



Università
Ca' Foscari
Venezia

Master's Degree programme – Second Cycle
(*D.M. 270/2004*)
in Economics

Erasmus Mundus QEM Models and Methods
in quantitative Economics

—
Ca' Foscari
Dorsoduro 3246
30123 Venezia

Final Thesis

Stochastic volatility with big data

Supervisor

Ch. Prof. Roberto Casarin

Graduand

Livia Andreea Covaciu

Matriculation Number 850874

Academic Year

2013/ 2015

CORSO DI LAUREA MAGISTRALE IN ECONOMIA. PROVA FINALE DI
LAUREA
UNIVERSITA CA FOSCARI
QEM- MATHEMATICS MODELS IN QUANTITATIVE
ECONOMICS

STOCHASTIC VOLATILITY WITH BIG DATA

Coordinator:

Prof.

CASARIN ROBERTO

STUDENT:

COVACIU Livia-Andreea

JUNE 2015

Content

1	INTRODUCTION	3
1.1	Some variance models	3
1.2	Variance modeling	6
2	PRESENTATION OF THE MODEL	9
2.1	A stochastic volatility panel model	9
2.2	Bayesian inference	11
3	EMPIRICAL RESULTS	16
3.1	Technical part	16
3.2	Conclusions	42
	Bibliography	44

Abstract

The thesis aims to discuss stochastic volatility when a big amount of data is involved. Therefore I follow Windle and Carvalho (2015) and Casarin (2015) papers where a state-space model for observations and latent variables in the space of positive symmetric matrices is introduced. Moreover, I use Gibbs sample and MCMC method in order to discuss the bayesian inference. One-step ahead and multi-step-ahead forecasting are evaluated because of their importance in economics and business. Since this model can have important applications in finance, one can use realized covariance matrices as data to predict latent time-varying covariance matrices. I present factor-like models, GARCH-like model and univariate stochastic volatility models to give an alternative to the model from the mentioned papers.

It is known that financial markets data often expose volatility clustering, where time series have periods of high volatility and periods of low volatility. As a matter of fact, time-varying volatility appears more than constant volatility, and accurate modeling of time-varying volatility is of great importance, considering economic and financial data. In our case working with a nonlinear model by using MCMC posterior approximation can be a quite challenging issue. Computational time in Monte Carlo simulations is reduced by implementing a parallel algorithm in Matlab which is able to split our database and run the blocks in the same time.

Keywords: forward filter, stochastic volatility, realized covariance, state-space models

Chapter 1

INTRODUCTION

1.1 Some variance models

We briefly review some statistical models for modeling dynamic covariance matrices and a short description about big data.

One can define **big data** as an enormous amount of unstructured data, that is obtained using applications with a high degree of performance, in a big heterogeneous family of applications cases: for instance, scientific computing applications to social network, data obtained from the market and so forth. When data is stored in the underlying layers of the application mentioned (or other) they have some characteristics: we observe that we get large-scale data and this is about the size and distribution, there exists applications that can run big data, it is supporting advanced, meaning that some processes can achieve structured information from it, and we can develop and design some useful and interpretable analytics over it.

Factor like models capture variations in asset returns using multiple linear regression with latent covariates. Moreover, integrating over the covariates yields the marginal covariation of the returns.

Consider the vector of asset returns r_t depends linearly on the covariate x_t such that:

$$r_t = \beta x_t + \epsilon_t, \epsilon_t \sim N(\mu, \Sigma_t).$$

Suppose the regression coefficient β is fixed and the covariates are independent of the error terms, then the marginal variance of r_t is given by:

$$\text{var}(r_t) = \beta \text{var}(x_t) \beta' + \Sigma_t.$$

1.1. SOME VARIANCE MODELS

The individual elements of the vector r_t are independent, but marginally they are correlated. One way to continue is to take known covariates x_t in order to capture as much of the predictable marginal variation in r_t as possible, like in the work of Fama and French (1993) who studies the common risk factors in stock and bond returns and tests whether these shared risks capture the cross-section of average returns. Another way is to use data to infer a set of latent covariates $\{x_t\}_{t=1}^T$. Moreover a prior is placed on the covariates $\{x_t\}_{t=1}^T$ and the errors' variances $\{D_t\}_{t=1}^T$ such that $\text{var}(x_t)$ and D_t are diagonal and change slowly. The latter model is called factor stochastic volatility. But there are a few factors that determine the correlation between elements of r_t , moreover Windle and Carvalho classify this model as factor-like.

David Ruppert, in Statistics analysis for financial engineering (2011), defines a factor model for excess equity returns in the following way:

$$r_{j,t} = \beta_{0,j} + \beta_{1,j}f_{1,t} + \dots + \beta_{p,j}f_{p,t} + \epsilon_{j,t}$$

where $r_{j,t}$ is the return or the excess return on the j th asset at time t , $f_{1,t}, \dots, f_{p,t}$ are factors or risk factors (the state of the financial markets and world economy at time t), and $\epsilon_{1,t}, \dots, \epsilon_{n,t}$ represent the unique risks of the individual stocks and they are uncorrelated (all cross-correlation between the returns is due to the factors), mean-zero random variables. Moreover the factors do not depend on j since they are common to all returns. The parameter $\beta_{i,j}$ is the factor loading and gives the sensitivity of the j th return to the i th factor.

If $p = 1$ and $F_{1,t}$ is the excess return on the market portfolio, then this factor model is the famous CAPM. Here, the market risk factor is the only source of risk besides the unique risk of each asset.

The equation above is a regression model. If j is fixed, then it is a univariate multiple regression model because there is one response (the return on the j th asset) and multiple factors. A multivariate regression model is if we combine these models across j . Multivariate regression is used when fitting a set of returns to factors.

A cross-sectional factor model is a regression model using data from many assets but it is considered only a single holding period. Ruppert in his book talks about two important differences between time series factor models and cross-sectional factor models. The first is that with a time series factor model the parameters are estimated, one asset at a time, using multiple holding periods, but in a cross-sectional model they are estimated, one single holding period at a time, using multiple assets. The second difference is that in the

1.1. SOME VARIANCE MODELS

first kind of model, the factors are directly measured and the loadings are the unknown parameters to be estimated by regression. In a cross-sectional factor model the loadings are directly measured and the factor values are estimated by regression.

GARCH-like model

GARCH is an acronym meaning Generalized AutoRegressive Conditional Heteroscedasticity. We should remember that heteroscedasticity means non-constant variance and homoscedasticity stands for constant variance. Thus GARCH is unconditionally homoscedastic, but conditionally heteroscedastic. Since deficiency of ARCH(p) models is that the conditional standard deviation process has high-frequency oscillations with high volatility coming in short outbreaks.

GARCH models permit more persistent volatility. Let ϵ_t be Gaussian white noise with unit variance. The GARCH(p, q) model is

$$r_t = \sigma_t \epsilon_t$$
$$\sigma_t = \sqrt{\omega + \sum_{i=1}^p \alpha_i a_{t-i}^2 + \sum_{i=1}^q \beta_i \sigma_{t-i}^2}$$

where σ_t is the conditional standard deviation of r_t given the past values r_{t-1}, r_{t-2}, \dots of this process.

Past values of the σ_t process are reintroduced in the present value, therefore the conditional standard deviation can display more persistent periods of high or low volatility than seen in an ARCH process. ARCH models are a special case of GARCH models (β_i are zeros).

