

FINANCIAL TIME SERIES VOLATILITY ANALYSIS USING GAUSSIAN PROCESS STATE-SPACE MODELS

by

Jianan Han

Bachelor of Engineering, Hebei Normal University, China, 2010

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, 2015

©Jianan Han 2015

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my dissertation may be made electronically available to the public.

Financial Time Series Volatility Analysis Using Gaussian Process State-Space Models

Master of Applied Science 2015

Jianan Han

Electrical and Computer Engineering

Ryerson University

Abstract

In this thesis, we propose a novel nonparametric modeling framework for financial time series data analysis, and we apply the framework to the problem of time varying volatility modeling. Existing parametric models have a rigid transition function form and they often have over-fitting problems when model parameters are estimated using maximum likelihood methods. These drawbacks effect the models' forecast performance. To solve this problem, we take Bayesian nonparametric modeling approach. By adding Gaussian process prior to the hidden state transition process, we extend the standard state-space model to a Gaussian process state-space model. We introduce our Gaussian process regression stochastic volatility (GPRSV) model. Instead of using maximum likelihood methods, we use Monte Carlo inference algorithms. Both online particle filter and offline particle Markov chain Monte Carlo methods are studied to learn the proposed model. We demonstrate our model and inference methods with both simulated and empirical financial data.

Acknowledgements

I sincerely thank all the people who have helped and supported me during my graduate study at Ryerson University. Without their help I would never have been able to complete this thesis.

First and foremost I would like to express my deepest gratitude to my supervisor Dr. Xiao-Ping Zhang, whose expertise and enthusiasm for research had set an excellent example for me. I appreciate all his inspiration, understanding, and patience. Thank you for providing me such a great research atmosphere.

Also I am really grateful to Dr. Alagan Anpalagan, Dr. Bobby Ma and Dr. Karthi Umapathy. Thank you for the time, efforts, and contributions to this work.

It has been a pleasure working with all my colleagues in Communications and Signal Processing Applications Laboratory (CASPAL), who are always willing to discuss and share their knowledge with me. A very special thanks goes to Dr. Zhicheng Wei, without whose motivation and encouragement I would not have considered a graduate study in Electrical Engineering.

Finally, my heartfelt thanks to my parents for their continuous support, understanding and encouragement.

Contents

<i>Declaration</i>	ii
<i>Abstract</i>	iii
<i>Acknowledgements</i>	iv
<i>List of Tables</i>	xii
<i>List of Figures</i>	xiii
1 Introduction	1
1.1 Motivation and Objective	1
1.2 Volatility Modeling Literature Review	3
1.2.1 GARCH Models	4
1.2.2 Stochastic Volatility Models	6
1.2.3 Alternative Approaches	7
1.3 Main Contributions	8
1.4 Organization of Thesis	9
2 Background	11
2.1 Financial Time Series Data	11
2.2 Volatility Modeling Preliminaries	12
2.3 Bayesian Nonparametric Framework	15
2.4 Gaussian Process	16
2.4.1 Gaussian Process Regression	17
2.4.2 GP for Time Series	19
2.5 State-Space Models	20
2.6 Chapter Summary	21

3	Gaussian Process Regression Stochastic Volatility Model	23
3.1	GPRSV Models Framework	23
3.2	GP-SSM for Financial Time Series Modeling	25
3.3	Model Structure	27
3.4	Model Building Process	28
3.5	GPRSV with Exogenous Factors	34
3.6	Chapter Summary	36
4	GPRSV Models Inference	37
4.1	Bayesian Inference for State-Space Models	37
4.2	Monte Carlo Methods	39
4.2.1	Sequential Monte Carlo Methods	40
4.2.2	Particle MCMC methods	42
4.3	Chapter Summary	43
5	Volatility Analysis with GPRSV Models	45
5.1	Simulated Data	45
5.2	Empirical Data	50
5.2.1	Volatility Forecast Evaluation	50
5.2.2	Data	51
5.2.3	Results	52
5.3	Chapter Summary	54
6	Conclusion and Future Work	57
	References	63

List of Tables

2.1	Descriptive Statistic of S&P 500 Daily Return Data	12
3.1	Descriptive Statistic of GE Daily Return Data	34
5.1	Descriptive Statistic of IBM Daily Return Data	52
5.2	Estimated GPRSV Model Hyper-parameters Results for IBM Daily Return Data	52
5.3	Results of IBM Volatility Forecast Using Loss Functions	55

List of Figures

2.1	Standard & Poor 500 Index Return and Close Price Data	13
2.2	Graphical Model Representation of Standard Gaussian Process Regression	18
2.3	Graphical Model Representation of State-Space Model	21
3.1	Graphical Model for Gaussian Process Regression Stochastic Volatility Model	24
3.2	Graphical Model Representation of a Gaussian Process State-Space Model	26
3.3	Flowchart of Volatility Model Building Process	29
3.4	GE Daily Return Data	32
3.5	Sample ACF and PACF Functions Plot for GE Daily Returns	33
3.6	Graphical Model for Gaussian Process Regression Stochastic Volatility Model with Exogenous Factor	35
5.1	Return and Variance Values of Simulated Data	46
5.2	Estimated Hidden States Densities of Simulated Data	47
5.3	Results of Simulated Data Hyper-parameters learned from RAPCF . . .	48
5.4	Predictive Log-likelihood value of Simulated Data	49
5.5	Results of IBM Daily Data Volatility Analysis	53

Chapter 1

Introduction

1.1 Motivation and Objective

Financial time series data analysis is one of the most studied areas in financial economics research, also it is a highly empirical discipline. Both academic researchers and finance market practitioners are interested in questions like these: what is the mechanism of financial market? What are the determining reasons for asset prices change? To answer these questions we need to find a proper way to describe the market and the vast data it generates. The financial market is a huge complex system determined by many factors such as political, corporate, and individual decisions. Financial data contain both meaningful information and random noises. So from an information processing point of view, it is nature to take the statistical modeling approach to the problem of financial time series analysis. For last decades, we have seen a lot of similar applications in engineering areas. Recent years there has been a dramatic growth of statistical models and related techniques used in finance as well. In this research, we would like to apply the engineering modeling techniques to the financial time series data. When working on financial time series data, we take the same methodology and utilize the same set of mathematical tools as we process other signals like audio, image and video in engineering applications.

In this research, we would like to exploit the problem of financial time series analysis using Bayesian nonparametric (BNP) models. One key feature of financial time series data is that there exists certain levels of uncertainty in the data [52]. For example, the asset volatility is not directly observed and some data is generally corrupted by noise. As

a result, we can use probability theory and related methods to express aspects of these uncertainties in our models. Another problem with financial time series data modeling is the huge amount and rapid growth of data. Ideally, a model should be adaptive enough to handle this. If the forecast output of the model is probability distribution, when more data comes, we should increase the probability of events which actually happened [21]. Based on above requirement, we would prefer a model which is flexible and robust enough to fit the financial time series data in. There is a very large body of current research on ways of doing approximate Bayesian machine learning [17]. The Bayesian nonparametric framework can provide an appropriate platform on which to approach massive data analysis [10].

Volatility plays a uniquely important role in the financial market, and modeling volatility becomes an increasingly important task in financial time series research. The main objective of this research is apply the Bayesian nonparametric modeling framework to analyze the conditional volatility. The volatility of the asset return series is an important factor in measuring risk. Because volatility describe the magnitude and speed of the time series' fluctuations, it can be interpreted as the variability of a financial time series. Although volatility is not the same as risk, its importance in conveying the uncertainty when making investment decisions makes it one of the most important variables. There are three main purposes of modeling asset volatility:

Risk Management potential future losses of one asset are measured because they account for large part of risk management. When we calculate these losses, the future volatilities are needed as an input.

Portfolio Optimization the standard approach of Markowitz [38] by minimizing risk for given level of expected return, the estimate of the variance-covariance matrix is required to proxy the risk. Like the application to risk management, the volatility for each asset in the portfolio is crucial important to optimize the portfolio.

Option Pricing all option traders try to develop their own volatility trading strategy, and based on that to compare the estimate for one option's value and the market price. Hence they can take bets on future volatility. This is perhaps the most challenging application. Since the Chicago board of option exchange (CBOE) introduced the ground-breaking volatility index VIX in 1993 [44], Many investors

worldwide consider VIX as the world's premier barometer of investor sentiment and market volatility. Because of its importance, the volatility index of a market has become a financial instrument. The VIX volatility index has been traded in futures since March 26, 2004.

1.2 Volatility Modeling Literature Review

In this section, we review main parametric volatility models, their advantages and disadvantages and techniques for estimate model parameters. Some alternative approaches for analyzing volatility are presented as well. Volatility which is often expressed as conditional standard deviation of asset return. In Equation (1.1), r_t denotes the return of an asset at time t , and I_{t-1} describe all the information we can obtain until time $t - 1$. The expected value μ_t and variance σ_t^2 of the return series are:

$$\mu_t = E(r_t|I_{t-1}) \quad (1.1a)$$

$$\sigma_t^2 = Var(r_t|I_{t-1}) = E[(r_t - \mu_t)^2|I_{t-1}] \quad (1.1b)$$

There are two general categories of volatility models: the generalized autoregressive conditional heteroscedasticity (GARCH) models and Stochastic Volatility (SV) models. The first category models describe the evolution of σ_t^2 using an exact function of all variables available until time $t - 1$, while those belong to the second category assume a stochastic process governs σ_t^2 . Both these two type of models share the same structure which can be expressed:

$$r_t = \mu_t + a_t \quad (1.2a)$$

$$\sigma_t^2 = Var(r_t|I_{t-1}) = Var(a_t|I_{t-1}) \quad (1.2b)$$

where a_t is called as the innovation of the asset return at time t .

1.2.1 GARCH Models

Autoregressive conditional heteroscedasticity (ARCH) model [16] was first introduced by Engle in 1982. An ARCH(p) model can be specified as follows:

$$a_t = \sigma_t \epsilon_t \quad (1.3a)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \dots + \alpha_p a_{t-p}^2 = \alpha_0 + \sum_{i=1}^p \alpha_i a_{t-i}^2 \quad (1.3b)$$

where a_t is the innovation of the asset return at time t . ϵ_t is assumed to be a sequence of independent and identically distributed(i.i.d.) random variables with zero mean and unit variance. α_0 and $\alpha_i, \dots, \alpha_p$ are model parameters, and $\alpha_0 > 0$, and $\alpha_i \geq 0$ for $i > 0$. For ϵ_t , it is often assumed to follow the standard Gaussian distribution or a generalized error distribution (GED) or standardized Student- t distribution.

ARCH model is the first systemic framework for volatility modeling, and it gives a good way to describe the asset return series features such as volatility clustering. The ARCH model is not only suitable to asset return data but also does well with other financial time series. Since the introduction of ARCH model, a lot of variants and extensions have been proposed. Bollerslev extended the model, and give a form of the Generalized Autoregressive Conditional Heteroskedasticity Model (GARCH) [6]. Similar to (1.3), the GARCH(p, q) can be summarized as follows:

$$a_t = \sigma_t \epsilon_t \quad (1.4a)$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^p \alpha_i a_{t-i}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2 \quad (1.4b)$$

where a_t , σ_t and ϵ_t are with the same meaning in (1.3). The GARCH(1,1) with ϵ_t follows a standard Gaussian distribution is easy to estimate and widely used in many real world financial applications. Here we can simplify (1.4) to obtain GARCH(1,1) with Gaussian innovation:

$$a_t \sim \mathcal{N}(0, \sigma_t^2) \quad (1.5a)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \beta \sigma_{t-1}^2 \quad (1.5b)$$

Although ARCH and GARCH are good candidate models to represent the properties of financial asset return series, such as *volatility clusters*, they are not perfect. There are some weaknesses like: Both models can not handle the *leverage* effect which is found by empirical financial data that volatility tend to react differently with positive and negative return shocks. This asymmetry effect in volatility equation is not captured in ARCH and GARCH models. Some GARCH extensions are developed to fix this problem. In 1991, Nelson [43] proposed the Exponential GARCH (EGARCH) model.

$$\log(\sigma_t^2) = \alpha_0 + \sum_{j=1}^q \alpha_j g(a_{t-j}) + \sum_{i=1}^p \beta_i \log(\sigma_{t-i}^2) \quad (1.6a)$$

$$g(a_t) = \theta a_t + \lambda |a_t| \quad (1.6b)$$

In Nelson's model, the logarithm of σ_t^2 is modeled instead of σ_t .

Another popular GARCH extension Threshold-GARCH [22] model was introduced by Glosten et al. in 1993, also see Zakoian [56]. The model is also called as GJR-GARCH,

$$\sigma_t^2 = \alpha_0 + \beta \alpha_{t-1}^2 + \gamma a_{t-1}^2 H_{t-1} \quad (1.7a)$$

$$H_{t-1} = \begin{cases} 0, & \text{if } a_{t-1} \geq 0 \\ 1, & \text{if } a_{t-1} < 0 \end{cases} \quad (1.7b)$$

where H_{t-1} is the threshold function. α_0, β and γ are model parameters.

The last GARCH typed model we want discuss in this section is the Markov Regime-switching GARCH model. The idea of regime switching models for economic data analysis was introduced at least three decades ago. See [25], [24], and [26] for details of regime switching models. The literature on Markov Regime-switching GARCH models (MRS-GARCH) begins with Cai [8]. In the paper of Marcucci [37] compared a set of standard GARCH models with a group of Markov Regime-Switching GARCH models. The main feature of MRS-GARCH is that it allows the parameters to switch across different regimes according to a Markov chain process. If we denote the regime variable as s_t , the transition probability is:

$$\Pr(s_t = j | s_{t-1} = i) = p_{ij} \quad (1.8)$$

Usually we assume there are two regimes, and the matrix form of Equation 1.8:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{21} \\ p_{12} & p_{22} \end{bmatrix} = \begin{bmatrix} p & (1-q) \\ (1-p) & q \end{bmatrix} \quad (1.9)$$

If the regime variable s_t takes the value of i , then the conditional mean and the conditional variance can be expressed in GARCH(1,1)-like form:

$$r_t = \mu_t^{(i)} + \sigma_t^{(i)} \epsilon_t \quad (1.10a)$$

$$a_t = r_t - \mu_t^{(i)} \quad (1.10b)$$

$$h_t^{(i)} = \alpha_0^{(i)} + \alpha_1^{(i)} a_{t-1}^2 + \beta_1^{(i)} h_{t-1} \quad (1.10c)$$

where h_t denotes the conditional variance, so we have $h_t = \sigma_t^2$.

