BAYESIAN CHANGEPOINT AND TIME-VARYING PARAMETER LEARNING IN REGIME SWITCHING VOLATILITY MODELS

by

Mustafa Serdar Yümlü

B.S., Computer Engineering, Yıldız Technical University, 2000

M.S., Computer Engineering, Boğaziçi University, 2004

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2015

BAYESIAN CHANGEPOINT AND TIME-VARYING PARAMETER LEARNING IN REGIME SWITCHING VOLATILITY MODELS

APPROVED BY:

Prof. Fikret S. Gürgen (Thesis Supervisor)	.....
Prof. Nesrin Okay	.....
Assoc. Prof. A. Taylan Cemgil	.....
Prof. Tunga Güngör	.....
Assoc. Prof. Songül Albayrak	.....

DATE OF APPROVAL: 25.05.2015

To my lovely children Ege & İnci Defne

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Prof. Fikret S. Gürgen for his endless support and patience, his invaluable guidance and suggestions throughout the study. I greatly appreciate him for his contribution. It was an invaluable experience for me to work with him.

I would like to express my special thanks to Prof. Nesrin Okay for her endless support, guidance in the field of economics, and for sparing her time for our continuous discussions in statistics.

I am grateful to Assoc. Prof. A. Taylan Cemgil for serving on my thesis committee and for his invaluable guidance and contributions in Bayesian time series and changepoint modeling.

I would like to present my thanks to Prof. Tunga Güngör and Assoc. Prof. Songül Albayrak for their participation to my thesis jury.

This research has been partially supported by Boğaziçi University Scientific Research Fund (BAP) with grants 10A0102 and 12A01D3.

Finally, I would like to express my deepest thanks to my wife Mehnur, my son Ege (8) and my daughter İnci Defne (4), my parents (Abdullah & Hamdiye), my brother (Murat) and my grandparents (Mehmet Yümlü, Ayşe Yümlü, Şerife Sevim and İsmail Sevim) for their love, encouragement, patience and support which made this study possible.

ABSTRACT

BAYESIAN CHANGEPOINT AND TIME-VARYING PARAMETER LEARNING IN REGIME SWITCHING VOLATILITY MODELS

This dissertation proposes a combined state and piecewise time-varying parameter learning technique in regime switching volatility models using multiple changepoint detection. This approach is a Sequential Monte Carlo method for estimating GARCH & EGARCH based volatility models with an unknown number of changepoints. Modern auxiliary particle filtering techniques are used to calculate the posterior densities and online forecasts. This approach also automatically deals with the common ancestral path dependence problem faced in these type volatility models. The model is tested on Borsa Istanbul (BIST) formerly known as Istanbul Stock Exchange (ISE) market data using daily log returns. A full structural changepoint specification is defined in which all parameters of the conditional variance of the volatility models are dynamic. Finally, it is shown with simulation experiments that the proposed approach partitions the series into several regimes and learns the parameters of each regime's volatility model in parallel with the multiple changepoint detection process and shows better forecasting power compared to previous techniques.

ÖZET

REJİM DEĞİŞTİREN VOLATİLİTE MODELLERİNDE BAYES BAZLI DEĞİŞİM NOKTASI VE ZAMANA BAĞLI PARAMETRE KESTİRİMİ

Bu tez, rejim değiştiren volatilite modellerinde değişim noktası tespit yöntemini kullanarak bütünleşik durumların tespiti ve parça parça zamana bağlı değişim gösteren parametreler için bir kestirim tekniği sunmaktadır. Bu yaklaşım değişim nokta sayısı bilinmeyen GARCH ve EGARCH bazlı volatilite modellerinin tahmini için bir Sıralı Monte Carlo yöntemidir. Arka yoğunlukları ve çevrimiçi tahminleri hesaplamak için modern harici parçacık filtreleme teknikleri kullanılmıştır. Bu yaklaşım, aynı zamanda bu tip volatilite modellerinde karşılaşılan ortak atalara yol bağımlılığı sorununa da otomatik olarak çözüm sağlamaktadır. Bu model daha önce İstanbul Menkul Kıymetler Borsası (İMKB) olarak bilinen Borsa İstanbul günlük getiri verisi üzerinde test edilmiştir. Volatilite modelleri için koşullu varyansın tüm parametrelerinin dinamik olarak değişim gösterdiği bir tam yapısal değişim noktası spesifikasyonu tanımlanmıştır. Sonuç olarak, önerilen yaklaşım ve modelin zaman serisini farklı rejimlere bölerek her rejime ait volatilite modeli parametrelerini çoklu değişim noktası tespit süreci ile birlikte öğrendiği ve geçmiş yöntemlere göre daha iyi bir tahmin gücü ortaya koyduğu deneylerle gösterilmiştir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv
ABSTRACT v
ÖZET vi
LIST OF FIGURES ix
LIST OF SYMBOLS xii
LIST OF ACRONYMS/ABBREVIATIONS xiv
1. INTRODUCTION 1
1.1. Contributions 4
1.2. Outline of the thesis 5
2. FINANCIAL TIME SERIES & PREDICTION 7
2.1. Financial Time Series 8
2.2. Financial Time Series Prediction 13
2.2.1. Why Stock Market Prediction? 14
2.2.2. Is the Market Predictable? 14
2.3. Prediction Methods 15
2.3.1. Time Series Prediction using Machine Learning Algorithms . . . 18
2.3.2. Artificial Neural Networks 19
2.3.2.1. Feed-forward Neural Networks 21
2.3.2.2. Recurrent Neural Networks 23
2.3.3. Support Vector Machines (SVM) 24
2.3.3.1. Support Vector Regression (SVR) 27
2.3.4. Mixture of Experts 28
3. STOCHASTIC VOLATILITY MODELS 31
3.1. Financial Time Series & Volatility Models 31
3.2. Volatility & Financial Time Series Characteristics 32
3.2.1. Financial Time Series Characteristics 33
3.2.2. Volatility in Turkish Stock Market (BIST) 34
3.2.3. Risk Management 34
3.3. Traditional Financial Time Series Prediction 36

3.3.1. Autoregressive Conditional Heteroskedasticity (ARCH) 36
3.3.2. Generalized Autoregressive Conditional Heteroskedasticity (GARCH) Models 38
3.3.3. Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) Models 42
4. STATE SPACE REPRESENTATION FOR MULTIPLE CHANGEPOINT MODELS 45
4.1. Inference in Probabilistic Time Series Models 45
4.1.1. Linear Dynamical System (LDS) - Kalman Filters 48
4.1.2. Importance Sampling 51
4.2. State Space Representation for Multiple Changepoint Models 53
4.3. Multiple Changepoints for Volatility Models 57
5. BAYESIAN CHANGEPOINT AND TIME-VARYING PARAMETER LEARNING IN REGIME SWITCHING VOLATILITY MODELS 62
5.1. Sequential Monte Carlo (SMC) methods 62
5.1.1. Sequential Monte Carlo (SMC) methods and Bayesian Filtering 64
5.1.2. Sequential Importance Sampling (SIS) & Resampling (SIR) . . 66
5.2. Auxiliary Particle Filtering (APF) 67
5.3. Mixture Kernel Smoothing (MKS) Based Parameter Filtering 68
5.4. Auxiliary Particle Filtering based Multiple Changepoint Detection in GARCH/EGARCH Models 70
5.4.1. Auxiliary Particle Filtering based Multiple Changepoint Detection Algorithm 71
6. EXPERIMENTS & RESULTS 73
6.1. Experiments on Synthetic Data Series 73
6.1.1. Experiments on GARCH Synthetic Data Series 73
6.1.2. Experiments on EGARCH Synthetic Data Series 77
6.2. Experiments on Real Financial Time Series (BIST 100) 83
7. CONCLUSION & DISCUSSIONS 90
REFERENCES 92

LIST OF FIGURES

Figure 2.1. BIST 100 Index USD based Close prices. 11

Figure 2.2. BIST 100 Index Log Returns. 11

Figure 2.3. Skewness. 12

Figure 2.4. The structure of a basic multilayer artificial neural network [1]. . . 20

Figure 2.5. Elman Network. 24

Figure 2.6. Maximum margin hyperplane. 27

Figure 2.7. A system of expert and gating networks. 29

Figure 3.1. Volatility Realization by GARCH Parameters (changepoints, returns, volatility respectively). 41

Figure 3.2. Volatility Realization by EGARCH Parameters (changepoints, returns, volatility respectively). 43

Figure 4.1. State space model graphical representation. 46

Figure 4.2. Hidden Markov Model. 47

Figure 4.3. Schematic presentation of the expectation. 52

Figure 4.4. Importance Sampling [2]. 54

Figure 4.5. A 4-state left-right HMM model. 55

Figure 4.6. Graphical model for changepoints. 56

Figure 4.7. Run length indicator r_k [3]. 56

Figure 4.8. Graphical representation for the multiple changepoint model. . . . 57

Figure 5.1. Steps for Sequential Monte Carlo (SMC) methods. 63

Figure 5.2. Illustration of Particle Filtering operations [4]. 65

Figure 5.3. Filtering and Prediction Steps in Bayesian Estimation. 66

Figure 6.1. States & Filtered Changepoints, Log Returns and Volatility respectively. 75

Figure 6.2. GARCH: Top Left: Log Returns, Top Right: Changepoint Estimates, Center Left: True Volatility, Center Right: Volatility Estimates, Bottom Left: Effective Sample Size(ESS). 76

Figure 6.3. Filtered parameters in each regime $(\omega_k, \alpha_k, \beta_k, \gamma_k)$ and their %2.5 and %97.5 quantiles. 77

Figure 6.4. States & Filtered Changepoints, Log Returns and Volatility respectively. 79

Figure 6.5. EGARCH: Top Left: Log Returns, Top Right: Changepoint Estimates, Center Left: True Volatility, Center Right: Volatility Estimates, Bottom Left: Effective Sample Size(ESS). 81

Figure 6.6. Filtered parameters in each regime $(\omega_k, \alpha_k, \beta_k, \gamma_k)$ and their %2.5 and %97.5 quantiles. 82

Figure 6.7. Comparison of Volatility Estimates in APF vs MH Models using EGARCH & GARCH simulations respectively. 83

Figure 6.8. Comparison of model parameters in different regimes (EGARCH simulations). 84

Figure 6.9. Comparison of model parameters in different regimes (EGARCH simulations). 85

Figure 6.10. Comparison of model parameters in different regimes (GARCH simulations). 86

Figure 6.11. Comparison of model parameters in different regimes (GARCH simulations). 87

Figure 6.12. BIST100 Changepoint Estimates. 88

Figure 6.13. BIST100 Filtered Parameters. 89

LIST OF SYMBOLS

B_k	The value of divisor at time k (Adjuested base market value)
e	Error function
E	Error vector
E_k	The value of index at time k
$E[.\|.]$	Conditional Expectation operator
$f(x)$	Approximation Function
f_x	State Transition Function
f_y	Observation Function
F_{ik}	The value of i^{th} stock at time k
h_k	Volatility at time k
H_{ik}	The flotation weight (publicly held portion) of the stock i at time k
k	Time index or day of trading
ln	Natural logarithm
n	The number of components used in the index calculation
N_{ik}	The total number of shares outstanding of stock i at time k
P	Transition Probabilities Matrix
$p(x_{1:k})$	Joint distribution of Observations
p_{high}	The highest traded price during the day
p_k	Closing price at the end of the day
p_{low}	The lowest traded price during the day
r_k	Continuously compounded return at k
r_s	i.i.d. variable
s_k	Latent state variable at time k, Regime Identifier
u_k	Process noise
v_k	Observation noise
(V_k)	Volatility series
Vol_k	The total number of traded stocks during the day
$w^{(m)}$	Importance weights

$\bar{X}$	Average of time series
x	Latent state variables
X_i	Time series variable at i
$x_{1:K}$	A collection of time series variables indexed 1 to K
y	Observations
y_k	Observation at time k
z_k	Residuals at time k
α	Volatility parameter for residuals
β	Volatility parameter for historical volatility
γ	Asymmetry or Leverage parameter
γ_1	Skewness
ϵ	Residuals, Innovation Process
Θ	Set of parameters for all regimes
Θ_k	Set of parameters at time k
Θ_m	Set of parameters for the regime m
μ_3	Third central moment
ν	Degree of freedom
σ	Standard Deviation
σ_0^2	Initial Volatility
σ_k	Volatility at time k
ϕ_{k-1}	The information set available at time $k-1$
ω	Volatility long term mean parameter

LIST OF ACRONYMS/ABBREVIATIONS

ANN	Artificial Neural Network
APF	Auxiliary Particle Filtering
AR	Auto-Regressive
$ARCH$	Autoregressive Conditional Heteroskedasticity
$BIST$	Borsa Istanbul Stock Exchange
$BOCPD$	Bayesian Online Changepoint Detection
$CGARCH$	Compound GARCH
CPI	Consumer Price Index
DT	Decision Tree
$EGARCH$	Exponential GARCH
EM	Expectation Maximization
EMH	Efficient Market Hypothesis
EKF	Extended Kalman Filter
FA	Fundamental Analysis
$FTSP$	Financial Time Series Prediction
$GARCH$	Generalized Autoregressive Conditional Heteroskedasticity
GM	Grey Model
GNP	Gross National Product
HME	Hierarchical Mixture of Experts
HMM	Hidden Markov Model
$i.i.d.$	Independent and identically distributed
ISE	Istanbul Stock Exchange
IV	Implied Volatility
$J-B$	Jarque-Bera statistic for normality test
KF	Kalman Filter
MCD	Multiple Changepoint Detection
$MCMC$	Markov Chain Monte Carlo
MKS	Mixture Kernel Smoothing

MLP	Multilayer Perceptron
MoE	Mixture of Experts
MSE	Mean square error
$PARCH$	Partial Autoregressive Conditional Heteroskedasticity
PDF	Probability Density Function
PF	Particle Filtering or Particle filters
PL	Particle Learning
QP	Quadratic Programming
RBF	Radial Basis Function
$RMSE$	Root mean square error
RNN	Recurrent Neural Networks
$RTRL$	Real-time Recurrent Learning
SMC	Sequential Monte Carlo
SOM	Self Organizing Maps
$SPKF$	Sigma Point Kalman Filters
SVM	Support Vector Machines
SVR	Support Vector Regression
TA	Technical Analysis
$TDNN$	Time Delay Neural Networks
UPM	Underlying Predictive Model
USD	United States Dollar
VaR	Value-at-Risk

1. INTRODUCTION

The financial market is a complex, evolving, and nonlinear dynamical system. The financial time series are inherently noisy, non-stationary, and chaotic [5]. This brings the altering structure of the distribution of financial time series over time. The mean and variance of the time series is non-stationary in itself. Even the relationship between different time series such as currency rates and stock exchanges may change over time. Financial Time Series Prediction (FTSP) is first investigating the underlying structure of the time series, fitting a model based on the data changing in time and using this model for prediction of the following time steps.

Modeling such dynamical and non-stationary time series is expected to be a challenging task. FTSP is also a difficult signal processing problem that has hidden variables and lacks observable data for determining the underlying structure of the series, if one exists. We used conditional variance (volatility) in this dissertation which is the time-dependent heteroskedastic variance. Briefly, volatility is the measure of time domain changeability of asset returns [6]. Stock prices fluctuate due to many different riskiness in the market causing heteroskedasticity in the observations. As a result, accurate prediction can only be achieved with a forecast of the time dependent variance of the stock. Volatilities of asset returns are used to analyze market risks, portfolio selection and market timing [7].

Recent financial time series prediction studies were based on applications of Artificial Neural Networks, Mixture of Experts [7], and Support Vector Regression (SVR) [8]. In contrast, our focus is on Sequential Monte Carlo (SMC) methods to perform online prediction based on joint probability distribution in Hidden Markov Model (HMM) for non-linearity and non-Gaussian scenarios [9]. SMC methods, a general class of Monte Carlo methods, are mostly used for sampling from sequences of distributions. Simple examples of these algorithms are extensively used in the tracking and signal processing literature. Recent developments indicate that these techniques have much more general applicability, and can be applied to statistical inference problems in signal processing.

Financial time series includes several switching regimes which are not possible to be modeled by a single model. For this reason, we aim to address challenges in estimating of volatility models such as Generalized Autoregressive Conditional Heteroskedasticity (GARCH) & Exponential GARCH (EGARCH) models that are subject to structural regime switches by using a Bayesian SMC methods approach.

For most real-world problems, the optimal Bayesian inference using recursions is intractable and approximate solutions must be used. Within the space of approximate solutions, the extended Kalman filter (EKF) has become one of the most widely used algorithms with applications in state, parameter and dual estimation. Van der Merwe extended these algorithms to a family of Sigma Point Kalman Filters (SPKF) in dynamic state-space models [10]. Matsumoto attempt to construct a model from time-series data and make an online prediction when the linear assumption is not valid. The problem is formulated within a Bayesian framework implemented by the SMC method [11].

Financial time series includes several switching regimes, for this reason we have to detect the unknown number of changepoints together with the regime parameters. Previous studies on changepoint detection have the roots of Chib's approach. Chib (1998) proposed a convenient state-space formulation of structural break models in which a discrete latent state variable drives structural breaks and follows a constrained Markov chain. An efficient Markov Chain Monte Carlo (MCMC) algorithm for estimation and computing Bayes factors for the purpose of model selection is provided in his paper [12]. Chopin (2007) has developed a particle filtering algorithm for estimating structural break models in which the fixed model parameters are formulated as part of the latent variables [13]. He and Maheu proposed a SMC method for estimating GARCH models subject to an unknown number of structural breaks. They applied their model to daily NASDAQ returns and followed Chib's formulation of structural breaks [14]. Turner et al. proposed an adaptive sequential approach for Bayesian Change Point Detection using Bayesian Online Changepoint Detection (BOCPD) of Mackay by introducing an Underlying Predictive Model (UPM) and a hazard function [15].

Tsay et al. reviewed SMC methods (particle filters (PF)) with special emphasis on its potential applications in financial time. They analyzed Liu-West filter (2001), Storvik filter (2002) and particle learning (PL) of Carvalho, Johannes, Lopes and Polson (2010) and the Auxiliary Particle Filter (APF) of Pitt and Shephard (1999) [16]. Eckley and Fearnhead also analyzed different type of change point models in time series problems including single change point models and multiple change point models. They compared Bayesian based and other approaches in change point modeling [17]. Whiteley et. al. provided explanations about particle filtering, sequential importance sampling and auxiliary particle filtering techniques. He also introduced recent developments in auxiliary particle filtering with applications to several problems [18]. In empirical Bayesian change point detection, Paquet built an online algorithm of Adams and Mackay (2006), who cast the product partition into a Bayesian graphical model. Adams and Mackay computed the probability distribution of the length of the current "run" or time since the last change point, using a simple message-passing algorithm [3]. This model enables an infinite number of hidden states and change points in observations and with this property it has an advantage over Chib' s HMM based approach where the number of change points have to be specified previously [19].

A particle filter is a class of SMC methods that approximates posterior distribution of the latent state variables by a set of particles and associated importance weights. In order to understand the underlying switching models we propose a combined multiple changepoint detection and time-varying parameter learning in regime switching state space models. Our approach includes both state estimation and parameter estimation steps. One approach to learning about fixed parameters is the mixture kernel smoothing method of Liu and West (2001). This dissertation addresses the challenges in estimating the unknown number of changepoints in volatility models using Bayesian inference and SMC methods approaches. A particle filtering approach based on APF to sequential estimation is built on top of the changepoint model of Chib (1998). The unknown number of changepoints and the regime parameters are estimated jointly.

The objective of this research is to investigate the use of recent SMC methods approaches for the Multiple Changepoint Detection (MCD) problems in financial time

series prediction problems without knowing the number of changepoints [20–22]. Financial time series, esp. stock market time series show non-linearity and stochasticity. In order to solve this, several volatility models are used including GARCH & EGARCH and its derivatives. The proposed approach is a SMC method that combines Auxiliary Particle Filtering (APF) with Mixture Kernel Smoothing (MKS) for time-varying parameter and changepoint estimation in regime switching GARCH & EGARCH based volatility models. Recent APF techniques are used to calculate the posterior densities and forecasts in real-time. This approach also automatically deals with the common path dependence problem of these type volatility models.

1.1. Contributions

The contributions in this dissertation which shows a different solution to the problem of multiple changepoint detection, time-varying parameter learning and volatility modeling by extending previous approaches [13, 14, 16, 23, 24] are:

- Providing multiple regime switching state space models instead of fitting a global and single GARCH & EGARCH based model to the time series
- Combined multiple changepoint detection and time-varying parameter learning in regime switching state space models
- Learning without knowing the exact number of the changepoint locations
- Online estimation approach by combining APF based SMC methods and Kernel Smoothing based parameter learning to provide combined state and parameter learning
- Applied extensive simulations using GARCH & EGARCH based models, Student-t distribution over synthetic dataset and validated using an emerging market data of Borsa Istanbul (BIST)

With these features, the proposed approach has advantages over traditional and single model approaches [25]. The proposed solution is first tested over synthetic dataset and then validated using an emerging market data of Borsa Istanbul where volatility shows more fluctuations than US markets. Besides Gaussian distributions,

Student-t distribution which is more suitable for financial time series data is considered with both GARCH and EGARCH based volatility models. This dissertation provides a different and multi-regime auxiliary particle filtering based piecewise time varying parameter learning approach in regime switching volatility models. The originality of this dissertation lies on the proposed approach that combines APF based SMC methods and Kernel Smoothing based parameter learning to provide combined state and piecewise time-varying parameter learning in regime switching volatility models using multiple changepoint detection techniques.

1.2. Outline of the thesis

The rest of the dissertation is organized as follows.

Chapter 2: Financial Time Series & Prediction provides financial time series characteristics and financial time series prediction methods.

Chapter 3: Stochastic Volatility Modeling briefly introduces the stochastic volatility and volatility modeling techniques including ARCH, GARCH and EGARCH.

Chapter 4: State space Representation for Multiple Changepoint Models first provides inference in probabilistic time series models. This chapter then introduces state space representation for multiple changepoint models and multiple changepoints for volatility models.

Chapter 5: Bayesian Changepoint & Time-varying Parameter Learning in Regime Switching Volatility Models introduces the proposed approach for bayesian changepoint & time-varying parameter learning in regime switching volatility models.

Chapter 6: Experiments & Results explains experimental results on synthetic and real financial time series data sets and compares the performance of the proposed approach with previous techniques.

Chapter 7: Conclusion & Discussions summarises the theoretical and experimental results. It discusses their relevance and suggests a few directions for further research.

2. FINANCIAL TIME SERIES & PREDICTION

A time series is a sequence of data points, typically consisting of successive measurements made over a time interval. A time series may consist of a sequence of values that a randomly varying attribute accumulates over time. Common time series examples are stock markets, weekly weather reports, yearly sales of consumer goods or measurements of earthquakes. Time series are used in statistics, signal processing, pattern recognition, econometrics, weather forecasting, earthquake prediction and in any domain of applied science and engineering which involves temporal measurements.

Real world time series data tend to be continuous, and are usually a sequence of observations or values separated by equal time intervals. The corresponding probability model of a time series is called a stochastic process that represents a family of random variables with real values. For example, considering a time series sample represented by a collection of 10 random variables $x_1, x_2, ..., x_{10}$; random variable x_1 represents the value of the series at the first time point, x_2 at time $k = 2$ and so on. A collection of random variables $x_{1:k}$ indexed by k is also referred to as a stochastic process. For this reason, observations in a time series are not purely deterministic.

A well known characteristic of stochastic time series is that they are non-linear and nonstationary. Time series are often presumed to consist of components that enable us to predict future patterns. These components are [26]:

- Trend,
- Cycle,
- Seasonality and
- Random/Irregular fluctuations.

A trend is a variation in the movement of the time series that happens over a period of time. This component shows us patterns of long term growth or decrease in a time series. A cycle is a recurrent oscillatory pattern associated with trend levels.

Oscillations can be quantified as per the duration of similar movements that frequently occur over the length of the trend. Seasonal variations are periodic patterns found in a time series and shows seasonality. Coca-Cola sales statistics in Turkey, for instance, shows seasonal variations. Random movements other than components defined above are called random/irregular fluctuations in a time series.

2.1. Financial Time Series

Most financial data is available in time series form and therefore the statistics and modelling of time series data are essential components underpinning mathematical finance. The information about the market comes from the study of relevant data. Financial time series data can be a technical data such as the closing price at the end of the day (p_k), the highest traded price during the day (p_{high}), the lowest traded price during the day (p_{low}), and the total number of traded stocks during the day (Vol_k) where k represents the day of trading. It can also be a fundamental data that is based on a fundamental analysis of an asset or a market. This can be inflation rates, interest rates, trade balance, prices of related commodities such as oil, metal, etc. or it can be the net profit margin of a firm. There are several different influencing factors that may be valuable in a financial time series analysis and prediction problems. Financial time series data can also be a combination or derivation of the predefined data. Returns or volatility are good examples of this derived type of data.

- Returns: Continuously compounded return r_k is defined as the relative increase in price since the previous point in the time series.
- Volatility: Describes the variability of a stock and is used as a way to measure the risk of an investment.

There are several reasons for focusing on returns rather than on prices. First, for the average investor, financial markets may be considered close to perfectly competitive, so that the size of the investment does not affect price changes. The return is scale free. The prices vary greatly and make it difficult to create a valid model for a longer period of time and prices for different stocks may easily differ over several decades and

therefore cannot be used as the same type of input in a model. Statistically, returns brings stationarity into the model.

