7}$$

for $k = 1, \ldots, N$, where $\bar{m}_t(x_t^k)$ is the mode of $\log[g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t^k)]$ with respect to x_{t+1} and hence is the unique solution of

$$-\frac{1}{\tau}(x - \rho x_t^k) + \frac{y_{t+1}^2}{2\beta^2}\exp(-x) - \frac{1}{2} = 0, \tag{4.8}$$

and

$$\bar{\sigma}_t^2(x_t^k) = \left\{ \frac{1}{\tau} + \frac{y_{t+1}^2}{2\beta^2}\exp[-\bar{m}_t(x_t^k)] \right\}^{-1}. \tag{4.9}$$

Define the function

$$f_{t+1}^k(x) = -\frac{1}{\tau}(x - \rho x_t^k) + \frac{y_{t+1}^2}{2\beta^2}\exp(-x) - \frac{1}{2}, \tag{4.10}$$

since the first derivative of $f_{t+1}^k(x)$

$$f_{t+1}^{k}{}'(x) = -\frac{y_{t+1}^2}{2\beta^2}\exp(-x) - \frac{1}{\tau} < 0, \tag{4.11}$$

the Newton-Raphson method can be used to solve Eq.(4.8). The proposal distribution becomes

$$q_2(x_{t+1} \mid x_t^k) = N\left[\bar{m}_t(x_t^k), \bar{\sigma}_t^2(x_t^k)\right], \tag{4.12}$$

and the second-stage unnormalized weights become

$$\bar{\omega}_{t+1} = \frac{g(y_{t+1} \mid x_{t+1}) f(x_{t+1} \mid x_t^k)}{\bar{\sigma}_t(x_t^k) g[y_{t+1} \mid \bar{m}_t(x_t^k)] f[\bar{m}_t(x_t^k) \mid x_t^k] p_N[x_{t+1}; \bar{m}_t(x_t^k), \bar{\sigma}_t^2(x_t^k)]}, \tag{4.13}$$

where $p_N(\cdot; \mu, \sigma^2)$ is the density function of a normal distribution with mean μ and variance σ^2. Since the target distribution is the joint smoothing distribution rather than the filtering distribution, the APF algorithm is different from Algorithm 4, and this new method is given in Algorithm 10.

Algorithm 10 The auxiliary particle filter in the SV model

```
Ensure: $\{\tilde{x}^i_{0:t}\}_{i=1}^N$ approximates $\phi_{0:t|t}(x_{0:t})$.
Compute the first-stage unnormalized weights $\tilde{\lambda}^k_{t+1}$ by Eq.(4.7), $k = 1, \ldots, N$.
for i=1, ..., N do
    draw index $k^i$ from the set $\{1, \ldots, N\}$ with respect to the probability
        $P(k^i = j) = \frac{\tilde{\lambda}^j_{t+1}}{\sum_{m=1}^N \tilde{\lambda}^m_{t+1}}, j = 1, \ldots, N;$
    sample $\tilde{x}^i_{t+1} \sim q_2(\cdot \mid x^{k^i}_t)$;
    calculate the second-stage unnormalized weights $\bar{\omega}^i_{t+1}$ by Eq.(4.13).
end for
Compute the second-stage normalized weights
    $\tilde{\omega}^j_{t+1} = \frac{\bar{\omega}^j_{t+1}}{\sum_{m=1}^N \bar{\omega}^m_{t+1}}, j = 1, \ldots, N;$
draw index $I^1_{t+1}, \ldots, I^N_{t+1}$ independently with the probability
    $P(I_{t+1} = j) = \tilde{\omega}_{t+1}, j = 1, \ldots, N;$
set $x^i_{0:t+1} = \tilde{x}^{I^i_{t+1}}_{0:t+1}$ for $i = 1, \ldots, N$.
Take $\{x^i_{0:t+1}\}_{i=1}^N$ as an approximation for $\phi_{0:t+1|t+1}(x_{0:t+1})$.
```

Another key issue in the implementation of the PIMH algorithm is to obtain the estimator of the likelihood $p_{\boldsymbol{\theta}}(y_{0:T})$. The estimator in Eq.(2.23) is based on the simplest SMC method (i.e. SISR algorithm). However, in the framework of the APF method, there are two weights. Hence, Pitt and Silva (2012) proposes an unbiased estimator of $p_{\boldsymbol{\theta}}(y_i \mid y_{0:i-1})$ as

$$\hat{p}_{\boldsymbol{\theta}}(y_i \mid y_{0:i-1}) = \left\{ \sum_{k=1}^{N} \frac{\bar{\omega}_{t+1}^{i}}{N} \right\} \left\{ \sum_{k=1}^{N} \tilde{\lambda}_{t+1}^{k} \right\}. \tag{4.14}$$

With the above APF algorithm and the estimator of the likelihood, we can use the PIMH method in Algorithm 7 to estimate the MLE for the SV model.

4.1.2 Maximization Procedure

From Eq.(4.5), it is seen that the parameter β^2 is independent of the other two parameters ρ and τ, and these two groups of parameters can be considered separately. If $\{x_{0:T}^{i}\}_{i=1}^{M}$ is generated from $p(x_{0:T} \mid y_{0:T}, \boldsymbol{\theta}')$, then the estimation of parameters is obtained by

$$(\rho, \tau) = \arg\max_{\rho,\tau} \frac{1}{M} \sum_{i=1}^{M} \left[-\frac{T}{2} \log(\tau) - \frac{1}{2\tau} \sum_{t=1}^{T} (x_t^i - \rho x_{t-1}^i)^2 \right] \tag{4.15}$$

and

$$\beta^2 = \arg\max_{\beta^2} \frac{1}{M} \sum_{i=1}^{M} \left[-\frac{T}{2} \log(\beta^2) - \sum_{t=1}^{T} \frac{y_t^2}{2\beta^2 \exp(x_t^i)} \right]. \tag{4.16}$$

Set the first derivative of the target functions equal to zero and solve the equations, we have

$$\begin{aligned} \hat{\rho} &= \frac{\sum_{t=1}^{T} \sum_{i=1}^{M} x_t^i x_{t-1}^i}{\sum_{t=1}^{T} \sum_{i=1}^{M} (x_{t-1}^i)^2} \\ &= \frac{\sum_{t=1}^{T} (\hat{x}_{t-1}\hat{x}_t + \hat{P}_{t-1,t})}{\sum_{t=1}^{T} (\hat{x}_{t-1}^2 + \hat{P}_{t-1})}, \end{aligned} \tag{4.17}$$

$$\begin{aligned} \hat{\tau} &= \frac{1}{MT} \sum_{t=1}^{T} \sum_{i=1}^{M} (x_t^i - \hat{\rho} x_{t-1}^i)^2 \\ &= \frac{1}{T} \left\{ \sum_{t=1}^{T} (\hat{x}_t^2 + \hat{P}_t) - \frac{\left[\sum_{t=1}^{T} (\hat{x}_{t-1}\hat{x}_t + \hat{P}_{t-1,t}) \right]^2}{\sum_{t=1}^{T} (\hat{x}_{t-1}^2 + \hat{P}_{t-1})} \right\}, \end{aligned} \tag{4.18}$$

and

$$\hat{\beta}^2 = \frac{1}{MT} \sum_{t=1}^{T} \sum_{i=1}^{M} \frac{y_t^2}{\exp(x_t^i)}, \tag{4.19}$$

where $\hat{x}_t = \frac{1}{M}\sum_{i=1}^{M} x_t^i$, $\hat{P}_t = \frac{1}{M}\sum_{i=1}^{M}(x_t^i - \hat{x}_t)^2$ and $\hat{P}_{t-1,t} = \frac{1}{M}\sum_{i=1}^{M}(x_{t-1}^i - \hat{x}_{t-1})(x_t^i - \hat{x}_t)$ for $t = 1, \ldots, T$.

4.1.3 Other Issues

A disadvantage of the EM algorithm is its slow convergence rate, thus it is necessary to find a good starting value. As in Kim (2005), the moment method can be used to calculate the initial value of parameters. For parameters ρ and τ, we use similar starting points as used Kim (2005). Take $v_t = \log(y_t^2)$, then

$$\rho^{(0)} = \text{sign}\left[\frac{\sum_{t=3}^{T}(v_t - \bar{v})(v_{t-2} - \bar{v})}{\sum_{t=3}^{T}(v_{t-1} - \bar{v})(v_{t-2} - \bar{v})}\right] \cdot \min\left[\frac{\sum_{t=3}^{T}(v_t - \bar{v})(v_{t-2} - \bar{v})}{\sum_{t=3}^{T}(v_{t-1} - \bar{v})(v_{t-2} - \bar{v})}, 0.99\right] \tag{4.20}$$

and

$$\tau^{(0)} = \max\left\{\frac{1}{n}\sum_{t=2}^{T}\left[(v_t - \bar{v}) - \rho^{(0)}(v_{t-1} - \bar{v})\right]^2 - \hat{\sigma}^2[1 + (\rho^{(0)})^2], 0.01\right\} \tag{4.21}$$

where $\bar{v}$ is the average of $v_{0:T}$ and $\hat{\sigma}^2$ is set as 5, which approximates the variance of the log-χ^2 distrbution. The term 0.99 is included in Eq.(4.20) to ensure the stationarity of the process $\{x_t\}$, and the term 0.01 in Eq.(4.21) ensures the positive variance. For parameter β^2, we square and take the logarithm of Eq.(4.2) to get

$$v_t = \log(\beta^2) + x_t + \log(\epsilon_t^2). \tag{4.22}$$

By taking the expectation of Eq.(4.22), we have

$$\log(\beta^2) = E(v_t) - E(x_t) - E[\log(\epsilon_t^2)]. \tag{4.23}$$

Due to the stationarity, $E(x_t) = 0$, the initial value of β^2 is set as

$$(\beta^2)^{(0)} = \exp(\bar{v} + 1.3), \tag{4.24}$$

where -1.3 approximates the expectation of of the log-χ^2 distrbution.

The stopping rule and the sample size issue in this case are dealt with in Section 3.1.1 and Section 3.1.2 respectively. The MCEM algorithm for the MLE of the SV model is given in Algorithm 11, and the proof of the convergence of this algorithm is presented in Appendix A.1.

4.2 Bayesian Analysis for the Stochastic Volatility Model

We now discuss the application of Bayesian analysis with type II maximum likelihood prior to the SV model. In the setting of priors, it is assumed that the priors of the parameters are independent. Because ρ is supposed to lie in $(-1, 1)$ to ensure the stationarity of the sequence $X_{0:T}$, we take ρ to be uniformly distributed in the interval $(-1, 1)$, denoted as $\rho \sim U(-1, 1)$. For the parameters τ and β, the conjugate priors are chosen, which are inverse Gamma distributions, and denoted as

$$\tau \sim IG(\alpha_0, \gamma), \tag{4.25}$$

and

$$\beta^2 \sim IG(\mu_0, \delta) \tag{4.26}$$

respectively, where the shape parameters α_0 and μ_0 are given, and the scale parameters γ and δ are hyper-parameters. If the maximization involves unknown parameters α_0 and μ_0, further numerical procedure are needed and it becomes more difficult and time-consuming to achieve convergence. Only γ and δ are unknown in this study.