The parameters of GARCH class models can be learned using maximum likelihood methods. There are a lot of papers on the topic and many software environment provide the program implementing the algorithms. Among these, see Kevin Sheppard's Matlab code of Oxford MFE Toolbox ¹. In [7], the authors explore the volatility forecasting performance of GARCH family models. There are two advantages of GARCH-type models: the analytical tractability and flexibility describing empirically observed features of asset returns.

1.2.2 Stochastic Volatility Models

Stochastic volatility (SV) models difference GARCH typed models with the process of how the conditional volatility evolves over time. For SV models, the volatility equation is expressed as a stochastic process, which means the value of volatility at time t is latent and unobservable. While for GARCH and its extensions, this value is totally determined by the information up to time t , which we defined as I_{t-1} in before. For example, Hull and White replaced Black-Scholes option-pricing formula [4] with a stochastic process [27].

The first discrete time-varying stochastic volatility model was introduced by Taylor, see [48] [50] [49]. The logarithm of variance was modeled by a latent AR(1) process.

¹The toolbox can be download from https://www.kevinsheppard.com/MFE_Toolbox. The earlier versions are called as UCSD_GARCH toolbox.

Taylor’s stochastic model can be presented as:

$$r_t = \mu_t + a_t = \mu_t + \sigma_t \epsilon_t \quad (1.11a)$$

$$\log(\sigma_t^2) = \alpha_0 + \alpha_1 \log(\sigma_{t-1}^2) + \sigma_n \eta_t \quad (1.11b)$$

where α_1 is a parameter which controls the persistence of logarithm variance, the value of α_1 is between (-1,1). There are two independent and identically distributed random variables ϵ_t and η_t . The original idea of SV model assume these two noise parts to be i.i.d. normal distributed. Recently, some researchers brought the idea of making ϵ_t and η_t negative correlated: $\text{corr}(\epsilon_t, \eta_t) < 0$. By doing this, the SV model can react asymmetric fashion to return shocks. This is similar to the way EGARCH extend GARCH model to reflect empirical observation of financial return series.

The inference of a SV model parameters is not as straightforward as the corresponding simple GARCH typed model. In [47], Shephard reviewed SV models and inference methods like *methods of moments* (MM) and *quasi-maximum likelihood* (QML). Simulation-based methods to learn SV models become more and more popular because their accuracy and flexibility of handling complicated models.

1.2.3 Alternative Approaches

Besides GARCH and SV models, there are some alternative approaches to solve this conditional volatility modeling problem. Here we discuss some of these methods: Using High-Frequency Data and Using Open, High, Low, and Close Price. High-Frequency Data, for example when we modeling daily conditional volatility, we can use the intra-day like 5-minute or 10-minute data to calculate the daily volatility. This approach sometimes is also called *realized volatility*. We will elaborate this later when this *realized volatility* is used as the proxy of the real volatility.

Another approach is called *implied volatility*, and it is related to option trading problems. If we assume that the prices are governed by some econometric models for example the Black-Scholes equation, we can use the price to calculate the “implied” volatility. Experience shows implied volatility is often larger than the value of GARCH type model. The VIX of CBOE is an example of implied volatility. The calculation of VIX is based

on this equation (see [14] for more details):

$$\sigma^2 = \frac{2}{T} \sum_i \frac{\Delta K_i}{K_i^2} e^{RT} Q(K_i) - \frac{1}{T} \left[\frac{F}{K_0} - 1 \right]^2 \quad (1.12)$$

where σ is VIX/100, T is time to expiration, F is forward index level desired from index option prices, K_0 is the first strike below the forward index level F . K_i is strike price of the i th out-of-the-money option. R is the risk-free interest rate to expiration, and $Q(K_i)$ is the midpoint of the bid-ask spread for each option with strike K_i .

1.3 Main Contributions

In this thesis, we propose a novel nonparametric model which we call it Gaussian process regression stochastic volatility (GPRSV) model. We use GPRSV model to solve the problem of modeling and forecasting time varying variance of financial time series data. For the standard econometric volatility models (including both GARCH and SV classes), model forecast performance is limited by the rigid linear transition function form. Moreover, the model parameters are usually learned by maximum likelihood methods, which can lead to over-fitting problems. We apply the recent development of Bayesian nonparametric modeling methods to unblock the bottleneck of financial time series volatility modeling. The Gaussian process regression stochastic volatility models are more natural to describe the financial time series dynamic behaviors.

The second contribution of this research is the development of algorithms to learn the proposed models. We applied the recent developed learning algorithms to learn the GPRSV models. We use tailed sequential Monte Carlo and particle Markov chain Monte Carlo methods to jointly learn the hidden states trajectory and Gaussian process hyper-parameters. Most of the previous work on state-space model inference has took the approach of separating the hidden states filtering and parameters estimating. The GPRSV model usually is more difficult to estimate than a GARCH or SV model. By taking a full Bayesian nonparametric approach we learn the hidden states or system variable distribution, so our inference method is free of the over-fitting problem as using maximum likelihood methods for the traditional parametric models.

1.4 Organization of Thesis

In chapter 2, we describe the background of this thesis. The characteristics of financial time series data, preliminaries of volatility modeling and basic of Bayesian nonparametric models are discussed. Also we present fundamental knowledge of Gaussian process and state-space models.

In chapter 3, we propose our Gaussian process regression stochastic volatility models. We discuss the model's structure, the process of building a GPRSV model and the issue of introducing exogenous factors to improve the forecasting performance.

In chapter 4, we discuss how to learn Gaussian process regression stochastic volatility models. We introduce a novel estimating approach to learn both the hidden volatility and model's hyper-parameters together. Monte Carlo methods are provided to learn the nonparametric models.

In chapter 5, we conduct experiment to prove the advantages of proposed modeling approach. Both simulated and empirical financial data are tested using our GPRSV model and tailored Sequential Monte Carlo algorithm.

In chapter 6, we conclude our work and discuss future work for this research.

Chapter 2

Background

In this chapter, we would like to present the background of this research. There are two parts, the first part is the data set characters we are study and the preliminaries of volatility modeling. The other part is the methodology we are going to use: Bayesian nonparametric framework, Gaussian process and state-space models.

2.1 Financial Time Series Data

Time series data are collected through time. One time series is a sequence of data points of measurement $z_t \in \mathbb{R}$ index by time t . Time series can be in discrete or continuous form. The discrete time series can be viewed as special case of continuous time series. If we measure the continuous time series exactly once per unit time we can obtain the discrete ones. This is often called as *uniform sampling* [53]. To simplify we often but not always focus on discrete time series. Another classification for time series is univariate or multivariate. Throughout this thesis, we mainly discuss the problem of discrete univariate time series analysis. The objective of time series analysis is utilizing the theory and methods to extract meaningful statistics and other characteristics from the data. Time series analysis is widely used for many real world applications in the domains of science, economic and engineering.

Financial time series analysis is a highly empirical discipline, and people more concern with the theory and practice how asset valuation change over time. In financial time series research, people usually analyze assets return instead of price [9]. See Figure 2.1 for the

differences between asset return and price data. The data we use here is Standard & Poor 500 (S&P 500) index data from January 3, 1990 to July 21, 2009. Clearly we can observe that consecutive prices are highly correlated and the variance increases with time. The return series (in percent) we get in Figure 2.1 and Table 2.1 is defined as:

$$r_t = 100[\log(p_t) - \log(p_{t-1})] \quad (2.1)$$

where r_t is the return at time t , p_t is the asset price at time t . In Table 2.1, we give the descriptive statistics of the data.

Table 2.1: Descriptive Statistic of S&P 500 Daily Return Data

Mean	Standard Deviation	Skewness	Kurtosis	Min	Max
0.0198	1.1761	-0.1936	12.2642	-9.4695	10.9572

2.2 Volatility Modeling Preliminaries

The idea behind volatility modeling is to express the relationship of the return and volatility and how these two processes evolve over time. Before we go on to discuss Bayesian nonparametric volatility models, there are several preliminaries and background we need to present in this chapter. Volatility modeling and forecasting is an important task in financial markets and it was born about 30 years ago ever since Engle introduced the autoregressive conditional heteroskedasticity (ARCH) model in 1982. Both academics and practitioners are interested in this problem.

The most special feature of asset return is that volatility can not be directly observed from the return data. For example, the daily data we used to plot Figure 2.1 consist of 4895 observations of Standard and Poor 500 daily returns. There is only one observation in a trading day, if we do not use intra-day data. Even we use the intra-day data we can only estimate part of the volatility. Another part is the overnight volatility which

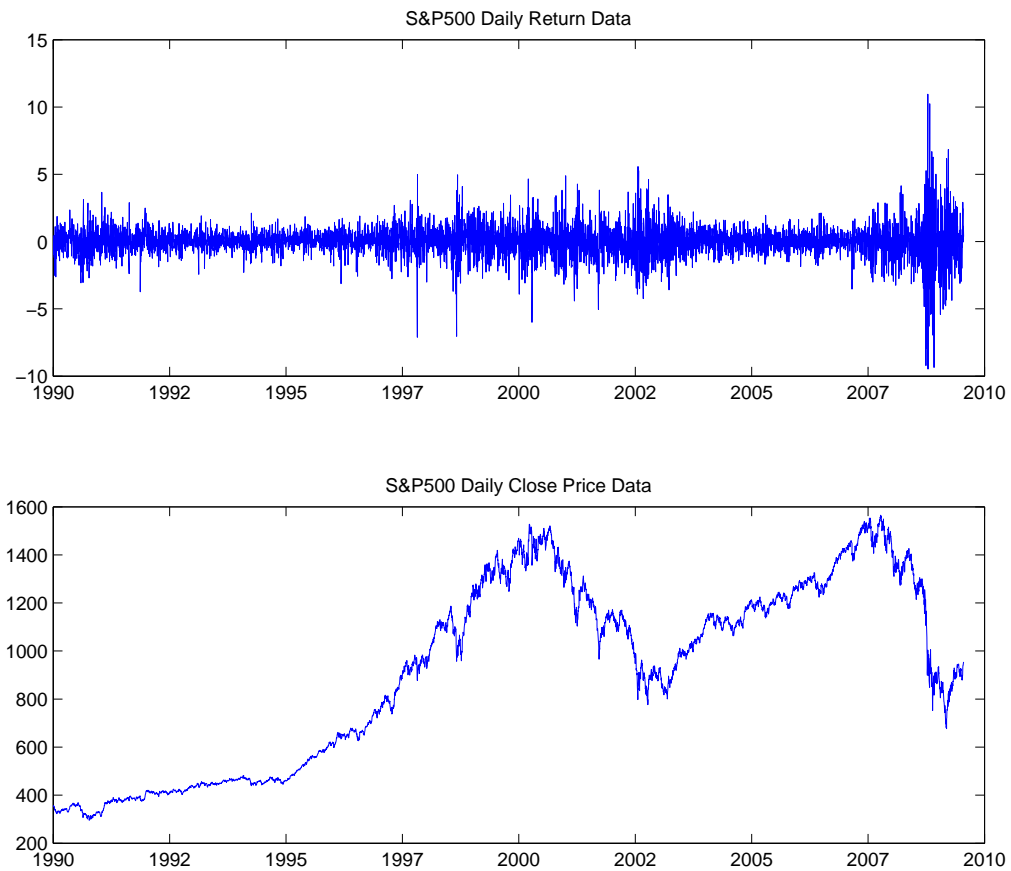


Figure 2.1: Standard & Poor 500 index return and close price data from January 1, 1990 to July 21, 2010

intra-day data can provide very little information. This observability of volatility makes evaluating the forecasting of candidate models difficult.

Besides the hidden feature, there are some other characteristics which are commonly observed in asset return series.

- *Heteroscedastic*: the volatility of asset return is not constant through time. This is also called Heteroscedasticity. The most common phenomenon of this heteroskedastic effect is the example of launching rocket. For asset return, the value of this conditional volatility is time varying.
- *Volatility Clustering*: it is widely accepted that the asset return tend to exist clusters for volatility, which also means there is some period the market is with high volatility and there is some period with lower volatility. In 1963, Mandelbrot [36] pointed out that "large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes."
- *Asymmetric Effect*: based on rich empirical observations of financial asset returns, volatility tend to react differently on positive and negative returns. This is one important character that the early models like ARCH, GARCH and basic SV all failed to capture, and there are many ways of modifying those models to deal with this *asymmetric effect*. For example, the EGARCH model was introduced to fix this problem based on the GARCH framework. For the SV models, one possible solution is to change the independence of ϵ_t and η_t in Equation (1.11), and make the correlation of the two innovation negative: $\text{corr}(\epsilon_t, \eta_t) < 0$.
- *Heavier tails*: volatility models should explain that the asset returns are not normally distributed. Actually rich evidences prove that financial asset return exhibit heavy tails and high-peakedness. Even in GARCH models, we assume that returns are conditionally normally distributed, the unconditional (marginal) distribution can be represented as a mixture of normal distributions. The tail of the mixture normals turn out to be heavier than the single normal distribution.
- *Stationary*: volatility usually changes within some fixed ranges, and it evolves over time in a continuous manner. Sudden jumps are rare for most asset returns. Before modeling the return series, we can use some statistical tests to test the stationary of the series.

2.3 Bayesian Nonparametric Framework

The Bayesian approach to data analysis and machine learning is based on using probability to represent all forms of uncertainty [41]. The process flow can be summarized as:

- *Define model*: We express qualitative aspects of the system, by defining random variables, their forms of distributions, and independence assumptions. Also we specify prior probability distribution for the unknown parameters.
- *Collect data*: We can compute the posterior probability distribution for the unknown parameters, given the collected data.
- *Make decisions*: With the posterior we can make scientific conclusion, predict future output by averaging over the posterior distribution and make decisions to minimize expected loss.

The Bayes' Rule for modeling:

$$P(\theta|D) = \frac{P(\theta)P(D|\theta)}{P(D)}$$

where $P(D|\theta)$ is likelihood of unknown parameters θ , $P(\theta)$ is prior probability of θ and $P(\theta|D)$ is the posterior of θ given data D . Also, we can write this relationship as:

$$P(\text{parameters}|\text{data}) \propto P(\text{parameters})P(\text{data}|\text{parameters})$$

or

$$\text{Posterior} \propto \text{Prior} \times \text{Likelihood}$$

We can see from above that choosing suitable prior is very important to Bayesian Modeling. In parametric models, finite set of parameters are assumed. Given the parameters, future predictions are independent of the observed data. so the unknown parameters capture everything about the data. The capability of the model is bounded even if the amount of data is unbounded [21]. While nonparametric models often assume that there are an infinite dimensional parameters θ . The nonparametric models make fewer assumptions about the dynamics, and thereby the data drive the complexity of the model

[17]. We can think of θ as function instead of a vector. By saying nonparametric, it is not mean there is no parameters, but actually infinitely many ones. The infinite dimensional θ often takes form of a function. From information channel viewpoint, all the models are like information channels, with the past data D as input and future prediction y as output. For parametric models, given the model parameters, future predictions, y are independent of the observed data. The model’s complexity or the channel’s capacity is bounded. That is to say the parameters constitute a bottleneck in channel. nonparametric models are free of this bottleneck problem, with more data D , the more information θ can capture. To make predictions, nonparametric models need to process a growing amount of training data D . This information channel view of nonparametric modeling was first pointed out in [21] by Ghahramani.

