UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

EARLY DETECTION OF HIGH VOLATILITY CLUSTERS USING PARTICLE FILTERS

TESIS PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

KAREL BOGOMIR MUNDNICH BATIC

PROFESOR GUÍA:
MARCOS ORCHARD CONCHA

MIEMBROS DE LA COMISIÓN:
JORGE SILVA SÁNCHEZ
HÉCTOR AGUSTO ALEGRÍA

SANTIAGO DE CHILE
SEPTIEMBRE 2013

RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO
POR: KAREL BOGOMIR MUNDNICH BATIC
FECHA: SEPTIEMBRE 2013
PROF. GUÍA: MARCOS ORCHARD CONCHA

EARLY DETECTION OF HIGH VOLATILITY CLUSTERS USING PARTICLE FILTERS

El presente trabajo explora y analiza el uso de herramientas de procesamiento de señales que son comunes en áreas de Ingeniería Eléctrica y Pronóstico y Gestión de Salud en el análisis de series de tiempo financieras. El objetivo principal de este trabajo es detectar eventos de alto riesgo en una etapa temprana. De esta forma, el algoritmo propuesto emplea la fuerte relación entre volatilidad y riesgo y detecta *clusters* de alta volatilidad mediante el uso de la información obtenida de los procesos de estimación a través de Filtro de Partículas.

Para alcanzar el objetivo mencionado, se utiliza la representación de espacio-estado estocástica uGARCH para modelar la volatilidad de retornos compuestos continuamente. Dada la no-observabilidad de la volatilidad, se implementan dos esquemas de Filtro de Partículas para su estimación: los enfoques clásico y sensible al riesgo. Este último incluye el uso de una Distribución de Pareto Generalizada como propuesta para el funcional de riesgo (y distribución de importancia) para asegurar la asignación de partículas en regiones del espacio-estado que están asociadas a variaciones rápidas de volatilidad del sistema.

Para evaluar correctamente el rendimiento de las rutinas de filtrado, se han generado seis conjuntos de datos, donde ambos el estado y las mediciones son conocidas. Además, se ha realizado un análisis de sensibilidad sobre los seis conjuntos de datos, para así obtener los parámetros que permiten la mejor estimación de volatilidad. De estos resultados, se calculan valores promedios de parámetros que son luego utilizados en el esquema de detección.

La etapa de detección explora tres diferentes técnicas. Primero, se propone la utilización de un test de hipótesis entre las estimaciones *a priori* y *a posteriori* de las distribuciones de probabilidad del Filtro de Partículas Sensible al Riesgo. Segundo, se utiliza el Discriminante de Fisher para comparar las estimaciones *a posteriori* de las densidades entre el Filtro de Partículas Clásico y el Sensible al Riesgo. Finalmente, se utiliza la Divergencia de Kullback-Leibler de la misma forma que el Discriminante de Fisher. Los algoritmos propuestos son probados en los datos generados artificialmente y en datos de acciones de IBM.

Los resultados demuestran que el Filtro de Partículas Sensible al Riesgo propuesto supera la precisión del Filtro de Partículas en momentos de alzas no esperadas de volatilidad. Por otra parte, el test de hipótesis empleado en el proceso de filtrado sensible al riesgo detecta correctamente la mayoría de las alzas repentinas de volatilidad que conducen a la detección temprana de *clusters* de alta volatilidad. Finalmente, los algoritmos de detección propuestos basados en Discriminante de Fisher y Divergencia de Kullback-Leibler llevan a resultados donde la detección no es posible.

Summary

This work explores and analyzes the use of signal processing tools common in areas of Electrical Engineering and Prognostics and Health Management in financial time series analysis. The main objective of this work is to detect high risk events in an early stage. The proposed algorithm employs the strict relation between volatility and risk and detects high volatility clusters by the use of information obtained through a Particle Filter estimation process.

To achieve the aforementioned objective, the stochastic state-space uGARCH representation is used to model the volatility of continuously compounded returns. Given the non-observability of volatility, two Particle Filter frameworks are implemented for its estimation, which include classic and risk sensitive approaches. The latter includes the use of the Generalized Pareto Distribution as the proposed risk functional (and importance density distribution) to ensure the allocation of particles in regions of the state-space that are associated with sudden changes in the volatility of the system.

To correctly assess the performance of the filtering routines, six data sets have been generated, where both the state and the measurements are known. A sensibility analysis over these six data sets is performed in order to obtain the parameters that permit the best estimation of volatility. From these results, average parameters are calculated in order to use them in the detection scheme.

The detection stage explores three different techniques. First, a proposed hypothesis test between the prior and posterior probability density function estimates of the Risk Sensitive Particle Filtering process is used. Second, the Fisher Discriminant is used to compare the posterior probability density function estimates of the Particle Filter and Risk Sensitive Particle Filter. Finally, the Kullback-Leibler Divergence is used in the same sense as the Fisher Discriminant. The proposed algorithms are tested in the artificially generated data and in IBM stock market data.

Results demonstrate that the proposed Risk Sensitive Particle Filter slightly outperforms the Classic Particle Filter in terms of accuracy, in events where unexpected volatility rises occur. Moreover, the hypothesis test employed in the risk sensitive filtering process correctly detects most of the sudden volatility variations conducive to early detection of high volatility clusters. Finally, the proposed detection scheme based on the Fisher Discriminant and Kullback-Leibler Divergence is conducive to results where detection is not possible.

A mi madre.

Acknowledgements

Muchas personas han influido positivamente en mi vida. Gran parte de ello se ve plasmado en este trabajo, el cual representa la conclusión de un largo proceso de aprendizaje. Familia, profesores y amigos me han ayudado crecer y ser una mejor persona a lo largo de estos 25 años.

Quisiera agradecer a Eduardo Abeliuk, primer Ingeniero Eléctrico que conocí y quien me permitió comprender todas las posibilidades que la Ingeniería Eléctrica brinda. También debo mencionar a Thomas Capelle, por todas las conversaciones acerca de matemáticas y la carrera, y por demostrar con hechos que las cosas pueden ser distintas. Gracias también a César, Cristóbal y Alfonso, por ser los buenos amigos que son.

El grupo de amigos y compañerismo que nació a partir del Divino Circuito F.C. es uno de los excelentes recuerdos que me llevo de la etapa universitaria. Muchas gracias al Pato, Tati, Piña y Wlady por crear este proyecto e invitarme a ser parte de él. Además, quisiera mencionar a Daniel, Tomás, Hugo y Ángela, por el apoyo y tiempo invertido en los buenos y malos momentos de la carrera. Mención especial para Luis Alberto "Luchito" Herrera, quien a través de su particular modo me ha enseñado muchísimo. Espero que le saques luego las rueditas chicas a tu bicicleta.

Es imprescindible agradecer a los Profesores que me dieron la oportunidad de trabajar en el proyecto Radiosonda: Marcos Díaz y Roberto Rondanelli. Por otro lado, debo mencionar a Patricio Parada y Jorge Silva (según palabras de P. P., "con quienes me arriesgué a trabajar"). Sin sus cursos y paciencia, comenzar a comprender las probabilidades y la estadística hubiera sido muy complejo. La experiencia, visiones y observaciones de todos Uds. me han permitido crecer académicamente y como persona.

Quisiera destacar el rol de mi Profesor Guía, Dr. Marcos Orchard, en mi formación. Siempre me ha apoyado en el ámbito académico, profesional y también personal. Siempre ha estado atento a mis ideas, inquietudes y motivaciones. Además, me permitió trabajar en un proyecto de mi interés y con plena confianza de su parte. Muchas gracias.

Debo agradecer también a todos quienes ayudaron a corregir este trabajo: Profesores Marcos Orchard, Jorge Silva y Héctor Agusto. También a Amy McDermott, quien desde lejos me ayudó con valiosos comentarios.

Finalmente, darle las gracias a mi mamá e Isabel, por estar siempre conmigo de forma incondicional. Terminar este etapa fue mucho más sencillo con su apoyo.

Contents

1 Introduction **1**

2 Financial Time Series **4**
2.1 Returns and Volatility 4
2.2 Volatility Models 5
2.2.1 The GARCH model 6
2.2.2 The uGARCH model 7
2.3 Final Words 8

3 A Bayesian Framework for Volatility Estimation and Event Detection **10**
3.1 The Tracking Problem 10
3.2 Monte Carlo Integration 11
3.3 Importance Sampling 13
3.4 Particle Filters 13
3.4.1 The Degeneracy Problem 15
3.4.2 Resampling 15
3.4.3 Importance Density Selection 15
3.5 Risk Sensitive Particle Filters 16
3.6 Online Parameter Estimation with Particle Filters 17
3.7 Distance Measures for Probability Density Functions 18
3.7.1 Fisher Discriminant 19
3.7.2 Kullback-Leibler Divergence 19
3.8 Final Words 20

4 Implementation of Volatility Estimation using Particle Filters for a High Volatility Cluster Detection Scheme **21**
4.1 Data 21
4.1.1 Simulated Data 22
4.1.2 Stock Market Data 22
4.2 Filtering Details 24
4.2.1 Estimation of Model Parameters 24
4.2.2 Particle Filters 25
4.3 Detection 27
4.3.1 Detection through Hypothesis Test 27
4.3.2 Detection using the Fisher Discriminant 29
4.3.3 Detection using the Kullback-Leibler Divergence 29

5 Results **31**
5.1 Performance measures 31
5.1.1 Accuracy Indicator for Particle Filter estimates 32
5.1.2 Detection Performance 32
5.2 Estimation through Particle Filters 33
5.2.1 Fitting GARCH Parameters 33
5.2.2 Particle Filter Parameter Selection 34
5.2.3 Estimation Results 37
5.3 Detection using a Hypothesis Test 47
5.4 Detection using Fisher Discriminant and Kullback-Leibler Divergence 55
5.5 Case Study: Early Detection of High Volatility Clusters in IBM's Stock Data 63

6 Conclusion **65**

Bibliography **68**

List of Tables

4.1 GARCH(1,1) model parameters for each data set. The arrow ($\rightarrow$) indicates a change in the parameter value at time step 250. Note that parameters μ and ω are constant for each set. 22

5.1 Parameter estimation using MATLAB®'s `garchfit` function from the Financial Toolbox. 33
5.2 Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH0. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0075$, $R_{th} = 0.6$. 35
5.3 Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH1. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0350$, $R_{th} = 0.6$. 35
5.4 Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH2. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0035$, $R_{th} = 0.5$. 35
5.5 Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH3. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0035$, $R_{th} = 0.7$. 36
5.6 Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH4. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0150$, $R_{th} = 0.7$. 36
5.7 Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH5. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.02$, $R_{th} = 0.6$. 36
5.8 Summary of sensibility analysis for the Classic Particle Filter 37
5.9 Detection results for GARCH0 data set early high volatility cluster detection. 49
5.10 Detection results for GARCH1 data set early high volatility cluster detection. 50
5.11 Detection results for GARCH2 data set early high volatility cluster detection. 51
5.12 Detection results for GARCH3 data set early high volatility cluster detection. 52
5.13 Detection results for GARCH4 data set early high volatility cluster detection. 53
5.14 Detection results for GARCH5 data set early high volatility cluster detection. 54
5.15 Parameter estimation of the GARCH(1,1) model through maximum likelihood for IBM's returns series between September 12$^{\text{th}}$, 2005 and September 1$^{\text{th}}$, 2009. 63

List of Figures

1.1 NASDAQ Composite value for the period January 1st, 2008 to December 31st, 2009. The data show sthe extreme value drop occurring in October 2008. . . 1

4.1 IBM's adjusted closing stock prices and corresponding daily returns from 1st, 1962 to May 17th, 2013. Notice the dashed lines from September 12th, 2005 to September 1st, 2009. 23
4.2 IBM's adjusted closing stock prices and corresponding daily returns from September 12th, 2005 to September 1st, 2009. This figure represents the details between the dashed blue lines of Figures 4.1a and 4.1b. 23
4.3 Comparison between $q_{PF}(\sigma_k^2|\sigma_{k-1}^{2(\mathrm{i})})$ and $q_{RSPF}(\sigma_k^2|\sigma_{k-1}^{2(\mathrm{i})})$. In this example, $\sigma_{k-1}^{2(\mathrm{i})} = 5 \times 10^{-4}, \alpha = 0.2, \beta = 0.6, \omega = 1.0468 \times 10^{-5}$, and $\sigma_\eta^2 = 0.7$. 27
4.4 Examples of prior and posterior densities of the RSPF in a volatility filtering process. 28
4.5 Hypothesis test example. The filled area under the smoothed prior density represents the 70% confidence interval for the smoothed prior density. Here, the null hypothesis is accepted. 29

5.1 Volatility estimation of GARCH0 data set using optimal parameters. Refer to Table 5.2 for percentage error. 40
5.2 Volatility estimation of GARCH1 data set using optimal parameters. Refer to Table 5.3 for error. 40
5.3 Volatility estimation of GARCH2 data set using optimal parameters. Refer to Table 5.4 for error. 41
5.4 Volatility estimation of GARCH3 data set using optimal parameters. Refer to Table 5.5 for error. 41
5.5 Volatility estimation of GARCH4 data set using optimal parameters. Refer to Table 5.6 for error. 42
5.6 Volatility estimation of GARCH5 data set using optimal parameters. Refer to Table 5.7 for error. 42
5.7 Volatility estimation of GARCH0 data set using average parameters. 44
5.8 Volatility estimation of GARCH1 data set using average parameters. 44
5.9 Volatility estimation of GARCH2 data set using average parameters. 45
5.10 Volatility estimation of GARCH3 data set using average parameters. 45
5.11 Volatility estimation of GARCH4 data set using average parameters. 46
5.12 Volatility estimation of GARCH5 data set using average parameters. 46
5.13 Hypothesis test-based detection for data set GARCH0. 49

5.14 Hypothesis test-based detection for data set GARCH1. 50
5.15 Hypothesis test-based detection for data set GARCH2. 51
5.16 Hypothesis test-based detection for data set GARCH3. 52
5.17 Hypothesis test-based detection for data set GARCH4. 53
5.18 Hypothesis test-based detection for data set GARCH5. 54
5.19 PDF comparison-based detection for data set GARCH0. 57
5.20 PDF comparison-based detection for data set GARCH1. 58
5.21 PDF comparison-based detection for data set GARCH2. 59
5.22 PDF comparison-based detection for data set GARCH3. 60
5.23 PDF comparison-based detection for data set GARCH4. 61
5.24 PDF comparison-based detection for data set GARCH5. 62
5.25 Early hypothesis test-based detection of high volatility clusters in IBM's stock data. 64

Chapter 1

Introduction

Electrical Engineering is a discipline which deals with systems and information processing. It uses tools that are transversal to many areas of knowledge; including various branches of Mathematics, Physics, Computer Science, Economics, and even Psychology. Moreover, Electrical Engineering intersects dramatically with different areas of Finance, where signal processing tools (also know as time series analysis) and Information Theory are of paramount importance.

In the past decades, there have been events in the financial world that have led to chaos and desperation in the societies of the occidentalized world. Figure 1.1 shows the big market fall of year 2008, where about 40% of the market value was lost. These rare events are studied by researchers in the areas of extreme event modeling (Embrechts et al., 1997) and in analysis of financial time series (Tsay, 2010). The intention of this work is to explore the use of tools that are well-known in Electrical Engineering and Prognostics and Health Management (PHM) in financial time series analysis, specifically in the early detection of high risk events. This is achieved through the extrapolation of estimation and detection schemes for high risk and low-likelihood events that can be found in PHM to the case of detection of extreme and rare events occurring in the stock market.

Volatility is a key abstraction in finance, since it is the concept necessary to quantify risk. Its calculation makes possible the computation of other features related to risk, such as *Value at Risk* (VaR). Informally, volatility is defined as the standard deviation of the continuously compounded returns of a price series. Albeit its straightforward definition, there are various

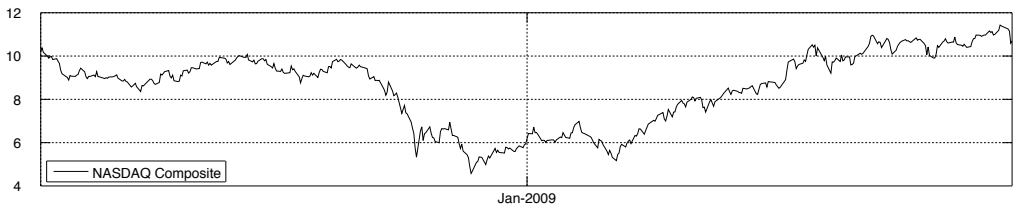


Figure 1.1: NASDAQ Composite value for the period January 1st, 2008 to December 31st, 2009. The data show sthe extreme value drop occurring in October 2008.

problems related to volatility. From the perspective of state-space modeling of financial time series, where volatility is the state and measurements are the continuously compounded returns, volatility is a non-observable state. Consequently, the resulting volatility value is intrinsically related to the modeling technique and the employed estimation scheme. Given that volatility is necessary to detect high risk events, it is necessary to estimate its value accurately and precisely, and to propose detection frameworks based on the estimates of this non-observable state.

The description of the problem leads to the introduction of the specific objectives of this work. The volatility model used is the uGARCH (unobserved GARCH) model (Tobar, 2010), which is a stochastic extension of the well-known GARCH(1,1) (Generalized Autoregressive Conditional Heteroskedasticity) model (Bollerslev, 1986). In particular, this work focuses in a first instance on the implementation of a Classic Particle Filtering framework (Gordon et al., 1993; Bergman, 1999; Doucet et al., 2000; Maskell and Gordon, 2001; Arulampalam et al., 2002), and the proposition and implementation of a Risk Sensitive Particle Filter (Thrun et al., 2002) for the estimation of volatility. Moreover, the information derived from these estimation schemes is used to propose three different detection algorithms, including an hypothesis test-based detection scheme, a Fisher Discriminant-based detection scheme, and a Kullback-Leibler Divergence-based detection scheme. The proposed algorithms are tested with simulated data (where volatility is known) and IBM's stock market data, where volatility is estimated.

The structure of the document is as follows. Chapter 2 includes key concepts of Finance, such as formal definitions of returns and volatility, and introduces two volatility models. Chapter 3 sets the base for Bayesian estimation of time-varying variables, such as volatility. In particular, the tracking problem from a Bayesian perspective is introduced, and Monte Carlo integration is presented as the foundation of Particle Filters. These algorithms are carefully explained, and implementation problems are mentioned together with the currently employed solutions to these problems. Finally, different distance measures for probability density functions are introduced, which set the base for two of the proposed detection algorithms. Chapter 4 describes the implementation details for data generation and acquisition, explains details about the employed filtering frameworks and describes the use of the detection schemes. In Chapter 5, performance measures are presented, and filtering and detection performance are discussed for simulated and real data. Chapter 6 offers the conclusions of this work.

As a note, this document employs the following code for the presentation of figures:

- Data is always presented in black,
- Classic Particle Filter estimates are presented in blue,
- Risk Sensitive Particle Filter estimates are presented in red,
- Confidence intervals are presented in transparent cyan,
- Prior estimates are drawn in dashed lines (- -),
- Posterior estimates are drawn in continuous lines (–),
- Training intervals are presented as red transparent rectangular blocks,
- Regime shifts are drawn as gray dashed lines (- -),

- Detections are shown as magenta circles (over data) and as blue stems,
- Ground truth for detection algorithms is illustrated with green asterisks.

Chapter 2

Financial Time Series

Financial time series analysis is concerned with the theory and practice of asset valuation over time (Tsay, 2010). A key aspect that differentiates financial time series analysis from other diciplines is that both financial theory and its empirical time series contain an element of uncertainty. For example, there are various definitions of asset volatility, and for a stock return series, the volatility is not directly observable. As a result of the added uncertainty, statistical theory and methods play an important role in financial time series analysis.

This chapter sets the theoretical foundation to analyze financial time series. In particular, Section 2.1 presents mathematical definitions of returns and volatility that sets the base for volatility modeling. In this same section, their behavior and properties are presented. Section 2.2 introduces volatility models, which try to capture the empirically observed characteristics. Section 2.3 concludes.

2.1 Returns and Volatility

The study of varying assets usually deals with returns, rather than asset prices. According to (Tsay, 2010), there are two main reasons for using returns instead of prices:

- They are a complete and scale-free summary of the investment opportunity,
- They posses more attractive statistical properties than prices.

There are various possible definitions of asset returns, each with their own properties and applications. This work is founded upon the definition of continuously compounded returns, given in Definition 2.1.

Definition 2.1 (Continuously Compounded Return (Tsay, 2010)) *The natural logarithm of the simple gross return of an asset is called the* continuously compounded return *or* log return*:*

$$r_k = \log \frac{p_k}{p_{k-1}}, \tag{2.1}$$

where p_k corresponds to the asset price at time k.

One can represent the returns as the sum of a conditional expected return $\mu_{k|k-1}$ and an innovation component u_k, that is (Rachev et al., 2008):

$$r_k = \mu_{k|k-1} + u_k, \tag{2.2}$$

where $\mu_{k|k-1} = \mathbb{E}[r_k|\Sigma_{k-1}]$, and Σ_{k-1} denotes the information up to time $k-1$. This information set includes, for example, past asset returns and information about past trading volume. A further decomposition gives:

$$r_k = \mu_{k|k-1} + \sigma_{k|k-1}\epsilon_k, \tag{2.3}$$

where $\sigma_{k|k-1} > 0$ is the volatility and $\epsilon_k \sim \mathcal{N}(0,1)$ is an independent and identically distributed (iid) process. Definition 2.2 formally introduces the concept of volatility.

Definition 2.2 (Conditional Volatility (Rachev et al., 2008)) *Conditional volatility is defined as*

$$\sigma^2_{k|k-1} = \mathbb{V}ar[r_k|\Sigma_{k-1}] = \mathbb{E}[(r_k - \mu_{k|k-1})^2|\Sigma_{k-1}], \tag{2.4}$$

where $\mu_{k|k-1} = \mathbb{E}[r_k|\Sigma_{k-1}]$ is the conditional expected value of the returns and Σ_{k-1} is the σ-algebra generated by the variables observed up to time $k-1$.

Empirical evidence shows some features that interestingly relate returns and volatility. First, volatility is not constant, therefore returns are *heteroskedastic* (Rachev et al., 2008). Moreover, clusters of high and low returns may be observed, a phenomenon first observed by (Mandelbrot, 1963). This characteristic implies that volatility also comes in clusters. Second, volatility displays an asymmetric behavior towards positive and negative return variations, since it tends to be higher when the market falls. This phenomenon is known as the *leverage* effect. Third, volatility evolves over time in a continuous manner. Fourth, volatility does not diverge to infinity.

Volatility is important in the analysis of financial time series for diverse reasons. In particular, it has a strict relation with risk, and provides a simple approach towards the calculation of *Value at Risk* (VaR). According to (Tsay, 2010), volatility also plays an important role in asset allocation, and modeling the volatility of a time series can improve efficiency in parameter estimation and the accuracy in interval forecasting.

These two concepts set the foundation for the application of signal processing tools in the analysis of financial time series, and are the basic principles to be used in this work.

2.2 Volatility Models

Volatility modeling rises as a necessity to better understand the structure of returns series. In state-space models, volatility is considered a variable that drives a dynamics process, which conditions a measurement process for the returns. Classic statistical analysis of time series

is used for the construction of these models, such as autocorrelation functions and partial autocorrelation functions of the square of the returns series.

Empirical data shows that returns processes exhibit excess kurtosis, skewness and heavier tails than the normal distribution (Rachev et al., 2008; Tsay, 2010). Moreover, since present volatility depends on past volatilities, it is clear that today's returns depend on yesterday's returns, as described in Equation 2.3. Volatility models try to explain these features. This section describes the well-known GARCH model and a stochastic variation –the uGARCH model–, which are attempts to explain these observed characteristics.

2.2.1 The GARCH model

The GARCH model, first presented by (Bollerslev, 1986), is widely used to model volatility because it provides a large degree of flexibility in capturing empirically observed features of returns, and because of the ease of parameter estimation.

Definition 2.3 (GARCH model (Rachev et al., 2008)) *The GARCH(1,1) model is given by*

$$\sigma^2_{k|k-1} = \omega + \alpha u^2_{k-1} + \beta\sigma^2_{k-1|k-2}, \tag{2.5}$$

$$r_k = \mu_{k|k-1} + \sigma_{k|k-1}\epsilon_k, \tag{2.6}$$

where $u_k = r_k - \mu_{k|k-1} = \sigma_{k|k-1}\epsilon_k$, *and* $\omega > 0$. *Moreover,* $\alpha, \beta > 0$ *to ensure that* $\sigma^2_{k|k-1} > 0$, *for any time step* k.

From Definition 2.3, using Equation 2.6 in 2.5 results in:

$$\sigma^2_{k|k-1} = \omega + \alpha\sigma^2_{k-1|k-2}\epsilon^2_{k-1} + \beta\sigma^2_{k-1|k-2}, \tag{2.7}$$

where $\epsilon^2_{k-1} \sim \chi^2_1$. Equation 2.7 shows that new information in the volatility process is embodied by the term ϵ^2_{k-1}, this is, the innovations in the volatility process are driven by the previous step innovations of the returns process.

