ON PARAMETER ESTIMATION OF STATE SPACE MODELS AND ITS APPLICATIONS

BY

LIANG WANG

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Statistics and Biostatistics

Written under the direction of

Rong Chen

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2018

ABSTRACT OF THE DISSERTATION

On parameter estimation of state space models and its applications

by Liang Wang

Dissertation Director: Rong Chen

State space model is a class of models where the observations are driven by underlying stochastic processes. It is widely used in computer vision, economics and financial data analysis, engineering, environmental sciences and etc. My thesis mainly addresses the parameter estimation problem of state space model and the applications of it.

This thesis starts with a brief introduction and the motivation for studying the problems in the first chapter. The second chapter follows the first one by covering the main tools used to study the topics in the thesis. The general framework of state space models and its related filtering methods, Kalman Filtering for linear Gaussian models and sequential Monte Carlo for other cases, are introduced. The information criteria, as a tool for model selection, are also covered in this chapter.

The parameter estimation problem is mainly discussed in the third chapter. Two algorithms under the general framework of Stochastic Approximation methods are proposed. These two algorithms attain much faster convergence rate and less computational cost by variance reduction

techniques which utilize the property of sequential Monte Carlo methods. Two numerical examples are examined to compare the performance. Another contribution of Chapter 3 is the application of sequantial Monte Carlo methods in modeling and predicting the bond yield curve with regime-switching Dynamic Nelson-Siegel model.

The fourth chapter, which is a joint work with Hao Chang, develops a state space model with regime switching to detect periodically collapsing rational bubbles in stock price. The present-value stock-price model is expressed in a state space form and the bubble process is modeled as a conditional dynamic linear system. The asset-bubble system is estimated by a novel sequential Monte Carlo based method, Mixture Kalman Filter (MKF). The efficacy of the proposed method is examined by simulated observations and real stock index of the US market.

Another application of state space model with regime switching is discussed in the fifth chapter, in which real-time Blood Glucose Monitoring problem is addressed using a conditional dynamic linear system modeling. A study with a biostatistical dataset, Star 1 dataset, has shown the advantage of the proposed novel estimation framework.

In the sixth chapter, a nonparametric regression model, $l1$ trend filtering method is discussed. Two trend filtering models out of state space representation, both of which have similar property as $l1$ trend filtering, are proposed. With the implementation of sequential Monte Carlo methods as well as a greedy Viterbi algorithm, both trend filtering models can operate on-line rather than just on batch data. To better emphasize the two models' improvement in on-line trend filtering, a real world econometrics topic is introduced. The econometric example shows the competence of trend filtering as well as the efficiency of the proposed models.

Acknowledgements

First I would like to express my deepest gratitude to my thesis advisor, Professor Rong Chen. Without his continuous guidance, brilliant ideas and strong support, I could not stand a chance to finish this dissertation. I still remember the smile on his face and the Coke in his hand when I first met with him in his office four years ago. Since then there have been so many precious advice, encouragement, as well as critism, from which I have benefited so much and will never forget. More than an advisor, he's a role model that I would like to follow throughout my life, not only for his curiosity to the unknown, but also positive attitude towards life.

I wish to thank the faculties in the department of statistics and biostatistics for providing an inspring and accessible environment to do statistical research. I am grateful to the former graduate director, Professor John Kolassa, for his valuable advices and continuous support for the graduate study and life. Sincerely thanks to the department chair, Professor Regina Liu, for her efforts to take care of every student.

I also want to say thanks to Professor Han Xiao, Professor Zhiqiang Tan and Professor Yangru Wu for being my committee members and providing helpful comments on my dissertation.

This is also an opportunity to thank my colleague, Hao Chang, in the department of both statistics and finance, for his important contribution to our collaborated work. His insights to potential finance applications of state space models always inspire me.

Last but not least, I want to show my gratitude to my former and current colleagues in the statistics program for the enlightening discussions on and beyond statistical research. They made my life in the department enjoyable.

Finally, great thanks to my family for their unconditional love and my girlfriend for her continuous encouragement and help.

Dedication

To my parents Feng and Lingqin; To my girlfriend Linglin

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Statistical modeling and analysis for sequential data have many applications in both scientific and industrial fields. In many of the applications, the driving forces behind the evolution of the sequential data are not observable or measurable. State space models (Cappé et al., 2009; Liu and Chen, 1998; West and Harrison, 1998) allow the researcher to model an observed time series as being explained by a vector of unobserved state time series in a dynamic system. The long history of state space models can be traced back to the early Kalman Filter literatures (see Kalman, 1960; Kalman and Bucy, 1961) as an engineering problem. Early applications of Kalman Filter include tracking objects, such as airplanes and missiles, from noisy measurements, such as radar. Since its early success in engineering, state space models have gained increasing attentions in time series forecasting (see Harvey, 1990; West and Harrison, 1998; Petris et al., 2009; Prado and West, 2010). Due to their flexibility and easy interpretation in time series modeling, countless applications can be found under the state space representation. Examples of state-observation variables include the bubble and stock price in finance, original and received signal in wireless communication, running speed and position in real-time tracking and many others. Its popularity leads to continuously active discussions on state space modeling among researchers from engineering, statistics, finance and many other disciplines (Kantas et al., 2015; Durbin and Koopman, 2012; Aoki, 2013).

1.1 Motivation

Despite of significant progress that have been made in state space modeling during the past decades, there are still many unsolved issues that need to be explored. Moreover, there are growing

needs to calibrate the state space model to solve a real data application issue from practitioners in different fields. Motivated by this, the thesis tries to make improvements in the following two topics in state space models: model fitting or parameter estimation for a specific state space model, and the applications of state space models.

State space models are first proposed from an engineering background, where the parameters are often previously specified. Therefore early researches focused on model inference based on given parameters(Doucet et al., 2001) while few mentioned the model fitting topic (Kantas et al., 2015). Yet with the development of data-driven modeling, the task of calibrating the state space model is an important problem frequently faced by practitioners and the observed data may be used to estimate the parameters of the model. Attempts to address the parameter estimation issue include gradient and EM-based maximum likelihood estimation algorithms (Poyiadjis et al., 2011; Andrieu et al., 2004), which hardly accommodate the growth of data length due to efficiency issues. Therefore, more efficient parameter estimation schemes for state space models are in deep need. With the above motivation, the first topic of this thesis tries to fill this gap by proposing two gradient free stochastic approximation (Kushner and Yin, 2003) algorithms for maximum likelihood estimation under the framework of state space models. We will show in Chapter 3 that with variance reduction techniques, the algorithms attain fast convergence rate with reduced computational cost.

The second topic is driven by the increasing need from various disciplines to make inferences from sequentially obtained observations that has connections to underlying latent factors. For example, Figure 1.1.1 shows a dynamic system of stock price and dividend for S&P500 index. It is believed in financial literatures (Campbell and Shiller, 1988; LeRoy and Porter, 1981; Wu, 1997) that this system is driven by an unobservable bubble series. A well-designed bubble detection model would benefit the econometric analysis of the market periodicity and the understanding of the economy situation. This leads to our proposed regime swithing state space model in Chapter 4, which achieves better bubble detection performance and explains more of the bubble dynamic. In clinic trial, the interstitial signals from a continuous glucose sensor and fingerstick measurement are driven by a latent blood glucose series (Dicker et al., 2013), which can be illustrated in Figure 1.1.2.

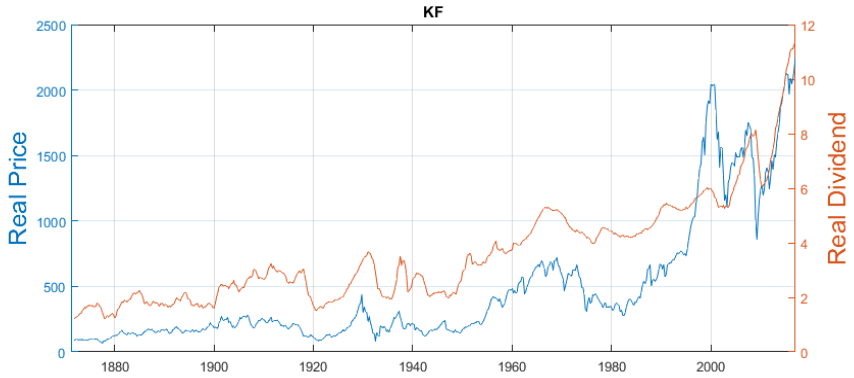


Figure 1.1.1: US S&P500 index quaterly price and dividend series.

An improvement in the continuous glucose monitoring algorithm will greatly advance the study of diabetes treatment. This improvement can also result from our proposed state space model, which is elaborated in Chapter 5. Furthermore, one might find state space models' application in an existing statistical model. For example, a generalized regression model, l_1 trend filtering (Kim et al., 2009) achieves the filtering function given a set of batch data. However, in time series analysis, the filtering objective are often required to be carried out on-line in real time to accommodate the fact that new observation often comes in sequentially. Chapter 6 addresses this issue by converting this statistical model into state space representation. To summarize, the latter part of this thesis is devoted to the above financial, biostatistics and statistical applications to provide a standard procedure of sequential data analysis using state space representation.

1.2 Outline of the thesis

Given sequentially observed data, the framework of calibration via state space model consists of three steps: converting the dynamic system into state space representation, fitting the model based on the observed data and making inference based on the fitted model. The rest of the thesis deals with several important aspects among the procedures. An outline of the subsequent chapter

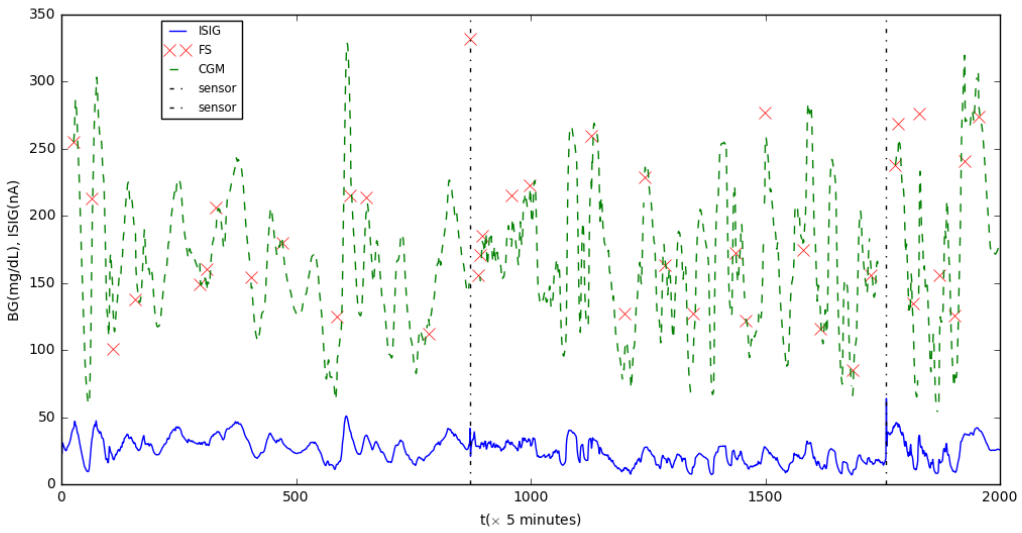


Figure 1.1.2: An illustration of the blood blucose (estimated by CGM), interstitial signal (ISIG) and finger-stick measurement (FS) system. Vertical lines represents the sensor replacement cycle.

contents can be given as follows:

Chapter 2 covers the main tools used to study the above topics, including state space models and its related filtering methods, Kalman Filtering for linear Gaussian models and sequential Monte Carlo for other cases. As a special class of state space models, conditional dynamic linear models (CDLM) provide certain convenience for model inference. CDLM and its associated filtering algorithm, Mixture Kalman Filter, are also widely adopted in the thesis. The toolkit also includes information criteria, which provide a standard for model selection.

In Chapter 3, we discuss the problem of fitting a state space model using maximum likelihood estimator. Two algorithms under the general framework of Stochastic Approximation methods are proposed. These two algorithms attain much faster convergence rate and less computational cost by variance reduction techniques which utilize the property of sequential Monte Carlo methods. On each iteration, a finite difference estimate of the score function is calculated with smoothed

approximate likelihood function which are calculated with one run of SMC algorithm. Two numerical examples are examined to compare the performance. Another contribution of Chapter 2 is the application of sequantial Monte Carlo methods in modeling and predicting the bond yield curve with regime-switching Dynamic Nelson-Siegel model, which also proves the improvements of the two algorithms in parameter estimation.

Chapter 4 presents a joint work with Hao Chang, as an illustration of state space modeling in finance applications. In this chapter we develop a state space model with regime switching to detect periodically collapsing rational bubbles in stock price. The present-value stock-price model is expressed in a state space form and the bubble process is modeled as a conditional dynamic linear system. We allow two to three regimes that switch by Markovian transition probability matrices while keeping the system conditionally linear and Gaussian given the regime. The asset-bubble system is then estimated by Mixture Kalman Filter (MKF). The efficacy of the proposed method is examined by simulated observations and real stock index of the US market. We demonstrate that with the associated likelihood-based model selection techniques, our proposed model with regime-switching better fits the bubble process and can detect most of the bubble collapsing periods in history.

Another application of CDLM is explored in the Chapter 5, in which real-time Blood Glucose Monitoring problem is discussed. Inspired by the biological structure of the biosensor signal, fingerstick measurement and blood glucose system, we employ the CDLM framework to address the continuous Glucose Monitoring problem. Detailed implementation of this SSM-based CGM algorithm includes two main component: periodical and proper parameter estimation and statistical inference(including estimation, prediction, etc) on blood glucose levels. The carefully designed algorithm is applied and assessed via an important dataset, Star 1 dataset. The performance comparison in both estimation and prediction shows the advantage of the proposed model.

In Chapter 6, a nonparametric regression model, $l1$ trend filtering method (Kim et al., 2009) is discussed as an example of state space models' application in statistical methods. Two trend filtering models out of state space representation, both of which have similar property as $l1$ trend

filtering, are proposed. With the implementation of sequential Monte Carlo methods as well as a greedy Viterbi algorithm, both trend filtering models can operate on-line rather than just on batch data. To better emphasize the two models' improvement in on-line trend filtering, a real world econometrics topic is introduced. The econometric example shows the competence of trend filtering as well as the efficiency of the proposed models.

Chapter 2

Preliminary

2.1 State space models

State space models (Doucet et al., 2001; West and Harrison, 1998; Liu and Chen, 1998) are a very popular class of time series models that have found numerous of applications in fields as diverse as statistics, ecology, econometrics, engineering and environmental sciences . A generalized state space model consists of two time series $\{X_t\}_{t\geq 0}$ and $\{Y_t\}_{t\geq 0}$, which are $\mathbb{R}^{n_x}$ and $\mathbb{R}^{n_y}$-valued respectively. $\{X_t\}_{t\geq 0}$ is a latent time series of initial density $\mu_\theta(x)$ and follows a Markov transition density $f_\theta(\cdot|x_{t-1})$ called state density, that is,

$$X_t|X_{t-1} = x_{t-1} \sim f_\theta(\cdot|x_{t-1}), \tag{2.1}$$

and $\{Y_t\}_{t\geq 0}$ is observable and dependent on $\{X_t\}_{t\geq 0}$ by the following observation density

$$Y_t|X_t = x_t \sim g_\theta(\cdot|x_t). \tag{2.2}$$

The state and observation densities can also be expressed as state and observation equations as

$$\begin{aligned} X_t &= s_t(\boldsymbol{X}_{t-1}, \epsilon_t), \\ Y_t &= h_t(\boldsymbol{X}_t, e_t), \end{aligned}$$

where ϵ_t and e_t are known state and observation innovations. A simple example can be given from an AR(1) plus noise system as below,

$$X_t = \theta X_{t-1} + V_t,\ V_t \sim N(0,1) \tag{2.3}$$

$$Y_t = X_t + W_t,\ W_t \sim N(0,1), \tag{2.4}$$

which is linear and Gaussian. However the state and observation density can be non-Gaussian and may also involve non-linearities. An example can be given by the Stochastic Volatility model (Sandmann and Koopman, 1998):

$$\begin{aligned} X_t &= \phi X_{t-1} + \sigma_x \eta_t, \\ Y_t &= \sigma_y e^{\frac{X_t}{2}} \xi_t, \end{aligned} \tag{2.5}$$

where η_t and ξ_t are independent standard Gaussian variables, Y_t is the demeaned return of a portfolio obtained by subtracting the average of all returns from the actual return and σ_y is the average volatility level. X_t drives the specific volatility level at time t.

For fixed parameters θ, the main objective of state space models is to, at each time, obtain or understand the latent states $\{X_t\}_{t\geq 0}$, given the entire sequence of observations $\{Y_t\}_{t\geq 0}$. All the information about $\{X_t\}_{t\geq 0}$ are given in the posterior distribution

$$p_\theta(\boldsymbol{x}_t|\boldsymbol{y}_t) \propto \prod_{s=1}^{t} g_\theta(y_s|x_s) f_\theta(x_s|x_{s-1}). \tag{2.6}$$

There are three main objectives under the framework of state space models:

(1) Filtering: One is interested in the marginal posterior distribution $p_\theta(x_t|\boldsymbol{y}_t)$, and estimating the current state $E(h(X_t)|Y_1, ..., Y_t)$.

(2) Prediction: One is interested in the marginal posterior distribution $p_\theta(x_{t+1}|\boldsymbol{y}_t)$, and estimating the current state $E(h(X_{t+1})|Y_1, ..., Y_t)$.

(3) Smoothing: One is interested in the posterior distribution $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{y}_t)$, and estimating the current state $E(h(\boldsymbol{X}_{t-1})|Y_1, ..., Y_t)$.

It is often desired to perform the inference in an on-line manner. Therefore real-time iterative algorithms are usually preferred due to its adaptivity to new incoming information. Depending on model specifications, there are two kinds of iterative algorithms for state space models: Kalman Filter (Kalman, 1960) for linear Gaussian models and sequential Monte Carlo (Liu and Chen, 1998; Doucet et al., 2001) for the other forms.

2.2 Kalman Filter

Linear Gaussian state space model (Kitagawa, 1996) is a widely adopted dynamic system in time series analysis. The popularity of linear Gaussian state space model comes from the model simplicity for estimation and prediction. A linear Gaussian state space model takes the form

$$\begin{aligned} x_t &= H_t x_{t-1} + W_t w_t, \\ y_t &= G_t x_t + V_t v_t, \end{aligned} \tag{2.7}$$

when $H_t,\ G_t,\ W_t,\ V_t$ are given matrices and $w_t,\ v_t$ follow standard normal distributions. The linear Gaussian state space model has connections to many classical time series model. The ARMA model with Gaussian innovation (Brockwell and Davis, 2013) can be translated into a linear Gaussian state space modle. For example, a AR(2) with Gaussian innovation model, denoted as

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + e_t, \ \ e_t \sim N(0, \sigma^2),$$

can be denoted as the following linear Gaussian model with no observation noise,

$$\begin{aligned} y_t &= \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{x}_t, \\ \boldsymbol{x}_t &= \begin{bmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}_{t-1} + \boldsymbol{w}_t, \end{aligned}$$

where $\boldsymbol{x}_t = [y_t, y_{t-1}]'$ and $\boldsymbol{w}_t = [e_t, 0]'$.The state space representation allows on-line inference and forecasting using the below Kalman Filtering techniques. Equation (2.3) is another example of linear Gaussian state space model.

With linear and Gaussian assumption, the posterior distribution $p_\theta(x_t|\boldsymbol{y}_t)$ is still Gaussian and can be obtained on-line with Kalman Filter (Kalman, 1960; Kalman and Bucy, 1961). Here we give a brief review of Kalman Filter. More details can be found in Harvey (1990) and Durbin and Koopman (2012). Denote $\boldsymbol{\mu}_t = E(x_t|y_1, ..., y_t),\ \Sigma_t = Cov(x_t|y_1, ..., y_t)$ as the posterior mean and covariance given the observation series up to time t. Kalman Filter runs the following recursion to update its estimation on $\boldsymbol{\mu}_t$ and Σ_t:

$$\begin{aligned} P_{t+1} &= H_{t+1}\Sigma_t H'_{t+1} + W_{t+1}W'_{t+1}, \\ S_{t+1} &= G_{t+1}P_{t+1}G'_{t+1} + V_{t+1}V'_{t+1}, \\ \boldsymbol{\mu}_{t+1|t} &= H_{t+1}\boldsymbol{\mu}_t, \\ \boldsymbol{e}_t &= Y_{t+1} - G_{t+1}H_{t+1}\boldsymbol{\mu}_t, \\ \boldsymbol{\mu}_{t+1} &= \boldsymbol{\mu}_{t+1|t} + P_{t+1}G'_{t+1}S^{-1}_{t+1}\boldsymbol{e}_t, \\ \Sigma_{t+1} &= P_{t+1} - P_{t+1}G'_{t+1}S^{-1}_{t+1}G_{t+1}P_{t+1}, \end{aligned} \tag{2.8}$$

where P_{t+1} and S_{t+1} stands for the predictive state and error covariances while $\boldsymbol{\mu}_{t+1|t}$ and $\boldsymbol{e}_t$ are the predictive state and error estimations. The matrix $P_{t+1}G'_{t+1}S^{-1}_{t+1}$ is often called the forwards (optimal) Kalman gain matrix. In addition, the data likelihood can be obtained as a by-product of

Kalman Filter with

$$L(\theta|\boldsymbol{y}_T) = -Tp/2\log(2\pi) - 1/2\sum_{t=1}^{T}\log|S_t| - 1/2\sum_{t=1}^{T}\boldsymbol{e}_t' S_t^{-1}\boldsymbol{e}_t, \tag{2.9}$$

where p is the observation dimension.

Although initiated as a filtering algorithm, Kalman Filter can be extended to do prediction and smoothing as well. Kalman prediction is a direct extension as the derivation of the predictive mean and covariance $\boldsymbol{\mu}_{t+1|t}$ and P_{t+1}. One needs to conduct those two equations for k steps if $\boldsymbol{\mu}_{t+k}|t = E(x_{t+k}|\boldsymbol{y}_t)$ is needed. As for smoothing where the conditional mean and covariance, $\boldsymbol{\mu}_{t|T} = E(x_t|\boldsymbol{y}_T)$ and $\Sigma_{t|T} = Cov(x_t|\boldsymbol{y}_T)$, the below backward Kalman smoothing (Rauch et al., 1965) steps are needed:

$$\begin{aligned} J_t &\triangleq \Sigma_t H_{t+1}' P_{t+1}^{-1}, \\ \boldsymbol{\mu}_{t|T} &= \boldsymbol{\mu}_t + J_t(\boldsymbol{\mu}_{t+1|T} - \boldsymbol{\mu}_{t+1|t}), \\ \Sigma_{t|T} &= \Sigma_t + J_t(\Sigma_{t+1|T} - P_{t+1})J_t'. \end{aligned} \tag{2.10}$$

The matrix J_t is the backwards Kalman gain matrix.

Moreover, Kalman Filter can be further adapted to accommodate for general state space models where linearity and Gaussian are partially violated. For example, the extended Kalman Filter (Gelb, 1974) and unscented Kalman Filter (Julier and Uhlmann, 1997) deals with non-linear but still Gaussian state space models while Gaussian Sum Kalman Filtering (Sorenson and Alspach, 1971) deals with state space models with mixture of Gaussian residuals. However, when faced with a non-linear and non-Gaussian state space model, the above approximations fail and numerical methods are needed.

2.3 The sequential Monte Carlo framework

In most of the applications, a state space model is often non-linear and non-Gaussian. For example, many important economic time series exhibit strong patterns of non-Gaussian or time-varying behavior. Regime switching, stochastic volatility, and time-varying parameter models have become increasingly popular over the last decade. In this section we discuss an effective Monte Carlo approach designed specifically for dealing with non-linear or non-Gaussian state space models. It is referred to as the sequential Monte Carlo (SMC) (Kong et al., 1994; Liu and Chen, 1998; Doucet et al., 2001) method. This method tries to utilize fully the sequential and dynamic nature of the state space models, yet enjoys the flexibility of the powerful Monte Carlo approaches. In this section, we briefly introduce the sequential Monte Carlo framework. We first briefly discuss the important concept of importance sampling, then build the SMC framework on it.

2.3.1 Importance sampling

One of the key components of SMC is importance sampling, which comes from the general Monte Carlo methods (Robert, 2004). A Monte Carlo method tries to generate a set of samples $x^{(1)}, ..., x^{(m)}$ from a target distribution $\pi(x)$. Statistical inferences, such as estimating $E_\pi(h(x))$ can then be given by the Monte Carlo samples like the mean estimator

$$E_\pi[h(x)] \approx \frac{\sum_{i=1}^{m} h(x^{(i)})}{m}.$$

Importance sampling operates in cases that the Monte Carlo samples are generated from a trial distribution q different from the target distribution π. Then due to the fact that

$$E_\pi[h(x)] = \int h(x)\pi(x)dx = \int h(x)\frac{\pi(x)}{q(x)}q(x)dx = E_q[h(x)w(x)],$$

where $w(x) = \frac{\pi(x)}{q(x)}$, the mean estimator under such Monte Carlo samples can be obtained by

$$E_\pi[h(x)] \approx \frac{\sum_{i=1}^{m} h(x^{(i)})w^{(j)}}{m},$$

where $w^{(j)} = \frac{\pi(x^{(j)})}{q(x^{(j)})}$.

It is often hard to calculate the exact value of $w^{(j)}$ due to the partial absence of the target distribution $\pi(x)$. Instead, if a normalized weight $\tilde{w}^{(j)} \propto w^{(j)}$, which differs from the true weight with only a normalizing constant, can be calculated. Then combined with the fact that

$$E_q[w(x)] = E_q[\frac{\pi(x)}{q(x)}] = 1,\ E_q[\sum_{j=1}^{m} w^{(j)}] = m,$$

A weighted average

$$\frac{1}{\sum \tilde{w}^{(j)}} \sum_{j=1}^{m} \tilde{w}^{(j)} h(x^{(j)}) = \frac{1}{\sum w^{(j)}} \sum_{j=1}^{m} w^{(j)} h(x^{(j)}) \approx E_\pi[h(x)],$$

can therefore be taken for the estimation. With this formulation, the normalized weight $\tilde{w}(x)$ often avoids the evaluation of the normalization constants in the distributions of π and q.

The estimation variance,

$$Var_q\{h(x^{(i)})w^{(j)}\} = \frac{1}{m} \int \{h(x)\frac{\pi(x)}{q(x)}\}^2 q(x)dx - E_\pi^2[h(x)],$$

is often considered as a criterion to measure the efficiency of a Monte Carlo estimation. This quantity depends on the selection of trial distribution. Therefore a set of Monte Carlo samples from a carefully selected trial distribution might provide an even more efficient estimator than that from the target distribution. This is also one of the motivations of importance sampling. Alternatively, efficient measurement can come from another quantity, named as effictive sample size (Kong et al.,

1994)

$$\boldsymbol{ESS} = \frac{m}{1 + cv^2(w)},$$

which represents the equivalent number of samples from target distribution π.

2.3.2 The sequential Monte Carlo methods

Based on the importance sampling structure, Liu and Chen (1998) summarized the following sequential Monte Carlo (SMC) framework, which consists the sequential importance sampling(SIS) and resampling step. The generalized SMC algorithm, under the state space model setting is given below.

In the sequential Importance Sampling (SIS) step, when the system evolves from stage t to $t+1$, a set of Monte Carlo paths $\{\boldsymbol{x}_{t+1}^{(j)}\}_{j=1,\ldots,m}$ are generated sequentially from a known trial distribution $q_\theta(x_{t+1}|\boldsymbol{x}_t, \boldsymbol{y}_{t+1})$ and assigned a normalized weight

$$\tilde{w}_{t+1}^{(j)} \propto \frac{\pi_{t+1}(\boldsymbol{x}_{t+1}^{(j)})}{\prod_{i=0}^{t} q_\theta(x_{i+1}^{(j)}|\boldsymbol{x}_i^{(j)}, \boldsymbol{y}_{i+1})},$$

where π denotes the target distribution. Adapted to the sequential characteristics of state space models, the weight calculation are carried out step-wisely, where the incremental weight

$$u_{t+1}^{(j)} \propto \frac{\pi_{t+1}(\boldsymbol{x}_{t+1}^{(j)})}{\pi_t(\boldsymbol{x}_t^{(j)}) q_\theta(x_{t+1}^{(j)}|\boldsymbol{x}_t^{(j)}, \boldsymbol{y}_{t+1})},$$

is calculated and the new weight updated by

$$\tilde{w}_{t+1}^{(j)} = u_{t+1}^{(j)} \tilde{w}_t^{(j)}.$$

Samely as importance sampling, the Monte Carlo estimator of any function $h(\cdot)$ can be obtained

by

$$\widehat{h(\boldsymbol{x}_{t+1})} = \frac{1}{\sum \tilde{\omega}_{t+1}^{(j)}} \sum_{j=1}^{m} h(\boldsymbol{x}_{t+1}^{(j)}) \tilde{\omega}_{t+1}^{(j)}.$$

Specifically, in the scope of this thesis, $\pi_{t+1}(\boldsymbol{x}_{t+1}^{(j)}) = p_\theta(\boldsymbol{x}_{t+1}^{(j)}|\boldsymbol{y}_{t+1})$. The weight becomes

$$w_{t+1}^{(j)} = \frac{\pi_{t+1}(\boldsymbol{x}_{t+1}^{(j)})}{\prod_{i=0}^{t} q_\theta(x_{i+1}^{(j)}|\boldsymbol{x}_i^{(j)}, \boldsymbol{y}_{i+1})},$$

and the normalized weight $\tilde{\omega}_{t+1}^{(j)}$ is set to be

$$\tilde{w}_{t+1}^{(j)} = p_\theta(\boldsymbol{y}_{t+1})\omega_{t+1}^{(j)} = \frac{\prod_{i=0}^{t} g_\theta(y_{i+1}|x_{i+1}^{(j)}) f_\theta(x_{i+1}^{(j)}|x_i^{(j)})}{\prod_{i=0}^{t} q_\theta(x_{i+1}^{(j)}|\boldsymbol{x}_i^{(j)}, \boldsymbol{y}_{i+1})}, \tag{2.11}$$

where the normalizing constant is the unknown likelihood function $p_\theta(\boldsymbol{y}_{t+1})$ while $f_\theta(\cdot)$ and $g_\theta(\cdot)$ denote the state and observation densities. Therefore the sequential decomposition of $\tilde{\omega}_{t+1}^{(j)}$ leads to the following formulation of sequential weight update,

$$\begin{aligned} u_{t+1}^{(j)} &= \frac{g_\theta(y_{t+1}|x_{t+1}^{(j)}) f_\theta(x_{t+1}^{(j)}|x_t^{(j)})}{q_\theta(x_{t+1}^{(j)}|\boldsymbol{x}_t^{(j)}, \boldsymbol{y}_{t+1})}, \\ \tilde{\omega}_{t+1}^{(j)} &= u_{t+1}^{(j)} \omega_t^{(j)}. \end{aligned} \tag{2.12}$$

A good trial distribution that approximates the target distribution well is key to SMC. There are continuous discussions on that (see Kitagawa, 1996; Lin et al., 2005; Pitt and Shephard, 1999). For example, the basic bootstrap filter (Kitagawa and Gersch, 1996), proposes to use the state density $f_\theta(x_{t+1}|x_t)$ as the trial distribution thus $u_{t+1}^{(j)} = g_\theta(y_{t+1}|x_{t+1}^{(j)})$. This algorithm is usually easy and fast when state density is easy to generate and observation density is easy to evaluate. However it might lose efficiency if the state equation is not representative for the whole posterior.

It is shown that variance of $\tilde{\omega}_t^{(j)}$ increases stochastically as t increases in the SIS step (Kong

et al., 1994). The increasing variance of $\tilde{\omega}_t^{(j)}$ leads to an undesired outcome called sample degeneracy (Liu and Chen, 1995), leading to a shrinking effective sample size. Instead of carrying the weight $\tilde{\omega}_t^{(j)}$ as the system evolves, it is legitimate and sometimes preferable to insert a resampling step between SIS recursions in order to stabilize the weight distribution (see Liu and Chen, 1995). Suppose at time t, streams of $\boldsymbol{x}_t's$ are generated as $\{\boldsymbol{x}_t^{(j)}\}_{j=1,...,m}$ from the trial distribution $q_\theta(x_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}_t)$ with the weight $\{\tilde{w}_t^{(j)}\}_{j=1,...,m}$. The resampling step samples a new set of$\{\boldsymbol{x}_t'^{(j)}\}_{j=1,...,m}$ from the old one according to probabilities proportional to $\alpha_t^{(j)}$. In order to offset the resampling distribution, the new weight of $\boldsymbol{x}_t^{(j)}$ is then set to $\tilde{\omega}_t^{(j)}/\alpha_t^{(j)}$. A typical choice of α_t could be $\tilde{\omega}_t^{\rho}$ with $0 < \rho \leq 1$. In many cases, $\rho = 1$.