As presented in recent report [10], “big data” arises in many areas. Terabytes of data, in some cases petabytes of data are generated. The rapid growth heralds an era of “data-centric science”. The Bayesian nonparametric modeling framework is adaptive, robust and flexible way of analyzing data and it could be promising technique for the problem of “big data” analysis.

2.4 Gaussian Process

Gaussian process, together with Dirichlet process (DP), constitute the fundamental tools in Bayesian nonparametrics. For a Gaussian process, it places distribution over functions, while a Dirichlet Process is a distribution on distribution. Here we give a compressed introduction to Gaussian process (GP), more detailed discussion can be found in the textbook of Rasmussen and Williams [45], the paper of Neal [42]. Gaussian process was first introduced in the statistics community as *kriging*. Most probability distributions are over finite dimensional objects (scalars, vectors), while functions are infinite dimensional. So a Gaussian process can be viewed as an extension of multivariate Gaussian distribution to infinite dimensions. Like a Gaussian distribution is specified by a mean vector and covariance matrix, a Gaussian process is determined by mean function and covariance function. Similarly we can have the definition of Gaussian process:

$$f \sim \mathcal{GP}(m(x), k(x, x')) \quad (2.2)$$

Here we use this notation to say function f is drawn from a Gaussian process. $m(x)$ and $k(x, x')$ are the mean function and covariance function. The covariance function $k(x, x')$ is also called as kernel function.

From the definition we can see that any finite subset of sampled function values \mathbf{y} from the process follows a multiple Gaussian distribution. The mean function is often set to be zero: $m(x) = 0$ for many Gaussian process applications, but this is not case for our Gaussian process regression stochastic volatility (GPRSV) model in Chapter 3. In GPRSV model, the mean functions are not simply assumed to be zero but to be adjusted to the specific application requirements, we will discuss the reason in details in that chapter. The covariance function $k(x, x')$ measures the “similarity” between inputs \mathbf{x} and \mathbf{x}' . The parameters in mean and covariance functions are called hyper-parameters. These hyper-parameters will control the sampled function’s properties: smoothness, input output scale and so on (see [53]). One of the most used covariance function is the squared-exponential (SE) function:

$$k(x, x') = \gamma \exp\left(-\frac{1}{2l^2}|x - x'|^2\right) \quad (2.3)$$

where γ and l are the hyper-parameters for the covariance function.

2.4.1 Gaussian Process Regression

Gaussian process can be used for the non-linear regression problems in many fields, such as machine learning, statistics and engineering. In this thesis we would like to apply this useful tool to financial time series data analysis. The object of a non-linear regression problem is to find how to express y using the covariates x . Simply we use the following equation to describe the relationship:

$$y_i = f(x_i) + \epsilon_i \quad (2.4)$$

where f is called the regression function, and ϵ_i is the random residuals which are assumed to be i.i.d. Gaussian distributed with mean zero and constant variance σ^2 . A Gaussian process represents a powerful way to perform Bayesian inference about functions.

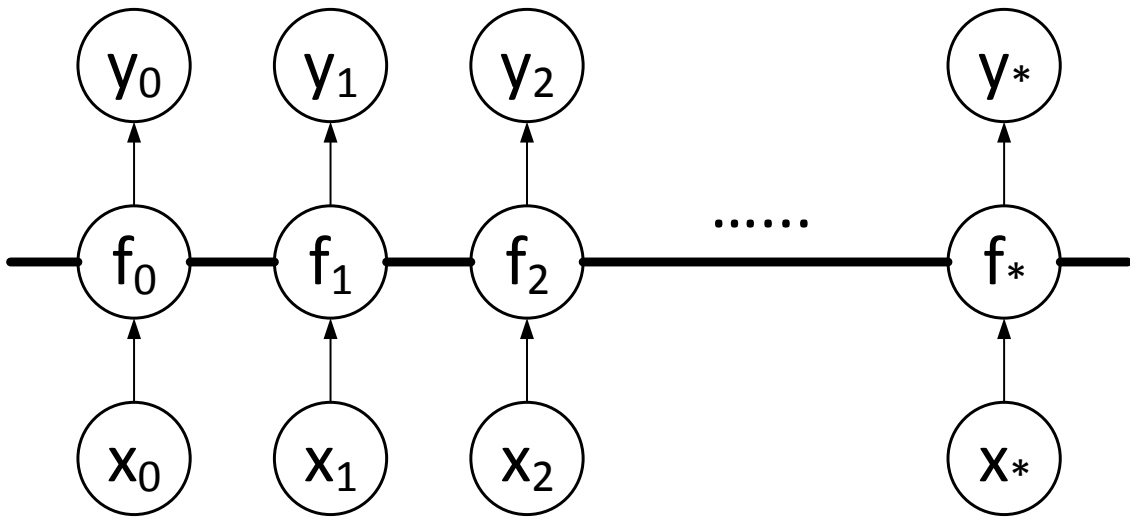


Figure 2.2: Graphical model representation of standard Gaussian process regression, where x_t is the input at time t , and y_t are the output of regression process at time t . The Gaussian process regress function value is f . The thick horizontal line represent fully connected nodes.

2.4.2 GP for Time Series

In [53], Turner pointed out that there are two approaches for Gaussian Process time series modeling: *Gaussian process time series (GPTS)*, and *autoregressive Gaussian process (ARGP)*. The first one GPTS can be described below:

$$y_t = f(t) + \epsilon_t \quad (2.5a)$$

$$f \sim \mathcal{GP}(0, k) \quad (2.5b)$$

$$\epsilon_t \sim \mathcal{N}(0, \theta^2) \quad (2.5c)$$

where the time series input is time index t , and the time series output is y_t . ϵ_t is standard normal distributed noise. The GPTS models generalize a lot of the classic time series models: the autoregressive (AR), the autoregressive moving average (ARMA).

The other Gaussian process to time series modeling approach is ARGP. Compared to GPTS, ARGP is more general and powerful but more computational. In Equation (2.6) we present an ARGP with order p , $y_{t-p:t-1}$ are the p previous values of the output y_t .

$$y_t = f(y_{t-p:t-1}) + \epsilon_t \quad (2.6a)$$

$$f \sim \mathcal{GP}(0, k) \quad (2.6b)$$

$$\epsilon_t \sim \mathcal{N}(0, \theta^2) \quad (2.6c)$$

In this thesis, we choose the ARGP instead of GPTS to model financial time series because ARGP is more general than GPTS and ARGP can model more complex dynamics than GPTS do.

Both the GPTS and ARGP can handle external inputs with the regress process, the ARGP with external inputs z_t can be generalized as below:

$$y_t = f(y_{t-p:t-1}, z_{t-p:t-1}) + \epsilon_t \quad (2.7a)$$

$$f \sim \mathcal{GP}(0, k) \quad (2.7b)$$

$$\epsilon_t \sim \mathcal{N}(0, \theta^2) \quad (2.7c)$$

2.5 State-Space Models

Discrete time series data can be modeled mainly with two approaches: *autoregressive* and *state-spaces*. The main difference of the two methodologies is that state-space models assume the system is governed by a chain of hidden states, and we can inference the system by a sequence of observations which are determined only the hidden states. The advantages of state-space models over autoregressive models are: not requiring the specification of an arbitrary order parameters and a richer family of naturally interpretable representation. For most of the times, a state-space model can be a general framework for describing dynamical phenomena. Only when inference is difficult the autoregressive models are more advantageous, but this special case is not discussed in this thesis.

State-space models (SSM) or hidden Markov models (HMM) are one of the most widely used types of methods for effectively modeling time series and describing dynamical system. In different areas SSM maybe named differently: structural models (Econometrics), linear dynamic models (LDS) or Bayesian forecasting models (Statistics), linear system models (Engineering). In Finance, state-space models can generalize other popular time series models such as ARMA, ARCH, GARCH and SV.

A state-space model consists of two parts: hidden state \mathbf{x}_t and observation variable \mathbf{y}_t . The essential idea is that behind the observed time series \mathbf{y}_t there is an underlying process \mathbf{x}_t which itself is evolving through time in a way that reflects the structure of the system. The general form of SSM can be summarized as:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) + \epsilon, \mathbf{x}_t \in \mathbb{R}^M \quad (2.8a)$$

$$\mathbf{y}_t = g(\mathbf{x}_t) + \nu, \mathbf{y}_t \in \mathbb{R}^D \quad (2.8b)$$

where ϵ and ν are both i.i.d. noise with zero mean and unit variance. The unknown function f describe the system dynamics and function g links the observation and the system hidden state. Both the f and g functions can be linear or non-linear. The hidden state \mathbf{x}_t follows a Markov chain process which is the reason we use the terminology of hidden Markov models. In figure 2.5, we give the graphical model representation of one SSM.

We use $p(\mathbf{x}_t|\mathbf{x}_{t-1};\theta)$ and $p(\mathbf{y}_t|\mathbf{x}_t;\theta)$ to define the conditional distributions of both

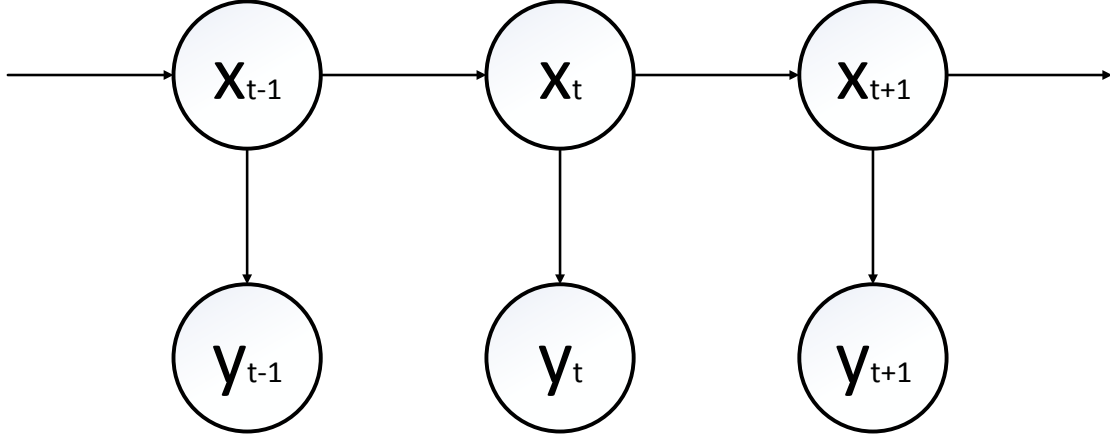


Figure 2.3: Graphical model representation of a state-space model, where \mathbf{x}_t is the hidden state at time t , and \mathbf{y}_t is the observation at time t . The system state follows a Markov chain process, and the unknown parameter vector θ is omitted in the figure.

state and observation variables, θ is a vector of unknown parameters. Also we define $\mathbf{x}_{1:t} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_t\}$ and $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t\}$. Since we cannot observe the system variable $\mathbf{x}_{0:t}$, we are interested in estimate $\mathbf{x}_{0:t}$ using the observation $\mathbf{y}_{1:t}$. The conditional probability distribution of $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}; \theta)$ is calculated using Bayes' Rule:

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}; \theta) = \frac{p(\mathbf{x}_{0:t}, \mathbf{y}_{1:t}; \theta)}{p(\mathbf{y}_{1:t}; \theta)} \quad (2.9)$$

where the integration $p(\mathbf{y}_{1:t}; \theta)$ is called the likelihood of the state-space model. Besides the joint smoothing distribution, there are three marginal distributions people are interested as well: the one-step ahead predictive distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t-1}; \theta)$, the filtering distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t}; \theta)$, and the smoothing distribution $p(\mathbf{x}_t|\mathbf{y}_{1:T}; \theta)$.

2.6 Chapter Summary

In this chapter, we present the basic background knowledge for this thesis, the financial time series data we are studying and the methodology we use, and most important

the preliminaries for volatility modeling. The characteristics for volatility models try to capture, the prepare work we need to do before we apply our modeling procedure to the asset return. And we present the structure of Gaussian process for time series analysis and state-space models which are the fundamental knowledge for our Bayesian nonparametric modeling approach to volatility analysis.

Chapter 3

Gaussian Process Regression Stochastic Volatility Model

In this chapter we introduce the Gaussian process regression stochastic volatility (GPRSV) model to solve the problem of financial time series volatility modeling and forecasting. Like GARCH and basic stochastic volatility (SV) models, we model the financial asset return and volatility in state-space way. The logarithm of variance is modeled as the system's unobserved latent variable in our model. We use Gaussian process (GP) to sample unknown hidden states transition function. A GPRSV model can be viewed as an instant of Gaussian process state-space model (GP-SSM). Gaussian process which has been discussed in chapter 2 is a flex and powerful tool to model time series data. After introducing the GPRSV model, we continue to discuss the procedure of building a GPRSV model, and the issue of introducing exogenous factors to improve GPRSV model forecast performance.