To guarantee the convergence of $\sigma^2_{k|k-1}$, it is necessary that $\alpha + \beta < 1$. To prove this fact, note that $\sigma^2_{k|k-1}$ and ϵ^2_k are independent (Tobar, 2010):

$$\begin{aligned}
\mathbb{E}[\sigma^2_{k|k-1}\epsilon^2_k] &= \mathbb{E}[(\omega + \alpha\epsilon^2_{k-1}\sigma^2_{k-1|k-2} + \beta\sigma^2_{k-1|k-2})\epsilon^2_k] \\
&= \mathbb{E}[\omega\epsilon^2_k] + \mathbb{E}[\alpha\epsilon^2_{k-1}\sigma^2_{k-1|k-2}\epsilon^2_k] + \mathbb{E}[\beta\sigma^2_{k-1|k-2}\epsilon^2_k] \\
&= \omega\mathbb{E}[\epsilon^2_k] + \alpha\mathbb{E}[\epsilon^2_{k-1}\sigma^2_{k-1|k-2}]\mathbb{E}[\epsilon^2_k] + \beta\mathbb{E}[\sigma^2_{k-1|k-2}]\mathbb{E}[\epsilon^2_k] \\
&= \mathbb{E}[\epsilon^2_k](\omega + \alpha\mathbb{E}[\epsilon^2_{k-1}\sigma^2_{k-1|k-2}] + \beta\mathbb{E}[\sigma^2_{k-1|k-2}]) \\
&= \mathbb{E}[\epsilon^2_k]\mathbb{E}[\sigma^2_{k|k-1}].
\end{aligned} \tag{2.8}$$

With the result from Equation 2.8 and considering that $\epsilon_k \sim \mathcal{N}(0,1)$ implies that $\mathbb{E}[\epsilon^2_k] = 1$, it is possible to calculate the expected value of $\sigma^2_{k|k-1}$:

$$\begin{aligned}
\mathbb{E}[\sigma^2_{k|k-1}] &= \mathbb{E}[\omega + \alpha\epsilon^2_{k-1}\sigma^2_{k-1|k-2} + \beta\sigma^2_{k-1|k-2}] \\
&= \omega + \alpha\mathbb{E}[\epsilon^2_{k-1}]\mathbb{E}[\sigma^2_{k-1|k-2}] + \beta\mathbb{E}[\sigma^2_{k-1|k-2}] \\
&= \omega + (\alpha + \beta)\mathbb{E}[\sigma^2_{k-1|k-2}].
\end{aligned} \tag{2.9}$$

Expression 2.9 proves that $\alpha + \beta < 1$ is needed for $\mathbb{E}[\sigma^2_{k|k-1}] < \infty$. The value of $\alpha + \beta$ is known as the GARCH *process persistence parameter*, since it determines the speed of the mean-reversion of volatility to its long-term average. A higher value for $\alpha + \beta$ implies that the effect of the shocks of volatility, u^2_k, dies out slowly (Rachev et al., 2008). Empirical evidence shows that the value of $\alpha + \beta$ is usually close to 1.

The GARCH(1,1) model has interesting properties that make it suitable for volatility modeling and easy practical handling, including:

- Parameter selection may be easily accomplished through maximum likelihood,
- If $\alpha + \beta < 1$, then the process is stationary in the weak sense,
- It is able to model volatility clusters,
- Can partially explain nonnormality of asset returns.

These reasons make the GARCH(1,1) a widely used alternative for volatility modeling. Nevertheless, the evolution of volatility in the GARCH(1,1) is conditionally deterministic (see Equation 2.7), not allowing possible statistical analysis of volatility. Inference of statistical properties such as asymmetry, kurtosis, and confidence intervals is simply not possible in estimation and prediction problems. These difficulties lead to the development of the uGARCH model.

2.2.2 The uGARCH model

The uGARCH model (Tobar, 2010) assumes that the dynamics of volatility are not driven by the observed process $u_k = r_k - \mu_{k|k-1}$. Instead, they are driven by a non-observable process u'_k which has the same distribution as u_k.

Definition 2.4 (uGARCH model (Tobar, 2010)) *The uGARCH model is given by*

$$\sigma^2_k = \omega + \alpha\sigma^2_{k-1}\eta^2_k + \beta\sigma^2_{k-1}, \tag{2.10}$$
$$r_k = \mu + \sigma_k\epsilon_k, \tag{2.11}$$

where r_k is a process of returns, σ_k is the stochastic volatility, $\mu \in \mathbb{R}^+, \omega \in \mathbb{R}^+, \alpha, \beta > 0$ are parameters, and $\alpha + \beta < 1$. Furthermore, $\epsilon_k \sim \mathcal{N}(0,1)$ and $\eta_k \sim \mathcal{N}(0, \sigma^2_\eta)$ are iid *processes for every time step k.*

It is necessary to note from Definition 2.4 that the subscripts are not written conditionally: At time step k, σ^2_k is not known without uncertainty, given Σ_{k-1}.

To completely define the model, it is necessary to present the state transition distribution

$p(\sigma_k^2|\sigma_{k-1}^2)$ and the likelihood $p(r_k|\sigma_k^2)$:

$$p(\sigma_k^2|\sigma_{k-1}^2) = \frac{1}{\sqrt{2\pi\alpha\sigma_{k-1}^2(\sigma_k^2 - \omega - \beta\sigma_{k-1}^2)}} \exp\left(\frac{\omega + \beta\sigma_{k-1}^2 - \sigma_k^2}{2\alpha\sigma_{k-1}^2}\right), \quad \sigma_k^2 \geq \omega + \beta\sigma_{k-1}^2 \tag{2.12}$$

$$p(r_k|\sigma_k^2) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(r_k - \mu)^2}{2\sigma_k^2}\right). \tag{2.13}$$

The derivation of $p(\sigma_k^2|\sigma_{k-1}^2)$ in this case has considered that $\eta_k \sim \mathcal{N}(0,1)$ and therefore, $\eta_k^2 \sim \chi_1^2$ (Tobar, 2010). To derive an expression for the state transition distribution when $\eta_k \sim \mathcal{N}(0, \sigma_\eta^2)$ and σ_η^2 is the variance of the Normal distribution that is not necessarily equal to 1, it is imperative to calculate the probability density function of η_k^2. According to the Fundamental Theorem for the distribution of $g(X)$ (Papoulis and Pillai, 2002, chap. 5, pg. 130), $\eta_k^2 = X$ has an analytical probability density function:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma_\eta^2 x}} \exp\left[-\frac{x}{2\sigma_\eta^2}\right]. \tag{2.14}$$

Using Expression 2.14, it is possible to recalculate $p(\sigma_k^2|\sigma_{k-1}^2)$:

$$\begin{aligned}
p(\sigma_k^2|\sigma_{k-1}^2) &= p(\omega + \alpha\sigma_{k-1}^2\eta_k^2 + \beta\sigma_{k-1}^2|\sigma_{k-1}^2) \\
&= \frac{\partial}{\partial\sigma_k^2}\mathbb{P}[\omega + \alpha\sigma_{k-1}^2\eta_k^2 + \beta\sigma_{k-1}^2 < \sigma_k^2] \\
&= \frac{\partial}{\partial\sigma_k^2}\mathbb{P}\left[\eta_k^2 < \frac{\sigma_k^2 - \omega + \beta\sigma_{k-1}^2}{\alpha\sigma_k^2}\right] \\
&= \frac{1}{\sqrt{2\pi\sigma_\eta^2\left(\frac{\sigma_k^2 - \omega + \beta\sigma_{k-1}^2}{\alpha\sigma_k^2}\right)}} \exp\left[\frac{\sigma_k^2 - \omega + \beta\sigma_{k-1}^2}{2\sigma^2\alpha\sigma_k^2}\right] \frac{\partial}{\partial\sigma_k^2}\left(\frac{\sigma_k^2 - \omega + \beta\sigma_{k-1}^2}{\alpha\sigma_k^2}\right) \\
&= \frac{1}{\sqrt{2\pi\sigma_\eta^2\alpha\sigma_{k-1}^2\left(\sigma_k^2 - \omega + \beta\sigma_{k-1}^2\right)}} \exp\left[\frac{\sigma_k^2 - \omega + \beta\sigma_{k-1}^2}{2\sigma_\eta^2\alpha\sigma_k^2}\right].
\end{aligned} \tag{2.15}$$

Notice that σ_η^2 is introduced as a scale factor. The calculation and presentation of Expression 2.15 makes possible the use of a generic Particle Filtering approach for volatility estimation in this model.

2.3 Final Words

In this chapter, an introduction of basic concepts of financial time series has been given. Notions such as returns and volatility have been defined, and the problem of volatility modeling has been addressed through the introduction of two models, each one representing one of the two possible trends for volatility analysis: the GARCH model, which considers that the dynamics of the process are deterministic and completely known at time k with measurements

up to time $k-1$, and the uGARCH model that considers that the dynamics of volatility are driven by a non-observable process $\eta_k^2 \sim \chi_1^2$.

These definitions open the path for the possibility of volatility estimation in a Bayesian framework, in which the *a posteriori* $p(\sigma_k^2|r_k)$ may be inferred. The estimated statistics involved in this algorithm also let one define hypotheses tests. These subjects are addressed in the following chapters.

Chapter 3

A Bayesian Framework for Volatility Estimation and Event Detection

Monte Carlo (MC) methods and *Markov Chain Monte Carlo* (MCMC) methods have been widely used to approximate integrals and probability density functions (Tobar, 2010). Nevertheless, their use in Bayesian inference is not direct, since this problem involves a sequence of time-variant probability density functions, while MCMC assumes that the objective density is time-invariant. This prompted the development of a sequential version of Monte Carlo integration, one that is able to use measurements to improve recursive estimation.

The first section of this chapter introduces the tracking problem, which gives insight into the problems encountered in a Bayesian filtering framework. The following section presents Monte Carlo integration and the *importance sampling* method. This opens the possibility to explore the Particle Filter and the Risk Sensitive Particle Filter. Finally, distance measures for probability density functions are given, which are the core of detection schemes proposed in the following chapters.

3.1 The Tracking Problem

Many problems in time series analysis can be formulated with the help of state-space models. These models consider a transition equation that describes the prior distribution of a hidden Markov process $\{x_k; k \in \mathbb{N}\}$, called the state process, and an observation equation describing the likelihood of the observation $\{z_k; k \in \mathbb{N}\}$ (Doucet et al., 2000). To define the problem of tracking, consider the state-space model (Arulampalam et al., 2002)

$$x_k = f(x_{k-1}, v_{k-1}), \tag{3.1}$$

$$z_k = h(x_k, w_k), \tag{3.2}$$

where $f(\cdot, \cdot)$ is a state-transition function with corresponding $\{v_{k-1}, k \in \mathbb{N}\}$ iid innovation process, and $h(\cdot, \cdot)$ is the observation function with $\{w_k, k \in \mathbb{N}\}$ its corresponding iid noise process. In particular, the objective of tracking is to recursively estimate x_k from all available measurements $z_{1:k} = \{z_\mathrm{i};\ \mathrm{i} = 1, \ldots, k\}$ up to time k.

Within a Bayesian framework, all relevant information about $x_{0:k}$ given the observations $z_{0:k}$ can be obtained from the posterior distribution $p(x_{0:k}|z_{0:k})$. In many applications, nevertheless, it is sufficient to calculate the marginal conditional distribution $p(x_k|z_{0:k})$. In particular, the intention of the Bayesian approach is to recursively construct $p(x_k|z_{1:k})$, using

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})}. \tag{3.3}$$

In principle, $p(x_k|z_{1:k})$ might be obtained in two stages: prediction and update. Let one assume that the initial probability density function $p(x_0|z_0) \equiv p(x_0)$ and $p(x_{k-1}|z_{k-1})$ are available. The prediction step involves using the system dynamics in Equation 3.1 to obtain the prior probability density function of the state at time k via the Chapman-Kolgomorov equation (Arulampalam et al., 2002; Papoulis and Pillai, 2002):

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\mathrm{d}x_{k-1}. \tag{3.4}$$

At time step k, a new measurement becomes available, which may be used to update the prior distribution $p(x_k|z_{1:k-1})$. Considering that the normalizing constant $p(z_k|z_{1:k-1})$ can be expressed as

$$p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1})\mathrm{d}x_k, \tag{3.5}$$

one can calculate the posterior distribution $p(x_k|z_{1:k})$, using Equations 3.4 and 3.5 in 3.3:

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)\int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})\mathrm{d}x_{k-1}}{\int p(z_k|x_k)p(x_k|z_{1:k-1})\mathrm{d}x_k}. \tag{3.6}$$

Expression 3.6 forms the basis for the Bayesian optimal solution in the mean square error sense. In most cases, this expression is only conceptual, and cannot be determined analytically. In a restricted set of cases, the optimal solution may be found (Kalman, 1960). This is possible only if the noises v_k and w_k are additive and Gaussian and the functions $f(\cdot,\cdot)$ are $h(\cdot,\cdot)$ are linear. The following sections builds upon Monte Carlo integration to describe a general algorithm for suboptimal solutions to Expression 3.6.

3.2 Monte Carlo Integration

The problem of tracking described in the previous section presents no analytical solution for a vast majority of applications; since Expressions 3.4 and 3.5 can be calculated only for a small subset of models and noise processes. To attack this problem, it is necessary to use numerical techniques to approximate the integrals in order to obtain an estimate $\hat{p}(x_k|z_{1:k})$ of the posterior distribution.

A common problem encountered in applied statistics deals with the numerical calculation of expressions such as:

$$I = \int_{\mathbb{R}^n} g(x)\mathrm{d}x. \tag{3.7}$$

Monte Carlo methods for numerical integration regard problems of the form (Bergman, 1999):

$$I = \int_{\mathbb{R}^n} f(x)\pi(x)\mathrm{d}x, \tag{3.8}$$

where $\pi(x)$ is a positive function that integrates to unity,

$$\pi(x) \geq 0, \qquad I = \int_{\mathbb{R}^n} \pi(x)\mathrm{d}x = 1. \tag{3.9}$$

Many of the problems that can be defined as Expression 3.7 can be converted into Expression 3.8 through a suitable transformation $g(x) = f(x)\pi(x)$. The assumptions imposed over $\pi(x)$ resemble a probability density function, and in a Bayesian context, the density of interest is the posterior density of the parameters given the observed data, i.e., $\pi(x) = p(x|z)$ (see Equations 3.4 and 3.5).

The Monte Carlo methods rely on the assumption that it is possible to draw $N >> 1$ samples $\{x^{(\mathrm{i})}\}_{\mathrm{i}=1}^{N}$ distributed according to $\pi(x)$. The Monte Carlo estimate of the integral in Expression 3.8 is formed by taking the average over the set of samples

$$f_N = \frac{1}{N_s}\sum_{\mathrm{i}=1}^{N_s} f(x^{(\mathrm{i})}), \tag{3.10}$$

where N_s is assumed to be large. If the samples in the set $\{x_{(\mathrm{i})}\}_{\mathrm{i}=1}^{N_s}$ are independent, f_{N_s} will be an unbiased estimate and will almost surely converge to I,

$$\mathbb{P}\left(\lim_{N_s\to\infty} f_{N_s} = I\right) = 1, \tag{3.11}$$

by the strong law of large numbers. Moreover, if the variance of $f(x)$,

$$\sigma^2 = \int_{\mathbb{R}^n} \left(f(x) - I\right)^2 \pi(x)\mathrm{d}x = \int_{\mathbb{R}^n} f^2(x)\pi(x)\mathrm{d}x - I^2, \tag{3.12}$$

is finite, the central limit theorem yields convergence in distribution of the error:

$$\lim_{N_s\to\infty} \sqrt{N_s}(f_{N_s} - I) \sim \mathcal{N}(0, \sigma^2). \tag{3.13}$$

The convergence results in Equation 3.11 and Equation 3.13 are asymptotic, so the approximation error tends to zero as $N_s \to \infty$. In fact, Expression 3.11 yields that the error $\varepsilon = f_{N_s} - I$ of the Monte Carlo estimate is of order $\varepsilon = \mathcal{O}(N_s^{-1/2})$, independent of the state dimension n. This results in an interesting characteristic from a filtering perspective, since one can control the approximation error for the estimation as a function of the samples used.

The factorization of Expression 3.7 presented in Expression 3.8 has some interesting implications that need further revision. The effectiveness of the method depends deeply on how the factorization $g(x) = f(x)\pi(x)$ is performed. Since the set $\{x^{(\mathrm{i})}\}_{\mathrm{i}=1}^{N_s}$ is automatically chosen from $\pi(x)$, one can conclude that the more informative $\pi(x)$ is compared to $f(x)$, the better the choice of automatic samples will be. This claim is verified by Expression 3.13, where the relative smoothness of $f(x)$ compared to $\pi(x)$ determines the size of the variance σ^2.

3.3 Importance Sampling

Importance sampling arises as a method for sampling from $\pi(x)$ when this distribution is not part of the standard probability density functions, such as the exponential family. In the Bayesian framework, where the factorization $g(x) = f(x)\pi(x)$ is given, if one deals with $\pi(x) = p(x|z)$, where $p(x|z)$ comes from a nonlinear/non-Gaussian model, it is not always possible to extract samples from $\pi(x)$. Therefore, a more general approach towards sampling is necessary to have a general sequential Monte Carlo filtering algorithm.

Importance sampling deals with a proposal distribution $q(x)$ from which is easy to generate samples (Bergman, 1999). The general assumption on the *importance function* $q(x)$ is that the support set covers the support of $\pi(x)$, this is, $\pi(x) > 0 \Rightarrow q(x) > 0$ for all $x \in \mathbb{R}^n$. Under this assumption, any integral of the form 3.8 can be rewritten into

$$I = \int_{\mathbb{R}^n} f(x)\pi(x)\mathrm{d}x = \int_{\mathbb{R}^n} f(x)\frac{\pi(x)}{q(x)}q(x)\mathrm{d}x. \tag{3.14}$$

A Monte Carlo estimate is computed by generating $N_s \gg 1$ independent samples from $q(x)$, and forming the weighted sum:

$$f_{N_s} = \frac{1}{N_s}\sum_{\mathrm{i}=1}^{N_s} f(x^{(\mathrm{i})})w(x^{(\mathrm{i})}), \quad \text{where} \quad w(x^{(\mathrm{i})}) \propto \frac{\pi(x^{(\mathrm{i})})}{q(x^{(\mathrm{i})})} \tag{3.15}$$

are the importance weights.

If the normalizing factor of the density $\pi(x)$ is unknown, the importance weights in Expression 3.15 can be evaluated up to a normalizing factor. Then, the weights can be formed using a function proportional to the target density and then normalized afterwards, forming the estimate:

$$f_{N_s} = \frac{1}{N_s}\frac{\sum_{\mathrm{i}=1}^{N_s} f(x^{(\mathrm{i})})w(x^{(\mathrm{i})})}{\sum_{\mathrm{i}=1}^{N_s} w(x^{(\mathrm{i})})}, \quad \text{where} \quad w(x^{(\mathrm{i})}) \propto \frac{\pi(x^{(\mathrm{i})})}{q(x^{(\mathrm{i})})} \tag{3.16}$$

This technique is practically applied in the Bayesian framework. Although the estimate 3.16 is biased for finite N_s, the law of large numbers and central limit theorem hold.

The importance sampling principle works as the basis for the most common Particle Filter algorithms. These are introduced in the following section.

3.4 Particle Filters

Particle Filters are a class of algorithms developed to solve Expression 3.6 through sequential Monte Carlo simulations when the integrals are intractable due to possible nonlinearities in the model involved or noise processes that do not possess standard distributions. Solving these integrals is achieved through the *importance sampling* principle. The key idea is to

represent the required posterior density function by a set of random samples which serve as support points with associated weights and to compute estimates based on these samples and weights. As the number of samples becomes very large, this Monte Carlo characterization becomes an equivalent representation to the usual functional description of the posterior probability density function, and the Particle Filter approaches the optimal Bayesian state in the mean square error sense.

According to (Arulampalam et al., 2002), to develop the details of the algorithm, let $\{x_{0:k}^{(\mathrm{i})}, w_k^{(\mathrm{i})}\}_{\mathrm{i}=1}^{N_s}$ denote a random measure that characterizes the posterior probability density function $p(x_{0:k}|z_{1:k})$, where $\{x_{0:k}^{(\mathrm{i})}, \mathrm{i} = 0, \ldots, N_s\}$ is a set of support points with associated weights $\{w_k^{(\mathrm{i})}, \mathrm{i} = 1, \ldots, N_s\}$ and $x_{0:k} = \{x_j, j = 0, \ldots, k\}$ is the set of all states up to time k. The weights are normalized such that $\sum_{\mathrm{i}} w_k^{(\mathrm{i})} = 1$. Then, the posterior density at k can be approximated as:

$$p(x_{0:k}|z_{1:k}) \approx \sum_{\mathrm{i}=1}^{N_s} w_k^{(\mathrm{i})} \delta(x_{0:k} - x_{0:k}^{(\mathrm{i})}). \tag{3.17}$$

The former approximation may be obtained using an *importance density* $q(x_{0:k}|z_{1:k})$, and the weights are calculated using Equation 3.16 in the sequential case:

$$w_k^{(\mathrm{i})} \propto \frac{p(x_{0:k}^{(\mathrm{i})}|z_{1:k})}{q(x_{0:k}^{(\mathrm{i})}|z_{1:k})}. \tag{3.18}$$

To effectively use Expression 3.18 in a sequential algorithm, it is necessary to express 3.18 in a recursive manner. At each iteration, one could have samples constituting an approximation to $p(x_{0:k-1}|z_{1:k-1})$ and want to approximate $p(x_{0:k}|z_{1:k})$ with a new set of samples. If the importance density is chosen to factorize such that (Doucet et al., 2000; Arulampalam et al., 2002):

$$q(x_{0:k}|z_{1:k}) = q(x_k|x_{0:k-1}, z_{1:k}) q(x_{0:k-1}|z_{1:k-1}) \tag{3.19}$$

$$= q(x_0|z_0) \prod_{j=1}^{k} q(x_j|x_{0:j-1}, z_{0:j}), \tag{3.20}$$

then one can obtain samples $x_{0:k}^{(\mathrm{i})} \sim q(x_{0:k}|z_{1:k})$ by augmenting each of the existing samples $x_{0:k-1}^{(\mathrm{i})} \sim q(x_{0:k-1}|z_{1:k-1})$ with the new state $x_k^{(\mathrm{i})} \sim q(x_k|x_{0:k-1}^{(\mathrm{i})}, z_{1:k})$. Then, the weights are updated recursively using (Arulampalam et al., 2002):

$$w_k^{(\mathrm{i})} \propto w_{k-1}^{(\mathrm{i})} \frac{p(z_k|x_k^{(\mathrm{i})}) p(x_k^{(\mathrm{i})}|x_{k-1}^{(\mathrm{i})})}{q(x_k^{(\mathrm{i})}|x_{k-1}^{(\mathrm{i})}, z_k)}, \tag{3.21}$$

and the posterior filtered density $p(x_k|z_{1:k})$ can be approximated as:

$$p(x_k|z_{1:k}) \approx \sum_{\mathrm{i}=1}^{N_s} w_k^{(\mathrm{i})} \delta(x_k - x_k^{(\mathrm{i})}), \tag{3.22}$$

where the weights are defined in Expression 3.21. This algorithm is generally called *Sampling Importance Sampling* (SIS), and denotes the simplest form to solve Equation 3.6. Nevertheless, to obtain good performance from the algorithm, some issues need to be addressed properly.

3.4.1 The Degeneracy Problem

A common problem with the SIS algorithm is the negligible weight all but one particle has after some iterations. (Doucet et al., 2000) have shown that the variance of weights can only increase over time, and therefore it is not possible to avoid this problem. This phenomenon produces that only one particle has a significant contribution to the overall estimation, and thus, heavy computational effort is lost.

A common measure for degeneracy is (Kong et al., 1994; Maskell and Gordon, 2001):

$$N_{eff} = \frac{N_s}{1 + \mathbb{V}ar\left[w_k^{*\mathrm{i}}\right]}, \tag{3.23}$$

where $w_k^{*\mathrm{i}} = p(x_k^{(\mathrm{i})}|z_{1:k})/q(x_k^{(\mathrm{i})}|x_{k-1}^{(\mathrm{i})}, z_k)$ is referred to as the *true weight*. Expression 3.23 cannot be evaluated and as a consequence, the approximation

$$\hat{N_{eff}} = \frac{1}{\sum_{\mathrm{i}=1}^{N_s}\left(w_k^{(\mathrm{i})}\right)^2} \tag{3.24}$$

must be used, where $w_k^{(\mathrm{i})}$ are the normalized weights. Big weight unconditional variance implies great variations in weight values and hence, small values of $\hat{N_{eff}}$ indicate severe degeneracy. Different techniques have been used to confront this problem, in which resampling methods and selection of an adequate importance density are mentioned.

3.4.2 Resampling

One of the methods to overcome the problem of degeneracy is resampling whenever the unconditional variance of the weights is high, this is, $N_{eff} < N_T$, where N_T is a given threshold. The idea behind resampling is to concentrate on the particles which possess more weight and discard the particles with negligible weight. There are various techniques to do so. In this work, residual resampling is used (see Algorithm 2 in (Maskell and Gordon, 2001)).

3.4.3 Importance Density Selection

The selection of the importance density is crucial for the correct performance of the filter being implemented. Since the importance density defines the position where particles are evaluated, incorrect integration domains may lead to poor estimations or high degeneracies.

The importance density function which minimizes the variance of the true weights $w_k^{(\mathrm{i})*}$, conditioned upon $x_{k-1}^{(\mathrm{i})}$ and z_k has been shown to be (Doucet et al., 2000; Maskell and Gordon, 2001):

$$q(x_k|x_{k-1}^{(\mathrm{i})}, z_k)_{opt} = p(x_k|x_{k-1}^{(\mathrm{i})}, z_k) \tag{3.25}$$

$$= \frac{p(z_k|x_k, x_{k-1}^{(\mathrm{i})})p(x_k|x_{k-1}^{(\mathrm{i})})}{p(z_k|x_{k-1}^{(\mathrm{i})})}. \tag{3.26}$$

Substituting Expression 3.26 into 3.21 yields:

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(z_k|x_{k-1}^{(i)}) \tag{3.27}$$

$$= w_{k-1} \int p(z_k|x'_k) p(x'_k|x_{k-1}^{(i)}) \mathrm{d}x'_k. \tag{3.28}$$

This choice of importance density is optimal since for a given $x_{k-1}^{(i)}$, $w_k^{(i)}$ takes the same value whatever sample is drawn from $q(x_k|x_{k-1}^{(i)}, z_k)_{opt}$. Nevertheless, it is almost never possible to draw samples from this importance density.

A common and convenient important density is:

$$q(x_k|x_{k-1}^{(i)}, z_k) = p(x_k|x_{k-1}^{(i)}). \tag{3.29}$$

Substitution of Expression 3.29 over 3.21 yields:

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(z_k|x_k^{(i)}). \tag{3.30}$$

This is one of the most common elections for the importance density since it is intuitive and simple to implement.

3.5 Risk Sensitive Particle Filters

Particle Filters described in Section 3.4 approximate $p(x_k|z_{0:k})$ through a random set of N_s particles. The position of the particles and consequent performance of the filter is greatly determined by the importance density $q(x_k|x_{k-1}^{(i)}, z_k)$ from which particles are drawn. The structure of the Particle Filter algorithm and importance densities usually employed do not regard the problem of high risk and low-likelihood event tracking. In the case where unlikely events may conduce to great loss or high costs, it is natural extend the Particle Filter algorithm to track these low probability states.