In this case, the goal is to find the hyper-parameters γ and δ that maximize the marginal distribution, meaning

$$\arg\max_{\gamma,\delta} L(\gamma, \delta \mid y_{0:T}) = \arg\max_{\gamma,\delta} p(y_{0:T} \mid \gamma, \delta). \tag{4.27}$$

Algorithm 11 The MCEM algorithm for MLE of the SV model

```
Ensure: The initial value $\boldsymbol{\theta}^{(0)} = (\rho^{(0)}, \tau^{(0)}, (\beta^2)^{(0)})^T$ is obtained by Eq.(4.20), Eq.(4.21) and Eq.(4.24) respectively; and set $r = 0$ and $M^{(0)} = C_0$, where $C_0$ is a positive integer.
repeat
    for k=1 ..., $M^{(r)}$ do
        run Algorithm 10 targeting $\phi^{\boldsymbol{\theta}^{(r)}}_{0:T|T}(x_{0:T})$, obtain the corresponding weighted particles $\{x^j_{0:T}, \omega^j_T\}^N_{j=1}$, and compute $\hat{p}_{\boldsymbol{\theta}^{(r)}}(y_{1:T})^*$ by Eq.(2.23) with $\hat{p}_{\boldsymbol{\theta}}(y_i \mid y_{0:i-1})$ estimated by Eq.(4.14);
        calculate the acceptance rate
            $$r = \begin{cases} 1 & \text{for } k = 1 \\ \min\{1, \frac{\hat{p}_{\boldsymbol{\theta}^{(r)}}(y_{1:T})^*}{\hat{p}_{\boldsymbol{\theta}^{(r)}}(y_{1:T})(k-1)}\} & \text{for } k = 2, \dots, M^{(r)} \end{cases};$$
        draw $U$ uniformly on $[0, 1]$;
        if $U \leq r$, draw index $I$ with the probability $P(I = j) = \tilde{\omega}^j_T, j = 1, \dots, N$, where $\tilde{\omega}_T$ is the normalized weight, set $x_{0:T}(k) = x^I_{0:T}$ and $\hat{p}_\theta(y_{1:T})(k) = \hat{p}_\theta(y_{1:T})^*$; otherwise set $x_{0:T}(k) = x_{0:T}(k-1)$ and $\hat{p}_\theta(y_{1:T})(k) = \hat{p}_\theta(y_{1:T})(k-1)$.
    end for
    Burn the first $\lfloor M^{(r)}/c \rfloor$ steps, take $\{x_{0:T}(k)\}^{M^{(r)}}_{k=\lfloor M^{(r)}/c \rfloor + 1}$ as an approximation for $\phi^{\boldsymbol{\theta}^{(r)}}_{0:T|T}(x_{0:T})$, where $c$ is a positive constant.
    Calculate the estimator $\boldsymbol{\theta}^{(r+1)} = (\rho^{(r+1)}, \tau^{(r+1)}, (\beta^2)^{(r+1)})^T$ by Eq.(4.17)-Eq.(4.19) respectively;
    set $r = r + 1$;
    calculate the estimator of variance of relative likelihood by Eq.(3.15) ;
    obtain the subsample $\{x_{0:T}(t_k)\}^{N_{M^{(r)}}}_{k=1}$ by renewal theory in Section 3.1.2, and then calculate the estimator of $RL^{(r)}$ by Eq.(3.14);
    construct the $100(1-\alpha)\%$ confidence region of $RL^{(r)}$ by Eq.(3.16);
    if this interval contains 1, then $M^{(r)} = cM^{(r-1)}$ where $c > 1$, otherwise $M^{(r)} = M^{(r-1)}$.
until satisfying Eq.(3.7) for three adjacent iterations.
```

The MCEM algorithm is used to deal with this problem. Because the priors of τ and β^2 are independent, according to Eq.(3.22), the sequences $\{\gamma^{(r)}\}$ and $\{\delta^{(r)}\}$ are generated by

$$\begin{aligned}
\gamma^{(r+1)} &= \arg\max_{\gamma} \frac{1}{M}\sum_{j=1}^{M}\log\left[\frac{\gamma^{\alpha_0}}{\Gamma(\alpha_0)}(\tau^{(j,r)})^{-\alpha_0-1}\exp\left(-\frac{\gamma}{\tau^{(j,r)}}\right)\right] \\
&= \arg\max_{\gamma} \frac{1}{M}\left[M\alpha_0\log(\gamma)-(\alpha_0+1)\sum_{j=1}^{M}\log(\tau^{(j,r)})-\sum_{j=1}^{M}\frac{\gamma}{\tau^{(j,r)}}\right] \\
&= M\alpha_0/\sum_{j=1}^{M}\frac{1}{\tau^{(j,r)}} \qquad (4.28)
\end{aligned}$$

and

$$\begin{aligned}
\delta^{(r+1)} &= \arg\max_{\delta} \frac{1}{M}\sum_{j=1}^{M}\log\left[\frac{\delta^{\mu_0}}{\Gamma(\mu_0)}((\beta^2)^{(j,r)})^{-(\delta_0+1)}\exp\left(-\frac{\delta}{(\beta^2)^{(j,r)}}\right)\right] \\
&= \arg\max_{\delta} \frac{1}{M}\left[M\mu_0\log(\delta)-(\mu_0+1)\sum_{j=1}^{M}\log((\beta^2)^{(j,r)})-\sum_{j=1}^{M}\frac{\delta}{(\beta^2)^{(j,r)}}\right] \\
&= M\mu_0/\sum_{j=1}^{M}\frac{1}{(\beta^2)^{(j,r)}}, \qquad (4.29)
\end{aligned}$$

where $\{\tau^{(j,r)},(\beta^2)^{(j,r)}\}_{j=1}^{M}$ are sampled from $p(\tau,\beta^2 \mid \gamma^{(r)},\delta^{(r)},y_{0:T})$. To obtain this sample, draw particles $\{x_{0:T}^{(j,r)},\tau^{(j,r)},(\beta^2)^{(j,r)}\}_{j=1}^{M}$ from the joint distribution $p(x_{0:T},\tau,\beta^2 \mid \gamma^{(r)},\delta^{(r)},y_{0:T})$. In the remaining section, the application of the resample-move algorithm and the PMMH algorithm in this setting are presented.

4.2.1 Resample-move Algorithm in the Stochastic Volatility Model

In this case, the vector including all unknown variables at time t is denoted by

$$\boldsymbol{z}_t = (x_{0:t},\rho,\tau,\beta^2), \qquad (4.30)$$

and the target distribution is

$$\pi_t(\boldsymbol{z}_t) = p(\boldsymbol{z}_t \mid \gamma,\delta,y_{0:t}). \qquad (4.31)$$

For the resample-move algorithm, the intuitive transition distribution $f(x_{t+1} \mid x_t, \rho, \tau)$ is chosen to draw the new-coming variable x_{t+1} in the augmentation step. Then in the resampling step, the importance weight is

$$\tilde{\omega}_{t+1} \propto \omega_t g(y_{t+1} \mid x_{t+1}, \beta^2). \tag{4.32}$$

In the MCMC step, Gibbs sampling and the Metropolis-Hastings (M-H) algorithm are combined to perform a one-step move, and Algorithm 12 presents the corresponding procedure.

The resample-move algorithm is used to obtain the sample $\{x_{0:T}^{(j,r)}, \rho^{j,r}, \tau^{j,r}, (\beta^2)^{(j,r)}\}_{j=1}^M$ from the joint distribution $p(\boldsymbol{z}_T \mid \gamma^{(r)}, \delta^{(r)}, y_{0:T})$ at each iteration. Then the sample can be used to obtain $\gamma^{(r)}$ and $\delta^{(r)}$, as well as to construct the confidence region of $(\gamma^{*(r+1)}, \delta^{*(r+1)})$. Define $\boldsymbol{\lambda} = (\gamma, \delta)^T$, then

$$Q_M^{(2)}(\boldsymbol{\lambda}^{(r+1)}, \boldsymbol{\lambda}^{(r)}) = \begin{pmatrix} -\frac{\alpha_0}{(\gamma^{(r+1)})^2} & 0 \\ 0 & -\frac{\mu_0}{(\delta^{(r+1)})^2} \end{pmatrix} \tag{4.33}$$

and $\hat{\text{var}}\{Q_M^{(1)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)})\}$ is given by

$$\frac{1}{M^2}\sum_{j=1}^{M}\begin{pmatrix} (\frac{\alpha_0}{\gamma^{(r+1)}} - \frac{1}{\tau^{(j,r)}})^2 & (\frac{\alpha_0}{\gamma^{(r+1)}} - \frac{1}{\tau^{(j,r)}})(\frac{\mu_0}{\delta^{(r+1)}} - \frac{1}{(\beta^2)^{(j,r)}}) \\ (\frac{\alpha_0}{\gamma^{(r+1)}} - \frac{1}{\tau^{(j,r)}})(\frac{\mu_0}{\delta^{(r+1)}} - \frac{1}{(\beta^2)^{(j,r)}}) & (\frac{\mu_0}{\delta^{(r+1)}} - \frac{1}{(\beta^2)^{(j,r)}})^2 \end{pmatrix}.$$

Therefore, the estimator of $\text{var}(\boldsymbol{\lambda}^{(r+1)} \mid \boldsymbol{\lambda}^{(r)})$ is calculated by

$$\hat{\text{var}}(\boldsymbol{\lambda}^{(r+1)} \mid \boldsymbol{\lambda}^{(r)}) = Q_M^{(2)}(\boldsymbol{\lambda}^{(r+1)}, \boldsymbol{\lambda}^{(r)})^{-1}\hat{\text{var}}\{Q_M^{(1)}(\boldsymbol{\lambda}^{*(r+1)}, \boldsymbol{\lambda}^{(r)})\}Q_M^{(2)}(\boldsymbol{\lambda}^{(r+1)}, \boldsymbol{\lambda}^{(r)})^{-1},$$

and an approximate $100(1-\alpha)\%$ confidence region of $\boldsymbol{\lambda}^{*(r+1)}$ at the $(r+1)$th iteration is then constructed using Eq.(3.31), and whether to increase the sample size at next iteration is determined according to this confidence region. The entire procedure is presented in Algorithm 13.