3.1 GPRSV Models Framework

In GPRSV model, the conditional volatility is modeled in a Bayesian nonparametric way. We assume that the hidden system states process is governed by a stationary stochastic process. The main difference between GPRSV and traditional stochastic volatility models is the driven force for the stochastic process. In traditional stochastic volatility models, the process is assumed to follow a rigid linear autoregressive form. In GPRSV model the

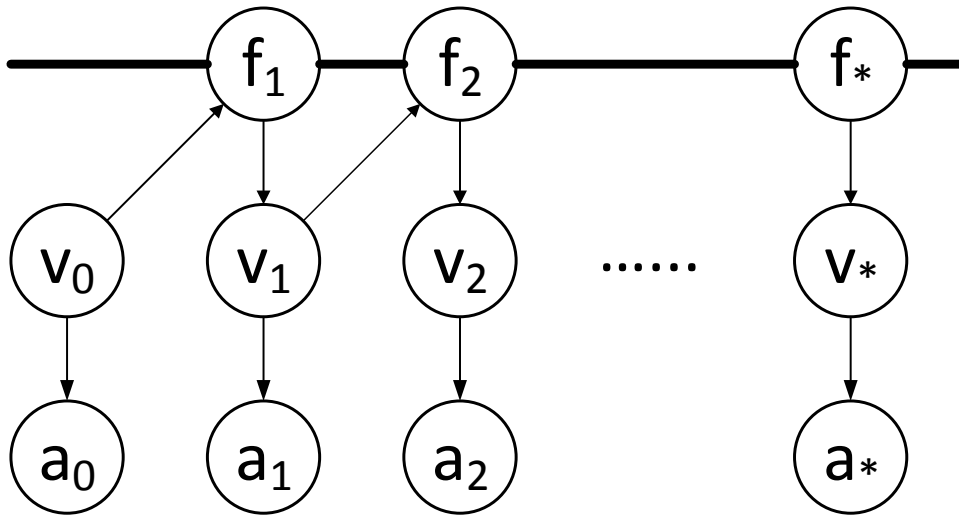


Figure 3.1: Graphical model representation of a Gaussian process regression stochastic volatility model, where a_t is the observation variable at time t , and v_t are the hidden variable (logarithm of volatility) at time t . f_t is the Gaussian process sampled function value at time t , and the thick horizontal line represent fully connected nodes. Hyper-parameters of the Gaussian process are omitted in the figure.

process is not limited to take a rigid form but a Gaussian process prior is placed over the transition function. The basic framework of a GPRSV model can be presented with the following equations:

$$a_t = r_t - \mu = \sigma_t \epsilon_t \quad (3.1a)$$

$$v_t = \log(\sigma_t^2) = f(v_{t-1}) + \tau \eta_t \quad (3.1b)$$

$$f \sim \mathcal{GP}(m(x), k(x, x')) \quad (3.1c)$$

where r_t is the asset return at time t and μ is the mean of asset return series, and a_t is the innovation of the return series. v_t is the logarithm of variance at time t , ϵ_t and η_t are i.i.d. Gaussian (or student's t) distributed noises. τ is unknown scaling parameters to be estimated. The function f is the hidden state transition function. Here we assume this function f follow a Gaussian process, which is defined by the mean function $m(x)$ and covariance function $k(x, x')$. We use logarithm of variance instead of standard deviation directly in our model. The advantage of using logarithm form can be found in [48] and [43]. Taylor's SV model and Nelson's EGARCH model used logarithm form in their models as well.

In Gaussian process, the mean function $m(x)$ can encode prior knowledge of system dynamics. For volatility modeling problems, we can encode the asymmetric effect in the mean function. The covariance function $k(x, x')$ is defined by covariance between function values $Cov(f(v_t), f(v_{t'}))$, so the covariance function is used to describe the correlation relationship of the time-varying volatility values. In Figure 3.1 we give the graphical model representation of a GPRSV model. We model the logarithm of variance instead of volatility directly, which is the same way as in EGARCH model and Taylor's stochastic volatility model.

3.2 GP-SSM for Financial Time Series Modeling

Our GPRSV model can be viewed as an instant of Gaussian process state-space models (GP-SSM) which are proved to be powerful tool to describe the nonlinear dynamic systems. Gaussian process are widely used as dimensionality reduction technique in machine learning community. In [34], Lawrence introduced the Gaussian Process Latent Variable

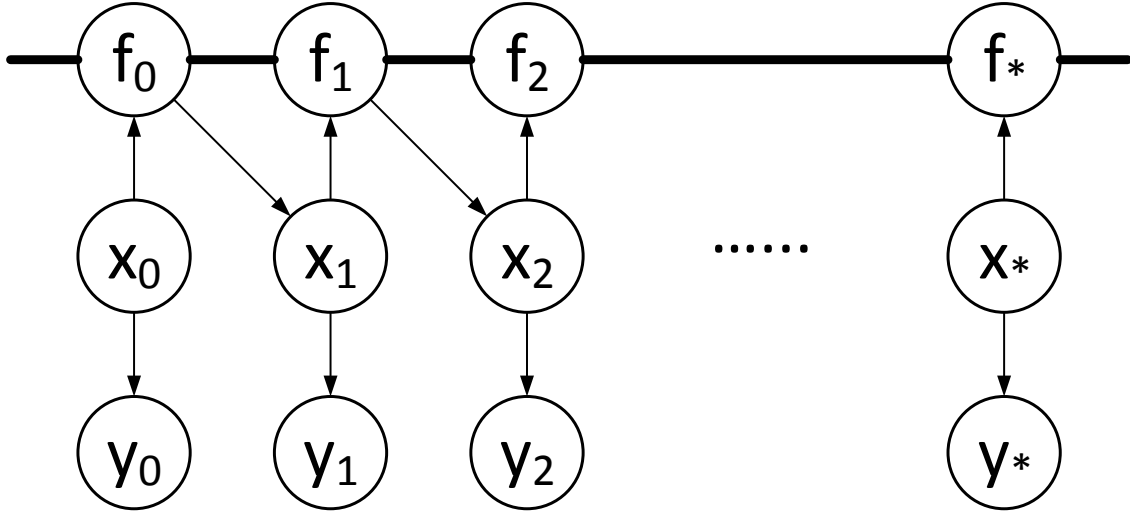


Figure 3.2: Graphical Model Representation of a Gaussian Process State-Space Model, where x_t is the hidden state variable value at time t , and y_t is the observation value at time t . f_t is the Gaussian process regression function (the hidden state transition function) value at time t . The thick horizontal line represent fully connected nodes

Model (GPLVM) for principal component analysis (PCA). In Lawrence's model, Gaussian process prior is used to map from a latent space to the observed data-space which is high dimensional. In [33], Ko and Fox proposed a Gaussian process based Bayesian Filter which is a nonparametric way of recursively estimating the state of a dynamical system. Wang et al. proposed the Gaussian Process Dynamical Model (GPDM) in [54]. GPDM enriches the GPLVM to capture temporal structure by incorporating a Gaussian process prior over the dynamics in the latent space. Frigola et al. pointed out that Gaussian process can represent functions of arbitrary complexity and provide a straightforward way to specify assumptions about the unknown function in [18]. Gaussian process related regression and classification problem has emerged as a major research field for time series modeling in machine learning community, however the advantage of this Bayesian nonparametric framework has got enough attention of financial researchers and market practitioners. We would like to apply this flexible and powerful modeling tool to the problem of financial time series analysis. We can combine Gaussian process and state-space

model together. The way of combining the two is to use the state-space model’s structure and apply Gaussian process to describe the hidden state transition function. The essence of the GP-SSMs is to change the rigid form of states transition function of traditional state-space models to a Gaussian process prior. In Figure 3.2, we show the Gaussian process state-space model graphical model representation. Financial data exhibits many dynamics because of the market is changing all the time and a lot of small change of the involved factors can result in significant fluctuation. As more and more data is available to process, the rigid form of state transition function in traditional state-space models becomes the bottleneck of improving the models’ forecast performance.

To learn a GP-SSM is more difficult than a standard SSM. In our work, we take the Bayesian approach to solve the GP-SSMs inference problem. Bayesian filtering is a type of technique used to estimate the hidden states in dynamic systems. The goal of Bayesian filtering is to recursively compute the posterior distribution of the current hidden state given the whole history of observations. One of the most fundamental and widely used Bayesian filters is the Kalman filter, but one of Kalman filter’s limitation is that it can only deal with linear and Gaussian noisy models. Two popular extensions for non-linear systems are the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) (see [39], [46], [29] and [30]). Markov chain Monte Carlo (MCMC) methods can be used to learn a state-space model parameters. Particle filter (PF) is another Monte Carlo method to the problem of filtering a state-space model. We will discuss the details of learning GP-SSMs in Chapter 4.

3.3 Model Structure

All volatility models try to describe how the hidden volatility value evolve over time and capture the characteristics of the asset return series we have discussed in Chapter 2. To achieve these goals, we need to put the conditional volatility modeling problem in a reasonable structure. Compared with auto regression approach, state-space models provide us a more general and flex framework to describe dynamic systems. Both the GARCH and SV models can be viewed as instants of state-space models. The hidden volatility is naturally modeled as the system’s state variable, and the return is observable to us. In Chapter 1 literature review part, we have reviewed these two categories of

conditional volatility models. Both these types of models follow the same structure to presenting how the return and variance process with time. In this structure there are two equations: *mean equation* and *variance equation*. For all these models the mean equations are similar. Usually we can assume r_t follows simple time series models like stationary AR, ARMA models (we can add some explanatory variables if necessary). For most of the times, the conditional mean can be simply described like in Equation (1.2a).

What distinguish one volatility model from others is the variance equation. In Equation (1.2b), we can see the statistical meaning of the conditional volatility but it does not give any information about the manner how σ_t^2 evolves over time. All the GARCH family models use an exact function to describe the evolution of σ_t^2 , while all stochastic volatility models use a stochastic equation to describe σ_t^2 . Although these two categories models are quite different at this point, but both the two typed of models use a linear regress form in *variance equation*. Both GARCH and standard SV models belong to parametric models. In [55], the Gaussian process was introduced to volatility modeling problems, the authors proposed the Gaussian process volatility (GP-Vol) model which can be viewed as an extension of GARCH model. Our GPRSV model apply the Gaussian process regression tool to replace the linear regression function in standard stochastic volatility model. In GP-Vol model and our GPRSV models, the volatility as the hidden state variable is modeled in a nonparametric way. The state transition function is assumed to generated from a Gaussian process. The function is not specified to follow a certain linear or non-linear form as in standard economic models. Functions sampled from a Gaussian process can take many forms despond on the mean and covariance functions associated with the Gaussian process.

3.4 Model Building Process

In [52], Tsay gave a four steps process for building a conditional volatility model and applied it to analyze empirical stock market data using GARCH models. Similarly, we can build our GPRSV model with the following steps:

- **Step 1: specify mean equation.** First we need to test the serial dependence in the return series. If the series are linear dependent, we should use an econometric model (e.g. an ARMA model) to remove the linear dependence in return series.

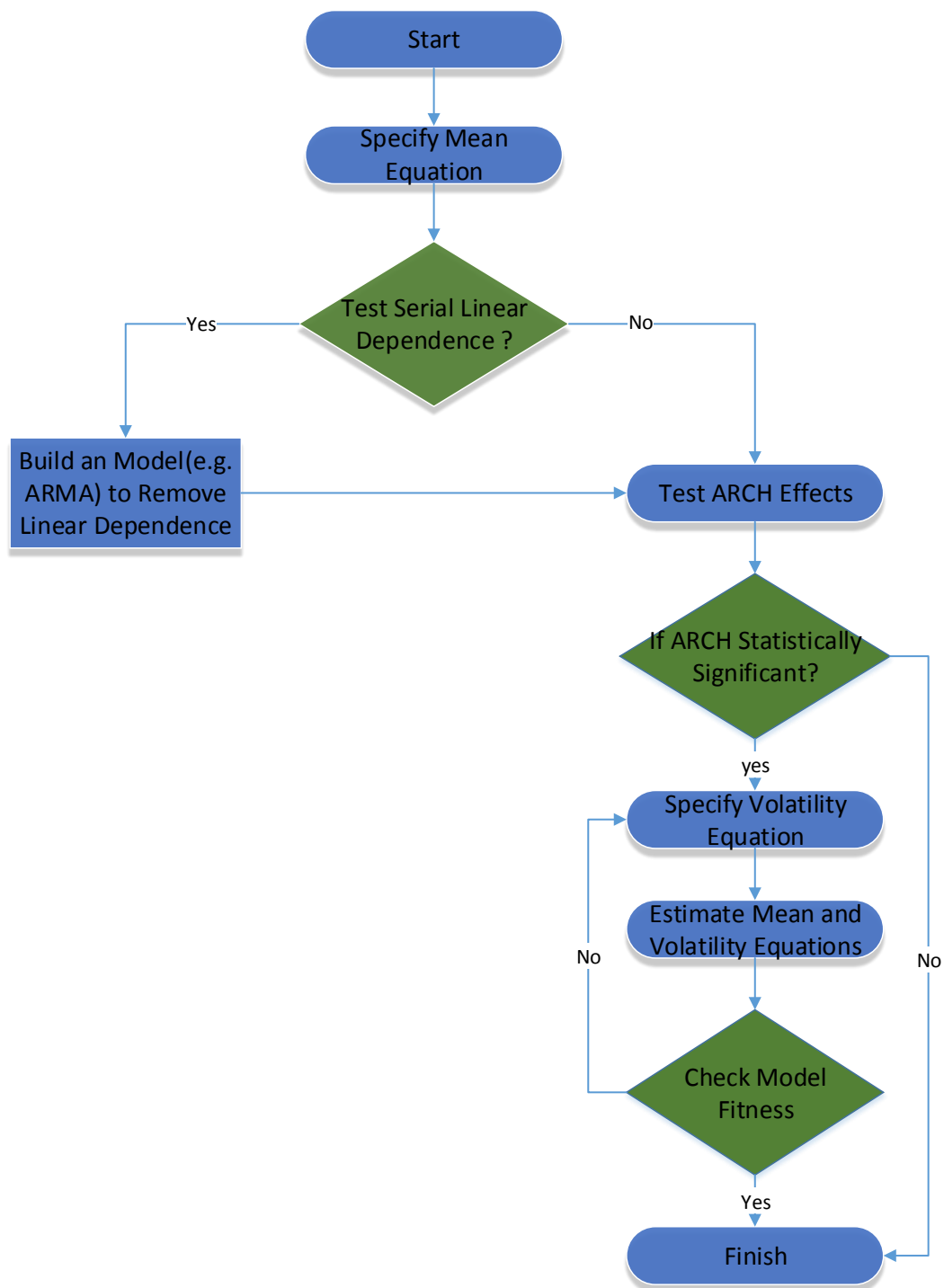


Figure 3.3: Flowchart of Volatility Model Building Process

Depending on the data we want to model we can use different methods to remove the linear dependence. After doing that, we can specify the distribution the return variable. In Equation (3.1a), we simply normalize the return series to remove the linear dependence part. If mean of the return series is significant small, we can use the return series directly, otherwise we model the innovation or residuals a_t , and we specify ϵ_t as a Gaussian or Student's- t distribution.

- **Step 2: test ARCH effect.** The residuals of the asset return a_t expressed in (3.1a) are often used to test the series' conditional heteroscedasticity. This conditional heteroscedasticity is also known as the ARCH effects [52]. There are two kind of test for ARCH effect, the first one is to apply the Ljung-Box statistics $Q(m)$ to a_t^2 [40], and the second test is the Lagrange multiplier (LM) test [16]. The null hypothesis of Ljung-Box test is that the first m lags of autocorrelation function (ACF) of the testing series are zero. For the Lagrange multiplier test, we assume in the linear regression form:

$$a_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \dots + \alpha_m a_{t-m}^2 + c_t \quad (3.2)$$

where $t = m + 1, \dots, T$, c_t is the noise term and T is the sample size. Additional we define

$$SSR_0 = \sum_{t=m+1}^T (a_t^2 - \bar{\omega})^2 \quad (3.3a)$$

$$SSR_1 = \sum_{t=m+1}^T \hat{c}_t^2 \quad (3.3b)$$

$$F = \frac{(SSR_0 - SSR_1)/m}{SSR_1/(T - 2m - 1)} \quad (3.3c)$$

where $\bar{\omega} = (1/T) \sum_{t=1}^T a_t^2$ is the sample mean of a_t^2 . F is asymptotically distributed as a chi-squared distribution χ_m^2 under null hypothesis. m is the degree of freedom. The null hypothesis H_0 is $\alpha_1 = \dots = \alpha_m = 0$. The decision rule is to reject H_0 if $F > \chi_m^2(\alpha)$ (here $\chi_m^2(\alpha)$ is the upper $100(1 - \alpha)$ th percentile of χ_m^2), or type-I error: the p value of F is less than α (see [52] for details).