The Risk Sensitive Particle Filter is proposed as an extension of the 'Classic' Particle Filter, where the particles are generated from an importance density that is the product of the combination of the posterior density function and a risk functional.

Risk Sensitive Particle Filters generate samples that are distributed according to (Thrun et al., 2002):

$$q(x_k|x_{k-1}^{(i)}, z_k) = \gamma_k r(x_k) p(x_k|z_{1:k}), \tag{3.31}$$

where

$$\gamma_k = \frac{1}{\int r(x_k) p(x_k|z_{1:k})} \tag{3.32}$$

is a normalizing constant that ensures that the importance density is indeed a probability density function. Hence, the position of samples are generated according to the likelihood of a certain state event $x_k^{(i)}$ and its risk $r(x_k^{(i)})$.

Considering the former approach, the Classic Particle Filter needs a simple modification. First, the initial set of particles $x_0^{(\mathrm{i})}$ is generated from $\gamma_0 r(x_0)p(x_0)$, and Expression 3.30 is updated to

$$w_l^{(\mathrm{i})} = \frac{r(x_k^{(\mathrm{i})})p(z_k|x_k^{(\mathrm{i})})}{r(x_{k-1}^{(\mathrm{i})})}. \tag{3.33}$$

To find an adequate risk functional, authors in (Thrun et al., 2002) formulate a Markov Decision Process and define it over an extended state-space $\langle x, c\rangle$. $c \in \{0, 1\}$ is a binary variable that models the event of correct tracking ($c_k = 1$) or incorrect tracking ($c_k = 0$). Then, the known probabilities are extended to:

$$p(\langle x_k, c_k\rangle|\langle x_{k-1}, c_{k-1}\rangle) = p(x_k|x_{k-1})p(c_k|c_{k-1}) \tag{3.34}$$
$$p(z_k|\langle x_k, c_k\rangle) = p(z_k|x_k) \tag{3.35}$$
$$p(\langle x_0, c_0\rangle) = p(x_0)p(c_0) \tag{3.36}$$
$$C(\langle x_k, c_k\rangle) = C(x_k), \tag{3.37}$$

where the only unknown components are $p(c_0)$ and $p(c_k|c_{k-1})$. In (Thrun et al., 2002), the Markov Decision Process is solved via value iteration.

Another approach that may be used for Risk Sensitive Particle Filters is to select an *importance density* $q(x_k|x_{k-1}^{(\mathrm{i})}, z_k)$ with higher likelihoods for state values associated to higher risk. One may assume that for a particular importance density, a risk functional exists, and therefore, Expression 3.21 is used for updating the weights.

3.6 Online Parameter Estimation with Particle Filters

In the context of state estimation, it is sometimes necessary to handle an online estimation scheme for a model parameter vector. Although parameters α and β have been presented as fixed in the GARCH(1,1) and uGARCH models, this is not necessarily adequate. The stock market suffers from variations and regime shifts, and these variations may be considered as parameter changes through time. This is true not only for time series derived from the stock market, but for a diverse range of applications where state tracking is intended.

To understand the problems of parameter estimation outside a Bayesian context, let θ be a vector parameter. The maximum likelihood estimate of the vector parameter θ is obtained by maximizing the log-likelihood function (Kitagawa and Sato, 2001):

$$l(\theta) = \log[L(\theta)] = \sum_{k=1}^{\kappa} \log[p(z_k|z_{1:k-1}, \theta)], \tag{3.38}$$

where

$$p(z_k|z_{1:k-1}, \theta) = \int p(z_k|x_k, \theta)p(x_k|z_{1:k-1}, \theta)\mathrm{d}x_k \tag{3.39}$$

needs to be approximated through Monte Carlo.

The maximization of Expression 3.38 for the estimation of θ is not always direct, and approximations over Expression 3.39 make this method impractical, due to the high computational costs involved if parameter estimation is intended for every time step. Thus, a different perspective is necessary to approach the online parameter estimation problem. This idea is attacked through the artificial evolution of parameters.

The first ideas about introducing random disturbances to particles were proposed by (Gordon et al., 1993). In their work, the authors propose to introduce random disturbances to the positions of particles (called *roughening penalties*) in order to combat degeneracy. This idea has been extended in order to estimate online a vector of fixed model parameters, which is referred to as *artificial evolution* (Liu and West, 2001). Artificial evolution of parameters is a simple and powerful idea, nevertheless, it requires careful handling because of the inherent model information loss given by the consideration of time-varying parameters that are fixed.

Instead of estimating the vector parameter θ through maximum likelihood, the Bayesian framework may be introduced to estimate θ online. This is achieved by augmenting the state vector x_k with unknown parameters θ as:

$$x'_k = \begin{bmatrix} \theta_k \\ x_k \end{bmatrix}, \tag{3.40}$$

where $\theta_k = \theta$ implies the consideration of an extended model where parameters are time-varying. Then, an independent, zero-mean normal increment is added to the parameter at each time step (Liu and West, 2001):

$$\theta_k = \theta_k + \zeta_k, \tag{3.41}$$

$$\zeta_k \sim \mathcal{N}(0, W_k), \tag{3.42}$$

where W_k is a variance matrix and θ_k and ζ_k are conditionally independent given Σ_k. The key motivation is that the artificial evolution of parameters gives new values for each iteration, and thus, weight assignment in Particle Filters considers the likelihood of the state and parameter values.

3.7 Distance Measures for Probability Density Functions

The current chapter describes Particle Filters (Sections 3.4 and 3.5), which aim to solve the Bayesian tracking problem through Monte Carlo simulations. In particular, Section 3.4 describes how prior and posterior estimates are formulated through a Bayesian estimation framework.

Detection schemes are commonly based in the comparison of probability density functions generated from data. This comparison may be executed through different techniques, including hypothesis tests and computation of the distance between probability densities. Considering that Particle Filter outputs are obtained as probability density functions (prior and posterior), it seems straightforward to process the estimates in order to propose a detection scheme. As a consequence, it is necessary to introduce distance measures suitable for the comparison of probability density functions.

3.7.1 Fisher Discriminant

The Fisher Linear Discriminant is a tool developed in the area of pattern recognition in order to find the optimum line in terms of class separability. Suppose that there exists a set of n d-dimensional samples $x_1, \ldots, x_n$, in which $n_1 < n$ are in the subset $\mathcal{D}_1$ and are labelled w_1 and $n_2 < n$ are in the subset $\mathcal{D}_2$ and labelled as w_2. If we form a linear combination of the components of x, we obtain the scalar dot product (Duda et al., 2000):

$$y = w \cdot x, \tag{3.43}$$

which divides the corresponding set of of n samples $y_1, \ldots, y_n$ into subsets $\mathcal{Y}_1$ and $\mathcal{Y}_2$. Geometrically, if $\|w\| = 1$, each y_i is the projection of the corresponding x_i onto the line in the direction of w.

The Fisher Discriminant maximizes the criterion function

$$J(w) = \frac{|m_1 - m_2|^2}{s_1^2 + s_2^2} \tag{3.44}$$

over the linear function $w \cdot x$ to find the line that best separates both classes. In this case, m_i is the sample mean, and s_i^2 is the class variance.

In this work, the criterion presented in Equation 3.44 is used as a distance measure between estimated probability density functions, where m_i and s_i^2 are directly obtained from the probability density function estimates.

3.7.2 Kullback-Leibler Divergence

The Kullback-Leibler (KL) divergence is a measure of the distance between two probability measures. In statistics, it arises as an expected logarithm of the likelihood ratio (Cover and Thomas, 2006). In essence, the KL divergence measures the inefficiency of assuming that a given distribution is $q(x)$ when the true distribution is $p(x)$. In other words, the KL divergence quantifies the necessary amount of information that needs to be used to represent $p(x)$ through $q(x)$.

Definition 3.1 (Kullback-Leibler Divergence (Cover and Thomas, 2006)) *The* relative entropy *or* Kullback-Leibler distance *between two probability mass function $p(x)$ and $q(x)$ is defined as*

$$D(p(x)\|q(x)) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \tag{3.45}$$

$$= \mathbb{E}_{X \sim p} \left[\log \left(\frac{p(X)}{q(X)} \right) \right]. \tag{3.46}$$

Definition 3.1 uses the convention $0 \log \frac{0}{0} = 0$, $0 \log \frac{0}{q} = 0$, and $p \log \frac{p}{0} = \infty$. Thus, if there is any symbol $x \in \mathcal{X}$ such that $p(x) > 0$ and $q(x) = 0$, then $D(p(x)\|q(x)) = 0$.

3.8 Final Words

In this chapter, an introduction of the general concepts of Bayesian estimation frameworks has been presented. The Bayesian tracking problem has been introduced, and theoretical foundations for the understanding of the Particle Filter algorithms have been explained. Moreover, the development of the Classic Particle Filter algorithm has been shown, with usual implementation problems and possible solutions. Finally, some information has been given about the possibilities of information processing that the Particle Filter algorithm opens, and subsequently, two tools for this processing have been presented.

The exhibition and understanding of these tools open the path for the implementation details addressed in the following chapter.

Chapter 4

Implementation of Volatility Estimation using Particle Filters for a High Volatility Cluster Detection Scheme

The present chapter describes the implementation details followed to create a high volatility cluster detection scheme. In Chapter 2, it was mentioned that volatility is a non-observable variable from a given returns time series. Given this fact, to correctly implement, analyze, and assess a Bayesian filtering framework, it is necessary to use simulated data as ground truth.

The GARCH volatility model allows one to create a volatility time series -after defining some parameters- and consequently generating a returns series for the given volatility at every time step. This is useful to measure the effectiveness of estimation frameworks and detection schemes. The structure of this chapter includes a detailed description of the data sets that will be used in Section 4.1. Section 4.2, on the other hand, shows specific aspects about the use of Particle Filters for volatility estimation. Finally, the detection scheme is presented in Section 4.3.

4.1 Data

The assessment of Bayesian estimation frameworks and detection schemes requires data sets where observations and the state are known for every instant in a given period. This allows the evaluation of filtering schemes and consequent comparison of the implemented techniques. Given that the volatility of a returns series is a non-observable variable, it is mandatory to generate data sets where the algorithms can be tested and fine-tuned. This section provides details about artificially generated data used during this work, and presents the acquisition and post-processing necessary to apply the proposed algorithms in stock market data.

Table 4.1: GARCH(1,1) model parameters for each data set. The arrow ($\rightarrow$) indicates a change in the parameter value at time step 250. Note that parameters μ and ω are constant for each set.

Parameter \ Data set	GARCH0	GARCH1	GARCH2	GARCH3	GARCH4	GARCH5
μ	9×10^{-4}	9×10^{-4}	9×10^{-4}	9×10^{-4}	9×10^{-4}	9×10^{-4}
ω	10×10^{-6}	10×10^{-6}	10×10^{-6}	10×10^{-6}	10×10^{-6}	10×10^{-6}
α	$0.20 \rightarrow 0.10$	$0.20 \rightarrow 0.12$	$0.20 \rightarrow 0.14$	$0.20 \rightarrow 0.16$	$0.20 \rightarrow 0.18$	$0.20 \rightarrow 0.20$
β	$0.60 \rightarrow 0.85$	$0.60 \rightarrow 0.80$	$0.60 \rightarrow 0.75$	$0.60 \rightarrow 0.70$	$0.60 \rightarrow 0.65$	$0.60 \rightarrow 0.60$

4.1.1 Simulated Data

The simulated data has been generated using a GARCH(1,1) model, where parameters α and β are chosen in such a way that $\alpha + \beta$ is a value close to 1, as described in Section 2.2. In observed financial time series, it is not possible to assure that the values of the parameters α and β are fixed for a given time window. For this reason, volatility time series are created using time-dependent parameters over the studied time span. In particular, time series of 500 steps have been generated, with a parameter change in the step 250. This change resembles a regime shift in the market (Tobar, 2010).

The model used for data generation is:

$$\sigma^2_{k|k-1} = \omega + \alpha u^2_{k-1} + \beta \sigma^2_{k-1|k-2}, \tag{4.1}$$

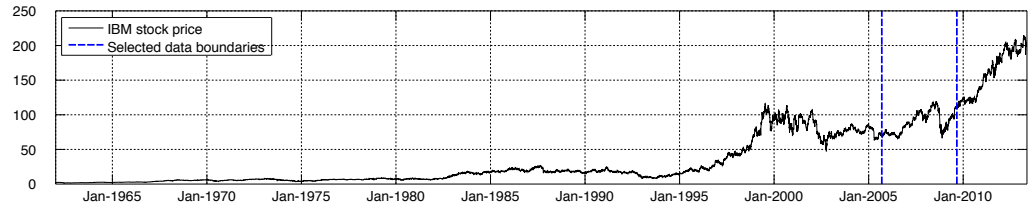
$$r_k = \mu + u_k, \tag{4.2}$$

where $\mu_{k|k-1} = \mu$ and ω are considered constant parameters in the considered time span, and $\sigma^2_0 = 0.5 \times 10^{-4}$.

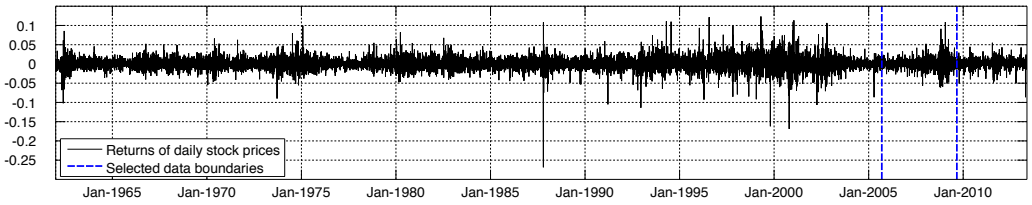
The implemented method for data generation creates data sets in which there are large volatility variations and small volatility variations. These are necessary for the correct assessment of the proposed detection algorithm.

4.1.2 Stock Market Data

The definition of volatility introduced in Chapter 2 considers the price time series of assets, which do not include stock market indexes such as *NASDAQ Composite*. In (Tsay, 2010), many of the examples for volatility concepts are given for the value of IBM stocks over time. For this reason, a section of IBM daily stock prices is used to apply the developed algorithm for early high volatility cluster detection. The data is extracted from (Yahoo!Finance, 2013), with information between January 1$^{\text{st}}$, 1962 and May 17$^{\text{th}}$, 2013 for a total of 12933 data points. The data considered for filtering is the *adjusted closing price*, which is commonly used for analysis of historical data. Figure 4.1 shows the data extracted from (Yahoo!Finance, 2013). Dashed lines have been used to show the data in which the proposed algorithm is applied, which corresponds to 1000 data points between September 12^{th}, 2005 and September 1^{st}, 2009. This data set includes the dramatic market fall occurred in October 2008 (see Figure 4.2).

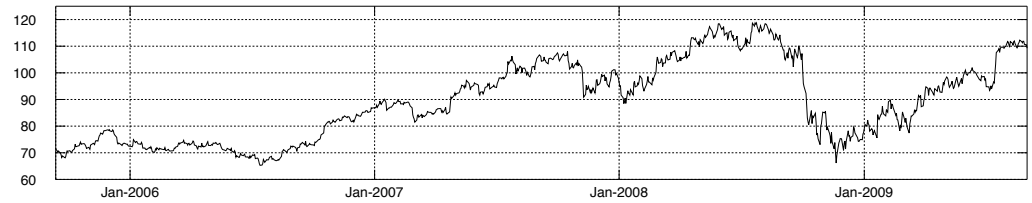


(a) Adjusted closing price.

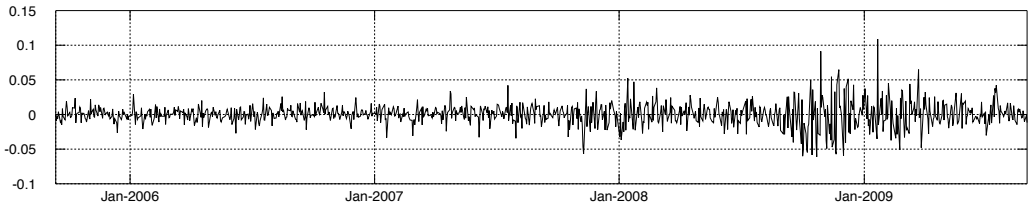


(b) Returns.

Figure 4.1: IBM's adjusted closing stock prices and corresponding daily returns from 1st, 1962 to May 17th, 2013. Notice the dashed lines from September 12th, 2005 to September 1st, 2009.



(a) Adjusted closing price.



(b) Returns.

Figure 4.2: IBM's adjusted closing stock prices and corresponding daily returns from September 12th, 2005 to September 1st, 2009. This figure represents the details between the dashed blue lines of Figures 4.1a and 4.1b.

4.2 Filtering Details

4.2.1 Estimation of Model Parameters

Model parameters α and β have a high impact on volatility series. These parameters have the power to drive the variation speed of a volatility series and to control the average of the series over time. Hence, it is of great importance to have good estimates of both parameters to adequately estimate volatility.

In financial time series, it is impossible to know if parameters α and β are fixed for a given time window in a data set. Therefore, it is necessary to estimate these parameters online. This work includes two stages of parameter estimation: Estimation through maximum likelihood in a training set and online estimation in test data points.

Estimation through Maximum Likelihood

Parameter estimation has been performed in both simulated data sets and stock market data through maximum likelihood. This is plausible due to the similar structure in both the GARCH(1,1) model and the uGARCH model. In particular, this task has been accomplished using the `garchfit` function available in the Financial Toolbox of `MATLAB`®.

In the simulated data sets, parameter estimation is performed using the first 150 steps for each returns time series. For IBM's stock prices, parameter estimation is performed over the first 200 time steps of the returns time series. These estimations are used as the initial conditions for online parameter estimation.

Online Estimation

Section 3.6 describes the reasons for using online parameter estimation in a Particle Filtering scheme. In this work, parameters α and β of the uGARCH model are allocated into an extended state vector,

$$(x_k)' = \begin{bmatrix} \alpha_k \\ \beta_k \\ \sigma_k^2 \end{bmatrix}, \tag{4.3}$$

where α_k and β_k are parameters considered to be time-variant, and are called pseudo-particles.

Maximum likelihood estimates α_0 and β_0 are used to compute the initial conditions α_1 and β_1, which include a random perturbation for every particle i:

$$\alpha_1^{(\mathrm{i})} = \mathcal{N}(\alpha_0, 0.1 \cdot \alpha_0), \tag{4.4}$$

$$\beta_1^{(\mathrm{i})} = \mathcal{N}(\beta_0, 0.1 \cdot \beta_0). \tag{4.5}$$

The initial conditions are used to drive the noise variance of the parameters in the extended state vector (see Equation 3.40):

$$\alpha_k^{(\mathrm{i})} \sim \mathcal{N}(\alpha_{k-1}^{(\mathrm{i})}, \alpha_1^{(\mathrm{i})}\sigma_{\alpha,\beta}^2) \tag{4.6}$$

$$\beta_k^{(\mathrm{i})} \sim \mathcal{N}(\beta_{k-1}^{(\mathrm{i})}, \beta_1^{(\mathrm{i})}\sigma_{\alpha,\beta}^2) \tag{4.7}$$

There are two major drawbacks with this method:

- $\mathbb{P}[\alpha_k^{(\mathrm{i})} < 0] > 0$ and $\mathbb{P}[\beta_k^{(\mathrm{i})} < 0] > 0$, $\forall \mathrm{i}, k$,
- $\mathbb{P}[\alpha_k^{(\mathrm{i})} + \beta_k^{(\mathrm{i})} > 1] > 0$, $\forall \mathrm{i}, k$,

both of which are not permitted in the uGARCH model. In particular, they have been addressed in the following way:

- For each $\alpha_k^{(\mathrm{i})} < 0$, let $\alpha_k^{(\mathrm{i})} = 10^{-5}$. Similarly, for each $\beta_k^{(\mathrm{i})} < 0$, let $\beta_k^{(\mathrm{i})} = 10^{-5}$.
- The Particle Filter self-regulates from the cases where $\alpha_k^{(\mathrm{i})} + \beta_k^{(\mathrm{i})} > 1$, provided that these cases have very low likelihood, which translate into very low values of corresponding weights. Hence, no saturation condition has been used for the upper bound.

4.2.2 Particle Filters

Volatility estimation in both simulated data and stock market data has been performed using two different Particle Filtering schemes, including classic and risk sensitive approaches. The estimation is performed using 100 particles. Due to the inherent randomness of the filtering processes, these are repeated 10 times. Also, both filters are implemented with a resampling stage, where residual resampling is used. Next, details about each particular filter are presented.

Classic Particle Filter

The Classic Particle Filter (PF) for volatility estimation in the uGARCH model has been implemented using an importance density equal to

$$q_{PF}(x_k|x_{k-1}^{(\mathrm{i})}, z_k) = p(\sigma_k^2|\sigma_{k-1}^{2(\mathrm{i})}). \tag{4.8}$$

A closed Expression for $p(\sigma_k^2|\sigma_{k-1}^{2(\mathrm{i})})$ has been given in Equation 2.15. Thus, samples are generated according to:

$$\sigma_k^{(\mathrm{i})} \sim p(\sigma_k^2|\sigma_{k-1}^{2(\mathrm{i})}), \tag{4.9}$$

which leads to the following weight update equation:

$$w_k^{(\mathrm{i})} = w_{k-1}^{2(\mathrm{i})} p(r_k|\sigma_k^{2(\mathrm{i})}). \tag{4.10}$$

Risk Sensitive Particle Filter

In the search for an importance density function that could be used to propose a risk sensitive approach towards volatility estimation, it was necessary to find a distribution with very specific characteristics. First, the probability density function needs a localization parameter that lets both the Classic Particle Filter's (PF) and Risk Sensitive Particle Filter's (RSPF) importance density have the same support. Second, the RSPF's importance density should have a fatter tail than the PF's density. The proposed RSPF uses the Generalized Pareto Distribution as the importance density function, which is commonly used to model the tails of other distributions; since it is able to model exponential, polynomial and even finite tails.

Definition 4.1 (Generalized Pareto Distribution (Embrechts et al., 1997; Mathworks, 2013)) *The probability density function of the Generalized Pareto Distribution (GPD) is defined by*

$$f_{GPD}(x|k,\sigma,\theta) = \begin{cases} \frac{1}{\sigma}\left[1+k\frac{x-\theta}{\sigma}\right]^{-1-\frac{1}{k}} & \text{if } \begin{cases} k>0, \text{ for } \theta<x \\ k<0, \text{ for } \theta<x<\sigma/k \end{cases} \\ \frac{1}{\sigma}\exp\left[-\frac{x-\theta}{\sigma}\right] & \text{if } \quad k=0, \text{ for } \theta<x \end{cases} \tag{4.11}$$

This distribution has two special cases, where it is reduced to other distributions:

- *If $k=0$ and $\theta=0$, the generalized Pareto distribution is equivalent to the exponential distribution.*
- *If $k>0$ and $\theta=\sigma/k$, the generalized Pareto distribution is equivalent to the Pareto distribution.*

As Definition 4.1 states, the probability density function of the GPD has three parameters. These can be interpreted as follows:

- k: Shape parameter,
- σ: Scale parameter,
- θ: Threshold (location) parameter.

These parameters cannot take any value if one wants to assure the convergence of the first and second moments of the the GDP, since

$$\begin{aligned} \mathbb{E}[X] &= \theta + \frac{\sigma}{1+k}, \text{ for } k<1, \\ \mathbb{V}ar[X] &= \frac{\sigma^2}{(1-k)^2(1-2k)}, \text{ for } k<1/2, \end{aligned} \tag{4.12}$$

Considering that the variance is defined for $k<1/2$, the parameters of the probability density function of the GPD have been used in the following way to utilize it as the importance density of the RSPF:

$$\boxed{\begin{aligned} &k=0.49, \\ &\sigma=0.3\sigma_{k-1}^{2(\mathrm{i})}, \\ &\theta=\omega+\beta_{k-1}^{(\mathrm{i})}\sigma_{k-1}^{2(\mathrm{i})}, \end{aligned}} \tag{4.13}$$

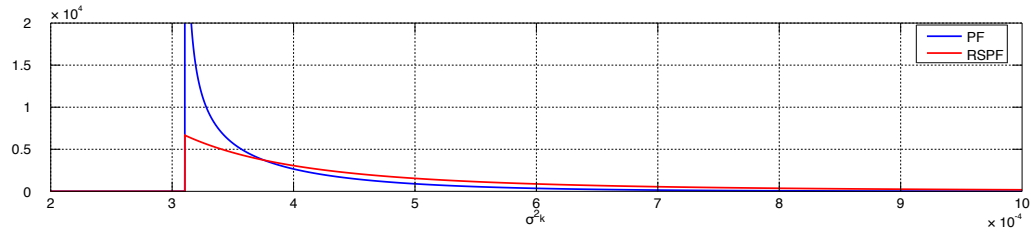


Figure 4.3: Comparison between $q_{PF}(\sigma_k^2|\sigma_{k-1}^{2(\mathrm{i})})$ and $q_{RSPF}(\sigma_k^2|\sigma_{k-1}^{2(\mathrm{i})})$. In this example, $\sigma_{k-1}^{2(\mathrm{i})} = 5 \times 10^{-4}, \alpha = 0.2, \beta = 0.6, \omega = 1.0468 \times 10^{-5}$, and $\sigma_\eta^2 = 0.7$.

where $\beta_k^{(\mathrm{i})}$ is the i^{th} pseudo-particle for the online estimation of the uGARCH parameter β. Parameter k has been fixed in the aforementioned value to reproduce the shape of $p(\sigma_k^2|\sigma_{k-1}^2)$ (see Equation 2.15). Parameter σ gives the scale to $f_{GPD}(x|k,\sigma,\theta)$. Given that $\max\{f_{GPD}(x|k,\sigma,\theta)\} = 1/\sigma$, using a scaled previous-step particle, a desired fat tail with similar shape to $p(\sigma_k^2|\sigma_{k-1}^2)$ is obtained. Parameter θ sets the location of the density of the GPD and is set to be equivalent to $\omega + \beta_{k-1}^{(\mathrm{i})}$ (see Equation 2.12), this is, the support of $f_{GPD}(x|k,\sigma,\theta)$ is set to be equivalent to the support of $p(\sigma_k^2|\sigma_{k-1}^2)$. Hence, the importance density function employed is

$$q_{RSPF}(x_k|x_{k-1}^{(\mathrm{i})}, z_k) = f_{GPD}(\sigma_k^2|0.49; 0.3\sigma_{k-1}^{2(\mathrm{i})}; \omega + \beta_{k-1}^{(\mathrm{i})}\sigma_{k-1}^{2(\mathrm{i})}). \tag{4.14}$$

Particles are drawn from

$$\sigma_k^{2(\mathrm{i})} \sim f_{GPD}(\sigma_k^2|0.49; 0.3\sigma_{k-1}^{2(\mathrm{i})}; \omega + \beta_{k-1}^{(\mathrm{i})}\sigma_{k-1}^{2(\mathrm{i})}), \tag{4.15}$$

and the weight update equation is

$$w_k^{(\mathrm{i})} = w_{k-1}^{(\mathrm{i})} \frac{p(r_k|\sigma_k^{2(\mathrm{i})})p(\sigma_k^{(\mathrm{i})}|\sigma_{k-1}^{(\mathrm{i})})}{f_{GPD}(\sigma_k^{2(\mathrm{i})}|0.49; 0.3\sigma_{k-1}^{2(\mathrm{i})}; \omega + \beta_{k-1}^{(\mathrm{i})}\sigma_{k-1}^{2(\mathrm{i})})}. \tag{4.16}$$

A visual comparison of $q_{PF}(x_k|x_{k-1}^{(\mathrm{i})}, z_k)$ and $q_{RSPF}(x_k|x_{k-1}^{(\mathrm{i})}, z_k)$ is shown in Figure 4.3. Notice that both importance densities are defined over the same support, and $q_{RSPF}(x_k|x_{k-1}^{(\mathrm{i})}, z_k)$ has a fatter tail than $q_{PF}(x_k|x_{k-1}^{(\mathrm{i})}, z_k)$. Hence, the design conditions for the RSPF's importance density are met.