4.2.2 Particle Marginal Metropolis-Hastings Sampler in the Stochastic Volatility Model

Two settings of the PMMH sampler in the SV model are presented. The first setting is the proposal distributions of unknown parameters ρ, τ and

Algorithm 12 MCMC step in the Resample-move algorithm for the SV model

```
Ensure: $\{\tilde{\boldsymbol{z}}_j^i\}_{i=1}^N$ approximates $\pi_j(\boldsymbol{z}_j \mid \gamma, \delta, y_{0:j})$ by SIS.
for i=1, . . . , N do
    simulate
        $\rho^i \sim N\left(\frac{\sum_{t=1}^{j} \tilde{x}_t^i \tilde{x}_{t-1}^i}{\sum_{t=1}^{j} (\tilde{x}_{t-1}^i)^2}, \frac{\tilde{\tau}^i}{\sum_{t=1}^{j} (\tilde{x}_{t-1})^2}\right)$ within $(-1, 1)$;
    simulate
        $\tau^i \sim IG\left(\frac{T}{2} + \alpha_0, \gamma + \frac{1}{2}\sum_{t=1}^{j} (\tilde{x}_t^i - \rho^i \tilde{x}_{t-1}^i)^2\right);$
    simulate
        $(\beta^2)^i \sim IG\left(\frac{T}{2} + \mu_0, \delta + \frac{1}{2}\sum_{t=1}^{j} \frac{y_t^2}{\exp(\tilde{x}_t^i)}\right);$
    for t=1, . . . , j-1 do
        simulate
            $\bar{x}_t^i \sim N\left(\frac{\rho^i}{1+(\rho^i)^2}(x_{t-1}^i + \tilde{x}_{t+1}^i), \frac{\tau^i}{1+(\rho^i)^2}\right);$
        calculate the acceptance rate
            $R = \min\left\{1, \frac{g(y_t \mid \bar{x}_t^i, (\beta^2)^i)}{g(y_t \mid \tilde{x}_t^i, (\beta^2)^i)}\right\};$
        draw $U$ uniformly on $[0, 1]$;
        if $U \leq R$, set $x_t^i = \bar{x}_t^i$ ; otherwise set $x_t^i = \tilde{x}_t^i$;
    end for
    simulate
        $\bar{x}_j^i \sim N\left(\rho^i x_{j-1}^i, \tau^i\right);$
    calculate acceptance rate
        $R = \min\left\{1, \frac{g(y_j \mid \bar{x}_T^i, (\beta^2)^i)}{g(y_j \mid \tilde{x}_T^i, (\beta^2)^i)}\right\};$
    draw $U$ uniformly on $[0, 1]$;
    if $U \leq R$, set $x_j^i = \bar{x}_j^i$ ; otherwise set $x_j^i = \tilde{x}_j^i$;
end for
take $\{\boldsymbol{z}_j^i\}_{i=1}^N$ to approximate $\pi_j(\boldsymbol{z}_j \mid \gamma, \delta, y_{0:j})$.
```

Algorithm 13 The MCEM alorithm with the resample-move algorithm in the SV model

```
Ensure: the initial value is given by $(\gamma^{(0)}, \delta^{(0)})^T$, set $r = 0$ and $M^{(0)} = C_0$,
    where $C_0$ is a positive integer.
repeat
    for i=1, . . . , N do
        simulate $\bar{\rho}^{(i,r)} \sim U[-1, 1]$, where U is uniform distribution;
        simulate
            $\bar{\tau}^{(i,r)} \sim IG(\alpha_0, \gamma^{(r)});$
        simulate
            $(\bar{\beta}^2)^{(i,r)} \sim IG(\mu_0, \delta^{(r)});$
        simulate
            $\bar{x}_1^{(i,r)} \sim f(\cdot \mid x_0, \bar{\rho}^{(i,r)}, \bar{\tau}^{(i,r)});$
        set $\bar{\boldsymbol{z}}_1^{(i,r)} = (\bar{\rho}^{(i,r)}, \bar{\tau}^{(i,r)}, (\bar{\beta}^2)^{(i,r)}, \bar{x}_1^{(i,r)});$
    end for
    get $\{\tilde{\boldsymbol{z}}_1^{(i,r)}, \omega_1^{(i,r)}\}_{i=1}^N$ by resample step in Algorithm 8;
    get $\{\boldsymbol{z}_1^{(i,r)}\}_{i=1}^N$ by Algorithm 12;
    for t=2, . . . , T do
        get $\{\boldsymbol{z}_t^{(i,r)}, \omega_t^{(i,r)}\}_{i=1}^N$ by Algorithm 8, where the MCMC step is conducted by Algorithm 12;
    end for
    calculate the estimator
        $\gamma^{(r+1)} = M^{(r)}\alpha_0 / \sum_{j=1}^{M^{(r)}} \frac{1}{\tau^{(j,r)}};$
    calculate the estimator
        $\delta^{(r+1)} = M^{(r)}\mu_0 / \sum_{j=1}^{M^{(r)}} \frac{1}{(\beta^2)^{(j,r)}};$
    construct an approximate $100(1-\alpha)\%$ confidence region of $\boldsymbol{\lambda}^{*(r+1)}$ by Eq.(3.31);
    if $(\gamma^{(r)}, \delta^{(r)})$ falls into this region, then $M^{(r+1)} = cM^{(r)}$ where $c > 1$,
        otherwise $M^{(r+1)} = M^{(r)}$ ;
    set $r = r + 1$.
until satisfying Eq.(3.26) for three adjacent iterations.
```

β^2. The simplest way is to select the prior distributions to be the proposal distributions, which draws the candidate point independently of the previous point. Therefore, the ratio of this algorithm becomes

$$\begin{aligned} R' &= \frac{\hat{p}_{\boldsymbol{\theta}^*}(y_{0:T})\pi(\boldsymbol{\theta}^* \mid \boldsymbol{\lambda})\pi(\boldsymbol{\theta} \mid \boldsymbol{\lambda})}{\hat{p}_{\boldsymbol{\theta}}(y_{0:T})\pi(\boldsymbol{\theta} \mid \boldsymbol{\lambda})\pi(\boldsymbol{\theta}^* \mid \boldsymbol{\lambda})} \\ &= \frac{\hat{p}_{\boldsymbol{\theta}^*}(y_{0:T})}{\hat{p}_{\boldsymbol{\theta}}(y_{0:T})}, \end{aligned} \tag{4.34}$$

which is the same as that of the PIMH algorithm. The other setting is to estimate the relative likelihood and its corresponding variance at each iteration. Again, the subsample generated using renewal theory is used to estimate the relative likelihood at the rth iteration

$$\begin{aligned} \hat{RL}^{(r)} &= \frac{1}{m}\sum_{k=1}^{m} \frac{\pi(\boldsymbol{\theta}^{(t_k,r-1)} \mid \boldsymbol{\lambda}^{(r)})}{\pi(\boldsymbol{\theta}^{(t_k,r-1)} \mid \boldsymbol{\lambda}^{(r-1)})} \\ &= \frac{1}{m}\sum_{k=1}^{m} \frac{p_{IG}(\tau^{(t_k,r-1)}; \alpha_0, \gamma^{(r)})p_{IG}((\beta^2)^{(t_k,r-1)}; \mu_0, \delta^{(r)})}{p_{IG}(\tau^{(t_k,r-1)}; \alpha_0, \gamma^{(r-1)})p_{IG}((\beta^2)^{(t_k,r-1)}; \mu_0, \delta^{(r-1)})}, \end{aligned} \tag{4.35}$$

where $p_{IG}(\cdot; a, b)$ is the density function of inverse gamma distribution with shape parameter a and scale parameter b. Now the function

$$\begin{aligned} h^{(r)}(\boldsymbol{z}_T) &= \frac{f(\boldsymbol{z}_T, y_{0:T} \mid \boldsymbol{\lambda}^{(r)})}{f(\boldsymbol{z}_T, y_{0:T}, \mid \boldsymbol{\lambda}^{(r-1)})} \\ &= \frac{p_{IG}(\tau; \alpha_0, \gamma^{(r)})p_{IG}(\beta^2; \mu_0, \delta^{(r)})}{p_{IG}(\tau; \alpha_0, \gamma^{(r-1)})p_{IG}(\beta^2; \mu_0, \delta^{(r-1)})}; \end{aligned} \tag{4.36}$$

can be substituted into Eq.(3.40) to obtain the estimator of variance of relative likelihood at the rth iteration. Then a confidence interval of relative likelihood at this iteration is constructed by Eq.(3.42), and whether to increase the sample size at the next iteration is determined based on this confidence interval. The whole procedure is presented in Algorithm 14, and the proof of the convergence of this algorithm is presented in Appendix A.2.

Algorithm 14 The MCEM algorithm with the PMMH algorithm in the SV model

```
Ensure: The initial value is given by $\boldsymbol{\lambda}^{(0)} = (\gamma^{(0)}, \delta^{(0)})^T$, and set $r = 0$ and
    $M^{(0)} = C_0$, where $C_0$ is a positive integer.
repeat
    for k=1, ..., $M^{(r)}$ do
        simulate $\rho^* \sim U[-1, 1]$, where U is uniform distribution; simulate
        $\tau^* \sim IG(\alpha_0, \gamma^{(r)})$ and $(\beta^2)^* \sim IG(\mu_0, \delta^{(r)})$; set $\boldsymbol{\theta}^* = (\rho^*, \tau^*, (\beta^2)^*)$;
        run Algorithm 10 targeting $\phi^{\boldsymbol{\theta}^*}_{0:T|T}(x_{0:T})$, obtain the corresponding
        weighted particles $\{x^j_{0:T}, \omega^j_T\}^N_{j=1}$, and compute $\hat{p}_{\boldsymbol{\theta}^*}(y_{1:T})^*$ by Eq.(2.23)
        with $\hat{p}_{\boldsymbol{\theta}}(y_i \mid y_{0:i-1})$ estimated by Eq.(4.14);
        calculate the acceptance rate
            $\tilde{R} = \begin{cases} 1 & \text{for } k = 1 \\ \min\left\{1, \frac{\hat{p}_{\boldsymbol{\theta}^*}(y_{0:T})}{\hat{p}_{\boldsymbol{\theta}(k-1)}(y_{0:T})}\right\} & \text{for } k = 2, \dots, M^{(r)} \end{cases};$
        draw $U$ uniformly on $[0, 1]$;
        if $U \leq \tilde{R}$, draw index $I$ with the probability $P(I = j) = \tilde{\omega}^j_T, j =$
        $1, \dots, N$, set $x_{0:T}(k) = x^I_{0:T}$, $\boldsymbol{\theta}(k) = \boldsymbol{\theta}^*$ and $\hat{p}_{\boldsymbol{\theta}(k)}(y_{1:T}) = \hat{p}_{\boldsymbol{\theta}^*}(y_{1:T})$;
        otherwise set $x_{0:T}(k) = x_{0:T}(k-1)$, $\boldsymbol{\theta}(k) = \boldsymbol{\theta}(k-1)$ and $\hat{p}_{\boldsymbol{\theta}(k)}(y_{1:T}) =$
        $\hat{p}_{\boldsymbol{\theta}(k-1)}(y_{1:T})$.
    end for
    Burn the first $\lfloor M^{(r)}/c \rfloor$ steps, take $\{x_{0:t}(k), \boldsymbol{\theta}(k)\}^{M^{(r)}}_{k=\lfloor M^{(r)}/c\rfloor+1}$ as an ap-
    proximation for $p(x_{0:T}, \boldsymbol{\theta} \mid y_{0:T}, \boldsymbol{\lambda}^{(r)})$, where $c$ is a positive constant,
    specially, set $\{\tau^{(j,r)}, (\beta^2)^{(j,r)}\}^M_{j=1} = \{\tau(k), \beta^2(k)\}^{M^{(r)}}_{k=\lfloor M^{(r)}/c\rfloor+1}$;
    calculate the estimator
        $\gamma^{(r+1)} = M\alpha_0 / \sum_{j=1}^{M} \frac{1}{\tau^{(j,r)}};$
    calculate the estimator
        $\delta^{(r+1)} = M\mu_0 / \sum_{j=1}^{M} \frac{1}{(\beta^2)^{(j,r)}};$
    set $r = r + 1$;
    calculate the estimator of variance of relative likelihood $\hat{\mathrm{var}}[h(\boldsymbol{z}_T)]^{(r)}$;
    obtain the subsample $\{\boldsymbol{\theta}(t_k)\}^{N_{M^{(r)}}}_{k=1}$ by renewal theory in Section 3.1.2,
    and then calculate the estimator of $RL^{(r)}$ by Eq.(4.35);
    construct an approximate $100(1-\alpha)\%$ confidence region of $RL^{(r)}$ by
    Eq.(3.42);
    if this interval contains 1, then $M^{(r)} = cM^{(r-1)}$ where $c > 1$, otherwise
    $M^{(r)} = M^{(r-1)}$.
until satisfying Eq.(3.26) for three adjacent iterations.
```