- **Step 3: specify volatility equation.** The key of volatility modeling is to specify how the hidden variable volatility or logarithm of variance evolves over time. In GPRSV models, this part is modeled using the flexible Bayesian nonparametric tool, Gaussian process regression. For GARCH and SV models this part is modeled in linear regression approach. Once we estimated the model's parameters, those parametric models are determined. When the hidden variable is modeled using Gaussian process regression, we need to specify both the mean and covariance functions. Besides these functions forms, the initial value of hyper-parameters (the parameters in mean and covariance functions are called hyper-parameters) associated with them need to be specified as well. How to choose the function forms and initial hyper-parameters are discussed in details in Chapter 5 when we analyze the empirical financial asset data.
- **Step 4: estimate model parameters and check model fitness.** After specifying both the mean and volatility equations and associated parameters and in Step 2 and Step 3, we can use training data to estimate the unknown parameters. Once we get our estimated parameters we can use testing data to test learned model, and it is necessary to check the fitness of model we obtained so far. Sometimes we need to go back to Step 3 to modify our Gaussian process mean and covariance function forms or hyper-parameters.

To demonstrate the four steps process, here we show the flowchart of this model building process in Figure 3.3. Also we use stock market data to further explain the process. We analyze the daily return data of GE corporation. The data is collected from January 1, 1990 to September 29, 1994 with 1200 observations. See Figure 3.4 for the return series. In Table 3.1 we give the descriptive statistic of the series. Because the mean value is very small, we can model the return series directly for this data set. In Figure 3.5, the sample autocorrelation function (ACF) and partial autocorrelation function (PACF) [28], [20] of the return and square return series are plotted. From these figures, we can clearly see that there is no significant serial correlation but the series are dependent for the GE daily return data during the period.

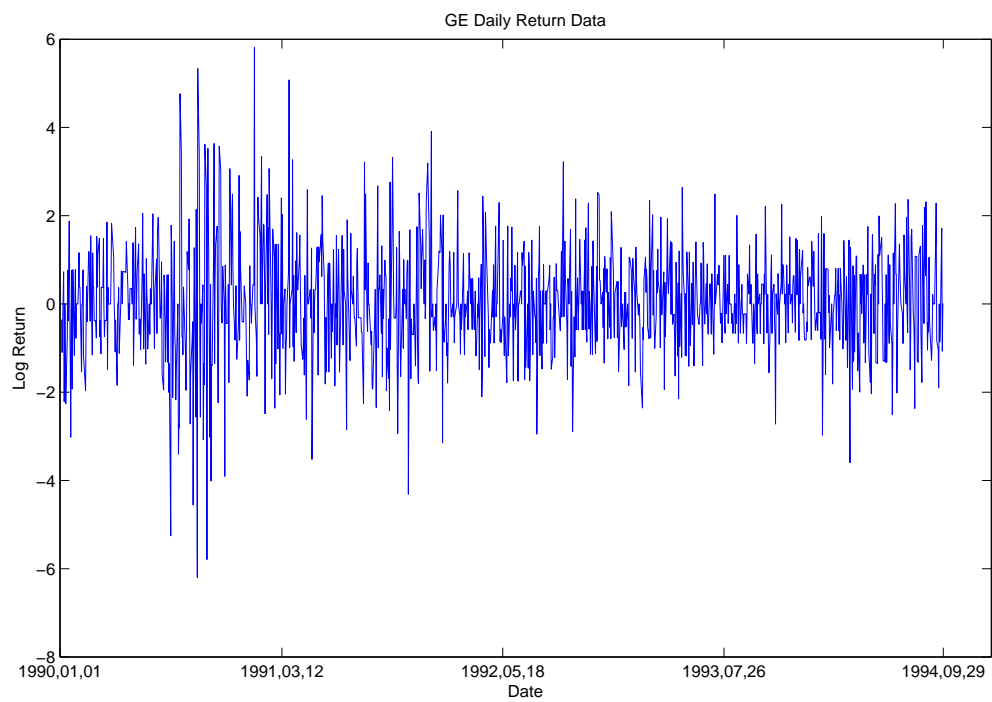


Figure 3.4: GENERAL ELECTRIC (NYSE: GE) daily return data from January 1, 1990 to September 29, 1994

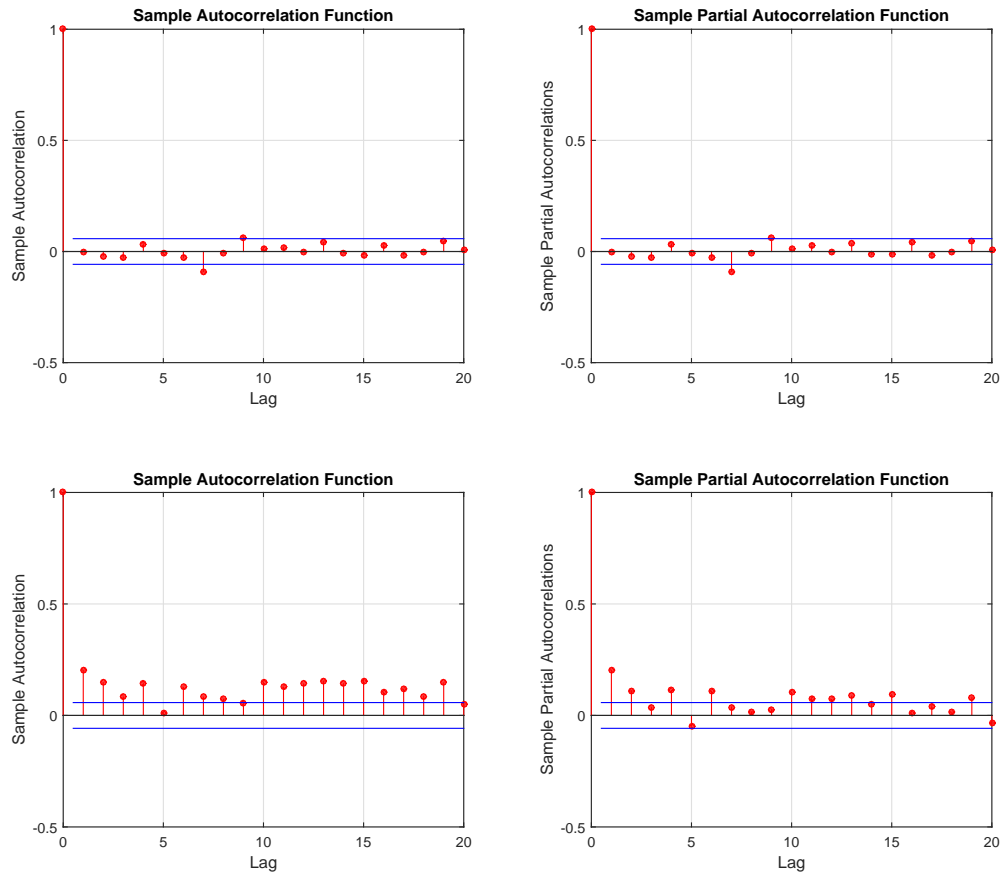


Figure 3.5: Sample ACF and PACF functions for GE Daily returns from January 1, 1990 to September 29, 1994. The first row: ACF and PACF of the returns; the second row: ACF and PACF of the squared returns.

Table 3.1: Descriptive Statistic of GE Daily Return Data

Mean	Standard Deviation	Skewness	Kurtosis	Min	Max
0.0433	1.2394	0.0092	5.2534	-6.3326	5.9952

3.5 GPRSV with Exogenous Factors

The GPRSV model with exogenous factors can be summarized as:

$$a_t = r_t - \mu = \sigma_t \epsilon_t \quad (3.4a)$$

$$v_t = \log(\sigma_t^2) = f(v_{t-1}, u_{t-1}) + \tau \eta_t \quad (3.4b)$$

$$f \sim \mathcal{GP}(m(x), k(x, x')) \quad (3.4c)$$

where τ is scale of variance process noise, and a_t , r_t , σ_t , ϵ_t , η_t and f have the same meaning as Equation 3.1. u_t is known exogenous factor data at time t . When modeling different financial time series, we can take different information into account. In Figure 3.6, we show the graphical model representation of a Gaussian process regression stochastic volatility model with exogenous factor. There are many macro-finance variables besides the asset return series itself which can be applied to volatility modeling, but how to manage fitting these variables can be complicated. The ultimate purpose of adding exogenous factors is to improve the forecasting performance of the model. If we treat these extra factors as simple linear regression variables, it can lead to the problem of over fitting and introducing too many parameters in which case learning such a model would be too difficult. By putting the exogenous factors in a Gaussian process, we can avoid the above problems. In [5], the authors investigated to use the mood measurements derived from large-scale Twitter feeds to predict the value of the Dow Jones Industrial Average (DJIA). They obtained an accuracy of 86.7% in predicting the daily up and down changes in the closing values of the DJIA and a reduction of the Mean Average Percentage Error (MAPE) by more than 6%. Although this is not a volatility forecast case, we can see that there are rich exogenous factors we can explore to improve our GPRSV models.

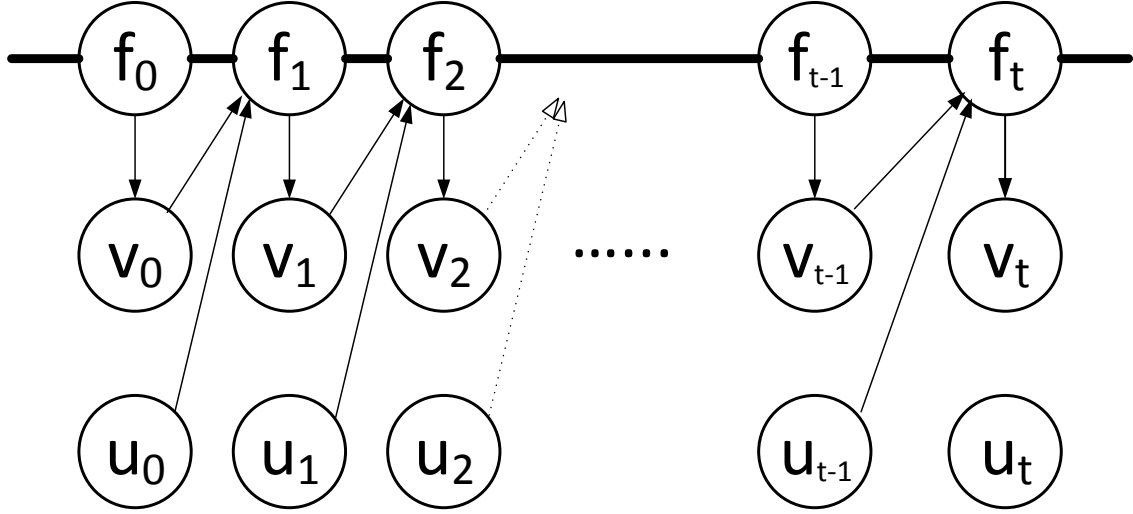


Figure 3.6: Graphical model representation of a Gaussian process regression stochastic volatility model with exogenous factor u_t , in this figure the nodes of observation variable (return series) is omitted. v_t is the hidden variable (volatility) at time t . f_t is the Gaussian process sampled function value at time t , u_t is the exogenous factor, and the thick horizontal line represent fully connected nodes. Hyper-parameters of the Gaussian process are omitted in the figure as well. In GPRSV models we can add more than one exogenous factors, we show one factor case for clarity

3.6 Chapter Summary

In this chapter, we introduced the GPRSV models, and discussed the advantages compared with traditional parametric volatility models. We gave the process how to build a GPRSV model, and discussed the model's structure. One possible way of improving the basic GPRSV model is to introduce exogenous factors into the model.

Chapter 4

GPRSV Models Inference

In this chapter, we discuss the problem of learning the proposed GPRSV models. It is much more challenging to learn a GPRSV model than its parametric competitors. As we have discussed a GPRSV model can be viewed as an instant of Gaussian process state-space model. Learning a GP-SSM model is a more complex task than learning a standard HMM model. Gaussian process dynamics are embedded in the hidden state transition equation. In GP-SSM models, we need to estimate two types of unknown variables from training data: the hidden states trajectory and the Gaussian process dynamics. In GPRSV models, the hidden state is the logarithm of variance and the Gaussian process dynamics are the hyper-parameters in the mean, covariance, likelihood functions. We specify the forms of these functions before the learning step, the only unknown part is the hyper-parameters. Jointly learning the hidden states trajectory, the unknown Gaussian process regression function values and hyper-parameters is computational challenging. Our approach is marginalizing out the Gaussian process regression function values, and then jointly learning the hidden volatility states and hyper-parameters using Monte Carlo methods.

4.1 Bayesian Inference for State-Space Models

In time series analysis settings, the task of learning a state-space model include two parts: the hidden system states and the parameters of the model. Bayesian Inference is to use the posterior distribution to answer questions of our interest. We have the prior

knowledge of the hidden states and the model's parameters. Although we can not update our knowledge of hidden states and parameters directly, we have the observation variable (return for volatility modeling problems) which is related to the hidden system states and unknown parameters with the likelihood function. With more observations come, we can update the posterior distribution applying Bayes' theorem.

First we consider estimate the unknown parameters. We denote the proposed model's parameters as a vector $\boldsymbol{\theta}$ and the observations $y_{1:T}$. We can consider estimation as a special case of inference as the parameter is our target of the posterior distribution. We use Bayes' rule to describe the estimation problem as:

$$p(\boldsymbol{\theta}|y_{1:T}) = \frac{p(y_{1:T}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(y_{1:T})} \quad (4.1)$$

where $p(\boldsymbol{\theta})$ is the distribution quantifying the modeler's belief of parameters' value before any observation data comes, $p(\boldsymbol{\theta}|y_{1:T})$ is the poster distribution, and $p(y_{1:T}|\boldsymbol{\theta})$ is the likelihood and $p(y_{1:T})$ is the marginal likelihood.