In the definition of GARCH model, if $\sum_{i=1}^p \alpha_i + \sum_{i=1}^q \beta_i = 1$ then the process is called Integrated GARCH or I-GARCH. This kind of process is either non-stationary or have an infinite variance.

Moreover until now researchers have found that stock returns have "heavy-tailed" or "outlier-prone" probability distributions. The stock returns have outliers because the conditional variance is not constant, and this event occur when the variance is large. Actually, GARCH processes can have heavy tails even if $\{\epsilon_t\}$ is Gaussian. In spite of that, a lot of financial time series have tails that are heavier than implied by a GARCH process with Gaussian $\{\epsilon_t\}$. To handle such data, one can assume that, instead of being Gaussian white noise, $\{\epsilon_t\}$ is an i.i.d. white noise process with a heavy-tailed distribution.

Univariate stochastic volatility model

A model with an unobserved variance component, for which the logarithm is modeled directly as a linear stochastic process, like an autoregression is called stochastic volatility models or stochastic variance(SV) models (Harvey, Ruiz and Shepard 1993). They are very used in finance in order to account for time varying volatility and clustering phenomena in volatility. SV models are the natural discrete-time versions of the continuous-time models developed in finance theory (Black-Scholes result on option pricing). The main disadvantage is that it is difficult to estimate them using maximum likelihood. From statistical point of view, an advantage is that their properties can be easily obtained from the properties of the process generating the variance component.

A simple stochastic volatility model, proposed by Taylor (1987) is the following one:

$$\begin{aligned} r_t &= e^{x_t/2} \epsilon_t, \epsilon_t \stackrel{i.i.d}{\sim} N(0, \sigma_\epsilon^2) \\ x_t &= \phi x_{t-1} + \eta_t, \eta_t \stackrel{i.i.d}{\sim} N(0, \sigma_\eta^2) \end{aligned}$$

where r_t is a sequence of observable variables (e.g. the log-return of a financial asset) with $t = 1, \dots, T$, and x_t a latent variable (called stochastic log-volatility process), $x_0 = 0$ and the independence assumption is $\epsilon_t \perp \epsilon_s, \forall s, t$ on the disturbance terms.

1.2 Variance modeling

It is known that Bayesian analysis of a state-space model includes computing the posterior distribution of the system's parameters and its latent states. An important topic is when the latent state is in \mathbb{R}^n or in a manifold embedded here (A manifold is defined as a topological space that is locally Euclidean, i.e., around every point, there is a neighborhood that is topologically the same as the open unit ball in \mathbb{R}^n). Different tools are used but when we refer to the second case the things got complicated. We are interested in state-space models whose latent states are covariance matrices because those have applications in finance. In this thesis I will follow the paper of Windle and Carvalho (2015) and Casarin (2015) where they present a state-space model whose observations and latent states are in the manifold of symmetric positive-definite matrices. Here it will be easier to compute the posterior distribution of the latent states and the system's parameters also the filtered

distributions and one-step ahead predictions. Given the fact that we are interested in the applications this model can have in finance, one can use realized covariance matrices as data to predict latent time-varying covariance matrices. According to the authors this idea out-performs factor stochastic volatility. We will give forward explanation of these terms.

State space model (SSM) is a probabilistic model that gives us the dependence between the latent state variables and the observed measurements (continuous or discrete). The term state space comes from control engineering (1960s). SSM gives a general framework for analyzing deterministic and stochastic dynamical systems that are measured or observed through a stochastic process. An example of SSM are the Hidden Markov models (HMMs) (Rabiner, 1989).

Another well studies SSM is the linear and Gaussian SSM. Inference on the hidden state is done by applying KALMAN FILTER.

Kalman filtering- linear quadratic estimation (LQE), is an algorithm that uses a series of measurements observed over time, containing noise (random variations) and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone. The Kalman filter operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. Under some conditions, this filter implement a predictor-corrector type estimator that minimizes the estimated error covariance.

The Kalman filter offers a prediction of a system's state with a new measurement using a weighted average. Weights are used because values with better estimated uncertainty are "trusted" more. They are computed from the covariance. The result of the weighted average is a new state estimate that lies between the predicted and measured state, and has a better estimated uncertainty than either alone. This process is repeated every time step, with the new estimate and its covariance informing the prediction used in the following iteration. Thus Kalman filter works recursively and requires only the last "best guess", and not the entire history, of a system's state to obtain a new state. Since is difficult to measure precisely one can observe the filter's behavior in terms of gain. The Kalman gain, which is a function of the relative certainty of the measurements and current state estimate, can achieve particular performance. The higher the gain the more weights are placed on measurements. With a low gain, the filter follows the model predictions more closely, smoothing out noise but decreasing the responsiveness. We shall remark the Kalman-like filter developed in the paper of Tyagi and

1.2. VARIANCE MODELING

Davis (2008) since it is used for symmetric positive-definite matrices.

Also we can see in Hauberg, Lauze and Peterson (2012) paper that they generalize the unscented transform and then the unscented Kalman filter to Riemannian manifolds. (The unscented transform is defined as a method for estimating the mean and covariance of a distribution of a non-linear transformation.)

Following the notation used in Windle and Carvalho(2015) 's paper, SSM is given by an observation density $f(y_t|x_t)$ for the responses $\{y_t\}_{t=1}^T$ and a transition density $g(x_t|x_{t-1})$ for the latent states $\{x_t\}_{t=1}^T$ with $x_t \in \mathbb{R}^n$. For our application to financial asset volatility it is good to consider a state-space model that has covariance matrix-valued responses (the statistics) characterized by $f(Y_t|V_t)$ and covariance matrix-valued latent quantities (the time-varying covariance matrices) characterized by $g(V_t|V_{t-1})$ because symmetric positive-definite matrix-valued statistics can be seen as noisy observations of the conditional covariance matrix of a vector of daily asset returns. In order to make methods easier filtering is used. Filtering or forward filtering means to iteratively derive the distributions $p(x_t|D_t, \theta)$ where D_t is the data $\{y_s\}_{s=1}^t$ and θ states for system's parameters.

However, to do Bayesian analysis of a state-space model means more than forward filtering. One need to be able to sample from posterior distribution of the latent states and the system's parameters. Let $S_{m,k}^+$ denote the set of order m, rank k, symmetric positive semi-definite matrices and let S_m^+ denote the set of order m, symmetric positive-definite matrices. Uhlig (1997) showed how to construct a state-space model with $S_{m,1}^+$ observations and S_m^+ hidden states and how one can forward filter in closed form.

Windle and Carvalho's (2015) approach is to extend the Uhlig (1997) model and to consider observations of arbitrary rank. They show how to forward filter, how to backward sample, and how to marginalize the hidden states, this avoiding Markov chain Monte Carlo (MCMC) schemes. Backward sampling means to take a joint sample of the posterior distribution of the latent states $p(\{x_t\}_{t=1}^T|D_T, \theta)$ using the conditional distributions $p(x_t|x_{t+1}D_T, \theta)$. The estimates and one-step ahead predictions of the model are exponentially weighted moving averages, known to provide simple and robust estimates and forecasts in many settings. The authors find that this will happen within multivariate volatility forecasting in finance.