Resampling step in algorithm allows more paths to naturally appear in areas of high posterior probability, which increases the effective sample size. Resampling methods involves simple random sampling, residual sampling (Liu and Chen, 1998), stratified sampling (Kitagawa, 1996) and many others. Design issues of resampling also involves the control of when to resample. One method is to resample with deterministic frequency every τ steps, where τ is some positive integer, while an adaptive resampling schedule using effective sample size as the monitor (Liu and Chen, 1995) could also be implemented. Detailed discussions can be found in (Doucet et al., 2001).

The following statement of SMC algorithm is listed as a summary:

Algorithm 1. *(Sequential Monte Carlo)*

- ***SIS step***

  (A) At each time $t+1$, for each $j = 1, ..., m$, generate an $x_{t+1}^{(j)}$ from the trial distribution $q_\theta(x_{t+1}|\boldsymbol{x}_t, \boldsymbol{y}_{t+1})$; attach it to $\boldsymbol{X}_t^{(j)}$ to form $\boldsymbol{x}_{t+1}^{(j)} = (\boldsymbol{x}_t^{(j)}, X_{t+1}^{(j)})$;

  (B) Compute the incremental weight $u_{t+1}^{(j)}$ thus normalized weight $\tilde{\omega}_{t+1}^{(j)}$ using Equation (2.12).

- ***Resampling step****(optional, when variance of $\omega_{t+1}^{(j)}$ are large)*

  (A) Sample a new set of $\{\boldsymbol{x}_t'^{(j)}\}_{j=1,...,m}$ from the old one proportional to $\alpha_t^{(j)}$;

(B) If $x_t^{(j)}$ is sampled, assign it a new weight $\tilde{\omega}_t^{(j)}/\alpha_t^{(j)}$.

2.3.3 Likelihood estimation under SMC

State space models in practice often contain unknown parameters. Maximum likelihood estimation is one important strand of parameter estimation techniques which requires the evaluation of the log likelihood function

$$l(\theta) = \log[L(\theta|\boldsymbol{y}_T)] = \log[p_\theta(\boldsymbol{y}_T)] = \log\big[\int p_\theta(\boldsymbol{y}_T, \boldsymbol{x}_T)d\boldsymbol{x}_T\big].$$

With the existence of latent variables in state space model, it is often difficult to directly obtain the likelihood function in closed form. However, SMC gives the likelihood estimation as a by-product (Pitt, 2002). Due to the fact that $p_\theta(\boldsymbol{y}_T)$ is just the normalization constant in the weight Equation (2.11), one can have

$$E_q\ \tilde{w}_{t+1}^{(j)} = E_q\ \omega_{t+1}^{(j)} p_\theta(\boldsymbol{y}_T) = E_p\ p_\theta(\boldsymbol{y}_T) = p_\theta(\boldsymbol{y}_T). \tag{2.13}$$

Therefore given a set of weighted Monte Carlo samples sequentially generated under θ, the particle approximation of $l(\theta)$ can be given by

$$\hat{l}(\theta) = \log \hat{p}_\theta(\boldsymbol{y}_T) = \log\Big(m^{-1}\sum_{j=1}^{m} \tilde{w}_T^{(j)}\Big). \tag{2.14}$$

Similarly, when resampling takes place in time $t = t_1, ..., t_k$,

$$\hat{l}(\theta) = \log \hat{p}_\theta(\boldsymbol{y}_T) = \sum_{i=1}^{k+1} \log\Big(m^{-1}\sum_{j=1}^{m} \tilde{w}_{t_i}^{(j)}\Big), \tag{2.15}$$

where we denote $t_{k+1} = T$. More discussions can be found in Chapter 3.

2.4 Conditional dynamic linear models and Mixture Kalman Filters

In this section, we focus on a special case of state space models, the conditional dynamic linear models (CDLM), which is a direct generalization of the linear Gaussian models and has been widely used in practice, such as regime-switching econometrics models, signal detection from jump dynamics, etc (see (Shephard, 1994) for more examples). A CDLM can be generally defined as follows:

$$\begin{aligned} x_t &= H_\lambda x_{t-1} + W_\lambda w_t, \\ y_t &= G_\lambda x_t + V_\lambda v_t \end{aligned} \qquad \text{if } \Lambda_t = \lambda \tag{2.16}$$

where $w_t \sim N(0, I)$, $v_t \sim N(0, I)$ and all coefficient matrices are known given λ. CDLM is a special case of state space models as it is linear and Gaussian given Λ_t. The Λ_t, which can be either continuous or discrete, is a latent indicator process with certain probabilistic structure. With discrete indicator variables, the model can be used to deal with ourliers, sudden jumps, system failures, regime changes and clutters. With carefully chosen continuous indicator variables, CDLMs can also accommondate state space models which is linear but non-Gaussian by approximating the non-Gaussian residuals using mixture of Gaussian distributions. In later chapters, calibration of sequential data which exhibits strong periodicity under the framework of CDLM will be elaborated. They are all good examples to see the flexibility and easy interpretation of CDLM.

The conditional linearity and Gaussian leads to certain convenience than ordinary state space models. Therefore an elegant and much more efficient marginalized SMC technique, called Mixture Kalman Filter (MKF), was proposed by Chen and Liu (2000). Whereas a straightforward sequential Monte Carlo filter uses a weighted sample of the state variable, $\{\boldsymbol{x}_t^{(j)}, \omega_t^{(j)}\}_{j=1,\ldots,m}$ to approximate $p_\theta(\boldsymbol{x}_t^{(j)}|\boldsymbol{y}_t)$, the MKF operates in the indicator space Λ_t, which is equivalent to marginalizing out the $\boldsymbol{x}_t$. Intuitively, MKF approximates the posterior of $\boldsymbol{x}_t$ by a mixture of Gaussian distributions based on the Monte Carlo samples of the indicator space $\Lambda_t^{(j)}$.

Note that given $\boldsymbol{\lambda}_{t+1}^{(j)},\ \boldsymbol{x}_t^{(j)},\ \boldsymbol{y}_{t+1}$, the posterior mean and covariance matrix of $\boldsymbol{x}_{t+1}$, marked

as $KF_{t+1}^{(j)} = \{\boldsymbol{\mu}_{t+1}(\boldsymbol{\lambda}_{t+1}^{(j)}), \Sigma_{t+1}(\boldsymbol{\lambda}_{t+1}^{(j)})\}$, can be calculated through a one step Kalman Filter (Kalman, 1960) by the following equations:

$$\begin{aligned} P_{t+1} &= H_{\lambda_{t+1}} \Sigma_t H'_{\lambda_{t+1}} + W_{\lambda_{t+1}} W'_{\lambda_{t+1}} \\ S_{t+1} &= G_{\lambda_{t+1}} P_{t+1} G'_{\lambda_{t+1}} + V_{\lambda_{t+1}} V'_{\lambda_{t+1}} \\ \boldsymbol{\mu}_{t+1} &= H_{\lambda_{t+1}} \boldsymbol{\mu}_t + P_{t+1} G'_{\lambda_{t+1}} S_{t+1}^{-1} (Y_{t+1} - G_{\lambda_{t+1}} H_{\lambda_{t+1}} \boldsymbol{\mu}_t) \\ \Sigma_{t+1} &= P_{t+1} - P_{t+1} G'_{\lambda_{t+1}} S_{t+1}^{-1} G_{\lambda_{t+1}} P_{t+1} \end{aligned} \tag{2.17}$$

Then by setting the target distribution as $p_\theta(\boldsymbol{\lambda}_t|\boldsymbol{y}_t)$ and trial distribution as $q_\theta(\lambda_{t+1}|\boldsymbol{\lambda}_t, KF_t, y_{t+1})$, the MKF algorithm, as a special case of sequential Monte Carlo algorithm, can be summarized as below:

Algorithm 2. *(Mixture Kalman Filter)*

- ***SIS step***

  (A) At each time $t+1$, for each $j = 1, ..., m$, generate a $\lambda_{t+1}^{(j)}$ from the trial distribution $q_\theta(\lambda_{t+1}|\boldsymbol{\lambda}_t, KF_t, y_{t+1})$; attach it to $\boldsymbol{\lambda}_t^{(j)}$ to form $\boldsymbol{\lambda}_{t+1}^{(j)} = (\boldsymbol{\lambda}_t^{(j)}, \lambda_{t+1}^{(j)})$;

  (B) Obtain $KF_{t+1}^{(j)}$ by the Kalman Filter described in (2.17);

  (C) Compute the incremental weight $u_{t+1}^{(j)}$ thus weight $\omega_{t+1}^{(j)} = \omega_t^{(j)} \times u_{t+1}^{(j)}$, where

$$u_{t+1}^{(j)} = \frac{p_\theta(\boldsymbol{\lambda}_{t+1}|\boldsymbol{y}_{t+1})}{p_\theta(\boldsymbol{\lambda}_t|\boldsymbol{y}_t) q_\theta(\lambda_{t+1}|\boldsymbol{\lambda}_t, KF_t, y_{t+1})}.$$

- ***Resampling step****(optional, when variance of $\omega_{t+1}^{(j)}$ are large)*

  (A) Sample a new set of $\{\boldsymbol{\lambda}_t'^{(j)}\}_{j=1,...,m}$ from the old one according to $\alpha_t^{(j)}$. Accordingly, if $\lambda_t^{(j)}$ is sampled, let $KF_t^{(j)}$ be the one with it;

  (B) If $\lambda_t^{(j)}$ is sampled, assign it a new weight $\omega_t^{(j)}/\alpha_t^{(j)}$.

$$
\begin{array}{ccccccc}
... & & y_t & & y_{t+1} & & ... \\
... & \rightarrow & \lambda_t & \xrightarrow{p(\lambda_{t+1}|\boldsymbol{\lambda_t}, KF_t, y_{t+1})} & \lambda_{t+1} & \rightarrow & ... \\
 & & \updownarrow & & \updownarrow & & \\
... & \rightarrow & KF_t & \xrightarrow{Kalman\ Filter} & KF_{t+1} & \rightarrow & ... \\
 & & \downarrow & & \downarrow & & \\
... & \rightarrow & \tilde{w}_t(\boldsymbol{\theta}) & \xrightarrow{\tilde{u}_{t+1}^{(j)}(\boldsymbol{\theta})} & \tilde{w}_{t+1}(\boldsymbol{\theta}) & \rightarrow & ...
\end{array}
$$

MKF algorithm

When Λ_t take values in a finite discrete set $\mathfrak{T}$, the incremental weight $u_{t+1}^{(j)}$ can be simplified as

$$u_{t+1}^{(j)} \propto p(y_{t+1}|KF_t^{(j)}) = \sum_{i \in \mathfrak{T}} p(y_{t+1}|KF_t^{(j)}, \Lambda_{t+1} = i) p(\Lambda_{t+1} = i|\boldsymbol{\lambda}_t^{(j)}),$$

where $p(\Lambda_{t+1} = i|\boldsymbol{\lambda}_t^{(j)})$ is the prior transition probability for the indicator and $p(y_{t+1}|KF_t^{(j)}, \Lambda_{t+1} = i)$ is a by-product of the Kalman Filter, that is,

$$p(y_{t+1}|\boldsymbol{y}_t^{(j)}, \lambda_{t+1}) \sim N(G_{\lambda_{t+1}} H_{\lambda_{t+1}} \boldsymbol{\mu}_t, S_{t+1}). \tag{2.18}$$

And the MKF SIS step becomes,

(A) At each time $t+1$, for each $j = 1, ..., m$ and then each $\Lambda_{t+1} = i,\ i \in \mathfrak{T}$, run the one-step Kalman Filter to obtain

$$v_i^{(j)} \propto p(y_{t+1}|KF_t^{(j)}, \Lambda_{t+1} = i) p(\Lambda_{t+1} = i|\boldsymbol{\lambda}_t^{(j)});$$

(B) Sample a $\lambda_{t+1}^{(j)}$ from the set $\mathfrak{T}$, with probability proportional to $v_i^{(j)}$. Let $KF_t^{(j)}$ be the one with it;

(C) The incremental weight $u_{t+1}^{(j)} = \sum_{i \in \mathfrak{T}} v_i^{(j)}$ and weight is $\omega_{t+1}^{(j)} = \omega_t^{(j)} \times u_{t+1}^{(j)}$.

2.5 Information criteria

In time series modeling, one generally needs to deal with the problem of model selection. A standardized criterion that favors the most appropriate model among a set of potentially good ones is desired by both researchers and practitioners. There are many successful proposals to this problems, among which the Akaike's information criterion (AIC) (Akaike, 1998) and Bayesian information criterion (BIC) (Schwarz et al., 1978) remain the most widely known and applied tools. Below we briefly introduce those two information criteria. Details could be found in Sakamoto et al. (1986).

Ultimately, AIC tries to evaluate the discrepancy between the approximating model represented by a set of parameters θ and the 'true' or generating model represented by another set of parameters θ_0. The discrepancy is then defined as

$$d_n(\theta, \theta_0) = E_0\{-2\log L(\theta|\boldsymbol{Y}_n)\},$$

where E_0 denote the expectation under the generating model, and $L(\theta|\boldsymbol{Y}_n)$ is the likelihood of the approximating model. With a realized set of estimates $\hat{\theta_n}$, the discrepancy could be further represented by

$$d_n(\hat{\theta_n}, \theta_0) = E_0\{-2\log L(\theta|\boldsymbol{Y}_n)\}|_{\theta=\hat{\theta_n}}. \tag{2.19}$$

However without knowledge of θ_0 the evaluation (2.19) is still impossible. Akaike et al. (1973) found that $-2\log L(\hat{\theta_n}|\boldsymbol{Y}_n)$ is a biased estimator of $d_n(\hat{\theta_n}, \theta_0)$ and the bias under θ_0,

$$E_0\{d_n(\hat{\theta_n}, \theta_0)\} - E_0\{-2\log L(\hat{\theta_n}|\boldsymbol{Y}_n)\},$$

can often be asymptotically estimated by $2k$ where k is the dimension of θ_n. Therefore

$$\text{AIC} = -2\log L(\hat{\theta_n}|\boldsymbol{Y}_n) + 2k, \tag{2.20}$$

can be a good measure of the distance between the examined model to the 'true' model. Intuitively, the first term $-2\log L(\hat{\theta_n}|\boldsymbol{Y}_n)$ explains the 'goodness of fit' and the second term $2k$ corresponds to the 'penalty' of parameter numbers. If one candidate model A exhibits lower AIC than another candidate model B, then model A is estimated to have better goodness of fit than model B by AIC measurement.

There are several variants of AIC that also have gained acknowledgement, among which Bayesian information criterion (BIC) is usually included in association of AIC. BIC approximates the same discrepancy between the approximating model and the 'true' model under the Bayesian structure and exponential family assumption. The different construction leads to

$$\text{BIC} = -2\log L(\hat{\theta_n}|\boldsymbol{Y}_n) + 2k\log n, \tag{2.21}$$

where n is the sample size. Posada and Buckley (2004) presents more detailed discussion over BIC. Intuitively, BIC penalizes more on the dimension of parameter space than AIC. Therefore in practice AIC tends to favor complicated modles with more parameters while BIC agrees with simpler models.

Chapter 3

A smoothed Stochastic-Approximation approach for likelihood estimation of State space models and conditional dynamic linear models

3.1 Introduction

When performing paramter estimation of state space model, one issue of simulation-based likelihood inference is that the approximate likelihood function is not continuous due to the independent Monte Carlo noise on each evaluation. This makes maximising the approximate likelihood surface questionable and the standard optimisation algorithms unstable. For sequential Monte Carlo, even when the same random seed is used for each evaluation, continuity is still not guranteed due to the fact that a small change in the parameter may cause a different ordering in the resampling step (Kantas et al., 2009). Poyiadjis, Doucet and Singh (2011) proposes to bypass this difficulty by approximating the score function and observed information matrix, instead of likelihood, and apply gradient ascent methods. The score functions is estimated through the Fisher identity where it can be written as the expectation of $\nabla p_\theta(\boldsymbol{x}_T, \boldsymbol{y}_T)$, where ∇ denotes the gradient with respect to θ, under the density $p_\theta(\boldsymbol{x}_T \mid \boldsymbol{y}_T)$, given the fact that particles from $p_\theta(\boldsymbol{x}_T \mid \boldsymbol{x}_T)$ can be obtained SMC algorithms. This requires the ability to calculate $\nabla p_\theta(\boldsymbol{x}_T, \boldsymbol{y}_T)$. For situations where it is possible to sample from the state equation but difficult to evaluate, Ionides et al (2006) and Ionides et al (2011) propose a gradient-free method by applying the finite-difference estimate of score function. They propose to add an artificial noise term with mean 0 and variance τ to the likelihood estimation. Then the score function is estimated by the finite difference between the estimated posterior mean and the parameter value. However a successful implementation requires proper tuning of the

decreasing series of τ, for more runs of SMC algorithms are required when τ gets smaller. On the other hand, Hrzeler, Knsch (2001) proposes to estimate the likelihood function by using the filtered or smoothed particles simulated for a single parameter value as an importance sample. Then maximum likelihood estimation can be obtained by grid-based optimisation on the approximate likelihood surface, and the parameter value used to construct the importance sampler is updated when the importance weights are dominated by a few of them. The drawback is that when using filtered particles, the computation cost is $O(TN^2)$, and smoothing techniques are needed for the cost to be $O(TN)$. For some other techniques such as Bayesian estimation and on-line estimation, see Kantas et al (2015) for a comprehensive review.

Here we propose to perform maximum likelihood estimation through a gradient-free stochastic approximation algorithm using filtered sample and $O(TN)$ computational cost. On each iteration, a finite difference estimate of the score function is calculated with smoothed approximate likelihood function which are calculated with one run of SMC algorithm. Specifically, at each iteration, the filtered sample from the current parameter value is used as an importance sample to approximate the likelihood function in a small neighbourhood. Even if the neighbourhood is very small in order for the finite difference to be an accurate approximation of the score, smoothness is guranteed by using the same set of filtered sample.

The remainder of this chapter is organized as follows. In Section 3.2, we review the two schemes of finite-difference stochastic approximation. In Section 3.3, two new algorithms are proposed and applied to the conditional dynamic linear model, which is a special case of general state space model. In Section 3.4, the new algorithms are illustrated in two simulated examples and a real data analysis with the regime-switching Dynamic Nelson Siegel model. The chapter is concluded in Section 3.5.

3.2 Review of Stochastic Approximation

Stochastic approximation implements the gradient ascent optimisation with numerical approximation of the gradient function. It is used when the gradient is in the form of an expectation. In

the well-known Robbins-Monro algorithm (Robbins and Monro, 1951), in each iteration, instead of using an accurate approximation by averaging many noisy measurements of the gradient, only one measurement is used to update the parameter. The finite difference stochastic approximation (FDSA) (Kiefer and Wolfowitz, 1952), or Kiefer-Wolfowitz algorithm, is a gradient-free stochastic approximation which approximates the gradient function by finite-difference estimate. A direct application of FDSA for maximising the likelihood of state space model, studied in Poyiadjis et al. (2003), gives the following recursion

$$\begin{aligned}\theta_n &= \theta_{n-1} + a_n \widetilde{\nabla l}(\theta_{n-1}),\\ \widetilde{\nabla_i l}(\theta_{n-1}) &= \frac{\widetilde{l}(\theta_{n-1} + c_n e_i) - \widetilde{l}(\theta_{n-1} - c_n e_i)}{2c_n},\end{aligned} \tag{3.1}$$

where e_i denotes the p dimension vector with 1 in the i^{th} entry and 0 elsewhere, $i = 1, \ldots, p$, and $\{a_n\}_{n\geq 1}$, $\{c_n\}_{n\geq 1}$ are two sequences that are typically small, positive and converge to zero as n goes to ∞. When the following conditions hold,

$$a_n \to 0,\ c_n \to 0, \sum_{n=1}^{\infty} a_n = \infty,\ \sum_{n=1}^{\infty} (a_n/c_n)^2 < \infty,$$

θ_n, and further the moving average of θ_n, $\sum_{i=1}^{n} \theta_i$ converges to the maximiser of $l(\theta)$ in probability (Broadie et al., 2011). A typical choice of the tuning parameters is $a_n = n^{-1}$ and $c_n = n^{-1/3}$. Choices in the form of $a_n = \beta/(n+\gamma)$, $c_n = n^{-1/3}$ are studied in Broadie et al. (2011). Futher discussion of tuning parameters is out of the scope of this chapter.

Each iteration of FDSA requires $2p$ evaluation of $l(\theta)$, and the computation cost increases with p. Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm (Spall, 1992) is proposed to address this problem by introducing a p dimension random perturbation vector Δ_n and setting

$$\widetilde{\nabla_i l}(\theta_{n-1}) = \frac{\widetilde{l}(\theta_{n-1} + c_n \Delta_n) - \widetilde{l}(\theta_{n-1} - c_n \Delta_n)}{2c_n \Delta_{n,i}}. \tag{3.2}$$

Each iteration of SPSA only needs to evaluate $l(\theta)$ twice, and so the cost of each step when searching in the parameter space is does not increase with p. Although theory shows that the number of iterations needed for SPSA to achieve convergence may be similar to that for FDSA (Spall, 1992), it is found that its performance is more sensitive to the tuning parameters and generally needs more iterations (Poyiadjis et al., 2003).

Poyiadjis et al. (2003) applies FDSA and SPSA to state space model by evaluating the likelihood values at each iteration with independent runs of SMC algorithms. There are two drawbacks with this implementation. One is that due to the independent Monte Carlo noise in $\widetilde{l}(\theta)$, the variances of the difference in (3.1) and (3.2) do not decrease with c_n, and so the variance of $\widetilde{\nabla l}(\theta)$ increases to infinity as c_n decreases to 0. As pointed out in (Reference: Klaiman, Spall and Naiman 1994), when the variance of gradient estimate is unbounded, the convergence of θ_n in classical FDSA or SPSA typically has the rate $O(n^{-1/3})$, slower than the rate $O(n^{-1/2})$ of Robbins-Monro algorithm, hence requiring much more iterations. The other one is that even if common random number is used to calculate $\widetilde{l}(\theta)$, as in (Reference: Kleinman et al, 1999), $\widetilde{l}(\theta)$ is still discontinuous due to the fact that a small change in θ changes the importance weights $\{w_t^{(j)}\}_{j=1}^N$ in the resampling step and generates a different set of resampled particles (Reference: Kantas et al, 2015). Therefore the variance of $\widetilde{\nabla l}(\theta)$ is still unbounded.

3.3 Two Stochastic Approximation Schemes

3.3.1 Smoothed likelihood approximation

Suppose the resampling takes place at time $t_1, \ldots, t_k$. The likelihood function at any θ can be written as

$$\frac{p_\theta(\boldsymbol{y}_T)}{p_{\theta_0}(\boldsymbol{y}_{t_k})} = \int a(\theta, \theta_0, \boldsymbol{x}_T, \boldsymbol{y}_T) w_T(x_{t_k:T}; \theta_0, t_k) p_{\theta_0}(\boldsymbol{x}_{t_k} | \boldsymbol{y}_{t_k}) \prod_{t=t_k+1}^{T} q_{\theta_0}(x_t | x_{t-1})\, d\boldsymbol{x}_T, \quad (3.3)$$

$$\text{where } a(\theta, \theta_0, \boldsymbol{x}_T, \boldsymbol{y}_T) = \frac{p_\theta(\boldsymbol{x}_T, \boldsymbol{y}_T)}{p_{\theta_0}(\boldsymbol{x}_T, \boldsymbol{y}_T)} = \prod_{t=1}^{T} \frac{g_\theta(y_t|x_t) f_\theta(x_t|x_{t-1})}{g_{\theta_0}(y_t|x_t) f_{\theta_0}(x_t|x_{t-1})}.$$

Equation (3.3) suggests that with one run of SMC algorithm given parameter value θ_0, after the weighting step at time T, $p_\theta(\boldsymbol{y}_T)/p_{\theta_0}(\boldsymbol{y}_{t_k})$ can be estimated by $N^{-1}\sum_{j=1}^{N} w_T^{(j)} a(\theta, \theta_0, \boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T)$. Therefore we propose the following estimate of $p_\theta(\boldsymbol{y}_T)$,

$$\widehat{p}_\theta(\boldsymbol{y}_T) = \prod_{i=1}^{k} \left(N^{-1} \sum_{j=1}^{N} w_{t_i}^{(j)} \right) \left(N^{-1} \sum_{j=1}^{N} w_T^{(j)} a(\theta, \theta_0, \boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T) \right), \tag{3.4}$$

and the estimate of $l(\theta)$, $\widehat{l}(\theta) = \log(\widehat{p}_\theta(\boldsymbol{y}_T))$. By comparing $\widehat{p}_\theta(\boldsymbol{y}_T)$ with $\widetilde{p}_\theta(\boldsymbol{y}_T)$, it can be seen in the last factor of $\widetilde{p}_\theta(\boldsymbol{y}_T)$ is replaced by the SMC estimate of $p_{\theta_0}(\boldsymbol{y}_T) E_{\theta_0}[a(\theta, \theta_0, \boldsymbol{x}_T, \boldsymbol{y}_T) \mid \boldsymbol{x}_T]$. As $N \to \infty$, by slightly modifying the proof of asymptotic normality of $\widetilde{p}_\theta(\boldsymbol{y}_T)$ in Del Moral (2004), and that of $\widehat{h}_t$ in Chopin (2004), it is easy to show that $\widehat{p}_\theta(\boldsymbol{y}_T)/p_\theta(\boldsymbol{y}_T)$ is asymptotic normal with mean being unity and variance given by,

$$\begin{aligned}
&\frac{1}{N} \left(\sum_{i=1}^{k} \int \frac{p_{\theta_0}(\boldsymbol{x}_{t_i} \mid \boldsymbol{y}_T)^2}{p_{\theta_0}(\boldsymbol{x}_{t_{i-1}} \mid \boldsymbol{y}_{t_{i-1}}) q_{\theta_0}(x_{t_{i-1}+1:t_i} \mid \boldsymbol{x}_{t_i-1})} \, d\boldsymbol{x}_{t_i} - k \right. \\
&\left. + \int \frac{p_{\theta_0}(\boldsymbol{x}_T \mid \boldsymbol{y}_T)^2}{p_{\theta_0}(\boldsymbol{x}_{t_k} \mid \boldsymbol{y}_{t_k}) q_{\theta_0}(x_{t_k+1:T} \mid \boldsymbol{x}_{t_k})} \left[\frac{p_\theta(\boldsymbol{x}_T \mid \boldsymbol{y}_T)}{p_{\theta_0}(\boldsymbol{x}_T \mid \boldsymbol{y}_T)} - 1 \right]^2 d\boldsymbol{x}_T \right).
\end{aligned}$$

Asymptotic variances of $\widetilde{p}_\theta(\boldsymbol{y}_T)$ and $\widehat{h}_t$ can also be seen in Doucet and Johansen (2009). Furthermore, $\widehat{p}_\theta(\boldsymbol{y}_T)$ is also unbiased of $p_\theta(\boldsymbol{y}_T)$, the proof of which is given in the appendix. Therefore by delta method, $\widehat{l}(\theta)$ is asymptotically unbiased to $l(\theta)$ and asymptotic normal with convergence rate $\sqrt{N}$.

Specifically for CDLM with indicator process in discrete space, the weight adjusting function $a(\theta, \theta_0, \boldsymbol{\lambda}_T^{(j)}, \boldsymbol{y}_T)$ equals,

$$\frac{p_\theta(\boldsymbol{\lambda}_T^{(j)}, \boldsymbol{y}_T)}{p_{\theta_0}(\boldsymbol{\lambda}_T^{(j)}, \boldsymbol{y}_T)} = \frac{\prod_{i=1}^{T} p_\theta(y_i \mid \boldsymbol{\lambda}_i^{(j)}, \boldsymbol{y}_{i-1}) p_\theta(\lambda_i^{(j)} \mid \tilde{\lambda}_{i-1}^{(j)})}{\prod_{i=1}^{T} p_{\theta_0}(y_i \mid \boldsymbol{\lambda}_i^{(j)}, \boldsymbol{y}_{i-1}) p_{\theta_0}(\lambda_i^{(j)} \mid \tilde{\lambda}_{i-1}^{(j)})},$$

where $p_\theta(y_i \mid \boldsymbol{\lambda}_i^{(j)}, \boldsymbol{y}_{i-1})$ is given in (2.18) and can be calculated by Kalman filter given the value of θ. Since $p_\theta(\boldsymbol{\lambda}_T, \boldsymbol{y}_T) = p_\theta(\boldsymbol{y}_T \mid \boldsymbol{\lambda}_T) p_\theta(\boldsymbol{\lambda}_T)$, $a(\theta, \theta_0, \boldsymbol{\lambda}_T^{(j)}, \boldsymbol{y}_T)$ can be interpreted as the product

of likelihood ratio of the linear-Gaussian model, conditional on the indicator process $\boldsymbol{\lambda}_T$, and the ratio of prior density of $\boldsymbol{\lambda}_T$, between the new and simulator parameters, θ and θ_0.

Example. To illustrate the accuracy of $\widehat{l}(\theta)$, consider estimating the log-likelihood function of an autoregressive model with order one. The true parameter value generating observations is 0.7. We compare $\widetilde{l}(\theta)$ and three $\widehat{l}(\theta)$ using different simulation parameters θ_0 for generating the particles, and results are reported in Figure 3.3.1. We can see that when the true parameter value is used for simulation, $\widehat{l}(\theta)$, using one set of particles, gives very similar approximation as $\widetilde{l}(\theta)$ which simulates a new set of partics for each θ. When θ_0 is far away from the true parameter value, both methods have similar approximation accuracy for θ around θ_0, which is needed for an accurate estimate of the gradient function at θ_0 via finite difference. Although $\widehat{l}(\theta)$ is increasingly biased when θ is far away from θ_0, it still gives the correct direction of the maximum of likelihood function. This is very useful in gradient descent algorithms.

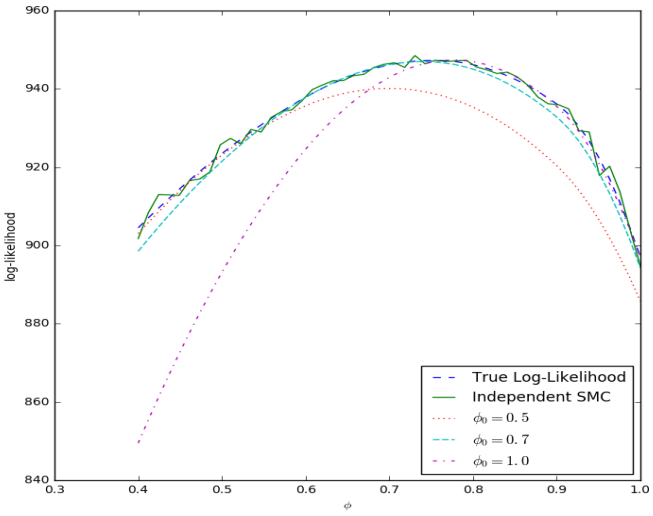


Figure 3.3.1: Likelihood approximation of an AR(1) model using SMC estimate and smoothed SMC estimates simulating particles from different θ_0.

3.3.2 Stochastic approximation with smoothed SMC likelihood

Let θ_n be the n_{th} update of the parameter θ. Then with a single run of SMC algorithm at θ_n and the proposed loglikelihood estimate $\widehat{l}(\theta)$, the finite-difference estimate of the score function at θ_n can be given as

$$\begin{aligned}\widehat{\nabla_i l}(\theta_n) &= \frac{\widehat{l}(\theta^f_{n,i}) - \widehat{l}(\theta^b_{n,i})}{2c_n\Delta_{n,i}} \\ &= \frac{\log\left(N^{-1}\sum_{j=1}^N w_T^{(j)} a(\theta^f_{n,i}, \theta_n, \boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T)\right) - \log\left(N^{-1}\sum_{j=1}^N w_T^{(j)} a(\theta^b_{n,i}, \theta, \boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T)\right)}{2c_n\Delta_{n,i}},\end{aligned}$$

where $\theta^f_{n,i}$ and $\theta^b_{n,i}$ are the forward and backward parameters that, together with $\Delta_{n,i}$, are algorithmically specified. With $\widehat{\nabla l}(\theta_n)$, we propose the following algorithm to maximise $l(\theta)$.

Algorithm 3. *Stochastic Approximation with smoothed SMC likelihood(SA-SMCw)*

Suppose the parameter estimate at iteration n is θ_n.