Chapter 5

Simulation and Empirical Study

Based on the methodology developed in Chapter 4, the simulation and empirical study are presented in this chapter. Section 5.1 presents the results of the simulation, and Section 5.2 shows the performance of the proposed methods in modeling RMB/dollar exchange rates.

5.1 Simulation Study

In this section, the proposed methods in Chapter 4 are applied to the simulated data sets. Section 5.1.1 presents and compares the performances of three proposed methods for parameter estimation. Comparison of the performances of the MCEM algorithm with increasing sample size by the proposed automated rules and predetermined sample size is provided in Section 5.1.2, and robust analysis is presented in Section 5.1.3 .

5.1.1 Estimation Results

In this section, the simulated data sets from the SV models with $\rho = 0.5$, $\tau = 4$, and $\beta^2 = 1$, and $\rho = -0.2$, $\tau = 1.8$, and $\beta^2 = 0.15$ are considered. For simplicity, these two models are defined as Model I and Model II respectively. Assume that the initial value $x_0 = 0$, and 30 data sets with 50 observations from each model are simulated to show the performance of the proposed methods. For simplicity, define the MCEM algorithm with the PIMH algorithm for the MLE as Method I, and the MCEM algorithm with the resample-move algorithm and the MCEM algorithm with the PMMH algorithm for Bayesian estimation as Method II and Method III respectively. Initial values of Method I are calculated by the formulas in Section 4.1.3, while $\lambda^{(0)} = 16$ and $\delta^{(0)} = 4$ are set as the initial values of Method II and Method III. A small particle size of the SMC algorithm in the PMCMC method is sufficient to obtain an estimator of the marginal distribution $p_\theta(y_{0:t})$; therefore, for Method I and Method III, the particle sizes of the PIMH algorithm and the PMMH algorithm N are set as 25, which is the same particle sizes set in Douc et al. (2014). In addition, set $\alpha_0 = 4$ and $\mu_0 = 3$ under the Bayesian framework. Table 5.1 displays the estimation results by using these three methods, which summarizes the mean and the corresponding standard deviation of estimators for 30 data sets.

Table 5.1 Summary of estimation results for data sets with 50 observations by three different methods.

	Model I			Model II		
	Method I	Method II	Method III	Method I	Method II	Method III
$\hat{\rho}$	0.421	0.347	0.322	-0.299	-0.156	-0.041
(s.d.)	(0.220)	(0.188)	(0.054)	(0.303)	(0.194)	(0.073)
$\hat{\tau}$	3.540	3.843	4.464	1.750	1.968	1.298
(s.d.)	(1.451)	(1.614)	(1.630)	(0.986)	(1.040)	(0.688)
$\hat{\beta}^2$	1.128	1.182	1.831	0.142	0.155	0.185
(s.d.)	(0.689)	(0.729)	(0.308)	(0.048)	(0.054)	(0.033)

From the above table, the estimators for τ and β^2 are relatively close to their true values, particularly for the estimators obtained by Method I and Method II. However, the performances of the estimators for ρ are poor, which are primarily because of the small number of observations. Furthermore, the distances between estimators of ρ and 0 are smaller than those between true values and 0 in the Bayesian analysis. The primary reason is that the prior of ρ

is uniformly on $(-1, 1)$, and pulls the posterior mean back to 0, which becomes influential when the number of observations is small. Therefore, increasing the number of observations may solve this problem.

To show this, the number of observations is increased up to 360, and the corresponding results are displayed in Table 5.2. Compared with the results in Table 5.1, all distances between the estimators and the true values are reduced to an acceptable range. Furthermore, the standard deviations of the estimators have been dramatically decreased.

Table 5.2 Estimation results for data sets with 360 observations by maximum likelihood estimation.

	$\hat{\rho}$ (s.d.)	$\hat{\tau}$ (s.d.)	$\hat{\beta}^2$ (s.d.)
Model I	0.477 (0.096)	3.891 (0.738)	1.199 (0.308)
Model II	-0.229 (0.143)	1.708 (0.334)	0.148 (0.015)

To achieve convergence, the average computing time for data sets with 360 observations is more than 1.5 times the one for data sets with 50 observations. Furthermore, because the estimators of τ and β^2 obtained by Method II and Method III in Table 5.1 are acceptable, 50 observations are sufficient to obtain plausible estimators for λ and δ. For the purpose of reducing time, the last 50 observations of the data set are used to estimate λ and δ, and these two values are then set as the hyper-parameters of the priors. Finally, the entire data set is handled by the ordinary Bayesian method, which provides a Bayesian estimator by means of the posterior distribution that derives from the priors with the hyper-parameters and the likelihoods of all observations. These methods are defined as modified Method II and modified Method III.

First, the phenomenon of underestimating ρ is explained by the ordinary Bayesian method and the suitable number of observations is selected in Bayesian analysis. For simplicity, Model I is used to illustrate these tasks. Because the true values are $\tau = 4$ and $\beta = 1$, to focus on the performance of ρ, $\lambda = 16$ and $\delta = 3$ are set, which make the corresponding inverse gamma distributions with means equaling the true values. The number of observations is increased from 50 to 500, and the estimation results for ρ are presented in Table 5.3. From the table, it is found that the estimators of ρ become closer to the true value 0.5, and 360 observations are sufficient to obtain a reasonable estimator.

Table 5.3 Estimation results for ρ with different numbers of observations (T) by Bayesian analysis with the Resample-move algorithm.

T	50	60	90	150	360	500
$\hat{\rho}$	0.381	0.413	0.441	0.477	0.496	0.517
(s.d.)	(0.201)	(0.249)	(0.139)	(0.119)	(0.069)	(0.054)

Now 30 data sets with 360 observations for Model I and Model II are simulated. As discussed, the observations 311-360 in each data set are used to determine the corresponding hyper-parameters for priors of τ and β^2. Based on these priors, the entire data set is analyzed using the ordinary Bayesian method. The results are presented in Table 5.4. It is found that the performances of the estimators are greatly improved by Method II, which verify the validation of the MCEM algorithm with the resample-move algorithm in this framework. Unfortunately, the estimators by the MCEM algorithm with the PMMH algorithm experience poor performance. This may be caused by bad selection of the proposal distribution or by the small particle size of the SMC algorithm in the PMMH algorithm. This problem will be considered in future research.

The performances of two proposed methods, Method I and Method II, are also compared. Table 5.1 shows that the estimators of ρ obtained by Method I are more accurate, although the performances of estimators for τ and β^2

Table 5.4 Summary of estimation results for data sets with 360 observations by two proposed Bayesian methods.

	Model I		Model II	
	modi. Method II	modi. Method III	modi. Method II	modi. Method III
$\hat{\rho}$ (s.d.)	0.481 (0.084)	0.317 (0.054)	-0.192 (0.169)	-0.038 (0.076)
$\hat{\tau}$ (s.d.)	4.129 (0.513)	4.517 (1.430)	1.660 (0.389)	1.421 (0.769)
$\hat{\beta}^2$ (s.d.)	1.135 (0.294)	1.547 (0.985)	0.157 (0.018)	0.185 (0.033)

obtained by these two methods are similar. Based on Table 5.2 and Table 5.4, the performances of estimators for τ and β^2 by these two methods are also similar, and Method II provides a better performance of $\hat{\rho}$. Furthermore, the average computation time by Method I is more than 1.5 times that of the modified Method II, and the minimum computation time by Method I is 10 times more than that of the modified Method II. Because the entire data set should be involved in the MCEM algorithm when performing the maximum likelihood estimator, and 50 observations are sufficient to determine hyper-parameters in modified Method II, the difference in computation time between these two methods becomes larger as the number of observations increases. Therefore, the computation time is another disadvantage for Method I when the number of observations is large. Overall, the MLE is suitable when the number of observation is small, and Bayesian estimators by modified Method II is recommended when the number of observations increases to a large value.

In addition, these two proposed methods are also compared with the existing methods. To obtain the MLE for the SV model, the MCEM algorithm with the FFBSi algorithm is discussed in Kim (2005). However, the convergence rate of the MCEM algorithm with the FFBSi algorithm is quite slower than the proposed method for MLE, because the PIMH algorithm is more efficient than the FFBSi algorithm to generate the sample from the joint smoothing distributions. And Bayesian estimators for the SV model are typically obtained by the Bayesian method with predetermined hyper-parameters, where both the resample-move algorithm and the PMCMC method can be used to handle the corresponding posterior distributions. However, the proposed Bayesian methods provide more robust estimators, which is verified by the comparison of the typical Bayesian methods and the proposed Bayesian methods presented in Section 5.1.3.

5.1.2 Sample Size Issue

An automated rule to determine whether the sample size should be increased at each iteration in the implementation of the particle MCEM algorithm is proposed in this thesis. This section demonstrates the performance of the proposed rules compared with the predetermined values in McCulloch (1997), in which the sample sizes for the first 19 iterations are set as 50, the sample sizes for next 20 iterations are set as 200, and the remaining sizes are set as 5000.

Figure 5.1 presents the numbers of sample sizes at each iteration according to the proposed rule and the predetermined rule when performing MLE by the MCEM algorithm. During the first 19 iteration, the sample sizes for the proposed method increase only slightly and are close to the ones for the predetermined case. The sample sizes start to increase at the 20th iteration, and increase sharply during the last 20 iterations. It is obvious that the proposed rule shows a better performance. The iterations in the proposed method are less than half of the ones in the predetermined method, and only about 10 iterations in the proposed method have larger sample sizes than those in the predetermined method. In addition, the computation time of predetermined method is almost three times that of the proposed method. When implementing the MCEM algorithm with the PMMH algorithm in the framework of Bayesian analysis, the increased sample sizes are similar, and the result is presented in Figure 5.2.