The optimized $\boldsymbol{\theta}$ value can be achieved by the Maximum-a-Posteriori (MAP) point estimate,

$$\boldsymbol{\theta}^{MAP} = \arg \max_{\boldsymbol{\theta}} p(y_{1:T}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

More conveniently we can use the logarithm posterior,

$$\boldsymbol{\theta}^{MAP} = \arg \max_{\boldsymbol{\theta}} \log(p(y_{1:T}|\boldsymbol{\theta})p(\boldsymbol{\theta}))$$

When the prior $p(\boldsymbol{\theta})$ is equal to constant, the the MAP solution becomes the maximum likelihood (ML) solution.

$$\boldsymbol{\theta}^{MAP} = \arg \max_{\boldsymbol{\theta}} p(y_{1:T}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = \boldsymbol{\theta}^{ML} = \arg \max_{\boldsymbol{\theta}} \log(p(y_{1:T}|\boldsymbol{\theta}))$$

This ML method is widely used to the parameter estimation problems in time series modeling, but one possible drawback of this approach is the over-fitting problem.

Secondly, we would like to discuss learning the hidden system states trajectory. If we assume that we know the parameters or we have estimated the them using ML methods, the distribution of the hidden states $x_{1:t}$ can be estimated iteratively. We can decompose

Equation 2.9 recursively as:

$$\begin{aligned}
p(x_{1:t}|y_{1:t}; \boldsymbol{\theta}) &= \frac{p(y_t|x_{1:t}, y_{1:t-1}; \boldsymbol{\theta})p(x_{1:t}|y_{1:t-1}; \boldsymbol{\theta})}{p(y_t|y_{1:t-1}; \boldsymbol{\theta})} \\
&= \frac{p(y_t|x_{1:t}, y_{1:t-1}; \boldsymbol{\theta})p(x_{1:t-1}|y_{1:t-1}; \boldsymbol{\theta})}{p(y_t|y_{1:t-1}; \boldsymbol{\theta})}p(x_{1:t-1}|y_{1:t-1}; \boldsymbol{\theta}) \\
&= \frac{p(y_t|x_t; \boldsymbol{\theta})p(x_t|x_{t-1}; \boldsymbol{\theta})}{p(y_t|y_{1:t-1}; \boldsymbol{\theta})}p(x_{0:t-1}|y_{1:t-1}; \boldsymbol{\theta})
\end{aligned} \tag{4.2}$$

From Equation 4.2 we can find the recursive relationship between $p(x_{1:t}|y_{1:t}; \boldsymbol{\theta})$ and $p(x_{0:t-1}|y_{1:t-1}; \boldsymbol{\theta})$. This is the foundation for designing recursive algorithms to solve the problem of learning the hidden states.

4.2 Monte Carlo Methods

We have discussed the general idea of Bayesian inference for state-space models, but the two parts: hidden states and unknown parameters are not learned together. As we discussed in Chapter 3, GP-SSMs provide us a flexible framework for time series analysis. However this great descriptive power comes with the expense of a computational cost. It is impossible to obtain analytic solution like learning linear Gaussian residuals state-space models using Kalman filter [31]. Our solution to this problem is applying the Monte Carlo methods to simulate the unknown densities. The core idea of Monte Carlo methods is to draw a set of i.i.d. samples (particles) from a target distribution density, and use the samples to approximate the target density with point-mass function [1].

$$p_N(x) = \frac{1}{N} \sum_{i=1}^N \delta_{x^{(i)}}(x) \tag{4.3}$$

where $x^{(i)}$ is the i th sample, N is the number of samples, and $\delta_{x^{(i)}}(x)$ denotes the Delta-Dirac mass function value at $x^{(i)}$. Further more we can approximate integrals of f which is function of interest. $I(f)$ can be achieved with tractable sums $I_N(f)$,

$$I_N(f) = \frac{1}{N} \sum_{i=1}^N f(x^{(i)}) \xrightarrow[N \rightarrow \infty]{a.s.} I(f) = \int f(x)p(x)dx \tag{4.4}$$

In a standard Gaussian process regression problem setting, the inputs and outputs are fully observed, so the regression function value f can be learned using exact Bayesian inference methods, for the details how to do that we refer readers to [45]. For GP-SSMS inference, both the hidden states and the Gaussian process dynamics are unknown. Direct learning the hyper-parameters, hidden states and Gaussian process function values is a challenging task. Most of previous work on inference GP-SSMs focused on filtering and smoothing hidden variables without jointly learning the Gaussian process hyper-parameters. In [35], a novel particle Markov chain Monte Carlo (particle MCMC) algorithm, particle Gibbs with ancestor sampling (PGAS) was proposed. In [19], Frigola et al. apply the algorithm to learn a GP-SSM’s hidden states and Gaussian process dynamics jointly. In [55], a regularized Auxiliary Particle Filter which the authors named as Regularized Auxiliary Particle Chain Filter (RAPCF) was introduced. The RAPCF algorithm belongs to the sequential Monte Carlo (SMC) methods.

To learn GPRSV models using the PGAS and RAPCF algorithms, we marginalize out the Gaussian process regression function value f first. Then we can targeting jointly learning the hidden states and hyper-parameters together. After marginalizing out f , the models become non Markovian state-space models. Traditional filter and smooth methods are not capable to learn such models. The Monte Carlo methods based algorithms we presented here provide us a powerful tool to solve this problem. Both of the hidden states and parameters can be represented using particles associated with normalized weights.

4.2.1 Sequential Monte Carlo Methods

Sequential Monte Carlo (SMC) concept was first introduced by Gordon et al. [23] in 1993, and Del Moral [13] gave the first consistency proof for that in 1996. SMC is also called as particle filter in some applications. Ever since its introduction, SMC method has been widely used in many areas to solve the problem of inference complex nonlinear models. References for SMC methods applied to engineers, finance and economics are [15], [11] and [12]. In Economic study, economists introduced many dynamic stochastic general equilibrium (DSGE) models to many real world time series which often exhibit strong non-Gaussian and time-varying behaviors. In this scenario, SMC methods are used to learn nonlinear, non-Gaussian state-space models. For volatility modeling research, Kim et al. first learned a stochastic volatility model using particle filter in [32]. In [55], Wu et

Algorithm 1 RAPCF for GPRSV Model

- 1: **Input:** return data $r_{1:T}$, number of particles N , shrinkage parameter $0 < \lambda < 1$, prior $p(\boldsymbol{\theta})$.
- 2: Remove linear dependence from $r_{1:T}$ to get the residuals $a_{1:T}$.
- 3: Sample N parameter particles from the prior, and set initial importance weights, $W_0^i = 1/N$
- 4: **for** $t = 1$ to T **do**
- 5: Shrink parameter particles towards empirical means

$$\bar{\boldsymbol{\theta}}_{t-1} = \sum_{i=1}^N W_{t-1}^i \boldsymbol{\theta}_{t-1}^i \quad (4.5a)$$

$$\tilde{\boldsymbol{\theta}}_t^i = \lambda \boldsymbol{\theta}_{t-1}^i + (1 - \lambda) \bar{\boldsymbol{\theta}}_{t-1} \quad (4.5b)$$

- 6: Compute expected states:

$$\boldsymbol{\mu}_t^i = \mathbb{E}(v_t | \tilde{\boldsymbol{\theta}}_t^i, v_{1:t-1}^i) \quad (4.6)$$

- 7: Compute important weights

$$g_t^i \propto W_{t-1}^i p(a_t | \boldsymbol{\mu}_t^i, \bar{\boldsymbol{\theta}}_t^i) \quad (4.7)$$

- 8: Resample N auxiliary indices $\{j\}$ according to $\{g_t^i\}$.
- 9: Propagate the chains of v_t forward, $\{v_{1:t-1}^j\}_{j \in J}$.
- 10: Add jitter: $\boldsymbol{\theta}_{t-1}^i \sim \mathcal{N}(\boldsymbol{\theta}_t^j, (1 - \lambda^2) \mathbf{V}_{t-1})$, and \mathbf{V}_{t-1} is empirical covariance of $\boldsymbol{\theta}_{t-1}$.
- 11: Propose new states $v_t^j \sim p(v_t | \boldsymbol{\theta}_t^j, v_{1:t-1}^j, a_{1:t-1})$
- 12: Adjust weights with newly proposed states:

$$W_t^i \propto p(a_t | v_t^j, \boldsymbol{\theta}_t^j) / p(a_t | \boldsymbol{\mu}_t^j, \tilde{\boldsymbol{\theta}}_t^j) \quad (4.8)$$

- 13: **end for**
 - 14: **Output:** particles of $v_{1:T}^j$, particles of $\boldsymbol{\theta}_t^j$ and particle weights W_t^j .
-

al. proposed the RAPCF algorithm to learn a Gaussian process based GARCH model. Here, we can modify the RAPCF algorithm to learn our GPRSV model.

In Algorithm 1, we present our version of RAPCF for jointly learning the hidden states and Gaussian process hyper-parameters for the GPRSV models.

4.2.2 Particle MCMC methods

Algorithm 2 PGAS for GPRSV models

- 1: **Input:** the return data $r_{1:T}$, the iteration times l .
 - 2: Remove linear dependence from $r_{1:T}$ to get the residuals $a_{1:T}$.
 - 3: Set $\theta[0]$ and $v_{1:T}[0]$ arbitrarily.
 - 4: **for** $l \geq 1$ **do**
 - 5: Draw particles of $\theta[l]$ conditionally on $v_{1:T}[l-1]$ and $a_{1:T}$.
 - 6: Run CPF-AS, targeting $p(v_{1:T}|\theta[l], a_{1:T})$, conditionally on $v_{1:T}[l-1]$.
 - 7: Sample k with $p(k=i) = w_T^i$ and set $v_{1:T}[l] = v_{1:T}^k$.
 - 8: **end for**
 - 9: **Output:** the hidden volatility $v_{1:T}$ and the hyper-parameter θ .
-

Besides SMC methods, we can learn the GPRSV models using Markov chain Monte Carlo methods as well. MCMC played significant important role in statistics, economics, computing science and physics over the last three decades. One of the MCMC methods: the Metropolis algorithm was considered to be one of the ten algorithms which have had the greatest influence on the development and practice of science and engineering in the 20th century [3]. In this section we focus on particle Markov chain Monte Carlo methods for learning the GPRSV models. The particle MCMC method was first introduced in [2]. The idea of particle MCMC is to use of a certain SMC sampler to construct a Markov kernel leaving the joint smoothing distribution invariant. In [35], Lindsten et al. proposed the PGAS algorithm. Frigola et al. applied the PGAS algorithm to the problem of Gaussian process state-space models inference [19]. Based their results, the PGAS algorithm is suitable to learn a non Markovian state-space model. In Algorithm 2, we show the PGAS algorithm to learn a GPRSV model. The main block of the PGAS algorithm is the conditional particle filter with ancestor sampling (CPF-AS) which is a particle filter like procedure. The CDF-AS part is presented in Algorithm 3. The two types of methods both can learn the proposed GPRSV models. Particle MCMC

Algorithm 3 CPF-AS conditional on $v'_{1:T}$

```
1: Initialize( $t=1$ )
2:   Sample  $v_1^i \sim p_1^\theta(x_1)$ , for  $i = 1, 2, \dots, N - 1$ .
3:   Set  $v_1^N = v'_1$ .
4:   Set  $w_1^i = W_1^\theta(v_1^i)$ , for  $i = 1, 2, \dots, N$ .
5:   for  $t \geq 2$  do
6:     Sample  $e_t^i \sim \text{Discrete}(\{w_{t-1}^j\}_{j=1}^N)$ , for  $i = 1, 2, \dots, N - 1$ .
7:     Sample  $v_t^i \sim p_t^\theta(v_t|v_{1:t-1}^{e_t^i})$ , for  $i = 1, 2, \dots, N - 1$ .
8:     Set  $v_t^N = v'_t$ .
9:     Sample  $e_t^N$  with  $w_{t-1}^i f_\theta(v'_t|v_{t-1}^i)$ .
10:    Set  $v_{1:t}^i = \{v_{1:t-1}^{e_t^i}, v_t^i\}$  and  $w_t^i = W_t^\theta(v_{1:t}^i)$ , for  $i = 1, 2, \dots, N$ .
11:   end for
```

methods are offline algorithms which are more accurate than the SMC methods, but the disadvantage is they are slower than SMC methods. In our experiment, we find that SMC method can provide us desired accuracy results. In Chapter 5, the empirical financial data are learned with SMC methods.

4.3 Chapter Summary

In this chapter, we discussed the inference methods for our proposed GPRSV models. Our approach is taking the SMC and particle MCMC algorithms to jointly learning the volatility and hyper-parameters posterior distribution. The advantage of our method is that the hidden states and the model's parameters are estimated simultaneously.

Chapter 5

Volatility Analysis with GPRSV Models

In this chapter, we apply both the simulated and empirical financial data to demonstrate our GPRSV models and inference methods. First, to prove the RAPCF and PGAS algorithms we discussed in Chapter 4 can be used to learn the proposed GPRSV models, we generated sets of simulated data. The results show that the algorithms can effectively learn the nonparametric models. Then we continue to demonstrate the GPRSV models with real financial data. The empirical data sets are used to demonstrate the forecasting performance our models.