Consider p_k, the price of an asset without dividends. The return of this asset between times $k-1$ and k, r_k, is defined as

$$r_k = p_k/p_{k-1} - 1 \tag{2.1}$$

The difficulty of manipulating geometric average operations over the whole time period brings another approach, the compounded returns which have also important implications on modeling of asset returns. The *continuously compounded returns* or *log returns* of an asset is defined as

$$r_k = ln(p_k/p_{k-1}) \tag{2.2}$$

Here ln represents the natural logarithm. In this dissertation work, r_k is taken as continuously compounded return of the price. Continuously compounded returns are chosen in order to achieve comparability and stationary series in the same domain.

The financial time series data set used in this dissertation is taken from an emerging market, Borsa Istanbul (BIST) formerly known as Istanbul Stock Exchange (ISE), BIST National 100 index (BIST 100).

In Turkey, the Istanbul Stock Exchange (ISE) was established in early 1986. ISE is the only securities exchange in Turkey established to provide trading in equities, bonds and bills, revenue-sharing certificates, private sector bonds, foreign securities and real estate certificates as well as international securities.

BIST 100 reflects all the characteristics of the market because it includes the most active and volumetric 100 stocks which are selected among the stocks of companies

traded on the national market. The time series data starting from 1988 up to year 2015 is gathered to analyze the behavior of the proposed approach. Data set includes 6674 daily observations starting from 04 January 1988 to 14 November 2014 and their log returns (Figure 2.1 & Figure 2.2).

BIST 100 index is an index that is a weighted average - it takes more than one parameter to calculate the final result and it assigns different weights to each financial argument. The basic formula for calculating BIST 100's float capitalization-weighted indices is as follows:

$$E_k = \frac{\sum_{i=1}^{n} F_{ik} N_{ik} H_{ik}}{B_k} \tag{2.3}$$

where E_k is the value of index at time k, n is the number of components that contribute to the index calculation, F_{ik} is the value of i^{th} stock at time k, N_{ik} is the total number of shares outstanding of stock i at time k, H_{ik} is the flotation weight (publicly held portion) of the stock i at time k and B_k is the value of divisor at time k (Adjusted base market value).

Financial time series shows different characteristics and features when analyzed in statistically. Table 2.1 shows some important statistical measures on BIST 100 index data.

Table 2.1. BIST 100 Log Returns Statistics.

Size	Mean	Var	Kurtosis	Skewness	J-B	Min	Max
6674	0.000252	0.000918	8.835498	-0.308427	9575.395	-0.25376	0.213246

Notable statistics presented above are kurtosis, skewness and Jarque-Bera (J-B) test. Skewness is a measure of asymmetry, visually a distribution with positive skewness would have a so-called tail much longer on its right side than on its left. The reverse statement is true for negative skewness - such distributions would have a longer

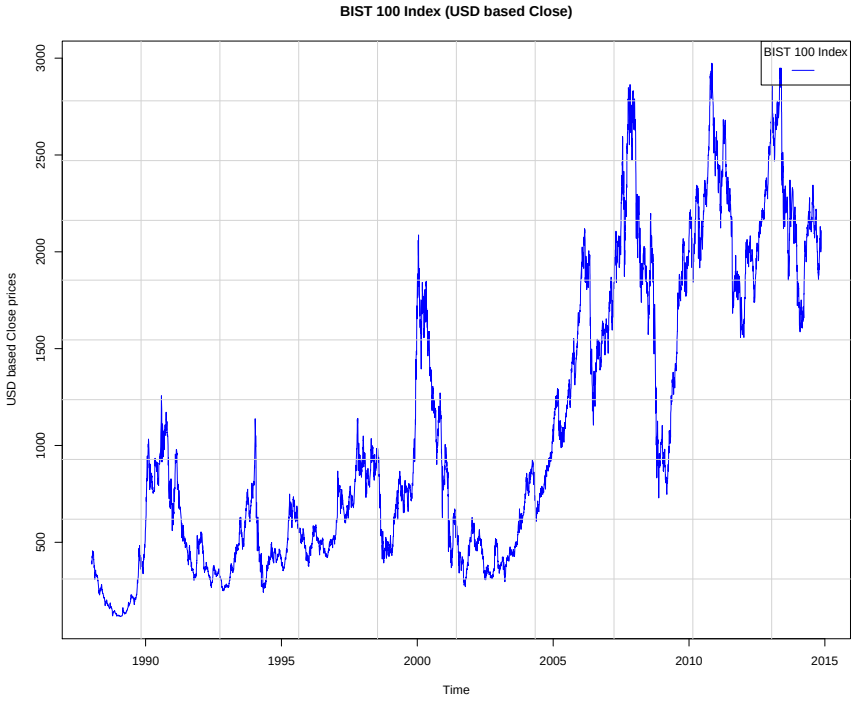


Figure 2.1. BIST 100 Index USD based Close prices.

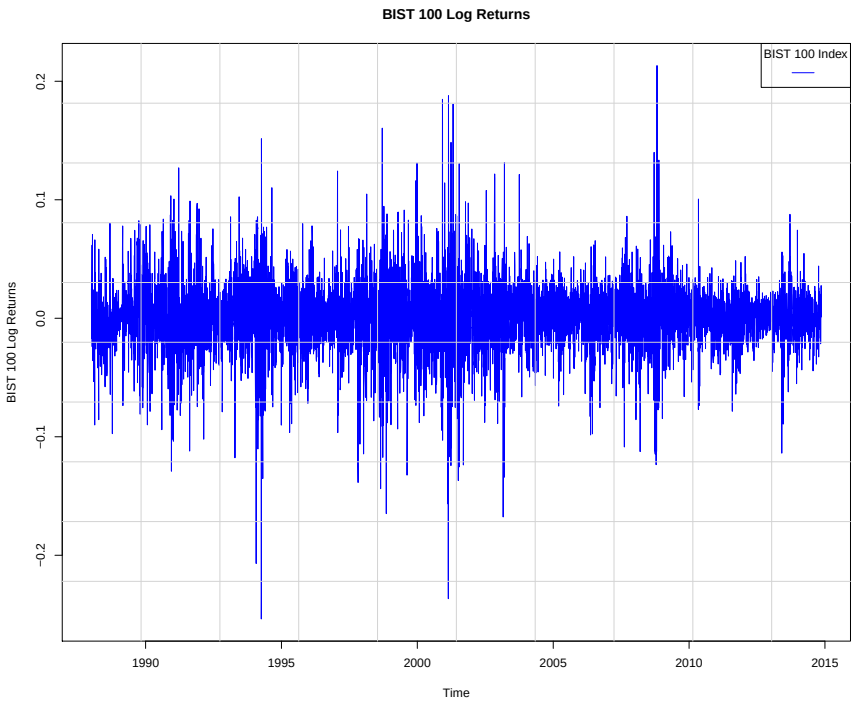


Figure 2.2. BIST 100 Index Log Returns.

tail on the left side of its graphic. Longer tail on one side mean more data would come from that side of the distribution.

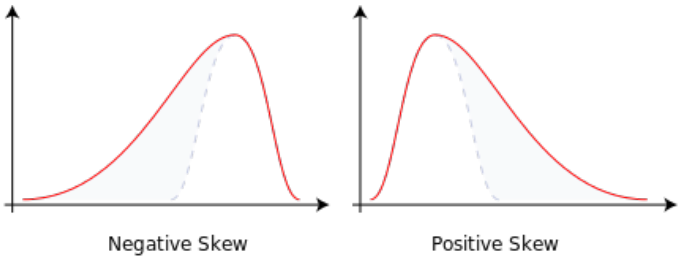


Figure 2.3. Skewness.

Mathematically, skewness is the third standardized moment γ_1 calculated as

$$\gamma_1 = \frac{\mu_3}{\sigma^3} \tag{2.4}$$

where μ_3 is the third central moment, μ is the mean and σ is the well-known standard deviation. Skewness can signal us if one can assume a distribution is Gaussian. Gaussian distribution has a skewness of zero, hence if sample skewness of a distribution is non-zero, we can conclude that this distribution is not a Gaussian (Normal) distribution.

Kurtosis shows the presence of extremely high values in the data. Standart normal distribution has a kurtosis of zero. Higher kurtosis values signal the presence of sharper peak in the distribution where lower kurtosis tend to have flatter peaks. Kurtosis shows whether extremely high values exists in a distribution or not. Kurtosis is calculated as

$$\frac{\sum_{i=1}^{n} T(X_i - \bar{X})^4}{(T-1)\sigma^4} \tag{2.5}$$

Taken together skewness and kurtosis can indicate if a distribution is normal or not. There is also a goodness-of-fit measure called Jarque-Bera test that can indicate

normality of a distribution. This test combines skewness and kurtosis to accept or reject the null hypothesis that the distribution in question is normal.

Looking at skewness an kurtosis for BIST 100, we can see high kurtosis and negative skewness. It is known that in finance literature stock markets are negatively skewed because decreases are much sharper than increases. This is also explained with a typical human behaviour in phychology: At time of crises, people behave in panic and in groups to sell their stocks at the same time, hence causing sharper decreases than increases in the stock market data.

2.2. Financial Time Series Prediction

Time series analysis comprises methods for analyzing time series data in order to extract meaningful statistics and other characteristics of the data. Time series forecasting is the use of a model to predict future values based on previously observed values. Financial Time Series Prediction (FTSP) aims to find underlying patterns, trends, cycles and aims to forecast future using historical and currently observable data [27].

In order to define the problem, first we start with a precise definition of the prediction before getting deeper into details:

Given a sample of N examples, $\{(x_k, y_k), k = 1, ..., N\}$, where $f(x_k) = y_k$ and for all k we try to return a function g that approximates f in the sense that error vector $E = (e_1, e_2, ..., e_N)$ is minimized. Each e_k is defined as $e_k = e(g(x_k), y_k)$ where e is the defined error function [28].

In other words, the definition above indicates that in order to predict the market one should search historic data and find relationships between data and the value of the market. Next step is to exploit these relationships found on future situations. Above definition is based on the assumption that such relationships do exist. But do they? Or do the markets fluctuate in a totally random way leaving us no space for prediction?

2.2.1. Why Stock Market Prediction?

The wish to find methods to predict asset returns has occupied the minds of investors and also academia since the foundation of financial markets. In general, financial assets are influenced by the real economy. The liberalization and globalization of world asset markets have caused interest rates, exchange rates, and also other asset markets to be intimately linked, and the need for tools to monitor as well as control risk levels has become obvious both for industrial companies and financial institutions. The question of predictability in the stock markets is therefore important even outside the trading rooms.

2.2.2. Is the Market Predictable?

The predictability of the market has been and is being predominantly studied by researchers and academia. As a result, a hypothesis formulated as the Efficient Market Hypothesis (EMH) implies that there is no way to make profit by predicting the market. The EMH states that the current market price reflects assimilation of all the information available. This means that given the information, no prediction of future changes in the price can be made. As new information enters the system, the unbalanced state is immediately discovered and it is quickly eliminated by a correct change in market price.

More specifically the EMH has three forms:

- The weak form: Only past price data is considered. This kind of EMH rules out any form of predictions based on the price data only.
- The semi-strong form: This EMH type states that you cannot even utilize published information to predict future prices.
- The strong form: This claims that you cannot predict the market no matter what information you have available.

According to Weak Form Efficiency Hypothesis, stock prices follow a ‘Random

Walk' model. Which more formally stated is

$$y_k = y_{k-1} + r_s \tag{2.6}$$

where y_k is the value of the market on time k and r_s is an independent and identically distributed (*i.i.d.*) variable. If this model is valid, the best prediction about tomorrow's value is today's value.

A natural benchmark for any prediction algorithm should be it's relative success aganist a random predictor. This idea was inspired from Fama [29] where he states "if [an] analyst can make meaningful judgements concerning the purchase and sale of individual securities, his choice should consistently outperform randomly selected securities of the same general riskiness".

2.3. Prediction Methods

The prediction of the market is without doubt an interesting task. In the literature, there are a number of methods applied to accomplish this task [30–33]. Various approaches have been used ranging from highly informal ways (e.g. the study of a chart with the fluctuation of the market) to more formal ways (e.g. linear or non-linear regressions).

These techniques are as follows:

- Technical Analysis Methods [31],
- Fundamental Analysis Methods [32],
- Traditional Time Series Prediction Methods [30],
- Machine Learning Methods [33]

The criterion to this categorization is the type of tools and the type of data that each method is using in order to predict the market. What is common to these techniques is that they are used to predict and thus benefit from the market's future

behaviour. None of them has proved to be the consistently correct prediction tool an investor would like to have. Furthermore many researchers question the usefulness of many of these prediction techniques.

There are both traditional and machine learning methods and studies published that can be related to our dissertation in the literature. Murphy used technical analysis for stock market prediction previously [31]. Brooks used value at risk metrics for forecasting of exchange rates [34–36]. Nelson proposed a model to capture asymmetric effects in time series prediction that is called Exponential GARCH [37]. Mazıbaş [38] evaluated the out-of-sample forecasting accuracy of fifteen symmetrical and asymmetrical GARCH models for daily, weekly and monthly volatility in composite, financial, services and industry indices of Istanbul Stock Exchange (ISE) including GARCH, EGARCH, PARCH and CGARCH. Some properties of financial data namely volatility clustering, asymmetrical price movements, leverage effects and fat-tail, has been investigated in stock market data of 1997-2004.

Technical analysis (TA) is defined as the study of market (price) actions for the purpose of forecasting future price trends [31]. Using technical data such as price, volume, and highest and lowest prices per trading period the technical analyst uses charts to predict future stock movements. One of the reasons for TA's popularity is that it forces a discipline and control on trading by providing traders with price and profit/loss objectives before trades are made. It is also a very useful tool for short-term as well as long-term trading strategies as it does not rely on any information other than market data. Another reason for its popularity is that, while its basic ideas are easy to understand, a wide variety of trading strategies can be developed from these ideas.

The second common technique, Fundamental analysis (FA), studies the effect of supply and demand on price. All relevant factors that affect the price of a security are analyzed to determine the intrinsic value of the security. If the market price is below its intrinsic value then the market is viewed as undervalued and the security should be bought. If the market price is above its intrinsic value, then it should be sold. Examples of relevant factors that are analyzed are financial ratios; e.g. Price to Earnings, Debt to

Equity, Industrial Production Indices, Gross National Product (GNP) and Consumer Price Index (CPI). Fundamental analysis studies the causes of market movements, in contrast to technical analysis, which studies the effect of market movements. Interest Rate Parity Theory and Purchasing Power Parity Theory are examples of the theories used in forecasting price movements using fundamental analysis.

The Traditional Time Series Prediction methods analyze historic data and attempts to approximate future values of a time series as a linear combination of these historic data. Basically, these methods attempt to model a nonlinear function by a recurrence relation derived from past values. The recurrence relation can then be used to predict new values in the time series, which hopefully will be good approximations of the actual values. In econometrics there are two basic types of time series forecasting: Univariate (simple regression) and multivariate (multivariate regression). Univariate models, like Box-Jenkins [30], contain only one variable in the recurrence equation. Box-Jenkins is a complicated process of fitting data to appropriate model parameters. The equations used in the model contain past values of moving averages and prices. Box-Jenkins is good for short-term forecasting but requires a lot of data, and it is a complicated process to determine the appropriate model equations and parameters. Multivariate models contain more than one variable in their equations. Traditional Time Series Prediction methods including Autoregressive Conditional Heteroskedasticity (ARCH), Generalized Autoregressive Conditional Heteroskedasticity (GARCH) and Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) are introduced in Chapter 3 in details.

Several methods for inductive learning also have been developed under the common label **"Machine Learning"**. All these methods use a set of samples to generate an approximation of the underlying function that generated the data. The aim is to draw conclusions from these samples in such a way that when unseen data are presented to a model it is possible to infer the explained variable from these data. These methods will be investigated in the next section.

2.3.1. Time Series Prediction using Machine Learning Algorithms

When we say prediction, we basically aim at understanding the dependency between the past and the future, the dependency between the input and the output in a financial time series prediction problem using machine learning techniques.

In this section, some of the popular and well known previous machine learning based time series prediction techniques are introduced. In fact, these techniques provides the basis for the machine learning approaches that are previously used for the problem of time varying parameter learning which is one of the main topics of this dissertation [39]. All of the techniques mentioned surrounded by Artificial Neural Networks (ANN) and Support Vector Machines (SVM) were trying to model the financial time series with a single model.

Prediction is finding an approximate mapping function between the input and the output data space. Financial time series prediction problem aims to find underlying patterns, trends, cycles and forecast future using historical and currently observable data. Over the past few years, neural networks have been successfully used for modeling financial time series ranging from options price [40], stock index volatility prediction [7, 39] to currency exchange [41]. Neural networks are universal function approximates that can map any nonlinear function without a priori assumption about the data. Unlike traditional statistical models, neural networks are data-driven; they use data as the primary factor for the model to be extracted [4, 42]. So neural networks are less susceptible to the model misspecification problem than most of the parametric models, and they are more powerful in describing the dynamics of financial time series than traditional statistical models [27, 43]. Tsai and Wand created a stock price forecasting model using artificial neural networks (ANNs) and decision trees and showed that ANN + DT models have significant predictive power over single ANN and DT models [44].

Yümlü, et al. has investigated the use of global, recurrent and smoothed neural network models in financial time series prediction [7, 27, 45, 46]. Girosi and Poggio have shown that radial basis functions possess the property of best approximation. An

approximation scheme has this property if, in the set approximating functions there is one function which has minimum approximating error for any given function to be approximated. They have also shown that this property was not shared by multilayer perceptrons [47]. Weigend used a single non-linear gate and non-linear experts, and called the model non-linear gated experts [48].

The following two sections describes two main machine learning techniques of Artificial Neural Networks (ANN) and Support Vector Machines (SVM). By the end of this chapter Mixture of Experts (MoE) technique is introduced which Yümlü has also previously used for financial time series prediction [7] to explain why a single model is not enough to model financial time series, esp. stock market time series and how MoE is used to model the multiple regimes and regime switching structure for financial time series. By considering all these techniques and literature review we have chosen to provide a Bayesian multiple changepoint detection and time-varying parameter learning solution for regime switching volatility models.

2.3.2. Artificial Neural Networks

Artificial Neural Networks (ANN) are one of the most well known and accepted techniques for prediction problems for the last two decades. They are relatively crude electronic models based on the neural structure of the brain. The brain basically learns from experience. Cognitive and perceptual powers of humans are much more powerful than today's computers. Humans can effortlessly recognize images in many inappropriate conditions, whereas machines fail to compete with human in these areas. But computers have trouble recognizing even simple patterns much less generalizing those patterns of the past into actions of the future. Artificial neural networks do not utilize traditional programming but involves the creation of massively parallel networks and the training of those networks to solve specific problems.

The fundamental processing element of a neural network is a neuron. A neuron is a processing unit that takes a number of inputs and gives a distinct output. This building block of human awareness encompasses a few general capabilities. Basically,

a biological neuron receives inputs from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result. Each of the inputs is multiplied by a connection weight which is the link between the neurons. The products are summed and passed through a transfer function. This function then turns this number into a real output via some algorithm. The transfer functions those are commonly supported are sigmoid, sine, hyperbolic tangent, etc.

Basic neural networks are single layer networks that do not have a hidden layer in their structure. The limitations with single layer networks are to some extent overcome by the introduction of multi-layer networks. The structure of a basic multilayer artificial neural network is shown in Figure 2.4. The output layer often has linear activation functions if the network is used for function approximation, and sigmoid if the network is used for classification purposes.

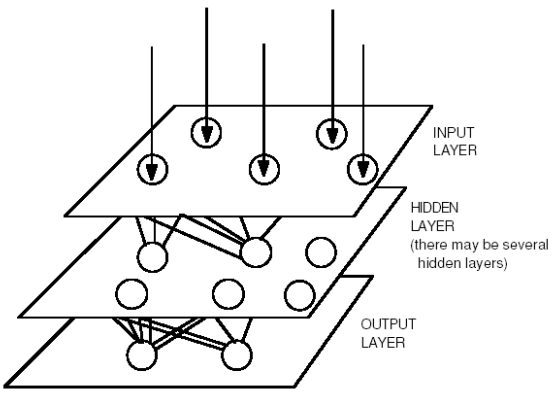


Figure 2.4. The structure of a basic multilayer artificial neural network [1].

ANNs can be used to perform classification and regression tasks. More specifically it has been proved by Cybenko that any function can be approximated to an arbitrary accuracy by a neural network [33].

The basic approach in learning is to start with an untrained network, present a training pattern to the input layer, pass the signals through the net and determine the output at the output layer. These outputs are compared to the target output values. Any difference in this situation corresponds to an error. This error function is some

scalar function of the weights and is minimized when the network outputs match the desired outputs. The weights are adjusted to reduce this measure of error. The back-propagation learning, which is based on gradient descent, is used to minimize the total error. It corresponds to a propagation of errors backwards through the network. The technique of back-propagation was popularized in a paper by Rumelhart, Hinton and Williams [49].

In neural network processing, there are several affecting factors to be considered to have a greater performance and a good generalization over the time series data. If these factors are not considered during the learning process, it is not possible to expect good generalization [4, 42]. These factors are

- Input Data Selection
- Preprocessing
- Cross-Validation
- Number of hidden units
- Initializing weights
- Learning and Momentum Rate
- Weight Decay
- Early Stopping

Besides the affecting factors, the structure of the neural network is also important. Both feed-forward neural networks like Multi-layer Perceptron (MLP) and Radial Basis Functions (RBF) and recurrent neural networks like Elman and Jordan type of neural networks have been used previously successfully for time series prediction problems [7]. These models will be briefly introduced in the following subsection.

2.3.2.1. Feed-forward Neural Networks. Multi-layer Perceptron (MLP) is a useful network that is able learn the non-linearity in data and is very useful in both regression and classification problems. As a subset of regression problem, function approximation in time series has been studied before and research at this area has not been come

across with results that satisfy any participant in the field. The Universal Approximation Property mentions that an MLP can approximate any nonlinear function to an arbitrary degree of accuracy with a suitable number of hidden layers [50, 51]. Most of the studies are Auto-Regressive (AR) model's simulations having assumed that the data depends on p previous elements in the time series. For this purpose, a window is defined that is including the previous p samples in the data set and having their weighted sums. This technique is known as the sliding windows technique.

The class of Radial Basis Functions (RBF) was first used to solve interpolation problems –fitting a curve exactly through a set of points. RBF methods have their origins in techniques for performing exact interpolation of a set of data points in a multi-dimensional space [52]. Consider a mapping from a $d-$dimensional input space x to a one-dimensional target space t. The data set consists of N input vectors x^n, together with corresponding targets t^n. The goal is to find a function $h(x)$ such that

$$h(x) = t^n, n = 1, ..., N \tag{2.7}$$

The RBF approach introduces a set of N basis functions, one for each data point, which take the form $\phi(||x - x^n||)$ where $\phi(.)$ is some non-linear function. The n^{th} function depends on the distance $||x - x^n||$, usually taken to be Euclidean, between x and x^n. The output of the mapping is then taken to be a linear combination of the basis functions.

$$h(x) = \Sigma_n w_n \phi(||x - x^n||) \tag{2.8}$$

For $\phi(.)$, several forms of basis functions have been considered, the most common being the Gaussian is

$$\phi(x) = exp(-\frac{x^2}{2\sigma^2}) \tag{2.9}$$

where σ is a parameter whose value controls the smoothness properties of the interpolating function.

Girosi and Poggio have shown that radial basis functions possess the property of best approximation. An approximation scheme has this property if, in the set approximating functions there is one function which has minimum approximating error for any given function to be approximated. They have also shown that this property was not shared by multilayer perceptrons [47].

2.3.2.2. Recurrent Neural Networks. Time can be introduced into ANN models in different ways. First possibility is to leave the time outside the ANN model and preprocess time to use it in standard ANN models. This method is used in Time Delay Neural Networks (TDNN) or in ANNs with a sliding window as it was applied in MLP [53]. Here the last n values of a time-dependent variable are given as an input to the network. Another possibility is to encode the temporal information into numerical values first and use these values in the ANN. Introduction of time into the model as an index of the state of the network is another way of handling time in ANNs. In this type of ANN, states of neurons are kept to be used in the following time steps. Order is the most important property for time in these models. Therefore these models are very suitable for data where order relation between observations is important. Stock market time series data is such a data type. These models are known as Recurrent Neural Networks (RNN).

Elman and Jordan networks raised importance in the literature [54, 55]. Jordan' recurrent neural network model uses connections from the output units back to the input of the network in order to achieve the learning of sequential tasks in language processing [55]. Instead of using recurrent connections from output units to the network input, Elman's recurrent neural network stores the activations of hidden units as representation of the network's internal state and feeds them back as part of the hidden layer's input at the next time step [54]. The activation of hidden units at time step k are directly copied into the so-called context units and kept there as input to

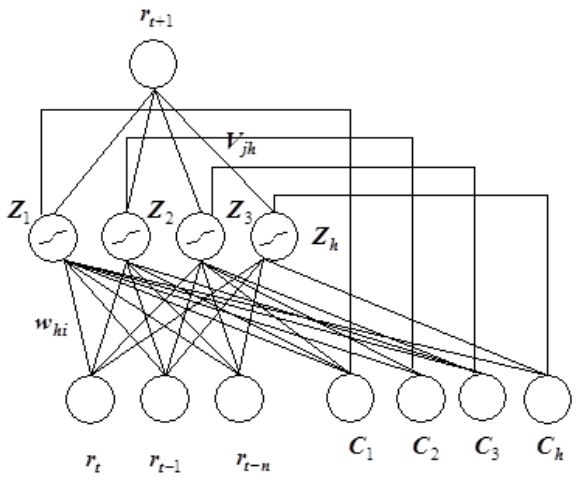


Figure 2.5. Elman Network.

the hidden units at the next time step $k + 1$ (Figure 2.5). There is a connection with a weight value of one and a time delay of one step. Real-time Recurrent Learning (RTRL) algorithm is used to learn the temporal effects of the time series for recurrent neural networks.