Figure 5.1 Sample size comparison of maximum likelihood estimation by the M-CEM algorith with the PIMH method: the proposed rule(+) and the predetermined values (∗).

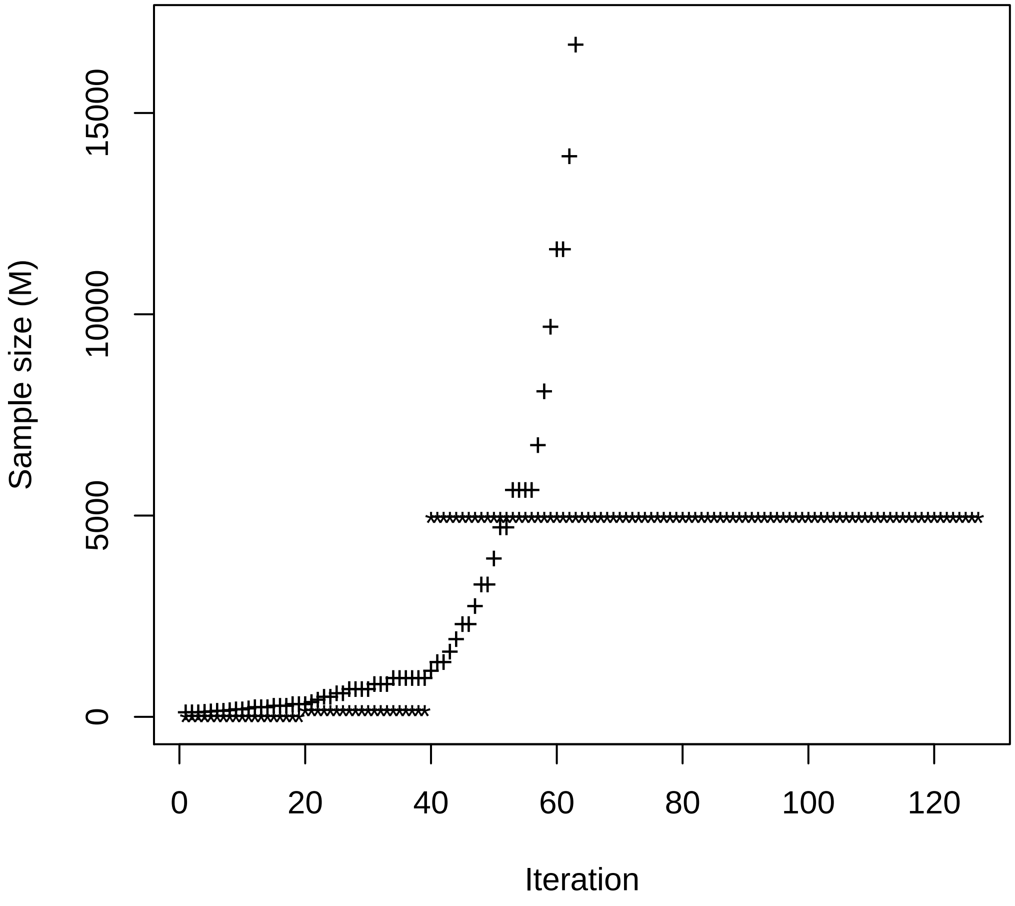
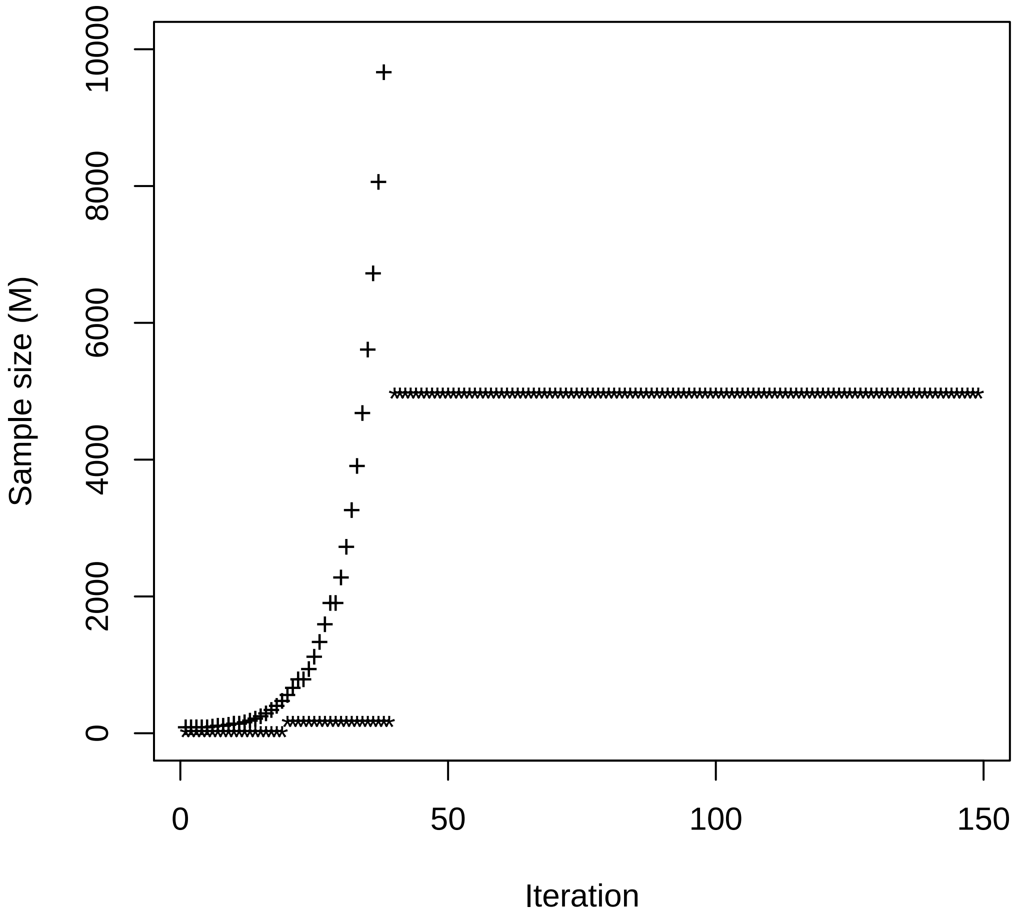


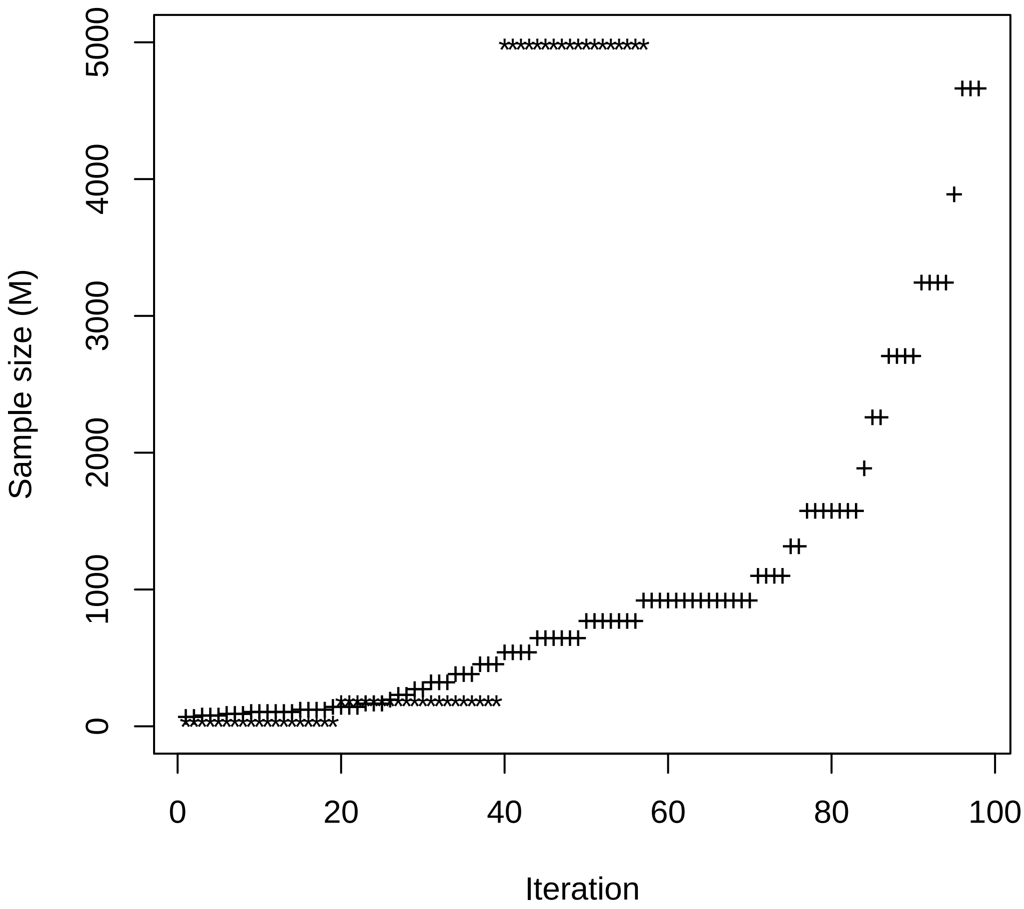
Figure 5.3 presents the numbers of sample sizes at each iteration by the proposed rule and the predetermined rule when performing Bayesian estimation by the MCEM algorithm with the resample-move algorithm. As in the

Figure 5.2 Increase in sample size by the MCEM algorithm with method III: the proposed rule (+) and the predetermined values (∗).



previous case, for the proposed method, the sample sizes in the first several iterations grow slowly, and increase faster as the number of iterations increases. Although the number of iterations in the proposed method is larger than the one in predetermined method, the sample sizes in the proposed method increase up to 4644, which is less than 5000, and thus the computation time in the proposed method is also less in this case.

Figure 5.3 Increase in sample size by the MCEM algorithm with the resample-move algorithm: the proposed rule (+) and the predetermined values(∗).



5.1.3 Robust Analysis

Type II maximum likelihood prior is introduced to obtain a robust estimator in this thesis. In typical Bayesian analysis, Bayesian estimators are highly influenced by hyper-parameters, and the changes of hyper-parameters typically cause the changes of estimators. The initial values of hyper-parameters in the proposed proposed algorithm are subjective. However, the changes in initial values rarely affect the estimators. To verify this, a robust analysis for the proposed methods within the Bayesian framework are conducted in the following section.