5.1 Simulated Data

We generated ten synthetic data sets of length $T = 200$ according the equations in Chapter 3. Based on Equation (3.1), we sample our hidden state transition function f from a Gaussian process prior. We specified that the mean function $m(x_t)$ and the covariance function $k(y, z)$ as follow:

$$m(x_t) = ax_{t-1} \tag{5.1a}$$

$$k(y, z) = \gamma \exp(-0.5|y - z|^2/l^2) \tag{5.1b}$$

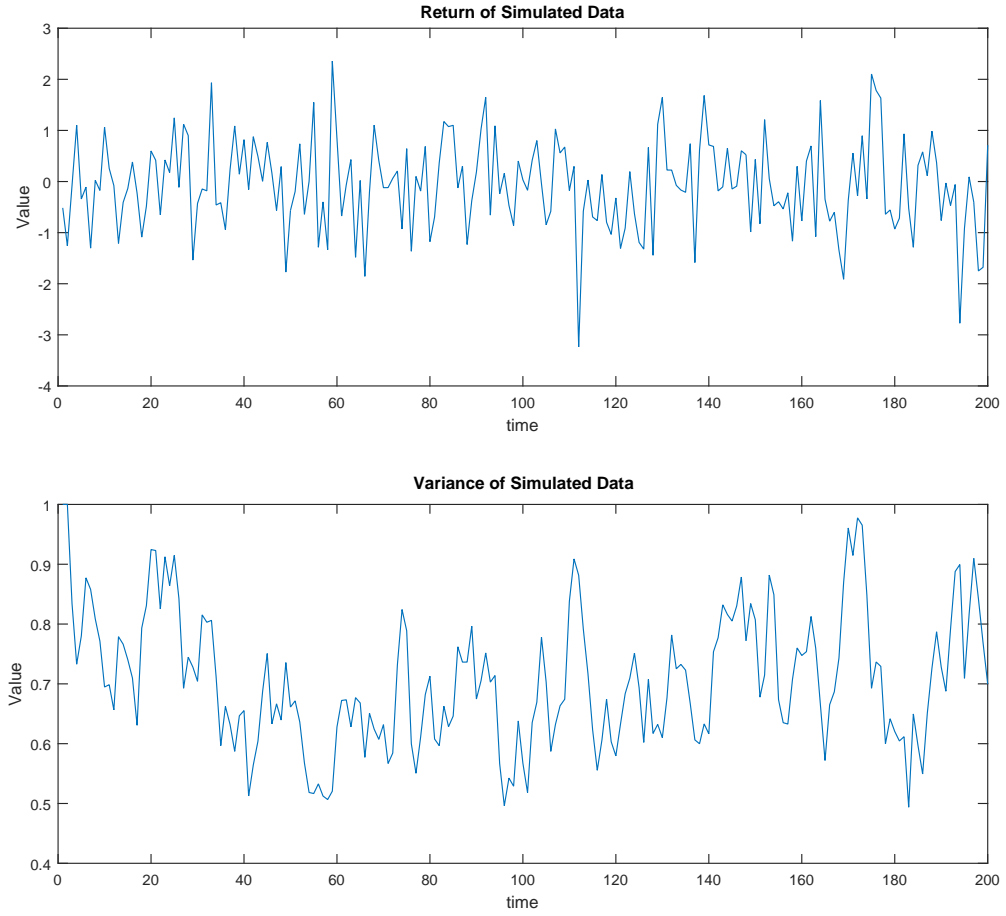


Figure 5.1: Return and variance values of one set of generated simulation data. Total number of observations is 200. The data are generated follow basic GPRSV model, and the Gaussian process mean and covariance functions are specified in Equation 5.1.

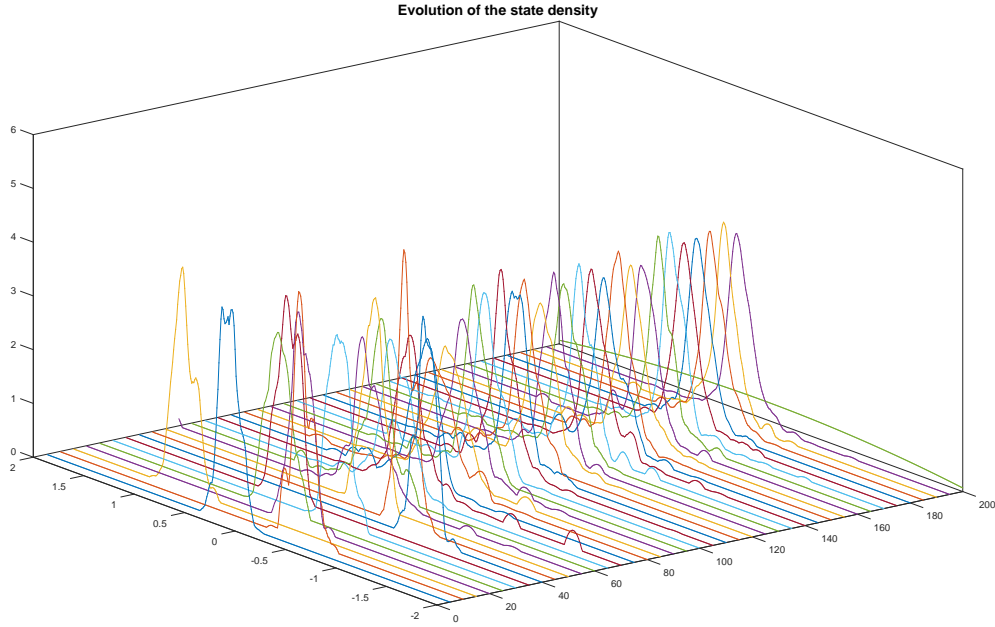


Figure 5.2: Estimated hidden states densities of simulated Data. There are 200 iteration steps for the simulated data, and we plot every 5 densities in this figure. The distribution densities are generated using particles and weights as described in Algorithm 1.

where a is the mean equation hyper-parameter, and γ and l are the covariance hyper-parameters.

In Figure 5.1, we show one set of the simulated data set. The return and variance are plotted. We applied the RAPCF algorithm to jointly learn the hidden states and the hyper-parameters. As we have explained in Chapter 4, although the Gaussian process regression parameters are fixed, we still learn these values using particles. The prior knowledge is given and at each iteration we update the unknown hidden state and parameter distribution with coming observation. The first 50 iteration is used as burn in period. The particles or samplers number we used for these simulated data sets is 1000. The shrink parameter in RAPCF $\lambda = 0.99$ is used.

One of the advantages of our approach for learning is the hidden states and Gaussian process dynamics are jointly learned together using particles. At each iteration step we

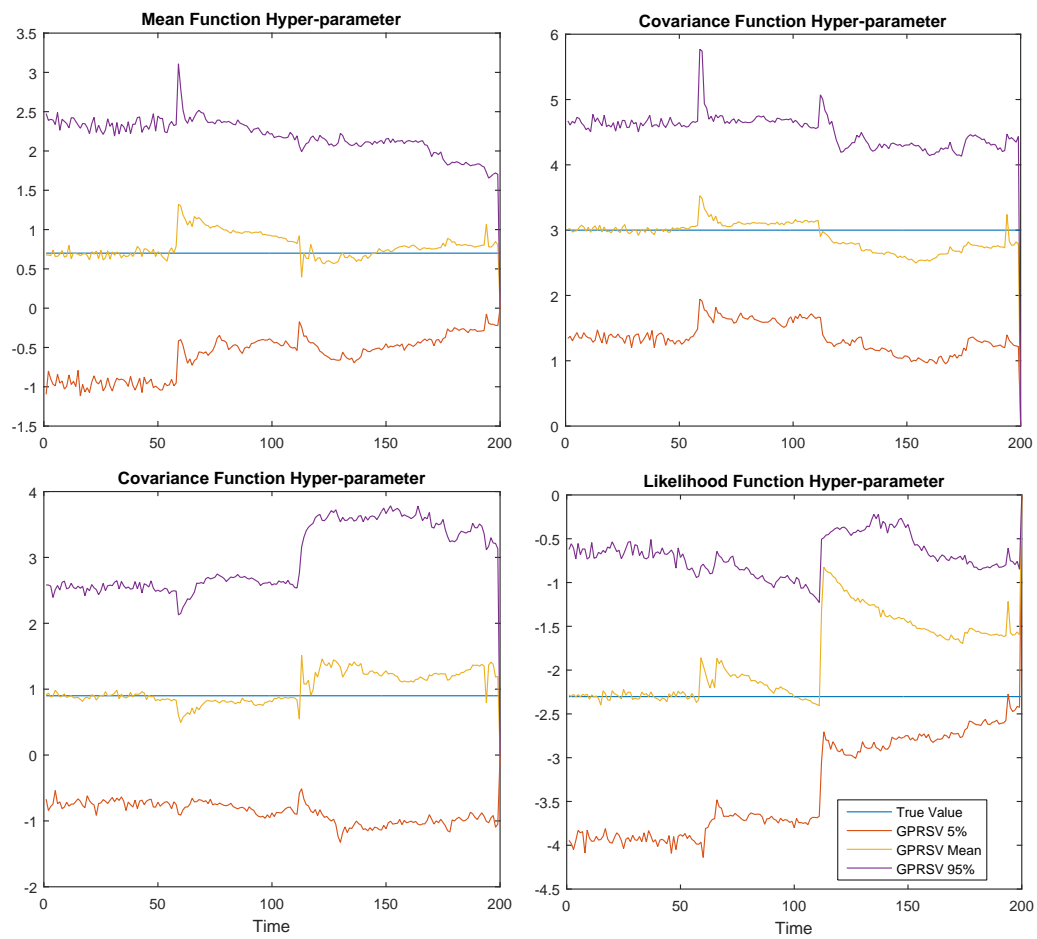


Figure 5.3: Results of the Gaussian process hyper-parameters. The hyper-parameters are learned from RAPCF algorithm using particles.

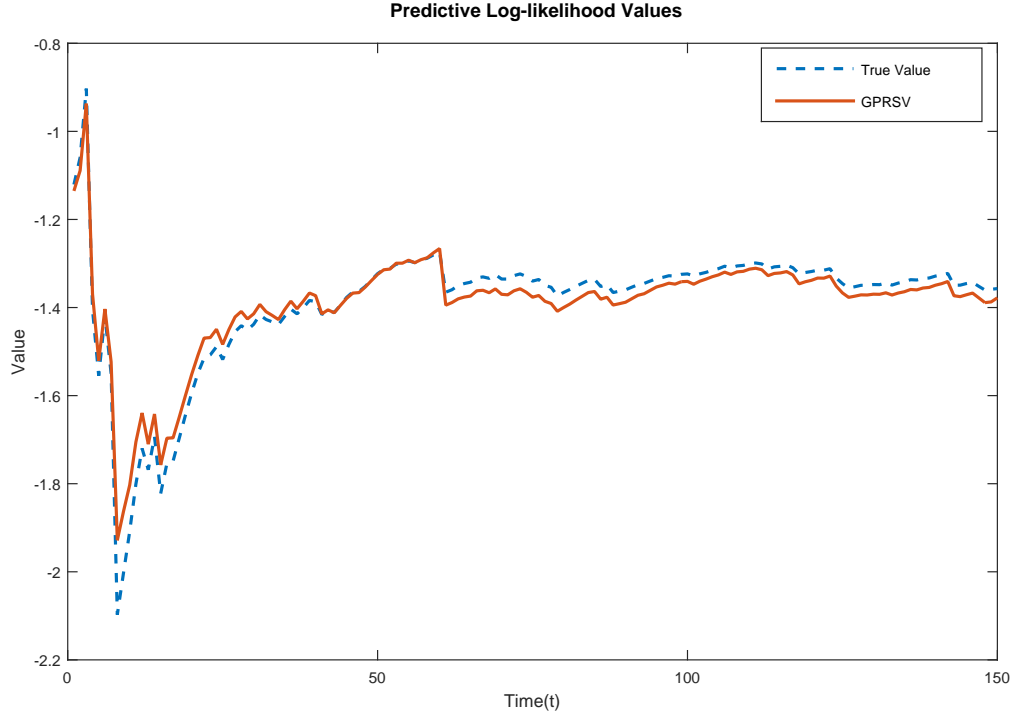


Figure 5.4: RAPCF algorithm learned predictive Log-likelihood value are compared with true value calculate from Equation 3.1. We discard the first 50 burn in iterations. The RAPCF learned predictive Log-likelihood result proves that the algorithm can successfully learn the hidden volatility.

can approximate the hidden state distribution. In Figure 5.2, we plot the hidden state variable density at every 5 iteration step. In Figure 5.3, we plots of the expected value and 90% posterior intervals for all the hyper-parameters learned from particles. Although the hyper-parameters are not random variables, we can learn those values using particles.

In Figure 5.4, we show the results of predictive log-likelihood. At each iteration step, we can calculate the log-likelihood with the learned hidden state value and the observation value. Compared with the values obtained from the true hidden state and observation, our particle filter based learned results are close enough. With more particles used the accuracy of results can improve. Based on our experiment, 800 to 1000 particles are enough to learn these sets of GPRSV models. With different Gaussian process function

forms and numbers of hyper-parameters, the more particles may be needed.

5.2 Empirical Data

In this part we apply our GPRSV model to the real financial data, and compared our model with a class of GARCH models which are the traditional parametric volatility models. We use the realized volatility calculated from intra day data as the proxy for the true daily volatility value. The process of the comparing is as follow : first we use in-sample data to train both the two typed models, and then we estimate the volatility values for the out-of-sample period. Finlay we use the average loss function values criterion to rank the models.

5.2.1 Volatility Forecast Evaluation

The evaluation of model's forecasting performance is the key step in the empirical data experiment. In finance study, it is rare to find a method that is consistently superior for forecasting the price of financial assets, and empirical studies are often inconclusive. The problem of volatility forecasting is that we cannot observe the variance directly. The evaluation of volatility forecasting can be complicated. There is many metrics to evaluate different forecast models. One of the most popular approach is using a particular statistical loss function, the model which achieved a minimized loss function value is the best forecasting model [7]. There are extensive choices of lost functions. We adopt a class of statistical loss functions instead of a particular one. Here we denote the unbiased ex post proxy of conditional variance as σ_{t+m}^2 and a model's forecast value as $\hat{\sigma}_{t+m}^2$. We take the following loss functions:

$$MSE : L(\hat{\sigma}_{t+m}^2, \sigma_{t+m}^2) = n^{-1} \sum_{t=1}^n (\hat{\sigma}_{t+m}^2 - \sigma_{t+m}^2)^2 \quad (5.2)$$

$$MAD_1 : L(\hat{\sigma}_{t+m}, \sigma_{t+m}) = n^{-1} \sum_{t=1}^n |\hat{\sigma}_{t+m} - \sigma_{t+m}| \quad (5.3)$$

$$MAD_2 : L(\hat{\sigma}_{t+m}^2, \sigma_{t+m}^2) = n^{-1} \sum_{t=1}^n |\hat{\sigma}_{t+m}^2 - \sigma_{t+m}^2| \quad (5.4)$$

$$MLAE_1 : L(\hat{\sigma}_{t+m}, \sigma_{t+m}) = n^{-1} \sum_{t=1}^n \log(|\hat{\sigma}_{t+m}^2 - \sigma_{t+m}^2|) \quad (5.5)$$

$$MLAE_2 : L(\hat{\sigma}_{t+m}, \sigma_{t+m}) = n^{-1} \sum_{t=1}^n \log(|\hat{\sigma}_{t+m} - \sigma_{t+m}|) \quad (5.6)$$

$$QLIKE : L(\hat{\sigma}_{t+m}^2, \sigma_{t+m}^2) = n^{-1} \sum_{t=1}^n (\hat{\sigma}_{t+m}^2 / \sigma_{t+m}^2 + \log \sigma_{t+m}^2) \quad (5.7)$$

$$HMSE : L(\hat{\sigma}_{t+m}^2, \sigma_{t+m}^2) = n^{-1} \sum_{t=1}^n (\hat{\sigma}_{t+m}^2 / \sigma_{t+m}^2 - 1)^2 \quad (5.8)$$

These loss functions include the typical mean squared errors, mean absolute deviation criteria and logarithmic loss functions which are more used in econometric literature.