Chapter 2

PRESENTATION OF THE MODEL

2.1 A stochastic volatility panel model

Taking into consideration Windle and Carvalho's (2015) paper, this report explores the state-space model with observations and latent states in the space of symmetric positive-definite matrices. Moreover, covariance matrices are used in order to predict latent time-varying covariance matrices. The model in their paper is based on Uhlig's (1997) work. Briefly, his model can be defined as:

$$\begin{cases} r_t \sim N(0, X_t^{-1}) \\ X_t = T_{t-1}' \Psi_t T_{t-1} / \lambda \quad \Psi_t \sim \beta_m(n/2, 1/2) \\ T_{t-1} = \mathcal{U}(X_{t-1}) \end{cases} \quad (1)$$

where $r_t \in \mathbb{R}^m, t = 1, \dots, T$ are the observations, $\{X_t\}_{t=1}^T \in S_m^+$ are the hidden states and \mathcal{U} stands for the upper-triangular Choleski factor (i.e. the upper triangular matrix T with positive diagonal elements which satisfies $X_{t-1} = T' T$). Here the mean of r_t is zero; $n > m - 1$ and β_m is the multivariate beta distribution given by the definition 1 in Uhlig's (1994b) paper:

Definition 1. β_m distribution

A random variable $X \sim \beta_m(n/2, k/2)$ if it can be written as $X = \mathcal{U}(A + B)'^{-1} A \mathcal{U}(A + B)^{-1}$, where $A \sim W_m(n, I_m)$ and $B \sim W_m(k, I_m)$ with A represented as $\sum_{j=1}^n r_j r_j'$, $r_j \sim N(0, \Sigma)$ and $\Sigma = I_m$ and the $Y_j, j = 1, \dots, n$, independent from B . $X \sim \beta_m(n/2, k/2)$ if $A \sim W_m(k, I_m)$ and $B \sim W_m(n, I_m)$ instead.

2.1. A STOCHASTIC VOLATILITY PANEL MODEL

Definition 2. Wishart distribution

Let $m, n, k \in \mathbb{Z}$. A random variable Y , with values in the space of positive semi-definite symmetric matrices, has a Wishart distribution $W_m(k, V)$ for $k \in \mathbb{N}$ and $V \in S_m^+$ if

$$Y \sim \sum_{j=1}^k \nu_j \nu_j'$$

where $\nu_j \stackrel{iid}{\sim} N(0, V), j = 1, \dots, k$.

When $k > m - 1$, the density for the Wishart distribution is

$$\frac{|Y|^{(k-m-1)/2}}{2^{mk/2} \Gamma_m(k/2) |V|^{k/2}} \exp\left(\text{tr} - \frac{1}{2} V^{-1} Y\right)$$

(Muirhead 1982) considering the volume element $(dY) = \bigwedge_{1 \leq i \leq j \leq m} dY_{ij}$.

When $k \leq m - 1$ and Y is rank deficient, the density for the Wishart distribution is

$$\frac{\pi^{-(mk-k^2)/2} |L|^{(k-m-1)/2}}{2^{mk/2} \Gamma_k(k/2) |V|^{k/2}} \exp\left(\text{tr} - \frac{1}{2} V^{-1} Y\right)$$

where the volume element in this case is

$$dY = 2^{-k} \prod_{i=1}^k l_i^{m-k} \prod_{i < j}^k (l_i - l_j) (H_1' dH_1) \wedge \bigwedge_{i=1}^k dl_i,$$

$Y = H_1 L H_1'$, H_1 is a matrix of orthonormal columns of order $m \times k$, and $L = \text{diag}(l_1, \dots, l_k)$ with decreasing positive entries (Windle and Carvalho-Wishart definition).

Thus when $k = 1$ and $r_t \sim N(0, 1)$ then $y_t = r_t^2$ follows a $Y_t \sim W_m(1, X_t^{-1})$, thar is a $Y_t \sim \chi_1^2$.

The authors consider the following univariate UE model, which is the extended version of Uhlig's (1997) model (there the degree of freedom is $k=1$ for the Wishart distribution of order m):

$$\begin{cases} Y_t \sim W_m(k, (kX_t)^{-1}) \\ X_t \sim T_{t-1}' \Psi_t T_{t-1} / \lambda & \Psi_t \sim \beta_m(n/2, k/2) \\ T_{t-1} = \text{upper chol} X_{t-1} \end{cases} \quad (2)$$

We consider y_{it} to be the observable variable equal to r_{it}^2 . For $m=1$ the model becomes:

$$\begin{cases} y_{it} \mid x_{it} \sim Ga(k/2, (kx_{it})^{-1}/2) \\ x_{it} \mid x_{it-1} \sim x_{it-1} \psi_t / \lambda & \psi_t \sim Be(n/2, k/2) \end{cases} \quad (3)$$

2.2. BAYESIAN INFERENCE

for $t = 1, \dots, T$, i is for every asset, with $x_0 = x$; y_{it} follows a gamma distribution given by the probability density function

$$f(y) = \Gamma\left(\frac{k}{2}\right)^{-1} \left(\frac{kx_{it}}{2}\right)^{\frac{k}{2}} e^{-\frac{kx_{it}}{2}y} y^{\frac{k}{2}-1} \quad (4)$$

and beta probability density is

$$f(x) = \frac{\Gamma\left(\frac{n+k}{2}\right)}{\Gamma\left(\frac{n}{2}\right)\Gamma\left(\frac{k}{2}\right)} x^{\frac{n}{2}-1} (1-x)^{\frac{k}{2}-1}. \quad (5)$$

2.2 Bayesian inference

Since y is thought of as the actual data that we have gathered the likelihood is a function of the parameters n, k, λ, x_t with y as a constant.

For our case, using the information from above we get the following likelihood:

$$L(y_{i1}, \dots, y_{iT} \mid x_1, \dots, x_T, n, k, \lambda) = \prod_{t=1}^T \frac{1}{\Gamma(k/2)} \left(\frac{kx_{it}}{2}\right)^{\frac{k}{2}} e^{-\frac{kx_{it}}{2}y_{it}} y_{it}^{\frac{k}{2}-1} \cdot \left(\frac{\lambda}{x_{it-1}}\right)^{\frac{n+k}{2}-1} \frac{\Gamma\left(\frac{n+k}{2}\right)}{\Gamma\left(\frac{n}{2}\right)\Gamma\left(\frac{k}{2}\right)} \psi_t^{\frac{n}{2}-1} \left(\frac{x_{it-1}}{\lambda} - \psi_t\right)^{\frac{k}{2}-1}.$$

We obtain this big formula because the likelihood is the product of the density functions from the model that we considered.

Because this likelihood is not a easy one, the usual method doesn't work; I am referring to the ordinary least squares (OLS) estimates, or maximum likelihood. Even if we take the *log*-likelihood is not that easy. Thus MCMC method and Gibbs samples will be utilized.

Next I want to generate samples from the posterior distribution and I apply Gibbs sampling algorithm. Let's consider the notation: $y = (y_1, \dots, y_T)$, $x = (x_1, \dots, x_T)$ and $x_{-t} = (x_1, \dots, x_{t-1}, x_{t+1}, \dots, x_T)$ describes the full conditional distributions in the Gibbs sampler. A Metropolis-Hastings will be used to generate samples from the full conditional of the latent process.