A data set with 50 observations of the SV model with $\rho = 0.5$, $\beta = 1$ and $\tau = 4$ is simulated, and assume that the initial value $x_0 = 0$. Eight values $\{5, 8, 10, 15, 16, 20, 24, 30\}$ are selected as the candidate set Λ_0 for λ, and six values $\{1, 3, 4, 6, 8, 10\}$ as the candidate set Δ_0 for δ. Forty-eight groups are defined by the following rule: select the first element in Δ_0, and then select all the elements in Λ_0 in order, and denote these eight groups by index $1 - 8$ in order; next, select the second element in Δ_0, then select all the elements in Λ_0 in order, and denote these eight groups by index $9 - 16$ in order; and so on. All 48 groups are considered to be the initial values in the proposed Bayesian methods and the predetermined hyper-parameters in the typical Bayesian analysis respectively. The selected estimation results are presented in Figure 5.4 and Figure 5.5.

Figure 5.4 shows the estimation results for β^2 by the MCEM algorithm with the resample-move algorithm and the Bayesian method with predetermined hyper-parameters. It is easy to see that the estimators $\hat{\beta}^2$ by the Bayesian method with predetermined hyper-parameters have a larger fluctuation than the ones by our method. In particular, the estimators $\hat{\beta}^2$ by the proposed method fluctuate little around a horizontal line, and the ones by the Bayesian method with predetermined hyper-parameters jump up once the

Figure 5.4 Estimation Results of β^2 for two methods: the MCEM algorithm with the resample-move algorithm (line) and the Bayesian analysis with predetermined hyper-parameters (○).

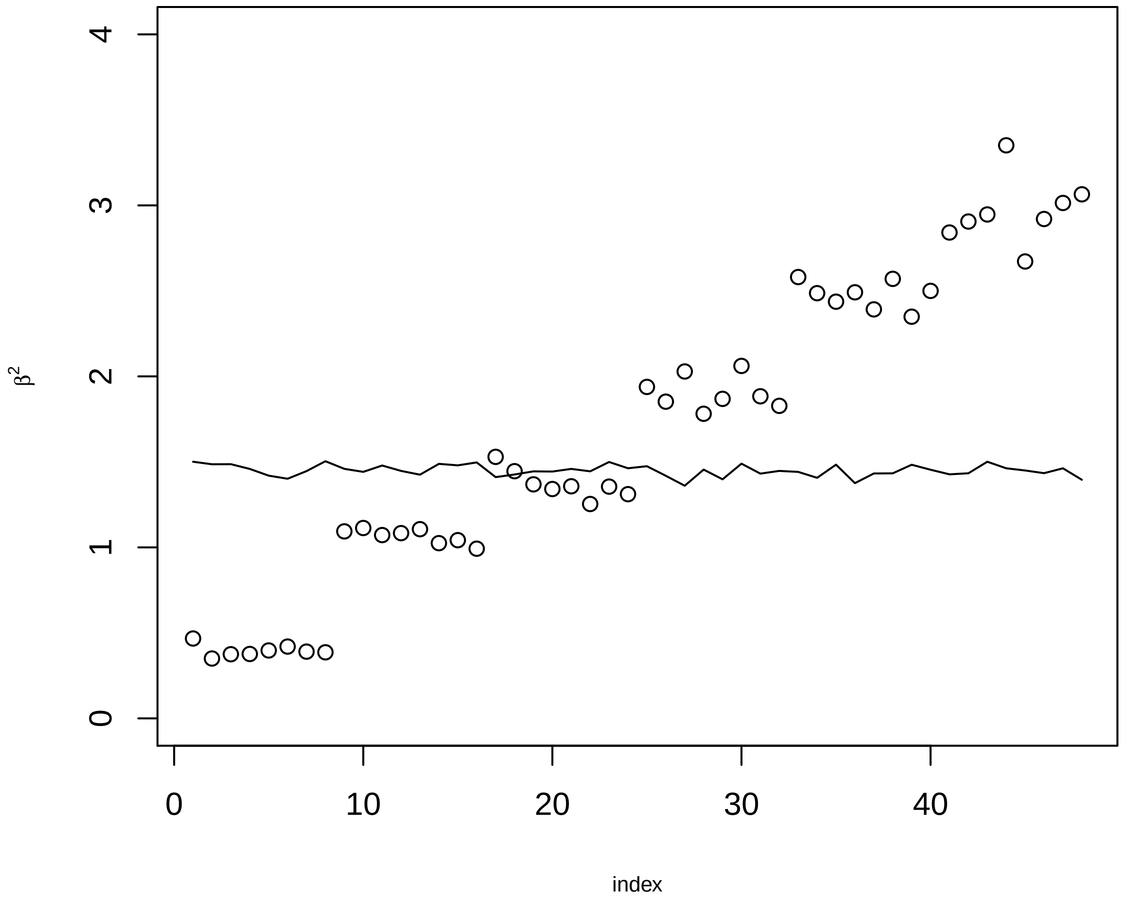
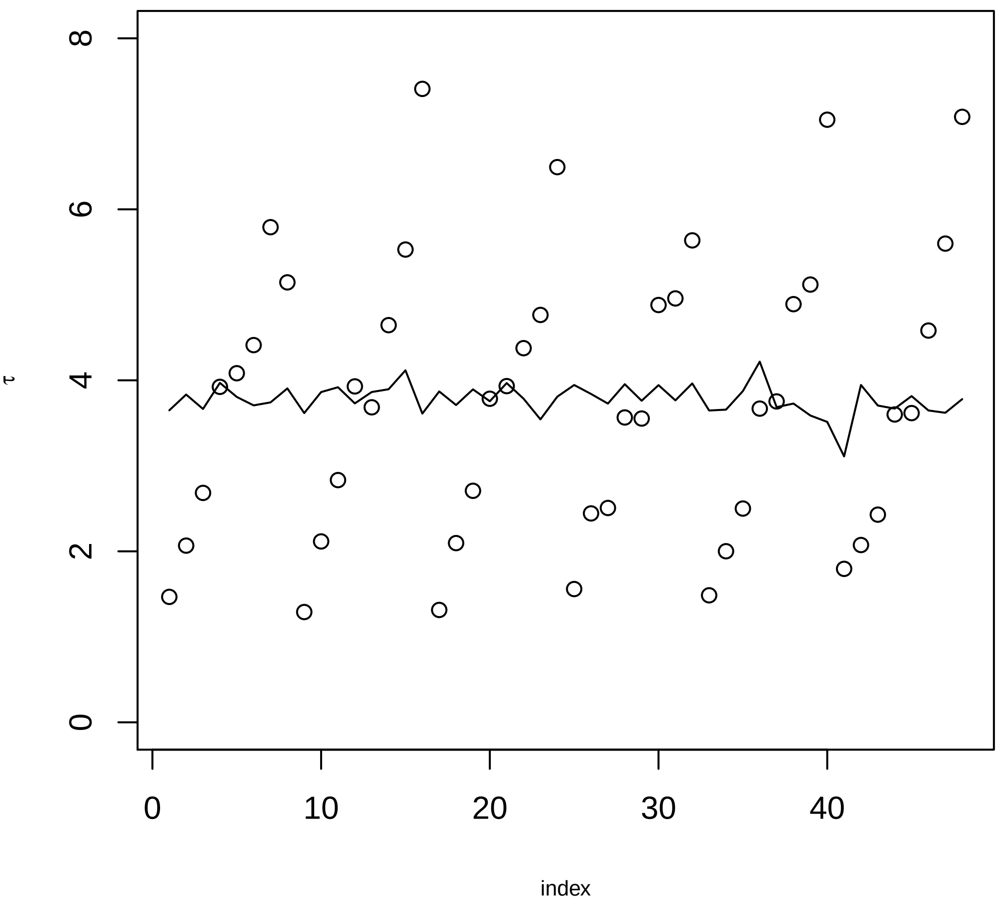


Figure 5.5 Estimation Results of τ for two methods: the MCEM algorithm with the PMMH algorithm (line) and the Bayesian analysis with predetermined hyperparameters (○).



hyper-parameter δ increases. Figure 5.5 shows the estimation results for τ by the MCEM algorithm with the PMMH algorithm and the Bayesian method with predetermined hyper-parameters. The estimators $\hat{\tau}$ by the proposed method fluctuate little around the horizontal line $y = 4$, and the estimators by the ordinary Bayesian method change greatly as the hyper-parameter λ moves. From these two figures, it can be seen that robustness is one of the advantages of the proposed Bayesian methods compared with the Bayesian method with predetermined hyper-parameters.

5.2 Empirical Study

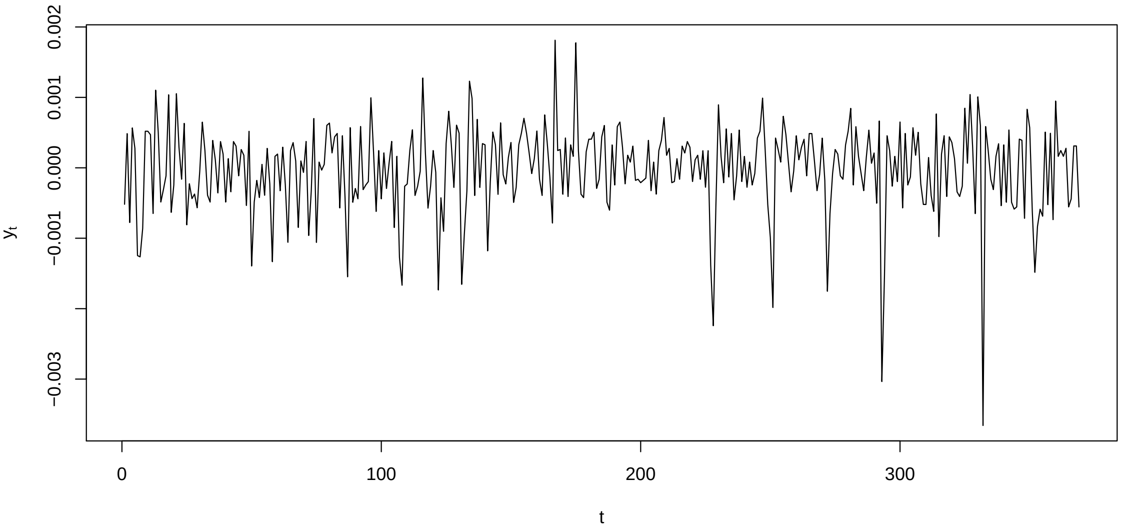
At first, an application of the methods to RMB/dollar daily exchange rates from July 1st 2013 to December 31th 2014 obtained from the State Administration of Foreign Exchange is conducted. Figure 5.6 shows the time plot of the log-return of the exchange rates, defined as

$$y_t = \log \frac{r_t}{r_{t-1}}, \tag{5.1}$$

where r_t is the daily exchange rate. From this return, we can find that the data shows various level of volatility, and more obviously, the last 100 observations fluctuate wildly, which is possibly caused by the quantitative easing monetary policy adopted the Federal Reserve in the second half of 2014. It can also be seen that this data set shows heteroscedasticity, and thus the SV model can be used to describe these log-returns.