(A) For $i = 1, \ldots, p$, let

- *$\theta^f_{n,i} = \theta_n + c_{n+1}e_i$ and $\theta^b_{n,i} = \theta_n - c_{n+1}e_i$, if FDSA is used,*
- *$\theta^f_{n,i} = \theta_n + c_{n+1}\Delta_{n+1}$ and $\theta^b_{n,i} = \theta_n - c_{n+1}\Delta_{n+1}$, if SPSA is used,*

where Δ_n is a random perturbation vector.

(B) Run the SMC algorithm using θ_n to obtain $\widehat{l}(\theta)$, and calculate the gradient function $\widehat{\nabla l}(\theta_n)$.

(C) Update the parameter by $\theta_{n+1} = \theta_n + a_n\widehat{\nabla l}(\theta_n)$

Comparing to FDSA or SPSA in (Reference: Poyiadjis et al, 2006) where $\widetilde{l}(\theta)$ is used to calculate the finite difference, Algorithm 3 makes improvement in two aspects. First, in each iteration, SMC algorithm only needs to run once for both FDSA and SPSA implementation, instead of $2p$ runs for FDSA or 2 runs for SPSA in Poyiadjis et al(2006). Second, the variance of gradient estimate is smaller than that when $\widetilde{l}(\theta)$ is used. Intuitively, this is because when $\theta^f_{n,i}$ and $\theta^b_{n,i}$ are close,

$\widehat{l}(\theta^f_{n,i})$ and $\widehat{l}(\theta^b_{n,i})$ are positively correlated while $\widetilde{l}(\theta^f_{n,i})$ and $\widetilde{l}(\theta^b_{n,i})$ are independant. Remarkably, the following result shows that as $c_n \to 0$, the variance of the gradient estimate is bounded, comapred to the unbounded variance when $\widetilde{l}(\theta)$ is used, as stated in Section 2.2. Assuming the gradient function of $p_\theta(\boldsymbol{x}_T, \boldsymbol{y}_T)$ with respect to θ exists, by Taylor expansion,

$$\begin{aligned}\widehat{\nabla_i l}(\theta_n) &= \frac{\sum_{j=1}^N \nabla_i a(\theta, \theta_n, \boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T) w_T^{(j)} \mid_{\theta=\theta'}}{\sum_{j=1}^N a(\theta', \theta_n, \boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T) w_T^{(j)}}, \\ &= \frac{\sum_{j=1}^N \nabla_i \text{log} p_\theta(\boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T) \mid_{\theta=\theta'} a(\theta', \theta_n, \boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T) w_T^{(j)}}{\sum_{j=1}^N a(\theta', \theta_n, \boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T) w_T^{(j)}},\end{aligned}$$

where θ' satisfies $\|\theta' - \theta^b_{n,i}\| \leq \|\theta^f_{n,i} - \theta^b_{n,i}\|$. Then as $c_n \to 0$, $\theta^b_{n,i}$ and $\theta^f_{n,i}$ converges to θ_n almost surely, and

$$\widehat{\nabla_i l}(\theta_n) \to \sum_{j=1}^N \nabla_i \text{log} p_{\theta_n}(\boldsymbol{x}_T^{(j)}, \boldsymbol{y}_T) \frac{w_T^{(j)}}{\sum_{j=1}^N w_T^{(j)}}, \tag{3.5}$$

almost surely. By standard SMC asymptotic results (Doucet and Johansen), as $N \to \infty$, the above limit is asymptotic normal with finite variance. Therefore $\widehat{\nabla_i l}(\theta_n)$ has bounded variance as $c_n \to 0$.

3.3.3 Accelerated stochastic approximation with smoothed SMC likelihood

To emphasize its dependence on the simulator parameter θ_0, we denote the quasi-loglikelihood function $\widehat{l}(\theta)$ from (3.4) by $\widehat{l}(\theta; \theta_0)$. Since $\widehat{l}(\theta; \theta_0)$ augments the observations with a sample of the latent process simulated using parameter θ_0, it is natural to consider maximising the likelihood by iterating between maximising $\widehat{l}(\theta)$ and simulating the latent process using the maximiser of $\widehat{l}(\theta; \theta_0)$. However when θ is further away from θ_0, the approximation error of $\widehat{l}(\theta; \theta_0)$ may increase, as illustrated in Figure 3.3.1 and the maxmiser of $\widehat{l}(\theta; \theta_0)$ may be further away from the maximum likelihood estimator than θ_0. This is because the particles generated using θ_0 may not cover the high density area of $p_\theta(\boldsymbol{x}_T \mid \boldsymbol{y}_T)$ well. Since many commonly used optimisation algorithms, e.g. DFP (Davidon, 1991) and BFGS (Broyden, 1970), are iterative algorithms, instead of using the

maximiser, it would help to be more conservative by running a fixed number of iteration along the line and using the updated parameter. Therefore we propose the following recursive algorithm.

Algorithm 4. *Accelerated Stochastic Approximation with smoothed SMC likelihood(ASA-SMCw)*

Suppose the parameter estimate at iteration n is θ_n.

(A) Run the SMC algorithm using θ_n to obtain $\widehat{l}(\theta;\theta_n)$.

(B) Iterates an optimisating algorithm for maximising $\widehat{l}(\theta;\theta_n)$ over θ and stops when either K of this inner iterations have been run or the algorithm converges.

(C) Let θ_{n+1} be the ending update in (B).

Algorithm 3 is a special case of Algorithm 4, which can be seen by using FDSA or SPSA for the maximisation and setting K to be 1 in step (B) above. The improvement of using $K > 1$ is illustrated in numerical example 3.4.1.

The ASA-SMCw algorithm is different from the EM algorithm used for state space model (Dempster et al., 1977) in that the M-step of those EM algorithms uses the maximiser of the likelihood estimate to update the simulator parameter, while here a maximum number of inner iteration is set to avoid using maximiser every time. It can be seen from the numerical examples that this is necessary.

3.4 Emiprical Studies

Here the perform of the new algorithms for approximating the maximum likelihood estimator are illustrated in three examples. Three methods are compared, including SA using the ordiniary SMC/MKF likelihood(OSA), SA using the smoothed SMC likelihood(SA-SMCw) as in Algorithm 3 and the accelerated SA using the smoothed SMC likelihood(ASA-SMCw) as in Algorithm 4. For ASA-SMCw, $K = 10$ is used. In all exmaples, results of using the FDSA scheme in all three methods are reported(Comparison of those using the SPSA scheme show similar patterns). Identical tuning parameters $a_n = c/(n+5)$ and $c_n = 1/n^{1/3}$ are used for all algorithms. For r

experiment runs, define the root mean square error(RMSE) of estimating true parameter value θ_c at iteration n by $\sqrt{r^{-1}\sum_{l=1}^{r}(\theta_n - \theta_c)}$.

3.4.1 Example 1: AR(1) observed with noise

Consider the following linear-Gaussian state space model,

$$\begin{aligned} X_t &= \theta X_{t-1} + V_t,\ V_t \sim N(0, 0.7^2) \\ Y_t &= X_t + W_t,\ W_t \sim N(0,1), \end{aligned}$$

where V_t and W_t are independent. Observations with length 200 are generated using $\theta_c = 0.7$.

Trajectories of paramter updates given by the three methods are compared in Figure 3.4.1. Results of two experiment runs are reported. It can be seen that both new algorithms are more stable and converge faster than OSA. Updates of both new algorithms converge to the true parameter value after several hundred iterations, while OSA needs over ten thousand iterations to converge, which is not reported in the plots. ASA-SMCw converges a bit faster than SA-SMCw in one run, and almost the same in the other run.

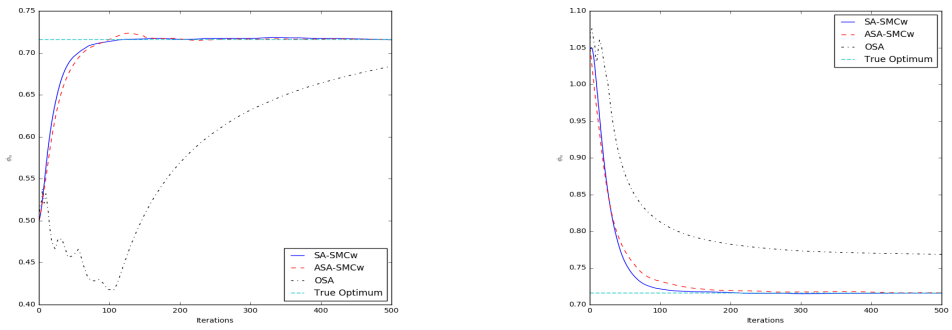


Figure 3.4.1: Trajectories of paramter updates using $N = 500$ in SMC.

Their average performance in estimation accuracy and computational cost are compared using

100 experiment runs. The accurarcy is compared by RMSEs, reported in Figure 3.4.2. The computational cost are compared by the runnning time under the same SMC particles size and number of iterations, reported in Table 3.4.1. It can be seen that both new algortihms perform similarly, and are more accurate than the OSA in all iterations and with all choices of particle sizes. Meanwhile, the running time of both new algorithms is siginicantly less than the OSA in all settings, with reduction of approximately 50%. This is because OSA uses $2p$ SMC runs in each iteration while the new algorithms use only 1 run. The running time of ASA-SMCw is approximately 5% of that of OSA, 10% of SA-SMCw, since the latter simulates particles in every iteration while the former reduces the frequence of simulation. In this example the simulation takes significantly larger computational cost than weight adjustment. Therefore, with $K = 10$, the SA-SMCw takes 10 times runtime with same total iteration.

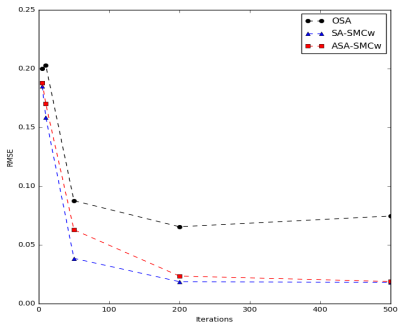


Figure 3.4.2: Trajectories of RMSE when each SMC run uses $N = 500$.

	OSA	SA-SMCw	ASA-SMCw
$m = 100, I = 50$	0.8180	0.3809	0.0429
$m = 1000, I = 50$	3.8645	1.7514	0.1855
$m = 100, I = 500$	8.2431	3.9567	0.4508
$m = 4000, I = 50$	15.3994	7.5079	0.8230

Table 3.4.1: Comparison of computational cost, measured by CPU time(s)

We then study the impact of inner iteration numbers K in ASA-SMCw. We pick $K \in \{1, 5, 10, 20, \infty\}$, where ASA-SMCw becomes SA-SMCw when $K = 1$. A total of 100 simulations are generated

and ASA-SMCw algorithms with different K's are conducted under the same a_n and c_n. Figure

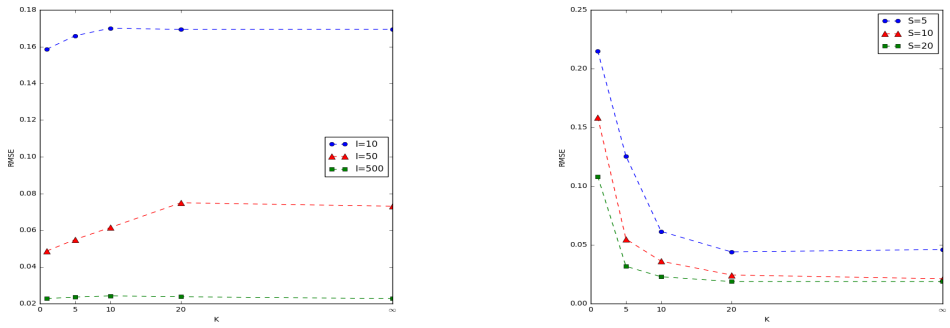


Figure 3.4.3: Left: Trajectories of RMSE when total iterations are set equal. Right: Trajectories of RMSE when total MC samples are set equal.

3.4.3 shows the comparison of RMSE's in two different ways. On the left, we compare the RMSE's for each K when total iterations are set equal. For example, when $I = 10$ and $K = 5$, then two set MC samples are generated. One can observe a small increase of RMSE with as K increases when $I = 50$. This is due to lack of MC samples. On the right, we compare the the RMSE's for each K when total MC samples are set equal. For example, when $S = 5$ and $K = 5$, then we have run a total of 25 iterations. Since the computation of likelihood adjustment is significantly cheaper than Monte Carlo simulation, this comparison can be treated as a comparison with almost same computational cost. This also illustrates the benefit of ASA-SMCw–it is a faster as well a efficient algorithm.

3.4.2 Example 2: Conditinal AR(1) plus noise model

Consider the following CDLM,

$$\begin{aligned} X_t &= \theta_{\lambda_t} X_{t-1} + V_t,\ V_t \sim N(0, 0.7^2), \\ Y_t &= X_t + W_t,\ W_t \sim N(0, 1), \end{aligned}$$

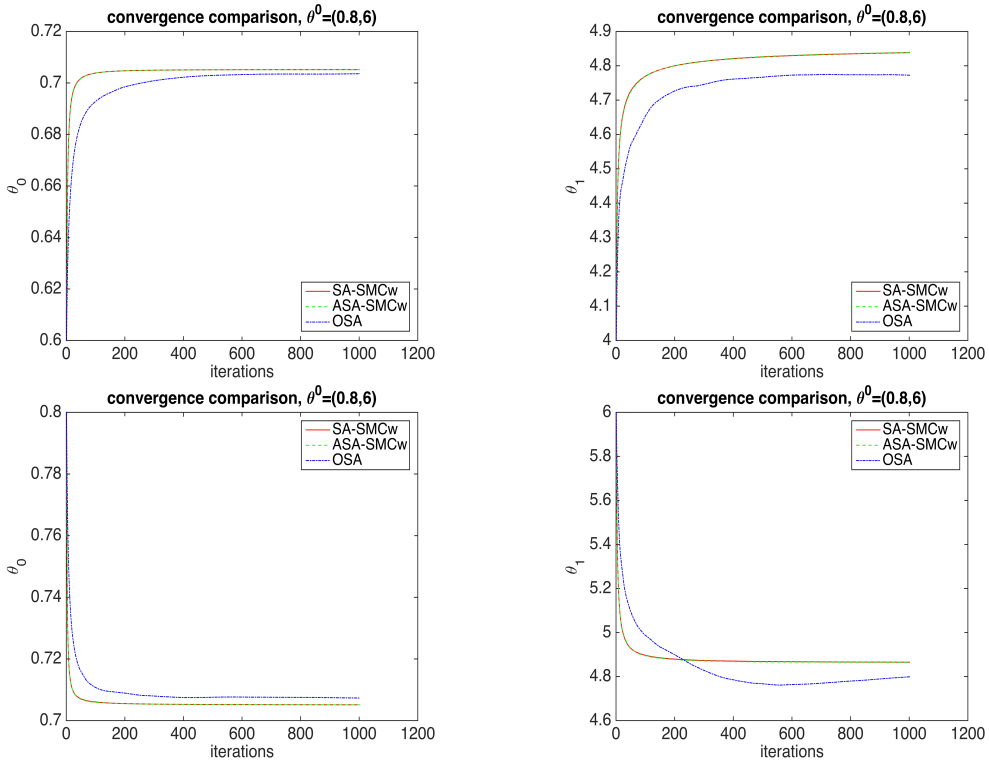


Figure 3.4.4: Comparison of three algorithms, under same tuning parameters. $m = 100$. $K = 10$ in ASA-SMCw. $a_n = \frac{c}{n+5}$, $c_n = 1/n^{1/3}$.

where V_t and W_t are independent and λ_t follows a Bernoulli distribution with parameter 0.1. When conditioning on λ_t, this becomes the model in Example 1. Obervations with length 200 are generated using $\theta_0 = 0.7$ and $\theta_1 = 5$. When $\lambda_t = 1$, the state equation is non-stationary since $\theta_1 > 1$.

Performane of the three algorithms are compared in the similar ways as in Example 1. Trajectories of paramter updates in two experiment runs are given in Figure 3.4.4. It can be seen that the new algorithms improves OSA in two aspects: convergence speed and bias. First, the performance of OSA are more stable than those in Example 1, since MKF is in general more efficient than SMC and particle size 100 is a descent size for MKF. Despite the improvement of OSA, new algorithms converge faster than OSA for both θ_0 and θ_1 due to the further improvement of efficiency of gradient estimate. Second, OSA shows more bias in the approximated MLE of θ_1 than that of θ_0. This is because the state $\lambda_t = 1$ has less data points, on average 20 out of 200 Y can be used to

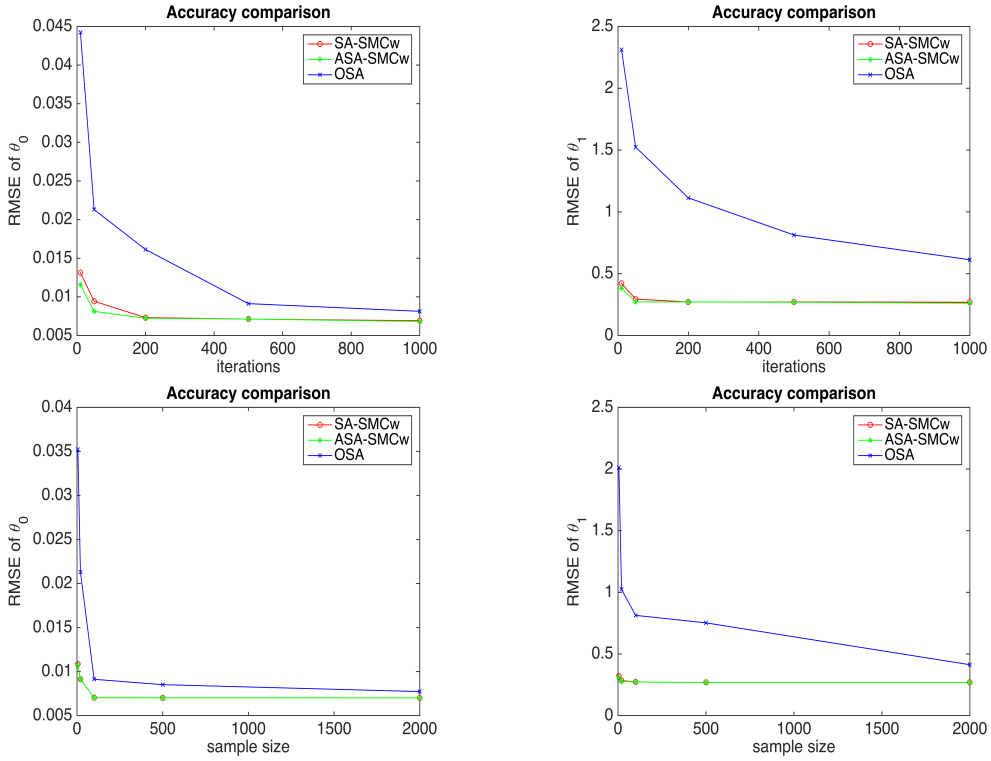


Figure 3.4.5: RMSE comparison based on 100 simulations. In the top two figures we set $m = 100$ and let total iterations change while the total iteration $I = 500$ and size m varies in the bottom two figures.

estiamte $l(\theta_1)$ with $p = 0.1$, and the bias of MLE of θ_1 is higher than that of θ_0. Note that all three algorithms contain the same amount of MLE bias. The new algorithms reduce the bias through reducing the Monte Carlo bias, although they are designed for reducing the Monte Carlo variance.

Similar comparison can be seen in the average performance of 100 experiment runs, reported in Figure 3.4.5. As iteration proceeds, new algorithms converge faster than OSA in RMSE. As particle size increases, RMSE of OSA gets closer to that of new algorithms since the Monte Carlo bias gets closer.

3.4.3 Example 3: Regime-Switching Nelson-Siegel term structure model

In this section we propose a multi-dimensional conditional dynamic linear system to model the yield curve dynamics. The yield curve at certain time t gives interest rates across different contract lengths, or maturities. Diebold and Li (2006) proposes the following state space model with latent vector-AR(1) dynamics for the yield curve,

$$i_{t(\tau)} = L_t + S_t\Big(\frac{1-e^{-\lambda\tau}}{\lambda\tau}\Big) + C_t\Big(\frac{1-e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau}\Big) + \epsilon_{t(\tau)}, \tag{3.6a}$$

$$\begin{bmatrix} L_t \\ S_t \\ C_t \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} + \begin{bmatrix} \phi_1 & 0 & 0 \\ 0 & \phi_2 & 0 \\ 0 & 0 & \phi_3 \end{bmatrix} \begin{bmatrix} L_{t-1} \\ S_{t-1} \\ C_{t-1} \end{bmatrix} + \begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \end{bmatrix} \begin{bmatrix} \eta_{1t} \\ \eta_{2t} \\ \eta_{3t} \end{bmatrix}, \tag{3.6b}$$

where $\epsilon_{t(\tau)}$ is a Gaussian noise with mean 0 and variance σ_ϵ^2, $\eta_{1t},\ \eta_{2t},\ \eta_{3t}$ are independant standard Gaussian noises and $i_{t(\tau)}$ is the zero-coupon yield for maturity τ at time t. Here the 9-dimensional vector $\{i_{t(\tau)}\}_\tau$ is an observable time series, and L_t, S_t, C_t, the so called level, slope and curvature parameters, are latent states. This is known as the dynamic Nelson-Siegel model. This model is Gaussian and linear, thus can be estimated by standard Kalman Filtering procedure. Recent researches (see Ang and Bekaert, 2002; Xiang and Zhu, 2013) show that it would be more reasonable to take into account the change of economic environment, or the regime for more accurate filtering and prediction.

From a statistical perspective, the incorporation of underlying dynamics of level, slope, curvature factors balances the mean-variance trade-off in hope that the sacrifice in real-time Nelson-Siegel fitting helps to stabilize the system and thus improve the prediction performance. Yet it is obvious that a simple linear and Gaussian pattern is unrealistic. With the incorporation of Markov regime-switching in (3.6), the conditional dynamic linear model further balances the goal of real-time fitting and prediction. With a set of properly estimated parameters, Mixture Kalman Filter can conduct the filtering and prediction task in an efficient way. Moreover, with the appropriate likelihood estimation, we can do model selection using the AIC and BIC criterion.

Here consider the situation of two different regimes, i.e.

$$\begin{bmatrix} L_t \\ S_t \\ C_t \end{bmatrix} = \begin{bmatrix} \mu_1^s \\ \mu_2^s \\ \mu_3^s \end{bmatrix} + \begin{bmatrix} \phi_1 & 0 & 0 \\ 0 & \phi_2 & 0 \\ 0 & 0 & \phi_3 \end{bmatrix} \begin{bmatrix} L_{t-1} \\ S_{t-1} \\ C_{t-1} \end{bmatrix} + \begin{bmatrix} w_1^s & 0 & 0 \\ 0 & w_2^s & 0 \\ 0 & 0 & w_3^s \end{bmatrix} \begin{bmatrix} \eta_{1t} \\ \eta_{2t} \\ \eta_{3t} \end{bmatrix},$$

where $s = H, L$ indicates the regime change. When $s = H$ we have a higher drift and volatility term, indicating the regime of higher interest rate. Here we assume that s follows a Markov process with transition probability

$$P = \begin{bmatrix} p_1 & 1 - p_1 \\ 1 - p_2 & p_2 \end{bmatrix}.$$

This is a model with 17-dimensional unknown parameter and used to illustrate our proposed modelling and estimation strategy. Similar strategy can be applied to more complicated regime-switching Nelson-Siegel models, e.g. incorporating the interaction between level, slope and curvature factors and containing more regimes.

3.4.3.1 Simulation study

To begin with, observed yield curve simulated under the parameter specification in Table 3.4.2 and with maturities $\tau \in \Gamma = \{1, 3, 6, 12, 24, 36, 60, 84, 120\}$ are studied. Consider the observed curve with a time length of 200, and denote it by $\boldsymbol{i}_{1:200}$.

	μ_1	μ_2	μ_3	w_1	w_2	w_3
L	0.1969	-0.0651	-0.0158	0.2569	0.2214	0.6686
H	0.4924	-0.1628	-0.0396	0.4382	0.498	1.3387
	λ	p_1	p_2	ϕ_1	ϕ_2	ϕ_3
	0.039	0.922	0.934	0.957	0.969	0.901

Table 3.4.2: Parameter specification

For simplicity we set $\lambda,\ p_1,\ p_2$ and the observation volatilities σ_ϵ as fixed and try to estimate

the rest of the parameters in Table 3.4.2. For a single realisation $\boldsymbol{i}_{1:200}$, a comparison of independent and smoothing SMC likelihood estimations is given in Figure 3.4.6. First it can be seen that the SMC likelihood contains large Monte Carlo variation and a lot of local maximums, even with a relatively large sample size. Especially for the variance parameters w_1–w_3, the likelihood functions are relatively flat and the maximisers are unstable due to the Monte Carlo noise. In contrast, the weight adjusted SMC likelihood does not have local maximums due to its smoothness, and when using θ_c as the simulator parameter, it is fairly close to the 'smoothed' SMC likelihood and hence gives similar and more stable maximiser than the latter. Second, it can be seen that the weight adjusted SMC likelihood around simulator parameters are accurate. Hence if we start from the triangular point which is far away from θ_c, the SA update will lead us towards the right direction. When the simulator parameter gets close to θ_c, the new likelihood estimation would become more reliable. Third, Figure 3.4.6 also shows that it is necessary to avoid using the maximizer in Algorithm 4 due to the inaccuracy of the likelihood estimates beyond the close neighborhood of the simulator parameter.

For the same realisation as above, trajectories of parameter updates using the three algorithms are compared in Figure 3.4.7. It can be seen that the OSA algorithm fails in estimating most parameters. One important reason is that once the AR parameters ϕ_1–ϕ_3 got larger than 1, as ϕ_2 estimated by OSA here, the model using these parameters losed stationarity. Then generated particles would be very different from the true states, and cripple the likelihood estimation and further parameter updates in SA. In this example, this often happens in other experiment runs due to the instability of OSA. It can also be seen that ASA-SMCw converges faster than SA-SMCw, hence reducing the computational cost.

Again all algorithms are compared using 100 experiment runs. Figure 3.4.8 shows the boxplot of the 100 approximate MLEs. Clearly the new algorithms estimate the true parameters with much less variance and bias than OSA.

3.4.3.2 Real data nalysis

Here we study the same dataset as that in Xiang and Zhu (2013) which is the zero-coupon yield curve from 1983:01 to 2010:08 with maturities $\tau = \{1, 3, 6, 12, 24, 36, 60, 84, 120\}$ months. Figure 3.4.9 shows the yield surface.

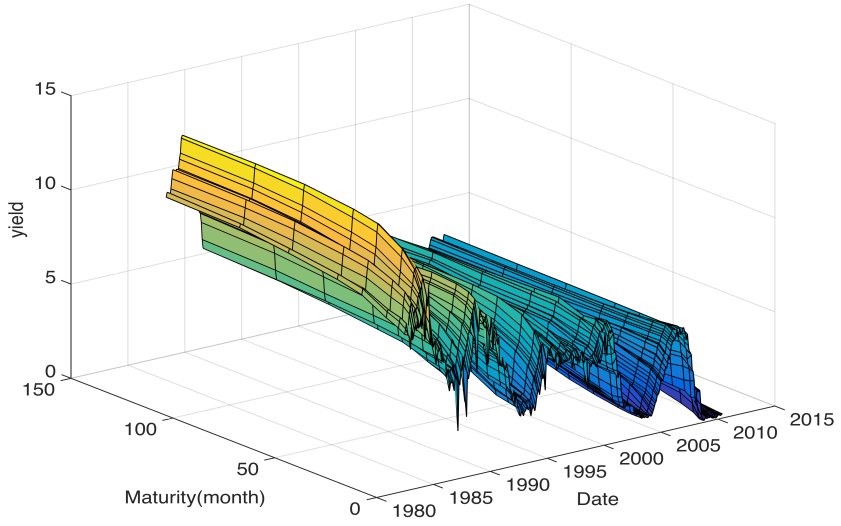


Figure 3.4.9: Yield surface from 1983:01 to 2010:08.

We compare three methods to fit the yield curve which are stated below.

Single-regime LS: In the single-regime model (3.6), The naive method is to treat Equation (3.6a) and (3.6b) separately. Given observations $\{i_{t(\tau)}\}_{t,\tau}$, first estimate $\hat{L}_t, \hat{S}_t, \hat{C}_t$, λ and σ_ϵ in (3.6a) by minimising the squared error loss, then estimate the parameter in (3.6b) by plugging the estimated series of $\hat{L}_t, \hat{S}_t$ and $\hat{C}_t$ and weighted least square.

Single-regime KF: The second method is to apply SA recursion to maximise the likelihood of (3.6) obtained by Kalman filter, then obtain filtered states by Kalman filter. The SA recursion is initialised at the estimated parameter value given by single-regime-LS method. Here parameters λ is fixed at the estimated values from single-regime-LS method due to

the model's insensitivity to λ, and σ_ϵ is also fixed in the similar way. The parameter value estimated by this method is given in the following table:

	L	S	C
$\hat{\mu}$	0.0385	-0.0810	-0.0763
$\hat{\phi}$	0.9909	0.9648	0.9136
$\hat{w}$	0.2870	0.3858	1.1036

This is the method used in Diebold and Li (2006).

Two-regime: The third method uses the regime-switching structure with two regimes, i.e. state model (3.7), and estimate the parameter by the proposed ASA-SMCw algorithm with particles generated by MKF, as done in the simulation study. Then the filtered states are obtained by MKF. The fitted parameter is listed in the following table.

	μ_1	μ_2	μ_3	w_1	w_2	w_3
H	0.2346	0.1636	0.7466	0.3591	0.3144	1.2078
L	-0.0837	-0.4378	-0.9348	0.2024	0.2917	0.9936
	ϕ_1	ϕ_2	ϕ_3		p_1	p_2
	0.9962	0.9506	0.9217		0.8716	0.9278

Table 3.4.3: Fitted Parameters of two-regime method

The comparison of three parameter estimation algorithms in this real data analysis is similar to that in the simulation study, thus not reported here.

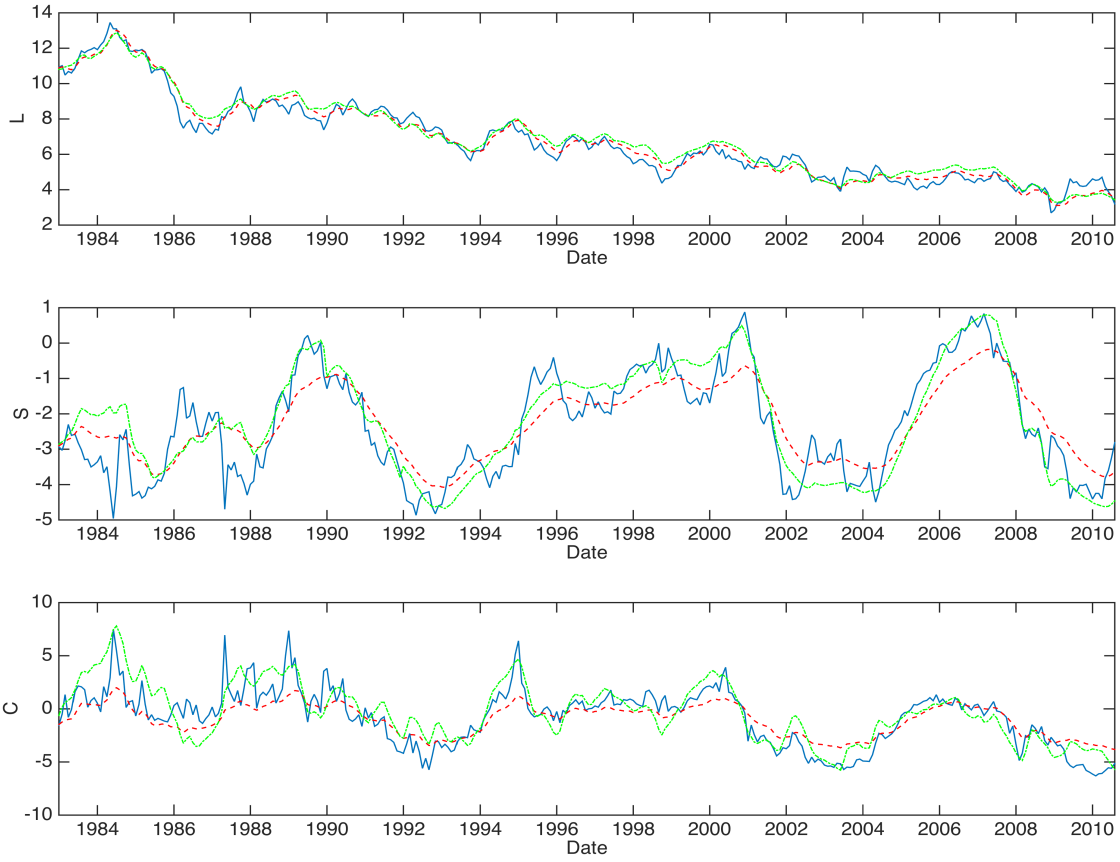


Figure 3.4.10: Comparison of filtered L_t, S_t and C_t using single-regime LS(blue and solid), single-regime KF(red and dashed) and two-regime (green and dash-dotted).