Also capacitive RNNs were considered, which has a capacitive layer for the context layer to store the previous state of the context state in addition to the hidden layer's state storage [39].

2.3.3. Support Vector Machines (SVM)

Support Vector Machine (SVM), based on Statistical Learning Theory, was first developed by Vapnik [56, 57]. It has become a hot topic of intensive study due to its successful application in classification tasks and regression tasks, especially on time series prediction and finance related applications.

Support vector machine is a very specific type of learning algorithms characterized by the capacity control of the decision function, the use of the kernel functions and the sparsity of the solution. Established on the unique theory of the structural

risk minimization principle to estimate a function by minimizing an upper bound of the generalization error, SVM is shown to be very resistant to the over-fitting problem, eventually achieving a high generalization performance [57–60]. Another key property of SVM is that training SVM is equivalent to solving a linearly constrained quadratic programming problem so that the solution of SVM is always unique and globally optimal, unlike neural networks training which requires nonlinear optimization with the danger of getting stuck at local minima.

Van Gestel applied the Bayesian evidence framework to least squares support vector machine (LS-SVM) regression in order to infer nonlinear models for predicting a time series and the related volatility [61]. Cao proposed a model using the support vector machines (SVMs) experts for time series forecasting on sunspot data and SantaFe data sets in two stages. First Self-organizing feature map (SOM) is used as a clustering algorithm to partition the whole input space into several disjointed regions. A tree-structured architecture is adopted in the partition to avoid the problem of predetermining the number of partitioned regions. In the second part, multiple SVMs that best fit partitioned regions are constructed by finding the most appropriate kernel function and the optimal free parameters of SVMs [62]. Cao and Tay had dealt with the application of SVM in financial time series forecasting. They investigate the variability in performance of SVM using adaptive parameters on Chicago Mercantile Market future contracts. According to Cao and Tay SVM outperforms back-propagation neural networks in financial forecasting [63]. Tay and Cao also proposed a modified version of SVMs, called C-ascending SVM, to model non-stationary financial time series. They modified the regularized risk function in SVM where E-insensitive errors are penalized heavily than distant E-insensitive error on Chicago Mercantile Market data [64].

Kim applied SVM to stock price index prediction and compared the results with back-propagation neural networks and case-based reasoning and showed that SVMs plays a significant role of an alternative to neural networks [65]. Smola and Schölkopf worked on Support Vector Regression and their tutorial is a valuable introduction to the topic of Support Vector Regression (SVR) [66]. Yang et al. applied SVR to financial prediction by varying the margins of the SVR to reflect the change in volatility

of the financial data in Hang Seng Index data [67]. Yümlü et al. applied SVR to propose a solution for financial time series prediction of Istanbul Stock Exchange using Support Vector Predictors [8,68]. Wang et al. used a modularized connectionist model introducing a time variant HMM model for time series prediction. Their approach achieves significant improvement in the generalization performance over global models [69].

Chen and Shih applied SVM and neural networks on six Asian stock markets rather than US or European markets [70]. Kaynak et al. used a different model of Grey system theory-based models in time series prediction. They investigated the use of grey models using Fourier series on highly noisy data of USD/Euro parity between 2005 and 2007. Their simulations have shown that modified grey model GM (1, 1) using Fourier series in time is the best in model fitting and forecasting [71].

In SVM, for the classification case, the aim is to find an optimal hyperplane that separates two classes. In order to find an optimal hyperplane, we need to minimize the norm of the vector w, which defines the separating hyperplane. This is equivalent to maximizing the margin between two classes. The maximum margin hyperplane is shown in Figure 2.6. In the case of regression, the goal is to construct a hyperplane that lies close to as many of the data points as possible. Therefore, the objective is to choose a hyperplane with small norm while simultaneously minimizing the sum of the distances from the data points to the hyperplane.

Both in classification and regression, we obtain a quadratic programming (QP) problem where the number of variables is equal to the number of observations. Basically, the SVM regressor maps the inputs into a higher dimensional feature space in which a linear regressor is constructed by minimizing an appropriate cost function. The regressor is obtained by solving a finite dimensional QP problem in the dual space avoiding explicit knowledge of the high dimensional mapping and using only the related kernel function.

When using SVM in regression tasks, the Support Vector Regressor must use a

cost function to measure the empirical risk in order to minimize the regression error. There are many choices of the loss functions to calculate the cost, e.g., least modulus loss function, quadratic loss function, the intensive loss function, etc.

2.3.3.1. Support Vector Regression (SVR). Support Vector Regression is the application of SVMs into the regression problems. The goal is to construct a hyperplane that lies close to as many of the data points as possible. The problem is to minimize the sum of distances from the data points to the hyperplane defined. In the following section, we briefly described the mathematical formulation of SVR using $\epsilon - intensive$ loss function.

Let us assume, $x_1, x_2, ..., x_N$ as our data set. Our goal is to find a function $f(x)$ that has a ϵ deviation from the target data points.

$$f(x) = wx + b \tag{2.10}$$

The decision boundary defined by the hyperplane can be found by solving the problem given in 2.11 subject to $y_i(w^T x_i + b) \geq 1, \forall_i$.

$$min \frac{1}{2} \parallel w \parallel^2 \tag{2.11}$$

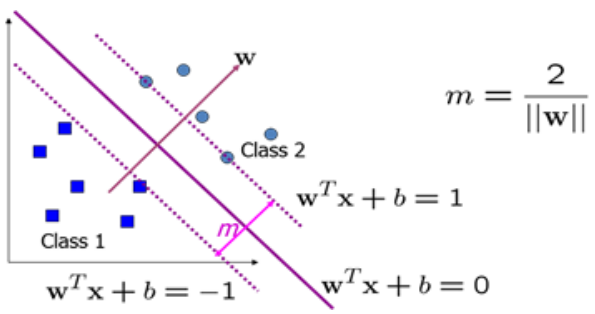


Figure 2.6. Maximum margin hyperplane.

The Lagrangian function used in the dual problem is given in 2.12.

$$L = \frac{1}{2} w^T w + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + b)) \tag{2.12}$$

In the non-linear case, the problem is converted into a minimization problem of defined in 2.13 subject to $y_i(w^T x_i + b) \geq 1 - \xi_i$, $\xi_i \geq 0$.

$$min \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{n} \xi_i \tag{2.13}$$

2.3.4. Mixture of Experts

Until now, we have investigated single model approaches for modeling time series and for financial time series prediction. Yümlü et. al. showed that stock market time series does not have a constant mean and variance and this variance has a time-varying structure which is analyzed in Chapter 3 in details. Stock market time series consists of several changepoint locations where the model behind the variance changes and consists of several regimes which makes the financial time series prediction problem hard to model with single modeling techniques either ANN or SVM. For this purpose mixture models have been introduced to model different parts (regimes) of time series.

The basic idea behind mixture models is to divide the input space into several parts, to build a model for each part, and then to combine the local models to get a better global model. This approach is based on the principle of divide-and-conquer and is widespread in different fields of computer science and applied mathematics. There are two problems in this scenario that is considered. The first one is the splitting of the input space into a set of regions and the other one is to fit each region with a simpler function compared with a global fitting function. Against the problems, the divide-and-conquer approach in mixture of experts applied to forecasting has several advantages. Each local model can specialize on a different part of the input space. The combination

of these specialized models, experts can result in a better model than possible with a single global model. Each expert is a feed-forward network and all experts receive the same input and have the same number of outputs. The gating network is also feed forward, and typically receives the same input as the expert networks.

The Mixture of Experts (MoE) [72] has emerged as a powerful divide and conquer algorithm. It consists of a set of competing experts moderated by a gate. When the experts are predictors and the data set is a time series, the algorithm is capable of segmenting and identifying the time series into stationary regions, in a completely unsupervised fashion, by observing the output of the gate over time. Figure 2.7 gives the structure of the Mixture of Experts architecture.

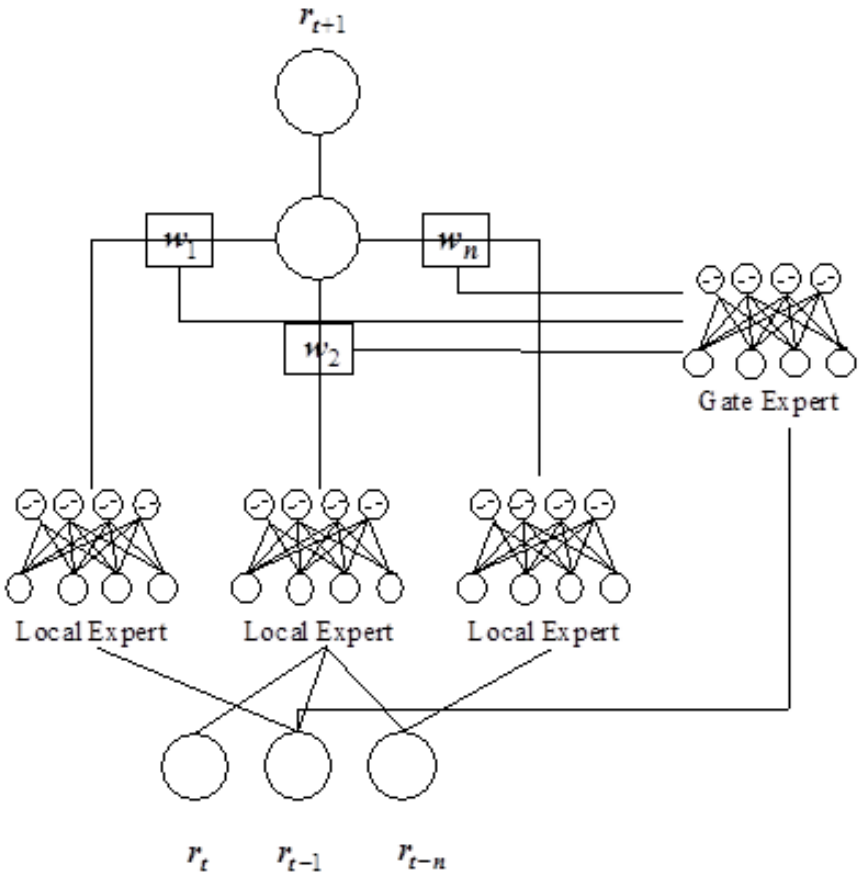


Figure 2.7. A system of expert and gating networks.

Stock market time series has multiple changepoint locations which has to be

identified to understand different regimes. Previously Yümlü et. al. modeled this multiple regime switching structure using Mixture of Experts (MoE) [7] and showed that MoE performs better than single models but has limitations in terms of changepoint location identification, regime identification and time-varying parameter learning. Financial time series, esp. stock market time series show non-linearity and stochasticity. In this dissertation, we have proposed a novel approach, an SMC method that combines Auxiliary Particle Filtering (APF) with Mixture Kernel Smoothing (MKS) for time-varying parameter and multiple changepoint estimation in regime switching GARCH & EGARCH based volatility models. This multiple changepoint model and proposed model is explained in Chapter 4 and Chapter 5.

3. STOCHASTIC VOLATILITY MODELS

Recent research in financial time series analysis has put a lot of emphasis on modeling and forecasting asset return volatilities. This is because of the fact that asset prices vary with the changes in volatilities of the underlying asset. Therefore, we need to forecast the volatility to have accurate predictions of future prices. Another important aspect is the market risk management. In order to calculate the market risk we need a measure known as value-at-risk (VaR), which requires a forecast of volatilities of the risk factors such as market returns, interest rates etc.

Conditional variances are known to be unobservable time dependent features and can be analyzed using Autoregressive Conditional Heteroskedasticity (ARCH) processes based heteroskedastic models proposed by Engle (1982) [73]. There is no doubt that in the literature the most prominent volatility model that estimates conditional variances of asset returns on the basis of historical observations is Bollerslev (1986)'s Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model in which conditional variances are governed by a linear autoregressive process of past squared returns and variances [6]. It is able to capture several important stylized facts of asset returns, namely heteroskedasticity, volatility clustering and excess kurtosis. In the following sections, we will survey volatility, characteristics of financial time series, risk management and some common volatility models also used throughout this dissertation.

3.1. Financial Time Series & Volatility Models

The financial time series data set is used from an emerging market, Borsa Istanbul formerly known as Istanbul Stock Exchange (ISE), BIST National 100 index. BIST 100 reflects all the characteristics of the market because it includes the most active and volumetric 100 stocks which are selected among the stocks of companies traded on the national market. ISE was established in 1988 and all the data starting from 1988 up to year 2015 is gathered to analyze the behavior of the proposed approach. Data set

includes 6674 daily observations starting from 04 January 1988 to 14 November 2014 (Figure 2.1).

Turkish stock market BIST, experiencing powerful fluctuations translated by large price movements, is a light star among volatile emerging markets. Closer look to the BIST reveal high degree of persistence and strong time dependence in conditional variances that makes forecasting volatility an important issue [6].

3.2. Volatility & Financial Time Series Characteristics

Stock prices vary with changes in volatilities of the underlying risk factor and as a consequence, accurate prediction of future stock prices requires a forecast of asset return's volatility. Financial time series exhibit time dependent heteroskedastic variance known as the conditional variance (volatility) that is not a directly observable feature. There are basically two notions of volatility in the literature: historical volatility and implied volatility. Historical measure is the popular approach where exponentially declining weights are given to past volatilities, approximated by squared returns. Historical volatility is an exponentially weighted moving average of squared returns. The returns are calculated from the closing values of the index. Returns are governed from $r_k = ln(p_k/p_{k-1})$ where p_k represents the closing index price and r_k represents the returns. Historical volatility (V_k) is then measured as

$$V_k = (1-\alpha)\sum_{k}^{K} \alpha^{K-k} r_k^2 \tag{3.1}$$

The weighing factor $\alpha \in (0,1)$ determines the impact of past returns on the actual volatility. V_k can also be represented as

$$V_k = \alpha V_{k-1} + (1-\alpha) r_k^2 \tag{3.2}$$

Historical volatility measure is similar to the basic volatility measure applied in RiskMetrics™. The implied volatility measure (IV-measure) is estimated from the extracted options' volatilities implied by the Black-Scholes model [74]. This is closely related to option prices and requires an option-pricing model in order to calculate the market driven volatilities.

Volatility models are used to forecast the risk, volatility of asset returns. These forecasts are used in market risk management, portfolio selection, market timing etc. and are used by other financial decision makers.

3.2.1. Financial Time Series Characteristics

Several characteristics for the volatility of asset returns have emerged throughout the years and these have been confirmed by numerous studies on the field of volatility modeling for the prediction of financial time series. A volatility model is expected to capture and reflect these properties of the series. These are,

- Volatility Clustering suggests that large changes in the price of an asset are often followed by other large changes and small changes are often followed by small changes.
- According to volatility persistence volatility comes and goes. Volatility persistence states an existence of normal level of volatility. Volatility will go up and down, but in a reasonable time, it will converge to the normal level of volatility. Persistence in volatility implies that current information has no effect on the long run forecast.
- Innovations may have an asymmetric effect on the volatility. The sign of the innovation is very important for the volatility model. Negative shocks have been shown to be more volatility prone. This is known as the "asymmetric effect" first noted in [37], refers to the fact that changes in stock prices tend to be negatively correlated with changes in volatility. In general, negative return shocks raise more volatility than positive return shocks of same magnitude.

Financial Time Series, distribution of returns, is fat-tailed and exhibits leptokurtosis. In general, there is evidence for the negative relationship between the stock market volatility and fixed investment.

3.2.2. Volatility in Turkish Stock Market (BIST)

Research points to the fact that managers will take market volatility into account when they make investment decisions. In general, there is evidence for the negative relationship between the stock market volatility and fixed investment. Large stock market price fluctuations are related to low growth in real fixed investment. In Turkey, stock market volatility may have led to a reduction in the capital stock and hence long-run productivity and income growth. A more stable stock market will better serve as the forecasting mechanism for the economy as well as fulfill its role in challenging savings into capital investment [25].

A historical analysis of the financial crises of the Turkish economic history indicates that serious examples of financial instability are always associated with substantial deterioration in the balance sheets of firms, households and banks. Thus, increased financial volatility that is not linked to the deterioration in balance sheets is unlikely to produce financial instability, which has harmful effects on the economy. The large fluctuations cannot be fully explained by the macroeconomic parameters such as inflationary money growth and excess consumption. They may be generated by trading activity as well as by political risk. Stock market engineering makes volatility desirable in a rising market to the extent that volatility is not perfectly correlated with loss [25].

3.2.3. Risk Management

Two important developments, one in academia and one on Wall Street have facilitated the advancement in knowledge about risk management. The development of volatility models for measuring and forecasting volatility dynamics was proposed by Engle [73]. RiskMetrics^{TM} by JP Morgan has enabled companies with just a minimum of computational power and technical ability to compute simple measures of market

risk for a given portfolio of assets. Value at Risk in RiskMetrics is an estimation of likely losses, which could arise from changes in market prices. More precisely, it is defined as the money-loss in a portfolio that is expected to occur over a pre-determined horizon and with a pre-determined degree of confidence.

Measuring the risk on specific assets has become increasingly important during the last decades. In a broad sense, companies want to have good control of their risk profile. The definition of risk and its translation into mathematics is of practical importance. Risk is generally defined in terms of the probability of returns and particularly of the returns variance. In the financial market this is of even greater importance. There has been rapid development of techniques for measuring and managing financial risk, partially motivated by recent financial disasters involving derivative securities over the last decade. One of the most popular approaches to risk measurement is by calculating what is known as an institution's 'Value at Risk' (VaR). Broadly speaking, Value at Risk is an estimation of likely losses, which could arise from changes in market prices. More precisely, it is defined as the money-loss in a portfolio that is expected to occur over a pre-determined horizon and with a pre-determined degree of confidence. The advantage of VaR is that it provides a single number which encapsulates the portfolio risk and which can be applied easily by non-technically minded financial risk managers. Explicitly, VaR expresses the expected loss resulting from potential adverse market movements with a specified probability over a period of time.

The roots of VaR' s popularity stem from the simplicity of its calculation, its ease of interpretation, and from the fact that VaR can be suitably aggregated across an entire firm to produce a single number which broadly encompasses the risk of the positions of the firm as a whole. Its origin can be traced to the "4:15 report" of Dennis Weatherstone, chairman of JP Morgan who demanded that a one page report be delivered to him every day summarizing the company's market exposure and providing an estimate of the potential loss over the next trading day. Dowd provides thorough introductions to VaR, and Brooks and Persand present recent discussions of VaR model estimation issues [34, 35, 75].

3.3. Traditional Financial Time Series Prediction

In the business and economics world, it is very important to accurately predict various kinds of financial variables to develop proper strategies and avoid the risk of potentially large losses.

Many economic time series do not have a constant mean and most show periods of tranquility followed by high volatility. Financial time series data do not have a constant mean and variance. These processes may contain both stochastic and deterministic components. A stochastic variable with a constant variance is called homoskedastic. Heteroskedasticity is a time-varying variance. For series exhibiting volatility, the unconditional variance may be constant even though the variance during some periods is unusually large. We have to decompose the series into its stochastic and deterministic components. Time series are not stationary; the sample means do not appear to be constant and/or there is the strong appearance of heteroskedasticity and they usually have a trend. For some series, increasing trend is interrupted by a decline and it is hard to maintain a time-invariant mean in these series. Any shock to these series displays a high degree of persistence causing the change over time. These series are called conditionally heteroskedastic.

3.3.1. Autoregressive Conditional Heteroskedasticity (ARCH)

The ARCH-model was first presented by Engle (1982) and since then received a lot of attention [73]. Professor R.F. Engle shared The Bank of Sweden Prize in Economic Sciences in Memory of Alfred Nobel 2003 with Professor Clive W.J. Granger for methods of analyzing economic time series with time varying volatility (ARCH). Engle's work explains how random fluctuations in the value of financial markets can be smoothed out, allowing the risk of holding shares to be calculated.

Conditionality of the variance implies a dependence on the observations of the past and autoregressive conditionality describes a feedback mechanism that incorporates past observations into the model. Consider an ordinary $AR(q)$ model of the

stochastic process y_k.

$$y_k = c + \alpha_1 y_{k-1} + ... + \alpha_p y_{k-p} + \epsilon_k \tag{3.3}$$

and it's functional form is defined as

$$y_k = E[y_k|\phi_{k-1}]\epsilon_k \tag{3.4}$$

where ϵ_k is a white noise disturbance term, ϕ_{k-1} is the information set available at time $k-1$ and $E[.|.]$ denotes the conditional expectation operator. This expectation is conditional to all past information available at time $k-1$. ϵ_k is the innovation process. The Autoregressive Conditional Heteroskedasticity (ARCH) process of Engle is any ϵ_k of the form

$$\epsilon_k = \sigma_k z_k \tag{3.5}$$

where ϵ_k is an independently and identically distributed (*i.i.d.*) process with zero mean and a variance equal to one [73]. By definition, ϵ_k is serially uncorrelated with mean zero, but its conditional variance equals σ_k^2 and, therefore, may change over time. The $ARCH(q)$ process is then,

$$h_k = \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{k-i}^2 \tag{3.6}$$

where $\alpha_0 > 0$, $\alpha_i > 0$, $i = 1, ..., q$ and h_k is the conditional variance represented also by σ_k^2.

3.3.2. Generalized Autoregressive Conditional Heteroskedasticity (GARCH) Models

Although ARCH models provide a good description of many return series, it suffers from a practical problem. This means that a large lag q is needed for the $ARCH(q)$ process to provide an adequate description of the returns, but as an $ARCH(q)$ process has $q + 1$ parameters, the immediate consequence of this is that a large number of parameters have to be estimated, all subject to parameter restrictions. This imposes a serious computational burden. To circumvent this problem, Bollerslev proposed the GARCH process [6].

GARCH is a mechanism that includes past variances in the explanation of future variances. More specifically, GARCH is a time series modeling technique that uses past variances and past variance forecasts to forecast future variances. A GARCH (Generalized Auto Regressive Conditional Heteroskedasticity) process is a common model used in time series analysis for analyzing stochastic volatility [3]. Specifically, a GARCH(1,1) model is the following:

$$y_k = \sigma_k z_k \tag{3.7}$$

$$z_k \sim N(0,1) \tag{3.8}$$

$$\sigma_k^2 = \omega + \alpha y_{k-1}^2 + \beta \sigma_{k-1}^2 \tag{3.9}$$

In this model y_k is the demeaned log return series, σ_k^2 represents the volatility and ω, α and β are the volatility parameters which control the effects of the log returns and historical volatility. ARCH parameter α and GARCH parameter β controls the persistence of a GARCH model which shows how fast volatilities decay after critical changes. This model assumes that ω, α and $\beta > 0$ and $\alpha + \beta < 1$.

Formally, the GARCH (p, q) process only differs from the ARCH (q) process in the way that the function σ_k^2, conditional variance, is specified:

$$\sigma_k^2 = \omega + \sum_{i=1}^{q} \alpha_i y_{k-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{k-j}^2 \tag{3.10}$$

where p integer, q integer

$$\omega > 0, \alpha_i > 0, i = 1, 2, ..., q \tag{3.11}$$

$$\beta_j > 0, j = 1, 2, ..., p \tag{3.12}$$

thus the additional feature is that the process now also includes lagged σ_k^2 values. For $p = 0$ the process is an $ARCH(q)$.