Three proposed methods are used to perform parameter estimation, and the initial setting of each method is the same as in Section 5.1.1. The estimation results by these three methods are presented in Table 5.5. The standard deviations of maximum likelihood estimators are estimated as in Kim (2005) to ensure the positive definite property of the estimated observed information matrix. The standard deviations of Bayesian estimators are calculated using

Figure 5.6 The time plot of log-return of RMB/dollar exchange rates from July 1st 2013 to December 31th 2014.



the sample standard deviations of the samples for the corresponding parameters. Similar to the simulation study in Section 5.1.1, the estimation results by Method I and Method are close, and $\hat{\rho}$ in Method III is substantially smaller than Method I and II. According to the simulation study, the estimators in Method I and Method II are plausible. In addition, the estimators of β^2 are of a small order of magnitude, because the relatively few changes in the RMB/dollar exchange rates cause the values in log-returns to be small.

Table 5.5 Estimation results for the log-return of RMB/dollar exchange rates from July 1st 2013 to December 31th 2014.

	$\hat{\rho}$ (s.d.)	$\hat{\tau}$ (s.d.)	$\hat{\beta}^2$ (s.d.)
Method I	0.593 (0.093)	0.318 (0.050)	2.83×10^{-7} (2.87×10^{-8})
Method II	0.541 (0.148)	0.366 (0.103)	2.90×10^{-7} (3.07×10^{-8})
Method III	0.217 (0.483)	0.397 (0.192)	3.74×10^{-7} (1.55×10^{-7})

To further investigate the performance of Method I and Method II, the RMB/dollar daily exchange rates from January 1st 2008 to June 30th 2009 are also studied. Figure 5.7 shows the time plot of the log-return of the corresponding exchange rates. A large level volatility is presented at the first half of data, which was primarily caused by the financial crisis happened in 2008. Due to the heteroscedasticity property of this data set, the SV model is used to handle this data. The estimation results are listed in Table 5.5.

Figure 5.7 The time plot of log-return of RMB/dollar exchange rates from January 1st 2008 to June 30th 2009.

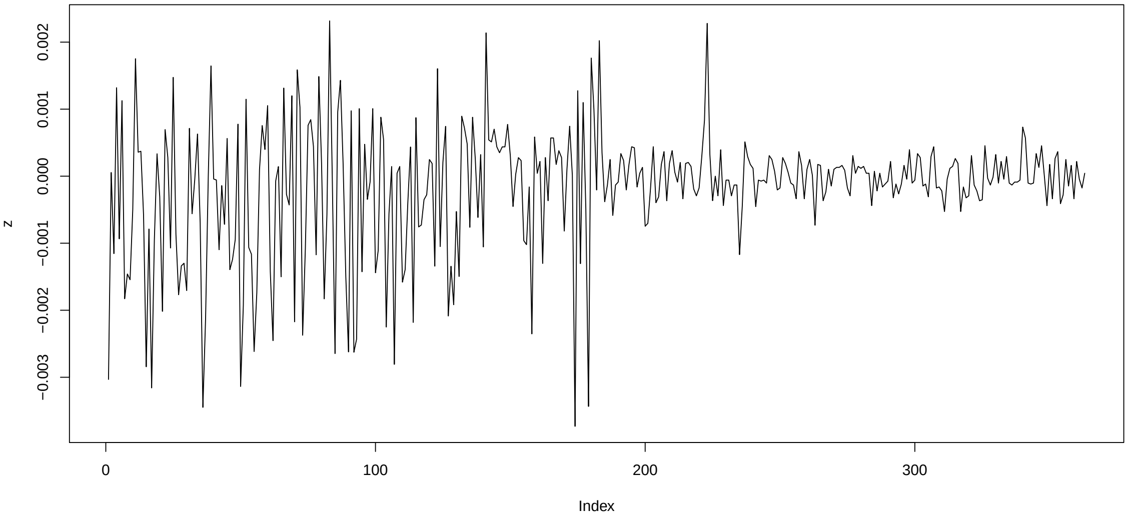


Table 5.6 Estimation results for the log-return of RMB/dollar exchange rates from January 1st 2008 to June 30th 2009.

	$\hat{\rho}$ (s.d.)	$\hat{\tau}$ (s.d.)	$\hat{\beta}^2$ (s.d.)
Method I	0.989 (0.010)	0.093 (0.024)	9.14×10^{-7} (5.76×10^{-7})
Method II	0.992 (0.007)	0.054 (0.024)	9.76×10^{-7} (2.95×10^{-7})

The estimators of ρ and β^2 by these two methods are close, and there are substantial difference between the estimators of τ. However, $\hat{\tau}$ by Method II is covered by the confidence interval of τ by Method I, and the confidence interval of τ by Method II also includes $\hat{\tau}$ by Method I. Therefore, the difference between these two estimators are probably caused by the Monte Carlo errors. Compared to the estimation results for the data from July 1st 2013 to December 31th 2014, the model for this data has higher $\hat{\rho}$ but smaller $\hat{\tau}$, which shows that the unobserved volatilities in this model are more highly related to each other with a smaller fluctuation. Furthermore, the higher level volatility in this data set makes the estimators for β^2 larger.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This thesis focuses on the inference for NLSS models by using particle methods. In the classical framework, the EM algorithm and the PIMH algorithm are combined to obtain the maximum likelihood estimators. In Bayesian analysis, type II maximum likelihood prior is introduced to obtain robust Bayesian estimators. To obtain the estimators, the EM algorithm and the particle methods are used together. More specifically, the MCEM algorithm with the resample-move algorithm and the MCEM algorithm with the PMMH algorithm are used to analyze this Bayesian problem respectively.

Another contribution of this thesis is to propose an automated rule to decide whether the Monte Carlo sample size should be increased at each iteration in the implementation of the MCEM algorithm with the PMCMC algorithm. Renewal theory is used to construct a confidence interval of relative likelihood at each iteration. Based on this confidence interval, an automated rule is designed for increasing the sample size. Furthermore, it is shown that the rule for sample size proposed in Booth and Robert (1999) is effective in the implementation of the MCEM alorithm with the resample-move algorithm based on the central limit theory of particle filters, which was validated in Chopin (2004).

For the purpose of illustration, the proposed methods are applied to stochastic volatility model, which is presented in Section 4. In the simulation study, the validation of MLE by the MCEM algorithm with the PIMH algorithm and Bayesian estimators by the MCEM algorithm with the resample-move algorithm is established. However, Bayesian estimators by the MCEM algorithm with the PMMH algorithm fail to converge to the true values. Comparisons of the performances of these methods are also discussed. When the number of observations is small, the MLE performs relatively well. However, Bayesian estimators by the MCEM algorithm with the resample-move algorithm are the most effective in terms of estimation accuracy and computation time, and when the number of observations is large (e.g., 360 or more). The empirical study also provides evidence of these two viewpoints. The robustness of the proposed Bayesian methods compared with the ordinary Bayesian method is also presented in the simulation study.

6.2 Future Work

The first task in the future is to find out the primary reason for poor performance of the MCEM algorithm with the PMMH algorithm in the Bayesian framework, and to propose a corresponding solution. More proposal distributions in the PMMH algorithm can be tried to observe and summarize the changes, and find out the suitable classes of proposal distributions. Increasing the particle size N to strike a balance between the computational time and the acceptance rate in the PMMH algorithm is another possible method.

When doing the maximum likelihood estimator in the SV Model, the scheme introduced in Kim (2005) is used to determine the initial values of the MCEM algorithm to save computation time. However, there is no discussion of the initial value in the implementation of the MCEM algorithm within the Bayesian framework. A more effective method is required to develop for a more efficient convergence in this situation.

Finally, to further investigate the performance of the proposed methods, they can be applied to other special cases of NLSS models. The SV models with particular modifications, such as the SV model with fat tails and the SV model with jumps are typically used in empirical studies. The proposed methods provide new approaches to handle these models.

Bibliography

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72:269–342, 2010.

[2] C. Andrieu, N. D. Freitas, and A. Doucet. Sequential MCMC for Bayesian model selection. In *Higher-Order Statistics, 1999. Proceedings of the IEEE Signal Processing Workshop on*. IEEE, 1999.

[3] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725, 2009.

[4] M. J. Bayarri and J. O. Berger. The interplay of Bayesian and frequentist analysis. *Statistical Science*, 19(1):58–80, 2004.

[5] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer series in statistics. Springer-Verlag, New York, second edition, 1985.

[6] J. G. Booth and J. P. Hobert. Maximizing generalized linear mixed model likelihood with an automated Monte Carlo EM algorithm. *Journal of Royal Statistical Society: Series B*, 61:265–285, 1999.

[7] O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer Science+Business Media, Inc., New York, 2005.

[8] J. Carpenter, P. Clifford, and P. Fearnhead. An improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proceedings*, 146:2–7, 1999.

[9] K. S. Chan and J. Ledolter. Monte Carlo EM estimation for time series models involving counts. *Journal of the American Statistical Association*, 90(429):242–252, 1995.

[10] N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–551, 2002.

[11] N. Chopin. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411, 2004.

[12] J. Cornebise, E. Moulines, and J. Olsson. Adaptive methods for sequential importance sampling with application to state space models. *Statistics and Computing*, 18:461–480, 2008.

[13] D. Creal. A survey of sequential Monte Carlo methods for economics and finance. *Econometric Reviews*, 31(3):245–296, 2012.

[14] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society: Series B*, 39:1–38, 1977.

[15] R. Douc, E. Moulines, and J. Olsson. Optimality of the auxiliary particle filter. *Probability and Mathematical Statistics*, 29:1–28, 2009.

[16] R. Douc, É. Moulines, and D. S. Stoffer. *Nonlinear Time Series: Theory, Methods and Applications with R Examples.* Texts in Statistical Science. CRC Press, Boca Raton, first edition, 2014.

[17] A. Doucet, N. D. Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer, New York, 2001.

[18] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.

[19] A. Doucet and A. M. Johansen. A tutorial on particle filtering and s-moothing: Fifteen years later, 2011.

[20] G. Fort and E. Moulines. Convergence of the Monte Carlo expectation maximization for curved exponential families. *The Annals of Statistics*, 31(4):1220–1259, 2003.

[21] W. R. Gilks and C. Berzuini. Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Journal of Royal Statistical Society: Series B*, 63:127–146, 2001.

[22] S. Godsill, A. Doucet, and M. West. Maximum a posteriori sequence estimation using Monte Carlo particle filters. *Annals of the Institute of Statistical Mathematics*, 53:82–96, 2001.

[23] S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, 2004.

[24] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140:107–113, 1993.

[25] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Methuen's monographs on applied probability and statistics. Methuen, London, first edition, 1964.

[26] J. E. Handschin and D. Q. Mayne. Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. *International Journal of Control*, 9(5):547–559, 1969.

[27] M. Hüzeler and H. R. Künsch. Monte Carlo approximations for general state-space models. *Journal of Computational and Graphical Statistics*, 7(2):175–193, 1998.

[28] J. Kim. *Parameter Estimation in Stochastic Volatility Models with Missing Data Using Particle Methods and the EM Algorithm*. PhD thesis, University of Pittsburgh, 2005.