We assume the following prior distribution for the parameters n, k, λ :

$$\begin{cases} n \sim \exp\{\alpha\} \\ k \sim \exp\{\beta\} \\ \lambda \sim \exp\{\gamma\} \end{cases}$$

2.2. BAYESIAN INFERENCE

uniform prior on interval $[0, 1]$ for λ and flat priors on \mathbb{R}_+ for n and k .

Knowing that posterior distribution is the product of the likelihood and the priors distributions one can get:

$$\begin{aligned}
L(y_1, \dots, y_T \mid x_1, \dots, x_T, n, k, \lambda) &= \\
\prod_{t=1}^T \frac{1}{\Gamma(k/2)} \left(\frac{kx_t}{2} \right)^{\frac{k}{2}} e^{-\frac{kx_t}{2} y_t} y_t^{\frac{k}{2}-1} \cdot \left(\frac{\lambda}{x_{t-1}} \right)^{\frac{n+k}{2}-1} \frac{\Gamma(\frac{n+k}{2})}{\Gamma(\frac{n}{2}) \Gamma(\frac{k}{2})} \psi_t^{\frac{n}{2}-1} \left(\frac{x_{t-1}}{\lambda} - \psi_t \right)^{\frac{k}{2}-1} \\
&\quad \cdot \alpha e^{-\alpha n} \cdot \beta e^{-\beta k} \cdot \gamma e^{-\gamma \lambda} \\
&= \left(\frac{k}{2} \right)^{\frac{Tk}{2}} \left(\frac{x_T}{x_0} \right)^{\frac{k}{2}} e^{-\frac{k}{2} \sum_{t=1}^T x_t y_t} \left(\frac{1}{\Gamma(k/2) B(n/2, k/2)} \right)^T \alpha \beta \gamma e^{-(\alpha n + \beta k + \gamma \lambda)} \\
&\quad \cdot \prod_{t=1}^T y_t^{\frac{k}{2}-1} \frac{\psi_t}{x_{t-1}} \left(\frac{x_{t-1}}{\lambda} - \psi_t \right)^{\frac{k}{2}-1}
\end{aligned}$$

Next I will write the full conditionals.

Full conditional for n :

$$\pi(n \mid \lambda, k, y, x) \propto \prod_{t=1}^T \frac{\lambda^{n/2} \psi_t^{n/2-1}}{B(n/2, k/2)} e^{-\alpha n} \alpha$$

Full conditional for λ :

$$\pi(\lambda \mid n, k, y, x) \propto \gamma e^{-\gamma \lambda} \prod_{t=1}^T \lambda^{\frac{n+k}{2}-1} \left(\frac{x_{t-1}}{\lambda} - \psi_t \right)^{k/2-1}$$

Full conditional for k :

$$\begin{aligned}
\pi(k \mid n, \lambda, y, x) &\propto \left(\frac{k}{2} \right)^{\frac{Tk}{2}} \left(\frac{x_T}{x_0} \right)^{\frac{k}{2}} e^{-\frac{k}{2} \sum_{t=1}^T x_t y_t} \left(\frac{1}{\Gamma(\frac{k}{2}) B(\frac{n}{2}, \frac{k}{2})} \right)^T \beta e^{-(\beta k + \gamma \lambda)} \\
&\quad \cdot \prod_{t=1}^T y_t^{\frac{k}{2}-1} \left(\frac{x_{t-1}}{\lambda} - \psi_t \right)^{\frac{k}{2}-1}
\end{aligned}$$

2.2. BAYESIAN INFERENCE

Full conditional for $x_t, t = 2, \dots, T - 1$:

$$\pi(x_t \mid n, \lambda, k, y, x_{-t}) \propto e^{-\frac{k}{2} \sum_{t=1}^T x_t y_t} \prod_{t=1}^T \frac{1}{x_{t-1}} \left(\frac{x_{t-1}}{\lambda} - \psi_t \right)^{n/2-1}$$

Full conditional for x_1 :

$$\pi(x_1 \mid n, \lambda, k, y, x_{-1}) \propto e^{-\frac{k}{2} x_1 y_1} \left(\frac{1}{x_1} \right)^{n/2-1} \left(\frac{x_1}{\lambda} - \psi_1 \right)^{k/2-1}$$

Full conditional for x_T :

$$\pi(x_T \mid n, \lambda, k, y, x_{-T}) \propto x_T^k / 2e^{k/2 x_T y_T}.$$

Let's consider $D_t = \{y_t\} \cup D_{t-1}$, for $t = 1, \dots, T$, with $D_0 = \{\sigma^2\}$ where σ^2 is the variance. Consider the prior, like in Windle and Carvalho's paper, but for case $m = 1$, for $(x_1 \mid D_0)$ be $Ga(\frac{n}{2}, \frac{\lambda k \sigma_0^2}{2})$. Proposition 1 from their paper becomes:

Proposition 3. Suppose $x_t \mid D_{t-1} \sim Ga(\frac{n}{2}, \frac{\lambda k \sigma_{t-1}^2}{2})$. After observing y_t , the updated distribution is:

$$x_t \mid D_t \sim Ga\left(\frac{n+k}{2}, \frac{k \sigma_t^2}{2}\right)$$

where $\sigma_t^2 = \lambda \sigma_{t-1}^2 + y_t$. One step ahead leads to $x_{t+1} \mid D_t \sim Ga(\frac{n}{2}, \frac{\lambda k \sigma_t^2}{2})$.

Proposition 4. The joint distribution of $\{x_t\}_{t=1}^T \mid D_t$ can be decomposed as

$$p(x_t \mid D_t) \prod_{t=1}^{T-1} p(x_t \mid x_{t+1}, D_t) \text{ where:}$$

$$(x_t \mid x_{t+1}, d_t) \sim \lambda x_t + z_{t+1}, \quad z_{t+1} \sim Ga\left(\frac{k}{2}, \frac{k \sigma_t^2}{2}\right).$$

Proposition 5. The joint distribution of $\{y_t\}_{t=1}^T$ is given by

$$p(\{y_t\}_{t=1}^T \mid D_0) = \prod_{t=1}^T p(y_t \mid D_{t-1}).$$

Since $m = 1$ ($k > m - 1$) we obtain

$$p(y_t \mid d_{t-1}) = \frac{1}{Be(\frac{n}{2}, \frac{k}{2})} \frac{y_t^{\frac{k}{2}-1} c_t^{\frac{n}{2}}}{(c_t + y_t)^{\frac{n+k}{2}}}$$

with respect to $(dY) = \bigwedge_{1 \leq i \leq j \leq m} dY_{ij}$, $c_t = \lambda \sigma_{t-1}^2$, $\sigma_t^2 = \lambda \sigma_{t-1}^2 + y_t$.

2.2. BAYESIAN INFERENCE

Moreover, using the law of iterated expectations, one finds that $\mathbb{E}[y_{t+1} | D_t] = \mathbb{E}[X_{t+1}^{-1} | D_t]$. Since $(x_{t+1}^{-1} | D_t) \sim \text{Inv} - \text{Gamma}(n/2, (\frac{2}{k\lambda\sigma_t^2}))$ is an inverse gamma distribution, its expectation is $\frac{\frac{k\lambda\sigma_t^2}{2}}{(\frac{n}{2}-1)}$. Solving the recursion

for σ_t^2 one can get $\sigma_t^2 = \sum_{i=0}^{t-1} \lambda^i y_{t-i} + \lambda^t \sigma_0^2$.