Figure 3.1 shows a generated synthetic series by GARCH model to explain effect of GARCH parameters. We divided the series into 5 different regimes. At each 200^{th} time step a changepoint occurs and the parameters of the GARCH model is changing. The first, third and fifth regimes have constant parameters of $\omega = 0$, $\alpha = 0.05$ and $\beta = 0.05$. As a result of this volatility shows smooth persistence. In the second regime we first see the effect of volatility constant by changing ω to $\omega = 1.75$. This controls range of the returns but does not have any changing effect on the volatility. To see the effect of parameter α, the parameter value is set to 0.85 in regime 3. Parameter α is the ARCH parameter and shows the effect of the log returns on volatility. Here the historical volatility is not incorporated yet as β is still the same as previous regimes. In regime 4, we revert back the ARCH parameter α to its original state and increasing historical volatility effect by changing the GARCH parameter β to 0.85. By the effect of historical volatility, the volatility started to increase with little fluctuations coming from log returns which are controlled by α. Figure 3.1 shows the changepoint locations, generated return series and the realization of volatility by different parameters. Here, the unconditional variance of returns is assumed to be $\omega/(1 - \alpha - \beta)$. We assume σ_0^2 at time $k = 0$ is to be the unconditional variance $\omega/(1 - \alpha - \beta)$. $(y_k \sim N(0, \sigma_k^2))$

GARCH has gained a lot of interest and is widely accepted. It takes into account excess kurtosis (i.e. fat tail behavior) and volatility clustering, two important characteristics of financial time series. It provides accurate forecasts of variances and covariance of asset returns through its ability to model time varying conditional variances. As a consequence, you can apply GARCH models to such diverse fields as risk management, portfolio management and asset allocation, option pricing, foreign exchange, and the term structure of interest rates. Although GARCH models are useful across a wide range of applications, they do have some limitations such as their in-

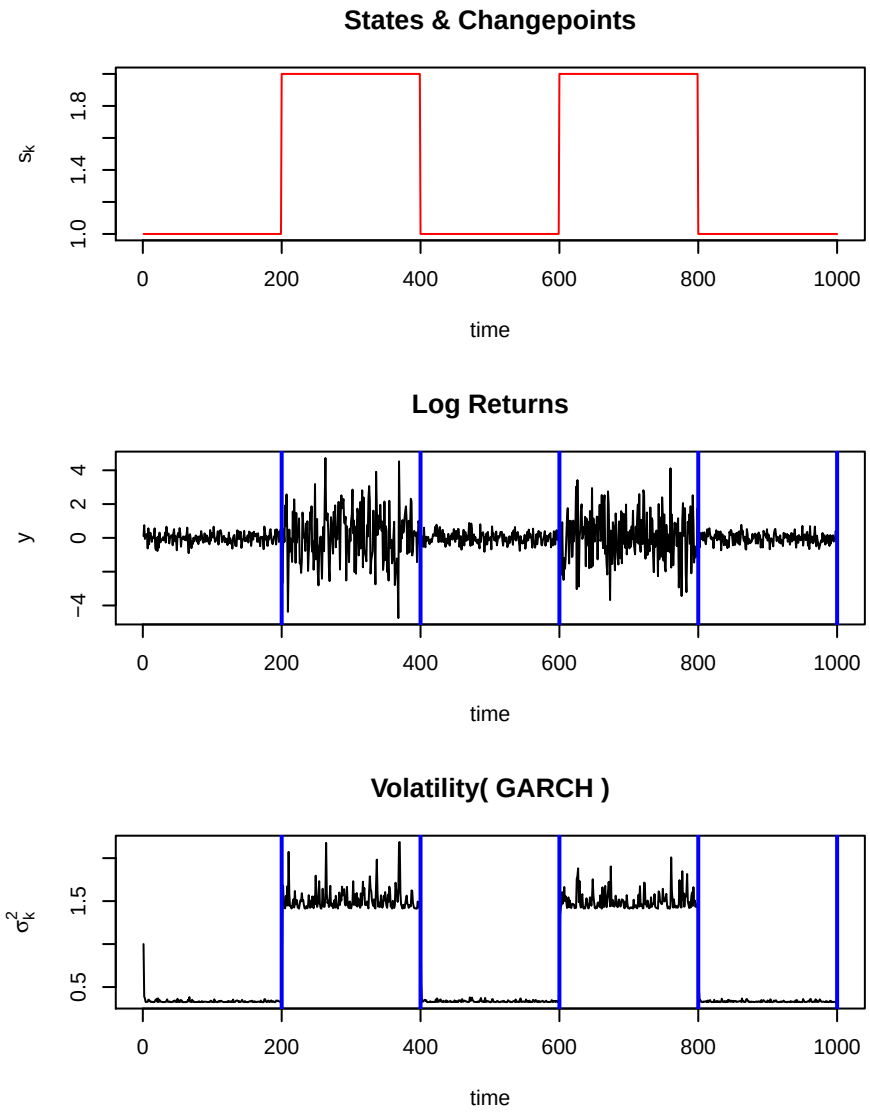


Figure 3.1. Volatility Realization by GARCH Parameters (changepoints, returns, volatility respectively).

ability to capture irregular market movements and their parametric specification that operates better under stable market conditions. GARCH models often fail to fully capture the fat tails observed in asset return series. Heteroskedasticity explains some of the fat tail behavior, but typically not all of it.

3.3.3. Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) Models

Asymmetric leverage volatility models incorporates the hypothesis that negative shocks cause more volatility than positive shocks. News Effect causing asymmetric volatility is a necessary result to be included in the estimations [25]. Despite their successful applications ARCH and GARCH models cannot capture some important facts in the data. The most important fact is the leverage or asymmetric effect discovered by Black and confirmed by the findings of French et. al., Nelson, Pagan and Engle [37, 76–79]. This effect claims that a bad news, a drop in price increases volatility more than an unexpected increase in price of similar magnitude. Because we take square of the innovations in ARCH and GARCH, we are unable to seize this effect and understand if there will be a difference from the point of view of volatility. This effect suggests that a symmetry constraint over y_k is not appropriate. Nelson proposed a model to capture such asymmetric effects called Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) [37]. This model is defined as below:

$$\log(\sigma_k^2) = \omega + \beta \log(\sigma_{k-1}^2) + \gamma y_{k-1}/\sigma_{k-1} + \alpha(|y_{k-1}|/\sigma_{k-1} - \sqrt{2/\pi}) \tag{3.13}$$

where ω, α, β and γ are constant parameters. This model is asymmetric and it is able to cover the effect of the sign of the returns because y_{k-1}/σ_{k-1} has a coefficient of γ. When this coefficient is negative, it will generate more volatility than positive return shocks because of the sign effect. The EGARCH model allows good news and bad news to have a different impact on volatility on the other hand GARCH does

not. Bad news has a greater impact than good news. The EGARCH model also allows big news to have a greater impact on volatility than GARCH model. We here assume that observed time series is governed by multiple GARCH or EGARCH based processes. The time series is governed by multiple regimes and our aim is to estimate the changepoint locations and regime parameters between switching regimes.

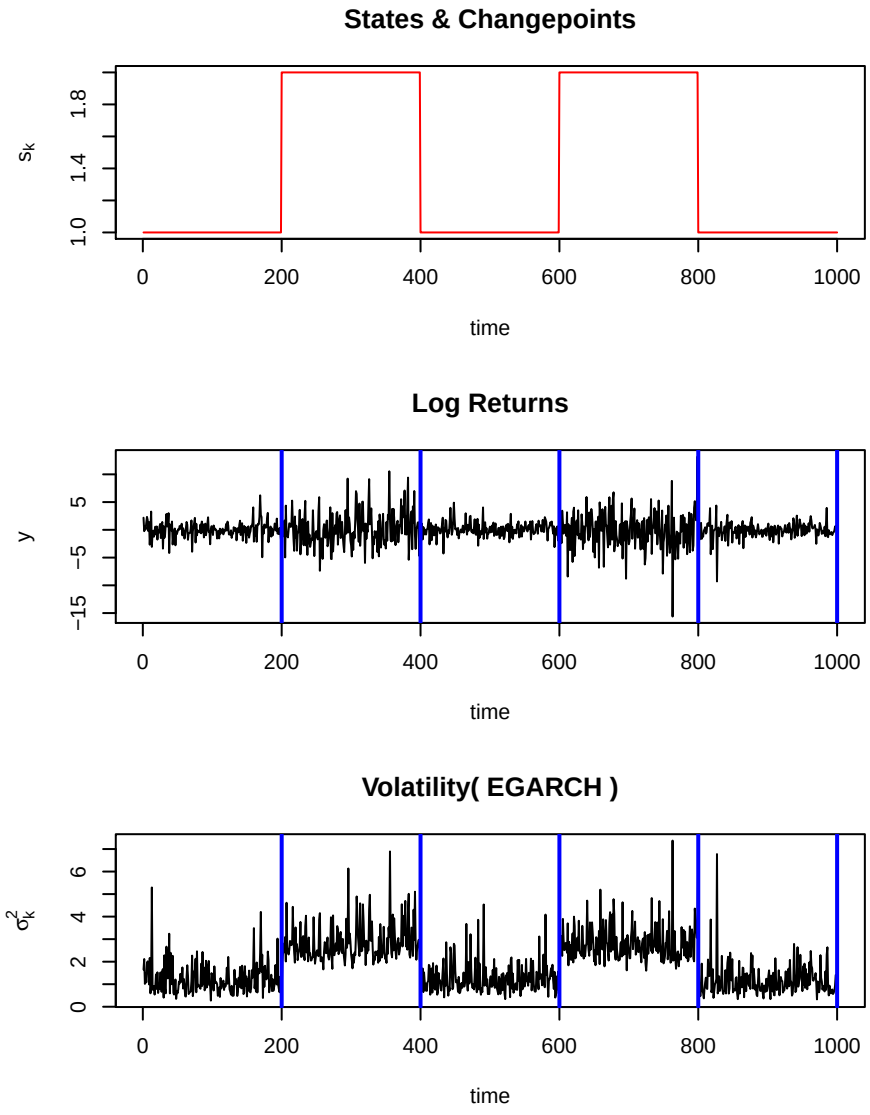


Figure 3.2. Volatility Realization by EGARCH Parameters (changepoints, returns, volatility respectively).

Figure 3.2 shows the changepoints, return series and realizations by using EGARCH model to explain effect of both GARCH and EGARCH parameters. The series is di-

vided into 5 different regimes with changepoints at each 200^{th} time step. The 1^{st}, 3^{rd} and 5^{th} regimes have constant parameters of $\omega = 0$, $\alpha = 0.05$ and $\beta = 0.05$. As a result of this volatility shows smooth persistence. In the 2^{nd} regime we first see the effect of volatility constant by changing ω to $\omega = 1.75$. This controls range of the returns but does not have any changing effect on the volatility. To see the effect of parameter α, the parameter value is set to 0.85 in regime 3. Parameter α is the ARCH parameter and shows the effect of the log returns on volatility. The α parameter represents a magnitude effect or the symmetric effect of the model. β measures the persistence in conditional volatility irrespective of anything happening in the financial market. When β is relatively large, then volatility takes a long time to die out following a crisis in the market. Here the historical volatility is not incorporated yet as β is still the same as previous regimes. In regime 4, we revert back the ARCH parameter α to its original state and increasing historical volatility effect by changing the GARCH parameter β to 0.85. By the effect of historical volatility, the volatility started to increase with little fluctuations coming from log returns which are controlled by α. The γ parameter measures the asymmetry or the leverage effect, the parameter of importance so that the EGARCH model allows for testing of asymmetries. If $\gamma = 0$ then the model is symmetric (Regimes 1, 3 and 5). When $\gamma < 0$, then positive shocks (good news) generate less volatility than negative shocks (bad news) (Regime 4). When $\gamma > 0$, it implies that positive innovations are more destabilizing than negative innovations (Regime 2). We assume σ_0^2 at time $k = 0$ is to be the unconditional variance $\omega/(1 - \alpha - \beta)$, $(y_k \sim N(0, \sigma_k^2))$.

4. STATE SPACE REPRESENTATION FOR MULTIPLE CHANGEPOINT MODELS

This chapter briefly introduces inference in probabilistic time series models and explains briefly Hidden Markov Models (HMM) and Linear Dynamical Systems (LDS) as probabilistic time series models. State space representation of multiple changepoint models are introduced and multiple changepoints for volatility models of GARCH and EGARCH are explained using graphical modeling techniques.

4.1. Inference in Probabilistic Time Series Models

Probabilistic inference is the problem of estimating the hidden variables (states or parameters) of a system in an optimal and consistent fashion as a set of noisy or incomplete observations of the system becomes available online.

The optimal solution to this problem is given by the recursive Bayesian estimation algorithm which recursively updates the posterior density of the system state as new observations arrive. This posterior density constitutes the complete solution to the probabilistic inference problem, and allows us to calculate any "optimal" estimate of the state. Unfortunately, for most real-world problems, the optimal Bayesian recursion is intractable and approximate solutions must be used.

In a probabilistic model of a time series $x_{1:K}$, the joint distribution of the observations is defined as $p(x_{1:K})$ [2,80,81]. In order the probabilistic model to be consistent with the causality of the time-series, we can utilize the chain rule and obtain the following recursion

$$p(x_{1:K}) = \Pi_{k=1}^{K} p(x_k|x_{1:k-1}) \tag{4.1}$$

Particularly we will be addressing sequential (recursive) probabilistic inference problem of financial time series prediction within nonlinear systems and non-Gaussian noise models. Usually, the State Space Model (SSM) of a time series if depicted as given in the Figure 4.1. x are the latent state variables and y are the observations.

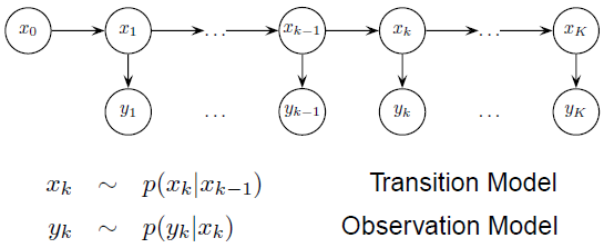


Figure 4.1. State space model graphical representation.

The hidden system state x_k with initial probability density $p(x_0)$, evolves over time (k is the discrete time index) as an indirect or partially observed first order Markov process according to the conditional probability density $p(x_k|x_{k-1})$. The observations y_k are conditionally independent given the state and are generated according to the conditional probability density $p(y_k|y_{k-1})$. The SSM can also be written as a set of nonlinear system equations

- State (Transition) equation: $x_k = f_x(x_{k-1}, u_k)$
- Observation equation: $y_k = f_y(x_k, v_k)$

where x_k is the state vector at time instant k, f_x is the state transition function, u_k is the process noise with known distribution, y_k are the observations at time instant k, f_y is the observation function and v_k is the observation noise with known distribution. The basic SSM is known as the Hidden Markov Models (HMM) [82] and it is depicted as the Figure 4.2.

In this HMM, there is one discrete hidden node, which is depicted as squares and one discrete or continuous observed node per time slice. X are the hidden variables, and Y are the observations. Structure and the parameters remains the same over time

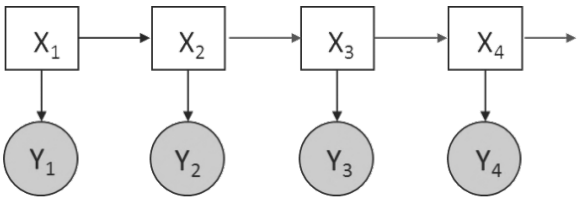


Figure 4.2. Hidden Markov Model.

in Hidden Markov Models.

Hidden Markov Models also have been directly for the purpose of time series forecasting. Rafiul Hassan introduced a novel method of hybridizing HMM to solve a real-world problem of time series forecasting. In addition to using soft computing to improve the initialization of parameter values, he used the HMM's output to identify similar data patterns, which are then used for clustering and forecasting mechanisms [26]. Zhang developed an extension of the Hidden Markov Model (HMM) that addresses two of the most important challenges of financial time series modeling: non-stationary and non-linearity. Specifically, Zhang extended the HMM to include a novel exponentially weighted Expectation-Maximization (EM) algorithm to handle these two challenges [83]. Bayramlı used HMM and Kalman Filters (KF) for financial time series prediction and he provided a mixture HMM, KF model called KMM [84].

Rao and Hong has analyzed the usage of Hidden Markov Models and Support Vector Machines in financial applications. They first discussed conventional methods of EMH and technical indicators. Then, they showed the usage of HMM and SVMs to help investors and enable them to make informed decisions [85]. Zhang in his master thesis discussed the use of HMMs in prediction of financial time series. He used S&P 500 Index from 1994 to 2002 in his studies [83].

The problem of inference in time series can be defined as follows: How do we optimally estimate the hidden system variables in a recursive fashion as incomplete and noisy observations become available online? In Bayesian approach, the posterior filtering density $p(x_k|y_{1:k})$ of the state given all observations $y_{1:k}$ constitutes the com-

plete solution to the sequential probabilistic inference problem. This also allows us to estimate the conditional mean of the state

$$E(f(x_{0:k})) = \int f(x_{0:k}) p(x_{0:k}|y_{1:k}) dx_{0:k} \tag{4.2}$$

We will go into details of how recursive Bayesian filtering approach will help estimation of financial time series problems in SMC methods. The optimal solution to recursively update the posterior density as new observations arrive is applying the recursive Bayesian estimation algorithm.

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})} \tag{4.3}$$

This recursive solution is usually only tractable for linear and Gaussian models in which exact inference is well known with Kalman Filter (KFL). For real world problems such as financial time series prediction, this solution is not tractable and approximation techniques have to be used. We will explain these in importance sampling and SMC sections.

4.1.1. Linear Dynamical System (LDS) - Kalman Filters

One of the tools that can be used for stochastic estimation from noisy sensor measurements is known as the Kalman Filter (KFL) or as Linear Dynamical Systems (LDS). LDS defines a probabilistic model that captures the time evolution and measurement processes and applies inference and learning approaches. KFL has the same topology as an HMM when they are represented as graphical models, but the state variable in LDS is continuous. KFL is the basic continuous Dynamic Bayesian Net-

work (DBN) and uses a continuous state variable with linear-Gaussian dynamics and measurements [86].

KFL is also used for Bayesian filtering and probabilistic inference and allows us to estimate the latent state of the system with linear-Gaussian distribution assumptions in all nodes. With this assumption, the probabilistic Bayesian filtering computations are tractable and in KFL it is shown that exact inference is possible unlike non-linear scenarios that we will discuss in the further sections.

The model in LDS is easily shown as a tree-structured directed graph and inference problems can be easily solved using the sum-product algorithm. The forward recursions like the alpha messages in HMM are Kalman Filter equations [87] and backward recursions like the beta messages in HMM are known as the Kalman smoother equations or Rauch-Tung-Striebel (RTS) equations [88].

The model has linear Gaussian conditional distributions, the transition and emission distributions are

$$p(x_k|x_{1:k-1}) = N(x_k|Ax_{k-1}, \Gamma) \tag{4.4}$$

$$p(y_k|x_k) = N(y_k|Cx_k, \Sigma) \tag{4.5}$$

Initial latent variable also has Gaussian distribution

$$p(x_1) = N(x_1|\mu_0, V_0); \tag{4.6}$$

Linear equations are defined as below:

$$x_k = Ax_{k-1} + Bw_k \tag{4.7}$$

$$y_k = Cx_k + v_k \tag{4.8}$$

$$x_1 = \mu_0 + u \tag{4.9}$$

where the noise terms have the following distributions

$$w \sim N(w|0, \Gamma) \tag{4.10}$$

$$v \sim N(v|0, \Sigma) \tag{4.11}$$

$$u \sim N(u|0, V_0) \tag{4.12}$$

The parameters of the models are denoted as $\Theta = A, \Gamma, C, \Sigma, \mu_0, V_0$.

The learning and inference occurs in two recursive steps in LDS. First we update time and calculate the error covariance and then in the measurement update process we correct our estimation with the constant K. The formulations and the recursion steps are given below in time update (predict) and measurement update (correct) processes.

(i) Time update (predict) $\hat{x_k} = Ax_{k-1} + Bw_k$

Project the error covariance as follows: $\hat{P_k} = AP_{k-1}A^T + \Gamma$

(ii) Measurement update (correct)

$$K_k = \hat{P}_k C^T (C\hat{P}_k C^T + \Sigma)^{-1}$$
$$x_k = \hat{x_k} + K_k(y_k - C\hat{x_k})$$
$$Pk = (I - K_k C)\hat{P}_k$$

4.1.2. Importance Sampling

For most of the probabilistic models; exact inference in intractable. Generally, in non-linear and non-Gaussian models, computational complexity is high and exact inference is intractable. We need flexible and efficient numeric integration techniques.

For probabilistic models in time series prediction problems, posterior distribution over unobserved variables is the primary interest in evaluating expectations and making predictions. The fundamental problem is to find the expectation of function $f(x)$ with respect to its probability distribution $p(x)$. The components of x can be continuous or discrete. In financial time series prediction, we will deal with the unobserved components of volatility and so most of the time x will be continuous. In the case of continuous variables, the expectation will be

$$E[f] = \int f(x)p(x)dx \tag{4.13}$$

The general idea behind sampling and making this expectation calculable is to obtain a set of samples $x(m)$ where $m = 1, ..., M$ that are drawn independently from the distribution $p(x)$. This allow the expectation to be approximated by a finite sum rather an integral.

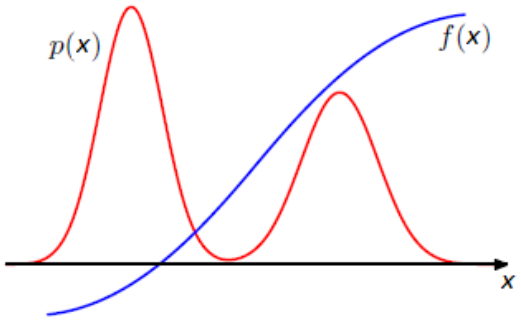


Figure 4.3. Schematic presentation of the expectation.

$$f = \frac{1}{M} \sum_{m=1}^{M} f(x^{(m)}) \tag{4.14}$$

Importance sampling provides a framework for approximating expectations directly but it does not itself provide a mechanism for drawing samples from the distribution $p(x)$.

The finite sum approximation depends on being able to draw samples from the distribution $p(x)$ but it is impractical to sample directly from $p(x)$ but we can evaluate $p(x)$ for any given x. The formula will be then

$$E(f(x)) \approx \frac{1}{M} \sum_{m=1}^{M} f(x^{(m)}) p(x^{(m)}) \tag{4.15}$$

Importance sampling is based on the use of proposal distribution $\pi(x)$ from which it is easy to draw samples and we can then define the expectation as

$$E(f(x)) \approx \int_x f(x)p(x)dx \approx \int_x f(x)\frac{p(x)}{\pi(x)}\pi(x)dx \tag{4.16}$$

$$E(f(x)) \approx \frac{1}{M}\sum_{m=1}^{M} f(x^{(m)})\underbrace{\frac{p(x^{(m)})}{\pi(x^{(m)})}}_{w^{(m)}} \tag{4.17}$$

$w^{(m)}$ are the importance weights and they correct the bias introduced by sampling from the wrong distribution.

In Importance Sampling, there are two steps:

- Draw samples from the proposal distribution $x^{(m)} \sim \pi(x)$
- Weight them according to how they fit to the original distribution

This is also shown in the Figure 4.4: The solid curve in the top figure denotes the real target distribution $p(x)$ and the dashed curve represents the proposal distribution $\pi(x)$. Weight function is calculated according to how the proposal distribution fit to the target distribution and this is represented in the second plot $w(x)$. The third figure shows the correspondences on the original distribution and gives us the particles and weights that are used for the approximation of the estimation.

4.2. State Space Representation for Multiple Changepoint Models

Many time series are characterised by abrupt changes in structure, such as sudden jumps in level or volatility. We consider change points to be those time points which divide a data set into distinct homogeneous segments. In practice the number of change points will not be known.

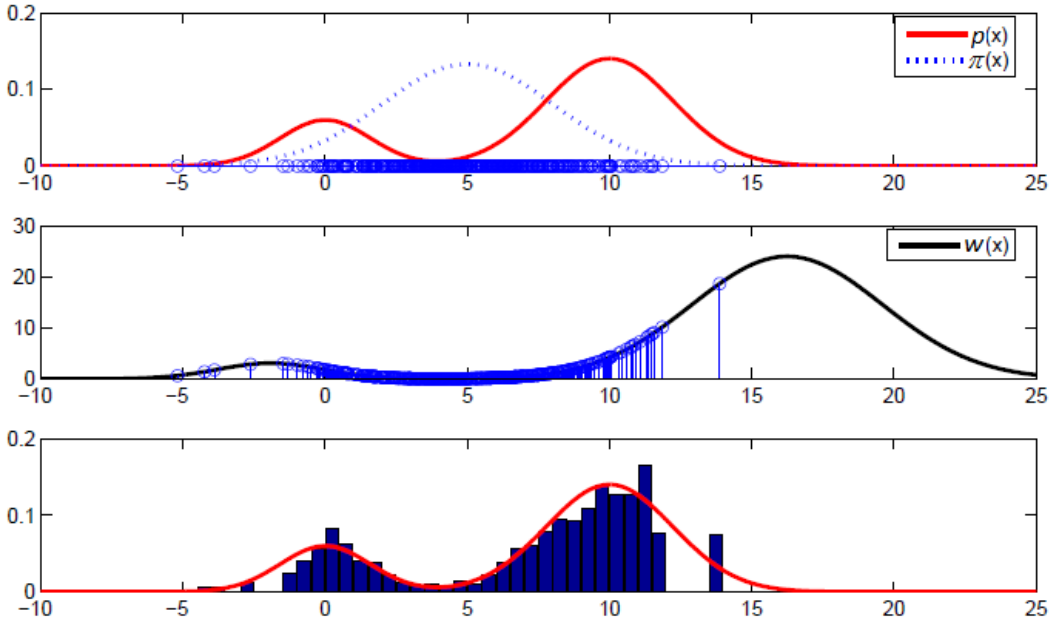


Figure 4.4. Importance Sampling [2].

State Space Model (SSM) of a time series is usually defined as s_k and y_k which are the latent state variables and the observations respectively. We will here use s_k as the latent state variable which represents the regime identifier. We here define the density of the observations as $p(y_k|y_{1:k-1}, \theta_k, s_k)$ where θ_k is the set of parameters at time k. The idea behind the model is the formulation of the changepoint model in terms of a latent variable that indicates the regime from which a particular observation has been drawn. The regimes of the process is defined first and the parameters θ_k are drawn by an unknown state variable s_k where $s_k = m$, $m \in (1, 2, \ldots, M)$ and $\theta_k = \theta_m$ which means the parameters θ_k at each time k are equal and same as the m^{th} regime θ_m. The latent variable is a discrete Markov process s_k with the transition probabilities P limited so that the state variable can either stay in the same regime or jump to a new regime [10]. In speech processing, such models are called Left-Right HMMs. Left-Right HMM which is also known as Bakis HMM can model signals changing over time like speech signals [82]. Figure 4.5 shows a 4-state left-right HMM model.

$$P = \begin{bmatrix} p_{11} & 1-p_{11} & 0 & 0 \\ 0 & p_{22} & 1-p_{22} & 0 \\ 0 & 0 & p_{33} & 1-p_{33} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.18}$$

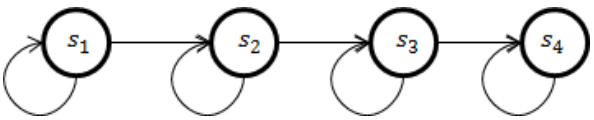


Figure 4.5. A 4-state left-right HMM model.