The filtered L_t, S_t and C_t using these methods are presented in Figure 3.4.10. For the single-regime model fitted with least square, filtered states are very noisy, since the estimation does not consider the underlying state dynamics and at each t only 9 observations are used to estimate the latent states. For the single-regime model fitted with Kalman filter, although it can capture the rough patterns of the three states, it fails to react to the 'peaks' timely. In contrast, the well estimated two-regime model performs better in balancing the smoothness and accuracy, or in other

words, the bias-variance tradeoff.

We compare the single-regime and two-regime models using AIC and BIC, which are popular criterion for model selection, with the estimated parameters, and the comparison in given below.

	Single-regime	Two-regimes
AIC	4913.9	4834.2
BIC	4967.9	4936.2

It is clear that the regime-switching model has a significant improvement in both AIC and BIC.

Secondly, to test the forecasting performance, we set the yield curve from 2005:09 to 2010:08(every 4 months in order to reduce computational cost) as the training set. We fit the KF and MKF parameters up to the time before forecasting and then use the fitted parameters. Table 3.4.4 shows the results: Overall those forecasting models cannot prove efficiency than random walk. But within

	3 months	6months	1 year	3 years	5 years	10 years
1-month-ahead forecasting RMSE						
RW	0.259	0.234	0.233	0.255	0.254	0.240
Single-Regime	0.262	0.235	0.234	0.256	**0.254**	**0.238**
Two-Regimes	**0.247**	**0.230**	**0.225**	**0.248**	0.255	0.239
6-months-ahead forecasting RMSE						
RW	0.999	0.990	0.950	**0.851**	0.771	0.621
Single-Regime	1.010	0.998	0.947	0.852	**0.768**	**0.620**
Two-Regimes	**0.957**	**0.963**	**0.932**	0.879	0.788	0.640
12-months-ahead forecasting RMSE						
RW	1.818	1.758	1.623	**1.218**	0.981	0.689
Single-Regime	1.821	1.764	1.626	1.231	**0.978**	**0.672**
Two-Regimes	**1.733**	**1.692**	**1.544**	1.370	1.165	0.873

Table 3.4.4: Forecasting comparison measured by RMSE. The bold ones are the best in each column

a shorter maturities and nearer forecasting window, the regime switching model has a better prediction power.

3.5 Conclusions

In this chapter, we have proposed the SA-SMCw and ASA-SMCw algorithms for parameter estimation in state space models. Both methods have very simple requirement while achieve stable and efficient results. All of our examples show that SA-SMCw and ASA-SMCw have significant gains over ordinary Stochastic-Approximation-based algorithms in both convergence rate and computational efficiency. In general, ASA-SMCw is preferable than SA-SMCw for its reduction in computational cost, while its performance is more sensitive to the smoothing estimation and thus needs special care.

Furthermore, since SA-SMCw and ASA-SMCw relies on a Stochastic Approximation scheme, the performance of them can be further enhanced by existing Stochastic Approximation techniques like adaptive tuning parameter selection. The development in this paper is thus an example to show that Stochastic Approximation methods is powerful in parameter estimation in state space models.

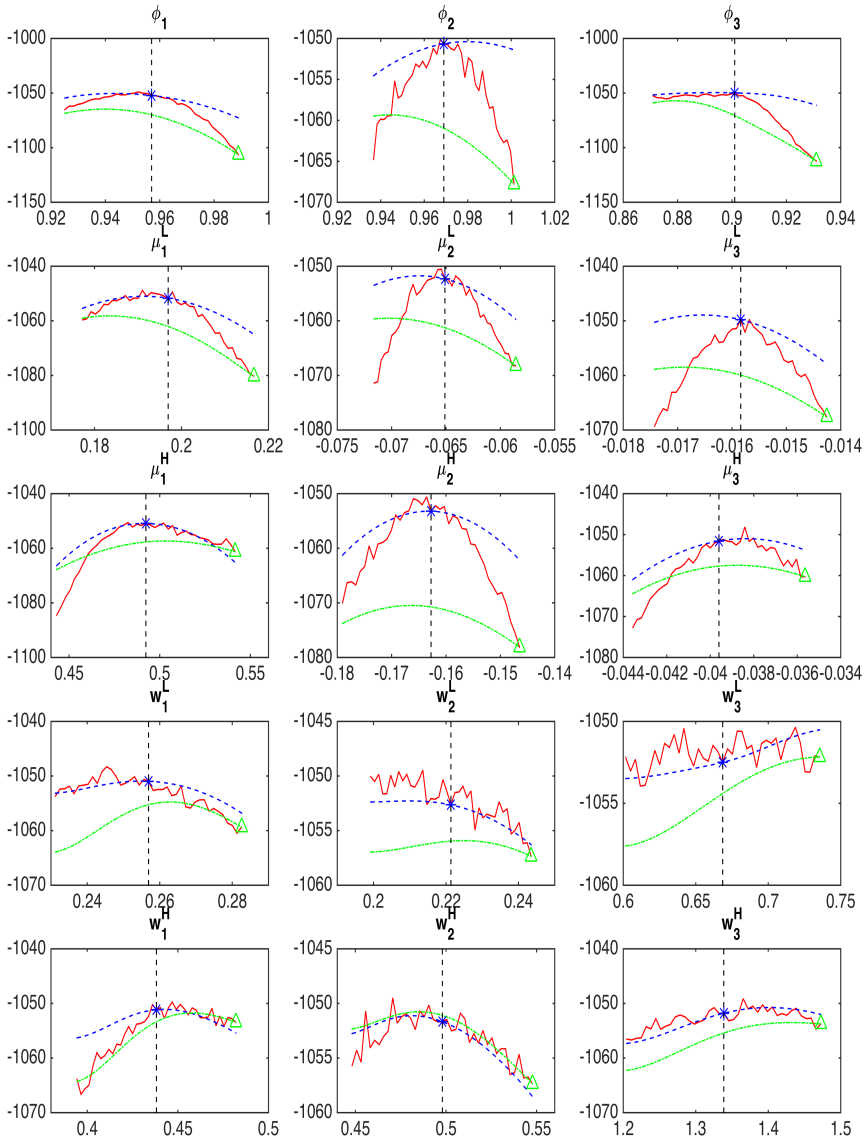


Figure 3.4.6: Likelihood estimates using independent and smoothing estimating. The solid line represents the independent estimation. The blue dashed line is the smoothing estimation starting from the cross mark. The dash-dot line is the smoothing estimation starting from the faraway triangular mark. $m = 500$

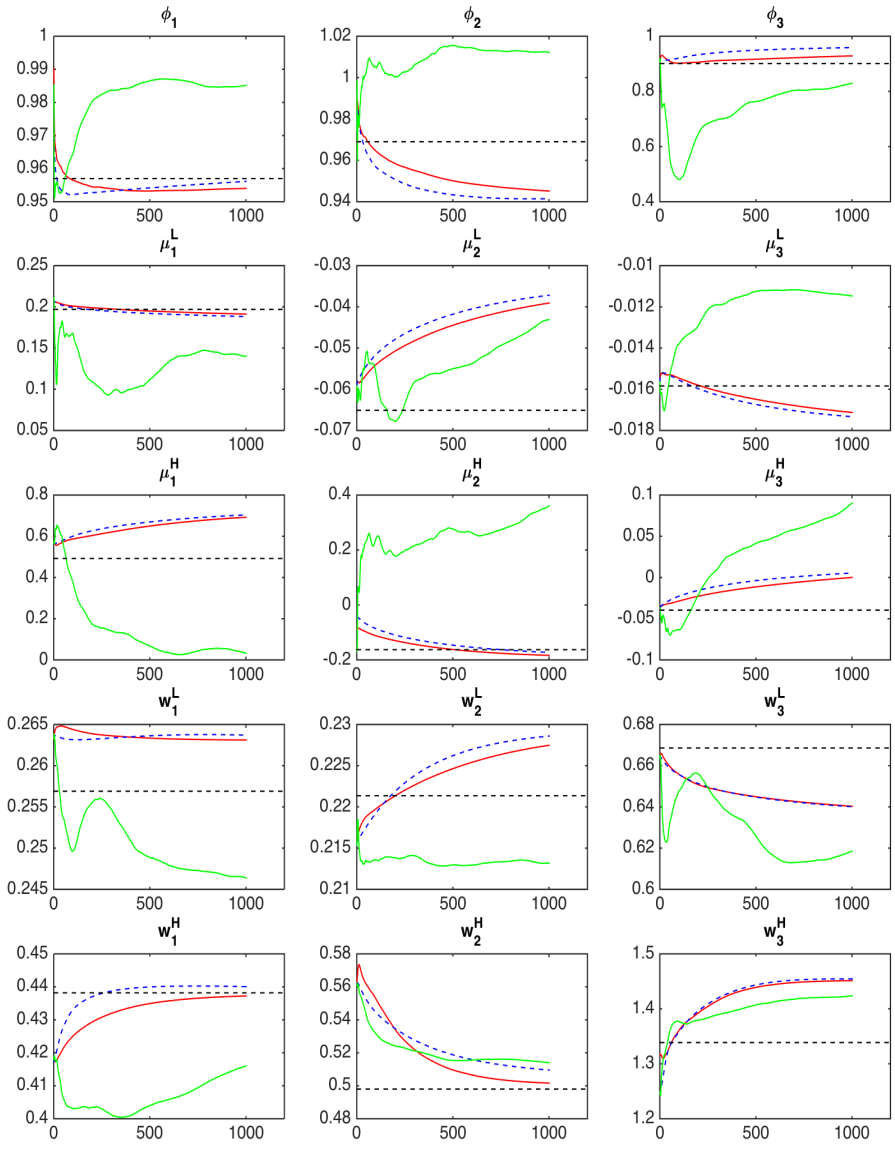


Figure 3.4.7: Convergence comparison of the three algorithms for 1000 iterations. Red solid line is for SA-SMCw algorithm while blue dashed line is for ASA-SMCw algorithm. Green solid line(rigid) is for OSA algorithm. Here Monte Carlo sample size $m = 100$. $K = 10$ in ASA-SMCw. Dashed horizontal line is the true parameter.

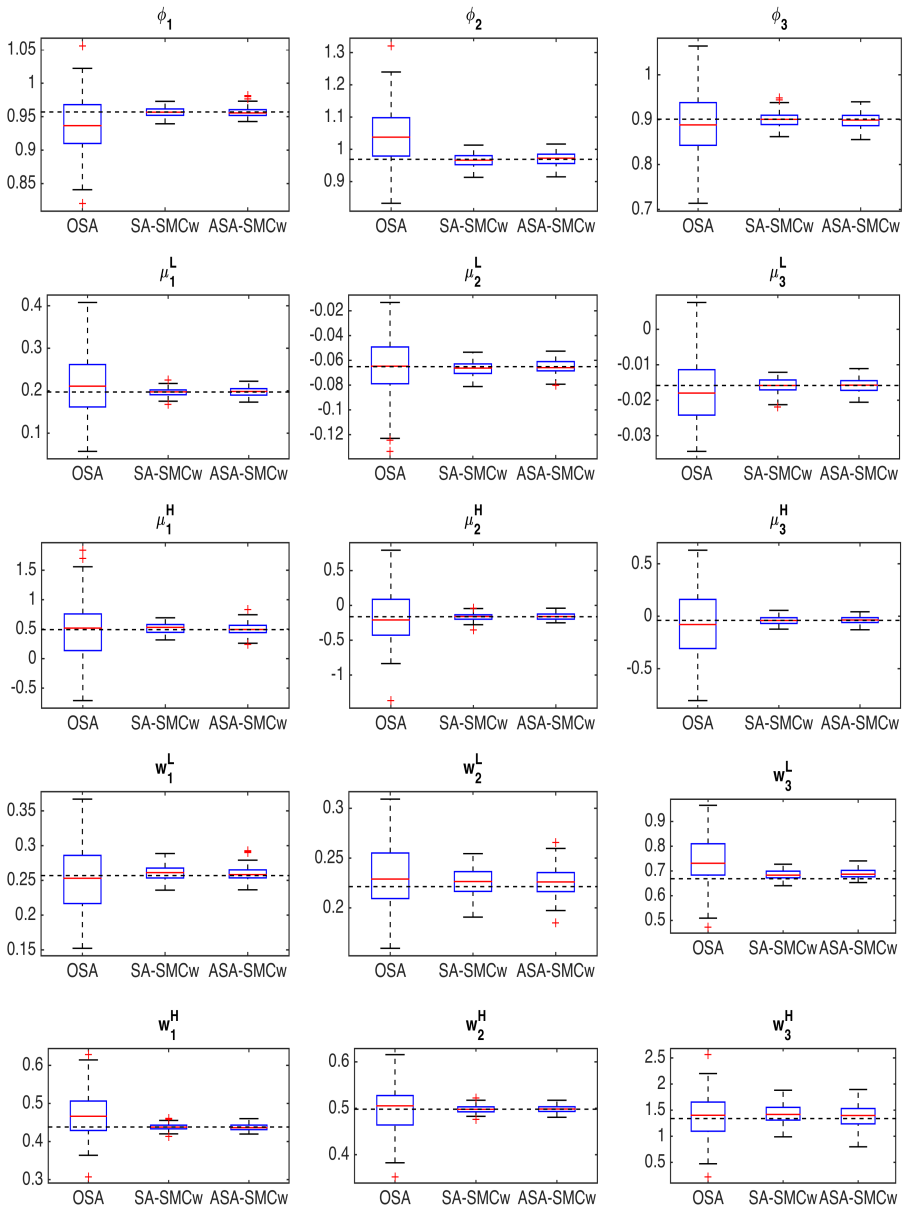


Figure 3.4.8: Boxplot of 100 estimations using the three algorithms. Each estimation is based on 1000 iterations and Monte Carlo sample size $m = 100$. $K = 10$ in ASA-SMCw. Dashed horizontal line is the true parameter.

Chapter 4

Estimating Periodically Collapsing Rational Bubble with Mixture Kalman Filter

4.1 Introduction

This chapter presents a joint work with Hao Chang on a regime-switching financial bubble model.

Financial bubbles have long intrigued economists and gained growing interests. As many economists believe that stock prices are too volatile to be attributed to market fundamentals, connecting the unexplained term to bubbles has become one practical and reasonable way of modeling (Campbell and Shiller, 1988; LeRoy and Porter, 1981; Wu, 1997). Bubbles refer to asset prices that exceed an asset's fundamental value because the holders believe that they can resell the asset at an even higher price in the future. Bubbles are typically associated with dramatic price increases followed by a collapse (Brunnermeier, 2009).

So far, a variety of literature has discussed the econometric detection of rational bubbles, which exhibit explosive behavior if investors have rational expectations and identical information. One strand utilizes co-integration and unit-root tests to examine whether asset prices are more explosive than dividends (Diba and Grossman, 1988; Hamilton and Whiteman, 1985). Evans (1991) criticizes this approach by arguing that it fails to detect periodically collapsing bubbles. This problem is overcome by Phillips et al. (2011), who propose a test procedure involving the recursive implementation of a right-side unit root test. Another strand, which this chapter follows, is to directly estimate the bubble as an unobservable state vector in a state space model. Specifying the bubble

dynamics by an AR(1) process, Wu (1995, 1997) identifies bubbles using the Kalman filtering technique. However, the linear specification here implies a continuous growth, a pattern inconsistent with inherent non-linear characteristics of real-world data. This non-linearity is believed to origin from the periodical switching of the underlying economic condition, also denoted as regime.

In line with Wu (1995, 1997) and the regime-switching idea, we adopt a conditional dynamic linear system (CDLM) (Harrison and West, 1999) to model the bubble process. We allow two to three regimes that switch by Markovian transition probability matrices while keeping the system conditionally linear and Gaussian given the regime. With a two-regime model, we specify the surviving and collapsing regimes, in which a bubble can grow at a speed greater or smaller than the asset's required rate of return. With a three-regime specification, we define the exploding, surviving and collapsing regimes, in which the bubble can grow at a speed greater than, equal to, or smaller than the asset's required rate of return. We impose the restriction that the expected growth rate of bubble is equal to the asset's required rate of return in order to make the bubble satisfy the rational expectations condition.

Presented as a state space model with regime switching, the asset-bubble system is estimated by a novel Monte Carlo based filtering scheme, Mixture Kalman filter (MKF), proposed by Chen and Liu (2000) and introduced in Section 2.4, in an efficient manner. This methodology has several advantages over the existing Gibbs sampling method (Kim et al., 1999). First, it is not subject to Bayesian bias from the choice of prior distribution of unknown parameters. Second, the MKF is a likelihood-based approach so the associated model selection rules such as AIC and BIC can be implemented to test whether our mixture linear framework yields a better fit than linear models.

We first examine the efficacy of the proposed method by applying it to simulated observations by Evans (1991). Then the method is applied to real stock index of the US stock market. With the associated likelihood-based model selection techniques, the proposed model with regime-switching is proved to yield a better fit in bubble process. In addition, the estimated model provides a filtered probability series of different regimes, which date stamps most of the bubble collapsing periods in history.

The remainder of this chapter is organized as follows. Section 4.2 reviews the classical asset-bubble model with the specification of a linear state space form. Section 4.3 is devoted to the elaboration of the proposed regime-switching model along with the corresponding estimation strategy, the Mixture Kalman filtering technique. Section 4.4 reports the estimation results in both simulated and real data. Section 4.5 concludes.

4.2 Basic bubble model with constant drift

4.2.1 Model specification and notation

In line with Wu (1997), we start from the stocks' fundamental price. Consider the standard linear rational expectations model of stock price determination,

$$[\mathrm{E}_t(P_{t+1} + D_t) - P_t]/P_t = r, \tag{4.1}$$

where:

$P_t =$ the real stock price at time t;

$D_t =$ the real dividend paid at time t;

$\mathrm{E}_t =$ the mathematical expectations conditional on information available at time t; and

$r =$ the required real rate of return, $r > 0$

After transferring the model in natural logarithms term and taking linear approximation (Campbell and Shiller, 1988), equation (4.1) can be written as follows:

$$q = \kappa + \psi \mathrm{E}_t p_{t+1} + (1 - \psi) d_t - p_t, \tag{4.2}$$

where:

$q =$ the required log gross return rate;

ψ is the average ratio of the stock price to the sum of the stock price and the dividend, $0 < \psi < 1$;

$\kappa = -ln(\psi) - (1-\psi)ln(1/\psi - 1);$

$p_t = ln(P_t);$

$d_t = ln(D_t).$

The unique forward-looking, no-bubble solution to (4.2), denoted by p_t^f is given by

$$p_t^f = (\kappa - q)/(1-\psi) + (1-\psi)\sum_{i=0}^{\infty} \psi^i \mathrm{E}_t(d_{t+i}), \tag{4.3}$$

provided that the transversality condition is satisfied,

$$\lim_{t\to\infty} \psi^i \mathrm{E}_t(p_{t+i}) = 0. \tag{4.4}$$

Equation (4.3) is the present value relation which states that the log stock price is equal to the present value of expected future log dividend streams. Notice that if the transversality condition is violated, then (4.3) is only a particular solution to (4.2). Nevertheless, the transversality can hardly hold in real-world stock market. To fill in the gap, a general solution to (4.3) brings in the bubble term which represent the difference between the stock price and its fundamental value:

$$p_t = (\kappa - q)/(1-\psi) + (1-\psi)\sum_{i=0}^{\infty} \psi^i \mathrm{E}_t(d_{t+i}) + b_t = p_t^f + b_t, \tag{4.5}$$

where b_t satisfies the following homogeneous difference equation:

$$\mathrm{E}_t(b_{t+i}) = (1/\psi)^i b_t, \quad \text{for } i = 1, 2, ... \tag{4.6}$$

In equation (4.5), the no-bubble solution p_t^f is exclusively determined by dividends and is often called the market-fundamental solution, while b_t can be driven by events extraneous to the market and is referred to as a rational speculative bubble. The existence of a bubble causes the actual stock price to deviate from its market-fundamental value.

Since the log dividends appear to be non-stationary, Wu (1997) specify the model in its difference form. Taking the first difference of (4.6) yields

$$\Delta p_t = (1-\psi)\sum_{i=0}^{\infty}\psi^i[\mathrm{E}_t(d_{t+i}) - \mathrm{E}_{t-1}(d_{t+i-1})] + \Delta b_t = \Delta p_t^f + \Delta b_t \tag{4.7}$$

To obtain a parsimonious specification, the log dividends can be assumed to follow an ARIMA(h,1,0) process as follows:

$$\Delta d_t = \mu + \sum_{j=1}^{h}\varphi_j \Delta d_{t-j} + \delta_t \tag{4.8}$$

where δ_t is an i.i.d. error term and distributed $N(0,\sigma_\delta^2)$. The autoregressive order h in (4.8) is to be determined by the data.

In companion form, equation (4.8) can be written as

$$Y_t = U + AY_{t-1} + \nu_t, \tag{4.9}$$

where $Y_t = (\Delta d_t, \Delta d_{t-1}, ..., \Delta d_{t-h+1})'$, $U = (\mu, 0, 0, ..., 0)'$, and $\nu_t = (\delta_t, 0, 0, ..., 0)'$ are all h-vector and $A = \begin{pmatrix} \varphi_1 & \varphi_2 & ... & \varphi_{h-1} & \varphi_h \\ 1 & 0 & ... & 0 & 0 \\ 0 & 1 & ... & 0 & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & ... & 1 & 0 \end{pmatrix}$ is an $h \times h$ matrix. Therefore, equation (4.7) becomes

$$\Delta p_t = \Delta d_t + M\Delta Y_t + \Delta b_t, \tag{4.10}$$

where:

$g = (1, 0, 0, ..., 0)$ is an h-row vector; and

$M = gA(I-A)^{-1}[I-(1-\psi)(I-\psi A)^{-1}]$ is an h-row vector and I is an $h \times h$ identity matrix.

When estimating the stock price equation (4.10), we are confronted with the problem that the

bubble component b_t is unobservable. In fact b_t is also the only unobserved series. Instead of directly estimating the bubble using equation (4.10), a more efficient way is to build up a state space model which utilizes both the information from equation (4.10) and the dynamic of bubble series itself.

A linear rational speculative bubble is believed Wu (1997) to have the following parametric dynamic:

$$b_t = (1/\psi)b_{t-1} + \eta_t, \tag{4.11}$$

where the innovation η_t is assumed to be serially uncorrelated and have zero mean and finite variance σ_η^2. It is also assumed that η is uncorrelated with the dividend innovation, δ in equation (4.8).

4.2.2 The state space form

In a companion form, a linear bubble-dividend-stock system follows the following state space model:

$$x_t = Hx_{t-1} + Ww_t, \tag{4.12}$$

$$z_t = Gx_t + Dg_t + Vv_t, \tag{4.13}$$

where

$x_t = (b_t, b_{t-1})'$ is the (2×1) vector of unobserved variable referred to as state variables;

$z_t = (\Delta d_t, \Delta p_t)'$ is the (2×1) vector of observable output variables;

$g_t = (1, \Delta d_t, \Delta d_{t-1}, ..., \Delta d_{t-h})'$ is $((h+1) \times 1)$ vector of observable input variables;

$H = \begin{pmatrix} 1/\psi & 0 \\ 1 & 0 \end{pmatrix}, G = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix}, W = \begin{pmatrix} \sigma_\eta & 0 \\ 0 & 0 \end{pmatrix}, V = \begin{pmatrix} \sigma_\delta & 0 \\ 0 & 0 \end{pmatrix}$

$D = \begin{pmatrix} \mu & 0 & \varphi_1 & \varphi_2 & ... & \varphi_{h-1} & \varphi_h \\ 0 & (1+m_1) & (m_2 - m_1) & (m_3 - m_2) & ... & m_h - m_{h-1} & -m_h \end{pmatrix}$ where m_i is the i-th component of the $(h \times 1)$ vector M. w_t and v_t are i.i.d two dimension standard normal random vector. Since this model only involves one regime, in the following part, it is denoted as the one-regime model or the model R1. Model R1 is both linear and Gaussian, thus can be filtered using

Kalman Filter in section 2.2 with equation (2.8). The only difference here is that the expected μ needs to include the drift Dg_t. Model fitting can also be done by maximizing (2.9).

4.3 New bubble model with periodically collapsing

Non-linear economic models have been more preferable due to its flexibility and accordance to financial data. This is also the case in bubble models. The dynamic of bubble series is periodically changing in their nature (Brunnermeier, 2009). The simplest argument for this goes with the following: a bubble in financial crisis usually collapses and has negative return. Therefore, in equation (4.11), it is reasonable and desirable to consider at least one more situation, in which the AR coefficient is smaller than 1. Moreover, besides a rational bubble which grows with the asset's required rate of return, there are scenarios that bubble grows irrationally with the rate higher than the required rate of return in an over-invested economic situation, and then followed by collapses. This dynamic is depicted in Evans (1991) and believed to be more realistic.

We now introduce two classes of conditional dynamic linear models to describe the stock-dividend-bubble system. Both of the models assume the state equation (4.12) have different parameters under different regimes. The underlying regime λ_t follows a Markov-switching process with transition probability matrix P. The first class assumes two regimes, the surviving ($\lambda_t = 1$) and collapsing regimes ($\lambda_t = 2$), in which a bubble can grow at a speed larger than or less than the asset's required rate of return. While the second class assumes three regimes, the surviving ($\lambda_t = 1$), exploding ($\lambda_t = 2$) and collapsing ($\lambda_t = 3$) regimes, in which the bubble can grow at a speed equal to, larger than, or less than the assets required rate of return. We impose the restriction that the expected growth rate of bubble is equal to the asset's required rate of return in order to make the bubble satisfy the rational expectations condition. Our model specification allows bubbles exhibit irrational behaviors in the short-run, but over the long-run, they still satisfy the rational expectations condition and can be classified as the rational bubbles.

4.3.1 Two-regimes model

In this subsection, we introduce the specification of the first class of models mentioned above. We assume the two regime-switching conditional linear bubble process as follows,

$$b_{t+1} = \begin{cases} a_1 b_t + \eta_t^{(1)}, \eta_t^{(1)} \sim N(0, \sigma_{\eta_1}^2) & \text{if } \lambda_t = 1, \\ a_2 b_t + \eta_t^{(2)}, \eta_t^{(2)} \sim N(0, \sigma_{\eta_2}^2) & \text{if } \lambda_t = 2, \end{cases} \tag{4.14}$$

where the probabilistic nature of the regime-indicator λ_t ($\lambda_t = 1, 2$) are specified by a first-order Markov-process with time-invariant transition probabilities $p_{ij} = Pr[\lambda_t = j | \lambda_{t-1} = i]$ which we collect in the transition-probability matrix

$$P = \begin{pmatrix} p_{11} & 1 - p_{11} \\ 1 - p_{22} & p_{22} \end{pmatrix}. \tag{4.15}$$

The unconditional probabilities for two regimes are

$$\pi_1 = \frac{1 - p_{22}}{2 - p_{11} - p_{22}}, \ \pi_2 = \frac{1 - p_{11}}{2 - p_{11} - p_{22}}. \tag{4.16}$$

To satisfy rational expectations condition (4.6), the constraint we put on the parameters are

$$a_1 \pi_1 + a_2 \pi_2 = 1/\psi, \ a_1 > a_2. \tag{4.17}$$

In two-regime Markov-switching CDLM, the collapsing regime helps to fit the downward shift in the filtered bubble series. In addition, volatilities in different regimes are allowed to be different in the belief that the bubble series may be more volatile in collapsing state. In the subsequent analysis, the two-regimes model is also denoted as the model R2.

4.3.2 Three-regimes nodel

It is more realistic and appealing to consider a three-regimes model to depict the dynamic of the bubble series. In reality, the bubble process starts with a period that most investors are rational and bubble grows with the rational drift term $a_1 = 1/\psi > 1$. Then gradually the bubble grows out of control with a drift term $a_2 > a_1$, which we call explosive drift, until it collapses with a drift term $a_3 < a_1$, i.e.

$$b_{t+1} = \begin{cases} a_1 b_t + \eta_t^{(1)}, \eta_t^{(1)} \sim N(0, \sigma_{\eta_1}^2) & \text{if } \lambda_t = 1, \\ a_2 b_t + \eta_t^{(2)}, \eta_t^{(2)} \sim N(0, \sigma_{\eta_2}^2) & \text{if } \lambda_t = 2, \\ a_3 b_t + \eta_t^{(3)}, \eta_t^{(3)} \sim N(0, \sigma_{\eta_3}^2) & \text{if } \lambda_t = 3, \end{cases} \tag{4.18}$$

where $a_1 = 1/\psi > 1,\ a_2 > a_1 > a_3$.

In a three-regimes model, the construction of transition probability matrices needs careful consideration in the sense that some of the transition should be restricted. For example, right after the financial crisis, the bubble is not likely to explode due to the cautious investment environment. Besides, in a pessimistic but realistic perspective, once in an explosive regime, the bubble can only collapse other than turning rational. Therefore we provide two relatively reasonable specifications of transition probability matrices.

In the first probability setting, a rational regime can only transfer to exploding regime but not collapsing one, i.e.

$$P = \begin{pmatrix} p_{11} & 1 - p_{11} & 0 \\ 0 & p_{22} & 1 - p_{22} \\ 1 - p_{33} & 0 & p_{33} \end{pmatrix}, \tag{4.19}$$

The stationary distribution can be calculated as follows,

$$\pi_1 = \frac{q_{11}^{-1}}{q_{11}^{-1} + q_{22}^{-1} + q_{33}^{-1}},\ \pi_2 = \frac{q_{22}^{-1}}{q_{11}^{-1} + q_{22}^{-1} + q_{33}^{-1}},\ \pi_3 = \frac{q_{33}^{-1}}{q_{11}^{-1} + q_{22}^{-1} + q_{33}^{-1}}, \tag{4.20}$$

where $q_{ii} = 1 - p_{ii},\ i = 1, 2, 3$. By the rational expectations condition

$$a_1\pi_1 + a_2\pi_2 + a_3\pi_3 = 1/\psi = a_1, \tag{4.21}$$

which suggests the following parameter constraint,

$$q_{22}(a_1 - a_3) = q_{33}(a_2 - a_1). \tag{4.22}$$

In the subsequent analysis, this model is also denoted as the first version of three-regimes model or the model R3V1.

The second probability setting allows the transition from rational regime directly to collapsing regime, thus

$$P = \begin{pmatrix} p_{11} & p_{12} & 1 - p_{11} - p_{12} \\ 0 & p_{22} & 1 - p_{22} \\ 1 - p_{33} & 0 & p_{33} \end{pmatrix}, \tag{4.23}$$

in which case the stationary distribution can be calculated as follows,

$$\pi_1 = \frac{q_{22}q_{33}}{q_{22}q_{33} + q_{11}q_{22} + p_{12}q_{33}},\ \pi_2 = \frac{p_{12}q_{33}}{q_{22}q_{33} + q_{11}q_{22} + p_{12}q_{33}},\ \pi_3 = \frac{q_{11}q_{22}}{q_{22}q_{33} + q_{11}q_{22} + p_{12}q_{33}}, \tag{4.24}$$

where $q_{ij} = 1 - p_{ij}$. With the same condition as equation (4.21) we can derive the following parameter constraint

$$q_{11}q_{22}(a_1 - a_3) = p_{12}q_{33}(a_2 - a_1). \tag{4.25}$$

In the subsequent analysis, this model is also denoted as the second version of three-regimes model or the model R3V2.

4.3.3 State space form with regime switching

The periodically collapsing bubble-dividend-stock system can be expressed as the following state space model with regime switching:

$$x_t = H_{\lambda_t} x_{t-1} + W_{\lambda_t} w_t, \tag{4.26}$$

$$z_t = G x_t + D g_t + V v_t, \tag{4.27}$$

where

$x_t = (b_t, b_{t-1})'$ is the (2×1) vector of unobserved variable referred to as state variables;

$z_t = (\Delta d_t, \Delta p_t)'$ is the (2×1) vector of observable output variables;

$g_t = (1, \Delta d_t, \Delta d_{t-1}, ..., \Delta d_{t-2})'$ is $((h+1) \times 1)$ vector of observable input variables;

$$H = \begin{pmatrix} a_{\lambda_t} & 0 \\ 1 & 0 \end{pmatrix}, G = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix}, W = \begin{pmatrix} \sigma_{\eta_{\lambda_t}} & 0 \\ 0 & 0 \end{pmatrix}, V = \begin{pmatrix} \sigma_\delta & 0 \\ 0 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} \mu & 0 & \varphi_1 & \varphi_2 & ... & \varphi_{h-1} & \varphi_h \\ 0 & (1+m_1) & (m_2 - m_1) & (m_3 - m_2) & ... & m_h - m_{h-1} & -m_h \end{pmatrix}$$

System (4.26) and (4.27) forms a conditional dynamic linear model. The inference of this CDLM, including estimating and predicting the x_t series as well as the λ_t series needs to be carried out by Mixture Kalman Filter introduced in Section 2.4. The more complicated while equivalently important parameter estimation mission needs to be conducted under the framework of Chapter 3.