4.3 Detection

4.3.1 Detection through Hypothesis Test

Section 3.4 describes that Particle Filters consist mainly of two steps: prediction and update. The Particle Filter, such as any other Bayesian filtering framework, predicts through model dynamics (Expression 3.1), and updates the estimation with the new measurement. Therefore, for every time step, the Particle Filter produces a prior estimate and a posterior

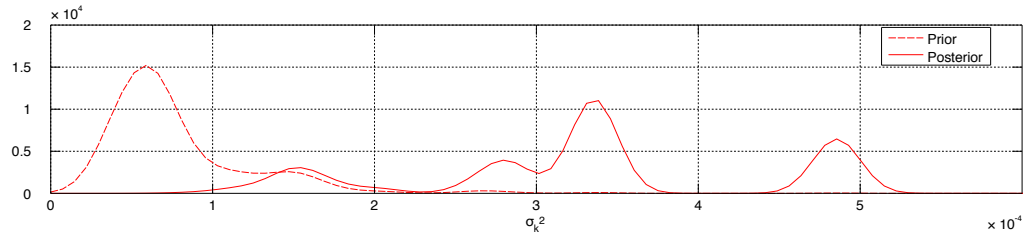


Figure 4.4: Examples of prior and posterior densities of the RSPF in a volatility filtering process.

estimate:

$$\text{Prior:} \quad \hat{p}(x_k|z_{1:k-1}) = \sum_{\mathrm{i}=1}^{N_s} w_{k-1}^{(\mathrm{i})} \delta(x_k - x_k^{(\mathrm{i})}), \tag{4.17}$$

$$\text{Posterior:} \quad \hat{p}(x_k|z_{1:k}) = \sum_{\mathrm{i}=1}^{N_s} w_{k}^{(\mathrm{i})} \delta(x_k - x_k^{(\mathrm{i})}), \tag{4.18}$$

Differences between prior and posterior densities may be considerable if model dynamics diverge from actual measurements. This is the case when unlikely events such as unexpected market falls or machine ruptures occur. Figure 4.4 shows the vast differences that may occur between prior and posterior density estimates. A detection scheme through hypothesis test exploits these differences to design rapid change detectors in the estimated state.

The designed test considers the following hypotheses:

- $\mathcal{H}_0$: A new high volatility cluster is starting.
- $\mathcal{H}_1$: There is no presence of new high volatility clusters.

To accept or reject the null hypothesis $\mathcal{H}_0$, the implemented test considers the 70% confidence interval of the prior density, and contrasts it with the mean of the posterior density. The confidence interval is calculated using Parzen windows (Principe, 2010) and a Normal kernel, whose bandwidth σ_{kernel} is obtained through Silverman's thumb rule (Principe, 2010). In particular, the following steps have been implemented for every time step k:

1. Calculate the maximum of the particle cloud: $maximum = \max\{\sigma_k^{2(\mathrm{i})}\}$.
2. Obtain 100 support points equivalently spaced between $[0, 1.5 \cdot maximum]$. The length of the support is increased in a 50% to correctly model the tails of the distributions.
3. Find the likelihoods for every point in the newly defined support through Parzen windows.
4. Obtain the support point where the 70% of the total mass of the density is covered.

If the mean of the posterior density is greater than the 70% interval bound of the prior density, the null hypothesis is accepted. Figure 4.5 shows an example of the designed hypothesis test, where an unlikely event occurs and the null hypothesis $\mathcal{H}_0$ is accepted.

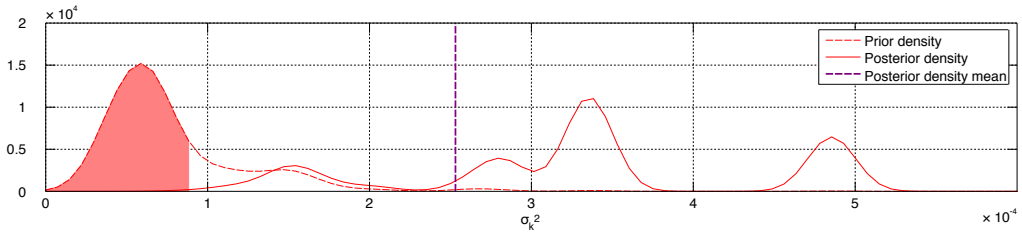


Figure 4.5: Hypothesis test example. The filled area under the smoothed prior density represents the 70% confidence interval for the smoothed prior density. Here, the null hypothesis is accepted.

4.3.2 Detection using the Fisher Discriminant

The previously presented framework involves volatility estimates from both Classic and Risk Sensitive Particle Filtering approaches. In particular, one obtains

$$p_{PF}(\sigma_k^2|r_k) = \sum_{\mathrm{i}=1}^{N_s} w_k^{(\mathrm{i})} \delta(\sigma_k^2 - \sigma_k^{2(\mathrm{i})}), \tag{4.19}$$

$$p_{RSPF}(\sigma_k^2|r_k) = \sum_{\mathrm{i}=1}^{N_s} w_k^{(\mathrm{i})} \delta(\sigma_k^2 - \sigma_k^{2(\mathrm{i})}). \tag{4.20}$$

The Fisher criterion is used to calculate the distance between the two weighted clouds of particles for every time step k of the filtering process as a function of the first two moments of the probability density functions.

4.3.3 Detection using the Kullback-Leibler Divergence

The Kullback-Leibler Divergence defined in Section 3.7.2 considers an equivalent support $\mathcal{X}$ for densities $p(x)$ and $q(x)$. Given that the supports for the posterior distributions $p_{PF}(\sigma_k^2|r_k)$ and $p_{RSPF}(\sigma_k^2|r_k)$ are obtained through Monte Carlo simulations, the KL divergence cannot be calculated between Particle Filter estimates. Moreover, the marginal densities $p(\sigma_k^2)$ and $p(r_k)$ cannot be calculated, so a theoretically accurate computation of the conditional densities cannot be introduced for the calculation of the KL divergence.

To calculate an approximation of this measure, Parzen windows are used to obtained a smooth version of the estimates $p_{PF}(\sigma_k^2|r_k)$ and $p_{RSPF}(\sigma_k^2|r_k)$ over a previously defined common support. The method includes the following steps for every k in the filtering window:

1. Calculate $maximum = \max\{\sigma_k^{2(\mathrm{i})}(PF), \sigma_k^{2(\mathrm{i})}(RSPF)\}$.
2. Obtain 100 support points equivalently spaced between $[0, 1.5 \cdot maximum]$. The length of the support Σ has been increased in a 50% to correctly model the tails of the distributions.
3. Find the likelihoods for every point in the newly defined support.

As in the case of the calculation of confidence intervals, Silverman's thumb rule has been used to obtain a reasonable bandwidth σ_{kernel}.

The used approximation is

$$D(p_{PF}(\sigma_k^2|r_k)\|p_{RSPF}(\sigma_k^2|r_k)) \approx \sum_{\sigma^2\in\Sigma} p_{PF}(\sigma_k^2|r_k)\log\frac{p_{PF}(\sigma_k^2|r_k)}{p_{RSPF}(\sigma_k^2|r_k)}. \tag{4.21}$$

Chapter 5

Results

This chapter describes the results obtained for volatility estimation using Particle Filters, and detection of high risk events using information derived from the filtering process. During the training stage, parameter selection for the setup of the PF algorithm is achieved through a sensibility analysis and subsequent selection through the smallest associated estimation error (in percentage). These results are used to select the PF algorithm parameters to be utilized in the detection scheme, where estimation through PF is crucial. After the parameter selection stage, results for three different detection approaches are presented.

Performance measures introduced in this chapter may only be used in simulated data, where the true volatility is known. Hence, quantitative results showing performance measures results are presented for simulated data, and qualitative results are presented for returns series derived from IBM stock prices.

The chapter is organized as follows. First, the performance measure used for Particle Filter estimates is presented, and an evaluation scheme for the detection framework is proposed. Filtering results are then introduced, which include parameter selection. Finally, detection results using the hypothesis test proposed in the previous chapter and distance measures for probability density functions are shown.

5.1 Performance measures

Chapters 2 and 4 describe the non-observability property of volatility. In this regard, this section presents the steps followed to give a possible solution for this problem. Performance measures described in this section assume knowledge of the ground truth data, and as a consequence, results may be analyzed only in simulated data. The following sections consider $\hat{\sigma}_k^2$ as the estimated volatility and σ_k^2 as the *true* volatility (this is, the ground truth).

5.1.1 Accuracy Indicator for Particle Filter estimates

Performance of Particle Filters, including the Classic and Risk Sensitive approaches, is compared in terms of error (in percentage). The performance indicator is defined as follows:

$$\mathrm{i}^{EX}(k) = \frac{|\hat{\sigma}_k^2 - \sigma_k^2|}{\sigma_k^2} \cdot 100. \tag{5.1}$$

Given that $\mathrm{i}^{EX}(k)$ is defined for every time step k of the filtering process, one can obtain an average of $\mathrm{i}^{EX}(k)$ over the filtering time window $T_\mathrm{i}, \ldots, T_f$:

$$I^{EX} = \frac{1}{T_f - T_\mathrm{i}} \sum_{k=T_\mathrm{i}}^{T_f} \mathrm{i}^{EX}(k) = \frac{1}{T_f - T_\mathrm{i}} \sum_{k=T_\mathrm{i}}^{T_f} \frac{|\hat{\sigma}_k^2 - \sigma_k^2|}{\sigma_k^2} \cdot 100. \tag{5.2}$$

Furthermore, given that I^{EX} is defined only for one filtering process, one can obtain an average of I^{EX} over the amount of realizations of the filtering process, which include 10 in this work:

$$\bar{I}^{EX} = \frac{1}{10} \sum_{n=1}^{10} I^{EX}(n). \tag{5.3}$$

In particular, index $\bar{I}^{EX}$ serves as the base to compare the error (in percentage) for each filtering process, for each set of parameters. Hence, the best set of parameters is obtained by observing the smaller index $\bar{I}^{EX}$.

5.1.2 Detection Performance

The detection scheme based on the hypothesis test approach that was proposed in the previous chapter has as an output indicatrix function that shows the time instants where volatility starts to increase. To correctly measure the effectiveness of this algorithm, it is necessary to label the time steps in which there is a *considerable* increase in true volatility, which is in essence subjective. To solve this problem, the one-step net differences of the ground truth volatility data series have been calculated. This first step lets one obtain direct information regarding increments in volatility. Next, a histogram is built with the volatility net differences obtained in the training interval, which accumulates the statistics of volatility increases. One can define a threshold using the value where the X% of mass has been obtained. This threshold defines over the time series a limit for volatility increases that are considerable, and lets one create the detection ground truth to which the detector performance is contrasted.

Detection results are presented using confusion matrixes, and the following statistical measures are calculated:

- Positive Predictive Value $\triangleq \frac{TP}{TP+FP}$ is the proportion of positive results that are correctly detected,
- Negative Predictive Value $\triangleq \frac{TN}{FN+TN}$ is the proportion of negative results that are correctly detected,

- Sensitivity $\triangleq \frac{TP}{TP+FN}$ is the proportion of true positive results from all positive results,
- Specificity $\triangleq \frac{FP}{FP+TN}$ is the proportion of true negative results out of all negative results,
- Accuracy $\triangleq \frac{TP+TN}{TP+TN+FP+FN}$ is the proportion of all correctly classified results and all results.

In all these definitions, TP stands for the number of true positives, FP stands for the number of false positives, FN stands for the number of false negatives and TN stands for the number of true negatives.

5.2 Estimation through Particle Filters

5.2.1 Fitting GARCH Parameters

The first step towards a filtering process starts with the estimation of model parameters that are used as the initial conditions in the extended GARCH model (refer to 4.2.1). As previously mentioned, for the uGARCH model this can be achieved by maximum likelihood, assuming that the model is in fact a GARCH model.

Table 5.1 shows the results of parameter estimation for the simulated data sets. The parameters are obtained using the `garchfit` function of MATLAB® over the training window of each data set. The true values for this time window are $\alpha = 0.2$ and $\beta = 0.6$ for every data set. From Table 5.1, GARCH1 is the data set that obtains the poorest parameter estimates from data contained within the training window.

This estimation has direct incidence over the filtering process, since these values are used as initial conditions for the extended uGARCH model, where the dynamics are non-observable. Initial conditions in non-observable systems are of great importance in the outcome of the a Bayesian filtering process. If the system is non-observable, the state may follow one of an infinite number of possible paths that match the current observations. Therefore, accurate initial conditions are necessary to achieve an unbiased estimate of the state.

Table 5.1: Parameter estimation using MATLAB®'s `garchfit` function from the Financial Toolbox.

Parameter \ Data set	GARCH0	GARCH1	GARCH2	GARCH3	GARCH4	GARCH5
α	0.2094	0.1477	0.2194	0.2612	0.2903	0.1149
β	0.6307	0.2768	0.5682	0.6699	0.5237	0.6533
$\alpha + \beta$ (< 1)	0.8401	0.4245	0.7876	0.9311	0.8140	0.7682

5.2.2 Particle Filter Parameter Selection

This subsection presents the results of the sensibility analysis of parameters of the Classic Particle Filter. These parameters have been tested to find the combination that minimizes the estimation error $\bar{I}^{EX}$. The tested parameters include R_{th} (resampling threshold), $\sigma_{\alpha,\beta}$ (pseudo-particle standard deviation), and σ_η (process noise).

To find the parameter values that minimize the estimation error, the following parameter mesh is used:

- $R_{th} = \{0.5, 0.6, 0.7\}$,
- $\sigma_{\alpha,\beta} = \left\{ \begin{array}{lllll} 0.0010, & 0.0015, & 0.0020, & 0.0025, & 0.0030 \\ 0.0035, & 0.0040, & 0.0050, & 0.0075, & 0.0100 \\ 0.0125, & 0.0150, & 0.0175, & 0.0200, & 0.0225 \\ 0.0250, & 0.0275, & 0.0300, & 0.0350, & 0.0400 \end{array} \right\}$,
- $\sigma_\eta = \{0.5, 0.6, 0.7\}$.

It should be noted that the selected values that are employed to create the mesh for $\sigma_{\alpha,\beta}$ were placed at irregular intervals. Since there is a tendency to have better estimations with lower values of $\sigma_{\alpha,\beta}$, a better resolution has been given to the interval of smaller values.

Every set of parameters is used to run 10 times each filtering process over the complete time window $T = \{1, \ldots, 500\}$ of every set of simulated data (GARCH0 to GARCH5). The error is computed over the interval $T' = \{151, \ldots, 500\}$, which excludes the training interval.

Tables 5.2 through 5.7 show the percentage error for the mean of the 10 filtering routines for each set of parameters. Since results are 3-dimensional, the tables show the results for $\sigma_\eta = 0.7$, which is the noise process value that minimizes the error for every data set.

The first thing to note in Tables 5.2 to 5.7 is that the minimum percentage error for the filtering process on these data sets is bounded approximately between 16% and 30%. Second, it is very important to notice that errors are very similar for each one of the columns of the tables. This means that the resampling threshold R_{th} has a limited impact on the estimates when contrasted to the ground truth from an accuracy perspective. This is very important since one can simply employ an average of R_{th} over all data sets without loosing estimation accuracy, or simply select the value that is most often the best value. Also, data shows that minimization occurs over a convex space which lets one assume that there is in fact a set of parameters which minimize the estimation error. These results also demonstrate that higher noises (this is, greater particle variability) do not translate into better estimates. In fact, there is a small subset of the parameter space where Bayesian filters such as the Particle Filter may work properly.

Table 5.8 shows the summary of selected parameters for each data set and its arithmetic mean, calculated using the information for every data set. This table clearly shows that the data sets GARCH0, GARCH2 and GARCH3 conduce to similar parameter values when minimizing the estimation error; in fact, the values for the parameter $\sigma_{\alpha,\beta}$ are in the same scale. On the other hand, data sets GARCH1, GARCH4 and GARCH5 introduce a major

Table 5.2: Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH0. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0075$, $R_{th} = 0.6$.

		$\sigma_{\alpha,\beta}$									
		0.0010	0.0015	0.0020	0.0025	0.0030	0.0035	0.0040	0.0050	0.0075	0.0100
R_{th}	0.5	45.6831	42.6858	39.8083	36.6233	33.045	34.0999	32.5206	30.8759	32.5593	32.3207
	0.6	45.3273	43.9501	39.2128	37.0836	35.0476	32.6049	32.5007	30.9371	30.5472	34.6344
	0.7	44.9466	44.6572	40.4360	37.7934	34.2450	33.0747	32.4398	31.5625	31.222	30.9074

		$\sigma_{\alpha,\beta}$									
		0.0125	0.015	0.0175	0.02	0.0225	0.025	0.0275	0.03	0.035	0.04
R_{th}	0.5	33.1869	33.2365	35.4018	35.4773	35.0612	36.7213	36.0474	38.4826	40.5602	39.4100
	0.6	33.1350	33.0238	34.4899	33.2348	33.7765	35.4858	37.0154	39.1493	38.0483	41.2012
	0.7	30.7984	34.9421	35.6370	34.6434	34.2616	35.1727	35.4310	38.5281	37.9884	40.6167

Table 5.3: Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH1. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0350$, $R_{th} = 0.6$.

		$\sigma_{\alpha,\beta}$									
		0.0010	0.0015	0.0020	0.0025	0.0030	0.0035	0.0040	0.0050	0.0075	0.0100
R_{th}	0.5	51.0611	51.2075	50.1367	50.1331	47.4300	48.0407	46.5103	44.3850	38.7528	30.2866
	0.6	51.4037	50.8755	50.6492	49.4681	49.1941	47.3560	47.2149	45.0067	36.1592	31.3635
	0.7	51.4245	50.8909	50.8915	49.6318	48.5846	47.6203	47.2209	45.2000	36.7997	30.1693

		$\sigma_{\alpha,\beta}$									
		0.0125	0.015	0.0175	0.02	0.0225	0.025	0.0275	0.03	0.035	0.04
R_{th}	0.5	26.1898	23.0201	21.7539	21.6545	21.4255	20.5647	20.3163	19.9770	19.7330	19.9307
	0.6	27.4839	25.0151	22.8430	21.6427	20.7188	21.0935	20.1355	19.7223	19.4162	20.1503
	0.7	27.4139	24.1124	22.9076	21.5613	21.4261	20.0818	20.5648	20.6290	19.9582	20.0205

Table 5.4: Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH2. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0035$, $R_{th} = 0.5$.

		$\sigma_{\alpha,\beta}$									
		0.0010	0.0015	0.0020	0.0025	0.0030	0.0035	0.0040	0.0050	0.0075	0.0100
R_{th}	0.5	34.3502	33.6726	30.2761	25.4804	26.3638	22.9285	24.0401	25.2195	26.1182	27.6950
	0.6	37.5176	34.6301	29.5473	27.0985	24.4056	24.1667	23.2317	24.6105	26.1960	28.6836
	0.7	35.7616	32.7430	30.3254	25.2158	23.6773	23.4249	23.9162	23.0610	27.1241	28.2041

		$\sigma_{\alpha,\beta}$									
		0.0125	0.015	0.0175	0.02	0.0225	0.025	0.0275	0.03	0.035	0.04
R_{th}	0.5	28.1978	29.1794	32.9497	31.5485	32.0718	34.6276	34.7882	36.1947	40.2270	41.2393
	0.6	27.6064	28.2799	29.7254	31.1746	32.4882	33.5473	33.9458	35.5584	37.3276	40.2381
	0.7	27.6898	28.2368	31.0338	31.9597	31.7439	35.0622	35.9027	33.3601	36.5086	41.3087

Table 5.5: Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH3. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0035$, $R_{th} = 0.7$.

		$\sigma_{\alpha,\beta}$									
		0.0010	0.0015	0.0020	0.0025	0.0030	0.0035	0.0040	0.0050	0.0075	0.0100
R_{th}	0.5	19.3664	20.0746	17.8894	17.7761	18.0384	17.3640	16.3491	18.7394	19.4852	22.1763
	0.6	18.1588	17.5985	17.4676	17.9119	17.2962	17.4517	18.1153	17.9320	19.3590	20.3719
	0.7	19.7638	18.9000	18.2602	18.5192	17.4313	16.1627	17.2223	18.4436	18.9302	21.1907

		$\sigma_{\alpha,\beta}$									
		0.0125	0.015	0.0175	0.02	0.0225	0.025	0.0275	0.03	0.035	0.04
R_{th}	0.5	24.1930	26.4535	27.9966	31.0447	32.1416	33.1691	35.9803	38.5001	38.3597	41.2628
	0.6	24.5801	25.5974	29.5183	31.5754	32.1902	32.8693	37.5007	37.0170	39.5580	42.0956
	0.7	24.3899	26.4036	27.6496	29.5650	35.0783	31.5716	37.9800	35.0379	39.5324	43.3656

Table 5.6: Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH4. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.0150$, $R_{th} = 0.7$.

		$\sigma_{\alpha,\beta}$									
		0.0010	0.0015	0.0020	0.0025	0.0030	0.0035	0.0040	0.0050	0.0075	0.0100
R_{th}	0.5	20.3345	19.4469	19.5720	19.9932	19.1090	19.1984	19.1401	19.5118	19.1902	19.3141
	0.6	20.7905	19.9680	20.1165	20.6015	19.2363	19.0756	18.9305	19.5775	19.1778	19.3428
	0.7	20.2243	19.7580	19.7437	19.2649	19.0240	19.2841	18.7612	19.3580	19.5441	19.2978

		$\sigma_{\alpha,\beta}$									
		0.0125	0.015	0.0175	0.02	0.0225	0.025	0.0275	0.03	0.035	0.04
R_{th}	0.5	18.9925	19.0681	19.3370	20.2175	19.5165	20.0991	20.9024	21.6774	22.6824	23.5475
	0.6	19.5260	19.4372	19.9382	19.8127	19.2165	19.9946	20.8392	20.8295	21.2633	22.8528
	0.7	19.5867	18.7214	19.5339	19.8361	19.1170	19.9311	19.7897	22.1874	21.7869	21.8020

Table 5.7: Estimation error $\bar{I}^{EX}$ for different parameter values for data set GARCH5. This table shows the results for $\sigma_\eta = 0.7$. The minimum error (shaded in gray) is obtained for $\sigma_{\alpha,\beta} = 0.02$, $R_{th} = 0.6$.

		$\sigma_{\alpha,\beta}$									
		0.0010	0.0015	0.0020	0.0025	0.0030	0.0035	0.0040	0.0050	0.0075	0.0100
R_{th}	0.5	31.9090	32.5511	33.8042	32.3486	33.4068	34.2595	33.6950	33.3682	33.0455	32.0685
	0.6	31.7091	32.0966	32.1253	32.6127	33.1701	33.0996	33.2717	31.6672	30.9641	31.2875
	0.7	31.5928	32.4281	31.2327	33.4183	32.7725	32.2182	32.6587	33.3244	32.4323	32.9893

		$\sigma_{\alpha,\beta}$									
		0.0125	0.015	0.0175	0.02	0.0225	0.025	0.0275	0.03	0.035	0.04
R_{th}	0.5	30.5636	31.8657	31.2062	30.6622	30.7736	30.1174	30.6500	31.6419	32.5699	32.3242
	0.6	30.9969	31.7978	30.5992	29.7220	31.4580	31.4766	30.9374	31.5223	31.9582	33.3153
	0.7	31.4595	30.6166	30.4844	30.6546	31.1387	30.7421	31.8942	30.9111	30.6593	33.3628

Table 5.8: Summary of sensibility analysis for the Classic Particle Filter

Parameter \ Data set	GARCH0	GARCH1	GARCH2	GARCH3	GARCH4	GARCH5	Mean
R_{th}	0.6	0.7	0.5	0.7	0.7	0.6	0.63
$\sigma_{\alpha,\beta}$	0.0075	0.0350	0.0035	0.0035	0.0150	0.0200	0.0141
σ_{η}	0.7	0.7	0.7	0.7	0.7	0.7	0.7
Minimum error $\bar{I}^{EX}$	30.5472	19.4162	22.9285	16.1627	18.7214	29.7720	22.9247
$\alpha + \beta$ (< 1)	0.8401	0.4245	0.7876	0.9311	0.8140	0.7682	0.7609

difference in this parameter value (there is a difference of one order of magnitude). Careful examination of the rows and columns of Table 5.6 (GARCH4) and Table 5.7 (GARCH5) shows that the error index $\bar{I}^{EX}$ is relatively constant for the parameters $\sigma_{\alpha,\beta}$ and R_{th} for these data sets. Nevertheless, this behavior is not present in Table 5.3 (GARCH1). Error values increase hugely towards the left side of the columns of Table 5.3. This most probably occurs due to the poor estimation of initial conditions through maximum likelihood. Since the initial conditions are far from the ideal values, more variability is needed in the artificial evolution equations included within the Particle Filter algorithm in order to effectively learn and find the correct intervals where these parameters lie.