This changepoint model is selected in this dissertation in order to give a limit to the parameter space but there are also other techniques for state space representation of multiple changepoint models in the literature. Fearnhead assigned changepoint flags and used the model representation given in Figure 4.6. Here s_k represents the changepoint flag, σ_k is the latent variable and y_k is the observation [22].$[s_k = 0]$ denotes the Iverson bracket which acts as an indicator function and results to 1 if the statement inside the bracket is satisfied, and 0 otherwise.

$$s_k \sim p(s_k|s_{k-1}), \text{ Changepoint flags} \in \{0, 1\}$$

$$\sigma_k \sim [s_k = 0]f(\sigma_k|\sigma_{k-1}) + [s_k = 1]\pi(\sigma_k), \pi(\sigma_k) : \text{Reinitialization}$$

$$y_k \sim p(y_k|\sigma_k), \text{ Observations}$$

Adams and Mackay used current "run length" or the time since the last changepoint to estimate the posterior distribution. Here r_k denotes the length of the current run at time k. Figure 4.7 shows the run length r_k as a function of time and r_k drops to zero when a changepoint occurs [3, 15].

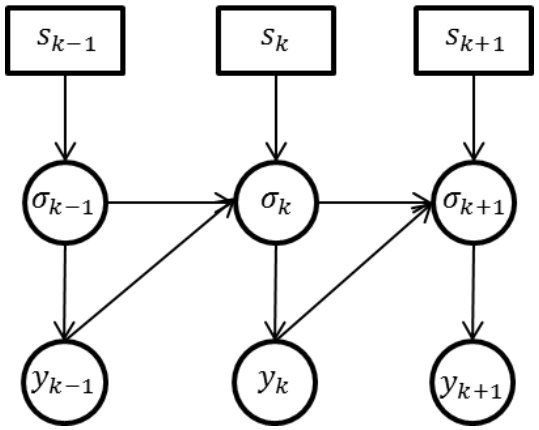


Figure 4.6. Graphical model for changepoints.

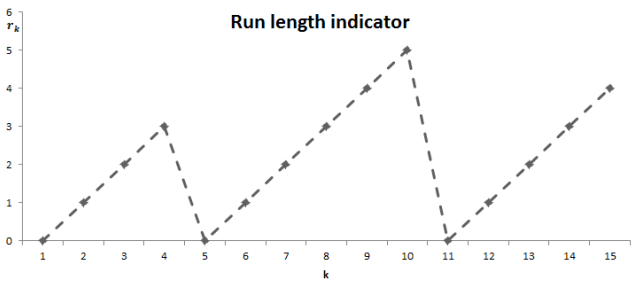


Figure 4.7. Run length indicator r_k [3].

4.3. Multiple Changepoints for Volatility Models

We used partial and full changepoint models where only subset of or all the parameters are subject to change. Both GARCH and exponential GARCH models are used as the volatility processes of the time series. We first used Normal distribution but empirical studies often suggest fat tails in the distribution of asset returns, therefore, we extended the proposed solution to incorporate student-t distribution with ν degrees of freedom as well. First we calculated volatility using GARCH [3] and Nelson's EGARCH [37]. Residuals and volatility is used in the process model and measurement model equations. Measurement and process model equations are adapted to GARCH and EGARCH models. SMC methods are implemented using the measurement and process model equations in Sections 3.3.2 and 3.3.3. Figure 4.8 shows the graphical representation for the multiple changepoint model defined in this dissertation.

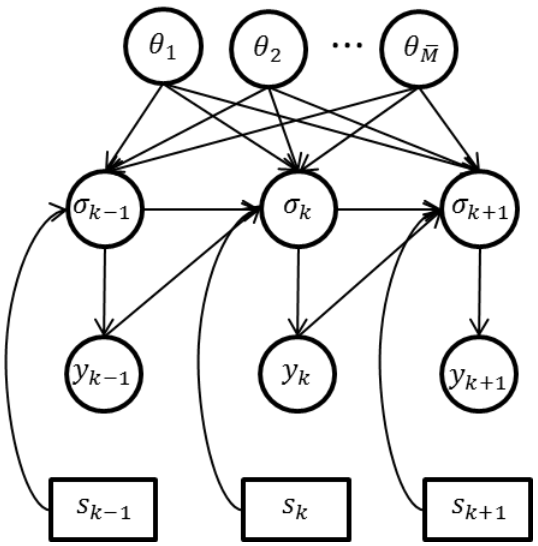


Figure 4.8. Graphical representation for the multiple changepoint model.

In this graphical model s_k and y_k represents the latent variables and the observations respectively where s_k is the regime identifier. $\theta = [\theta_1, \theta_2, \ldots, \theta_{\bar{M}}]$ is the set of parameters at different regimes. Probability densities of state and observations are given below:

$$s_k \sim p(s_k|s_{k-1}, \theta) \tag{4.19}$$

$$y_k \sim p(y_k|s_k, \theta) \tag{4.20}$$

The state model is a Markov process where it is conditional on the previous state and the parameters θ, and the observation at time k is conditional on state s_k. In order to determine the state at any given point k, it requires to estimate the filtered posterior distribution $p(s_k|y_{1:k}, \theta)$ recursively in time. The posterior distribution of the filtered states can be derived by using Bayes Rule as

$$p(s_k, \theta|y_{1:k}) = \frac{p(y_k|s_k, \theta)p(s_k, \theta|y_{1:k-1})}{p(y_k|y_{1:k-1}, \theta)} \tag{4.21}$$

where

$$p(s_k, \theta|y_{1:k-1}) = \int p(s_k|s_{k-1}, \theta)p(s_{k-1}, \theta|y_{1:k-1})ds_{k-1} \tag{4.22}$$

and

$$p(y_k|y_{1:k-1},\theta) = \int p(y_k|s_k,\theta)p(s_k,\theta|y_{1:k-1})ds_k \tag{4.23}$$

We model the observations as GARCH or EGARCH processes by using ω, α, β and γ parameters. Let $\theta_m = [\omega_m, \alpha_m, \beta_m, P_{mm}]$ or $\theta_m = [\omega_m, \alpha_m, \beta_m, \gamma_m, P_{mm}]$ be the model parameters and s_m is the latent variable in state m and $\theta = [\theta_1, \theta_2, \ldots, \theta_{\bar{M}}]$ is the parameter space in all states and regimes. $s_k = s_m$ at time step k where k^{th} time step resides in regime m. The likelihood of y_k for the full changepoint model is then defined as

$$p(y_k|y_{1:k-1}, s_k, \theta) = p(y_k|y_{1:k-1}, \theta_{s_k}) = (2\pi\sigma_k^2)^{-1/2} \exp(-\frac{y_k^2}{2\sigma_k^2}) \tag{4.24}$$

where $y_k = \sigma_k z_k$, $z_k \sim N(0,1)$.

In full changepoint model all the parameters of the GARCH model are subject to change as given below.

$$y_k = \sigma_k z_k, \sigma_k^2 = w_{s_k} + \alpha_{s_k} y_{k-1}^2 + \beta_{s_k}\sigma_{k-1}^2 \tag{4.25}$$

$$\theta_m = [\omega_m, \alpha_m, \beta_m, P_{mm}] \tag{4.26}$$

For EGARCH models:

$$\log(\sigma_k^2) = \omega_{s_k} + \beta_{s_k} \log(\sigma_{k-1}^2) + \gamma_{s_k} y_{k-1}/\sigma_{k-1} + \alpha_{s_k}(|y_{k-1}|/\sigma_{k-1} - \sqrt{2/\pi}) \tag{4.27}$$

and $\theta_m = [\omega_m, \alpha_m, \beta_m, \gamma_m, P_{mm}]$.

In partial changepoint model only parameter w_{s_k} is subject to change. GARCH model is then defined as follows:

$$\sigma_k^2 = \omega_{s_k} + \alpha y_{k-1}^2 + \beta \sigma_{k-1}^2 \tag{4.28}$$

$$\theta_m = [\omega_m, \alpha, \beta, P_{mm}]. \tag{4.29}$$

where $y_k = \sigma_k z_k$, $\sigma_k^2 = \omega_{s_k} + \alpha y_{k-1}^2 + \beta \sigma_{k-1}^2$.

For EGARCH models:

$$\log(\sigma_k^2) = \omega_{s_k} + \beta \log(\sigma_{k-1}^2) + \gamma y_{k-1}/\sigma_{k-1} + \alpha(|y_{k-1}|/\sigma_{k-1} - \sqrt{2/\pi}) \tag{4.30}$$

and $\theta_m = [\omega_m, \alpha, \beta, \gamma, P_{mm}]$.

As we mentioned above empirical studies often suggest fat tails in the distribution of asset returns, therefore, we extended the proposed solution to incorporate student-t distribution with ν degrees of freedom.

Let $\theta_m = [\omega_m, \alpha, \beta, \nu, P_{mm}]$ or $\theta_m = [\omega_m, \alpha, \beta, \nu, \gamma, P_{mm}]$ be the model parameters

and s_m is the latent variable in state m and $\theta = [\theta_1, \theta_2, \ldots, \theta_{\bar{M}}]$ is the parameter space in all states and regimes. The density of y_k for the partial changepoint model where $y_k = \sigma_k z_k$, $z_k \sim N(0,1)$ is

$$p(y_k|y_{1:k-1}, s_k, \theta) = p(y_k|y_{1:k-1}, \theta_{s_k}) = \frac{\Gamma((\nu+1)/2)}{\sigma_k \Gamma(\theta/2)\sqrt{\pi\nu}}(1 + \frac{y_k^2}{\nu\sigma_k^2})^{-}(\frac{\nu+1}{2}) \qquad (4.31)$$

$$\text{where } y_k = \sigma_k z_k,\ \sigma_k^2 = \omega_{s_k} + \alpha y_{k-1}^2 + \beta\sigma_{k-1}^2.$$

Full changepoint model and EGARCH models applies as the same in the normal distribution.

5. BAYESIAN CHANGEPOINT AND TIME-VARYING PARAMETER LEARNING IN REGIME SWITCHING VOLATILITY MODELS

Most of the real world problems require non-linear, non-Gaussian scenarios and in most real-world applications, the integrations with respect to s_{k-1} and s_k in (4.22), (4.23) and the implementation of Bayes' theorem in (4.21) are both analytically intractable and/or computationally costly. Hidden Markov Models (HMM) and Linear Dynamical Systems (LDS) systems fail into providing tractable filtering solutions for these type of non-linear and non-Gaussian scenarios. In this dissertation, a probabilistic time series modeling and changepoint detection approach for stock market time series using non-linear and non-Gaussian noise scenarios is introduced.

5.1. Sequential Monte Carlo (SMC) methods

Since their introduction in 1993, particle filtering techniques have become a very popular class of numerical methods for the solution of optimal estimation problems in non-linear non-Gaussian scenarios [89]. Recently, particle based sampling filters have been proposed and used successfully to recursively update the posterior distribution using sequential importance sampling and resampling [90–92].

The particle filter is an SMC method used for Bayesian filtering. Particles with corresponding weights are used to form an approximation of a probability density function (PDF). The particles are propagated over time by Monte Carlo simulation to obtain new particles and weights (usually as new information are received), hence forming a series of PDF approximations over time [21, 22]. SMC methods, particle filtering is based on a non-linear state space representation and is a method used for state estimation. Particle filtering is a technique for implementing for recursive Bayesian filtering by using Monte Carlo based sampling techniques. The idea is to approximate the posterior density by a set of random particles associated with weights. After defining

these particles using a proposal distribution the posterior density estimates are computed which are based on these particle samples and weights. So it is also applicable and a powerful method for filtering and/or prediction of time series problems. Particle filtering handles non-linear models with non-Gaussian noise. SMC methods provide a Monte Carlo based sampling method in itself. They approximate the target probability distribution (e.g. amplitude of speech signal and stock market returns). This method is also known as bootstrap filter [89], the survival of the fittest [93] or the condensation algorithm [94].

Particle filtering is a technique for implementing for recursive Bayesian filtering by using Monte Carlo based sampling techniques. The idea is to represent the posterior density by a set of random particles associated with weights. After defining these particles using a proposal distribution we compute the posterior density estimates based on these particle samples and weights. The approach is given in the Figure 5.1. First the problem is modeled as a state space model. The idea is to estimate the posterior distribution but the integrals in previously defined filtering approaches are not tractable. So we will need a sampling method to compute the estimation. It is difficult to draw samples from an unknown distribution. For these purpose, we incorporate importance sampling techniques which are defined in the previous section.

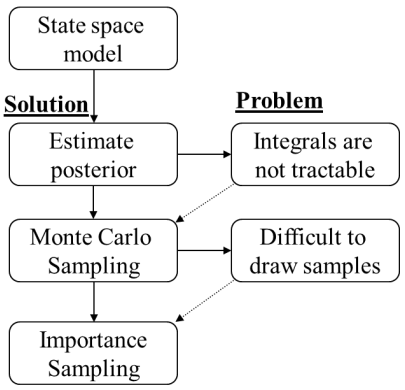


Figure 5.1. Steps for Sequential Monte Carlo (SMC) methods.

The state sequence and the observation sequence is a Markov random process.

We model the state (transition) equation as $s_k = f_s(s_{k-1}, u_k)$ where s_k is the state vector at time instant k , f_s is the state transition function and u_k is the process noise with a known distribution. The observation sequence is modeled as $y_k = f_y(s_k, v_k)$ where y_k represents the observation vector at time instant k, f_y is the observation function and v_k is the observation noise with a known distribution.

State (Transition) equation: $s_k = f_s(s_{k-1}, u_k)$

- s_k state vector at time instant k
- f_s state transition function
- u_k process noise with known distribution

Observation equation: $y_k = f_y(s_k, v_k)$

- y_k observations at time instant k
- f_y observation function
- v_k observation noise with known distribution

5.1.1. Sequential Monte Carlo (SMC) methods and Bayesian Filtering

The particle approximation of the posterior distribution is updated recursively by propagating and updating the particles according to filtering and predictive densities. In order to calculate the estimation we used recursive Bayesian filtering techniques. We first apply filtering to calculate the Filtering Density using Bayes Rules in (4.21) and calculate the Prediction Density by integrating transition probability and filtering densities in (4.22).

In summary Particle Filtering can be seen as a two steps process. This dissertation first represents the posterior distribution $p(s_k|y_{1:k})$ expressed as a mixture distribution of particles $s_k^{(i)}$ shown schematically as circles whose size are proportional to the corresponding weights $w_k^{(i)}$. When a new observation occur a set of new N samples from the distribution and the new observation is used to evaluate the new weights $w_{k+1}^{(i)}$ using

$p(y_{k+1}|s_{k+1}^{(i)})$. This is illustrated in the Figure 5.2. After this step, a resampling step is included where the particles are resampled in order to duplicate the particles with high weights and remove the particles with low weights. This step enables the particle filtering approach to remove degeneracy of the particle approximation.

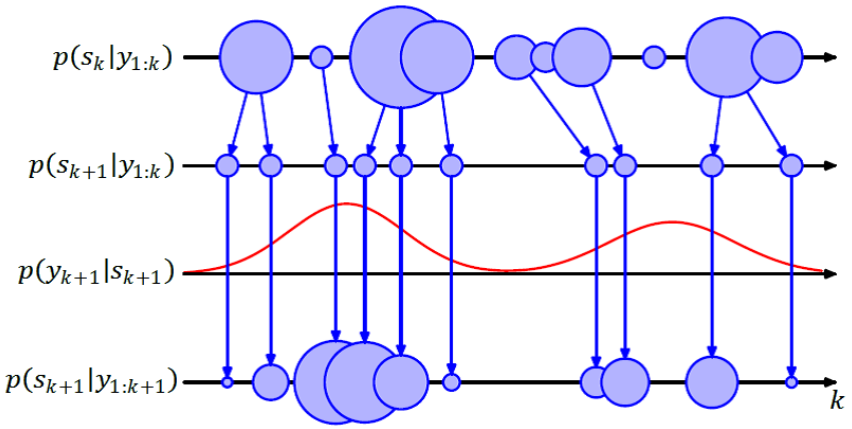


Figure 5.2. Illustration of Particle Filtering operations [4].

For prediction, we need to estimate the posterior distribution. For calculating the posterior distribution, we need to estimate the unknown state s_k based on a sequence of observation $y_{1:k}$. We can calculate the estimate of the posterior distribution using the integral given below:

$$E(f(s)) = \int f(s_{0:k})p(s_{0:k}|y_{1:k})ds_{0:k} \tag{5.1}$$

In order to calculate the estimation we will use recursive Bayesian filtering technique and first apply update and the propagate steps. These steps are illustrated as given in Figure 5.3.

We first apply filtering to calculate the Filtering Density using Bayes Rules as follows:

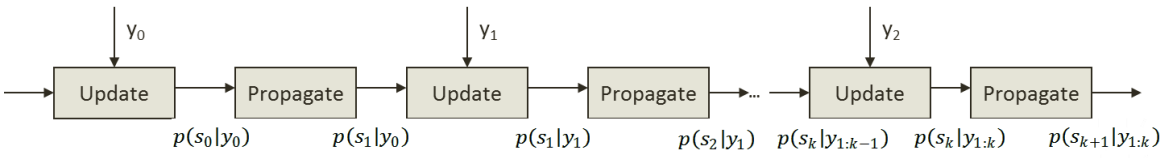


Figure 5.3. Filtering and Prediction Steps in Bayesian Estimation.

$$p(s_k|y_{1:k}) = \frac{p(y_k|s_k)}{p(y_k|y_{1:k-1})} p(s_k|y_{1:k-1}) \tag{5.2}$$

Then predict the next state and calculate the Prediction Density by integrating transition probability and filtering density.

$$p(s_{k+1}|y_{1:k}) = \int p(s_{k+1}|s_k) p(s_k|y_{1:k}) ds_k \tag{5.3}$$

5.1.2. Sequential Importance Sampling (SIS) & Resampling (SIR)

Sequential Importance Sampling is the basis of the Sequential Monte Carlo (SMC) methods. The idea is to first update filtering density using Bayesian filtering and then compute integrals using importance sampling.

Here in this dissertation we used the prior as the proposal distribution $s_k^{(i)} \sim p(s_k|s_{k-1})$ and weight computation and normalization is done as follows:

$$\hat{w}_k^{(i)} = w_{k-1}^{(i)} p(y_k|s_k^{(i)}) \tag{5.4}$$

$$w_k^{(i)} = \frac{\hat{w}_k^{(i)}}{\Sigma_{i=1}^{N} \hat{w}_k^{(i)}} \tag{5.5}$$

But, the SIS algorithm has some limitations. The variance of the importance weights increase stochastically over time [95]. This variance increase poses problems because the proposal density is preferred to be as close as to the posterior density and causes weight degeneracy with harmful effects on the accuracy of the simulations. Calculating the estimation on the same weight and sample set also wastes the computation resources. To address this degeneracy problem of SIS, a selection, resampling process is used to eliminate particles with low weights and multiply the particles with high important weights. There are different sampling approaches including residual, systematic, multinomial resampling and stratified sampling. In SIR, particles $s_k^{(i)}$ are replicated in proportion to their weights $w_k^{(i)}$ and produce N new samples all with equal weights $1/N$, $\{s_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}$.

5.2. Auxiliary Particle Filtering (APF)

SIR approach often wastes a lot of computation time on generating particles that have small weights. In order to solve this weight degeneracy problem, we can alternatively incorporate other information which will put more mass on relevant particles. This should improve how our algorithm performs. This is the motivation for auxiliary particle filters (APF), developed by Pitt and Sheppard (1999) [96].

Instead of applying a blind approach, we aim to update the most promising particles. The idea here is to use some form predictive power such as the likelihood in form $p(y_{k+1}|\mu_{k+1})$ in particle selection process to remove the weight degeneracy problem. We setup a $\mu_{k+1}^{(i)}$ as an estimate of the mode of new state s_{k+1}, the mode of the transition density $p(s_{k+1}|s_k^{(i)})$ for each particle. $i = 1, 2, 3, ...N$. The quality of the

estimate is evaluated by the auxiliary weights:

$$\delta_{k+1}^{(i)} = w_k^{(i)} p(y_{k+1}|\mu_{k+1}^{(i)}) \tag{5.6}$$

A large auxiliary weight indicates better representation power for the related particles in accordance to the underlying process. For particle selection, auxiliary weights are normalized and they are replaced by new samples with probabilities proportional to $\delta_{k+1}^{(i)}$ where $\{j\}$ is the new indices and $\{N^{'}\}$ represents the size of the new particle set. We kept the size of the sample set constant $\{N^{'}\}$ as the size of the particle set size $\{N\}$ throughout this dissertation and will use N as the new particle set size as well but it is also possible to change the size of the sample set at each time point. The generated set of indices $\{j\}$ represents the set of most likely paths of the volatility process will take given the arrival of the new observation y_{k+1}. We will use the new selected sample set to generate the new particles $s_{k+1}^{(j)} \sim p(s_{k+1}|s_k^{(j)})$. New auxiliary weights are evaluated as follows:

$$w_{k+1} = \frac{p(y_{k+1}|s_{k+1}^{(j)})}{p(y_{k+1}|\mu_{k+1}^{(j)})} \tag{5.7}$$

5.3. Mixture Kernel Smoothing (MKS) Based Parameter Filtering

State filtering problem is turned into a general joint state and parameter filtering problem. In this dissertation, we applied Liu and West's Mixture Kernel Smoothing approach for approximating the densities [97]. We will first introduce the regime parameter values at time k, $\theta_k^{(i)}$, and provide a joint importance sample set of state and regime parameter values for the model $\{s_k^{(i)}, \theta_k^{(i)}\}_{i=1}^N$ with associated weights $\{w_k^{(i)}\}_{i=1}^N$. The aim is converted to approximate a joint posterior $p(s_{k+1}, \theta|y_{1:k})$ instead of just a posterior for the states. We can decompose the joint posterior into following three

factors:

$$p(s_{k+1}, \theta | y_{1:k+1}) = p(y_{k+1}|\theta, s_{1:k+1}) p(s_{k+1}|\theta, y_{1:k}) p(\theta|y_{1:k}) \tag{5.8}$$

This is the product of the marginal likelihood given the state and parameters, prediction of latent variables given the parameters and past data and the posterior density for the regime parameters given the past data respectively. If we know the parameter values, Eq. 5.9 simplifies just to the state filtering problem. Here we follow Liu West and implement Mixture Kernel Smoothing (MKS) approach for approximating the density of $p(\theta|y_{1:k})$. We will have Monte Carlo samples, regime parameters $\theta_k^{(i)}$ and associated weights $\{\omega_k^{(i)}\}_{i=1}^N$ at time k from the posterior distributions. We approximate the parameter density by a smooth kernel density as follows:

$$p(\theta|y_{1:k}) = \Sigma_{i=1}^N w_k^{(i)} N(\theta|m_k^{(i)}, b^2 V_k) \tag{5.9}$$

where $N(\theta|m, V)$ is a multivariate normal distribution with mean vector m and covariance matrix V. Kernel smoothing method approximates the poster density for θ by a mixture of multivariate normal distributions and the weights that comes from the particles. b^2 is the kernel shrinkage parameter in Liu and West (2001) and is usually chosen to be $(0, 1)$ and usually decreases slowly as the particle set size increases so that all the parameter estimates are concentrated closer to the mean. V_k is the particle set variance and θ_k is the particle set mean and they are computed as follows:

$$V_k = \Sigma_{i=1}^N w_k^{(i)} (\theta_k^{(i)} - \bar{\theta}_k)(\theta_k^{(i)} - \bar{\theta}_k)^T \tag{5.10}$$

$$\bar{\theta}_k = \Sigma_{i=1}^N w_k^{(i)} \theta_k^{(i)} \tag{5.11}$$

Liu and West‘s shrinkage method is used to remove degeneracy problem which

will occur if $m_k^{(i)}$ is considered to be just $\theta_k^{(i)}$. Liu and West took $m_k^{(i)} = a\theta_k^{(i)} + (1-a)\bar{\theta}_k$ where $a = \sqrt{1-b^2}$. This mean value makes the centres of Gaussian centres to be closer and as a result the distribution will have thin tails. He and Maheu used a discount factor $\delta \in (0,1)$ to control the shrinkage of the kernel means as $b^2 = 1 - [3\delta - 1/2\delta]^2$ and $a = \sqrt{1-b^2}$ [14].

5.4. Auxiliary Particle Filtering based Multiple Changepoint Detection in GARCH/EGARCH Models

In this dissertation, APF is combined with kernel smoothing to design a SMC approach for MCD problems in financial time series. The proposed method is an APF based Bayesian inference approach to the MCD in financial time series using volatility models of GARCH and EGARCH.

The parameters θ are re-parameterized to take values on the real line since they will be sampled through a mixture of normal kernels. The repameterization approach is explained in the experiments sections in details. We will use θ for the re-parameterized values. The number of regimes and changepoints in the series are not known since we are dealing with the problem of estimating GARCH and EGARCH based volatility models which are subject to an unknown number of changepoints using Sequential Monte Carlo methods. A maximum number, an upper bound for the number of changepoints is defined which also represents the number of states, $\bar{M}$.