Another problem with volatility forecasting evaluation is that we do not have the real true volatility value in the loss functions. We have to use some proxy to standard for the real value. Some proxy like the square of return can be quite inaccurate. In our experiment, we apply the high frequency data to calculate the “realized volatility” [51]. In our experiment we want to model daily return series volatility, so we can use the intra-daily data as the high frequency data to estimate the daily volatility. Compared with the squared return, realized volatility is considered to more precise proxy for volatility forecast evaluation.

5.2.2 Data

The data set we analyzed is the IBM stock daily closing price data ¹. We used the daily closing price as our input. The data period is from January 1, 1988 to September 14, 2003. There is 1000 observations in total, the first 252 ones (from January 1, 1988 to September 27, 2001) are used as in-sample part for training purposes and the rest observations (from September 28, 2001 to September 14, 2003) are used as out-of-sample for evaluating forecasting performance.

Follow the process we proposed in Chapter 3, we can build our basic GPRSV model with the IBM return data, we measure that the in-sample data mean value is quite small

¹The data set can be obtained from YAHOO! finance website <http://finance.yahoo.com/>

and the standard deviation is around one. The detailed statistics are presented in Table 5.1.

Table 5.1: Descriptive Statistic of IBM Daily Return Data

Mean	Standard Deviation	Skewness	Kurtosis	Min	Max
0.1319	1.7840	0.9583	11.0807	-9.6498	12.0474

5.2.3 Results

We compare our GPRSV model with two standard parametric volatility models: GARCH and GJR-GARCH. For the parametric models, we use Kevin Sheppard's Oxford MFE Toolbox to estimate parameters and make prediction. For GPRSV model, the Gaussian process dynamics are specified as follows: the mean function is $m(x_t) = ax_{t-1}$ and the covariance function is the squared exponential covariance function $k(y, z) = \gamma \exp(-0.5|y - z|^2/l^2)$. The hyper-parameters include a , γ , l and likelihood function parameter $\log(sn)$. The learned parameters are presented in Table 5.2.

Table 5.2: Estimated GPRSV Model Hyper-parameters Results for IBM Daily Return Data

a	γ	l	$\log(sn)$
1.8777	3.3064	1.3044	-1.7664

In Figure 5.5, we plot the learned volatility values of GARCH, GJR-GARCH, standard SV and GPRSV models. In Table 5.3, we give the results of three models loss function values with realized volatility as proxy. The GPRSV achieved the lowest average loss function values for all functions except for MSE loss function. The GJR-GARCH obtained the lowest MSE loss function value. Our GPRSV model's performance is the best based on the loss function values.

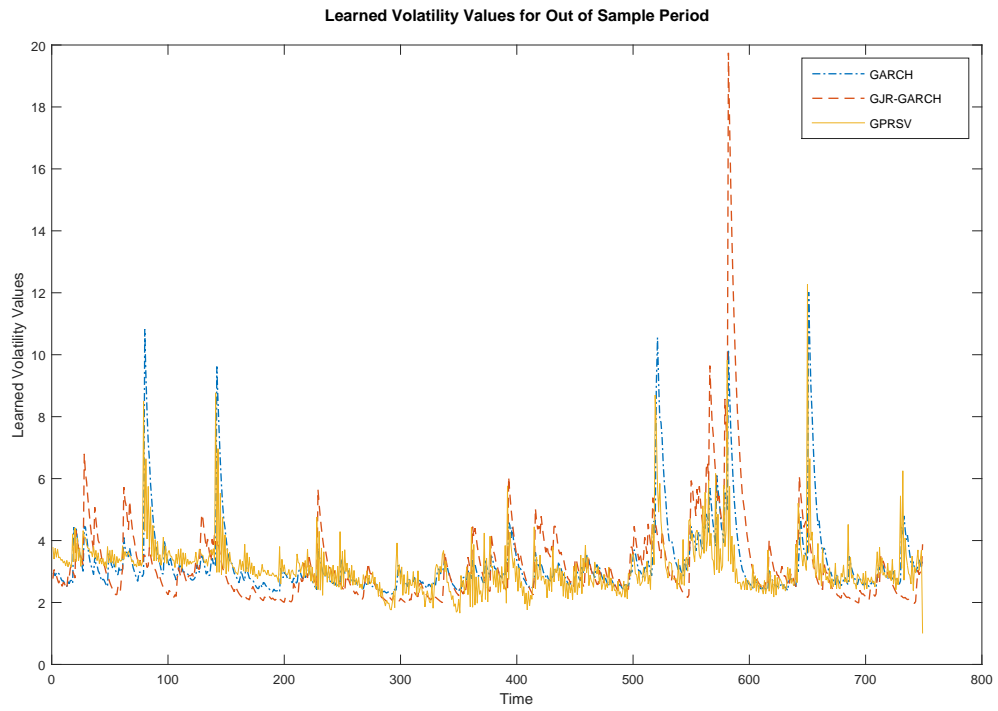


Figure 5.5: The volatility values learned from the three models are plotted. GARCH and GJR-GARCH results are both estimated using Kevin Shepperd's Oxford MFE Toolbox. GPRSV model results are learned using RAPCF algorithm

5.3 Chapter Summary

In this chapter we conducted experiments with both simulated and empirical data. Based on our results, the modified RAPCF algorithm can successfully learn a GPRSV model. We use loss functions to compare model forecasting performance, and the realized volatility is adopted as true volatility proxy instead of squared return. Our GPRSV model can provide better forecasting performance compared with standard parametric volatility models.

Table 5.3: Results of IBM Volatility Forecast Using Loss Functions

Model	<i>MSE</i>	<i>MAD</i> ₁	<i>MAD</i> ₂	<i>MLAE</i> ₁	<i>MLAE</i> ₂	<i>HMSE</i>	<i>QLIKE</i>
GARCH	3.656	3.7870	1.0751	0.7876	-0.2536	9.3797	2.13466
GJR-GARCH	0.7988	3.7920	1.0682	0.7717	-0.2577	6.9648	2.0917
SV	4.5024	3.800	1.077	0.7944	-0.2466	9.1739	2.1312
GPRSV	3.2330	3.3607	0.9682	0.6277	-0.3918	1.8617	1.8549

Note: The lowest loss function value is marked using the bold text. Except for MSE, GPRSV model obtained the lowest loss function values for all others. GJR-GARCH obtain the lowest MSE loss function. The volatility proxy is the 65-minutes sampled realized volatility.

Chapter 6

Conclusion and Future Work

In this thesis, we proposed a Gaussian process regression based volatility model for the problem of analyzing and predicting the time varying volatility of financial time series data. After we introduced the GPRSV model, we gave a solution to jointly learning the hidden volatility states and the Gaussian process dynamics. Also we discussed the possible way of adding exogenous factors to improve its forecasting performance.

Based on our experiment results, we can successfully learn the Gaussian process regression stochastic volatility model's hidden states and hyper-parameters. Also the Gaussian process regression based stochastic volatility models can achieve better performance compared with the standard economic parametric models.

For future research, there are several possible directions. First, we introduced the GPRSV framework to analyze the time varying volatility, and discussed that we can add exogenous factors to improve the forecasting performance. There are many factors that can be used. In different applications, depending on the data we analyze, we can study what information is more relevant. In our experiment, we used only one covariance function. The squared-exponential (SE) function gave us the best performance for our data set, but we can apply different covariance functions besides the most common used SE function. There are many more choices for molders to explore. Second, we think the way of learning the GPRSV model can be applied to other Gaussian process state-space models as well. Third, we can extend our Gaussian process regression stochastic volatility model to other financial data analyze. Any application using state-space models to analyze the data is also applicable using Gaussian process state-space model. Our

approach is marginalizing out the Gaussian process regression function values and then jointly learning the hidden states and the hyper-parameters. This methodology can be applied to those applications. Especially the learning procedure we used.

References

- [1] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43, 2003.
- [2] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.
- [3] Isabel Beichl and Francis Sullivan. The metropolis algorithm. *Computing in Science and Engineering*, 2(1):65–69, 2000.
- [4] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.
- [5] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.
- [6] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, April 1986.
- [7] Christian Brownlees, Robert Engle, and Bryan Kelly. A practical guide to volatility forecasting through calm and storm. *Journal of Risk*, 14(2):1–20, 2011.
- [8] Jun Cai. A markov model of switching-regime ARCH. *Journal of Business and Economic Statistics*, 12(3):309–316, 1994.
- [9] John Campbell, Andrew Wen-Chuan Lo, and Archie Craig MacKinlay. *The econometrics of financial markets*, volume 2. princeton University press Princeton, NJ, 1997.

- [10] National Research Council. *Frontiers in Massive Data Analysis*. The National Academies Press, Washington, DC, 2013.
- [11] Drew Creal. A survey of sequential monte carlo methods for economics and finance. *Econometric Reviews*, 31(3):245–296, 2012.
- [12] Dan Crisan and Arnaud Doucet. A survey of convergence results on particle filtering methods for practitioners. *Signal Processing, IEEE Transactions on*, 50(3):736–746, 2002.
- [13] Pierre Del Moral. Nonlinear filtering: Interacting particle solution. *Markov Processes and Related Fields*, 2(4):555–580, 1996.
- [14] Kresimir Demeterfi, Emanuel Derman, Michael Kamal, and Joseph Zou. More than you ever wanted to know about volatility swaps. *Goldman Sachs Quantitative Strategies Research Notes*, 1999.
- [15] Arnaud Doucet and Adam M Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.
- [16] Robert Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.
- [17] Emily Fox, Erik Sudderth, Michael Jordan, and Alan Willsky. Bayesian nonparametric methods for learning markov switching processes. 2010.
- [18] Roger Frigola, Yutian Chen, and Carl Rasmussen. Variational Gaussian process State-Space Models. In *Advances in Neural Information Processing Systems*, pages 3680–3688, 2014.
- [19] Roger Frigola, Fredrik Lindsten, Thomas B. Schön, and Carl E. Rasmussen. Bayesian inference and learning in gaussian process state-space models with particle MCMC. In *Advances in Neural Information Processing Systems 26*, pages 3156–3164, 2013.
- [20] Box George. *Time Series Analysis: Forecasting and Control*. Pearson Education India, 1994.

- [21] Zoubin Ghahramani. Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110553–20110553, 2012.
- [22] Lawrence Glosten, Ravi Jagannathan, and David Runkle. On the relation between the expected value and the volatility of the nominal excess return on stocks. *The Journal of Finance*, 48(5):1779–1801, 1993.
- [23] Neil Gordon, David Salmond, and Adrian Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. 140(2):107–113, 1993.
- [24] James Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57(2):357–384, 1989.
- [25] James Hamilton. *Time Series Analysis*. Princeton University Press, 1994.
- [26] James Hamilton and Raul Susmel. Autoregressive conditional heteroskedasticity and changes in regime. *Journal of Econometrics*, 64(1-2):307–333, 1994.
- [27] John Hull and Alan White. The pricing of options on assets with stochastic volatilities. *The Journal of Finance*, 42(2):281–300, 1987.
- [28] Gareth Janacek. Time series analysis forecasting and control. *Journal of Time Series Analysis*, 31(4):303–303, 2010.
- [29] Simon Julier and Jeffrey Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, Mar 2004.
- [30] Simon Julier, Jeffrey Uhlmann, and Hugh Durrant-Whyte. A new approach for filtering nonlinear systems. 3:1628–1632, 1995.
- [31] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [32] Sangjoon Kim, Neil Shephard, and Siddhartha Chib. Stochastic volatility: likelihood inference and comparison with ARCH models. *The Review of Economic Studies*, 65(3):361–393, 1998.

- [33] Jonathan Ko and Dieter Fox. Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models. *Autonomous Robots*, 27(1):75–90, 2009.
- [34] Neil Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.
- [35] Fredrik Lindsten, Michael Jordan, and Thomas Schön. Particle gibbs with ancestor sampling. *Journal of Machine Learning Research*, 15:2145–2184, 2014.
- [36] Benoit Mandelbrot. The Variation of Certain Speculative Prices. *The Journal of Business*, 36:394, 1963.
- [37] Juri Marcucci. Forecasting Stock Market Volatility with Regime-Switching GARCH Models. *Studies in Nonlinear Dynamics and Econometrics*, 9(4):1–55, December 2005.
- [38] Harry Markowitz. Portfolio selection*. *The Journal of Finance*, 7(1):77–91, 1952.
- [39] Bruce McElhoe. An assessment of the navigation and course corrections for a manned flyby of mars or venus. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-2(4):613–623, July 1966.
- [40] Allan McLeod and William Li. Diagnostic checking arma time series models using squared-residual autocorrelations. *Journal of Time Series Analysis*, 4(4):269–273, 1983.
- [41] Radford Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.
- [42] Radford Neal. Regression and classification using gaussian process priors. *Bayesian Statistics*, 6:475–501, 1998.
- [43] Daniel Nelson. Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica*, 59(2):347–70, March 1991.
- [44] Chicago Board of Option Exchange (CBOE). Vix index and volatility. <http://www.cboe.com/micro/vix-and-volatility.aspx>, January 2015.

- [45] Carl Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. the MIT Press, 2006.
- [46] Stanley Schmidt. The kalman filter-its recognition and development for aerospace applications. *Journal of Guidance, Control, and Dynamics*, 4(1):4–7, 1981.
- [47] Neil Shephard and Andersen Torben. Stochastic Volatility: Origins and Overview. Economics Series Working Papers 389, University of Oxford, Department of Economics, March 2008.
- [48] Stephen Taylor. Modelling financial time series, 1986.
- [49] Stephen Taylor. Modeling stochastic volatility: A review and comparative study. *Mathematical Finance*, 4(2):183–204, 1994.
- [50] Stephen Taylor. *Financial returns modelled by the product of two stochastic processes, a study of daily sugar prices*. Oxford University Press, 2005.
- [51] Andersen Torben, Tim Bollerslev, Francis Diebold, and Paul Labys. Modeling and forecasting realized volatility. *Econometrica*, 71(2):579–625, 2003.
- [52] Ruey Tsay. *Analysis of financial time series*. Wiley, 2010.
- [53] Ryan Turner. *Gaussian Processes for State Space Models and Change Point Detection*. PhD thesis, University of Cambridge, Cambridge, UK, July 2011.
- [54] Jack Wang, Aaron Hertzmann, and David Blei. Gaussian process dynamical models. In *Advances in neural information processing systems*, pages 1441–1448, 2005.
- [55] Yue Wu, José Miguel Hernández-Lobato, and Zoubin Ghahramani. Gaussian process volatility model. In *Advances in Neural Information Processing Systems*, pages 1044–1052, 2014.
- [56] Jean-Michel Zakoian. Threshold heteroskedastic models. *Journal of Economic Dynamics and Control*, 18(5):931–955, September 1994.