The inherent non-observability issues of volatility imply that using an average value for $\sigma_{\alpha,\beta}$ over all the data sets where the filtering process is applied will result in poor estimations for the data set GARCH1.

To choose specific parameter values R_{th}, $\sigma_{\alpha,\beta}$ and σ_{η}, it is necessary to consider that parameters R_{th} and σ_{η} have a very small incidence in the estimation error given the parameter mesh. Thus, both of these parameters are set to 0.7. For parameter $\sigma_{\alpha,\beta}$, if one considers the arithmetic mean, results for the GARCH1 data set are far from optimum. Nevertheless, this is the proposed value used in the detection scheme. As a summary, the values considered for the proposed detection algorithms are the following:

$$\boxed{\begin{aligned} R_{th} &= 0.6, \\ \sigma_{\alpha,\beta} &= 0.0141, \\ \sigma_{\eta} &= 0.7. \end{aligned}} \tag{5.4}$$

5.2.3 Estimation Results

Filtering results with optimum parameters for each data set

This subsubsection presents the results obtained for volatility estimation using the optimum parameters for the Classic Particle Filtering processes. These parameters have also been applied and used in the RSPF. Results are shown in Figures 5.1 to 5.6.

Analysis of the estimation performance in each data set uncovers many interesting findings that need to be addressed. The comments about results are discussed separately for every data set.

- GARCH0 (Figure 5.1): In this data set, parameters α and β of the GARCH model change dramatically at time step 250. This regime shift introduces a particular difficulty for the learning process of the Particle Filters, since adaptation needs to occur throughout the window. In particular, low volatility periods (period between time steps 150 and 250, except for the high volatility cluster) are not tracked correctly, given that only the trend is followed (slow reaction to changes). There is a small reaction towards tracking the high volatility cluster around time step 200, but the result is very biased. The behavior of the filtering scheme changes vastly after time step 300, where the filter is able to adapt. Estimation between time steps 350 and 500 has notable results.
- GARCH1 (Figure 5.2): Similar to data set GARCH0, there is a large change in the parameters α and β for the simulated data at time step 250. Up to time step 250, both the PF and RSPF are only able to track the trend of the volatility curve, but there is no reaction to sudden changes. This behavior changes in time step 250, where there is a tendency towards capturing rapid volatility changes. The filter demonstrates the results of the learning process at time step 290, where a hefty volatility cluster occurs. There is correct tracking of volatility shape with a very small estimation bias. This occurs between time steps 290 and 500, which corresponds to the end of the time window.
- GARCH2 (Figure 5.3): This filtering process presents extremely low quality estimation results, with biased estimations and deficient tracking in sudden state variations. As a matter of fact, employed noises are too small for the Particle Filters to learn in the first 100 time steps of the filtering process. The first big volatility cluster after time step 150 is somewhat correctly tracked, but between this event and time step 300, only trends are followed and there is poor rapid change tracking. The performance of these algorithms rises after time step 300, where variations of volatility are more efficiently followed. In fact, estimation bias diminishes considerably towards the end of the filtering window.
- GARCH3 (Figure 5.4): There is excellent filtering performance throughout the time window. Accurate estimation, excellent shape tracking. For this data set, only about 100 time steps are necessary for the algorithm to learn and adapt.
- GARCH4 (Figure 5.5): Similar to data set GARCH3, very good state estimation, with accurate results and very good tracking of sudden volatility changes. The training window is enough time to learn from the data.
- GARCH5 (Figure 5.6): This data set presents the poorest estimation results among the 6 data sets presented. Both the PF and the RSPF are not able to learn from the data, and only trend tracking is possible. Although there is decent reaction towards abrupt changes from low to high volatility, the PF and RSPF estimations are not able to react in cases where volatility diminishes. Since shape tracking extremely deficient, it is expected to have faulty detections in this data set.

The algorithms need around 300 to 400 data points to learn and correctly adapt to the observed data. In general, there is good tracking of trends before this turning point, but if the algorithm is able to adapt properly, both tendency and shape are correctly tracked.

One important aspect from these results is that estimation performance depends vastly on the value $\sigma_{\alpha,\beta}$. Previous experiments demonstrate that low values of $\sigma_{\alpha,\beta}$ (this is, lower

than the optimum) result in underestimation of both PF approaches, while higher values of $\sigma_{\alpha,\beta}$ are conducive to overestimation of volatility. From the purpose of tracking the shape of the envelope that characterizes the evolution of volatility in time, this is irrelevant, unless the filters lose the ability to track due to lack of particle variability. Given these results, it could be convenient to separate $\sigma_{\alpha,\beta}$ into σ_α and σ_β (this is, to consider separate sources of uncertainty on each pseudo-particle that extends the model). This is important for two reasons: First, is it necessary to understand that α multiplies the process noise σ_η and therefore, the process noise in the extended uGARCH model is the result of the multiplication of two random variables: $\alpha_k \sim \mathcal{N}(\alpha_{k-1}, \alpha_1\sigma_\alpha)$ and η_k^2. Second, variables α and β introduce different behaviors in the model, since the former is associated to innovations and the latter is associated to the memory of the model.

In Chapter 2, the value of $\alpha + \beta$ in the GARCH model was introduced as the "*process persistence parameter*, since it determines the speed of the mean-reversion of volatility to its long-term average. A higher value for $\alpha + \beta$ implies that the effect of the shocks of volatility, u_k^2, dies out slowly". In Table 5.8, the estimated value of $\alpha + \beta$ was included for each data set. Although there is no apparent relation between the filtering performance of the Particle Filters and the value of $\alpha + \beta$, there is in fact one relation that needs attention: The two data sets in which the estimated value of α was bigger, the filtering performance was more accurate and errors were systematically lower (Tables 5.5 and 5.6).

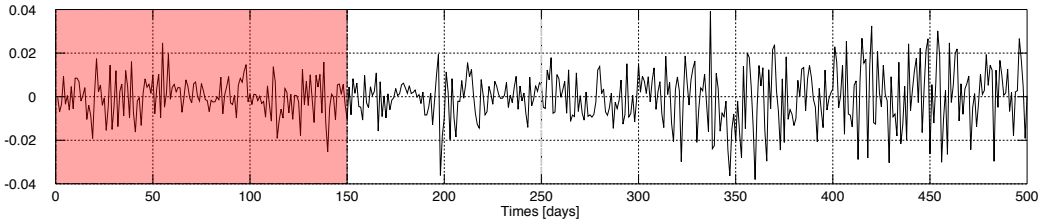
From a detection perspective, it is necessary to notice that the RSPF is usually more capable of tracking correctly sudden rises of volatility. In these cases, estimations of the RSPF are better than the PF estimations, since the latter tends to under estimate. This seems a natural result considering the construction of both PFs: The RSPF grants more resolution to high volatility areas, resulting in a better estimation of sudden volatility rises.

Continuing with the PF and RSPF comparison, the PF usually outperforms the RSPF in terms of estimation accuracy. Albeit sudden volatility changes from low to high values, the PF is less biased than the RSPF.

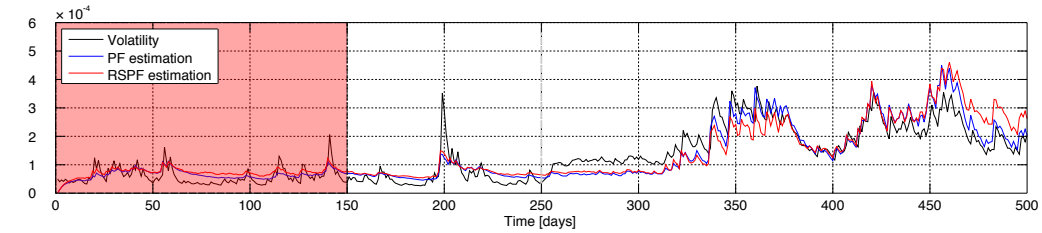
As a final comment, one should notice that the RSPF outperforms the Classic PF in terms of 1-step prediction in cases where volatility experiences sudden increments. This occurs due to the construction of the uGARCH model, in comparison to the GARCH model. Comparing both dynamics equations,

$$\begin{aligned} \text{GARCH: } \sigma_{k|k-1}^2 &= \omega + \alpha u_{k-1}^2 + \beta\sigma_{k-1|k-2}^2, \\ \text{uGARCH: } \sigma_k^2 &= \omega + \alpha\sigma_{k-1}^2\eta_k^2 + \beta\sigma_{k-1}^2, \end{aligned}$$

where $u_k = \sigma_{k|k-1}\epsilon_k$, the innovations process in the GARCH model depends on the value of the previous step of the returns process, while the volatility dynamics of the uGARCH model are time independent of the returns series. Since the simulated data was generated according to a GARCH model and cases associated with filtering through PF schemes is based on the uGARCH model, this 1-step prediction in sudden volatility rises is possible.

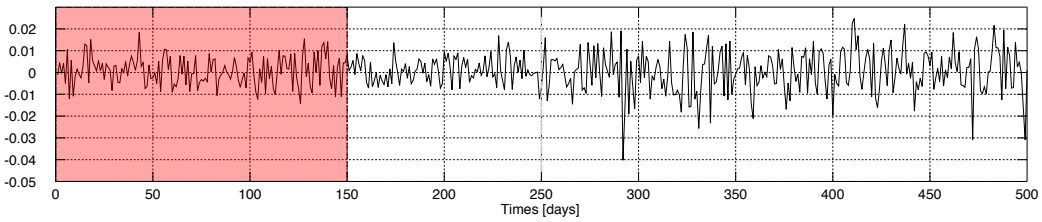


(a) Returns.

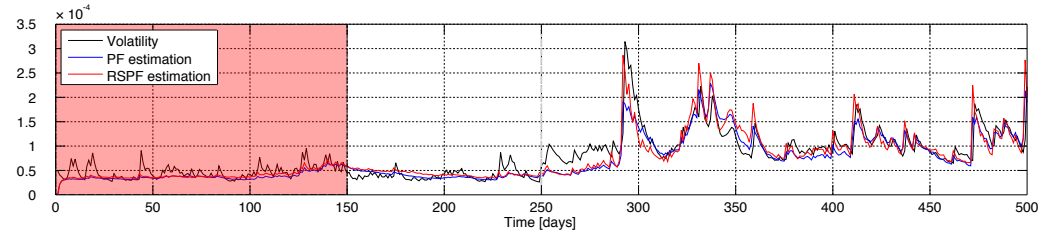


(b) Volatility.

Figure 5.1: Volatility estimation of GARCH0 data set using optimal parameters. Refer to Table 5.2 for percentage error.

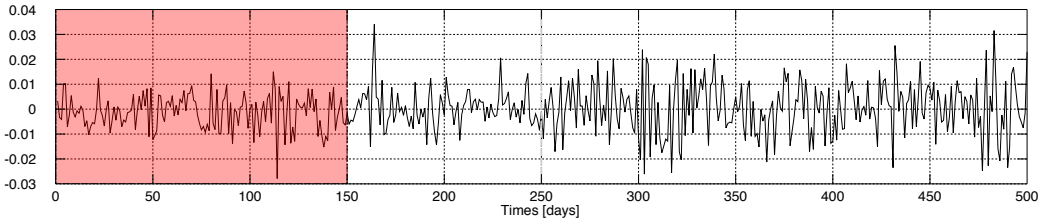


(a) Returns.

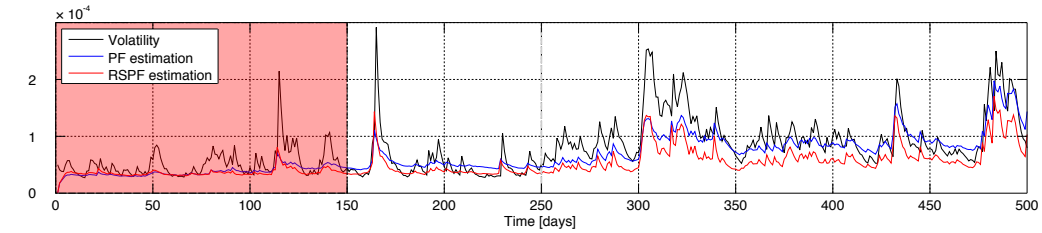


(b) Volatility.

Figure 5.2: Volatility estimation of GARCH1 data set using optimal parameters. Refer to Table 5.3 for error.

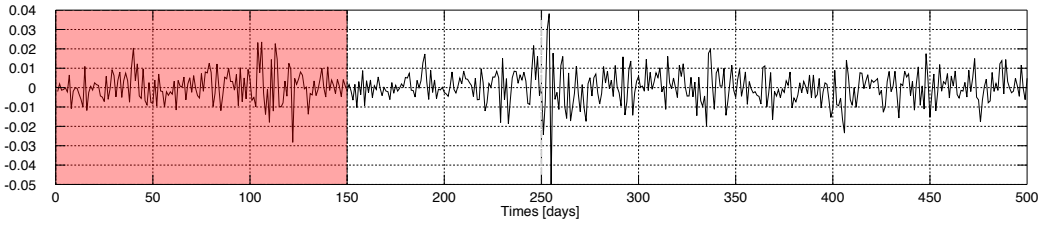


(a) Returns.

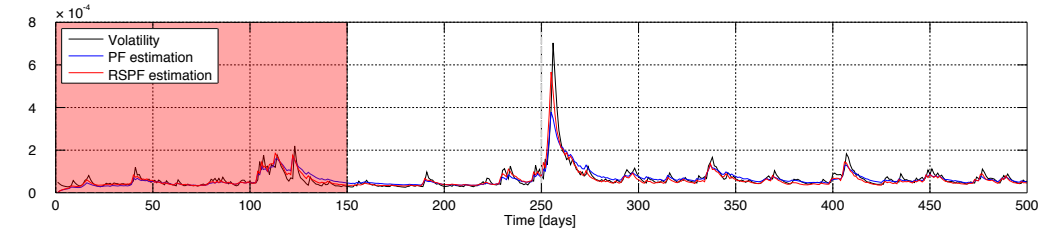


(b) Volatility.

Figure 5.3: Volatility estimation of GARCH2 data set using optimal parameters. Refer to Table 5.4 for error.

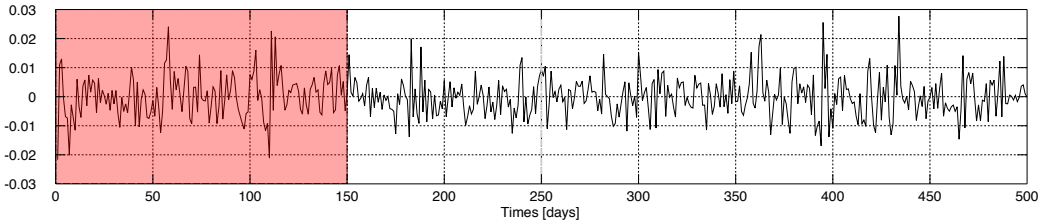


(a) Returns.

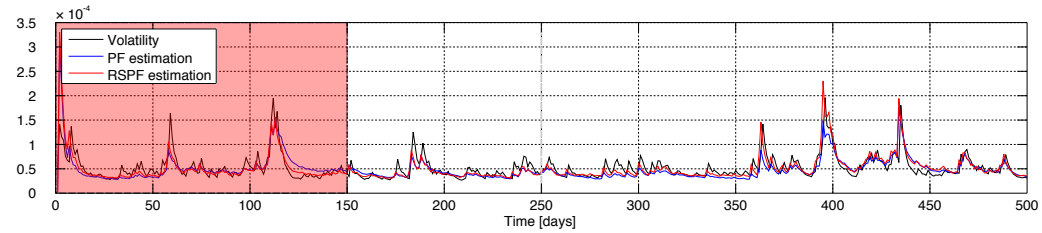


(b) Volatility.

Figure 5.4: Volatility estimation of GARCH3 data set using optimal parameters. Refer to Table 5.5 for error.

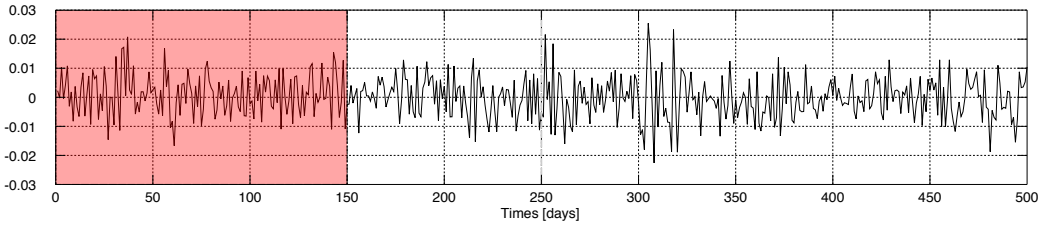


(a) Returns.

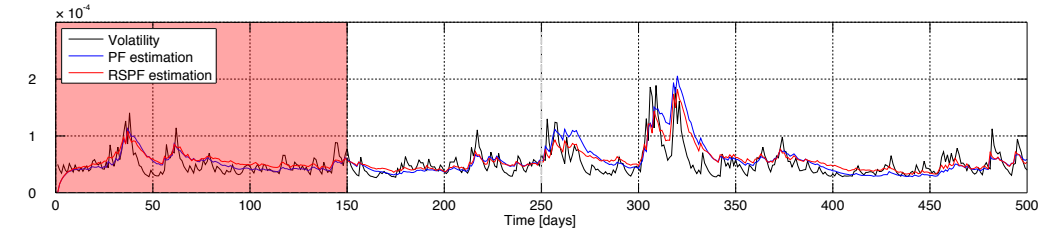


(b) Volatility.

Figure 5.5: Volatility estimation of GARCH4 data set using optimal parameters. Refer to Table 5.6 for error.



(a) Returns.



(b) Volatility.

Figure 5.6: Volatility estimation of GARCH5 data set using optimal parameters. Refer to Table 5.7 for error.

Filtering results with averaged parameters (as used in detection scheme)

This subsubsection presents the results obtained for volatility estimation using the average parameters for the Classic Particle Filtering processes (Expression 5.4). These parameters have also been applied and used in the RSPF. Results are shown in Figures 5.7 to 5.12.

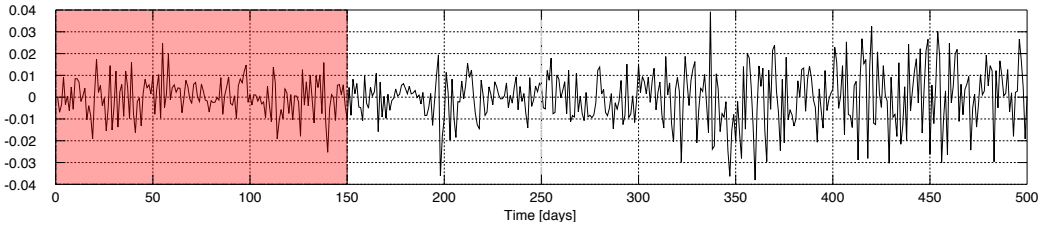
The previous subsubsection describes the phenomenon of over and underestimation related to the selected value of $\sigma_{\alpha,\beta}$. Given that the previous experiment showed results for the optimum value of this parameter, this situation was not apparent. Nonetheless, the new experiment makes this behavior palpable. A detailed analysis for each filtering process is given below.

- GARCH0 (Figure 5.7): The filtering process shows overestimation after the filters have adapted to the data in data set GARCH0. This is produced by the higher than optimum value of $\sigma_{\alpha,\beta}$ employed. This results in Particle Filters that clearly have less problems in learning from the data, which results in a more accurate shape tracking along the filtered time window.
- GARCH1 (Figure 5.8): An opposite behavior for the filtering process of GARCH1 may be observed, in contrast to GARCH0. The corresponding figure clearly shows underestimation of the state. Nevertheless, an interesting result is that shape tracking is extremely accurate, which is essential for the correct operation of the proposed detection algorithms.
- GARCH2 (Figure 5.9): Filters in data set GARCH2 have clearly less problems in learning from the data with the average parameters. This behavior translates into results with more accurate shape tracking. There is better accuracy in the estimation of sudden and vast volatility changes.
- GARCH3 (Figure 5.10): Results for this data set are extremely interesting because of the ample robustness of the filtering performance to variations of the value $\sigma_{\alpha,\beta}$. Shape tracking and estimation accuracy are almost intact in contrast to the use of optimum parameters.
- GARCH4 (Figure 5.11): A similar analysis can be presented for data set GARCH4 with respect to GARCH3. Although some accuracy is lost and mild high volatility clusters around time step 300 are not perfectly tracked, the overall filtering performance is very good. In this data set, filters are very robust to variations in the value of the parameter $\sigma_{\alpha,\beta}$.
- GARCH5 (Figure 5.12): Filtering performance for data set GARCH5 is extremely poor. Albeit trend tracking is available, there is no learning throughout the filtering window and consequently, sudden volatility changes are poorly tracked. For this reason, it is extremely unlikely for the proposed detectors to work correctly.

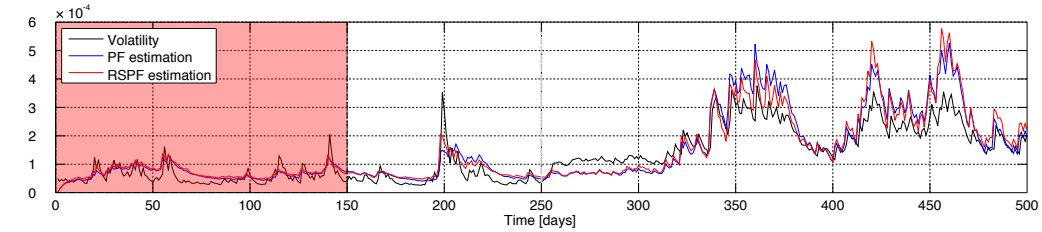
The anomalous behavior occurring in data sets GARCH1 (Figure 5.8) and GARCH5 (Figure 5.12) may be explained again by the estimation through maximum likelihood of parameter α.

Comparing the performance of the PF and RSPF, one needs to compare the behavior of these filters in Figures 5.7 to 5.11 (which exclude data set GARCH5). In comparison

to the previous experiment with optimum parameters, there is again a clear response from the RSPF towards estimating correctly sudden changes in volatility from low to high values. This is correct for sudden changes, since the PF tends to be less biased in average. This behavior is extremely important for the detection scheme, since correct performance from the proposed detection techniques can be obtained even though the optimal parameters are not used in simulated or real data.

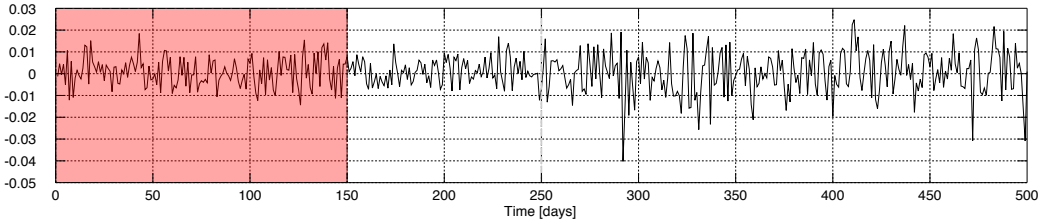


(a) Returns.

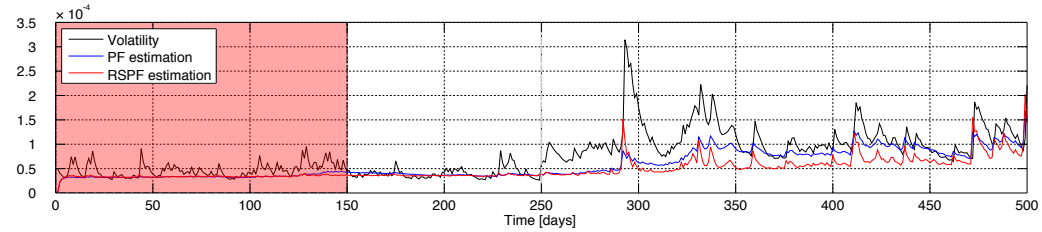


(b) Volatility.

Figure 5.7: Volatility estimation of GARCH0 data set using average parameters.

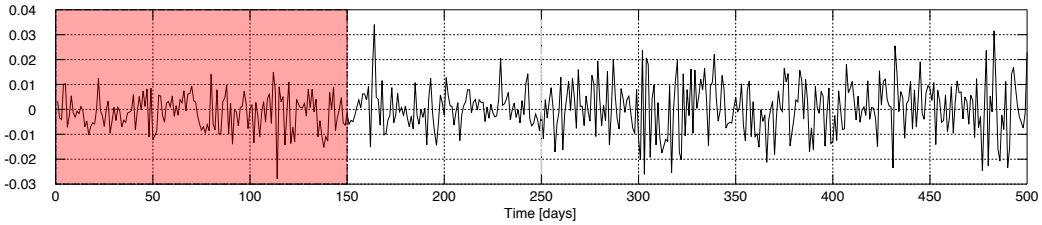


(a) Returns.

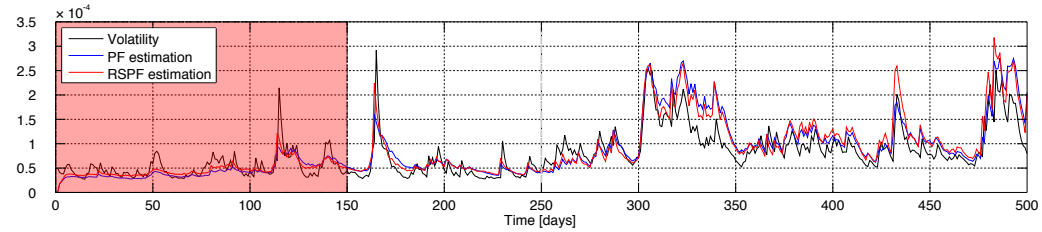


(b) Volatility.

Figure 5.8: Volatility estimation of GARCH1 data set using average parameters.