If one considers $\frac{1}{\lambda} = 1 + \frac{k}{n-2}$ then $\mathbb{E}[(x_t)^{-1} | D_t] = \mathbb{E}[(x_{t+1})^{-1} | D_t]$ (the latent state's mean is not changed).

The one step ahead point forecast of $(x_{t+1}^{-1} | D_t)$ is

$$\mathbb{E}[x_{t+1}^{-1} | D_t] = \frac{k\lambda\sigma_t^2}{n-2} = (1-\lambda)\sigma_t^2 = (1-\lambda) \sum_{i=0}^{t-1} \lambda^i y_{t-i} + (1-\lambda)\lambda^t \sigma_0^2.$$

What we can understand from the above equation is that the one step ahead forecast is the geometrically weighted average of past observations, in the constrained case. for this kind of average, the most recent observations are given more weight as λ decreases. This will provide decent one-step ahead forecast.

In addition to this idea, Carvalho and Windle showed that the one-step ahead predictions of the covariate matrix of daily returns of the assets, when realized covariance matrices are used as data, performs better than factor stochastic volatility model. Also, the parameter λ is important because it controls how much the model smooths observations when forming estimates and one-step ahead predictions, therefore we don't want to have constraints on λ . In this model, this parameter can take any value.

Now let see this model from another point of view. In chapter 14 of West and Harrison (1997)'s book is defined a exponential family state-space model. For the univariate case we will have:

$$p(y_t | \eta_t, v_t) = \exp\{v_t^{-1}(y(y_t)\eta_t - a(\eta_t))\}b(y_t, v_t)$$

which is the density for the observations y_t .

If we consider:

$$\begin{aligned} \eta_t &= x_t \\ v_t &= -2k \\ y(y_t) &= k^2 y_t \\ a(\eta_t) &= k^2 \log \eta_t = k^2 \log x_t \\ b(y_t, v_t) &= \frac{y_t^{-\frac{v_t}{4}} - 1}{\Gamma(-\frac{v_t}{4})(-\frac{v_t}{4})^{\frac{v_t}{4}}} \end{aligned}$$

2.2. BAYESIAN INFERENCE

we get the same density function as in (4).
Also they give the state posterior distributions

$$p(\eta_t \mid D_{t-1}) = \exp\{(r_t \eta_t - s_t a(\eta_t))\} c(r_t, s_t).$$

In our case we have to consider

$$r_t = -\frac{\lambda k \sigma_{t-1}^2}{2}$$

$$s_t = \frac{n}{2}, \quad c(r_t, s_t) = \frac{r_t^{s_t}}{\Gamma(s_t)}$$

in order to get the distribution of $x_t \mid D_{t-1}$ like in proposition 3.

From the properties of this family of exponential models one can remember that:

- a) η_t is the natural parameter of the distribution $p(y_t \mid \eta_t, v_t)$ a continuous quantity.
- b) $v_t > 0$ is a scale parameter and the precision parameter of the distribution is defined as $\phi_t = v_t^{-1}$.
- c) As a function of the natural parameter for fixed y_t , the density $p(y_t \mid \eta - t, v_t)$ can be viewed as a likelihood for η_t , and also depends on y_t through the transformed value $y(y_t)$.
- d) The function $a(\eta_t)$ is assumed twice differentiable in η_t . Moreover, the mean function is:

$$\mu_t = E[y(y_t) \mid \eta_t, v_t] = \dot{a}(\eta_t).$$

- e) The variance function

$$V[y(y_t) \mid \eta_t, v_t] = v_t \ddot{a}(\eta_t).$$

Chapter 3

EMPIRICAL RESULTS

3.1 Technical part

For illustrating the results of this model about how it behaves when big amount of financial data is involved I consider a panel of 12,933 assets of the US stock market and collect prices at a daily frequency from 29 December 2000 to 22 August 2014, which is 3,562 time observations, from Fame, (2012) Fame. Bureau van Dijk. [Online] ,available at: <http://fame2.bvdep.com/> . Then I computed logarithmic returns for all stocks and obtain a dataset of 46,067,346 observations, which represents a big data context where fitting a nonlinear model by using MCMC posterior approximation can be a quite challenging issue. In this work, in order to control for the liquidity of the assets and consequently for long sequences of zero returns, we impose that each stock has been traded a number of days corresponding to at least 40% of the sample size. After cleaning the dataset we obtain 6,799 series and the size of the dataset reduces to 2,846,038.

In this section I will focus on the analysis of the three parameters from our model: k , n and λ . Therefore I will show how the posterior distribution of them changes over iterations, or when we change the number of daily observations or when we use some specific methods to obtain a better computing time. Furthermore, I will focus also on computing time to see how fast or slow this model is for different values of size of the sample.

In order to show how our data looks like, I will consider, as in the previous chapter, a uniform prior on $[0, 1]$ for λ and a flat prior on \mathbb{R}^+ for n and k .

3.1. TECHNICAL PART

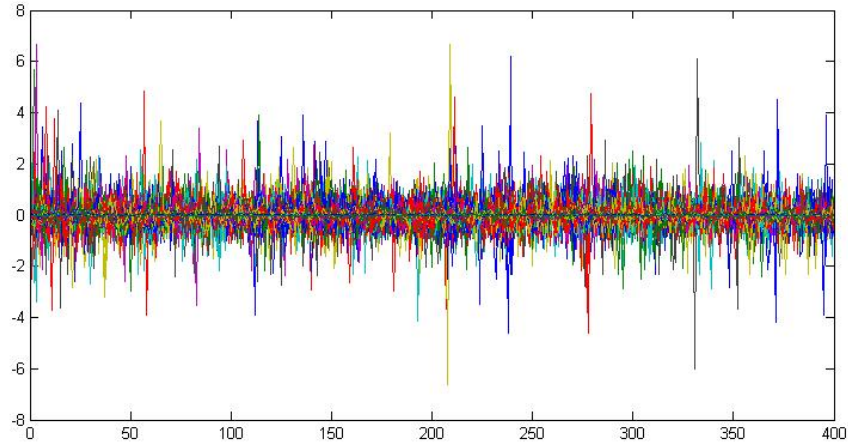


Figure 1: Plot of the dataset after cleaning. Observable variable y_{it} , which represents the logarithmic returns of daily prices, is obtained here.