4.4 Empirical analysis

In this section we apply both the one-regime model with the Kalman Filtering technique and our proposed multiple-regimes models with the Mixture Kalman Filtering scheme to both artificial data and real data. In order to obtain artificial data, we simulate the stock price, dividend, and bubble observations following the processes described in Evans (1991). For the real data, we mainly focus on the US stock market by studying its most representative market index — S&P-500 stock index

with quarterly frequency through the whole history.

4.4.1 Artificial data

We follow Evans (1991) to simulate periodically collapsing bubble series. The artificial bubbles have the form,

$$B_{t+1} = \begin{cases} (1+r)B_t u_{t+1}, & \text{if } \beta_t < \alpha, \\ \left[\delta + \frac{1+r}{\pi}(B_t - \frac{\delta}{1+r})\xi_{t+1}\right]u_{t+1} & \text{if } \beta_t > \alpha, \end{cases} \tag{4.28}$$

where δ and α are real scalars such that $0 < \delta < (1+r)\alpha$. $\{u_t\}$ is a sequence of non-negative exogenous i.i.d lognormal variables with $E_t(u_{t+1}) = 1$. Here we assume $\{u_t\}$ to be i.i.d. lognormally distributed and scaled to have unit mean, i.e., $u_t = \exp(y_t - \frac{\tau^2}{2})$ with $\{y_t\}$ being i.i.d. $N(0, \tau^2)$. $\{\xi_t\}$ is an exogenous i.i.d Bernoulli process independent of $\{u_t\}$ with $Pr(\xi_t = 0) = 1 - \pi$ and $Pr(\xi_t = 1) = \pi$ for $0 < \pi < 1$. The data-generating process for the dividends follows a pure random walk, $D_t = D_{t-1} + \epsilon_t$, where $\{\epsilon_t\}$ is a Gaussian white-noise process with mean zero and variance σ_ϵ^2. Therefore, the fundamental stock price is $P_{Dt} = \frac{D_t}{r}$. Hence the stock price is $P_t = P_{Dt} + B_t$. The parameters for simulation is listed in table 4.4.1,

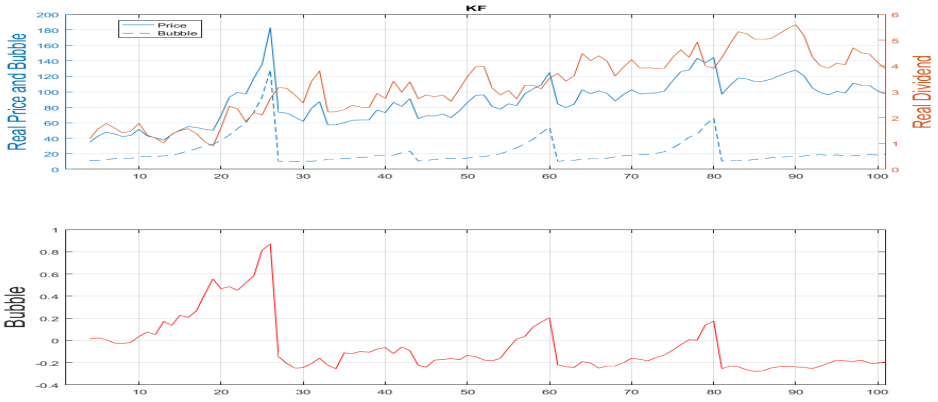
The upper panel in Figure 4.4.1 presents the realization of one trial of simulation using the Evans model with parameters in Table 4.4.1. The simulated stock price, dividend as well as the

Table 4.4.1: Parameter Specification for the Evans Bubble Process

No. of obs.	α	τ^2	r	δ	D_0	B_0	σ_ϵ^2	π
100	1	0.0025	0.05	0.5	1.3	0.5	0.1574	0.85

bubble series are plotted. One feature of the bubble series, which also aligns the equation (4.28), is the relatively sudden and strong collapse. The Evans process approximates the reality by creating a threshold, under which the bubble is rationally growing. However once the bubble grows out of the rational threshold, it starts to explode faster than the rational growing speed, and accompany with a probability to collapse to a certain low level, after which the bubble continues to behave rationally again.

Figure 4.4.1: Evans-one regime



One can either use the two-regimes model or the three-regimes model to filter out both the bubble and probability series. The first step is the model fitting. We adopt maximum likelihood estimation utilizing Mixture Kalman Filter and equation (2.9) to estimate unknown parameters. A by-product of this step, also a benefit from MKF, is the log-likelihood, which can be used to compute several criteria of model selection. To be more specific, Akaike information criterion (Akaike, 1998), also known as AIC can be calculated by

$$\text{BIC} = 2k - 2\log(\hat{L}), \tag{4.29}$$

where k is the number of parameters. AIC can be directly applied as a criterion as the goodness of fit of a model to the data. The smaller the AIC, the better the model fits the data. Another tool of model evaluation is the Bayesian Information Criterion (BIC), which is defined as

$$\text{AIC} = \log(n)k - 2\log(\hat{L}), \tag{4.30}$$

where n is the number of data points. Once the model is fitted, the series of bubble and probability of each stage can be filtered by Mixture Kalman Filter.

As we have discussed before, there are two versions of the three regime models. The first

version (R3V1) has the transition probability matrix specified by the equation (4.19), while the second version (R3V2) is equipped with the transition probability matrix specified by the equation (4.23). The estimated parameters and the associated log likelihood, AIC and BIC for all models are reported in Table 4.4.2.

Table 4.4.2: Estimation summary: Evans process

	1 Regime			2 Regimes			3 Regimes - V1			3 Regimes - V2		
Model Evaluation	loglike	108.4176		loglike	202.9590		loglike	205.0493		loglike	205.1933	
	AIC	-204.8351		AIC	-387.9181		AIC	-390.0986		AIC	-388.0986	
	BIC	-189.5119		BIC	-364.9332		BIC	-390.1863		BIC	-388.1951	
Estimated Paras	μ	0.0171***	(0.0024)	μ	0.0320***	(0.0024)	μ	0.0141***	(0.0051)	μ	0.0154 ***	(0.0053)
	φ_1	0.0170***	(0.0094)	φ_1	-0.1022***	(0.0094)	φ_1	-0.1167***	(0.0205)	φ_1	-0.1166 ***	(0.0263)
	φ_2	-0.2895***	(0.0097)	φ_2	-0.1808***	(0.0097)	φ_2	-0.1760***	(0.0237)	φ_2	-0.1747 ***	(0.0292)
	σ_δ	0.1480***	(0.0016)	σ_δ	0.1512***	(0.0016)	σ_δ	0.1483***	(0.0106)	σ_δ	0.1519 ***	(0.0103)
	σ_η	0.1309***	(0.0013)	σ_η	0.0418***	(0.0013)	σ_η	0.0410***	(0.0029)	σ_η	0.0406 ***	(0.0030)
	ψ	1.0000***	(0.0159)	ψ	0.9988***	(0.0159)	a_1	1.0034***	(0.0071)	a_1	1.0033 ***	(0.0139)
				a_1	1.0887***	(0.0213)	$a_1 - a_3$	0.9475***	(0.0481)	$a_1 - a_3$	0.9358 ***	(0.0511)
				p_{11}	0.9442***	(0.0552)	p_{11}	0.8059***	(0.0590)	p_{11}	0.8055 ***	(0.0596)
				p_{22}	0.3965***	(0.1211)	p_{22}	0.9149***	(0.0309)	$\frac{p_{12}}{1-p_{11}}$	0.9954 ***	(0.0020)
							p_{33}	0.1280	(0.0717)	p_{22}	0.9135 ***	(0.0550)
										p_{33}	0.1236 ***	(0.0683)
Implied Paras	R	0.0000		R	0.0012		R	0.0034		R	0.0033	
	a_1			a_1	1.0887		a_1	1.0034		a_1	1.0033	
	a_2			a_2	0.0544		a_2	1.0959		a_2	1.0961	
	a_3			a_3			a_3	0.0560		a_3	0.0674	
	π_1			π_1	0.9154		π_1	0.2853		π_1	0.2890	
	π_2			π_2	0.0846		π_2	0.6511		π_2	0.6469	
	π_3			π_3			π_3	0.0635		π_3	0.0642	

A quick glance at the model selection criteria reveals that the multiple-regimes models are much more preferable than the one- regime model. For example, after the number of regimes increase from 1 to 2, the log likelihood increase by 87.20%, while the AIC (BIC) decrease by 89.38% (92.56%). We can also find that the estimated collapsing drifts are small (around 0.05) for all multiple-regimes models, which are consistent with the fact that the bubble collapses dramatically only in one step in the Evans process.

By comparing three multiple-regimes models with AIC and BIC, we find the model R3V1 works best while the model R2 performs worst. The predominance of the model R3V1 stems from the fact that the the specification of the transitional probability matrix of its bubble process is consistent with the transitional process of the Evans' bubble, which recursively follows the order of rational surviving, exploding, and collapsing. The model R2 can not differentiate the rational surviving and exploding states and have to combine these two states into one regime, which leads

to smaller log likelihood. Although the model R3V2 allow the bubble in the surviving state to enter the collapsing regime directly, the estimate of the corresponding transition probability is close to zero and the overall parameter estimation is similar to that of R3V1. This is because the simulated evans bubble process doesn't have this feature. Because R3V2 has 1 more parameter, it is equipped with worse AIC and BIC than R3V1.

The second and third figures in Figure 4.4.2, 4.4.3 and 4.4.4 present the filtered bubble and state probability series for multiple-regime models. We can see all of them can detect the collapsing bubble periods accurately. And both three-regimes model can detect the exploding states, whose filtered probabilities grow gradually to almost 1 right before each collapsing.

Figure 4.4.2: Evans-two regimes

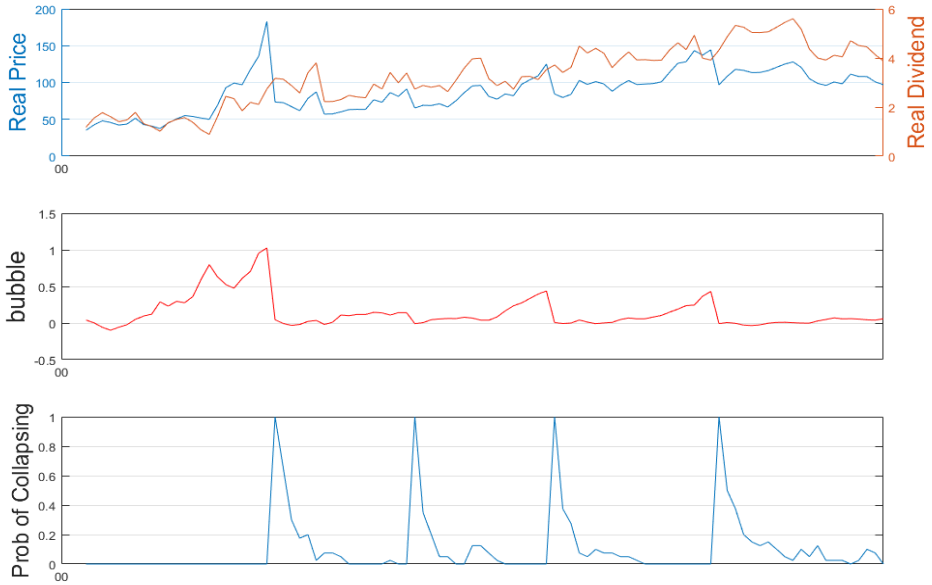
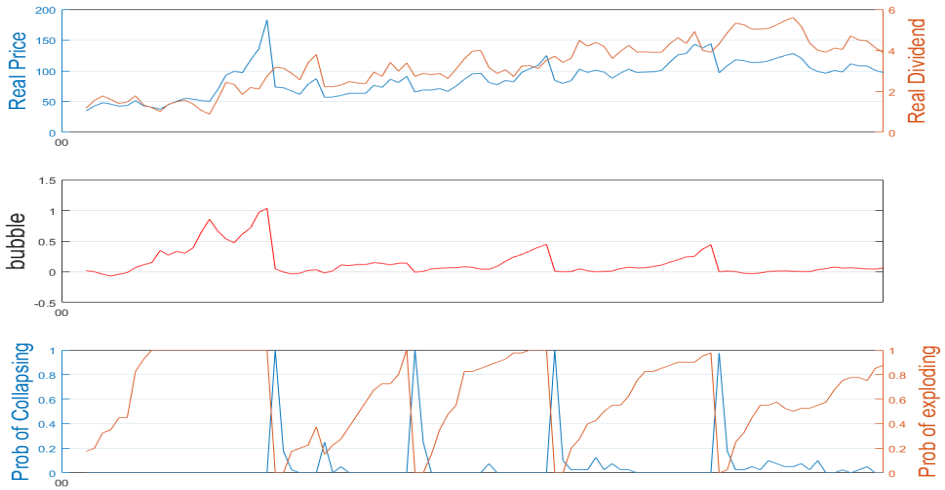


Figure 4.4.3: Evans-three regimes-V1

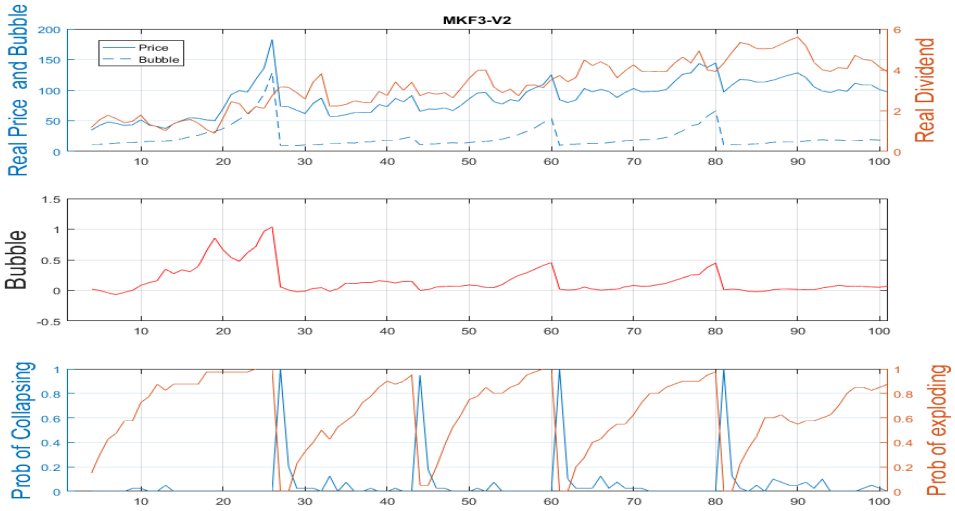


4.4.2 US real data

We follow the same modeling procedures as the one used in artificial data to the quarterly real S&P500 index and dividend data through the whole history from 1871 to 2017. The real data is obtained by adjusting the nominal data with CPI. The overall conclusions we can draw match those in the artificial data, which indicates the reasonableness of the Evans process, as well as our proposed models. To be more specific, we can infer from table 4.4.3 that the regime-switching models achieve better approximation to the data than one-regime model since both AIC and BIC are in favor of the latter three models.

In this real data example, the model R3V2 produces best AIC and BIC while the model R3V1 gets dominated by the other two. One possible explanation of this is that the collapse of bubble happens more unexpectedly in real life. More often a crisis happens while the bubble is still rationally growing. Therefore, the model that forces the bubble to collapse only after the exploding regime may be too strict. This can also be shown by the filtered probabilities exhibited in Figure 4.4.7 and 4.4.8. With strong restriction, R3V1 only detects one collapse, which is during the big

Figure 4.4.4: Evans-three regimes-V2



recession in 1930s, while R3V2 identifies most of the financial collapsing periods.

Since the data shows that it is more often that the collapse follows directly by rational regime in the US market, both R2 and R3V2 performs well in the probability filtering task in the sense that they successfully detects several major crises in history including: a series of panics during the Long Depression, the Great Depression followed by Wall Street Crash of 1929, the Secondary Banking Crisis of 1973-75, Black Monday in 1987 and the Subprime Mortgage Crisis in 2007. The only major crisis our model fails to detect is the Dot-com Bubble during which stock price decreased gradually rather than collapsed all of a sudden.

4.5 Conclusion

This chapter has proposed a new framework for modeling the periodically growing and collapsing bubble process. Under this framework, one set of discrete conditional Dynamic Linear models are introduced to capture the regime-switching characteristic of speculative bubbles. The novel Monte Carlo based Mixture Kalman Filtering method has been employed to fulfill the model

Table 4.4.3: Estimation summary: S&P500 Quaterly

	1 Regime			2 Regimes			3 Regimes - V1			3 Regimes - V2		
Model Evaluation	loglike	1666.2583		loglike	1709.9061		loglike	1699.2991		loglike	1719.9248	
	AIC	-3320.5167		AIC	-3401.8122		AIC	-3378.5981		AIC	-3417.8496	
	BIC	-3296.1061		BIC	-3362.5760		BIC	-3335.0024		BIC	-3369.8943	
Para Estimates	μ	0.0029***	0.0001	μ	0.0033***	0.0010	μ	0.0031***	0.0007	μ	0.0026	0.0027
	φ_1	0.2448***	0.0024	φ_1	0.2255***	0.0866	φ_1	0.2379***	0.0185	φ_1	0.2186***	0.0984
	φ_2	-0.0093***	0.0018	φ_2	-0.0278	0.0445	φ_2	-0.0151	0.0147	φ_2	-0.0072	0.0966
	σ_δ	0.0345***	0.0001	σ_δ	0.0356***	0.0033	σ_δ	0.0346***	0.0006	σ_δ	0.0346***	0.0031
	σ_η	0.0963***	0.0003	σ_η	0.0821***	0.0101	σ_η	0.0890***	0.0012	σ_η	0.0830***	0.0068
	ψ	1.0000***	0.0013	ψ	0.9999***	0.0925	a_1	1.0002***	0.0006	a_1	1.0001***	0.0028
				a_1	1.0118***	0.0196	$a_1 - a_3$	0.9890***	0.0692	$a_1 - a_3$	0.6635***	0.2118
				p_{11}	0.9851***	0.0900	p_{11}	0.9981***	0.0029	p_{11}	0.9922***	0.0701
				p_{22}	0.2200*	0.1288	p_{22}	0.2584***	0.0601	$\frac{p_{12}}{1-p_{11}}$	0.3920**	0.1701
							p_{33}	0.5077***	0.0316	p_{22}	0.0225	0.0789
										p_{33}	0.7004***	0.2047
Implied Paras	R	0.0000		R	0.0001		R	0.0002		R	0.0001	
	a_1			a_1	1.0118		a_1	1.0002		a_1	1.0001	
	a_2			a_2	0.3906		a_2	2.4900		a_2	1.5189	
	a_3			a_3			a_3	0.0112		a_3	0.3366	
	π_1			π_1	0.9812		π_1	0.9936		π_1	0.7007	
	π_2			π_2	0.0188		π_2	0.0025		π_2	0.2810	
	π_3			π_3			π_3	0.0038		π_3	0.0183	

fitting and estimation task.

To check the validity of this framework, we apply it to simulated observations and US stock market data. Results show a significant gain on goodness of fit as well as a set of theory-coherent estimators. Moreover, our model exhibits a strong capability in bubble detection in both artificial and real-data examples. In the mean-time, the rational expectation condition is satisfied in the model construction.

The proven goodness of fit of our proposed model also shows the fact that the speculative bubble process should be inherently non-linear. Certain consideration on this non-linearity is needed when specifying this periodical process.

Although we only show three Markov-switching models to approximate the bubble process, other more general specifications can be employed under this framework. For example, a four-regimes model with certain probability construction can be studied. A three regimes model which allows the bubble to explode immediately after collapsing could be another possibility. The same model fitting and estimation procedure as described above can be taken.

Figure 4.4.5: US-one regime

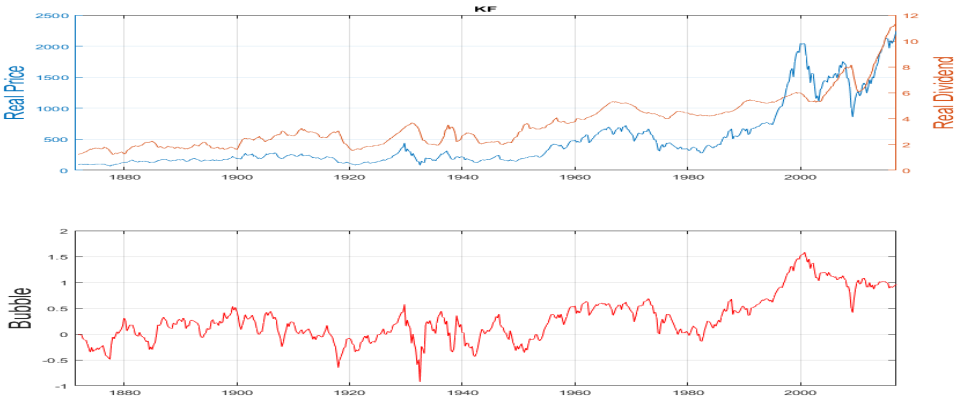


Figure 4.4.6: US-two regimes

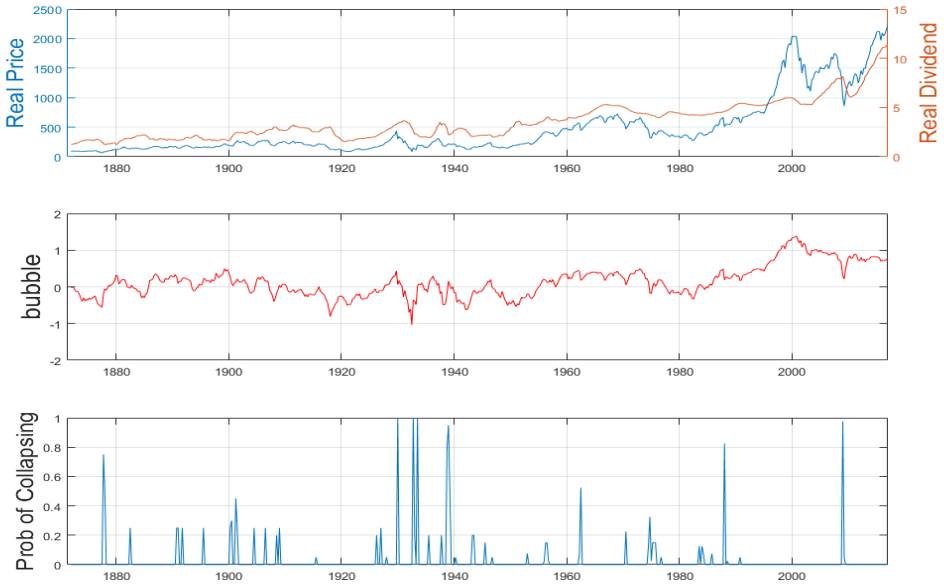


Figure 4.4.7: US-three regimes-V1

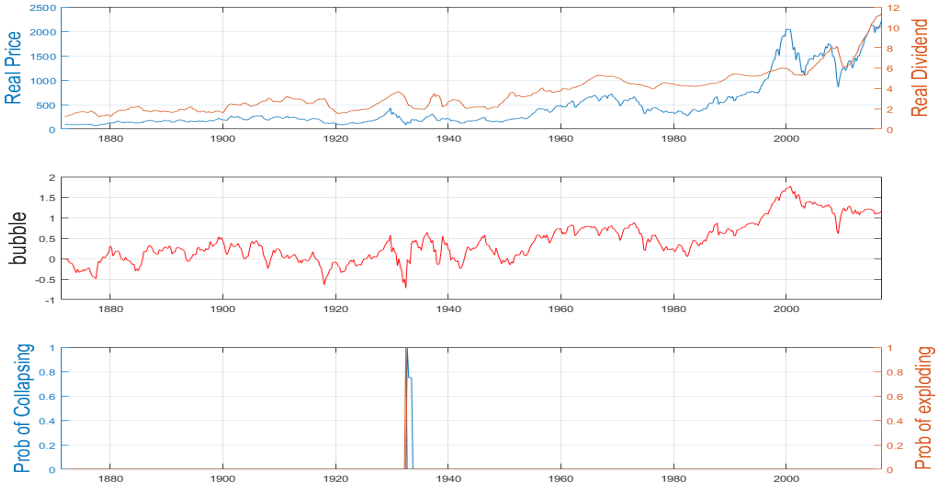
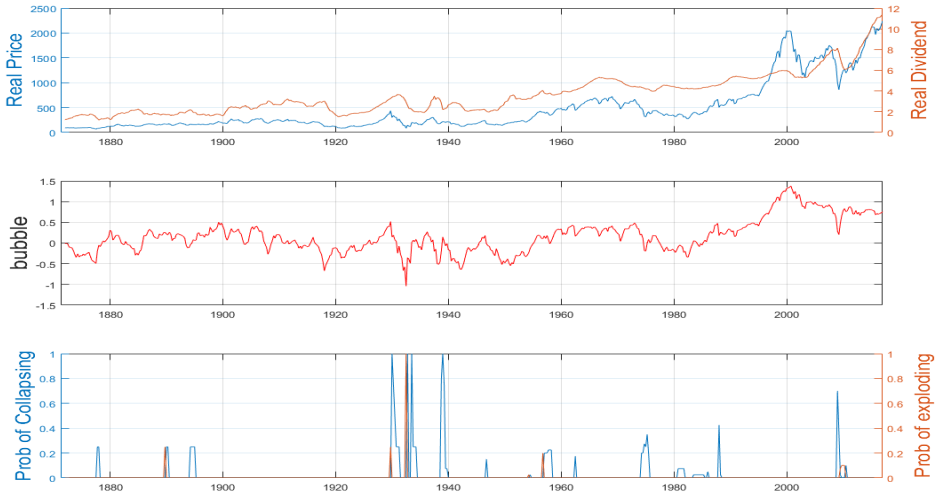


Figure 4.4.8: US-three regimes-V2



Chapter 5

A conditional dynamic linear model approach for real-time Blood Glucose Monitoring

5.1 Introduction

Diabetes mellitus (DM), commonly referred to as diabetes, is a group of metabolic disorders in which there are high blood sugar levels over a prolonged period (World Health Organization and others, 2014). According to World Health Organization (World Health Organization and others, 2016), an estimated 422 million adults were living with diabetes in 2014, which has nearly doubled in ratio since 1980. Type 1 diabetes mellitus, also referred to "insulin-dependent diabetes mellitus" (IDDM) or "juvenile diabetes", is characterized by loss of the insulin-producing beta cells of the pancreatic islets, leading to insulin deficiency (Rother, 2007). Insulin-injection type of treatments, including insulin pumps, multiple injections have been proved to be necessary and most effective methods to manage blood glucose levels for Type 1 Diabetes patients, according to a series of researches including the Diabetes Control and Complication Trial(DCCT) (Centers for Disease Control and Prevention and others, 2011). Following this, a system of closed-loop artificial pancreas (Weinzimer et al., 2008) incorporating continuous glucose sensors and insulin pumps has been proposed and proved significant effectiveness in glycemic management. The closed-loop artificial pancreas closely monitor the patient's blood glucose level in a real-time manner and conduct insulin injection accordingly, leading to a relatively stable blood glucose level of the patient. Relevant studies on artificial pancreas has shed some light on continuous blood glucose control and hyperglycemia prevention.

One of the keys to a powerful system of artificial pancreas is an accurate and robust real-time

continuous glucose monitoring(CGM) algorithm relying on both the glucose biosensors and the fingerstick measurements. Insulin pumps' injection dose rely on the output of CGM algorithm. An under- or overestimation of real-time blood glucose level may lead to over- or under-injection of insulin, which could result in patients' deteriorated health condition. However, the incompetence of accurate continuous glucose monitoring remains the main challenge in the artificial pancreas development. The difficulty comes from both the technical deficiency of the biosensor and the inefficient statistical modeling of the CGM algorithm. Due to the technical limit in the biosensor, the CGM algorithm requires routine calibration according to the device and patient. This procedure is in principle challenging and more care should be taken in statistical modeling of CGM algorithm.

In this chapter we address the above challenge by proposing a time series model which captures the nature of the CGM problem thus yields accurate and precise real-time blood glucose level estimation and prediction. Inspired by the biological structure of the biosensor signal, fingerstick measurement and blood glucose system as well as the on-line property of the algorithm, we employ the state space framework and associated filtering techniques to fit the real-time CGM problem. Detailed implementation of this SSM-based CGM algorithm includes two main component: periodical and proper parameter estimation and statistical inference(including estimation, prediction, etc) on blood glucose levels. The carefully designed algorithm is applied and assessed via an important dataset, Star 1 dataset. The performance exhibited in different dimensions all show a significant improvement over the existing CGM algorithm.

The rest of this chapter is organized as follows. In section 5.2, we introduce the dataset and derivation of the model. A detailed algorithm go-through based on a subsample will be illustrated in section 5.3. Section 5.4 covers the numerical analysis of the estimation and prediction performance over different CGM algorithms. A brief conclusion and discussion will be in section 5.5.

5.2 The state space representation of continuous glucose monitoring

5.2.1 The Star 1 dataset

In the Star 1 study (Hirsch et al., 2008), 137 subjects with type 1 diabetes were followed for 6 months, on average, using a CGM device (developed by Medtronic MiniMed). The first series of information comes from the blood glucose biosensor every 5 minutes in a form of electrical current measurement (Wang, 2008). Throughout this chapter, we denote this interstitial signal as ISIG(t). ISIG(t) is measured in nano amps, nA. Another more accurate but less frequent source of measurement is fingerstick measurement–each patient in the study pricks their finger to obtain a small droplet of blood to be analyzed by a blood glucose meter. Due to its strict procedure, the fingerstick measurement(in mg/dL), denoted as FS(t) is only taken every 6 hours on average. The above two measurements are entered into the CGM system and processed by the CGM algorithms to yield the blood glucose estimation.

Since most of the time there's only ISIG(t) available, the blood glucose estimation is based mainly on ISIG(t). Ultimately, for an artificial pancreas, one hopes to get free from fingerstick measurements. However, FS(t) is essential helping calibrating the algorithm due to the limit of the existing CGM device. For simplicity, we scale t as every 5 minutes since this is the smallest time interval in our study. Notation-wise, we record FS(t) = NA if there's no fingerstick measurement.

While one should build a continuous glucose monitoring algorithm based mainly on ISIG(t) and FS(t), there's other useful information in the Star 1 dataset. The biosensor are replaced approximately every three days on average. The sensor ID codes are recorded to help tracking the replacement of the sensor. There's also a time series of output of an existing, proprietary CGM algorithm denoted by CGM(t). CGM(t) provides a standard of comparison by which we can test the performance of our proposed methods. Figure 5.2.1 shows the ISIG and FS measurement with CGM output and indicator of replacement.