We studied over several alternatives including partial changepoint models where only a part of the parameters are subject to change or full changepoint models where all the parameters are subject to change. We used both GARCH and exponential GARCH models as the volatility processes of the time series in this dissertation. The EGARCH model allows good news and bad news to have a different impact on volatility on the other hand GARCH does not. Bad news has a greater impact on the volatility than good news. The EGARCH model also allows big news to have a greater impact on volatility than GARCH model. We first used Normal distribution in our studies but empirical studies often suggest fat tails in the distribution of asset returns, therefore,

we extended the proposed solution to incorporate student-t distribution with ν degrees of freedom as well. We define all these alternative models and the algorithm proposed below.

5.4.1. Auxiliary Particle Filtering based Multiple Changepoint Detection Algorithm

The proposed approach is used to approximate $p(s_k, \theta|y_{1:k})$ given a set of particles $\{s_k^{(i)}, \theta_k^{(i)}\}_{i=1}^N$ and weights $\{w_k^{(i)}\}_{i=1}^N$ for each regime where N is the size of the particle set. Latent variables s_k, and parameter estimations are done using the SMC approach in the algorithm below. Latent volatilites σ_k^2 are inferred through GARCH and EGARCH recursions using Equations 4.25 and 4.27.

The algorithm of the proposed approach is given below:

For $\forall i$, $i = 1, 2, \ldots N$ we define $\mu_{k+1}^{(i)}$ as the mode of the prediction of the latent variable $p(s_{k+1}|s_k^{(i)}, \theta_k^{(i)})$.

(i) Produce stratified uniform random variables $\{u_i\}_{i=1}^n$ within uniform intervals $u_i \sim \bigcup(\frac{i-1}{n}, \frac{i}{n})$

(ii) Compute the auxiliary weights $\delta_{k+1}^{(i)} \sim \omega_k^{(i)} p(y_{k+1}|y_{1:k}^{(i)}, m_k^{(i)})$ where $m_k^{(i)} = a\theta_k^{(i)} + (1-a)\bar{\theta}_k$ and $\bar{\theta}_k = \Sigma_{i=1}^N \omega_k^{(i)} \theta_k^{(i)}$ and draw a sample from $i = 1, 2, \ldots N$ with the corresponding auxiliary weights.

(iii) Sample the new regime parameter set from the k^{th} normal component of the kernel function given:

$$\theta_{k+1}^{(i)} \sim N(m_k^{(i)}, b^2 V_k) \tag{5.12}$$

$$V_k = \Sigma_{i=1}^N w_k^{(i)} (\theta_k^{(i)} - \bar{\theta}_k)(\theta_k^{(i)} - \bar{\theta}_k)^T \tag{5.13}$$

$$b^2 = 1 - [3\delta - 1/2\delta]^2 \tag{5.14}$$

(iv) Sample a new state value

$$s_{k+1}^{(i)} \sim p(s_{k+1}^{(i)} | s_k^{(i)}, \theta_{k+1}^{(i)}) \tag{5.15}$$

(v) Correct the weights

$$\omega_{k+1}^{(i)} \sim \frac{p(y_{k+1} | s_{k+1}^{(i)}, \theta_{k+1}^{(i)})}{p(y_{k+1} | \mu_{k+1}^{(i)}, m_k^{(i)})} \tag{5.16}$$

(vi) Repeat from 2^{nd} to 5^{th} steps using stratified sampling approach defined below. Stratified sampling here is used to stabilize estimates over multiple runs [14]. Stratified Sampling is based on ideas used in survey sampling consists in pre-partitioning the $(0, 1]$ interval into n disjoint sets. To produce a new sample size of $\{n\}$ from a population $\{s_k\}_{k=1}^K$ with weights $\{w_k\}_{k=1}^K$, stratified sampling first produces stratified uniform random variables $\{u_i\}_{i=1}^n$ within uniform intervals $u_i \sim \bigcup(\frac{i-1}{n}, \frac{i}{n})$. u_i's are drawn independently in each of these sub-intervals.

6. EXPERIMENTS & RESULTS

To analyze the performance of the proposed approach, we setup two simulation experiments, one with synthetic data and the other one with an emerging market time series, BIST 100 which is described in Chapter 2, using GARCH and EGARCH volatility models with both Gaussian and Student-t distributions and using partial and full parameter changepoint models where selected parameters or all the parameters of the volatility models are changing in different regimes. We also compared the results of the proposed APF based multiple changepoint detection algorithm with He and Maheu's GARCH based model and reported results below. The results for the synthetic data and real time series data are represented respectively.

6.1. Experiments on Synthetic Data Series

6.1.1. Experiments on GARCH Synthetic Data Series

Here in this experiment, synthetic time series is generated using a GARCH volatility model with Gaussian distribution and full parameter changepoint modeling. This time series includes switching regimes by providing different ω_k, α_k, β_k parameters at changepoint locations. Synthetic time series is generated using a predefined left-right HMM model. The parameter configuration of the algorithm includes the maximum number of the changepoints, prior parameters for each regime and volatility model parameter sets. First the maximum limit for the number of changepoints (the number of regimes) is defined. This has not to be the number of the exact number of changepoints but will be defined as the maximum number of changepoint limit that the algorithm can reach. Three different regime parameters are defined for each regime, $\theta_m = [\omega_k, \alpha_k, \beta_k]$. The upper bound for the number of states is defined as $M = 5$. State transition probabilities are defined as P. The prior parameters for each regime are randomly generated as $\omega_k \sim Gamma(1, 0.5)$, α_k with $Beta(4, 1)$ and β_k with $Beta(1, 8) * (1 - \alpha_k)$ for each regime. Gamma and Beta distributions are used to have parameters between (0, 1). θ_m is reparameterized as $\log(\theta_m)$.

This results with probabilities close to 1 which means a higher probability for not changing the states and not producing too many regimes. At the time where the transition probability is low enough to move to next state, we create a changepoint location and create the changepoint time series with a 500 time step. We define regimes as the series between two changepoint locations and for each regime the volatility model parameters are selected to be low if it is odd, high if the regime number is even in order to see the fluctuations in the synthetic time series. We then calculate the volatility using GARCH model with Gaussian distribution. Changepoint locations, volatility and the generated return series are plotted in Figure 6.1 below. The blue dashed vertical lines state the exact changepoint locations which allow us to easily identify the state switching regimes. We tried several alternatives regarding the parameters to see the performance of the algorithm. In the simulated data set several delta parameters are evaluated such as 0.25, 0.5, 0.75, 0.9 and 0.99 and selected as the best to be used in this dissertation. We had experiments with a 500 time step synthetic data series with different particle sizes of 500, 1000 and 5000.

Figure 6.1 presents the real changepoint locations for five different states and regimes in blue and corresponding filtered changepoints and estimated regimes in red. The second and third subfigures shows the generated log returns of the time series and GARCH based volatility respectively with blue dashed vertical lines stating the true changepoint locations in the corresponding series.

The results and estimates of the proposed APF based MCD algorithm are reported in Figure 6.2. The top left and right figures give the real log return series and corresponding filtered changepoint locations using the GARCH based volatility model. Here in the top right figure we will easily identify that the algorithm is able to estimate the changepoint locations very closely to real regime switching points. The middle part presents the filtered volatility estimates along with the true volatilities. We also plot the sequence of the effective sample size (ESS) of particles to check weight degeneracy. ESS of particles is equal to N when all weights are equal to $1/N$.

Figure 6.3 presents the filtered parameters and their %2.5 and %97.5 quantiles

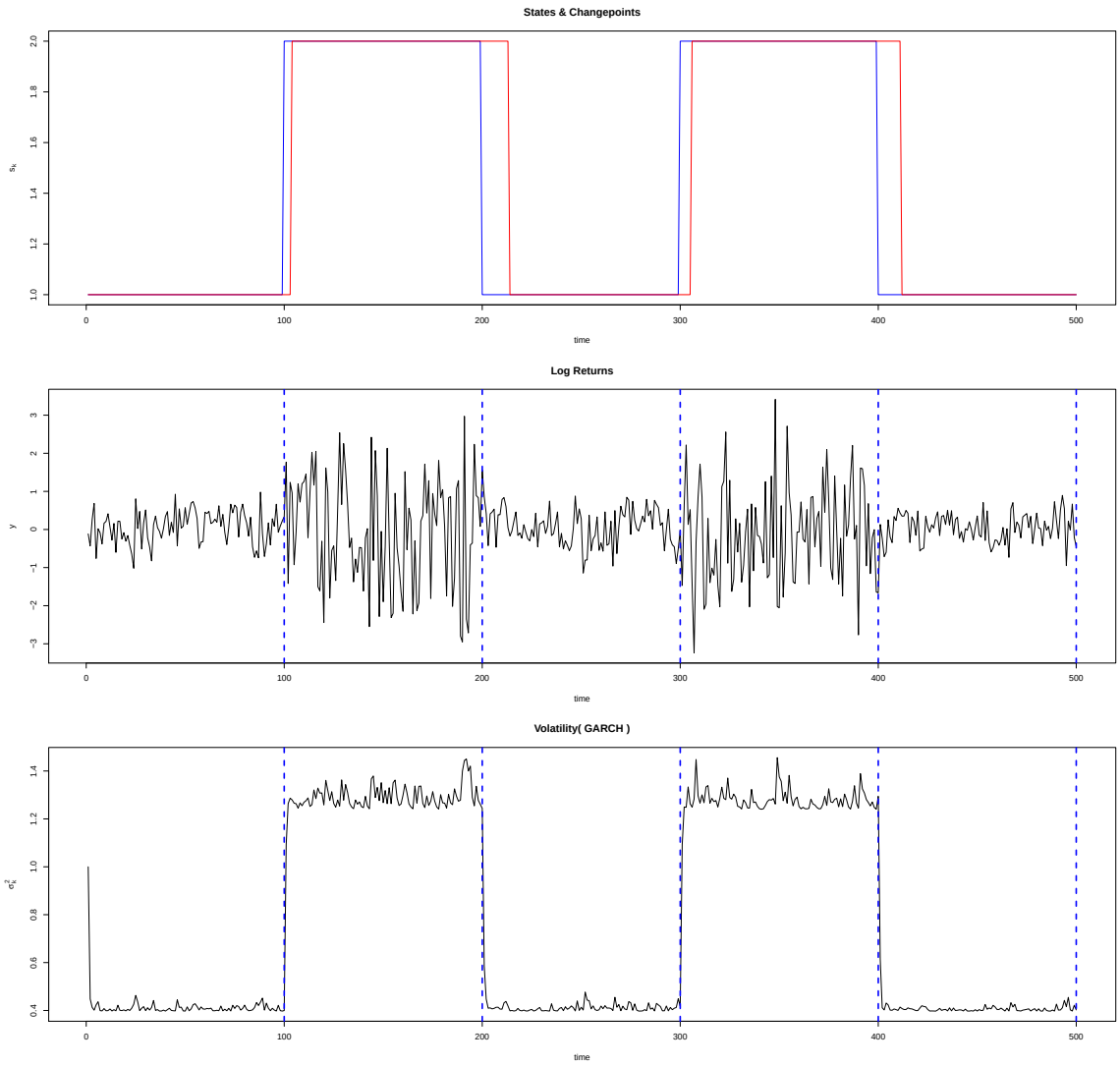


Figure 6.1. States & Filtered Changepoints, Log Returns and Volatility respectively.

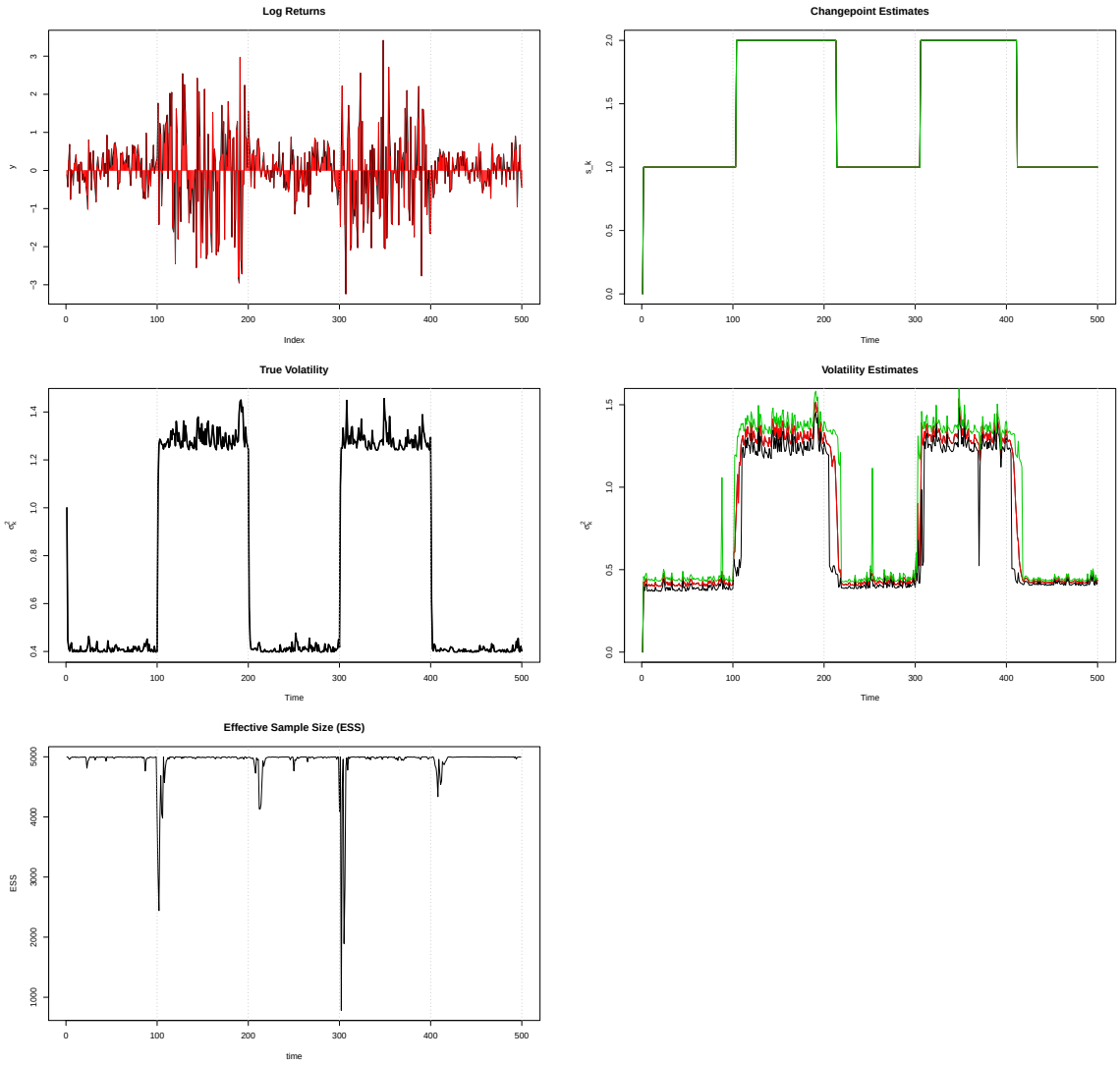


Figure 6.2. GARCH: Top Left: Log Returns, Top Right: Changepoint Estimates, Center Left: True Volatility, Center Right: Volatility Estimates, Bottom Left: Effective Sample Size(ESS).

in red and green and their mean values as black lines respectively for each parameter $\omega_k, \alpha_k, \beta_k$ in each regime. The dotted vertical lines identifies the exact changepoint locations and presents the regime identifiers. The filtered parameter estimates shows that the proposed approach estimates the parameters correctly in each regime and converges to very close values to real prior parameters which are shown as blue vertical lines in the plots.

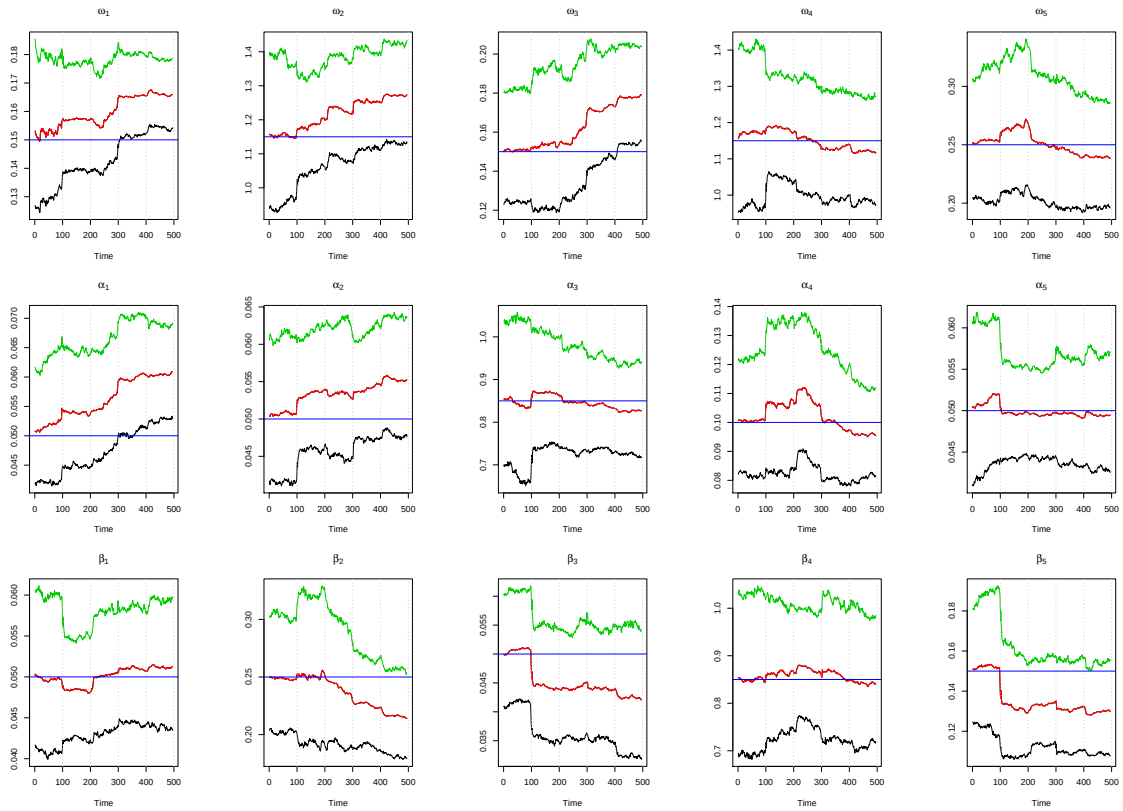


Figure 6.3. Filtered parameters in each regime $(\omega_k, \alpha_k, \beta_k, \gamma_k)$ and their %2.5 and %97.5 quantiles.

6.1.2. Experiments on EGARCH Synthetic Data Series

We generated the synthetic time series data using an EGARCH model with Gaussian distribution and full parameter changepoint modeling. This time series includes switching regimes by providing different ω_k, α_k, β_k and γ_k parameters at changepoint locations. Synthetic time series is generated using a predefined left-right HMM model. The parameter configuration of the algorithm includes the maximum number of the

changepoints, initial parameters for each regime and volatility model parameter sets. First the maximum limit for the number of changepoints (the number of regimes) is defined. This has not to be the number of the exact number of changepoints but will be defined as the maximum number of changepoint limit that the algorithm can reach. Four different regime parameters are defined for each regime, $\theta_m = [\omega_k, \alpha_k, \beta_k, \gamma_k]$. The upper bound for the number of states is defined as $M = 5$. State transition probabilities are defined as P. The prior parameters for each regime are randomly generated as $\omega_k \sim Gamma(1, 0.5)$, α_k with $Beta(4, 1)$, β_k with $Beta(1, 8) * (1 - \alpha_k)$ and γ_k with $Beta(1, 8)$ for each regime. Gamma and Beta distributions are used to have parameters between (0, 1). θ_m is reparameterized as $\log(\theta_m)$.

This results with probabilities close to 1 which means a higher probability for not changing the states and not producing too many regimes. At the time where the transition probability is low enough to move to next state, we create a changepoint location and create the changepoint time series with a 500 time step. We define regimes as the series between two changepoint locations and for each regime the volatility model parameters are selected to be low if it is odd, high if the regime number is even in order to see the fluctuations in the synthetic time series. We then calculate the volatility using EGARCH model with Gaussian distribution. Changepoint locations, volatility and the generated return series are plotted in Figure 6.4 below. The blue dashed vertical lines state the exact changepoint locations which allow us to easily identify the state switching regimes. We tried several alternatives regarding the parameters to see the performance of the algorithm. In the simulated data set several delta parameters are evaluated such as 0.25, 0.5, 0.75, 0.9 and 0.99 and selected as the best to be used in this paper. We had experiments with a 500 time step synthetic data series with different particle sizes of 500, 1000 and 5000.

Figure 6.4 presents the real changepoint locations for five different states and regimes in blue and corresponding filtered changepoints and estimated regimes in red. The second and third subfigures shows the generated log returns of the time series and EGARCH based volatility respectively with blue dashed vertical lines stating the true changepoint locations in the corresponding series.

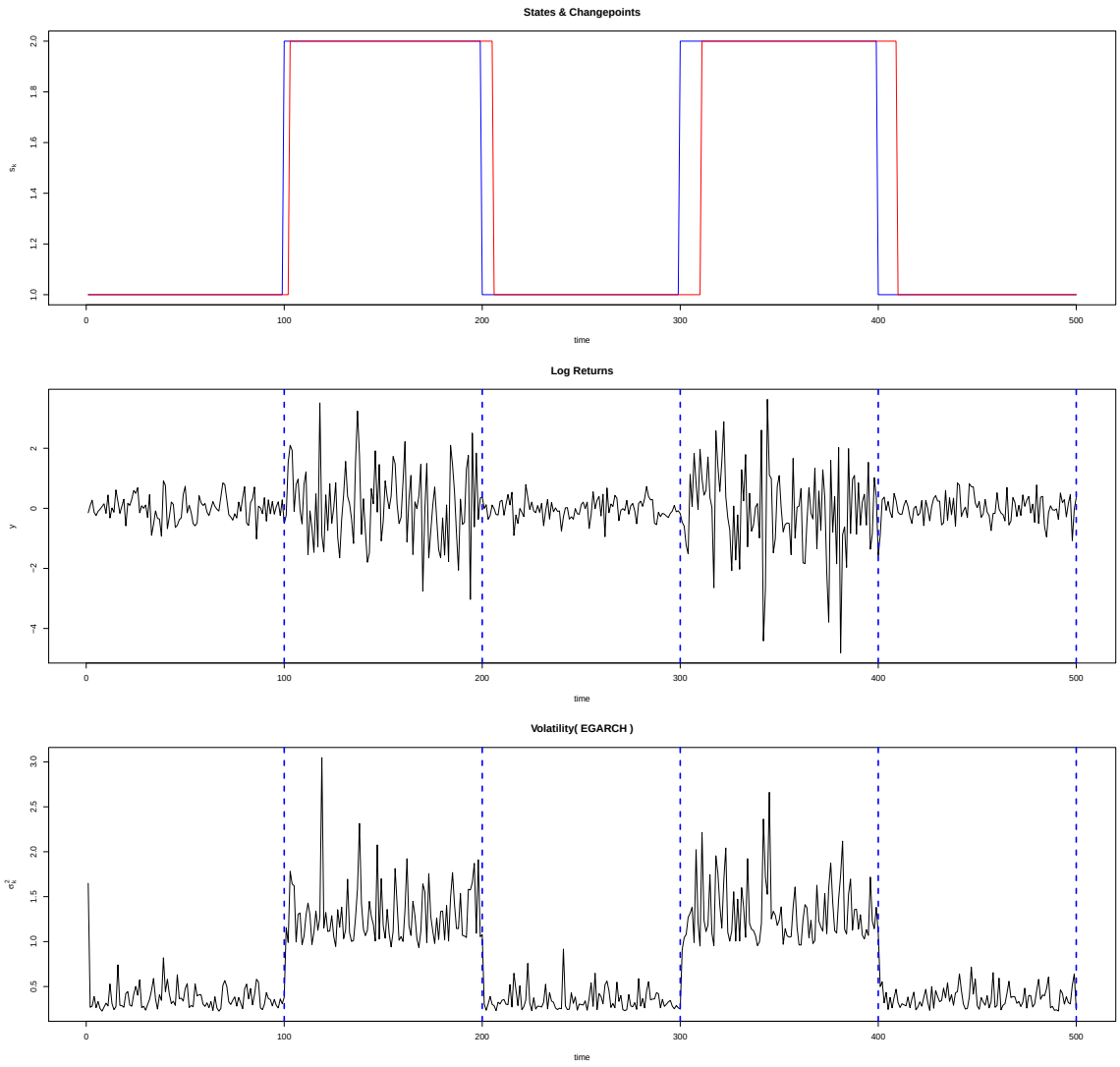


Figure 6.4. States & Filtered Changepoints, Log Returns and Volatility respectively.

We reported the results and estimates of the proposed APF based MCD algorithm in Figure 6.5. The top left and right figures give the real log return series and corresponding filtered changepoint locations using the volatility model. Here in the top right figure we will easily identify that the algorithm is able to estimate the changepoint locations very closely to real regime switching points. The middle part presents the filtered volatility estimates along with the true volatilities. We also plot the sequence of the effective sample size (ESS) of particles to check weight degeneracy. ESS of particles is equal to N when all weights are equal to 1/N. In the plot there exists a large fluctuation around regime switching points and outliers and they return to normal quickly after the outlier or the regime switching point.

Figure 6.6 presents the filtered parameters and their %2,5 and %97,5 quantiles in red and green and their mean values as black lines respectively for each parameter $\omega_k, \alpha_k, \beta_k, \gamma_k$ in each regime. The dotted vertical lines identifies the exact changepoint locations and presents the regime identifiers. The filtered parameter estimates shows that the proposed approach estimates the parameters correctly in each regime and converges to very close values to real prior parameters which are shown as blue vertical lines in the plots.