[29] J. Kim and D. S. Stoffer. Fitting stochastic volatility models in the presence of irregular sampling via particle methods and the EM algorithm. *Journal of Time Series Analysis*, 29:811–833, 2008.

[30] G. Kitagawa. Monte-Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[31] R. A. Levine and G. Casella. Implementations of the Monte Carlo EM algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

[32] J. S. Liu and R. Chen. Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995.

[33] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamical systems. *Journal of the American Statistical Association*, 93(443):1032–1044, 1998.

[34] H. F. Lopes and N. G. Polson. Bayesian inference for stochastic volatility modeling. In *Rethinking Risk Measurement and Reporting: Volume II*. Risk Books, 2010.

[35] C. E. McCulloch. Maximum likelihood variance components estimation for binary data. *Journal of the American Statistical Association*, 89(425):330–335, 1994.

[36] C. E. McCulloch. Maximum likelihood algorithm for generalized linear mixed models. *Journal of the American Statistical Association*, 92(437):162–170, 1997.

[37] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley series in probability and statistics. Wiley-Interscience, Hoboken, New Jersey, 2008.

[38] C. N. Morris. Parametric empirical Bayes inference: Theory and applications. *Journal of the American Statistical Association*, 78(381):47–55, 1983.

[39] M. L. A. Netto, L. Gimeno, and M. J. Mendes. On the optimal and suboptimal nonlinearing filtering problem for discrete-time system. *IEEE Transactions on Automatic Control*, 23(6):1062–1067, 1978.

[40] E. Nummelin. MC's for MCMC'ists. *International Statistical Review*, 70(2):215–240, 2002.

[41] J. Olsson, O. Cappé, R. Douc, and É. Moulines. Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state space models. *Bernoulli*, 14(1):155–179, 2008.

[42] J. Olsson and T. Rydén. Asymptotic properties of particle filter-based maximum likelihood estimators for state space models. *Stochastic Processes and their Applications*, 118:649–680, 2008.

[43] M. K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

[44] M. K. Pitt and R. S. Silva. On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171:134–151, 2012.

[45] H. Robbins. An empirical Bayes approach to Statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. University of California Press, California, 1956.

[46] C. P. Robert, T. Rydén, and D. M. Titterington. Convergence controls for MCMC algorithms, with applications to hidden Markov chains. *Journal of Statistical Computation and Simulation*, 64:327–355, 1999.

[47] D. B. Rubin. A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fraction of missing information is modest: The SIR algorithm. *Journal of the American Statistical Association*, 82:543–546, 1987.

[48] S. L. Scott. Bayesian methods for hidden Markov models. *Journal of the American Statistical Association*, 97(457):337–351, 2002.

[49] S. R. Searle, G. Casella, and C. E. McCulloch. *Variance components*. Wiley series in probability and mathematical statistics. Applied probability and statistics. Wiley, New York, 1992.

[50] J. R. Stroud, N. G. Polson, and P. Müller. Practical filtering for stochastic volatility models. In *State Space and Unobserved Component Models: Theory and Applications*. Cambridge University Press, Cambridge, 2004.

[51] G. C. G. Wei and M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithm. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

[52] N. Whiteley and A. M. Johansen. Auxiliary particle filtering: Recent developments. In *Bayesian Time Series Models*. Cambridge University Press, Cambridge, 2011.

[53] C. F. J. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11:95–103, 1983.

Appendix A

Proof of Convergence of the Proposed MCEM Algorithm in the SV Model

In this section, the same method as Kim and Stoffer (2008) is used to prove the convergence of the proposed MCEM algorithm in Section 4.1 and Section 4.2.2. The assumptions of convergence theorem proposed in Chan and Ledolter (1995) are verified to establish the convergence property of the proposed method. Below the theorem proposed in Chan and Ledolter (1995) is restated and the restatement of assumptions introduced in Kim and Stoffer (2008) is quoted. To simplify the notation, let X be the complete data and y be the observations.

Theorem A.1 Suppose that Assumptions 1-4 below are satisfied. Let θ^* be an isolated local maximizer of $l_y(\theta)$, the logarithm of the likelihood. Then there exists a neighborhood denoted by V_1 of θ^* such that for start values of the MCEM algorithm proposed in Chapter 4 inside V_1 and for all $\epsilon_1 > 0$, there exists a k_0 such that $P(\mid \theta_M^{(k)} - \theta^* \mid < \epsilon_1$ for some $k \leq k_0) \rightarrow 1$ as $M \rightarrow \infty$.

Assumptions

1. For all θ', $l_X(\theta') = q(Z, \theta')$, where Z is a measurable vector function of X, q is linear in Z and $l_X(\theta')$ is the complete likelihood at the parameter θ'.

2. $q(Z, \cdot)$ attains the unique global maximum at $\mathcal{M}(Z)$ and $\mathcal{M}$ is continuous in Z.

3. The convergence, in probability, of $\bar{Z}_M \to E_\theta(Z)$ as $M \to \infty$ is uniform over compact subsets of Ω where Ω is a parameter space, $\bar{Z}_M$ is the sample mean of Z values computed from the sample $X_1(\theta), \ldots, X_M(\theta)$ generated by particle methods.

4. $l_y(\theta)$ is continuous in θ.

A.1 Proof of Convergence of the Proposed MCEM Algorithm for MLE of the SV Model

In the following, Assumptions 1-4 within the framework of Section 4.1 are verified to prove the convergence of Algorithm 11.

In this case, the complete data is $X = (x_{0:T}, y_{0:T})$ and the parameters are $\theta = (\rho, \tau, \beta^2)$, then the log-likelihood of this complete data is

$$\begin{aligned} l_X(\theta) &= -\frac{1}{2}\left[Z_1 \cdot \frac{1}{\tau} - 2Z_2 \cdot \frac{\rho}{\tau} + Z_3\frac{\rho^2}{\tau} - Z_4 - Z_5\frac{1}{\beta^2} + C(\theta)\right] \\ &= q(Z, \theta), \end{aligned} \tag{A.1}$$

where

$$Z = (Z_1, \ldots, Z_5) = (\sum_{t=1}^{T} x_t^2, \sum_{t=1}^{T} x_{t-1}x_t, \sum_{t=1}^{T} x_{t-1}, \sum_{t=1}^{T} x_t, \sum_{t=1}^{T} \frac{y_t}{\exp(x_t)}) \tag{A.2}$$

and $C(\theta) = T\,[\log(4\pi^2) + \log(\tau) + \log(\beta^2)]$. It is easy to find that Z is a measurable vector function of X and q is linear in Z. Hence, Assumption 1 holds.

The log-likelihood function $q(Z, \cdot)$ attains the unique global maximum at $\mathcal{M}(Z)$, where

$$\begin{aligned} \mathcal{M}(Z) &= \left(\arg\max_{\rho} q, \arg\max_{\tau} q, \arg\max_{\beta^2} q\right) \\ &= \left(\frac{Z_2}{Z_3}, \frac{Z_1 - 2\rho Z_2 + \rho^2 Z_3}{T}, \frac{Z_5}{T}\right). \end{aligned} \tag{A.3}$$

Furthermore, $\mathcal{M}(Z)$ is continuous in Z. Therefore, Assumption 2 holds.

Andrieu et al. (2010) shows that the PIMH algorithm is a standard IMH algorithm. Furthermore, by Theorem 1 of Nummelin (2002), for any ϕ-integrable function f

$$\int f(x_{0:T}) \mathrm{d}\phi^{M}_{0:T|T}(x_{0:T};\theta) \xrightarrow{P} f(x_{0:T}) \mathrm{d}\phi_{0:T|T}(x_{0:T};\theta) \tag{A.4}$$

As same as the proof in the appendix of Kim and Stoffer (2008), it can be shown that Assumption 3 is satisfied.

Since

$$l_y(\theta) = \int \cdots \int l_X(\theta) \mathrm{d}x_0 \cdots \mathrm{d}x_T, \tag{A.5}$$

$l_y(\theta)$ is continuous in θ. Therefore, Assumption 4 is satisfied, and the convergence of Algorithm 11 holds.

A.2 Proof of Convergence of the Proposed M-CEM Algorithm for Bayesian Analysis in the SV Model

In this appendix, Assumptions 1-4 based on the Bayesian framework in Section 4.2.2 are verified to prove the convergence of Algorithm 14.

In this case, the complete data is $X = (x_{0:T}, y_{0:T}, \rho, \tau, \beta^2)$ and the parameters are $\theta = (\gamma, \delta)$, then the log-likelihood of this complete data is

$$\begin{aligned} l_X(\theta) =& -\frac{1}{2}\left[Z_1 - 2Z_2 + Z_3 - Z_4 - Z_5\right] - (\alpha_0 + 1 - \frac{T}{2})Z_6 - \gamma Z_8 \\ & - (\mu_0 + 1 - \frac{T}{2})Z_8 - \delta Z_9 \\ =& q(Z, \theta), \end{aligned} \tag{A.6}$$

where

$$\begin{aligned} Z =& (Z_1, \ldots, Z_9) \\ =& \Big(\frac{\sum_{t=1}^{T} x_t^2}{\tau}, \frac{\rho \sum_{t=1}^{T} x_{t-1}x_t}{\tau}, \frac{\rho^2 \sum_{t=1}^{T} x_{t-1}}{\tau}, \sum_{t=1}^{T} x_t, \sum_{t=1}^{T} \frac{y_t}{\beta^2 \exp(x_t)}, \\ & \log \tau, \frac{1}{\tau}, \log \beta^2, \frac{1}{\beta^2}\Big) \end{aligned}$$

and $C(\theta) = -T \log 2\pi + \alpha_0 \log \gamma - \log \Gamma(\alpha_0) + \mu_0 \log \delta - \log \Gamma(\mu_0)$. It is easy to find that Z is a measurable vector function of X and q is linear in Z. Hence, Assumption 1 holds.

The log-likelihood function $q(Z, \cdot)$ attains the unique global maximum at $\mathcal{M}(Z)$, where

$$\begin{aligned} \mathcal{M}(Z) &= \left(\arg\max_{\gamma} q, \arg\max_{\delta} q\right) \\ &= \left(\frac{\alpha_0}{Z_7}, \frac{\mu_0}{Z_9}\right). \end{aligned} \tag{A.7}$$

Furthermore, $\mathcal{M}(Z)$ is continuous in Z. Therefore, Assumption 2 holds.

Andrieu et al. (2010) shows that the PMMH algorithm is a special case of MH algorithm. The same as the proof in Appendix A.1, Assumption 3 can be verified.

Since

$$l_y(\theta) = \int \cdots \int l_X(\theta) \mathrm{d}x_0 \cdots \mathrm{d}x_T \mathrm{d}\rho \mathrm{d}\tau \mathrm{d}\beta^2, \tag{A.8}$$

$l_y(\theta)$ is continuous in θ. Therefore, Assumption 4 is satisfied, and the convergence of Algorithm 14 holds.