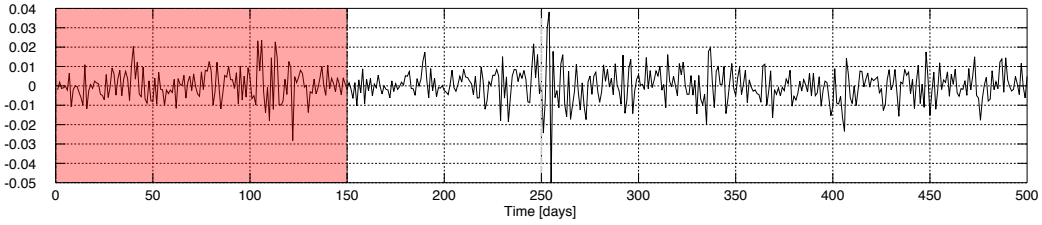


(a) Returns.

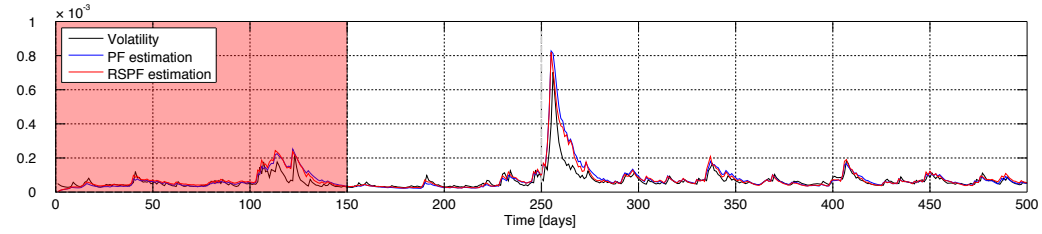


(b) Volatility.

Figure 5.9: Volatility estimation of GARCH2 data set using average parameters.

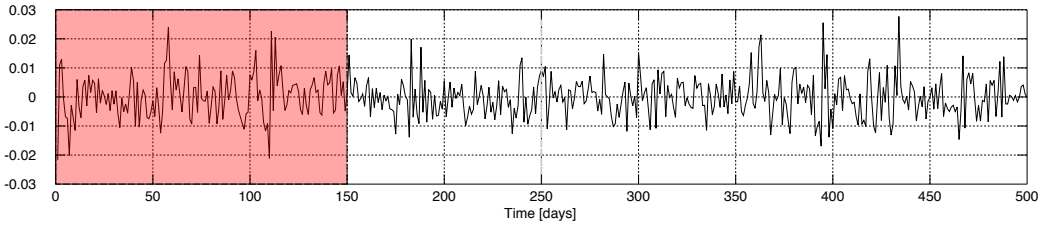


(a) Returns.

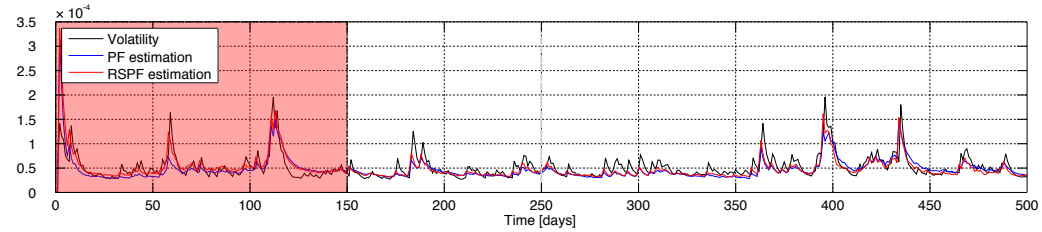


(b) Volatility.

Figure 5.10: Volatility estimation of GARCH3 data set using average parameters.

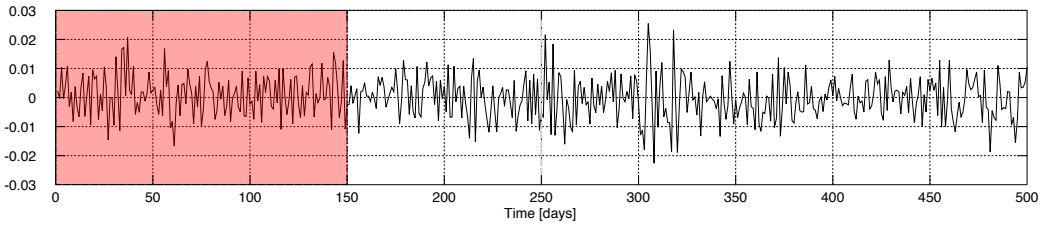


(a) Returns.

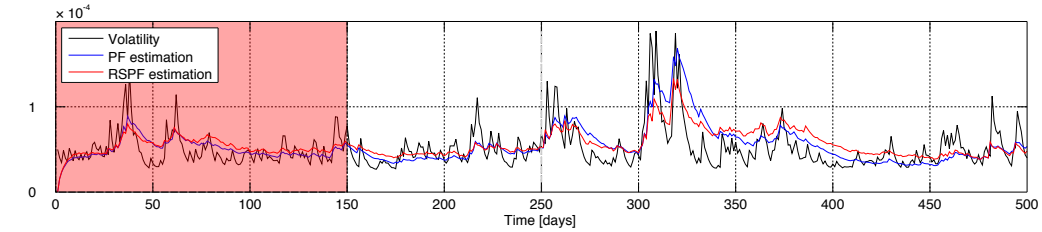


(b) Volatility.

Figure 5.11: Volatility estimation of GARCH4 data set using average parameters.



(a) Returns.



(b) Volatility.

Figure 5.12: Volatility estimation of GARCH5 data set using average parameters.

5.3 Detection using a Hypothesis Test

This section presents results obtained from the proposed hypothesis test to capture early rises in volatility. Figures 5.13 to 5.18 show these results. In all these figures, there are five subfigures which correspond to (a) price series, (b) returns series, (c) volatility series, RSPF prior and posterior estimation, and confidence interval, (d) volatility net differences, and (e) detection points. These set of subfigures encapsulate the associated information in the detection process, and help the reader comprehend the different aspects of the detection.

Figures 5.13 to 5.17 (notice the exclusion of data set GARCH5) show that the detector works correctly, since it is able to capture early rises of volatility which transform into high volatility clusters. This does not occur for data set GARCH5, in which estimation is extremely deficient and as a consequence, the detector does not work. The detector can also be interpreted as a local peak detector in the returns series, which is expected. Since the hypothesis test contrasts the dynamics of the model (prior) and the updated dynamics through the observations (posterior), it is clear that detections will occur mainly when local peaks of returns occur.

A detailed analysis of the results for each of the data sets is given ahead.

- GARCH0 (Figure 5.13): The filtering process and consequent detection in this data set offer early detection of both of the greatest high volatility clusters. There is also excellent detection in high volatility sub-clusters over high volatility regions. In particular, the cluster around time step 200 is detected and the cluster after time step 300 is detected. Over the last 200 filtering steps, rapid variations over the volatility cluster are also detected. Nevertheless, the proposed performance measure does not count detections of high volatility sub-clusters, which explains the low positive predictive value, given the relatively high number of false positives.
- GARCH1 (Figure 5.14): All of the high volatility clusters are detected, except for the high volatility variation due to regime shift at time step 250. This regime shift introduces a notorious mean variation in volatility, which the test is not able to capture, since there are no vast variations in the returns series. High volatility sub-clusters around time step 350 are also detected. From the performance measures perspective, the detector excels in this data set, given that there are no false positives and only 2 false negatives in 350 time steps, leading to a positive predictive value equal to 1.0.
- GARCH2 (Figure 5.15): As in the previous cases, all of the high volatility clusters are promptly detected. Nevertheless, in this data set there is a tendency towards higher sensibility in the detection process, since lower variations of volatility are identified as well. This behavior is observable in time steps 230 approximately, and various times between time steps 250 and 300. Given that the ground truth detection points in this data set leads to 8 false negatives, sensitivity of the test is not good. Nevertheless, positive predictive value has a fair value of 0.8666.
- GARCH3 (Figure 5.16): Table 5.11b shows a peculiarly low positive predictive value of 0.1818 in data set GARCH3. The reason for this low value may be observed in Figure 5.16d, where occurring detections are considered to be false positives due to the high value of the proposed threshold. Despite the very low positive predictive value,

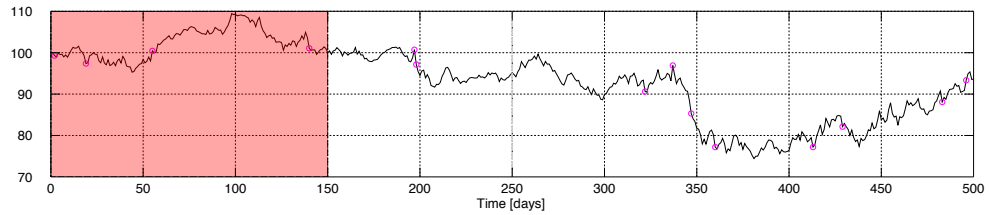
qualitative analysis of Figures 5.16c and 5.16e show that there is extremely accurate detection of higher volatility episodes along the filtered time window. All of the major sudden volatility rises are detected, except for the higher volatility episode starting at approximately time step 320. Continuing over this line of thought, the detector works as expected, although the performance measures do not correctly express the actual detector capacity.

- GARCH4 (Figure 5.17): The detection scheme applied to data set GARCH4 produces very similar results to the observed ones in data set GARCH3. An extremely low value of 0.0769 for the positive predictive value is given by the low value of detections classified as true positives (only 1) and the high value of detections classified as false positives (12). The high amount of false positives is given by the position of the proposed threshold. Nonetheless, a qualitative analysis of Figures 5.17c and 5.17e denotes that most of the rapid volatility variations are detected. High volatility clusters present in time steps 170, 270, 360, 390, 440 and 470 are detected, while only small clusters in time steps 240, 260 and 480 are missed.
- GARCH5 (Figure 5.18): This data set presents an anomalous behavior given the low quality of estimation. As previously analyzed, the RSPF is only capable of tracking the trend of the state, but it is not able to learn given the bias in the estimation of the parameter α of the model uGARCH. This data set demonstrates that the proposed hypothesis test-based detection scheme needs a decent shape tracking from the RSPF in order to work in a proper manner.

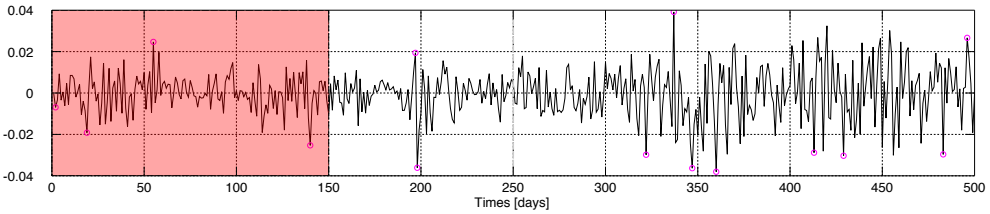
The detection results obtained through a proposed hypothesis test show that the detector is very sensitive even to mild high volatility clusters. This translates into the difficulty of measuring correctly the performance of the algorithm, since there is no possible definable hard limit between low and high volatility clusters. In fact, one can only use a diffuse definition. Other results that need to be addressed correspond to the robustness of the algorithm to the employed value of $\sigma_{\alpha,\beta}$ in the estimation stage. This parameter has tremendous implications over the estimation performance, but not over detection performance. If shape is tracked correctly, the hypothesis test-based detector performs exceptionally well, even under extreme estimation biases.

From an evaluation of the performance standpoint, the proposed threshold for introducing ground truth points for detection is not fair with the detection outcome (in the sense of positive predictive value). This produces low values in performance measures that do not exactly correspond to the observed outcome in qualitative analysis. Moreover, accuracy is not a very good indicator of performance since the amount of ground truth points for detection is extremely low compared to the length of the filtered time window.

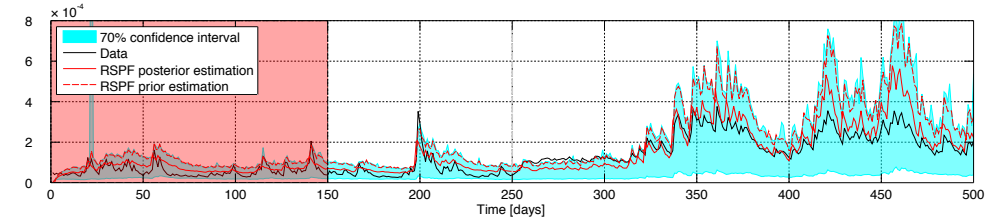
The proposed method for calculating the threshold intrinsically assumes that the mean volatility augments throughout the filtering window, given that it is calculated using only very rare volatility changes in the training period. Since the training window is not necessarily representative of the studied time span, there might be occasions where the proposed performance scheme for detection may not be fair with the actual detection results. This phenomenon occurs when the value $\alpha + \beta$ augments during the studied time span. This is present in the detection results of data set GARCH4, where a positive predictive value of 0.0769 does not correspond to the quality of the detections appreciable in Figure 5.17.



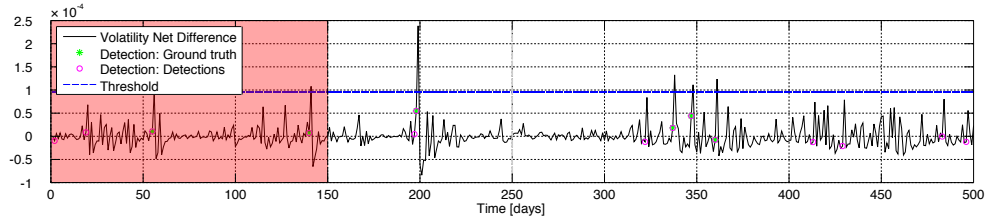
(a) Prices.



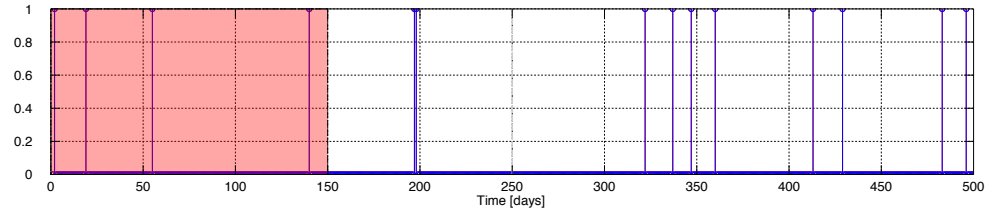
(b) Returns.



(c) Volatility estimation.



(d) Volatility net differences.



(e) Detections.

Figure 5.13: Hypothesis test-based detection for data set GARCH0.

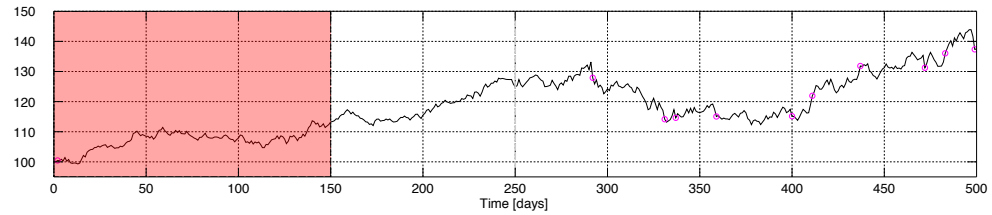
Table 5.9: Detection results for GARCH0 data set early high volatility cluster detection.

		Condition	
		$\mathcal{H}_0$	$\mathcal{H}_1$
Test outcome	$\mathcal{H}_0$	4	6
	$\mathcal{H}_1$	0	340

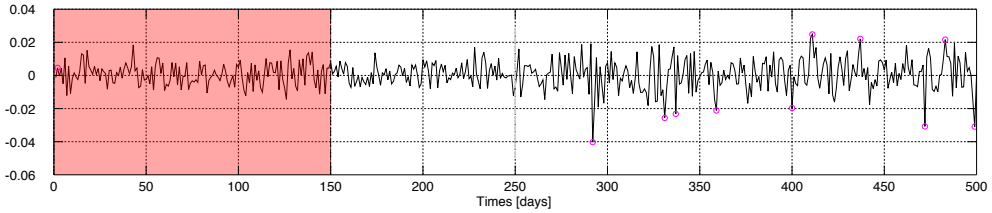
(a) Confusion matrix.

Positive Predictive Value	0.4000
Negative Predictive Value	1.0000
Sensitivity	1.0000
Specificity	0.9826
Accuracy	0.9828

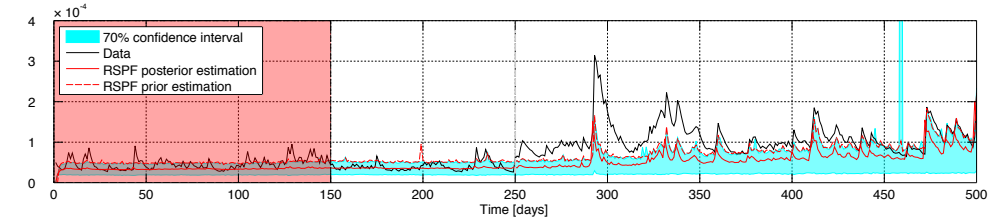
(b) Performance measures.



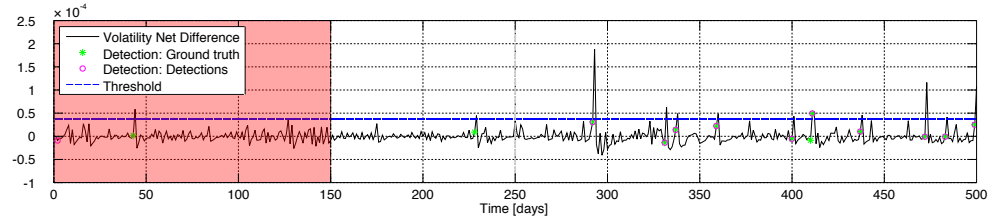
(a) Prices.



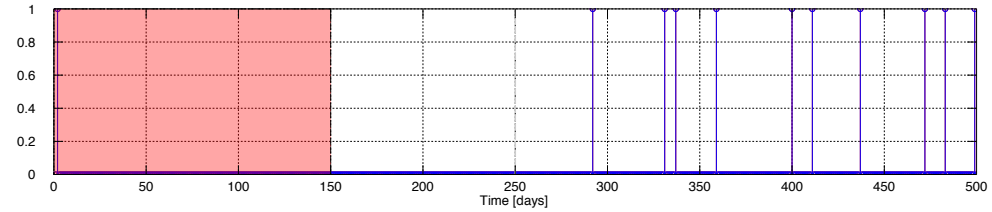
(b) Returns.



(c) Volatility estimation.



(d) Volatility net differences.



(e) Detections.

Figure 5.14: Hypothesis test-based detection for data set GARCH1.

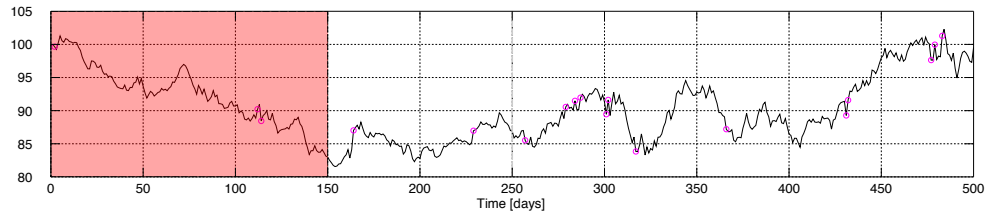
Table 5.10: Detection results for GARCH1 data set early high volatility cluster detection.

		Condition	
		$\mathcal{H}_0$	$\mathcal{H}_1$
Test outcome	$\mathcal{H}_0$	10	0
	$\mathcal{H}_1$	2	338

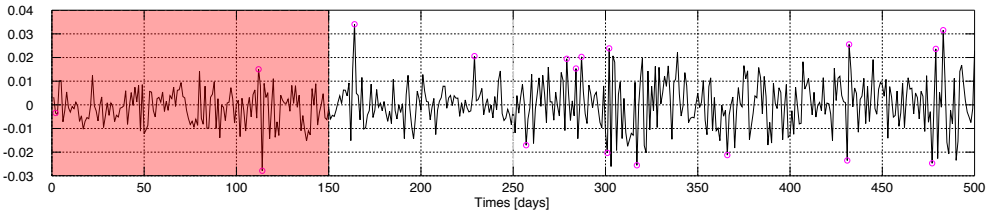
(a) Confusion matrix.

Positive Predictive Value	1.0000
Negative Predictive Value	0.9941
Sensitivity	0.8333
Specificity	1.0000
Accuracy	0.9942

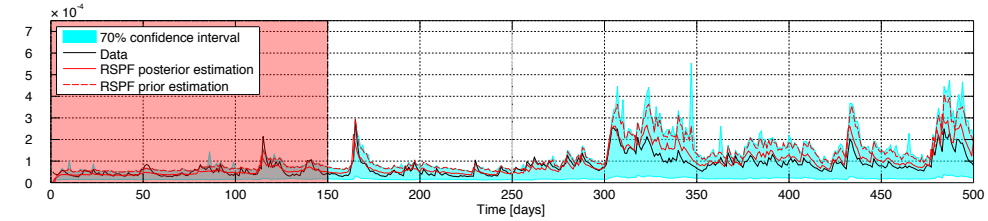
(b) Performance measures.



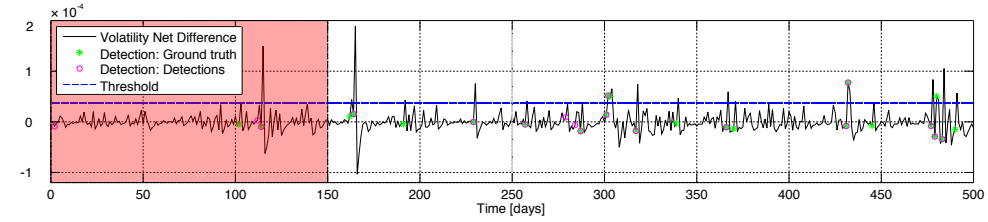
(a) Prices.



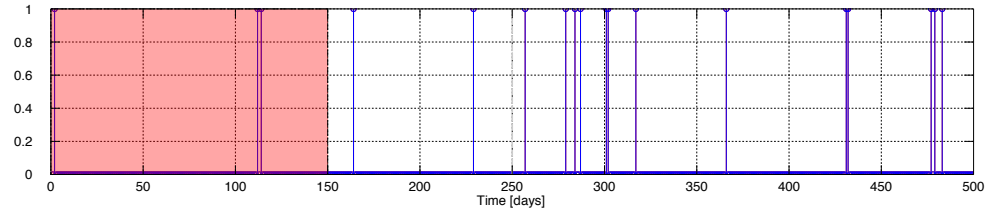
(b) Returns.



(c) Volatility estimation.



(d) Volatility net differences.



(e) Detections.

Figure 5.15: Hypothesis test-based detection for data set GARCH2.

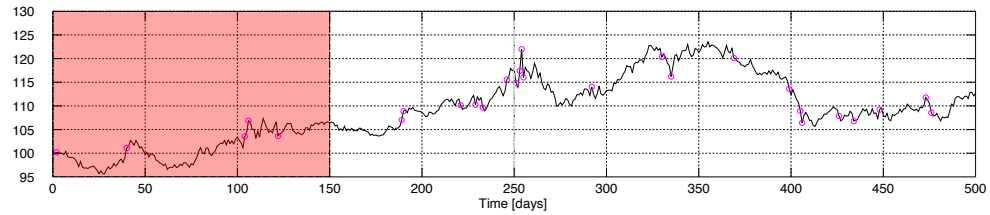
Table 5.11: Detection results for GARCH2 data set early high volatility cluster detection.

		Condition	
		$\mathcal{H}_0$	$\mathcal{H}_1$
Test outcome	$\mathcal{H}_0$	13	2
	$\mathcal{H}_1$	8	327

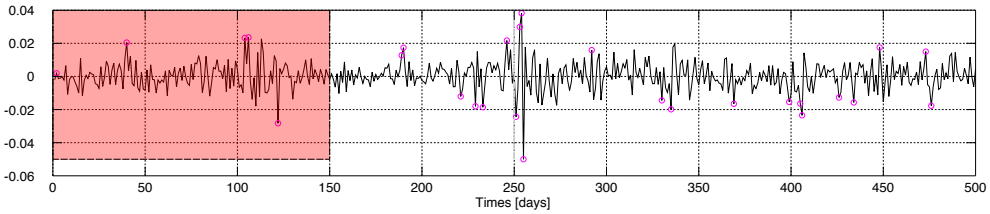
(a) Confusion matrix.

Positive Predictive Value	0.8666
Negative Predictive Value	0.9761
Sensitivity	0.6190
Specificity	0.9939
Accuracy	0.9714

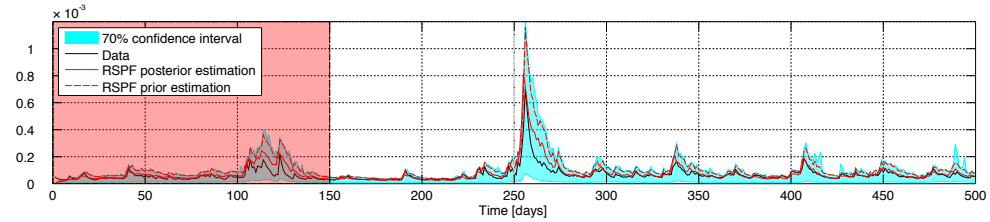
(b) Performance measures.



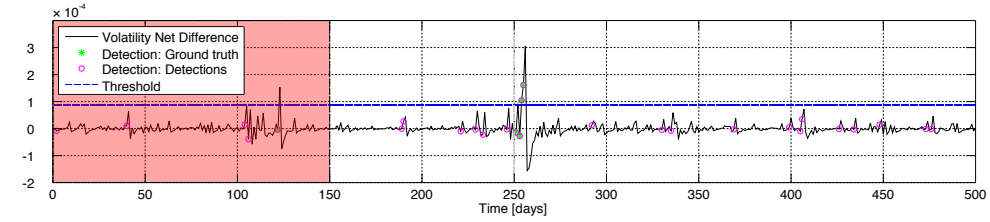
(a) Prices.