We have shown that APF based MCD algorithm estimates the changepoint locations and the parameters of the volatility model well using synthetic time series. In this section, we also compared our APF based technique with He and Maheu's model (MH) using both GARCH & EGARCH simulations and presented the mean square errors (MSE) of volatility and regime parameter estimates after 100 simulations. For GARCH simulations we have used exactly the same configuration as EGARCH based simulations and compared the results accordingly.

Figure 6.7 box plots of the MSE of the estimated volatility process compared to the real volatility process for both techniques. The left plot presents the comparison results for EGARCH simulations and the right plot shows the MSE comparison results for GARCH based simulations for APF and MH models respectively.

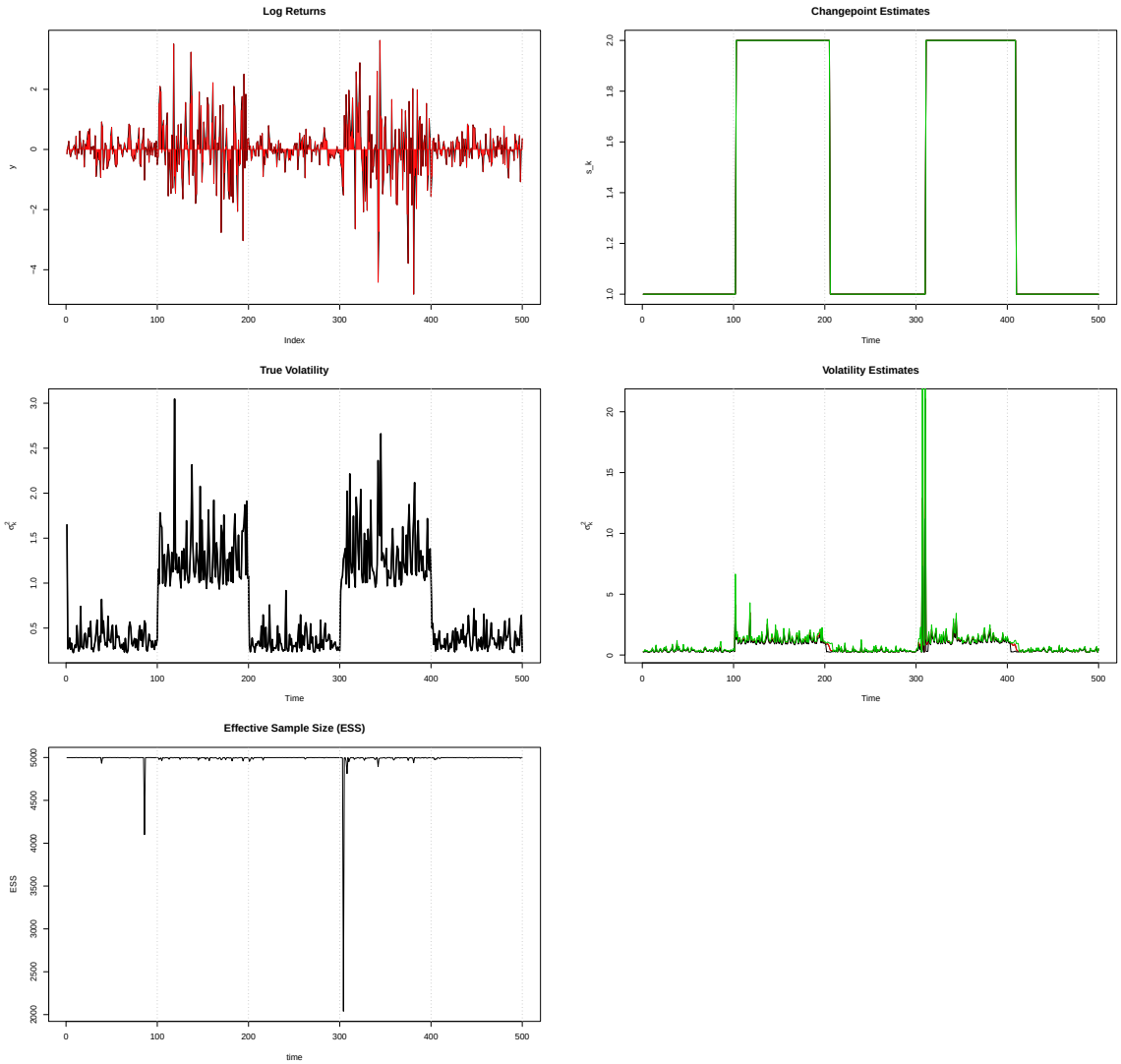


Figure 6.5. EGARCH: Top Left: Log Returns, Top Right: Changepoint Estimates, Center Left: True Volatility, Center Right: Volatility Estimates, Bottom Left: Effective Sample Size(ESS).

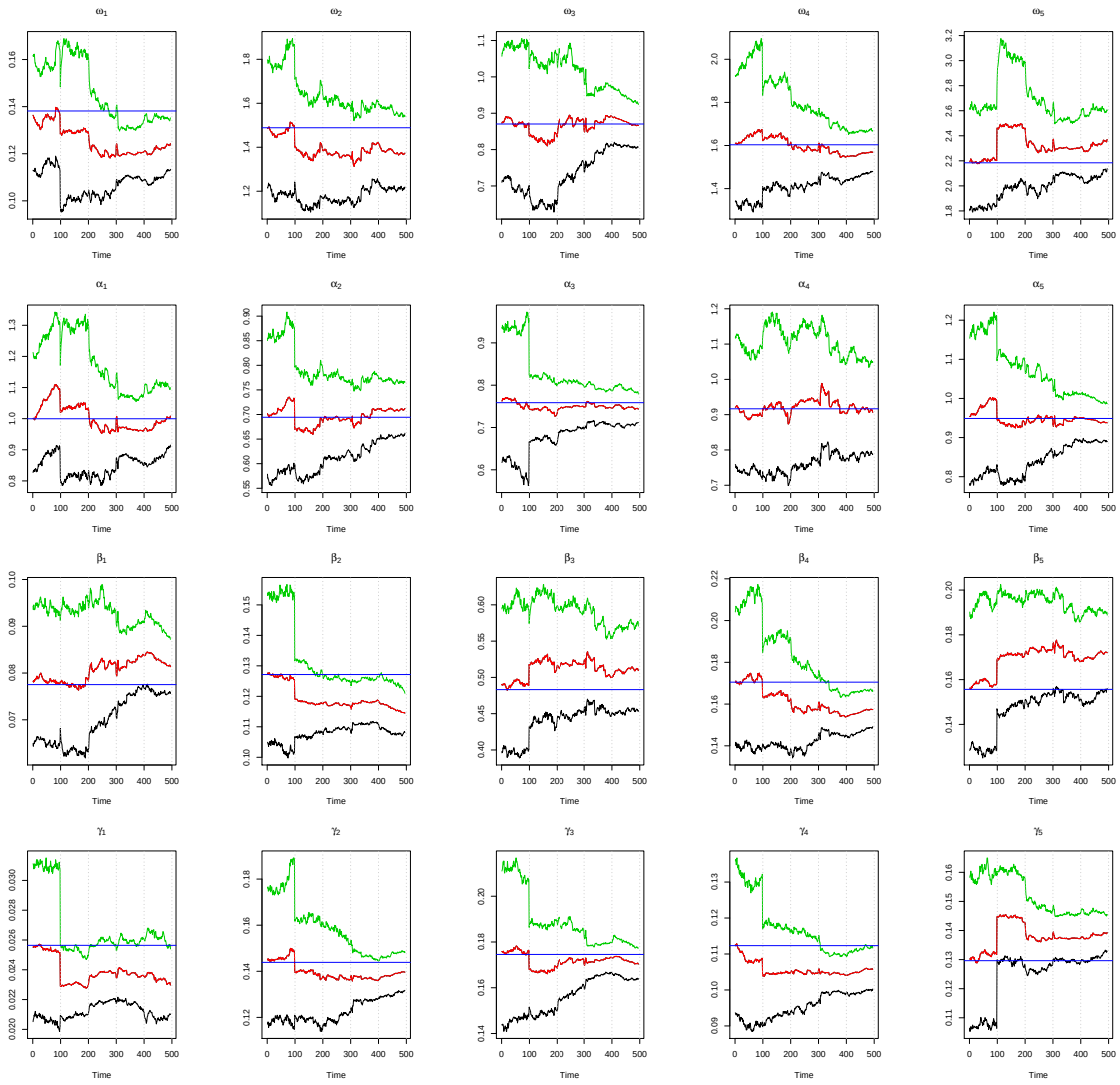


Figure 6.6. Filtered parameters in each regime $(\omega_k, \alpha_k, \beta_k, \gamma_k)$ and their %2.5 and %97.5 quantiles.

We also box plot MSE of the estimated parameters $\omega_k, \alpha_k, \beta_k, \gamma_k$ in Figures 6.8, 6.9, 6.10 and 6.11 for EGARCH & GARCH simulations respectively and have shown that APF shows better estimation power than MH technique for both models in volatility and parameter estimates.

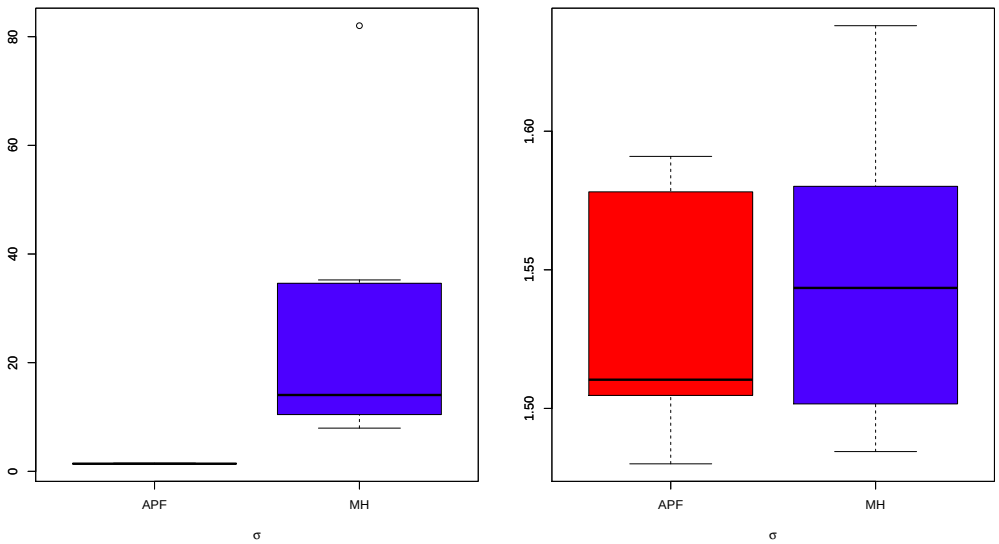


Figure 6.7. Comparison of Volatility Estimates in APF vs MH Models using EGARCH & GARCH simulations respectively.

6.2. Experiments on Real Financial Time Series (BIST 100)

In this section, we tested the APF based MCD algorithm with BIST 100 index financial time series data. Figure 6.12 shows the USD dollar based close prices of BIST 100 for the selected time range and its corresponding continuously compounding log returns with the estimated changepoints on top. A particle set of 5000 has been selected and initialization parameters are used as described in the synthetic time series data.

Figure 6.12 shows the log returns of the BIST100 series at the bottom, USD based price series in the middle and corresponding filtered changepoint locations on the top plot. Here we have identified three sustaining regimes one in 2010, the second one starting in the last quarter of 2010 (Ireland Debt Crisis) and continues for a one

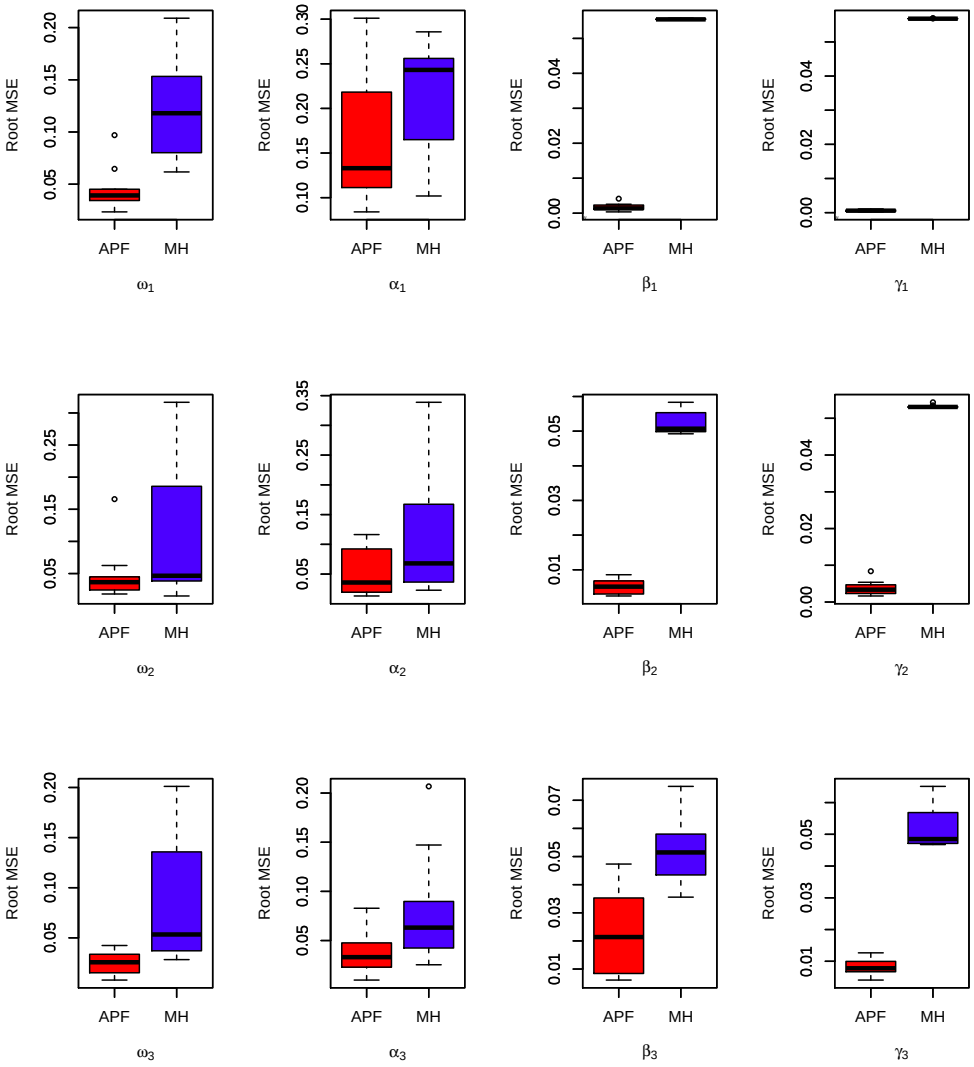


Figure 6.8. Comparison of model parameters in different regimes (EGARCH simulations).

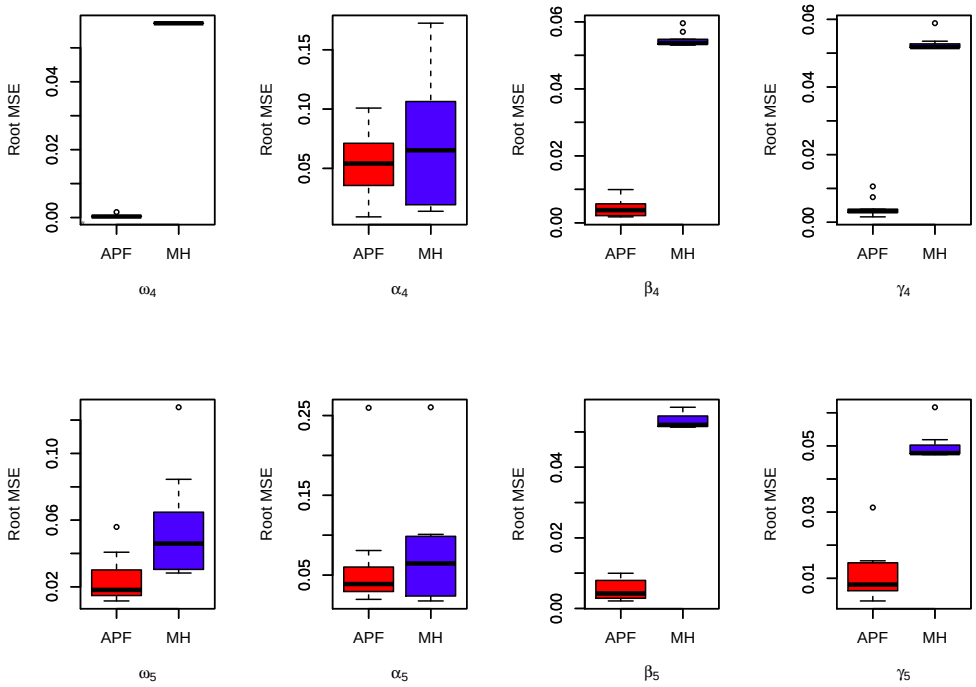


Figure 6.9. Comparison of model parameters in different regimes (EGARCH simulations).

and half year and the third one in the second half of 2012 up to early 2013 and a peak and change in the first quarter of 2013 (Moody's unexpected note for Turkey).

We have seen that parameter selection, sampling technique and APF approaches are all critical in the success of the proposed approach. Estimated parameters of the volatility model in each regime and their quantiles are given in Figure 6.13. $\theta_m = [\omega_k, \alpha_k, \beta_k, \gamma_k]$ parameter's evolution is given in time. This also shows the changes of each parameter among each regime throughout the time. α_k parameter is the driving parameter and shows stability besides the last regime. And also the counter part β_k parameter which controls the historical volatility effect shows stability throughout the regimes on the timeline. But this is not the case in the constant parameter (ω_k).

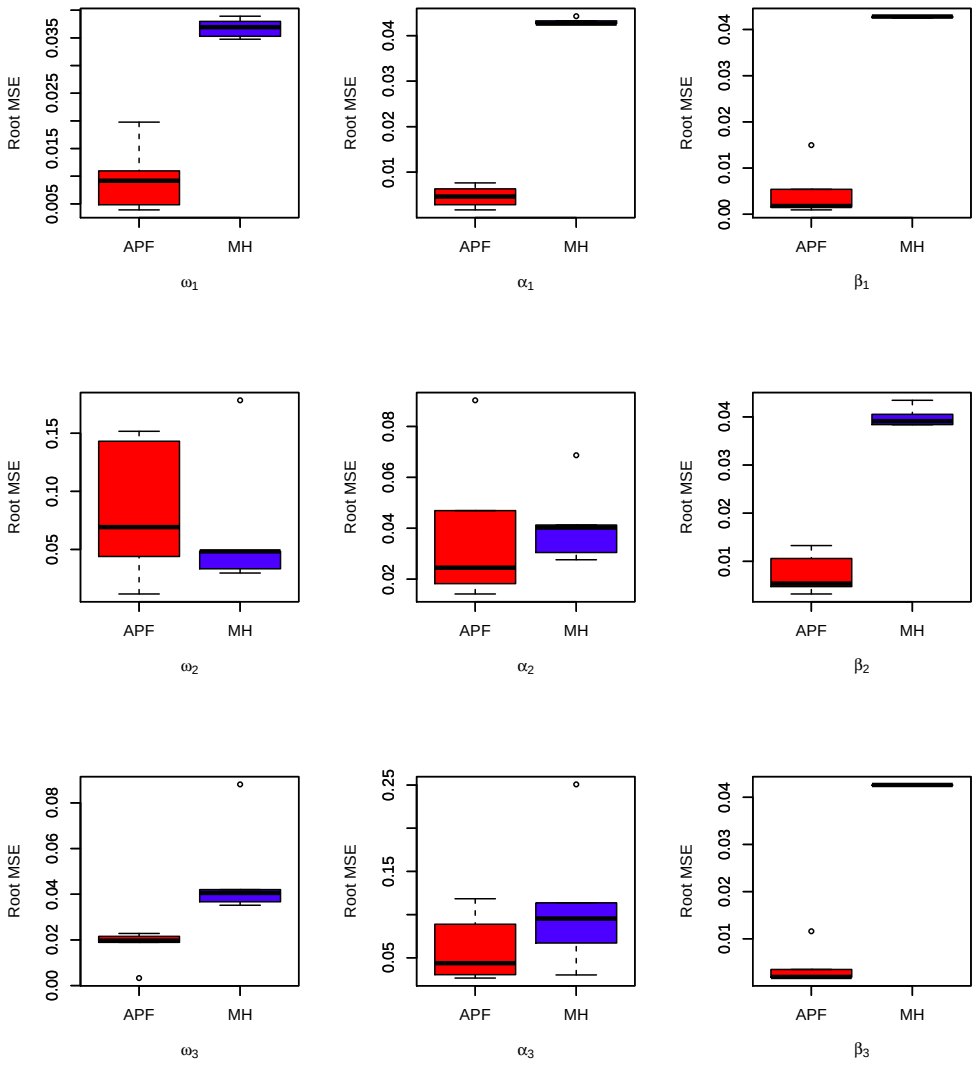


Figure 6.10. Comparison of model parameters in different regimes (GARCH simulations).

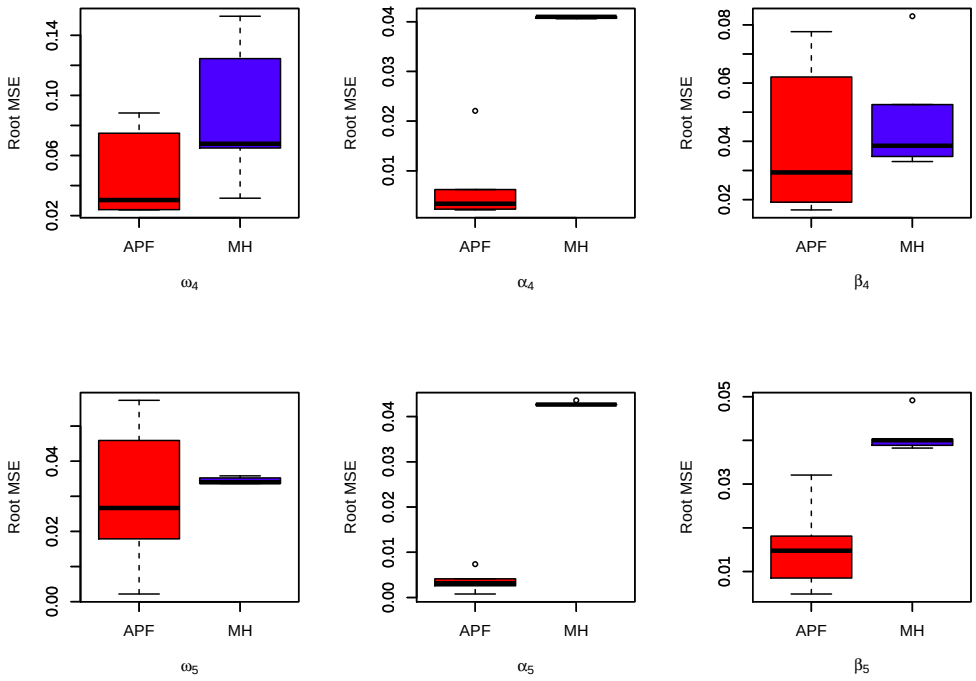


Figure 6.11. Comparison of model parameters in different regimes (GARCH simulations).

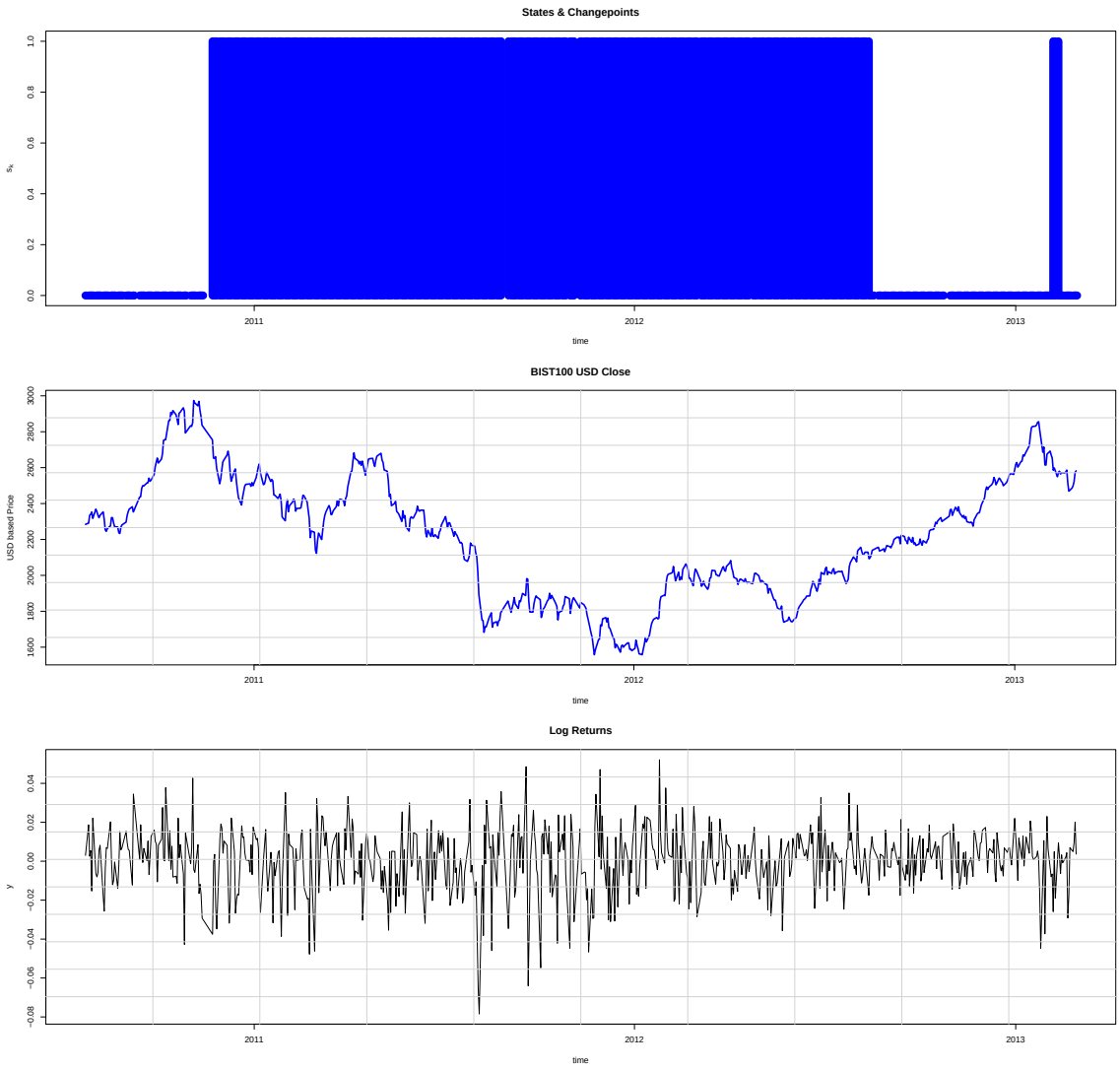


Figure 6.12. BIST100 Changepoint Estimates.