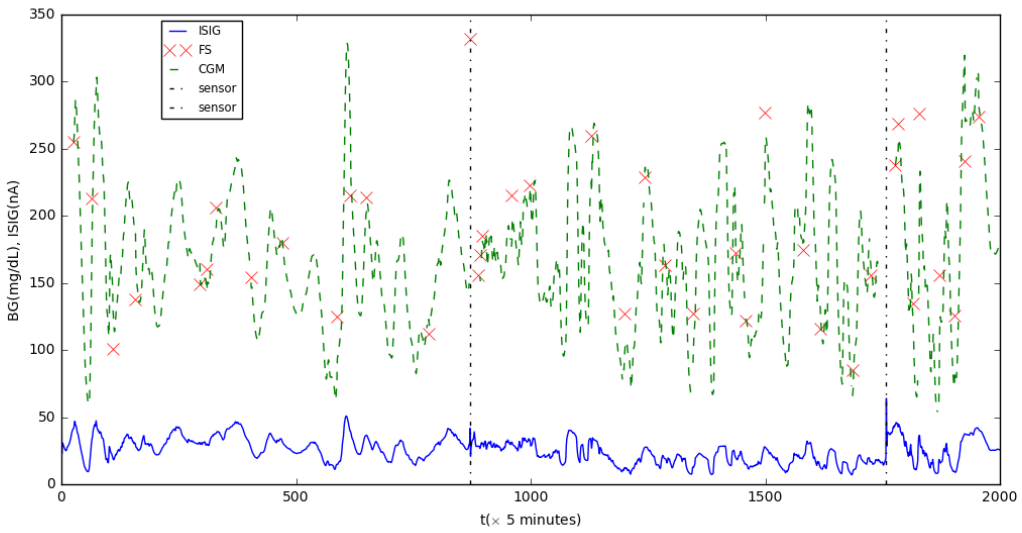


Figure 5.2.1: ISIG(t), FS(t), CGM(t) and sensor replacement for Subject 1 in Star 1 dataset

5.2.2 Modeling the blood glucose biosensors

As stated above, essentially the CGM algorithm aims at estimating the blood glucose density inside the patient's body, denoted as BG(t), given the interstitial signal ISIG(t) and fingerstick measurement FS(t). There have been discussions on the relationship between the measurements and the blood glucose density (see Khan et al., 2006; Mahoney and Ellison, 2007; Steil et al., 2005; Wang, 2008). The fingerstick measurements taken at $t = \tau_k$, are believed to be more reliable than CGM measurements, although there's still error term existing. Therefore we have

$$\text{FS}_k = \text{BG}_k + \sigma_1 \epsilon_k, \tag{5.1}$$

where ϵ_k are iid error terms with $\text{E}\epsilon_k = 0$ and $\text{Var}(\epsilon_k) = 1$. Here we assume $\epsilon_k \sim N(0,\ 1)$. There're studies on potential bias of fingerstick measurements($\text{E}\epsilon_k \neq 0$) (Khan et al., 2006), which haven't reached to any conclusive results. So for the scope of this Chapter we take FS(t) as the better measurement of the two which should in principle reflect the BG(t) with minor error. The main

and obvious drawback of FS(t) is that they are mostly NA's thus can only be taken for calibration.

On the other hand, the CGM measurement ISIG(t) reflects the glucose density near the interstitial fluid, denoted as IG(t). There is a time lag between IG(t) and BG(t) due to the diffusion of blood glucose molecules into interstitial fluid. The lag can be modeled (Steil et al., 2005) by the equation

$$\text{IG}(t) = \int_0^\infty \text{BG}(t-u)\rho^{-1}e^{-u/\rho}du.$$

Differentiating the equation leads to

$$\text{BG}(t) = \text{IG}(t) + \rho\text{IG}'(t).$$

In reality, ρ is relatively small and the second term in the above equation contributes very little to BG(t). Therefore, it is sometimes legitimate (Dicker et al., 2013) to assume $\rho = 0$ for simplicity. We'll assume $\text{BG}(t) = \text{IG}(t)$ in the following discussions while leave the scenario of $\rho \neq 0$ in the discussion.

It remains to discuss the relationship between ISIG(t) and IG(t). A linear approximation is suggested (Heller and Feldman, 2008), i.e.

$$\text{ISIG}(t) = \alpha(t)\text{IG}(t),$$

where $\alpha(t)$ is a slowly changing stochastic process. In principle the CGM biosensor attempts to get a stable estimation of IG(t), i.e. $\alpha(t) \equiv \alpha$. However, there are two obstacles to this due to the limit of current biosensors. Firstly, the biosensors' measurements base on interstitial signal, which is much more volatile than fingerstick measurements. So to be more rigorous, the equation should take the form

$$\text{ISIG}(t) = \alpha(t)\text{IG}(t) + \sigma_2\eta(t),$$

where $\eta(t)$ is modeled as iid standard normal distribution, and σ_2 is in natural larger than σ_1 after scaling. Secondly, the biosensors' sensitivity decays over time due to biofouling from the CGM device. Thus one can observe decreasing $\alpha(t)$ during the life period of one sensor. A basic approach on $\alpha(t)$ is to depict it as a linear process $\alpha(t) = \alpha_0 + \alpha_1(t - t_i),\ t \in (t_i, t_{i+1})$, where t_i is when a new sensor is installed. Figure 5.2.2 records the available ratios of ISIG(t)/FS(t) during the lifetime of a randomly picked sensor for Subject 1. Since FS(t) is approximately BG(t), the points

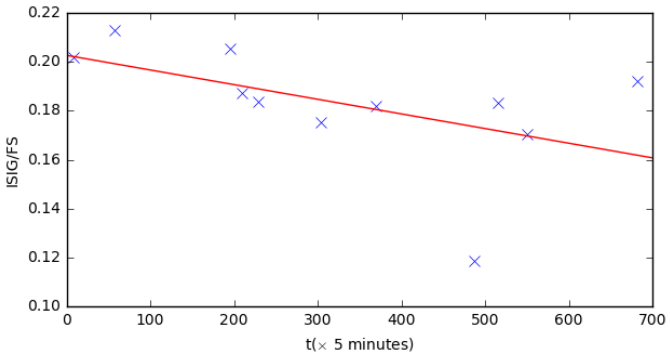


Figure 5.2.2: ISIG(t)/FS(t) for Subject 1 during the life period of sensor ID: A761_083210, with the linear regression line in red.

are approximately samples of $\alpha(t)$. This supports our linear assumption on $\alpha(t)$. Considering BG(t)=IG(t), we have the second observation equation

$$\text{ISIG}_t = (\alpha_0 + \alpha_1(t - t_i))\text{BG}_t + \sigma_2\eta_t,\ t \in (t_i, t_{i+1}). \tag{5.2}$$

Equations (5.1) and (5.2) together build the relationships between BG(t) with ISIG(t) and FS(t), with which a CGM algorithm can make statistical inference.

5.2.3 The state space representation

While Dicker et al. (2013) uses FS(t) as the true BG(t) to calibrate $\alpha(t)$ when it's available, we propose to take this series as an additional source of observation, only with much higher accuracy. To better support the estimation, we make assumptions on the patients' blood glucose dynamics to

form a state equation, i.e.

$$\mathrm{BG}_{t+1} \mid \mathrm{BG}_t \sim f(\cdot \mid \mathrm{BG}_t), \tag{5.3}$$

which, together with equations (5.1) and (5.2), forms a state space representation of continuous blood glucose monitoring problem.

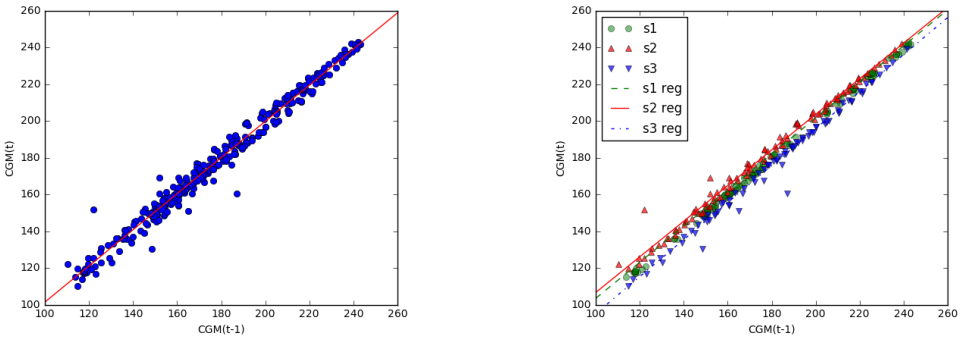
To start with, a tentative assumption one can make for blood glucose dynamic is simply setting the state space model as

$$\begin{aligned} \mathrm{ISIG}_t &= (\alpha_0 + \alpha_1(t - t_i))\mathrm{BG}_t + \sigma_2\eta_t,\ t \in (t_i, t_{i+1}), \\ \mathrm{FS}_t &= \mathrm{BG}_t + \sigma_1\epsilon_t,\ t = \tau_k, \\ \mathrm{BG}_{t+1} &= \mu + \beta\mathrm{BG}_t + \sigma_b\nu_{t+1}, \end{aligned} \tag{5.4}$$

where η_t, τ_t, ν_t are iid standard normal variables. This is an over-simplified model in the sense that we don't take the periodicity of blood glucose dynamic into account. Instead we assume a simple AR(1) dynamic to represent the state equation. This model can be efficiently fitted and estimated by Kalman Filter techniques since all components are linear and Gaussian. Figure 5.2.3a shows the scatter plot of CGM(t) against CGM($t-1$) from a subsample of Subject 1. In the belief that existing CGM time series can reflect the rough shape of BG dynamic, this preliminary study shows some reasoning of initiating with a simple AR(1) process.

The limit of model (5.4) is as obvious–the periodicity of the BG dynamic suggested by Figure 5.2.1 is ignored. This would lead to a relatively large σ_b, thus less estimating power from the state equation. Moreover, the prediction is mainly based on the state equation while the AR(1) assumption may not do well in prediction. For example, when the patient just finishes taking meal, a good CGM algorithm should expect a higher BG_{t+1} than BG_t while the AR(1) assumption wouldn't.

In hope to allow more flexibility to help blood glucose estimation and prediction, we introduce



(a) (b)

Figure 5.2.3: Scatter plot of CGM(t) against CGM($t-1$) for Subject 1 with one day, with the regression line. (a) takes the whole data while b seperates the data into three groups.

the three regimes Markov switching model as follows,

$$\begin{aligned} \text{ISIG}_t &= (\alpha_0 + \alpha_1(t - t_i))\text{BG}_t + \sigma_2\eta_t,\ t \in (t_i, t_{i+1}), \\ \text{FS}_t &= \text{BG}_t + \sigma_1\epsilon_t,\ t = \tau_k, \\ \text{BG}_{t+1} &= \text{BG}_t - \theta(\text{BG}_t - m) + J_{t+1}M_1 - I_{t+1}M_2 + \sigma_b(J_{t+1},\ I_{t+1})\nu_{t+1}, \end{aligned} \tag{5.5}$$

where the indicator series J_t and I_t takes value 0 and 1 respectively. The series J_t indicates the body's absorbing outside blood glucose molecules, leading to a constantly increasing blood glucose level. Similarly, I_t indicates the injection of insulin which cause a constant decreasing blood glucose level. When there's no meal activity or insulin injection, the blood glucose should be maintained in a relatively stable way through a mean reverting process caused by the insulin inside the patient's body. We also allow the variation term σ_b to be stage dependent. In practice, the injection of insulin is associated with the meal activity. That's also why there's always a sharp decreasing(insulin's effect) after a sharp increasing(meal's effect). So we restrict J_t and I_t to begin at the same time while J_t ends earlier than I_t. To be more specific, we combine the two indicators into one series λ_t which takes value 1, 2 and 3, representing stable($J_t = 0$, $I_t = 0$), increasing ($J_t = 1$, $I_t = 1$) and decreasing($J_t = 0$, $I_t = 1$) stages. And we set λ_t to be a Markovian stochastic process

with transition probability $\mathrm{P}(\lambda_{t+1} = j \mid \lambda_t = i) = p_{ij}$ forming the transition probability matrix

$$P = \begin{bmatrix} p_1 & 1-p_1 & 0 \\ 0 & p_2 & 1-p_2 \\ 1-p_3 & 0 & p_3 \end{bmatrix}. \tag{5.6}$$

Figure 5.2.3b shows a simple grouping of the three stages. The top red up triangles have a relatively larger drift while the bottom blue down triangles have smaller drift. The green round dots stays in between, representing the stable stage.

Ultimately, the regime detection job could be done by outside input in the CGM device since the timing of meal and insulin injection is determined. Yet the current biosensor does not take this input. Under the conditional dynamic linear model framework, we are still able to detect the regime from the observation and knowledge on the blood glucose dynamic. One might have doubt on the assumption of a Markovian structure on the trajectory transition, arguing that probability of being decreasing in the next step in a high blood glucose level cannot be the same as that in a low blood glucose level. Although the nature of blood glucose dynamic links the transition probability to the blood glucose level, it's hard to model this relationship in a concrete way. Instead, our flexible Markovian assumption leaves the detection of regime switching to the combination of both observation equation and state equation. The numerical study will show that with properly estimated parameters, the regime allocation is reasonable.

Combining (5.5) and (5.6) with $\beta = 1 - \theta$, $\mu_1 = \theta m$, $\mu_2 = \theta m + M_1 - M_2$, $\mu_3 = \theta m - M_2$, we have the following companion form of a conditional dynamic linear model of continuous blood glucose monitoring problem:

$$\begin{aligned} \mathrm{ISIG}_t &= (\alpha_0 + \alpha_1(t - t_i))\mathrm{BG}_t + \sigma_2\eta_t,\ t \in (t_i, t_{i+1}), \\ \mathrm{FS}_t &= \mathrm{BG}_t + \sigma_1\epsilon_t,\ t = \tau_k, \\ \mathrm{BG}_{t+1} &= \mu_{\lambda_{t+1}} + \beta\mathrm{BG}_t + \sigma_{b\lambda_{t+1}}\nu_{t+1}, \\ \mathrm{P}(\lambda_{t+1} = j \mid \lambda_t = i) &= p_{ij},\ i,\ j = 1,\ 2,\ 3, \end{aligned} \tag{5.7}$$

where p_{ij} follows (5.6).

5.3 Study on a subsample

Based on model (5.4) and (5.7), one can apply Kalman Filter and Mixture Kalman Filter to make blood glucose estimation and prediction. Beforehand, the model fitting can be done via maximum likelihood estimator discussed in chapter 3. There are several issues that need to be addressed in the CGM algorithm. Given the interstitial signal ISIG(t) and sometimes fingerstick measurement FS(t), the proposed CGM algorithm should yield the estimator $\hat{\text{BG}}(t)$ and predictor $\hat{\text{BG}}(t+h)$. In this section, we illustrate our proposed CGM algorithm based on techniques associated with Kalman Filter(in model (5.4)) and Mixture Kalman Filter(in model (5.7)). We start with a subsample of time series to go through the procedure. The estimation and prediction performance will be discussed in section 5.4.

As an example to discuss the detailed implementation of our proposed CGM algorithm, we pick a random subsample of Subject 1 when there are sufficient CGM algorithm estimation, which is missing approximately 17.5% of the whole length of Subject 1 in a clustered manner. Figure 5.3.1 shows the time series of the necessary information–ISIG and FS are the input of CGM algorithm while CGM is for comparison.

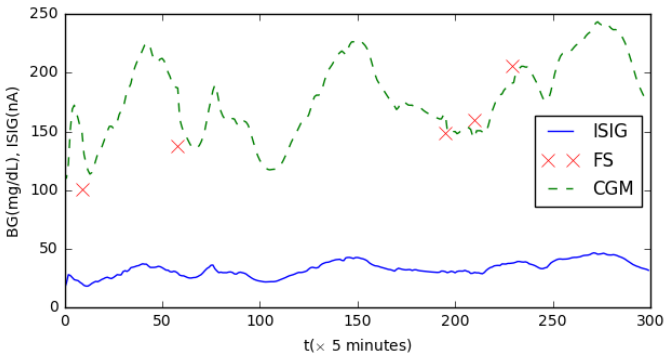


Figure 5.3.1: One day series of ISIG, FS and CGM for Subject 1, obtained by one biosensor with no replacement.

5.3.1 CGM algorithm based on Kalman Filter

Model (5.4) is linear and Gaussian and thus can be solved by Kalman Filter which is introduced in Section 2.2. Therefore the model can be fitted by finding the MLE $\hat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}}\{-\log L(\text{ISIG}_T, \text{FS}_K \mid \boldsymbol{\theta})\}$, the estimated parameters with standard deviation are reported in Table 5.3.1. One can observe that the parameter estimations from the observation equation are much sharper than those in the state equation(except for σ_1 which is due to the lack of fingerstick measurements). Also σ_2/α_0 is still smaller than σ_b, which means that the observation equation is the main contributor in filtering. This is due to the over-simplification of blood glucose dynamic modeling.

	α_0	α_1	σ_1^2	σ_2^2	μ	β	σ_b
M	0.213	-9.472$\times 10^{-5}$	5.097	3.365	2.903	0.985	20.250
SD	0.011	1.521$\times 10^{-5}$	2.585	0.756	1.384	0.012	4.112

Table 5.3.1: MLE parameter estimation results for model (5.4)

With the fitted model, the blood glucose estimator $\widehat{\text{BG}}_t$ and its associated variance $\text{Var}(\text{BG}_t)$ can be obtained by the posterior mean $\hat{\mu}_{t|t}$ and variance $\hat{\Sigma}_{t|t}$ from Kalman Filter. Figure 5.3.2 shows the estimated $\widehat{\text{BG}}_t$ and 95% confidence interval. The difference between the KF estimation and the CGM estimation in the early stage comes from our assumption of a gradually decreasing ISIG/IG ratio, which is shown in figure 5.2.2. In addition one would find $\widehat{\text{BG}}_{\tau_k}$'s be closer to FS_k's when the latter is available, which also supports the decreasing ISIG/IG ratio assumption. Moreover, the prediction based on Kalman Filter can be done by setting $\widehat{\text{BG}}_{t+1} = \hat{\mu}_{t+1|t}$ and $\text{Var}(\text{BG}_{t+1}) = \hat{\Sigma}_{t+1|t}$, the RHS of the two equations being the predictive mean and covariances in Kalman Filtering. We'll leave the details in the later section.

5.3.2 CGM algorithm based on Mixture Kalman Filter

Table 5.3.1 and Figure 5.3.2 both show the improvement of model (5.4) as a CGM algorithm along with its potential drawbacks. As stated in section 5.2.3, model (5.7) tries to cover the drawbacks of model (5.4) by a more realistic depiction on the blood glucose dynamics. As stated in

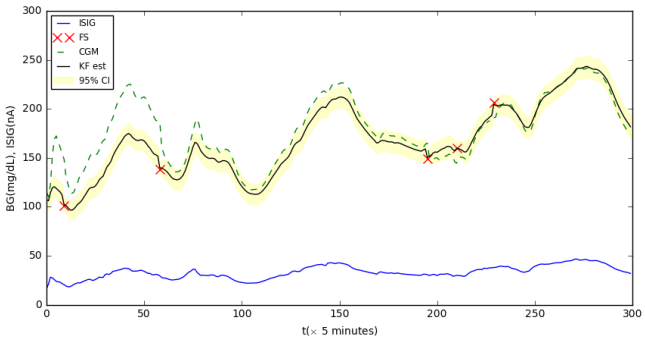


Figure 5.3.2: KF estimated $\widehat{\mathrm{BG}}_t$ with 95% confidence band, compared with CGM, FS and ISIG.

section 2.4, Mixture Kalman Filter(MKF) aims to efficiently do estimation and prediction of the conditional dynamic linear model. Still maximum likelihood estimator is employed for model fitting, referring to chapter 3. The estimated parameters and standard deviations are recorded in Table 5.3.2. The first line of the table includes the trajectory-independent parameters, which

	α_0	α_1	σ_1^2	σ_2^2	β	
M	0.202	-5.618$\times 10^{-5}$	5.535	3.423	0.982	
SD	0.013	1.716$\times 10^{-5}$	2.329	0.749	0.009	
	μ_1	μ_2	μ_3	σ_{b1}^2	σ_{b2}^2	σ_{b3}^2
M	3.096	7.617	-5.860	10.467	11.157	12.112
SD	0.926	1.361	1.070	3.428	3.275	4.174
	p_1	p_2	p_3			
M	0.9268	0.9251	0.8402			
SD	0.0295	0.0425	0.0566			

Table 5.3.2: MLE parameter estimation results for model (5.7)

are similar as those in Table 5.3.1, especially those parameters in the observation equation. The trajectory-dependent parameters, on the other side, gives much more information than model (5.4). Firstly, the σ_b^2's are approximately half than that in Table 5.3.1, which indicates a much larger contribution from state equation, i.e. the blood glucose dynamics. Secondly, the parameters in the original regime-switching model (5.5) are: $\hat{\theta} = 1 - \hat{\beta} = 0.018$, $\hat{m} = \hat{\mu_1}/\hat{\theta} = 172.00$, $\hat{M_1} = \hat{\mu_2} - \hat{\mu_3} = 13.477$, $\hat{M_2} = \hat{\mu_1} - \hat{\mu_3} = 8.856$. These implied parameters together reflect

the patient's blood glucose dynamics, thus can help understand the patients health condition. For example, m stands for the stationary blood glucose level while θ represents the mean reverting speed.

Applying the appropriately estimated parameters into the Mixture Kalman Filter algorithm, The blood glucose mean and variance estimator, $\widehat{\text{BG}}_t$ and $\text{Var}(\text{BG}_t)$, can be obtained by taking weighted average of $\hat{\mu}_{t|t}^{(j)}$ and $\hat{\Sigma}_{t|t}^{(j)}$, where j corresponds to sampled trajectory paths. To better detect the trajectory variable λ_t, a 4-steps delayed strategy is adopted, i.e. $p(\lambda_{t+1} \mid \boldsymbol{\lambda}_t, KF_t, \boldsymbol{y}_{t+5})$ is employed as the trial distribution. Here $\boldsymbol{y} = (\text{ISIG}, \text{FS})$. Figure 5.3.3 includes a summary of the MKF results on the specific subsample. Figure 5.3.3a covers the filtered series along with the confidence band. With more flexibility introduced by the state equation, one can observe both sharper increase and decrease than Figure 5.3.2. Due to the assumption that fingerstick measurements contains small error, both filter 'drags' the estimator $\widehat{\text{BG}}_t$ towards FS when the latter is available. Yet this transition is much smoother in the MKF case than that in the KF case, which also indicates a better fitting.

Figure 5.3.3b and 5.3.3c addresses the detection of trajectory in two dimensions: the former records the marginal probability of $P(\lambda_t = i \mid \boldsymbol{y}_T)$ while the latter records the optimal trajectory path, the paths with the highest weight. They both are coherent to the shape of the blood glucose level. In addition, the prediction algorithm can be derived utilizing $P(\lambda_t = i \mid \boldsymbol{y}_T)$. For example when $h = 1$,

$$\begin{aligned} \widehat{\text{BG}}_{t+1} &= \sum_i \left\{ \hat{P}(\lambda_t = i \mid \boldsymbol{y}_t) \sum_j \hat{p}_{ij} \hat{\mu}_{t+1|t} \right\} = \hat{\beta} \widehat{\text{BG}}_t + \sum_i \left\{ \hat{P}(\lambda_t = i \mid \boldsymbol{y}_t) \sum_j \hat{p}_{ij} \hat{\mu}_j \right\}, \\ \text{Var}(\widehat{\text{BG}}_{t+1}) &= \sum_i \left\{ \hat{P}(\lambda_t = i \mid \boldsymbol{y}_t) \sum_j \hat{p}_{ij} \hat{\Sigma}_{t+1|t} \right\}. \end{aligned} \tag{5.8}$$

	Kalman Filter	Mixture Kalman Filter
AIC	1303.15	1240.50
BIC	1329.07	1281.24

Table 5.3.3: Information criteria comparison between model (5.4) and (5.7)

One might also choose the most probable state to do prediction by setting

$$\widehat{\text{BG}}_{t+1} = \hat{P}(\lambda_t = i^* \mid \boldsymbol{y}_t) \sum_j \hat{p}_{ij} \hat{\mu}_{t+1|t},$$
$$\text{Var}(\widehat{\text{BG}}_{t+1}) = \hat{P}(\lambda_t = i^* \mid \boldsymbol{y}_t) \sum_j \hat{p}_{ij} \hat{\Sigma}_{t+1|t},$$

where $i^* = \text{argmax}_i \hat{P}(\lambda_t = i \mid \boldsymbol{y}_t)$. In practice both prediction methods have similar prediction accuracy. Here we adopt equation (5.8).

Since both Kalman Filter and Mixture Kalman Filter yield the data likelihood during model fitting, we provide the comparison of both Akaike Information Criterion AIC $= 2k - 2\log(\hat{L})$and Bayesian Information Criterion BIC $= \log(n)k - 2\log(\hat{L})$ as metrics of goodness of fit. Table 5.3.3 records the comparison of information criteria, both of which lean towards model (5.7).

5.3.3 Summary on SSM-based CGM algorithms

The CGM algorithm based on the state space model with or without regime-switching has been illustrated through the previous sections. A formalization of the state space model based CGM algorithm is now stated as follows:

Algorithm 5. *(ssm-based CGM algorithm)*

(A) Given $\boldsymbol{y}_t$, $t \in (t_i, t_i + K_i)$ where t_i is the time for sensor replacement, fit the model by either Kalman Filter or Mixture Kalman Filter, depending on the adopted model. Here $K_i = min\{k \geq K_1 | \#\{FS_t \neq NA\} \geq K_2\}$.

(B) Based on the estimated parameters, conduct model inference in either blood glucose estimation $\widehat{BG}_t$ or prediction $\widehat{BG}_{t+h}$.

(C) At time $t_i + K_i + jk$ where j is an integer and $t_i + K_i + jk < t_{i+1}$, re-estimate the model parameters. Repeat step (B) until sensor replacement.

Enough initial data is needed to generate an accurate state space model to fullfill the inference task. Therefore we introduce integers K_1 and K_2 for minimal ISIG and FS size to initiate the algorithm. Also the observation parameters $\alpha_0,\ \alpha_1,\ \sigma_2^2$ depend on the sensor's specification. So we set the period of the CGM algorithm as the life circle of the sensor. The small k is a tuning parameter balancing the performance and the computational cost. When $k = 1$, the CGM algorithm reset the parameters at every step, which may yield a more up-to-date model while requires more computational cost. Since the model parameters is in nature stable, a longer k is preferred from the consideration of efficiency.

5.4 Numerical study

5.4.1 Estimation and prediction accuracy

Ultimately the desired CGM algorithm should accurately estimate and predict the blood glucose level. Therefore we test the performance of the ssm-based CGM algorithms as stated in Algorithm 5 on the patient database. As for a metric of accuracy, we adopt a widely used overall measurement of performance for continuous blood glucose monitoring, mean absolute relative difference(MARD) (see Kovatchev et al., 2008; Dicker et al., 2013), in the form of

$$\text{MARD}(\widehat{\text{BG}}) = \text{Avg}(\text{ARD}) = \frac{1}{\#\{t\}} \sum_t \left\{ \frac{|\widehat{\text{BG}}_t - \text{BG}_t|}{\text{BG}_t} \right\}.$$

However BG_t is not attainable and FS_k is the only golden standard for testing performance. Thus replacing BG_t by FS_k, one gets

$$\text{MARD}(\widehat{\text{BG}}) = \text{Avg}(\text{ARD}) = \frac{1}{\#\{k\}} \sum_k \left\{ \frac{|\widehat{\text{BG}}_{\tau_k} - \text{FS}_k|}{\text{FS}_k} \right\}. \tag{5.9}$$

Following Dicker et al. (2013), Table 5.4.1 reports the summary of performance measurements

as a comparison of the three algorithms. The performance measurements include:

	Method	MARD(SD)	MedARD	ΔMARD	$N_{\text{MARD}}(N)$
Estimation	Mixture Kalman Filter	0.0132(0.0106)	0.0101	0.1589	10(10)
	Kalman Filter	0.0151(0.0109)	0.0110	0.1570	10(10)
	CGM	0.1721(0.1646)	0.1307		
Prediction w/ h=1	Mixture Kalman Filter	0.1317(0.1097)	0.1132	0.0563	10(10)
	Kalman Filter	0.1629(0.1293)	0.1408	0.0251	10(10)
	CGM	0.1880(0.1674)	0.1396		
Prediction w/ h=3	Mixture Kalman Filter	0.2098(0.1132)	0.1437	0.0239	10(10)
	Kalman Filter	0.2295(0.1963)	0.1688	0.0042	9(10)
	CGM	0.2337(0.2009)	0.1736		

Table 5.4.1: Summary statistics for accuracy comparison. $K_1 = 150$, $K_2 = 5$, $k = 20$ in Algorithm 5.

(1) MARD(SD): The total MARD's over all the subjects, associated with there standard deviations.

(2) MedARD: The Median of the ARD's over all the subjects.

(3) ΔMARD: The difference between MARD of the desired CGM alrogithm and the existing CGM algorithm.

(4) $N_{\text{MARD}}(N)$: N_{MARD} denotes the counts of subjects that the MARD of the desired CGM alrogithm is less than that of the existing CGM algorithm while N is the total subjects number.

The estimation performance of the SSM-based CGM algorithms is spuriously good because the target FS_k is also included in the model. In principle, one would always prefer to include FS_k into the observations for the purpose of absorbing more information. Yet if the true blood glucose level is presented, one can measure the accuracy at every step, with only a few FS_k's available. Still, accuracy performance would be better than the existing CGM algorithm with the incorporation of FS_k's.

To address the issue of double-incorporating FS_k's, we test the three CGM algorithms' prediction performance, replacing the $\widehat{\text{BG}}_{\tau_k}$ by $\widehat{\text{BG}}_{\tau_k - h + h}$ in equation (5.9). As discussed in the earlier

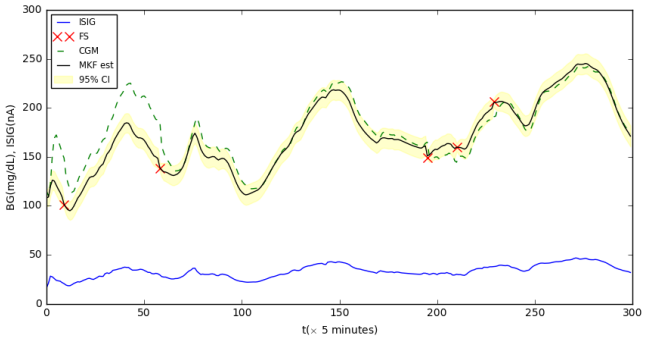
sections, $\widehat{\text{BG}}_{\tau_k-h+h}$ from SSM-based CGM algorithms are only based on observations up to time $\tau_k - h$, excluding FS_k. For the existing CGM algorithm, we directly set $\widehat{\text{BG}}_{\tau_k-h+h} = \widehat{\text{BG}}_{\tau_k-h}$. The cases $h = 1$ and $h = 3$ are examined with results shown in Table 5.4.1. Overall, both KF-based and MKF-based algorithms outperform the existing CGM algorithm in all metrics. Moreover, with the better fitted blood glucose dynamics, the MKF-based algorithm achieves significant prediction error reduction. Throughout the dataset, the Markov-switching MKF-based CGM algorithm is preferred to do both estimation and prediction.

5.5 Discussion

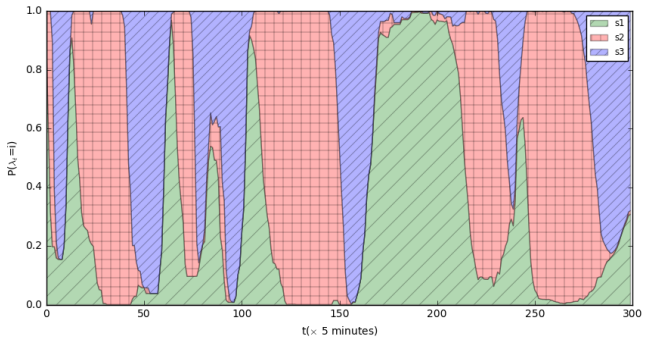
In this chapter the newly designed state space model for continuous blood glucose monitoring problem has been proposed. One can construct the CGM algorithms following the steps described in the chapter. The derivation of the model has shown its biological rightfulness as well as statistical efficiency. Then its great improvement in both blood glucose estimation and prediction are justified by comparing to the existing CGM algorithm under the Star 1 dataset. Among the two types of SSM-based CGM algorithms, the Markov-switching model yields further improvement in its goodness of fit to the human blood glucose dynamic, thus is preferred in clinical application, given sufficient computational resource.

Throughout the chapter, we have assumed that the time lag duration parameter $\rho = 0$ for simplicity. An extension to the model would be $\text{BG}(t) = \text{IG}(t) + \rho\text{IG}'(t)$ when $\rho \neq 0$. A possible way of modeling is taking $\text{IG}(t)$ instead of $\text{BG}(t)$ as the state variable and thus changing the fingerstick-associated observation equation to $\text{FS}_t = \text{IG}_t + \rho(\text{IG}_t - \text{IG}_{t-1}) + \sigma_1\epsilon_t$. However the dynamic of interstitial glucose level might differ from that of the blood glucose. Further consideration should be made.

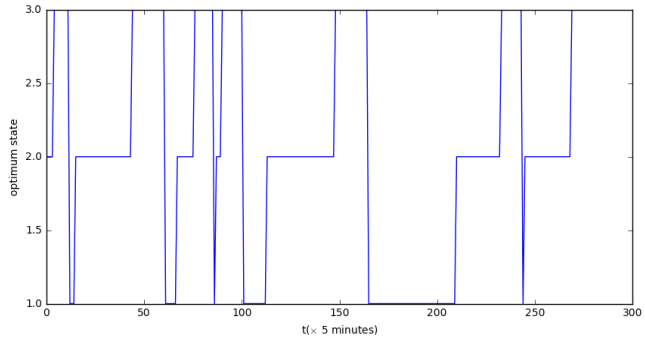
With the limitation of data, we take FS_k directly as BG_t to test performance, while FS_k may still contains error. If more information about the true blood glucose level can be obtained, a better performance comparison can be made. In principle, FS_k should be taken just as a more accurate measurement to be included in the model, as the SSM-based CGM algorithms do.