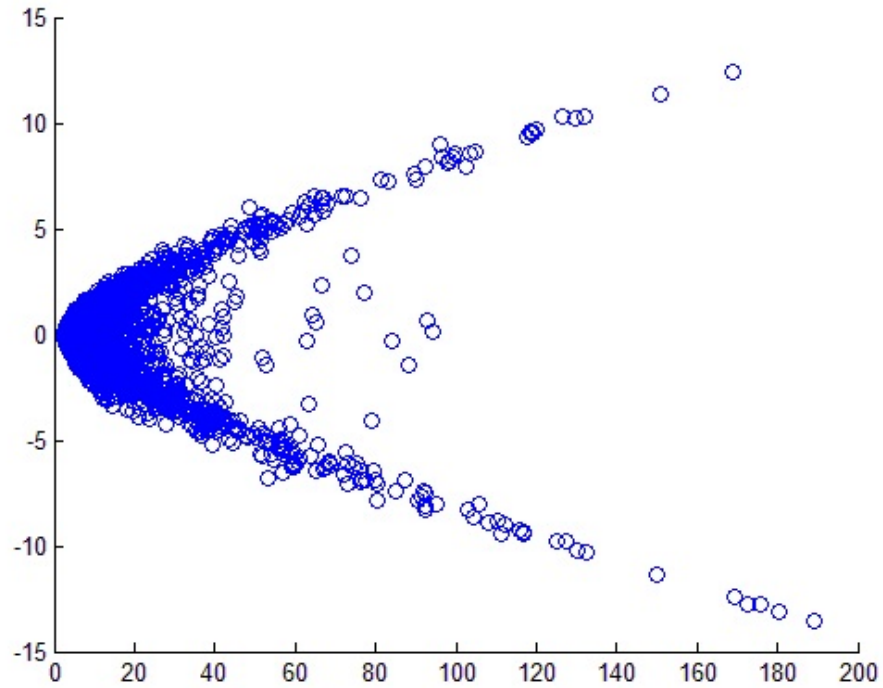


Figure 2: Scatter plot of the sample skewness and kurtosis. One can observe that between 0 and 40 observations there is a concentration of both variables centered in 0 on the vertical axes. From this figure, it is obvious

3.1. TECHNICAL PART

that the distribution of our data has the widest combination of skewness and kurtosis for values of kurtosis less than 40.

Furthermore, here high-dimensional data can be very challenging to analyze. They are difficult to visualize, need extensive computer resources and often require special statistical methodology. Fortunately, one can use dimension reduction techniques because in many practical applications, high-dimensional data have most of their variation in a lower-dimensional spaces.

Ruppert in his book (Statistics and data analysis for financial engineering) gives some methods designed for dimension reduction, such as, factor analysis and principal components analysis, often called PCA. The method PCA finds structure in the covariance or correlation matrix and uses this structure to locate low-dimensional subspaces containing most of the variation in the data. About factor analysis the Ruppert talks how this method explains returns with a smaller number of fundamental variables called factors or risk factors. According to him, factor analysis models can be classified by the types of variables used as factors (macroeconomic or fundamental) and by the estimation technique, time series regression, cross-sectional regression, or statistical factor analysis.

It is known that financial markets data often expose volatility clustering, where time series have periods of high volatility and periods of low volatility. As a matter of fact, time-varying volatility appears more than constant volatility, and accurate modeling of time-varying volatility is of great importance, considering economic and financial data.

I have to mention that I run the code in Matlab 2014a, on my laptop Lenovo with processor: Intel(R) Core(TM) i7-4710MQ CPU@ 2.50GHz and a 64-bit Operating System. Using the 64-bit version of MATLAB drastically increases the amount of data held in memory. We are interested to see how our data behaves to this model and how long does it takes to obtain the results. I also consider every 200's asset because I need to thin the dataset.

As I said before, we need to clean the data set and to make it more accurate. To do this, I first count the zero values on every line, then I divide this number by the total number of columns in our data. In this case I consider the ratio greater than 60% because is more appealing for economic analysis. Later on the paper I consider a case where the ratio is greater than 30% only. Our data is registered in *mOutputClear*.

3.1. TECHNICAL PART

```
[cN, cT] = size(Output);
mOutputClear = zeros(cN, cT);
for i = 1 : cN
    iNumZeros = sum(Output(i, :) == 0);
    iPerZeros = iNumZeros/cT;
    if iPerZeros > 0.60 %illiquidity rate
    else
    mOutputClear(i, :) = Output(i, :);
end
end
```

Case 1:

For the analysis, first I will compare the computing time when we run the code using `parpool` function in Matlab, which help us to achive results faster than running the code normally. This function enables the full functionality of the parallel language features (`parfor` and `spmd`) in MATLAB by creating a special job on a pool of workers, and connecting the MATLAB client to the parallel pool. Simple `parpool` starts a pool using the default cluster profile, with the pool size specified by your parallel preferences and the default profile. I use `parfor` which executes a series of statements for values of i between 0 and 202 , inclusive, which specify a vector of increasing integer values. The loop runs in parallel when you have `parpool` function. Unlike a traditional for-loop, iterations are not executed in a guaranteed order. I used this version: `parpool(6)` which overrides the number of workers (6 in this case) specified in the preferences or profile, and starts a pool of exactly that number of workers, even if it has to wait for them to be available.

I considered a panel dataset with time window of 200 daily observations and 400 series, thus one estimate for the whole sample. Using this parallel function I obtained the following results:

Iteration 1 CPU Time 4.446 and total time 1319s.

Without parallel pooling, I get:

Iteration 1 CPU Time 4951.9865, total time 4967.305 s.

Thus in the first case we get the results in 21 minutes and in the latter in 1 hour and 20 minutes. Also as we can see the results are similar and for both cases we have only one iteration.