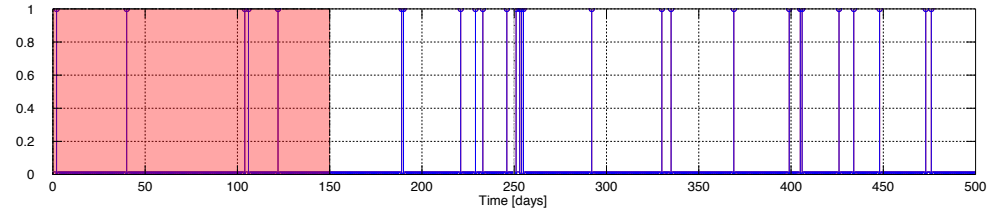
(b) Returns.



(c) Volatility estimation.



(d) Volatility net differences.



(e) Detections.

Figure 5.16: Hypothesis test-based detection for data set GARCH3.

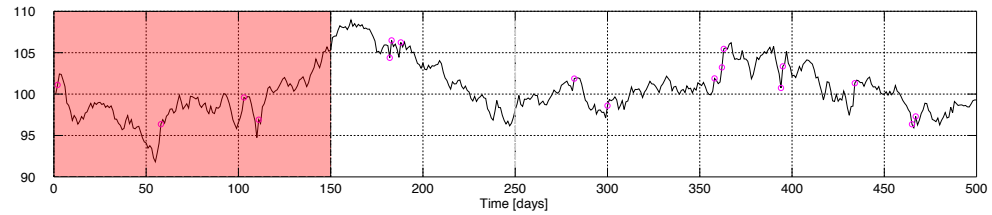
Table 5.12: Detection results for GARCH3 data set early high volatility cluster detection.

		Condition	
		$\mathcal{H}_0$	$\mathcal{H}_1$
Test outcome	$\mathcal{H}_0$	4	18
	$\mathcal{H}_1$	0	328

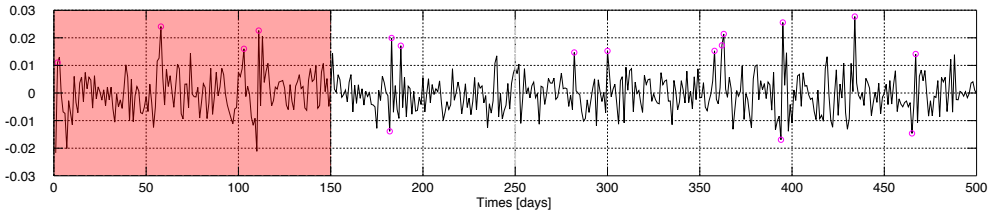
(a) Confusion matrix.

Positive Predictive Value	0.1818
Negative Predictive Value	1.0000
Sensitivity	1.0000
Specificity	0.9479
Accuracy	0.9485

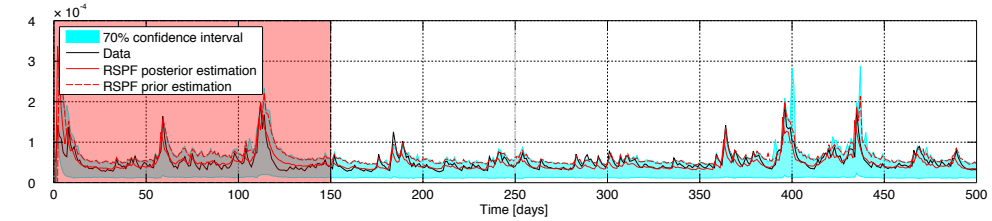
(b) Performance measures.



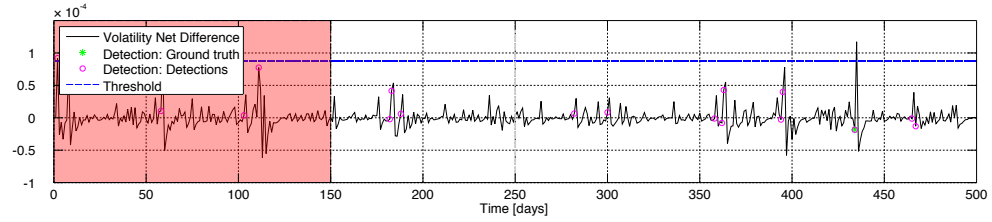
(a) Prices.



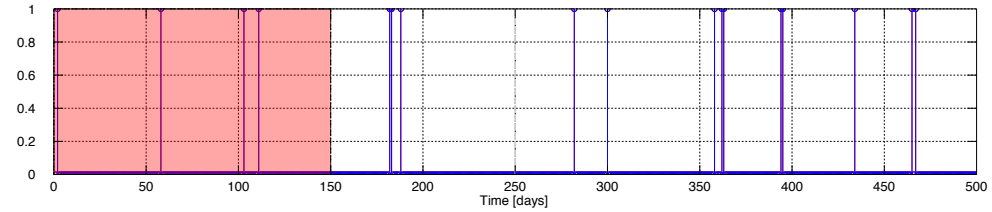
(b) Returns.



(c) Volatility estimation.



(d) Volatility net differences.



(e) Detections.

Figure 5.17: Hypothesis test-based detection for data set GARCH4.

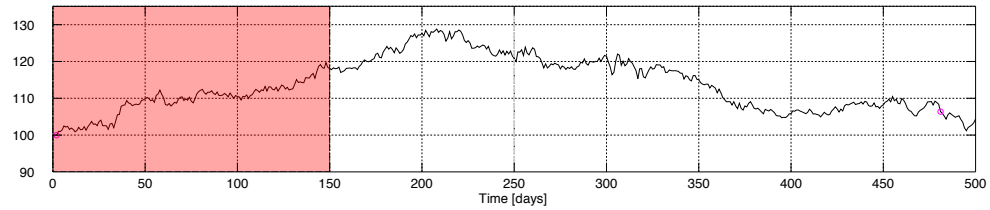
Table 5.13: Detection results for GARCH4 data set early high volatility cluster detection.

		Condition	
		$\mathcal{H}_0$	$\mathcal{H}_1$
Test outcome	$\mathcal{H}_0$	1	12
	$\mathcal{H}_1$	0	337

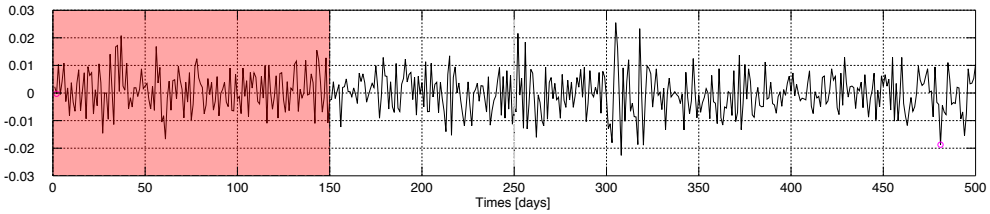
(a) Confusion matrix.

Positive Predictive Value = 0.0769
Negative Predictive Value = 1.0000
Sensitivity = 1.0000
Specificity = 0.9651
Accuracy = 0.9657

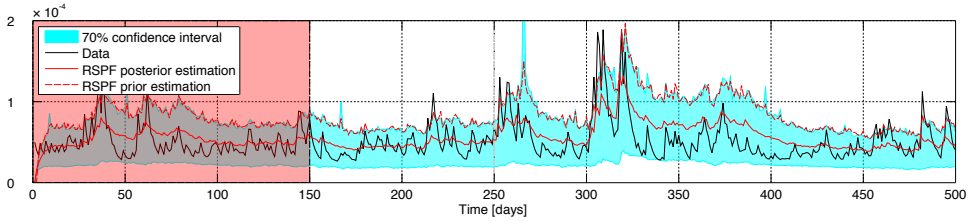
(b) Performance measures.



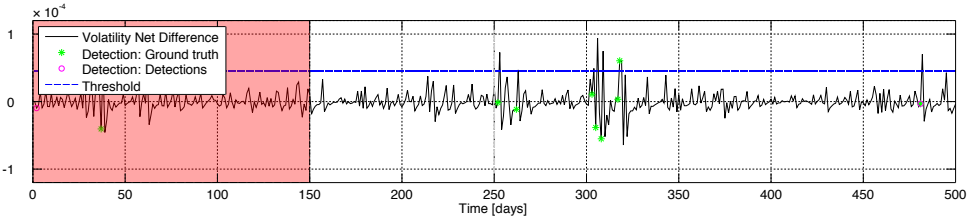
(a) Prices.



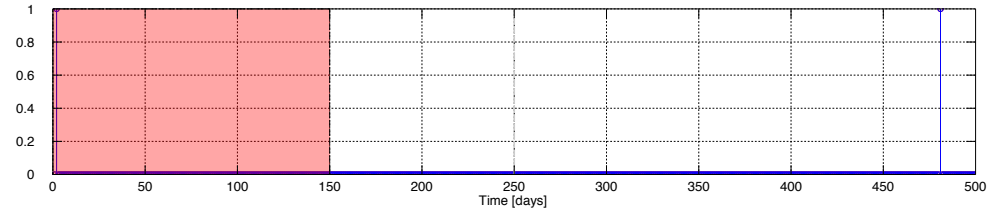
(b) Returns.



(c) Volatility estimation.



(d) Volatility net differences.



(e) Detections.

Figure 5.18: Hypothesis test-based detection for data set GARCH5.

Table 5.14: Detection results for GARCH5 data set early high volatility cluster detection.

		Condition	
		$\mathcal{H}_0$	$\mathcal{H}_1$
Test outcome	$\mathcal{H}_0$	1	0
	$\mathcal{H}_1$	7	342

(a) Confusion matrix.

Positive Predictive Value = 1.0000
Negative Predictive Value = 0.9799
Sensitivity = 0.1250
Specificity = 1.0000
Accuracy = 0.9800

(b) Performance measures.

5.4 Detection using Fisher Discriminant and Kullback-Leibler Divergence

This section presents, describes, and analyzes the results obtained by comparing the information obtained from the PF and RSPF filtering schemes in the simulated data sets. As previously described, the Fisher Discriminant (FD) and Kullback-Leibler Divergence (KLD) are tools to compare probability density functions. Therefore, it is natural to compare the estimated probability density functions by two different filtering schemes through these tools. In this section, the posterior distributions of both particle filtering schemes are compared, using average parameters obtained from the sensibility analysis of Section 5.2.2.

The results are introduced graphically, in a manner similar to what has been previously presented. These results are shown in Figures 5.19 through 5.24. In particular, subfigures (d) and (e) show both the values for the Fisher Discriminant and the Kullback-Leibler Divergence for every time step, respectively.

The results are presented analyzing each data set, in the same structure that has been used to discuss estimation results and hypothesis test-based detection results.

- GARCH 0 (Figure 5.19): The use of the FD in this data set lets one manually detect the volatility cluster around time step 200. Rises of the FD in the same time steps where sudden volatility rises occur may be observed. Nevertheless, the use of the FD is not determinant. The KLD rises do not show any visual qualitative correlation with the behavior of ground truth volatility.
- GARCH 1 (Figure 5.20): The FD is extremely close to 0 in this set until approximately time step 292, where there is a considerable rise in volatility. Since both the PF and RSPF give extremely similar estimates before time step 290, this is an expectable result. The big spike around time step 290 effectively detects the sudden volatility change. Nonetheless, other rapid state variations are not detected. With respect to the KLD, there is again no apparent correlation between volatility variations and divergence value.
- GARCH 2 (Figure 5.21): In this data set, the FD is able to capture high volatility clusters in an early stage three times: Around time step 170, in time step 430 and again around time step 470. Nevertheless, there is no detection for the biggest high volatility cluster starting at time step 300. With respect to the KLD, spike detecting the high volatility cluster at time step 170 coincides with the behavior of the FD. However, the are no other significant detections represented through spikes in its value that correspond to rapid state variations.
- GARCH 3 (Figure 5.22): This set of data is extremely favorable for the use of the FD, since there are rises in its value only when there are abrupt volatility changes. This response could lead to the use of a threshold for an automatic detection framework. On the other hand, the KLD suffers from the same problems present in the previous data sets.
- GARCH 4 (Figure 5.23): Again, presented results show that the FD rises when there are brusque rises in volatility. This behavior again could lead to the use of a threshold

to determine positive detections. On the other hand, the KLD suffers from the same problems, and spikes in its value are not representative of the events present in the estimation process.

- GARCH 5 (Figure 5.24): Data set GARCH5 clearly shows that neither the FD, nor the KLD are robust against low quality estimations for the volatility process. As in the case of the hypothesis test-based detection, these detection tools need from a decent estimation performance in order to perform adequately.

There are general conclusions that cover the results for each one of the data sets. Comparison of the Kullback-Leibler Divergence's performance and the Fisher Discriminant's performance show that the former is more noisy than the latter. There are several factors that may explain this behavior. The Kullback-Leibler Divergence considers all of the infinite moments of the probability density functions when comparing them. Given that the Fisher Discriminant only uses the first and second moments, more information may produce noisier results in this experiment. Additionally, the analysis includes multimodal distributions, which may intrinsically produce higher values for the Kullback-Leibler Divergence, given its sensibility towards higher valued moments. Since the Fisher Discriminant does not discriminate between uni and multimodal distributions, it is reasonable to obtain less noisy results.

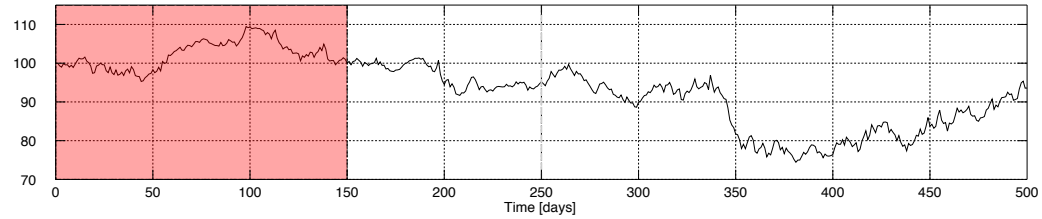
Another likely reason for the noisy results presented by the Kullback-Leibler Divergence is the approximations in its calculation. There are two main noise sources:

1. No consideration of the conditional values probability densities,
2. Redefinition of the posterior probability density functions of the PF and RSPF to get equal supports for correct evaluation.

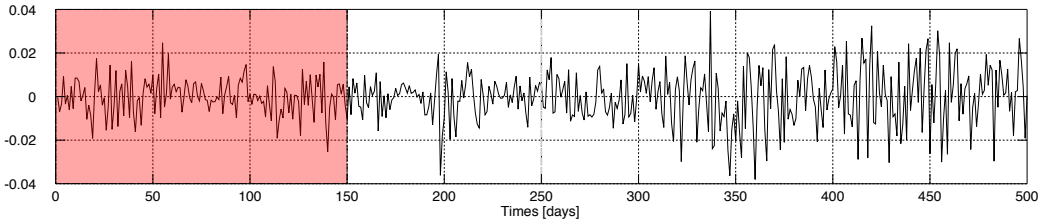
Since there is no possibility to calculate the Kullback-Leibler Divergence without approximations, it is unfortunately not possible to quantify the error introduced with these approximations. Nonetheless, it is clear that these approximations do not lead to a correct tool for early detection of high volatility clusters using Particle Filter estimates.

These tools are also important because they cannot aid the detection process by setting a detection threshold (as was possible, although with problems, in the hypothesis test-based detection). Albeit some clear detections from the Fisher Discriminant, this proposed algorithm may not be used as the base for an automatic algorithm for alert generation in the case of high-risk events.

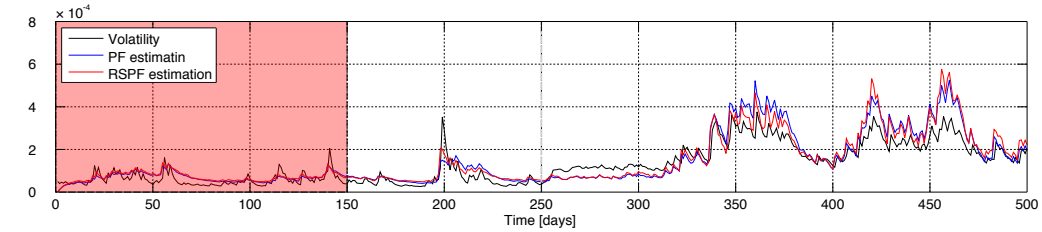
In these results, there is a tendency of the Fisher Discriminant to perform well in cases where the filtering algorithms perform well. In data sets GARCH3 and GARCH4, the estimation is very accurate, and there are evident signs of detection from part of the Fisher Discriminant. In data set GARCH5, where estimation is extremely deficient, the Fisher Discriminant performs poorly.



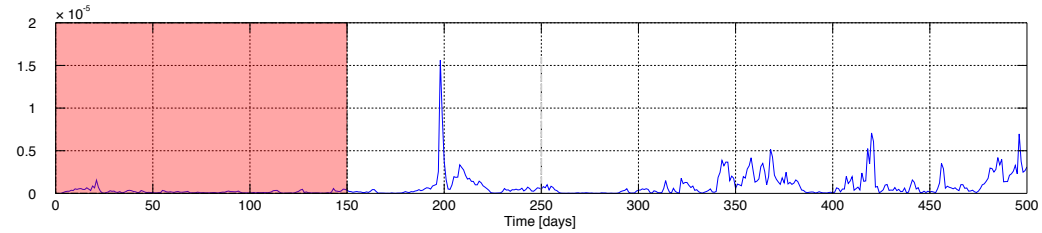
(a) Prices.



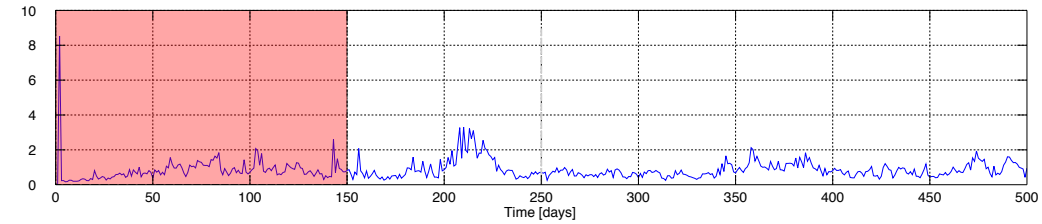
(b) Returns.



(c) Volatility estimation.

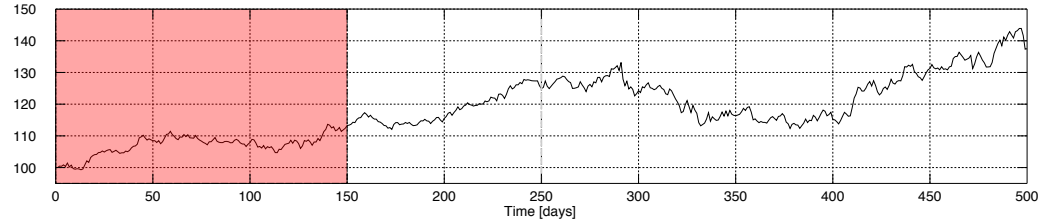


(d) Fisher discriminant.

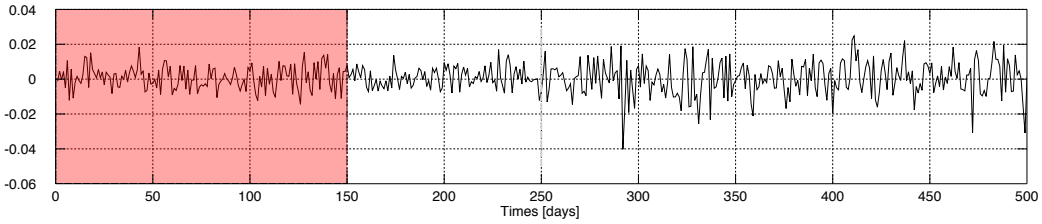


(e) Kullback-Leibler divergence.

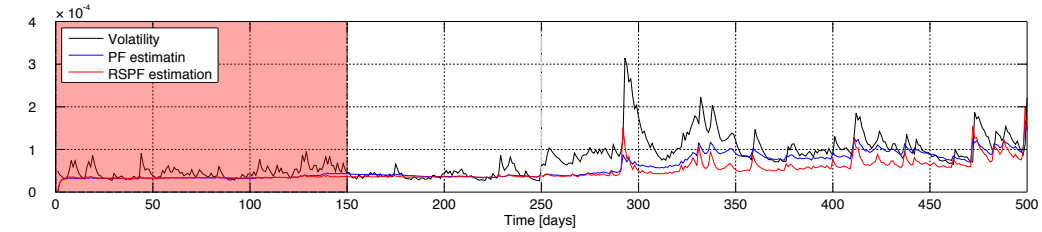
Figure 5.19: PDF comparison-based detection for data set GARCH0.



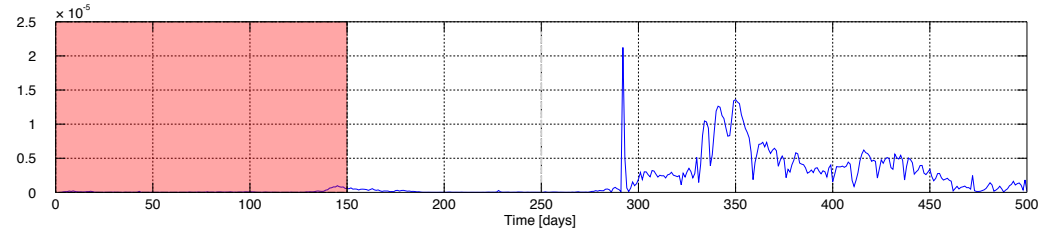
(a) Prices.



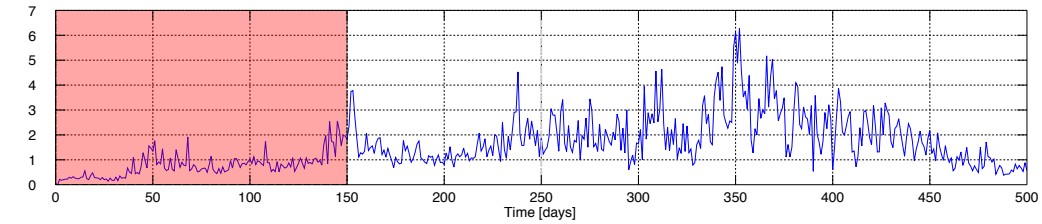
(b) Returns.



(c) Volatility estimation.

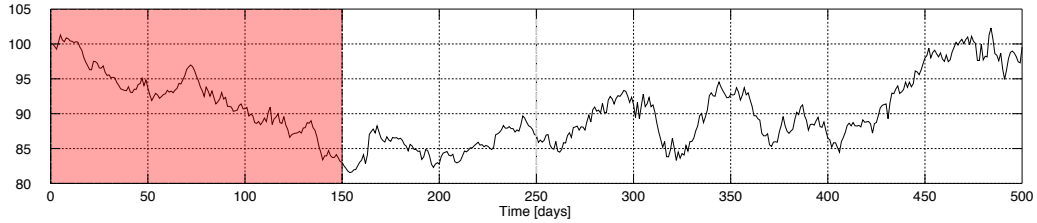


(d) Fisher discriminant.

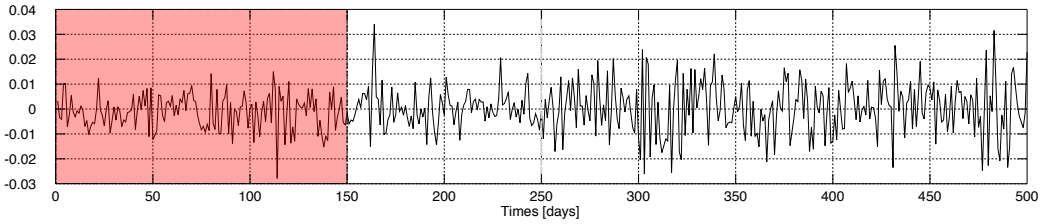


(e) Kullback-Leibler divergence.

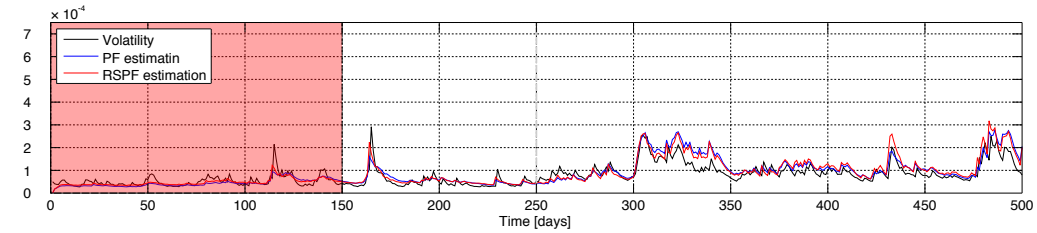
Figure 5.20: PDF comparison-based detection for data set GARCH1.



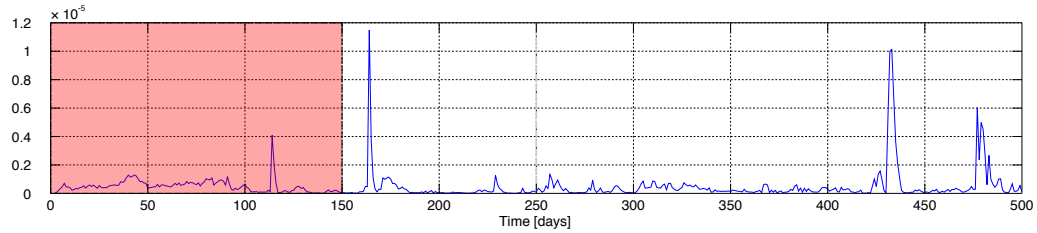
(a) Prices.



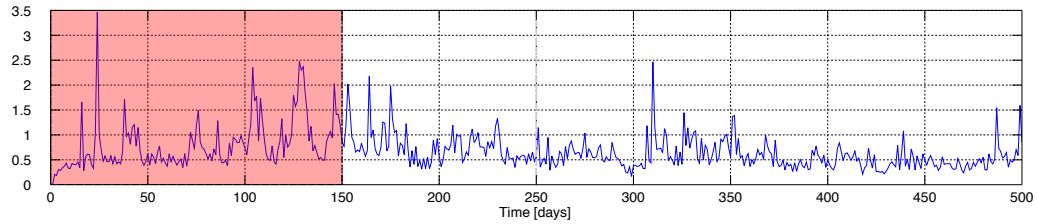
(b) Returns.



(c) Volatility estimation.