(a)



(b)



(c)

Figure 5.3.3: MKF filtered results. (a) records the estimated $\widehat{\mathrm{BG}}_t$ with 95% confidence band, compared with CGM, FS and ISIG. (b) records the estimated marginal probability for each state. (c) records the optimal state at each time.

Chapter 6

On-line Bayesian Trend Filtering

6.1 Introduction

A general nonparametric regression model, under a time series setting goes in the form

$$y_t = f_0(z_t) + \epsilon_t,\ t = 1, ..., T, \tag{6.1}$$

where the observation y_t's are generated via the function $f_0 : [0, 1] \to \mathbb{R}$ plus independent Gaussian errors ϵ_t. In most cases, x_t is evenly distributed, i.e., $z_t = t/n$.

A newly proposed nonparametric regression method, l_1 Trend Filtering (Kim et al., 2009), achieves the optimal(minimax) convergence rate (Tibshirani et al., 2014) to the true function by estimating $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, ..., \hat{\beta}_T)$ of $(f_0(z_1), ..., f_0(z_T))$ via a penalized least squares optimization problem,

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^n}{\operatorname{argmin}} \big\{ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{\beta}\|_2^2 + \lambda \|D^{(k)} \boldsymbol{\beta}\|_1 \big\},$$

where λ is the tuning parameter, and $D^{(k)} \in \mathbb{R}^{(T-k) \times T}, k \geq 1$ is the kth order difference operator on β. When $k = 1$,

$$D^{(1)} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \cdots & & & & & \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}.$$

For $k > 1$, we have

$$D^{(k)} = D^{(1)} \times D^{(k-1)} = \{D^{(1)}\}^k.$$

l_1 trend filtering can also be adapted to solve a dynamic regression problem where another series $\boldsymbol{x}_t$ is observed sequentially. The optimization problem then becomes

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^n}{\operatorname{argmin}} \{ \frac{1}{2} \|\boldsymbol{y} - \boldsymbol{\beta}\boldsymbol{x}\|_2^2 + \lambda \|D^{(k)}\boldsymbol{\beta}\|_1 \}, \tag{6.2}$$

where $\boldsymbol{\beta}$ is still of interest, representing the evolving coefficient between the observed $\boldsymbol{y}$ and $\boldsymbol{x}$.

Intuitively, l_1 trend filtering is a generalized LASSO method under a filtering framework. When $k = 1$ and $\boldsymbol{x} \equiv 1$, Equation (6.2) can be specified as

$$\frac{1}{2} \sum_{t=1}^{T} (y_t - \beta_t)^2 + \lambda \sum_{t=1}^{T-1} |\beta_{t+1} - \beta_t|, \tag{6.3}$$

which filters y_t via penalizing the absolute smoothness of β_t. Similar to LASSO, given a relatively large λ, l_1 trend filtering will shrink most of the $|\beta_{t+1} - \beta_t|$'s to 0. Therefore the β series is shown (Kim et al., 2009) to be piecewise constant except for some turning points (also called 'kinked points') where $|\beta_{t_i+1} - \beta_{t_i}| \neq 0$. Likewise, kth order l_1 trend filtering generates a piecewise $(k-1)$th polynomial series of β. Figure 6.1.1 gives an example of k th order l_1 trend filtering, with $k = 1, 2, 3$.

In this chapter we propose a Bayesian corresponding model to l_1 trend filtering. By reformulating (6.2) into a state space representation with a Laplace residual in the state equation, one can recover the optimal $\hat{\boldsymbol{\beta}}$ from the MAP(maximum a posteriori) estimation of that state space model. The benefit from the state space representation is that, with the adoption of sequential Monte Carlo methods, the trend filtering can be conducted on-line. In scenario where observation comes in sequentially, which is very often in econometrics, biology, etc, a batch algorithm like l_1 trend filtering

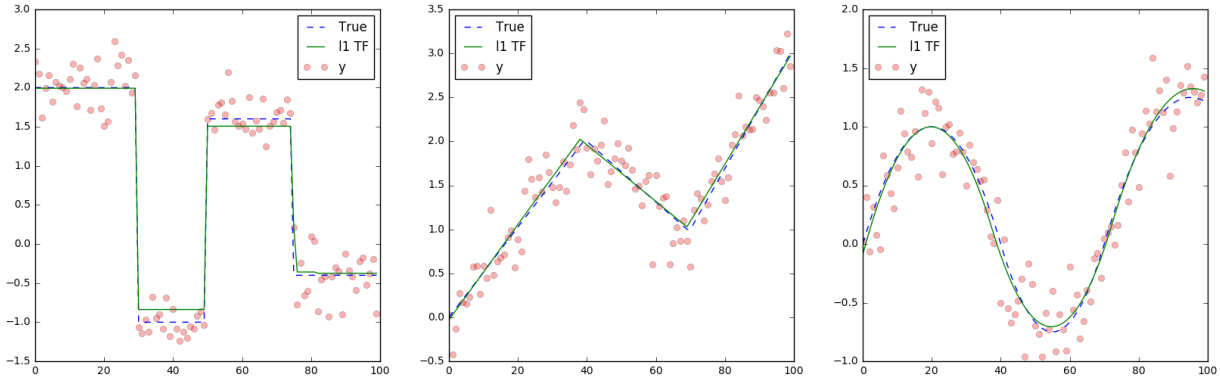


Figure 6.1.1: An example of l_1 trend filtering with $k = 1, 2, 3$ respectively. Data are simulated from piecewise polynomial function plus noise. The true level is included as blue dashed line.

suffers from multiple algorithm calls while the estimation update is minor. The worst-case computational complexity for l_1 trend filtering is $\mathcal{O}(n^{3/2})$, thus leads to an $\mathcal{O}(n^{5/2})$ rolling cost. Our proposed on-line Bayesian trend filtering algorithm would only require $\mathcal{O}(mn)$ time complexity with m the Monte Carlo sample size. It benefits a lot when n is relatively large.

Since l_1 trend filtering essentially finds some limited 'kinked points' where the kth difference of β is non-zero, it is legitimate and sometimes preferable to operate directly on this goal using regime switching state space model. Therefore we also propose a comparable Bayesian trend filtering model with a spike-and-slab state residual, i.e.

$$\{D^{(k)}\boldsymbol{\beta}\}_t = \sigma_{\lambda_t}\eta_t, \tag{6.4}$$

where $\eta_t \sim_{iid} N(0,\ 1)$, $\lambda_t \sim_{iid} Ber(p)$, $\sigma_0 = 0,\ \sigma_1 = \sigma_x$. This prior aims directly at finding the 'kinked points' and thus the MAP estimation of $\boldsymbol{\beta}$ would still be piecewise kth polynomial. In addition, now this conditional dynamic linear model can be solved even more efficiently using Mixture Kalman Filter with a rolling computational cost $\mathcal{O}(mn)$. Since MKF usually requires far less Monte Carlo samples, this model is easier to compute. The computational cost can be further reduced to $\mathcal{O}(Kn)$ by a greedy Viterbi algorithm, where K is some tens number which is smaller

than m.

The rest of this chapter is organized as follows. We introduce the Bayesian trend filtering with spike-and-slab prior and the associated MKF and Viterbi algorithm in section 6.2. The Laplace prior Bayesian trend filtering model, which corresponds to l_1 trend filtering, will be covered in section 6.3. Section 6.4 reports a numerical analysis with simulated data as well as an econometric application. A brief conclusion and discussion is included in 6.5.

6.2 Spike-and-slab trend filtering

Equation (6.3) gives the target function to be minimized for 1st order $l1$ trend filtering. In practice, as shown in Figure 6.1.1, it operates to shrink most of the $|\beta_{t+1} - \beta_t|$'s to zero while leaving only limited non-zero $|\beta_{t_i+1} - \beta_{t_i}|$'s, also known as 'kinked points' at t_i. In other words, 1st order $l1$ trend filtering achieves 'kinked points' selection in a time series setting, as LASSO achieves variable selection under the regression framework. Bayesians (George and McCulloch, 1997) have shown that a spike-and-slab prior can fulfill similar variable selection goals as LASSO while providing a more probabilistic insight than LASSO. In detail, while LASSO minimize the objective function

$$\frac{1}{2}\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i\boldsymbol{\beta})^2 + \lambda\|\boldsymbol{\beta}\|_1,$$

the spike-and-slab regression assumes that $y_i|\boldsymbol{x}_i \sim N(\boldsymbol{x}_i\boldsymbol{\beta}, \sigma_y^2)$ and $\beta_j \sim N(0, \sigma_{\lambda_j}), \lambda_j \sim Ber(p)$, $j = 1, ..., p$. The spike and slab prior standard deviation σ_0 and σ_1 are specified as 0 and σ_b. The variable is only selected when the estimated indicator $\hat{\lambda}_j = 1$.

Following this logic, one can build its time series equivalent under the state space construction. For example, when $k = 1$, the state space representation of spike-and-slab filtering is as follows,

$$\begin{aligned} y_t &= x_t\beta_t + \sigma_y\epsilon_t, \\ \beta_t &= \beta_{t-1} + \sigma_{\lambda_t}\eta_t, \end{aligned} \tag{6.5}$$

where $\epsilon_t,\ \eta_t \sim_{iid} N(0,\ 1)$, $\lambda_t \sim_{iid} Ber(p)$, $\sigma_0 = 0,\ \sigma_1 = \sigma_x$. The 'kinked point' is indicated when $\lambda_t = 1$. For $k > 1$, the state space representation goes as

$$\begin{aligned} y_t &= x_t\beta_t + \sigma_y\epsilon_t, \\ \{D^k\boldsymbol{\beta}\}_t &= \sigma_{\lambda_t}\eta_t, \end{aligned} \tag{6.6}$$

where the residual settings are the same as (6.5) and $\{D^k\boldsymbol{\beta}\}_t$ is the tth element of $D^k\boldsymbol{\beta}$. This can then be further written into a companion form as

$$\begin{aligned} y_t &= G_tB_t + \sigma_y\epsilon_t, \\ B_t &= HB_{t-1} + W_{\lambda_t}w_t, \end{aligned} \tag{6.7}$$

where $B_t = (\beta_t, ..., \beta_{t-k+1})'$, G_t is $k \times 1$ and H is $k \times k$. For example, when $k = 2$, the representation (6.6) becomes

$$\begin{aligned} y_t &= x_t\beta_t + \sigma_y\epsilon_t, \\ \beta_t - \beta_{t-1} &= \beta_{t-1} - \beta_{t-2} + \sigma_{\lambda_t}\eta_t, \end{aligned}$$

and $B_t = (\beta_t, \beta_{t-1})'$, $G_t = (x_t, 0)'$, $H = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}$, $W_{\lambda_t} = \begin{bmatrix} \sigma_{\lambda_t} & 0 \\ 0 & 0 \end{bmatrix}$, $w_t \sim_{iid} N(0, I_2)$ in the companion form (6.7).

With the state space representation, spike-and-slab filtering is even more intuitive than its equivalent in the regression setting and the corresponding l_1 trend filtering: the time series exhibits regime-switching patterns where they follow different polynomial trends between two 'kinked points' when $\lambda_t \neq 0$. Thus the optimal indicator series yields the best 'kinked points' selection and the MAP $\hat{\boldsymbol{\beta}}$ based on the optimal indicator series is therefore piecewise polynomial. While the MAP can be obtained similarly as spike-and-slab regression using MCMC methods based on the batch dataset, state space style on-line filtering methods are preferred for the speed consideration. Given the optimal indicator path $\boldsymbol{\lambda}_t^*$, computing the MAP $\hat{\beta}$ is faster due to the sparsity of $\boldsymbol{\lambda}_t^*$. In

the rest of this section we propose a two-step on-line filtering framework to obtain the $\hat{\beta}$. In the first step, two algorithms are employed to estimate the optimal indicator path $\boldsymbol{\lambda}_t^*$. The second step is simply optimizing based on $\boldsymbol{\lambda}_t^*$.

6.2.1 Delayed MKF with optimal path on $\boldsymbol{\lambda}_t$

Noticing (6.7) is essentially a conditional dynamic linear model, a Δ-steps look-ahead Mixture Kalman Filtering algorithm can be adopted to find the optimal path $\boldsymbol{\lambda}_t^*$ as follows:

(A) At each time $t+1$, for each $j = 1, ..., m$ and then each $\Lambda_{t+1} = i,\ i \in \mathfrak{T}$, run the Δ-steps Kalman Filter to obtain

$$\begin{aligned} v_{t+1,i}^{(j)} \quad &\stackrel{\Delta}{=} P(\Lambda_{t+1} = i | \boldsymbol{\lambda}_t^{(j)}, \boldsymbol{y}_{t+\Delta+1}) \\ &\propto p(\Lambda_{t+1} = i | \boldsymbol{\lambda}_t^{(j)}) \\ &\quad \times \sum_{\lambda_{t+2}^{t+\Delta+1}} [\prod_{\tau=0}^{\Delta} P(y_{t+\tau+1} \mid \boldsymbol{y}_{t+\tau}, \Lambda_{t+1} = i, \boldsymbol{\lambda}_t^{(j)}, \lambda_{t+2}^{t+\tau+1}) \\ &\quad \times \prod_{\tau=1}^{\Delta} P(\lambda_{t+\tau+1} \mid \Lambda_{t+1} = i, \boldsymbol{\lambda}_t^{(j)}, \lambda_{t+2}^{t+\tau})], \end{aligned}$$

where $\lambda_{t+2}^{t+\tau+1}$ denotes the path from $t+2$ to $t+\tau+1$, inclusive.

(B) Sample a $\lambda_{t+1}^{(j)}$ from the set $\mathfrak{T}$, with probability proportional to $v_i^{(j)}$. Let $KF_t^{(j)}$ be the one associated with it,

(C) The incremental weight

$$u_{t+1,i}^{(j)} = \frac{p(y_t | \boldsymbol{y}_{t-1}, \boldsymbol{\lambda}_t^{(j)}) p(\Lambda_t = i | \boldsymbol{\lambda}_{t-1}^{(j)})}{v_{t,i}^{(j)}} \sum_{k \in \mathfrak{T}} v_{t+1,k}^{(j)},$$

and weight is $\omega_{t+1}^{(j)} = \omega_t^{(j)} \times u_{t+1}^{(j)}$. The optimal path $\boldsymbol{\lambda}_{t+1}^*$ is $\boldsymbol{\lambda}_{t+1}^{(j)}$ that has the highest $\omega_{t+1}^{(j)}$.

(D) (Optional)Resample when variance of $\omega_{t+1}^{(j)}$ is large.

Intuitively the above Mixture Kalman Filter looks Δ steps ahead to check if the likelihood is implying a 'jump'. In practice, the prior frequency is relatively low to guarantee the sparsity. Therefore delayed sampling by trial distribution $P(\Lambda_{t+1} = i|\boldsymbol{\lambda}_t^{(j)}, \boldsymbol{y}_{t+\Delta+1})$ helps the on-line 'kinked points' detection.

6.2.2 Delayed top-K greedy Viterbi Algorithm

In the spike-and-slab filtering settings, the indicator series is binary and mostly 0's. Therefore a Monte Carlo sample of size m may end up with heavy duplications. To further utilize the sparsity of the $\boldsymbol{\lambda}_t$ series, we propose the following Δ-steps look-ahead top-K greedy Viterbi Algorithm:

(A) At each time $t+1$, suppose we have the top K paths of $\boldsymbol{\lambda}_t^{(j)}$ that has the highest $P(\boldsymbol{\lambda}_t^{(j)}|\boldsymbol{y}_{t+\Delta})$. We also stored the likelihood $P(\boldsymbol{y}_t|\boldsymbol{\lambda}_t^{(j)})$. For each $j = 1, ..., m$ and then each $\Lambda_{t+1} = i,\ i \in \mathfrak{T}$, run the Δ-steps Kalman Filter to obtain

$$\begin{aligned} p_{t+1,i}^{(j)} \quad &\triangleq P(\boldsymbol{\lambda}_t^{(j)}, \Lambda_{t+1} = i|\boldsymbol{y}_{t+\Delta+1}) \\ &\propto P(\boldsymbol{y}_t|\boldsymbol{\lambda}_t^{(j)}) \times P(\boldsymbol{y}_{t+\Delta+1}|\boldsymbol{\lambda}_t^{(j)}, \Lambda_{t+1} = i, \boldsymbol{y}_t) \times P(\boldsymbol{\lambda}_t, \Lambda_{t+1} = i) \\ &= P(\boldsymbol{y}_t|\boldsymbol{\lambda}_t^{(j)}) \times P(\boldsymbol{\lambda}_t^{(j)}, \Lambda_{t+1} = i) \\ &\quad \times \sum_{\lambda_{t+2}^{t+\Delta+1}} \Big[\prod_{\tau=0}^{\Delta} P(y_{t+\tau+1} \mid \boldsymbol{y}_{t+\tau}, \Lambda_{t+1} = i, \boldsymbol{\lambda}_t^{(j)}, \lambda_{t+2}^{t+\tau+1}) \\ &\quad \times \prod_{\tau=1}^{\Delta} P(\lambda_{t+\tau+1} \mid \Lambda_{t+1} = i, \boldsymbol{\lambda}_t^{(j)}, \lambda_{t+2}^{t+\tau}) \Big], \end{aligned}$$

(B) From the $K \times I$ paths, pick the highest k paths that maximizes $p_{t+1,i}^{(j)}$, along with the necessary KF quantities. The highest one is the optimal path estimation $\boldsymbol{\lambda}_{t+1}^*$.

(C) Update the likelihood

$$P(\boldsymbol{y}_{t+1}|\boldsymbol{\lambda}_{t+1}^{(j)}) = P(\boldsymbol{y}_t|\boldsymbol{\lambda}_t^{(j)}) \times P(y_{t+1}|\boldsymbol{y}_t, \boldsymbol{\lambda}_t^{(j)}, \lambda_{t+1}^{(j)}).$$

The top-K greedy Viterbi algorithm follows the logic of Viterbi algorithm (Forney, 1973) in the sense that it iteratively update the most likely path up to time t in a dynamic programming fashion. Yet with the existence of latent variable $\boldsymbol{\beta}_t$, there's no guarantee that the optimal path is covered among the top-K paths. However with the sparsity assumption where most of the $\boldsymbol{\lambda}_t$'s are 0's, the optimal path building aims to find the best 'kinked points' from a limited candidate pools. Therefore a size-K candidate pool suffices with the sparsity assumption.

Given the optimal path $\boldsymbol{\lambda}_t^*$, the MAP $\hat{\boldsymbol{\beta}}_t$ can be obtained by maximizing the full log-likelihood

$$\log\{p(\boldsymbol{y}_t, \boldsymbol{\beta}_t|\boldsymbol{\lambda}_t^{opt})\} = -\sum_t \frac{(y_t - x_t\beta_t)^2}{2\sigma_y^2} - \sum_t \frac{\{D^k\boldsymbol{\beta}\}_t^2}{2\sigma_{\lambda_t^*}^2}. \tag{6.8}$$

Although the maximization seems to be a batch method, it can be calculated on-line. Suppose $\boldsymbol{\lambda}_{t_i}^* = 1, i = 1, ..., l$, then (6.8) becomes

$$\log\{p(\boldsymbol{y}_t, \boldsymbol{\beta}_t|\boldsymbol{\lambda}_t^{opt})\} = -\sum_{i=1}^{l} \sum_{t\in(t_i,t_{i+1})} \frac{(y_t - x_t\beta_t)^2}{2\sigma_y^2} - \sum_{i=1}^{l} \frac{\{D^k\boldsymbol{\beta}\}_{t_i}^2}{2\sigma_1^2}.$$

Thus this maximization is size-l and the coefficients $\sum_{t\in(t_i,t_{i+1})} y_t$ and $\sum_{t\in(t_i,t_{i+1})} x_t$ can be calculated on-line with a total cost $\mathcal{O}(l^2 n)$ with l relatively small due to the sparsity assumption. Figure 6.2.1 conducts the Spike-and-slab filtering on the same data as Figure 6.1.1 with $\sigma_y = 1,\ p = 0.01$ and $\sigma_1 = 4, 1, 0.1$ respectively when $k = 1, 2, 3$.

6.3 On-line l_1 trend filtering with state space representation

The l_1 trend filtering can be rewritten into state space representation in the following form,

$$\begin{aligned} &y_t|x_t, \beta_t \sim N(x_t\beta_t, \sigma_y^2), \\ &\beta_t|\boldsymbol{\beta}_{t-k}^{t-1} \sim Laplace(f_k(\boldsymbol{\beta}_{t-k}^{t-1}), \sigma_b), \end{aligned} \tag{6.9}$$

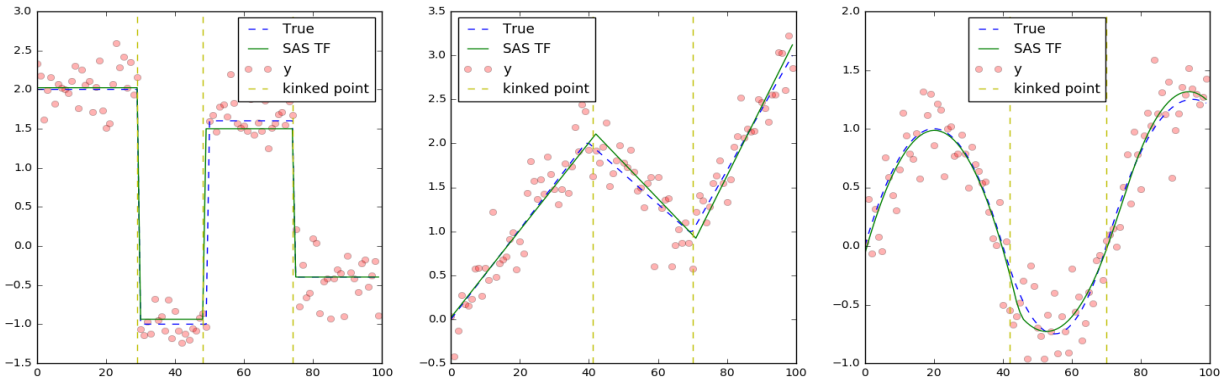


Figure 6.2.1: An example of spike-and-slab trend filtering(SAS TF) with $k = 1, 2, 3$ respectively using MKF with $m = 500$. Simulated data are the same as Figure 6.1.1 with $\boldsymbol{x}_t \equiv 1$. The true level is included as blue dashed line. The vertical yellow dashed lines mark where $\boldsymbol{\lambda}_t^* = 1$.

where $\boldsymbol{\beta}_{t-k}^{t-1} = (\beta_{t-k}, ..., \beta_{t-1})$, and $f_k(\boldsymbol{\beta}_{t-k}^{t-1})$ is a linear function. For example $f_1(\boldsymbol{\beta}_{t-1}^{t-1}) = \beta_{t-1}$ and $f_2(\boldsymbol{\beta}_{t-2}^{t-1}) = 2\beta_{t-1} - \beta_{t-2}$. We have the correspondence $\lambda = 2\sigma_y^2/\sigma_b$,

$$P(\boldsymbol{\beta}_t|\boldsymbol{y}_t) \propto \exp(-\sum_t \frac{(y_t - x_t\beta_t)^2}{2\sigma_y^2} - \frac{\|D^k\boldsymbol{\beta}\|_1}{\sigma_b})$$

Therefore l_1 trend filtering is equivalent to finding the MAP of $\boldsymbol{\beta}_t$ in (6.9).

6.3.1 Annealing method with empirical trial distribution

We here propose an Annealing sequential Monte Carlo method to approach the MAP

$$\hat{\boldsymbol{\beta}}_t = \underset{\boldsymbol{\beta}_t}{\text{argmax}} P(\boldsymbol{\beta}_t|\boldsymbol{y}_t).$$

Denote

$$P^\delta(\boldsymbol{\beta}_t|\boldsymbol{y}_t) \propto \exp(-\delta\sum_t \frac{(y_t - x_t\beta_t)^2}{2\sigma_y^2} - \delta\frac{\|D^k\boldsymbol{\beta}\|_1}{\sigma_b}),$$

then

$$\mathbb{E}_{P^\delta}\boldsymbol{\beta}_t \to \hat{\boldsymbol{\beta}}_t,\ \delta \to \infty \tag{6.10}$$

where $\hat{\boldsymbol{\beta}}_t$ is the MAP which maximizes $P(\boldsymbol{\beta}_t|\boldsymbol{y}_t)$, since P^δ converges to single point density at $\hat{\boldsymbol{\beta}}_t$. An Annealing sequential Monte Carlo algorithm starts with an initial trial distribution $q^{(0)}(\boldsymbol{\beta}_t|\boldsymbol{y}_t)$ and $\delta = 1$. It then repeatedly estimates a new trial distribution $q^{(\delta)}(\boldsymbol{\beta}_t|\boldsymbol{y}_t)$ from the weighted Monte Carlo samples where weight is calculated by $P^\delta(\boldsymbol{\beta}_t|\boldsymbol{y}_t)/q^{\delta-1}(\boldsymbol{\beta}_t|\boldsymbol{y}_t)$. The Monte Carlo samples will get denser as δ increases since q^δ gets sharper. After several iterations, $\hat{\boldsymbol{\beta}}_t$ is just the weighted average of the δth Monte Carlo samples. δ can grow exponentially as $\delta = 1, 2, 4, 8, ...$ until the sampled paths converge.

The key to the above annealing sequential Monte Carlo method is an appropriate initial trial distribution $q^{(0)}(\boldsymbol{\beta}_t|\boldsymbol{y}_t)$, which should approximate the posterior distribution $p(\boldsymbol{\beta}_t|\boldsymbol{y}_t)$ closely. However in practice the normal plus Laplace full distribution is hard to approximate sequentially with a large λ. Figure 6.3.1 shows an example of a one-step full likelihood to be approximated in sequen-

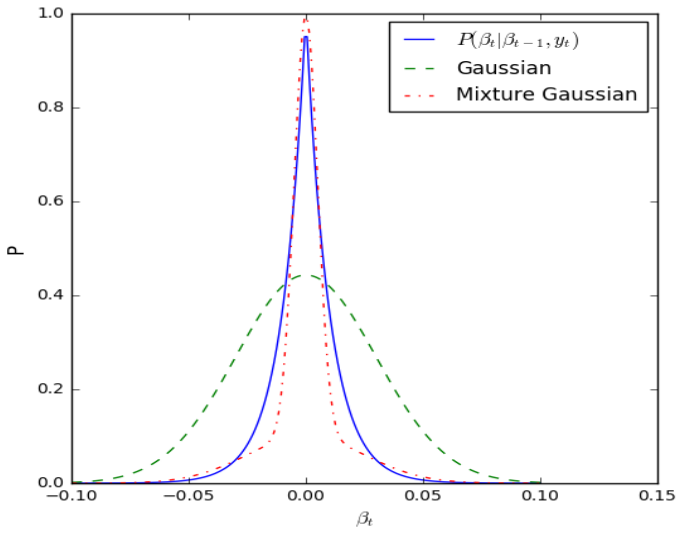


Figure 6.3.1: One-step full likelihood with $\lambda = 100$ and two approximations.

tial Monte Carlo algorithms. While it is impractical to simulate directly from the full likelihood due to its complexity, a simple Gaussian approximation usually fails to approximate it closely. Yet mixture of Gaussian provides a much better approximation since it emphasizes the density around zero as the Laplace part does. On the other hand, it is also important to adaptively increase the non-zero density for those potential 'kinked points'. Otherwise, a general $q^{(0)}$ will stuck in zeros, being reluctant to picking out 'kinked points'. Therefore, we infer an initial empirical distribution $q^{(0)}$ from the Monte Carlo samples $\boldsymbol{\beta}_t^{(j)}$ (up to time t) obtained in the SAS filtering procedure. As discussed in previous section, the samples $\boldsymbol{\beta}_t^{(j)}$ adaptively provides a good approximation to $p(\boldsymbol{\beta}_t|\boldsymbol{y}_t)$, having most of the $q(\beta_t|\beta_{t-1}, y_t)$ zero-dense while some potential 'kinked points'. Then the weight calculation calibrate the difference between the trial distribution and the full distribution.

In summary, the Annealing sequential Monte Carlo method goes like following:

Algorithm 6. *1 At each time t, run Δ-step look-ahead MKF algorithm to get a weighted sample $\beta_{t,0}^{(j)}$ with weight $w_{t,0}^{(j)}$, an implied empirical distribution can be estimated as $q_t^{(0)}(\beta_t|\beta_{t-1})$*

2 For $\delta = 1, 2, 4, 8...$, sample $\beta_{t,k}^{(j)}$ from $q_t^{(\delta-1)}(\beta_t|\beta_{t-1})$, set the cumulative weight

$$u_{t,\delta}^{(j)} = \exp\Big\{ -\frac{k(y_t - (x_t)\beta_t)^2}{2\sigma_y^2} - \frac{k|\beta_t - \beta_{t-1}|}{\sigma_b} \Big\} / q_t^{k-1}(\beta_t|\beta_{t-1}),$$

an implied empirical distribution can be estimated as $q_t^k(\beta_t|\beta_{t-1})$.

3 Repeat step 2 until convergence.

From the implementation's perspective, this procedure can be conducted on-line. At time t, the trial distributions q^δ can be updated one by one for $\delta = 1, 2, 4, ...$, given the stored Monte Carlo samples $\beta_{t-1,\delta}^j$. The total time complexity is $\mathcal{O}(Tm\log(\delta)))$. In practice a $\delta = 16$ or 32 suffices. Thus the time complexity can be treated as $\mathcal{O}(Tm))$.

Figure 6.3.2 shows the comparison between the original and Bayesian on-line $l1$ trend filtering estimations. One can observe almost identical behavior between the two estimations. However, the annealing step adds noise to implementing Monte Carlo sampling in each iteration. Therefore

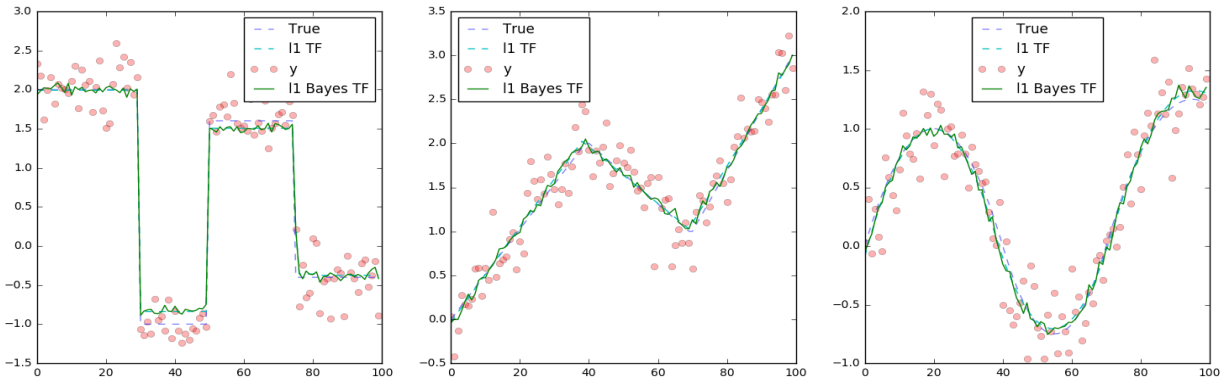


Figure 6.3.2: An example of on-line $l1$ trend filtering with $k = 1, 2, 3$ respectively. Simulated data are the same as Figure 6.1.1 with $\boldsymbol{x}_t \equiv 1$. The true level is included as blue dashed line. The cyan dashed line is the original $l1$ trend filtering. The green line is the Bayesian on-line $l1$ trend filtering estimation. $\delta = 16$.

the filtered $\hat{\boldsymbol{\beta}}_t$ is not strictly flat, but with some noise in vision. Yet the mechanism of annealing sequential Monte Carlo guarantees the convergence between $\hat{\boldsymbol{\beta}}_t$ and the optimal $\hat{\boldsymbol{\beta}}_t^*$.

6.4 Empirical studies

In this section we compare the traditional $l1$ trend filtering with the newly proposed Spike-and-Slab trend filtering(two algorithms) and Bayesian on-line $l1$ trend filtering in different metrics under both simulation and real data scenarios. For simulated data, we consider piecewise polynomial functions as the true value and add different levels of noises. We then apply the four methods with various tuning parameters using whole batch data to study their dependence on hyper-parameters. In the real data example, we introduce an econometric model where $l1$ trend filtering can have application while an on-line filtering algorithm is preferred. Run-time performance as well as predictive results are compared within the mentioned methods.