3.1. TECHNICAL PART

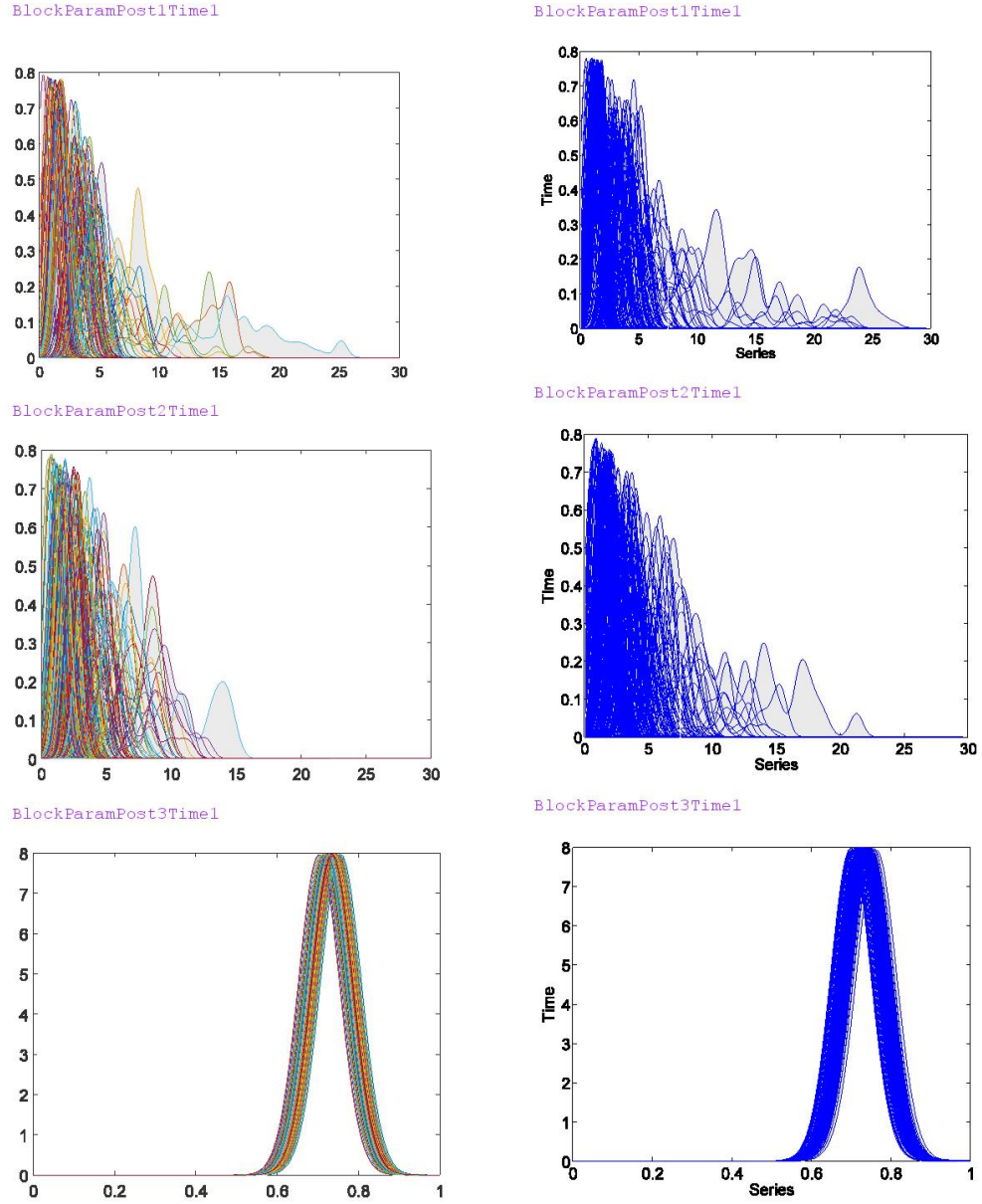


Figure 3: Posterior distribution for the parameters k , n , λ . The first column is obtained when parallel pooling is used, the second column is when this function is omitted. In this case, the posterior distribution are

3.1. TECHNICAL PART

obtained by rolling analysis with one estimate for the whole sample. Moreover, the results are similar in both cases, but the computing time is much longer in the latter case.

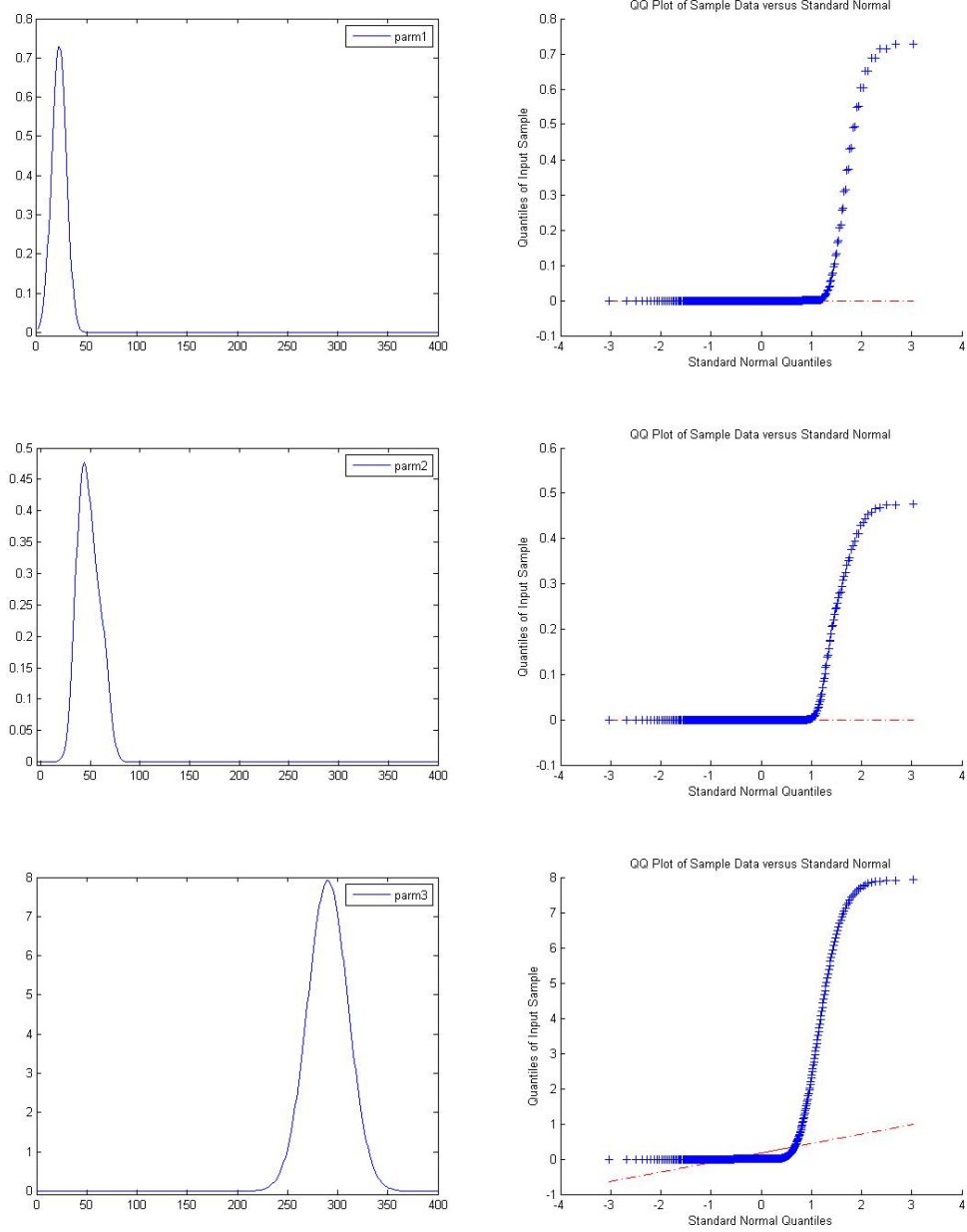


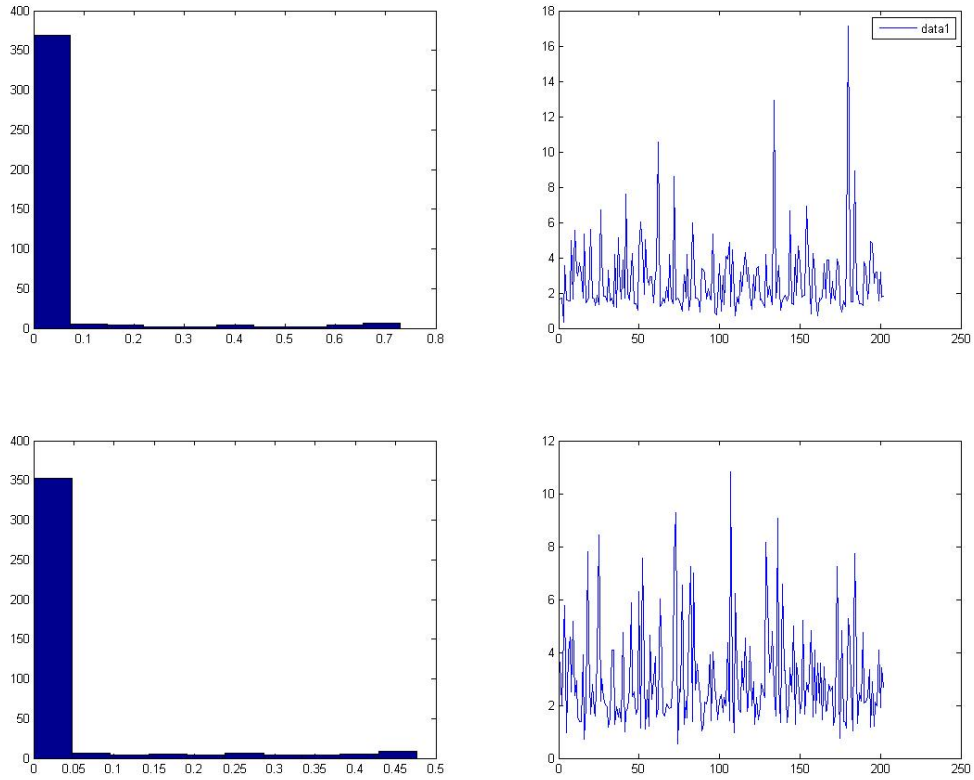
Figure 4: Plots of the parameters considering first asset line only (on the left side) and quantile-quantile plots for each (on the right side) which compares the parameter's quantile with the one corresponding to a normal