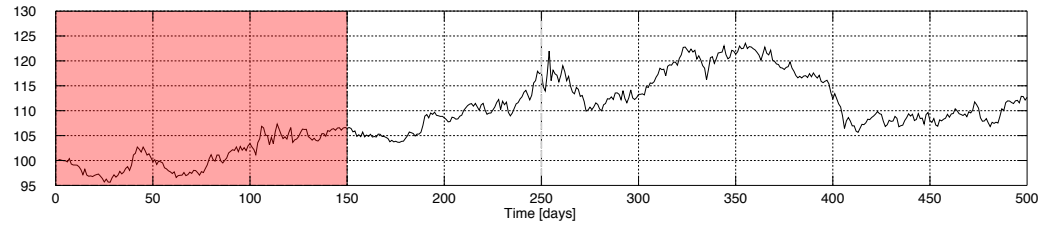


(d) Fisher discriminant.

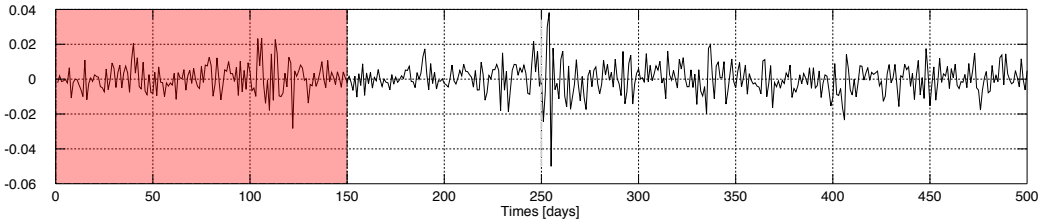


(e) Kullback-Leibler divergence.

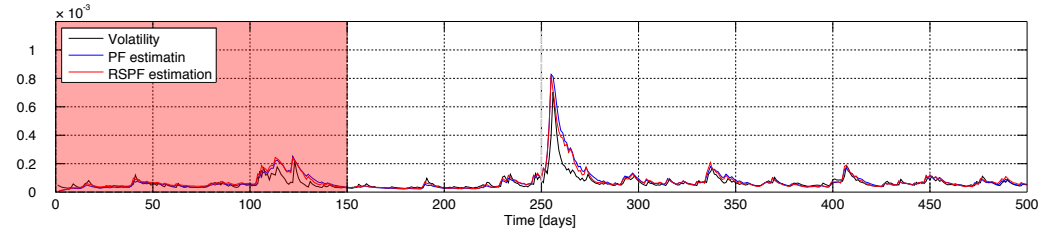
Figure 5.21: PDF comparison-based detection for data set GARCH2.



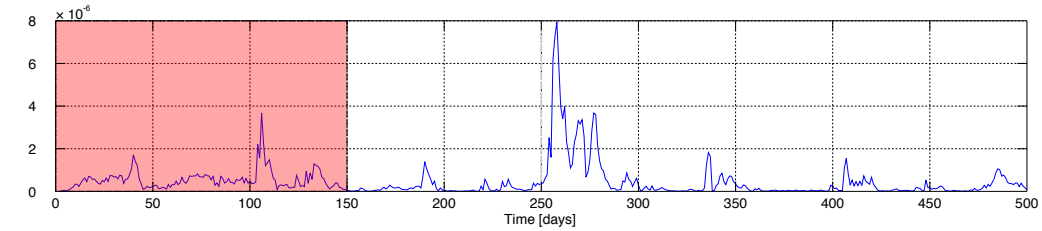
(a) Prices.



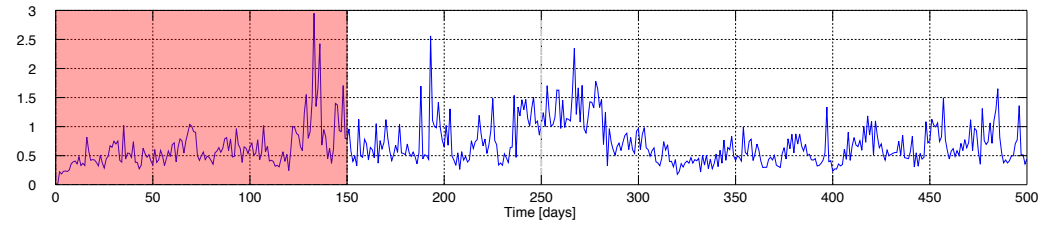
(b) Returns.



(c) Volatility estimation.

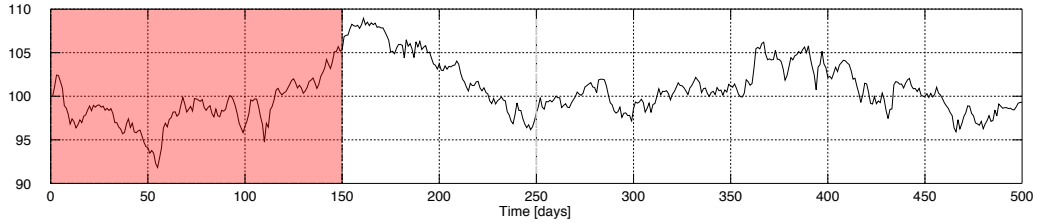


(d) Fisher discriminant.

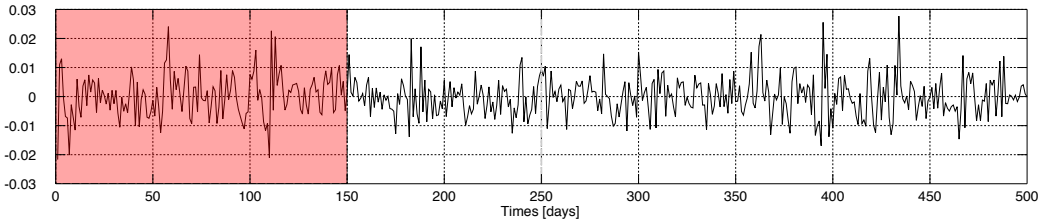


(e) Kullback-Leibler divergence.

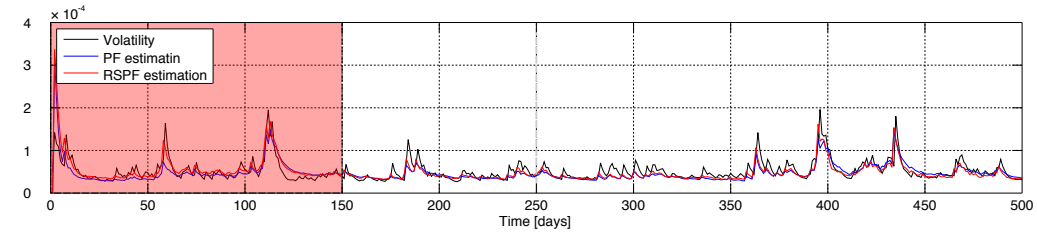
Figure 5.22: PDF comparison-based detection for data set GARCH3.



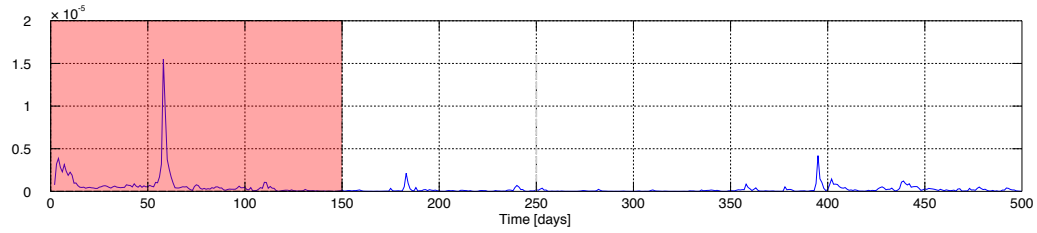
(a) Prices.



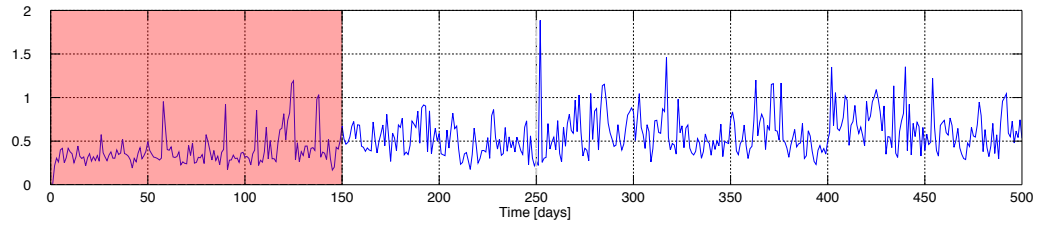
(b) Returns.



(c) Volatility estimation.

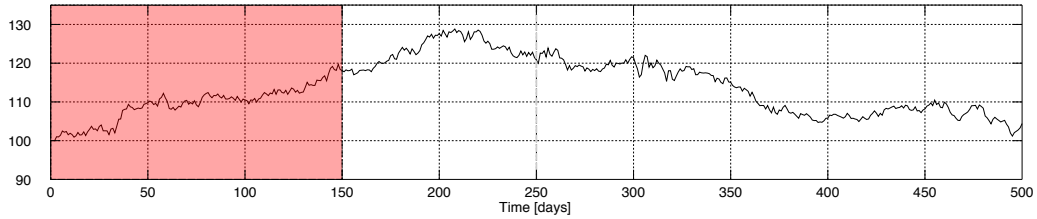


(d) Fisher discriminant.

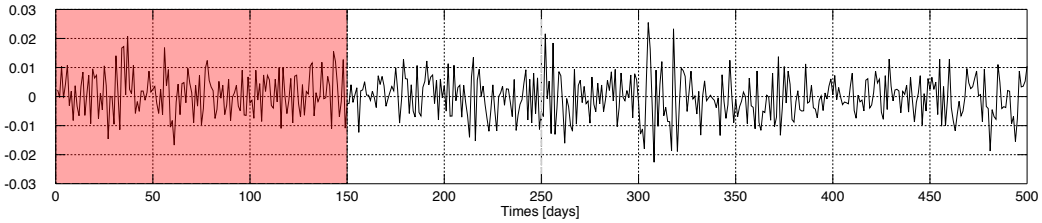


(e) Kullback-Leibler divergence.

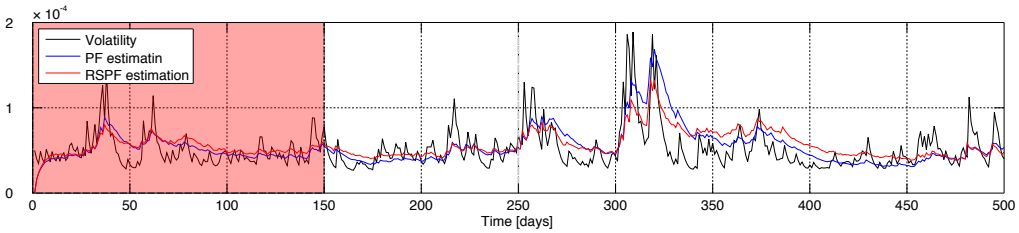
Figure 5.23: PDF comparison-based detection for data set GARCH4.



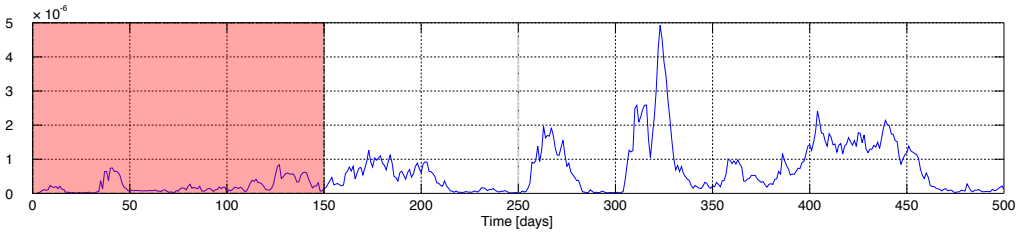
(a) Prices.



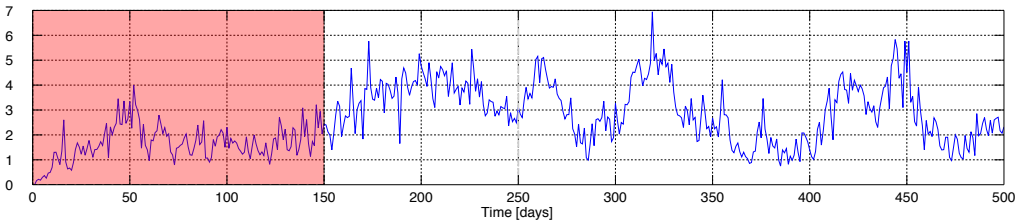
(b) Returns.



(c) Volatility estimation.



(d) Fisher discriminant.



(e) Kullback-Leibler divergence.

Figure 5.24: PDF comparison-based detection for data set GARCH5.

5.5 Case Study: Early Detection of High Volatility Clusters in IBM's Stock Data

IBM stock price series are usually used as examples for the study of returns series and volatility series (Tsay, 2010). There are various episodes since the year 1962 which are interesting events to explore, including the market falls of 1987 and 2008. As mentioned in Section 4.1.2, the data employed for this case study involves adjusted closing prices between September 12$^{\text{th}}$, 2005 and September 1$^{\text{st}}$, 2009.

The data observed here does not include the ground truth values for volatility, which means that volatility can only be estimated and therefore, there is no possibility to quantify the detector's performance. Analysis is solely based upon observation of the obtained results and qualitative interpretation of the data.

Table 5.15 displays the estimated parameters for the GARCH(1,1) model in the first 200 data points of the series, which serve as the training period. The parameters ω and μ are left fixed in the extended uGARCH model, while estimations of α and β are used as initial conditions for the online estimation of these parameters. The estimation exhibits a very low value for α, while β has a large value. Given that $\alpha + \beta = 0.9881$ and that evidence shows that usually $\alpha + \beta$ is close to 1, one might assume that the estimation is good. Given that the value of α is small, the pseudo-particle standard deviation used is equal to $\sigma_{\alpha,\beta} = 0.04$. Moreover, $R_{th} = 0.7$ and $\sigma_\eta = 0.7$.

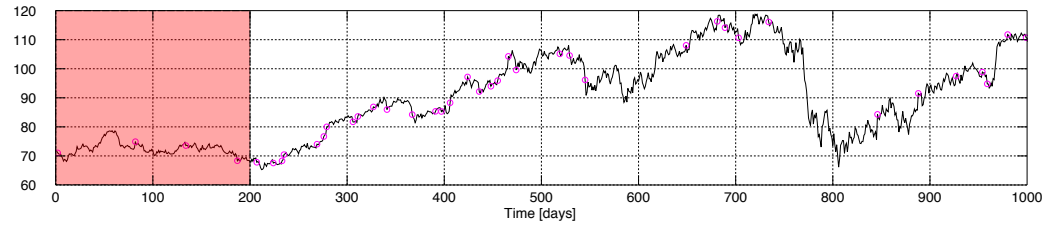
Figure 5.25 exhibits the obtained results from volatility estimation and early detection of high volatility clusters. In particular, details about the adjusted price series, returns series, volatility estimation, detections and the training window may be observed. Analysis of this Figure 5.25c shows that volatility estimation of both the PF and the RSPF are extremely close, and the differences between most estimations occur, although mildy, in sudden volatility rises, where the RSPF has a faster reaction towards unlikely values. This is more visible at the beginning of bigger high volatility clusters, from time step 500 and onwards.

Estimations obtained from the RSPF are used as the base of the hypothesis test-based detector, which showed the best results in the previous sections. One may observe that most of the small high volatility clusters between time steps 200 and 500 are detected. There are some false positives and false negatives, but these are minor. In the time window that includes time steps 500 to 1000, all of the major volatility clusters are detected in a very early stage, including the high volatility cluster starting at time step 750, conducive to the

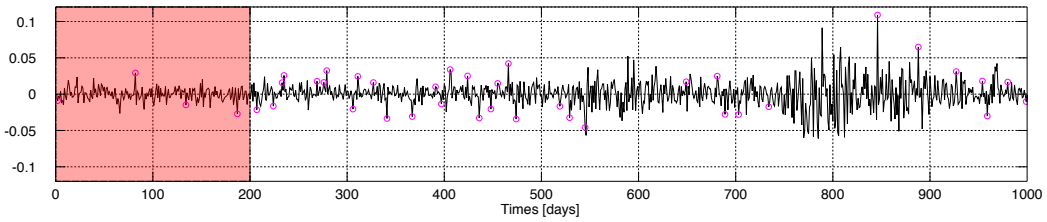
Table 5.15: Parameter estimation of the GARCH(1,1) model through maximum likelihood for IBM's returns series between September 12$^{\text{th}}$, 2005 and September 1$^{\text{th}}$, 2009.

GARCH parameter	Value
ω	2.5690×10^{-6}
α	0.0647
β	0.9234
μ	6.9333×10^{-4}

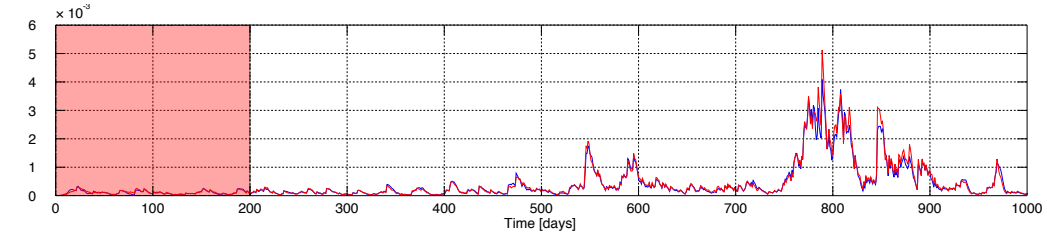
big stock market drop of the year 2008. Moreover, in this time window, there are only 2 false positives, which occur after the last high volatility cluster. All of the other detections need to be considered true positives.



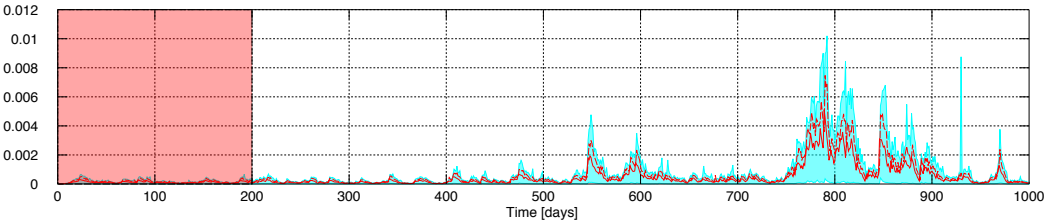
(a) Prices.



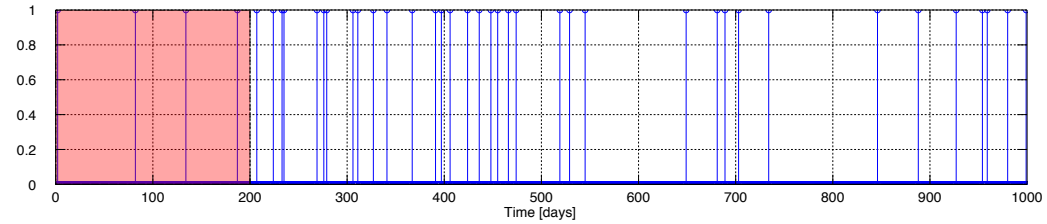
(b) Returns.



(c) Volatility estimation through PF and RSPF.



(d) Confidence interval, prior and posterior estimations for RSPF.



(e) Detections.

Figure 5.25: Early hypothesis test-based detection of high volatility clusters in IBM's stock data.

Chapter 6

Conclusion

This work presents and explores tools commonly used in Electrical Engineering for the early detection of high volatility clusters, to create an algorithm to manage high risk events from an early stage. The development of the proposed detection framework involves several stages that are addressed sequentially, including the creation of simulated volatility time series, the implementation of a Classic Particle Filtering scheme and the proposal of a Risk Sensitive Particle Filter and its posterior implementation, both used as tools for volatility estimation. After parameter selection and subsequent evaluation of estimation accuracy in performance, three proposed algorithms for early detection of high volatility clusters are implemented and tested, both in simulated time series and IBM's stock market data.

The implementation of the Classic Particle Filter and the Risk Sensitive Particle Filter demonstrate that these approaches need attention in the area of volatility estimation, since they can offer excellent estimation performance. Nevertheless, non-observability issues can produce poor estimation results, and this problem needs to be correctly addressed. Analysis of the parameter estimation of the GARCH model together with the sensibility analysis including noise values demonstrate that estimation performance is extremely dependent on four aspects:

1. Correct initial conditions for particle population,
2. Adequate characterization of process noise sources,
3. Correct initial conditions of pseudo-particles if the state-space model has been extended to include online parameter estimation,
4. Adequate characterization of process noise sources for pseudo-particle variability within artificial evolution-based approaches.

Inadequate values can lead to algorithms with inability to learn, or extremely biased estimates. Moreover, there is an important relationship between items 3 and 4: If estimates of the GARCH model (which in this case are used as the initial conditions) are too low or inaccurate, higher noise values for these pseudo-particles are needed to improve the learning capabilities of the PF algorithm. As a consequence, the parameter $\sigma_{\alpha,\beta}$ should be separated into σ_α and σ_β, this is, use a separate dispersion value for the noise process of each pseudo-particle which extends the model.

A performance comparison between the Classic Particle Filter and the proposed Risk Sensitive approach shows that the Risk Sensitive algorithm behaves better for purposes of tracking sudden volatility changes from low to high values. The greater particle resolution offered by the Risk Sensitive Particle Filter in areas of high volatility give this algorithm a very high performance in these cases.

From a detection perspective, the behavior of the Risk Sensitive Particle Filter is extremely important. This filtering approach, combined with the proposed detection technique based on the contrast of prior and posterior estimations of the Risk Sensitive Particle Filter through a hypothesis test proves that early detection of high volatility clusters is possible with a small error. Important aspects associated with the performance ensure that the detection is extremely robust to biased estimates, which are related to sub-optimal dispersion values of noise. In particular, if the Particle Filter does not lose the ability to learn and track the shape of volatility, the proposed hypothesis test-based detector excels in early detection of high volatility clusters.

With respect to the detection techniques based on the comparison of probability density function estimates, the proposed algorithms do not offer accurate detections. The implemented computation of the Kullback-Leibler Divergence produces extremely noisy results, which do not lead to any correlation of its value and volatility behavior. In the case of the Fisher Discriminant, there are a small number of cases where detection is very accurate, but there are many other cases where no detection occurs.

The Particle Filter and Risk Sensitive Particle Filter lead to excellent estimations of volatility if care is taken upon non-observability issues. The proposed detection scheme based on the hypothesis test of prior and posterior estimations of the Risk Sensitive Particle Filter leads to very good results. Nevertheless, the performance approach used to measure the detection error does not capture the effectiveness of the detector and therefore, qualitative analysis should be considered rather than presented confusion matrixes. Setting the limit between high and low volatilities is a complex task, and needs further exploration.

With respect to future work, there are many aspects of this work that need more attention. In a first instance, there are many parameters of the Risk Sensitive Particle Filter (such as the parameters of the Generalized Pareto Distribution) which were not considered in the sensibility analysis. The estimation of the filter and subsequent detection performance of the detector may depend greatly upon these parameters. Moreover, a sensibility analysis of the hypothesis test parameters should also be considered.

The performance measure used in the hypothesis test-based detector needs revision. As discussed earlier, the threshold used to set the limit between high and low variations is diffuse, and needs greater attention. It is extremely important to consider if the volatility rises or lowers throughout the considered time window. This is definitely a challenging problem, where aspects of prediction may be necessary.

On the other hand, the proposed algorithm sets a detection framework that might be introduced in other volatility models. In particular, it would be interesting to experiment with models such as the EGARCH (and stochastic extensions if available). The EGARCH model considers volatility rises and falls asymmetrically, which is a product of the leverage

effect. This is very interesting from a detection and risk management perspective.

Bibliography

M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.

Niclas Bergman. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. PhD thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1999.

Tim Bollerslev. Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 2 edition, 2006.

Arnaud Doucet, Simon Godsill, and Christhophe Andrieu. On Sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Pattern Classification and Scene Analysis: Pattern Classification. Wiley-Interscience, 2nd edition, 2000.

Paul Embrechts, Claudia Kluppelberg, and Thomas Mikosch. *Modelling Extremal Events for Insurance and Finance*, volume 33 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, 1997.

N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, April 1993.

Rudolph E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering*, 82(Series D):35–45, 1960.

Genshiro Kitagawa and Seisho Sato. *Sequential Monte Carlo Methods in Practice*, chapter 9: Monte Carlo Smoothing and Self-Organizing State-Space Model. Statistics for Engineering and Information Science. Springer-Verlag, first edition, 2001.

Augustine Kong, Jun S. Liu, and Wing Hung Wong. Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association*, 89(425):278–288,

March 1994.

Jane Liu and Mike West. *Sequential Monte Carlo Methods in Practice*, chapter 10: Combined Parameter and State Estimation in Simulation-Baes Filtering. Statistics for Engineering and Information Science. Springer-Verlag, first edition, 2001.

Benoit Mandelbrot. The Variation of Certain Speculative Prices. *The Journal of Business*, 36(4):394–419, October 1963.

Simon Maskell and Neil Gordon. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. Technical report, QinetiQ Ltd, 2001.

Mathworks. Generalized Pareto Distribution, 2013. URL `http://www.mathworks.com/help/stats/generalized-pareto-distribution.html`.

Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw Hill, fourth edition, 2002.

José C. Principe. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Information Science and Statistics. Springer, 1st edition, 2010.

Svetlozar Rachev, John Hsu, Biliana Bahasheva, and Frank Fabozzi. *Bayesian Methods in Finance*. The Frank J. Fabozzi Series. John Wiley & Sons, Inc., 1st edition, 2008.

Sebastian Thrun, John Langford, and Vandi Verma. Risk Sensitive Particle Filters. In *In Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

Felipe A. Tobar. Inferencia de Volatilitdad de Retornos Financieros usando Filtro de Partículas. Master's thesis, Departamento de Ingeniería Eléctrica, Universidad de Chile, June 2010.

Ruey Tsay. *Analysis of Financial Time Series*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Chicago, 3rd edition, 2010.

Yahoo!Finance. International Business Machines Corporation - NYSE, May 2013. URL `http://finance.yahoo.com/q?s=IBM&ql=1`.