6.4.1 Simulation results

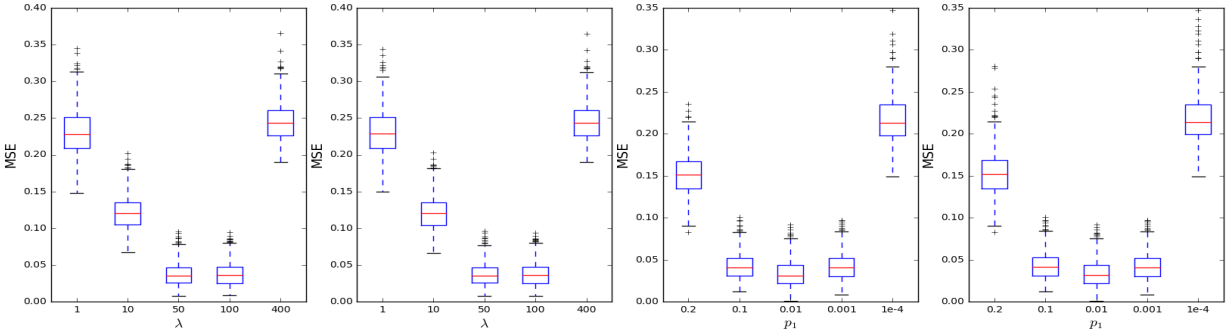
To begin with, we take a simple step function as follows,

$$f_0(t) = \begin{cases} 2, \ t \leq 30; \\ -1, \ 30 < t \leq 50; \\ 1.5, \ 50 < t \leq 75; \\ -0.5, \ 75 < t \leq 100, \end{cases}$$

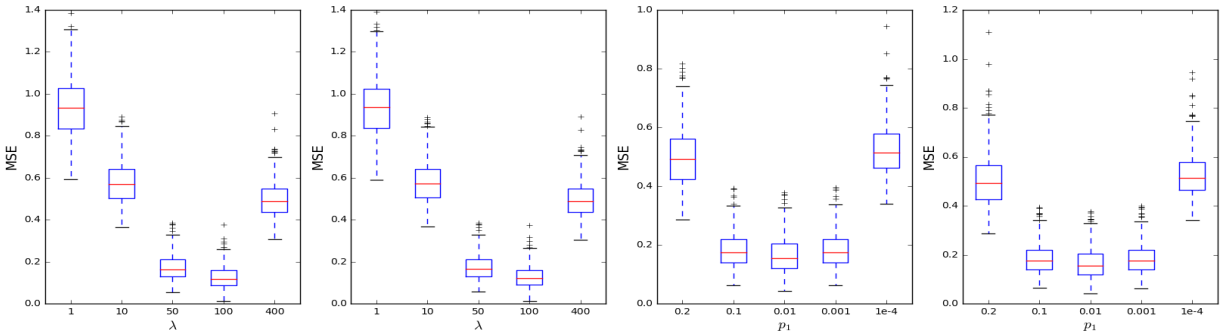
which is also shown in Figure 6.1.1. The observations are simulated following $y_t = f_0(t) + \epsilon_t$, where $\epsilon_t \sim N(0, \sigma^2)$, with $\boldsymbol{x}_t \equiv 1$. We test the performance of the four methods respectively on three levels of noises with $\sigma = \{0.5, 1, 2\}$, each with 500 realizations. For each noise level, the filtering algorithms are conducted with several levels of hyper-parameters. For l_1 and Bayesian l_1 trend filtering, we set the candidate hyper-parameter to be $\lambda \in \{1, 20, 50, 100, 400\}$ while $p_1 \in \{0.2, 0.1, 0.01, 0.001, 10^{-4}\}$ for the two Spike-and-slab trend filtering algorithms.

The boxplots of error metrics MSE $= \frac{1}{100}\sum(f_0(t) - \beta_t)^2$(truncated to adjust for the extreme variance from lagged effect in SAS TF) are recorded in Figure 6.4.1. In the l_1 trend filtering case, $\lambda = 1$ leads to over-fitting while $\lambda = 400$ penalizes the smoothness too much. The prior parameter p_1 in SAS trend filtering also serves similar functionality as λ in l_1 TF, with $p_1 = 0.2$ to over-fit and $p_1 = 10^{-4}$ beging too strong restriction. Overall the four methods have very similar performance. The left two share almost same boxplots since Bayesian on-line l_1 TF has the same target function as l_1 TF. The SAS TF has comparable MSE levels as l_1 TF. However it is more hyper-parameter indifferent in the sense that a wide range of p_1 can generate a reasonably good estimation while in l_1 TF one needs to be more careful in specifying λ. There's negligible differences between Mixture Kalman Filter and Top-K greedy Viterbi Algorithm, except for very rare cases in which the greedy Viterbi Algorithm fails to find the optimum path, thus leads to the extreme MSEs.

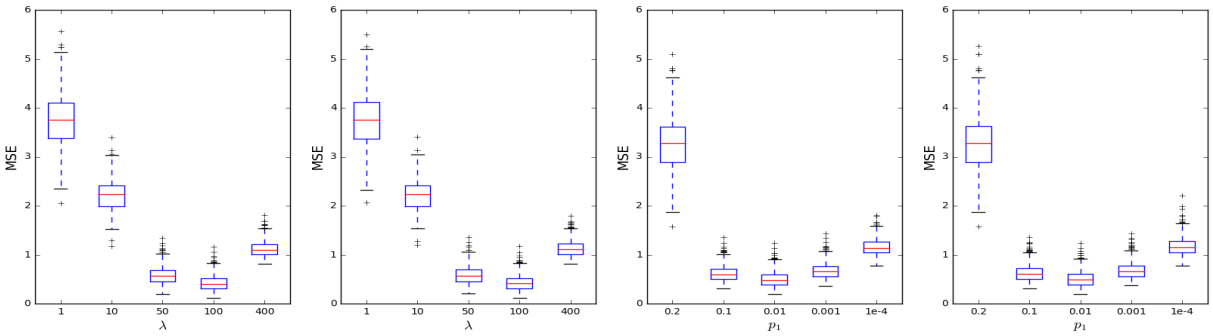
Then the same experiment was conducted on the first and second order case with piecewise



(a) $\sigma = 0.5$



(b) $\sigma = 1$



(c) $\sigma = 2$

Figure 6.4.1: Boxplots of MSE's on 500 simulations under different methods. From left to right are l_1 trend filtering, Bayesian on-line l_1 trend filtering and Spike-and-slab trend filtering type I and II. In Bayesian on-line $l1$ trend filtering, $m = 2000$ and $\delta = 16$. In SAS trend filtering, $\sigma_b = 2$ and $\sigma_y = 0.5, 1, 2$ respectively. $m = 500$ for type I and $K = 20$ for type II.

linear and quadratic functions as follows:

$$f_1(t) = \begin{cases} \frac{t}{20},\ t \leq 40; \\ -\frac{t}{30} + \frac{10}{3},\ 40 < t \leq 70; \\ \frac{t}{15} - \frac{11}{3},\ 70 < t \leq 100, \end{cases}$$

$$f_2(t) = \begin{cases} 1 - \left(\frac{t-20}{20}\right)^2,\ t \leq 40; \\ 0.75 \times \left\{\left(\frac{t-55}{15}\right)^2 - 1\right\},\ 40 < t \leq 70; \\ 1.25 - 1.25 \times \left(\frac{t-95}{25}\right)^2,\ 70 < t \leq 100. \end{cases}$$

f_1 and f_2 corresponds to the true levels in the second and third figures in Figure 6.1.1. The four methods are then applied to observations with independent noise levels with $\sigma = \{0.5, 1, 2\}$, 500 realizations each. We directly take the best hyper-parameter out of grid search and record the MSE's in Table 6.4.1(the case k=1 corresponds to Figure 6.4.1). In low noise cases, the SAS trend filtering outperforms l_1 trend filtering due to its flexibility in filtering after finding the 'kinked points'. However, with increasing noise level, finding turning points on-line becomes more difficult, which leads to a higher MSE of SAS TF algorithm than l_1 TF. Within the l_1 TF class, the MSE of Bayesian l_1 TF is slightly higher than l_1 TF estimation due to the rigidness of Bayesian l_1 TF estimations. Similarly, within SAS TF class, the greedy Viterbi algorithm has slightly higher MSE due to some extreme cases where the top-K paths fails to maintain the optimal.

6.4.2 Real data–Conditional beta in CAPM model

Conditional beta has been proven necessary in explaining the stock dynamic in the asset pricing literature.The general form of conditional CAPM model goes like below

$$R_{i,t} = \alpha_{i,t} + \beta_{i,t} R_{m,t} + \epsilon_{i,t}$$

k=1	l_1 TF ($\lambda = 100$)		Bayesian l_1 TF ($\lambda = 100$)		SAS TF-I ($p_1 = 0.01$)		SAS TF-II ($p_1 = 0.01$)	
	mean	std	mean	std	mean	std	mean	std
$\sigma = 0.5$	0.0374	0.0160	0.0378	0.0161	0.0335	0.0158	0.0338	0.0159
$\sigma = 1$	0.172	0.0591	0.174	0.0602	0.163	0.0594	0.165	0.0598
$\sigma = 2$	0.422	0.160	0.430	0.164	0.503	0.163	0.505	0.164
k=2	l_1 TF ($\lambda = 1000$)		Bayesian l_1 TF ($\lambda = 1000$)		SAS TF-I ($p_1 = 0.01$)		SAS TF-II ($p_1 = 0.01$)	
	mean	std	mean	std	mean	std	mean	std
$\sigma = 0.5$	0.0150	0.00792	0.0155	0.00801	0.0138	0.00753	0.0140	0.00754
$\sigma = 1$	0.0968	0.0409	0.0970	0.0406	0.0932	0.0430	0.0934	0.0433
$\sigma = 2$	0.243	0.0920	0.250	0.0975	0.263	0.105	0.267	0.107
k=3	l_1 TF ($\lambda = 5000$)		Bayesian l_1 TF ($\lambda = 5000$)		SAS TF-I ($p_1 = 0.01$)		SAS TF-II ($p_1 = 0.01$)	
	mean	std	mean	std	mean	std	mean	std
$\sigma = 0.5$	0.0165	0.00841	0.0167	0.00844	0.0160	0.00812	0.0162	0.00813
$\sigma = 1$	0.0878	0.0431	0.0881	0.0435	0.0846	0.0419	0.0848	0.0422
$\sigma = 2$	0.234	0.119	0.239	0.121	0.260	0.137	0.263	0.139

Table 6.4.1: Mean and standard deviation of best MSE's for different trend filtering methods under various combinations of original functions and noise levels.

where$R_{i,t} = r_{i,t} - r_{f,t}$ is the risk premium return of the stock, $r_{f,t}$ is the risk free rate, $R_{m,t} = r_{m,t} - r_{f,t}$ is the market risk premium. $\beta_{i,t}$ explains the risk exposure a single stock has to the market-a higher return corresponds to higher risk exposure.

One strand of research in estimating and interpreting conditional CAPM is Bali et al. (2009), which uses intra-month daily return to estimate monthly beta, i.e.

$$R_{i,d,t} = \alpha_{i,t} + \beta_{i,t} R_{m,d,t} + \epsilon_{i,d,t}$$

where t is the month index and d is the day index. For each single stock in every month, a regression model is fitted to get $\hat{\beta_{i,t}}$. After that a classical time series model is fitted on $\hat{\beta_{i,t}}$ with respect to t. For example an AR(1) process where

$$\beta_{i,t} = \mu + s\beta_{i,t-1} + \eta_{i,t},$$

is employed as a tool to predict $\hat{\beta_{i,t+1}}$ given all the information up to time month t. Compared to the classical CAPM model where monthly return is the main sourse to make beta inference, Bali's method utilize the daily stock return which enlarges the data resolution from a statistical perspective. But the drawback of this intra-monthly fitted beta is that it could be too noisy to produce a stable prediction. Thus adding a dynamic on beta to stabilize the prediction makes the model more preferable to the classical CAPM model.

However this method has its drawback in the sense that it is a two-step method–the time series fitting takes the estimated beta as observation and makes prediction on top of the pre-estimated betas. Moreover, the choice of period of daily stock return is fixed to be one month while the betas dynamic might not strictly follow a monthly pattern. While still utilizing the daily return series, we propose a piecewise-constant conditional beta dynamic follows

$$R_{i,d} = \alpha_i + \beta_{i,d} R_{m,d} + \epsilon_{i,d},$$
$$\beta_{i,d} = \beta_{i,d} + \eta_{i,d},$$

where $\epsilon_{i,d}$ are still normal residuals while $\eta_{i,d}$ are Laplace or Spike-and-slab residuals. The idea is to further utilize the daily return to automatically determine when to shift the beta and alpha level. With the l_1 or spike and slab penalty, the smoothing is automatically fulfilled thus further time series prediction is no longer needed.

6.4.2.1 Data and Results

The data we have are the whole stock data (from DataStream) from 1980.01.01 - 2016.12.31 including 12807 stock ID's. Due to stock enlist/unlist, on a single day, there are only around 3000 stocks available. The associated market return and risk-free return are also included. Four different models are applied to predict the next months' betas for each single stock:

I Apply the intra-month regression on daily stock return to get monthly observed beta. Fit the monthly-observed beta series with an AR(1) model in a rolling basis with lookback period

equal to 60 months to get the predicted beta.

II At the last day of each month, apply an l_1 trend filtering model using the time series tracing back 120 days. Take the last day's beta as the prediction of next month. The tuning parameter of the l_1 trend filtering is fixed at $\lambda = 100$.

III Run an on-line Spike-and-slab trend filtering algorithm using greedy Viterbi Algorithm with $k = 100$. For each month, take the last day's beta as the prediction of next month. The variance parameters are updated yearly and the probability parameters are pre-specified as $p_1 = 0.01$.

IV Run a Bayesian on-line l_1 trend filtering algorithm, which refreshes yearly, with $\lambda = 100$. The Annealing algorithm runs by $m = 2000$ and $\delta = 16$. For each month, take the last day's beta as the prediction of next month.

The amortized computational time per stock per month is recorded in Table 6.4.2 In each month

Model	I	II	III	IV
Amortized Runtime/ms	0.8528	4.591	0.1369	13.72

Table 6.4.2: Amortimzed computational time per stock per month for each model.

a 20 points regression and 60 points AR fit is needed for model I. The computational cost is thus moderate, mainly from AR fits. Although there's only 120 data points long l_1 trend filtering, which in principle takes only linear time, model II consumes more than model I. Model III is most time efficient since it only runs forward 20 steps per month with a relatively small k. Model IV is most time consuming in this case due to the multiplication from m and $\log(\delta)$. However its on-line property would be more advantageous if the rolling frequency is higher. For example, if a daily beta prediction is needed, both model I and II needs 20 times more runtime while model III and IV stays invariant to this change.

A monthly predicted beta series by stock is generated by each model. A cross sectional portfolio-wise comparison is conducted to test the predictive performance in the following way.

For each month, the stocks are grouped into 10 portfolios according to their rankings of predicted betas. Then the excess return of the portfolio is calculated to be the equally-weighted average of its covering stocks. Table 6.4.3 records the average excess return of the portfolios with the associated t statistics. According to classical CAPM theory, a high-beta stock is expected to have higher

Portfolio	Model I		Model II		Model III		Model IV	
	ExRet	t-stat	ExRet	t-stat	ExRet	t-stat	ExRet	t-stat
1(Low β)	0.96906	2.91184	0.909879	2.76661	0.954353	2.90829	0.939146	2.81359
2	0.526947	2.18122	0.559714	2.29602	0.531982	2.14402	0.57844	2.38901
3	0.782335	3.21431	0.778728	3.11595	0.7737	3.13909	0.769304	3.2084
4	0.874882	3.36867	0.66223	2.58902	0.798438	3.06969	0.725961	2.72817
5	0.933109	3.35959	0.890967	3.25586	0.937662	3.5292	0.91701	3.47295
6	1.02805	3.53906	0.947417	3.23868	0.964801	3.28126	0.944437	3.35897
7	1.09133	3.47708	1.09043	3.44471	1.06623	3.28797	1.08483	3.26455
8	1.0715	3.08341	1.05387	2.95146	1.01918	2.94474	1.03944	3.03203
9	1.34523	3.22055	1.26276	2.98852	1.27933	3.02787	1.23343	3.1412
10(High β)	1.78069	3.05721	1.78728	3.15944	1.88794	3.339	1.82304	3.20251
10 - 1	0.811635	2.03268	0.877405	2.17964	0.93359	2.35604	0.883892	2.22923

Table 6.4.3: Comparison of mean excess return between high and low β portforlios, associated with t-stat. The last row records the difference between the highest and lowest beta portfolios.

return due to larger market exposure. So a beta predictor should have predictive power in excess return, which leads to a diversified return profile among the 10 portfolios. The t-stat between the highest and lowest portfolio is also used to reflect the predictive power. One can observe from Table 6.4.3 that all three trend filtering models have a decent beta predictive power, outperforming Bali's method if measured by the t-statistics of highest-lowest difference.

6.5 Discussions

In this chapter we discussed two trend filtering models out of state space representation, both of which have similar property as l_1 trend filtering. With the implementation of sequential Monte

Carlo methods as well as a greedy Viterbi algorithm, both trend filtering models can operate on-line rather than just on batch data. We then compare the proposed two methods with the original l_1 trend filtering on various simulated datasets to check the validity of state space representation. To better emphasize the two models' improvement in on-line trend filtering, we introduced a real world econometrics topic where on-line trend filtering can be applied. The econometric example shows the competence of trend filtering as well as the efficiency of our proposed models.

Other than the discussed Spike-and-slab state equation or Laplace residual distribution, one can construct other versions of Bayesian Trend Filtering using different forms of state distribution, i.e.,

$$\{D^{(k)}\boldsymbol{\beta}\}_t \sim f_t(\theta_t),$$

and still apply sequential Monte Carlo methods to do on-line filtering. A possible choice would be $f_t(\theta_t) = N(0, \sigma_t^2)$ where σ_t^2 follows some prior distribution. This hierarchical structure allows the filter to adaptively shrink the σ_t's according to the data. However more considerations should be taken in picking appropriate trial distributions.

Bibliography

Akaike, H. (1998). Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, pages 199–213. Springer.

Akaike, H., Petrov, B., and Csaki, F. (1973). Information theory and an extension of the maximum likelihood principle.

Al-Anaswah, N. and Wilfling, B. (2011). Identification of speculative bubbles using state-space models with markov-switching. *Journal of Banking & Finance*, 35(5):1073–1086.

Andrieu, C., Doucet, A., Singh, S. S., and Tadic, V. B. (2004). Particle methods for change detection, system identification, and control. *Proceedings of the IEEE*, 92(3):423–438.

Ang, A. and Bekaert, G. (2002). Regime switches in interest rates. *Journal of Business & Economic Statistics*, 20(2):163–182.

Aoki, M. (2013). *State space modeling of time series*. Springer Science & Business Media.

Avitzour, D. (1995). Stochastic simulation bayesian approach to multitarget tracking. *IEE proceedings. Radar, sonar and navigation*, 142(2):41–44.

Bali, T. G., Cakici, N., and Tang, Y. (2009). The conditional beta and the cross-section of expected returns. *Financial Management*, 38(1):103–137.

Barlevy, G. (2007). Economic theory and asset bubbles. *Economic Perspectives*, 31(3).

Berzuini, C., Best, N., Gilks, W., and Larizza, C. (1997). Dynamic conditional independence models and markov chain Monte Carlo methods,. *J. Amer. Statist. Assoc*, 92:1403–1412.

Bliss, M. (2013). *The discovery of insulin*. University of Chicago Press.

Broadie, M., Cicek, D., and Zeevi, A. (2011). General bounds and finite-time improvement for the Kiefer-Wolfowitz stochastic approximation algorithm. *Operations Research*, pages 1211 – 1224.

Brockwell, P. J. and Davis, R. A. (2013). *Time series: theory and methods*. Springer Science & Business Media.

Broyden, C. G. (1970). The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1):76–90.

Brunnermeier, M. K. (2009). Deciphering the liquidity and credit crunch 2007–2008. *Journal of Economic Perspectives*, 23(1):77–100.

Campbell, J. Y. and Shiller, R. J. (1988). The dividend-price ratio and expectations of future dividends and discount factors. *Review of Financial Studies*, 1(3):195–228.

Cappé, O., Moulines, E., and Rydén, T. (2009). Inference in hidden markov models. In *Proceedings of EUSFLAT Conference*, pages 14–16.

Carlin, B. P., Polson, N. G., and Stoffer, D. S. (1992). A Monte Carlo approach to nonnormal and nonlinear state-space modeling. *Journal of the American Statistical Association*, 87:493–500.

Centers for Disease Control and Prevention and others (2011). National diabetes fact sheet: national estimates and general information on diabetes and prediabetes in the united states, 2011. *Atlanta, GA: US Department of Health and Human Services, Centers for Disease Control and Prevention*, 201(1).

Chan, B. L., Doucet, A., and Tadić, V. B. (2003). Optimisation of particle filters using simultaneous perturbation stochastic approximation. *Proc. IEEE ICASSP*, pages 681–684.

Chen, R. (2005). Sequential Monte Carlo methods and their applications. *IMS Lecture Notes Series, Markov Chain Monte Carlo*, 7:147–182.

Chen, R. and Liu, J. S. (2000). Mixture Kalman filters. *Journal of Royal Statistical Society B*, 62:493–508.

Chen, R., Wang, X., and Liu, J. (2000). Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering. *IEEE Transaction on Information Theory*, 46(6):2079–2094.

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

Coquelin, P.-A., Deguest, R., and Munos, R. (2009). Sensitivity analysis in hmms with application to likelihood maximization. In *Advances in Neural Information Processing Systems*, pages 387–395.

Davidon, W. C. (1991). Variable metric method for minimization. *SIAM Journal on Optimization*, 1(1):1–17.

Del Moral, P. (2004). Feynman-kac formulae. In *Feynman-Kac Formulae*, pages 47–93. Springer.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Diba, B. T. and Grossman, H. I. (1988). Explosive rational bubbles in stock prices? *American Economic Review*, 78(3):520–530.

Dicker, L. H., Sun, T., Zhang, C.-H., Keenan, D. B., and Shepp, L. (2013). Continuous blood glucose monitoring: a bayes-hidden markov approach. *Statistica Sinica*, pages 1595–1627.

Diebold, F. X. and Li, C. (2006). Forecasting the term structure of government bond yields. *Journal of econometrics*, 130(2):337–364.

Douc, R., Moulines, E., Rydén, T., et al. (2004). Asymptotic properties of the maximum likelihood estimator in autoregressive models with markov regime. *The Annals of statistics*, 32(5):2254–2304.

Doucet, A. (1998). On sequential simulation-based methods for bayesian filtering. *Technical report* TR.310, Department of Engineering, University of Cambridge.

Doucet, A., De Freitas, N., and Gordon, N. (2001). An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer.

Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3.

Durbin, J. and Koopman, S. J. (2012). *Time series analysis by state space methods*, volume 38. OUP Oxford.

Evans, G. W. (1991). Pitfalls in testing for explosive bubbles in asset prices. *American Economic Review*, 81(4):922–930.

Flood, R. P. and Garber, P. M. (1983). A model of stochastic process switching. *Econometrica*, pages 537–551.

Flood, R. P. and Hodrick, R. J. (1990). On testing for speculative bubbles. *Journal of Economic Perspectives*, 4(2):85–101.

Fong, W., Godsill, S. J., Doucet, A., and West, M. (2002). Monte Carlo smoothing with application to audio signal enhancement. *IEEE transactions on signal processing*, 50(2):438–449.

Forney, G. D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Gelb, A. (1974). *Applied optimal estimation*. MIT press.

George, E. I. and McCulloch, R. E. (1997). Approaches for bayesian variable selection. *Statistica sinica*, pages 339–373.

Godsill, S. J., Doucet, A., and West, M. (2012). Monte Carlo smoothing for nonlinear time series. *Journal of the american statistical association*.

Gordon, N. J., Salmond, D. J., and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F-Radar and Signal Processing*, volume 140, pages 107–113. IET.

Hamilton, J. D. and Whiteman, C. H. (1985). The observable implications of self-fulfilling expectations. *Journal of Monetary Economics*, 16(3):353–373.

Harrison, J. and West, M. (1999). *Bayesian Forecasting & Dynamic Models*. Springer.

Harvey, A. C. (1990). *Forecasting, structural time series models and the Kalman filter*. Cambridge university press.

Heller, A. and Feldman, B. (2008). Electrochemical glucose sensors and their applications in diabetes management. *Chemical reviews*, 108(7):2482–2505.

Hirsch, I. B., Abelseth, J., Bode, B. W., Fischer, J. S., Kaufman, F. R., Mastrototaro, J., Parkin, C. G., Wolpert, H. A., and Buckingham, B. A. (2008). Sensor-augmented insulin pump therapy: results of the first randomized treat-to-target study. *Diabetes technology & therapeutics*, 10(5):377–383.

Hürzeler, M. and Künsch, H. (1995). Monte Carlo approximations for general state space models. Research Report 73, ETH, Zürich.

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

Hürzeler, M. and Künsch, H. R. (2001). Approximating and maximising the likelihood for a general state-space model. In *Sequential Monte Carlo methods in practice*, pages 159–175. Springer.

Ionides, E. L., Bhadra, A., Atchadé, Y., King, A., et al. (2011). Iterated filtering. *The Annals of Statistics*, 39(3):1776–1802.

Ionides, E. L., Bretó, C., and King, A. (2006). Inference for nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 103(49):18438–18443.

Julier, S. J. and Uhlmann, J. K. (1997). New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–194. International Society for Optics and Photonics.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *J. Basic Engineering*, 82:35–45.

Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(1):95–108.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., Chopin, N., et al. (2015). On particle methods for parameter estimation in state-space models. *Statistical science*, 30(3):328–351.

Kantas, N., Doucet, A., Singh, S. S., and Maciejowski, J. M. (2009). An overview of sequential Monte Carlo methods for parameter estimation in general state-space models. In *15th IFAC Symposium on System Identification (SYSID), Saint-Malo, France.(invited paper)*, volume 102, page 117.

Khan, A. I., Vasquez, Y., Gray, J., Wians Jr, F. H., and Kroll, M. H. (2006). The variability of results between point-of-care testing glucose meters and the central laboratory analyzer. *Archives of pathology & laboratory medicine*, 130(10):1527–1532.

Kiefer, J. and Wolfowitz, J. (1952). Stochastic estimation of a regression function. *Ann. of Math. Stat*, 33:462–466.

Kim, C.-J., Nelson, C. R., et al. (1999). State-space models with regime switching: classical and gibbs-sampling approaches with applications. *MIT Press Books*, 1.

Kim, S.-J., Koh, K., Boyd, S., and Gorinevsky, D. (2009). \ell_1 trend filtering. *SIAM review*, 51(2):339–360.

Kindleberger, C. P. (2000). Manias, panics, and crashes: a history of financial crises. *The Scriblerian and the Kit-Cats*, 32(2):379.

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

Kitagawa, G. and Gersch, W. (1996). Linear gaussian state space modeling. *Smoothness Priors Analysis of Time Series*, pages 55–65.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288.

Kovatchev, B., Anderson, S., Heinemann, L., and Clarke, W. (2008). Comparison of the numerical and clinical accuracy of four continuous glucose monitors. *Diabetes care*, 31(6):1160–1164.

Kushner, H. and Yin, G. G. (2003). *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media.

LeRoy, S. F. and Porter, R. D. (1981). The present-value relation: Tests based on implied variance bounds. *Econometrica*, pages 555–574.

Lin, M. T., Zhang, J. L., Cheng, Q., and Chen, R. (2005). Independent particle filters. *Journal of the American Statistical Association*, 100(472):1412–1421.

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

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

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer Science & Business Media.

Mahoney, J. and Ellison, J. (2007). Assessing the quality of glucose monitor studies: a critical evaluation of published reports. *Clinical chemistry*, 53(6):1122–1128.

Malik, S. and Pitt, M. K. (2011). Particle filters for continuous likelihood evaluation and maximisation. *J. Econometrics*, 165:190209.

Marshall, A. W. (1954). The use of multi-stage sampling schemes in Monte Carlo computations. Technical report, RAND CORP SANTA MONICA CALIF.

Petris, G., Petrone, S., and Campagnoli, P. (2009). Dynamic linear models. In *Dynamic Linear Models with R*, pages 31–84. Springer.

Phillips, P. C., Shi, S., and Yu, J. (2015). Testing for multiple bubbles: Limit theory of real-time detectors. *International Economic Review*, 56(4):1079–1134.

Phillips, P. C., Wu, Y., and Yu, J. (2011). Explosive behavior in the 1990s nasdaq: When did exuberance escalate asset values? *International Economic Review*, 52(1):201–226.

Phillips, P. C. and Yu, J. (2011). Dating the timeline of financial bubbles during the subprime crisis. *Quantitative Economics*, 2(3):455–491.

Pitt, M. K. (2002). Smooth particle filters for likelihood evaluation and maximisation.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599.

Posada, D. and Buckley, T. R. (2004). Model selection and model averaging in phylogenetics: advantages of akaike information criterion and bayesian approaches over likelihood ratio tests. *Systematic biology*, 53(5):793–808.

Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80.

Poyiadjis, G., Singh, S., and Doucet, A. (2003). Gradient-free maximum likelihood parameter estimation with particle filters. *In Proc. Am. Control Conf*, page 30623067.

Prado, R. and West, M. (2010). *Time series: modeling, computation, and inference*. CRC Press.

Rauch, H. E., Striebel, C., and Tung, F. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Ann. Math. Statist.*, 22:400–407.

Robert, C. P. (2004). *Monte carlo methods*. Wiley Online Library.

Rother, K. I. (2007). Diabetes treatmentbridging the divide. *The New England journal of medicine*, 356(15):1499.

Rubinstein, R. (1981). *Simulation and the Monte Carlo Method*. Wiley.

Sakamoto, Y., Ishiguro, M., and Kitagawa, G. (1986). Akaike information criterion statistics. *Dordrecht, The Netherlands: D. Reidel*, page 81.

Sandmann, G. and Koopman, S. J. (1998). Estimation of stochastic volatility models via Monte Carlo maximum likelihood. *Journal of Econometrics*, 87(2):271–301.

Santos, M. S. and Woodford, M. (1997). Rational asset pricing bubbles. *Econometrica*, pages 19–57.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

Shephard, N. (1994). Partial non-gaussian state space. *Biometrika*, 81(1):115–131.

Shiller, R. J. (1981). The use of volatility measures in assessing market efficiency. *Journal of Finance*, 36(2):291–304.

Sorenson, H. W. and Alspach, D. L. (1971). Recursive bayesian estimation using gaussian sums. *Automatica*, 7(4):465–479.

Spall, J. C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37:332–341.

Steil, G., Rebrin, K., Hariri, F., Jinagonda, S., Tadros, S., Darwin, C., and Saad, M. (2005). Interstitial fluid glucose dynamics during insulin-induced hypoglycaemia. *Diabetologia*, 48(9):1833–1840.

Taylor, M. P. and Peel, D. A. (1998). Periodically collapsing stock price bubbles: a robust test. *Economics Letters*, 61(2):221–228.

Tibshirani, R. J. et al. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323.

Van Norden, S. and Schaller, H. (1993). The predictability of stock market regime: evidence from the toronto stock exchange. *Review of Economics and Statistics*, pages 505–510.

Wang, J. (2008). Electrochemical glucose biosensors. *Chemical reviews*, 108(2):814–825.

Weinzimer, S. A., Steil, G. M., Swan, K. L., Dziura, J., Kurtz, N., and Tamborlane, W. V. (2008). Fully automated closed-loop insulin delivery versus semiautomated hybrid control in pediatric patients with type 1 diabetes using an artificial pancreas. *Diabetes care*, 31(5):934–939.

West, K. D. (1986). A specification test for speculative bubbles.

West, M. and Harrison, J. (1998). Bayesian forecasting and dynamic models (2nd edn). *Journal of the Operational Research Society*, 49(2):179–179.

World Health Organization and others (2014). *About diabetes*. World Health Organization.

World Health Organization and others (2016). *Global report on diabetes*. World Health Organization.

Wu, G. and Xiao, Z. (2002). Are there speculative bubbles in stock markets? evidence from an alternative approach. Technical report, mimeo.

Wu, Y. (1995). Are there rational bubbles in foreign exchange markets? evidence from an alternative test. *Journal of International Money and Finance*, 14(1):27–46.

Wu, Y. (1997). Rational bubbles in the stock market: accounting for the us stock-price volatility. *Economic Inquiry*, 35(2):309.

Xiang, J. and Zhu, X. (2013). A Regime-Switching Nelson–Siegel term structure model and interest rate forecasts. *Journal of Financial Econometrics*, page nbs021.

Zhang, X., Graepel, T., and Herbrich, R. (2010). Bayesian online learning for multi-label and multi-variate performance measures. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 956–963.