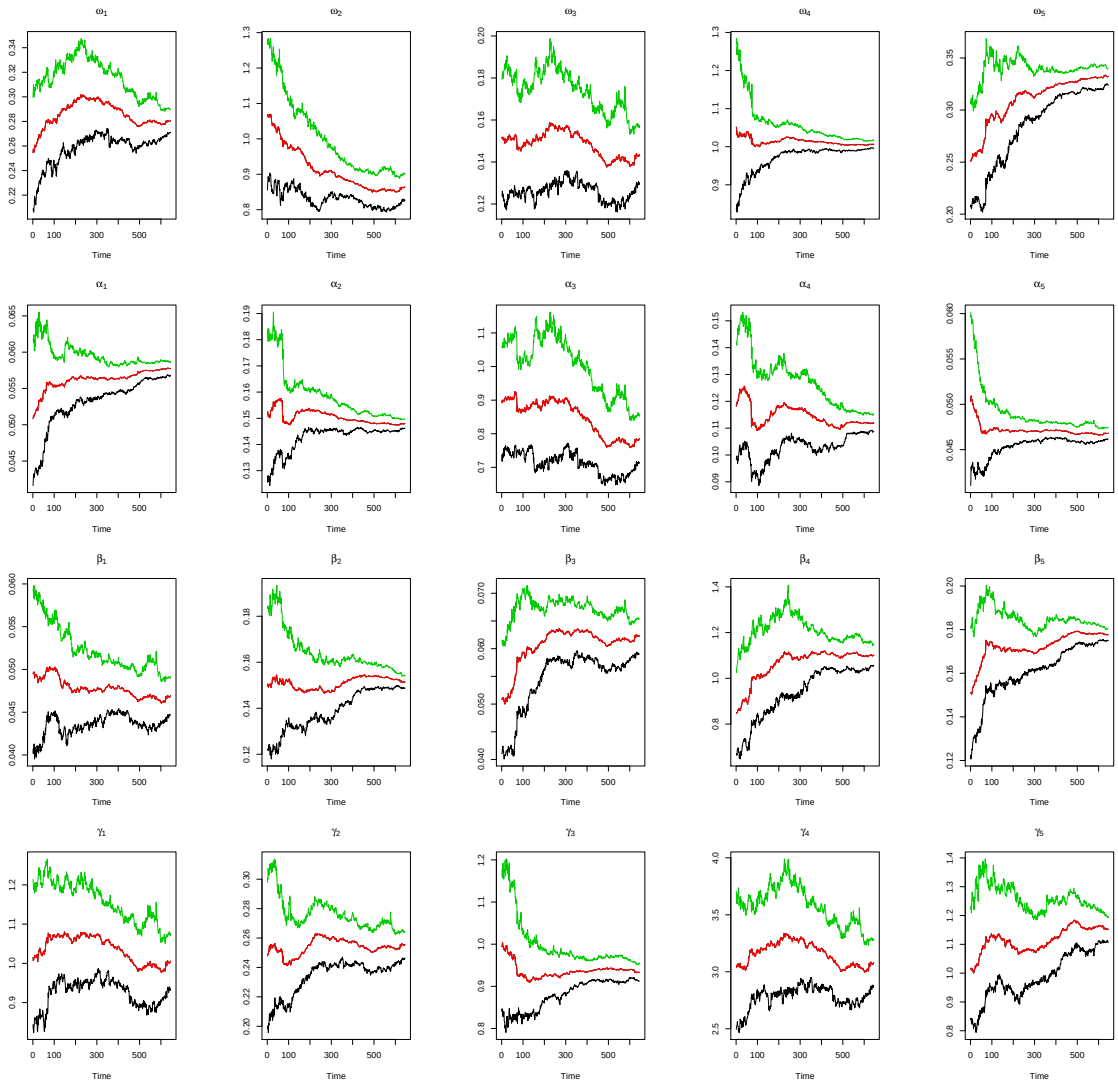


Figure 6.13. BIST100 Filtered Parameters.

7. CONCLUSION & DISCUSSIONS

In this dissertation we tried to address the challenges in estimating the unknown number of changepoints in GARCH and EGARCH based volatility models using Bayesian Inference techniques and Sequential Monte Carlo (SMC) approaches. The proposed approach is a SMC method for estimating GARCH and EGARCH based volatility models which are subject to an unknown number of changepoints. In real-world we face with non-linear and non-Gaussian structures especially in financial time series prediction problems. Generally, in non-linear and non-Gaussian models, computational complexity is high and exact inference is intractable. Recent Auxiliary Particle Filtering (APF) techniques are used to calculate the posterior densities and forecasts in real-time for non-linear problems. For this purpose we propose APF based MCD in GARCH and EGARCH models to accurately predict the occurrence of regime switches in GARCH and EGARCH based processes.

This dissertation proposes combined multiple changepoint detection and time-varying parameter learning in regime switching state space models. The proposed solution is first tested over synthetic dataset and then validated using an emerging market data of Istanbul Stock Exchange (BIST) where volatility shows more fluctuations than US markets. In synthetic data, we randomly generated regime switching locations and created several regimes using randomly generated GARCH & EGARCH parameters. The proposed algorithm was able to find structural changepoint locations and much more regarding the evolution of the volatility parameters. We also compared the results of the proposed APF based approach with He and Maheu's model and have shown that our approach provides better results. After this validation step, the proposed APF based MCD approach is tested on BIST 100 index time series and provided both state and parameter estimation online and shown encouraging results for understanding the switching regimes and the model behind for prediction purposes. We used both Gaussian and Student-t distribution in this dissertation which is more suitable for financial time series data and volatility models.

The originality of this dissertation lies on the proposed approach that combines APF based SMC methods and Kernel Smoothing based parameter learning to provide combined state and piecewise time-varying parameter learning in regime switching volatility models using multiple changepoint detection techniques. The main contributions are:

- Multiple regime switching state space models instead of fitting a global and single GARCH & EGARCH based model to the time series
- Combined multiple changepoint detection and time-varying parameter learning in regime switching state space models
- Learning without knowing the exact number of the changepoint locations
- Online estimation approach by combining APF based SMC methods and Kernel Smoothing based parameter learning to provide combined state and parameter learning
- Applied extensive simulations using GARCH & EGARCH based models, Student-t distribution over synthetic dataset and validated using an emerging market data of Borsa Istanbul (BIST)

Based on several experiments it is also seen that this algorithm is highly sensitive to the given prior parameters and needs optimization in terms of kernel smoothing parameters. With these features, the proposed approach has advantages over traditional and single model approaches. This dissertation provides a different and multi-regime auxiliary particle filtering based piecewise time varying parameter learning approach in regime switching volatility models which can also be applied to several signal processing problems besides time series prediction.

REFERENCES

1. Anderson, D. and G. McNeill, "Artificial Neural Networks Technology", *Kaman Sciences Corporation*, Vol. 258, pp. 13502–462, 1992.

2. Barber, D., A. T. Cemgil and S. Chiappa, *Bayesian Time Series Models*, Cambridge University Press, 2011.

3. Adams, R. P. and D. J. MacKay, "Bayesian Online Changepoint Detection", Cambridge, UK, 2007.

4. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer New York, 2006.

5. Abu-Mostafa, Y. S. and A. F. Atiya, "Introduction to Financial Forecasting", *Appl. Intell.*, Vol. 6, No. 3, pp. 205–213, 1996.

6. Bollerslev, T., "Generalized Autoregressive Conditional Heteroskedasticity", *Journal of Econometrics*, Vol. 31, pp. 307–327, 1986.

7. Yümlü, M. S., F. S. Gürgen and N. Okay, "A Comparison of Global, Recurrent and Smoothed-piecewise Neural Models for Istanbul Stock Exchange (ISE) Prediction", *Pattern Recognition Letters*, Vol. 26, No. 13, pp. 2093–2103, 2005.

8. Yümlü, M. S. and F. S. Gürgen, "SVR for Time Series Prediction", B. H. Boyle (Editor), *Support Vector Machines: Data Analysis, Machine Learning and Applications*, pp. 117–130, Nova Science Publishers, Commack, NY, USA, 2011.

9. Yümlü, M. S., F. S. Gürgen, A. T. Cemgil and N. Okay, "Auxiliary Particle Filtering based Multiple Changepoint Detection in Volatility Models.", *Proceedings of the IASTED International Conference Artificial Intelligence and Applications (AIA 2013)*, Vol. 793, 2013.

10. Van Der Merwe, R., *Sigma-point Kalman Filters for Probabilistic Inference in Dynamic State-space Models*, Ph.D. Thesis, Oregon Health & Science University, 2004, aAI3129163.

11. Matsumoto, T., "Online Bayesian Modeling and Prediction of Nonlinear Systems–Sequential Monte Carlo Approach.", *Methods Information in Medicine*, Vol. 46, pp. 96–101, 2007.

12. Chib, S., "Estimation and Comparison of Multiple Change-point Models", *Journal of Econometrics*, Vol. 86, No. 2, pp. 221–241, 1998.

13. Chopin, N., "Dynamic Detection of Change Points in Long Time Series", *Annals of the Institute of Statistical Mathematics*, Vol. 59, No. 2, pp. 349–366, 2007.

14. He, Z. and J. M. Maheu, "Real Time Detection of Structural Breaks in GARCH Models.", *Computational Statistics & Data Analysis*, Vol. 54, No. 11, pp. 2628–2640, 2010.

15. Turner, R., Y. Saatci and C. E. Rasmussen, "Adaptive Sequential Bayesian Change Point Detection", *Advances in Neural Information Processing Systems (NIPS): Temporal Segmentation Workshop*, 2009.

16. Lopes, H. F. and R. S. Tsay, "Particle Filters and Bayesian Inference in Financial Econometrics", *Journal of Forecasting*, Vol. 30, No. 1, pp. 168–209, 2011.

17. Eckley, I., P. Fearnhead and R. Killick, "Analysis of Changepoint Models.", D. Barber, A. Cemgil and S. Chiappa (Editors), *Bayesian Time Series Models*, Cambridge University Press, Cambridge, UK, 2011.

18. Whiteley, N. and A. M. Johansen., "Auxiliary Particle Filtering: Recent Developments", D. Barber, A. Cemgil and S. Chiappa (Editors), *Bayesian Time Series Models*, pp. 52–81, Cambridge University Press, Cambridge, UK, 2011.

19. Paquet, U., *Empirical Bayesian Change Point Detection*, Tech. rep., University of Stellenbosch, 2006.

20. Cemgil, A. T., H. J. Kappen and D. Barber, "A Generative Model for Music Transcription", *Audio, Speech, and Language Processing, IEEE Transactions on*, Vol. 14, No. 2, pp. 679–694, 2006.

21. Zoeter, O. and T. Heskes, "Change Point Problems in Linear Dynamical Systems.", *Journal of Machine Learning Research*, Vol. 6, pp. 1999–2026, 2005.

22. Fearnhead, P., "Exact and Efficient Bayesian Inference for Multiple Changepoint Problems", *Statistical Computing*, Vol. 16, pp. 203–213, 2006.

23. Yümlü, M. S., F. Gürgen, A. T. Cemgil and N. Okay, "Auxiliary Particle Filtering based Multiple Changepoint Detection in Volatility Models", *Artificial Intelligence and Applications*, Vol. 793, 2013.

24. Yümlü, M. S., F. S. Gürgen, A. T. Cemgil and N. Okay, "Bayesian Changepoint and Time-varying Parameter Learning in Regime Switching Volatility Models", *Digital Signal Processing*, 2015.

25. Okay, N., "Asymmetric Volatility Dynamics: Evidence from the Istanbul Stock Exchange", *Business & Economics for the 21st Century*, Vol. 2, pp. 207–216, 1998.

26. Hassan, R., *Hybrid HMM and Soft Computing Modeling with Applications to Time Series Analysis*, Ph.D. Thesis, University of Melbourne, 2007.

27. Yumlu, M. S., F. S. Gurgen and N. Okay, "Financial Time Series Prediction Using Mixture of Experts", *Computer and Information Sciences-ISCIS 2003*, pp. 553–560, Springer Berlin Heidelberg, 2003.

28. Hellström, T. and K. Holmström, *Predicting the Stock Market*, Tech. rep., Mälardalen University, Sweden, 1998.

29. Fama, E. F., "Efficient Capital Markets: A Review of Theory and Empirical Work", *The Journal of Finance*, Vol. 25, No. 2, pp. 383–417, 1970.

30. Box, G. E. P. and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, Holden-Day, San Francisco, 1976.

31. Murphy, J. J., *Technical Analysis of the Future's Markets*, Prentice Hall Press, New York, 1986.

32. Freisleben, B., "Stock Market Prediction with Backpropagation Networks", *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pp. 451–460, Springer Berlin Heidelberg, 1992.

33. Mitchell, T. M., *Machine Learning. WCB*, Vol. 8, McGraw-Hill Boston, MA:, 1997.

34. Brooks, C., "Linear and Non-linear (Non-) Forecastability of High-frequency Exchange Rates", *Journal of Forecasting*, Vol. 16, No. 2, pp. 125–145, 1997.

35. Brooks, C. and G. Persand, "Value-at-risk and Market Crashes", *Journal of Risk*, Vol. 2, pp. 5–26, 2000.

36. Brooks, C. and G. Persand, "Volatility Forecasting for Risk Management", *Journal of Forecasting*, Vol. 22, No. 1, pp. 1–22, 2003.

37. Nelson, D. B., "Conditional Heteroskedasticity in Asset Returns: A New Approach", *Econometrica*, Vol. 59, No. 2, pp. 347–370, 1991.

38. Mazıbaş, M., "İMKB Piyasalarındaki Volatilitenin Modellenmesi ve Öngörülmesi: Asimetrik GARCH Modelleri ile Bir Uygulama", *VII. Ekonometri ve İstatistik Sempozyumu, İstanbul Üniversitesi İktisat Fakültesi Ekonometri Bölümü, İstanbul*, Vol. 28, p. 2008, 2005.

39. Yümlü, M. S., *Financial Time Series Forecasting using Artificial Neural Networks*, Master's Thesis, Bogazici University, 2004.

40. Hutchinson, J. M., A. W. Lo and T. Poggio, "A Nonparametric Approach to Pricing and Hedging Derivative Securities via Learning Networks", *The Journal of Finance*, Vol. 49, No. 3, pp. 851–889, 1994.

41. Zhang, G. and M. Y. Hu, "Neural Network Forecasting of the British Pound/US Dollar Exchange Rate", *Omega*, Vol. 26, No. 4, pp. 495–506, 1998.

42. Bishop, C. M., *Neural Networks for Pattern Recognition*, Clarendon Press Oxford, 1995.

43. JingTao, Y. and C. L. Tan, "Guidelines for Financial Forecasting with Neural Networks", *International Conference of Neural Information Processing*, Citeseer, 2001.

44. Tsai, C. and S. Wang, "Stock Price Forecasting by Hybrid Machine Learning Techniques", *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 1, p. 60, 2009.

45. Yülü, S., F. S. Gürgen and N. Okay, "Global, Recurrent and Smoothed-Piecewise Neural Models for Financial Time Series Prediction.", *Artificial Intelligence and Applications*, pp. 134–139, 2005.

46. Yumlu, M. S., F. S. Gurgen and N. Okay, "Turkish Stock Market Analysis Using Mixture of Experts", *Proceedings of Engineering of Intelligent Systems (EIS), Madeira*, 2004.

47. Poggio, T. and F. Girosi, "Regularization Algorithms for Learning that are Equivalent to Multilayer Networks", *Science*, Vol. 247, No. 4945, pp. 978–982, 1990.

48. Weigend, A. S., M. Mangeas and A. N. Srivastava, "Nonlinear Gated Experts for Time Series: Discovering Regimes and Avoiding Overfitting", *International Journal of Neural Systems*, Vol. 6, No. 04, pp. 373–399, 1995.

49. Rumelhart, D. E., G. E. Hinton and R. J. Williams, "Learning Internal Representations by Error Propagation", D. E. Rumelhart, J. L. McClelland and the PDP research group. (Editors), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, MIT Press, 1986.

50. White, H., "Parametric Statistical Estimation with Artificial Neural Networks", *University of California at San Diego, Economics Working Paper Series*, 1992.

51. Alpaydin, E., *Introduction to Machine Learning*, The MIT Press, 2004.

52. Powell, M. J., "Radial Basis Functions for Multivariable Interpolation: A Review", *Algorithms for Approximation*, pp. 143–167, Clarendon Press, 1987.

53. Weigend, A. S., B. A. Huberman and D. E. Rumelhart, "Predicting the Future: A Connectionist Approach", *International Journal of Neural Systems*, Vol. 1, No. 03, pp. 193–209, 1990.

54. Elman, J. L., "Finding Structure in Time", *Cognitive Science*, Vol. 14, No. 2, pp. 179–211, 1990.

55. Jordan, M. I., "Attractor Dynamics and Parallelism in a Connectionist Sequential Machine", *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp. 531–546, Hillsdale, NJ: Erlbaum, 1986.

56. Vapnik, V. N., *The Nature of Statistical Learning Theory*, Springer, Berlin, 2nd edn., 2000.

57. Vapnik, V. N. and V. Vapnik, *Statistical Learning Theory*, Vol. 1, Wiley New York, 1998.

58. Burges, C. J., "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, pp. 121–167, 1998.

59. Müller, K.-R., A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen and V. Vap-

nik, "Predicting Time Series with Support Vector Machines", *Artificial Neural Networks—ICANN'97*, pp. 999–1004, Springer, 1997.

60. Platt, J., "Fast Training of Support Vector Machines using Sequential Minimal Optimization", *Advances in Kernel methods—support vector learning*, Vol. 3, 1999.

61. Van Gestel, T., J. A. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor and J. Vandewalle, "Financial Time Series Prediction using Least Squares Support Vector Machines within the Evidence Framework", *Neural Networks, IEEE Transactions on*, Vol. 12, No. 4, pp. 809–821, 2001.

62. Cao, L., "Support Vector Machines Experts for Time Series Forecasting", *Neurocomputing*, Vol. 51, pp. 321–339, 2003.

63. Cao, L.-J. and F. E. H. Tay, "Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting", *Neural Networks, IEEE Transactions on*, Vol. 14, No. 6, pp. 1506–1518, 2003.

64. Tay, F. E. and L. Cao, "Modified Support Vector Machines in Financial Time Series Forecasting", *Neurocomputing*, Vol. 48, No. 1, pp. 847–861, 2002.

65. Kim, K.-j., "Financial Time Series Forecasting using Support Vector Machines", *Neurocomputing*, Vol. 55, No. 1, pp. 307–319, 2003.

66. Smola, A. J. and B. Schölkopf, "A Tutorial on Support Vector Regression", *Statistics and Computing*, Vol. 14, No. 3, pp. 199–222, 2004.

67. Yang, H., L. Chan and I. King, "Support Vector Machine Regression for Volatile Stock Market Prediction", *Intelligent Data Engineering and Automated Learning—IDEAL 2002*, pp. 391–396, Springer, 2002.

68. Yümlü, M. S. and F. S. Gürgen, "Istanbul Stock Exchange (ISE) 100 Prediction using Support Vector Predictors (SVP)", *Proceedings of Artificial Intelligence and*

Applications, Vol. 718, 2011.

69. Wang, X., P. A. Whigham and D. Deng, "Time-line Hidden Markov Experts and Its Application in Time Series Prediction", , 2003.

70. Chen, W.-H., J.-Y. Shih and S. Wu, "Comparison of Support-vector Machines and Back Propagation Neural Networks in Forecasting the Six Major Asian Stock Markets", *International Journal of Electronic Finance*, Vol. 1, No. 1, pp. 49–67, 2006.

71. Kayacan, E., B. Ulutas and O. Kaynak, "Grey System Theory-based Models in Time Series Prediction", *Expert Systems with Applications*, Vol. 37, No. 2, pp. 1784–1789, 2010.

72. Jacobs, R. A., M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive Mixtures of Local Experts", *Neural Computation*, Vol. 3, No. 1, 1991.

73. Engle, R. F., "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation", *Econometrica*, Vol. 50, pp. 987–1008, 1982.

74. Black, F. and M. Scholes, "The Pricing of Options and Corporate Liabilities", *Journal of Political Economy*, pp. 636–659, 1973.

75. Morgan, J., "Introduction to Riskmetrics", *New York: JP Morgan*, 1994.

76. French, K. R., G. W. Schwert and R. F. Stambaugh, "Expected Stock Returns and Volatility", *Journal of Financial Economics*, Vol. 19, No. 1, pp. 3–29, 1987.

77. Black, F., "Stuedies of Stock Price Volatility Changes", *Proceedings of the 1976 Meetings of the American Statistical Association, Business and Economics Section*, pp. 177–181, Chicago, 1976.

78. Pagan, A. R. and G. W. Schwert, "Alternative Models for Conditional Stock

Volatility", *Journal of Econometrics*, Vol. 45, No. 1, pp. 267–290, 1990.

79. Engle, R. F. and V. K. Ng, "Measuring and Testing the Impact of News on Volatility", *The Journal of Finance*, Vol. 48, No. 5, pp. 1749–1778, 1993.

80. David Barber, A. T. C., "Graphical Models for Time Series", *IEEE Signal Processing Magazine*, Vol. 26, 2010.

81. Simsekli, U., *Bayesian Methods for Real-time Pitch Tracking*, Master's Thesis, Bogazici University, Turkey, 2010.

82. Rabiner, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of the IEEE*, pp. 257–286, 1989.

83. Zhang, Y., *Prediction of Financial Time Series with Hidden Markov Models*, Master's Thesis, Simon Frasey University, 2004.

84. Bayramlı, B., *Financial Time Series Analysis for Hidden Markov Models*, Master's Thesis, Bogazici University, Turkey, 2008.

85. Satish Rao, J. H., *Analysis of Hidden Markov Models and Support Vector Machines in Financial Applications*, Tech. rep., University of California at Berkeley, 2010.

86. Murphy, K. P., *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. Thesis, University of California, Berkeley, 2002.

87. Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems", *Journal of Fluids Engineering*, Vol. 82, No. 1, pp. 35–45, 1960.

88. Rauch, H. E., C. Striebel and F. Tung, "Maximum Likelihood Estimates of Linear Dynamic Systems", *AIAA Journal*, Vol. 3, No. 8, pp. 1445–1450, 1965.

89. Gordon, N., D. Salmond and A. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation", *IEEE Proceedings F, Radar and Signal Pro-*

cessing, Vol. 140, No. 2, pp. 107–113, 1993.

90. Doucet, A., S. Godsill and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering", *Statistics and Computing*, Vol. 10, No. 3, pp. 197–208, 2000.

91. Doucet, A., N. De Freitas and N. Gordon (Editors), *Sequential Monte Carlo Methods in Practice*, 2001.

92. Doucet, A. and A. M. Johansen, "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later", *Handbook of Nonlinear Filtering*, Vol. 12, pp. 656–704, 2009.

93. Kanazawa, K., D. Koller and S. Russell, "Stochastic Simulation Algorithms for Dynamic Probabilistic Networks", *Proceedings of the Eleventh Annual Conference on Uncertainty in AI (UAI '95)*, pp. 346–351, 1995.

94. Isard, M. and A. Blake, "A Smoothing Filter for Condensation", H. Burkhardt and B. Neumann (Editors), *ECCV (1)*, Vol. 1406 of *Lecture Notes in Computer Science*, pp. 767–781, Springer, 1998.

95. Mikosch, T. and C. Starica, *Long Range Dependence Effects and ARCH Modeling*, Econometrics 0412004, EconWPA, 2004.

96. Pitt, M. K. and N. Shephard, "Filtering via Simulation: Auxiliary Particle Filters", *Journal of the American Statistical Association*, Vol. 94, No. 446, pp. 590–599, 1999.

97. Liu, J. and M. West, "Combined Parameter and State Estimation in Simulation-based Filtering", *Sequential Monte Carlo Methods in Practice*, pp. 197–223, Springer, 2001.