3.1. TECHNICAL PART

distribution. The graphs are obtained using parallel pool (thus we are in the first case).

parameters	skewness	kurtosis
k	4.2148	20.1578
n	3.1811	12.0926
λ	2.1711	6.3421

Figure 4 shows that the first parameter has a long tail on the right since the plotted points appear to bend up and to the left of the normal line and it is centered in 25. For the second parameter, n , we have again a long tail on the right for the same reasons as before but a little bit smaller than the tail for the previous parameter, k ; it is centered in 50. The last graph is showing that the third parameter is centered in 300 and has a long tail on the left. Those results were obtained considering the priors for k, n, λ from Chapter 2.



3.1. TECHNICAL PART

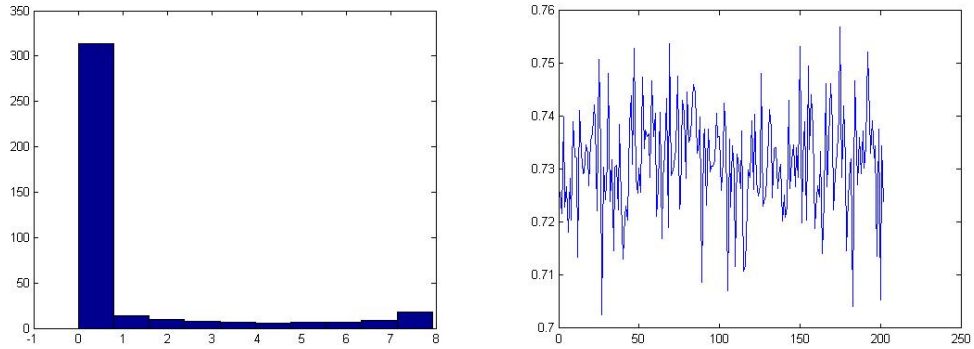
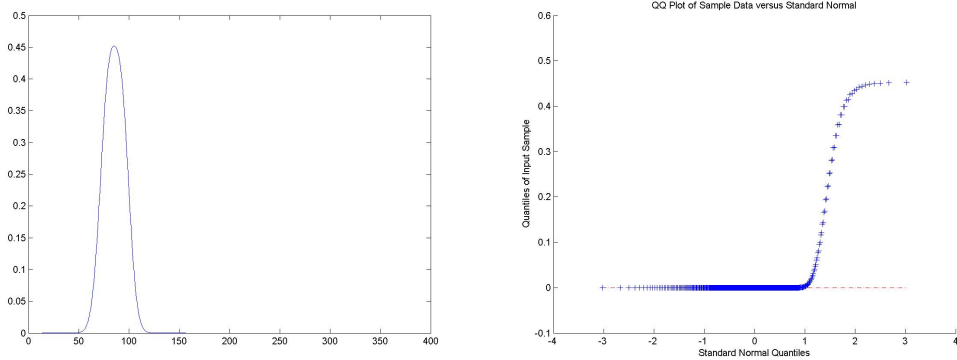


Figure 5: Histograms on the left column and progressive posterior mean over time of for each parameter, for univariate (UE) model.

The left column from Figure 5 proves again what we said before about k , n , and λ . One can say that histograms look unimodal and skewed right. On the right part of the figure I plotted the progressive mean for the cumulated parameters.

Case 2:

Let's move on the case where we don't use parallel pooling. I will consider the same number of observations and series as above, but I will delete *parpool* function and instead of *parfor* I will use only *for* statement in Matlab.



3.1. TECHNICAL PART

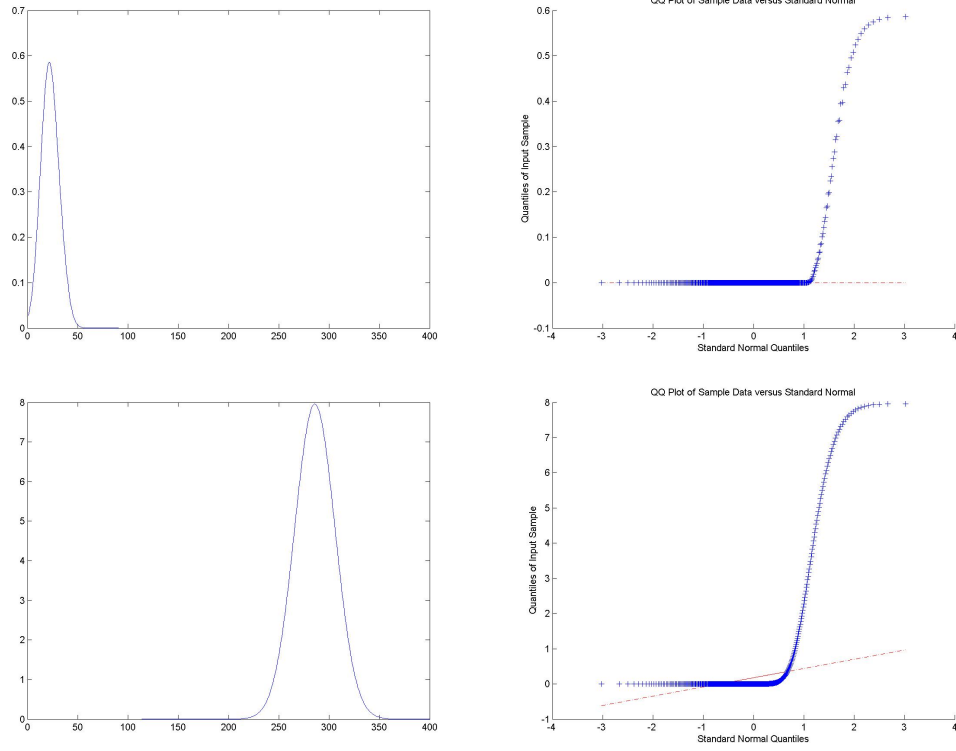


Figure 6: Distributions and QQ plots for each parameter in the case where we don't use parallel functions.

parameters	skewness	kurtosis
k	3.1765	11.7977
n	3.6937	15.8126
λ	2.1785	6.3758

Comparing the values of descriptive statistics from above of these three parameters we notice that with no parallel pooling, the skewness and kurtosis for k decrease, but for n and λ there is an increase in them. Moreover, for the latter parameter, the difference between the results from this case and the first case are very small.

3.1. TECHNICAL PART

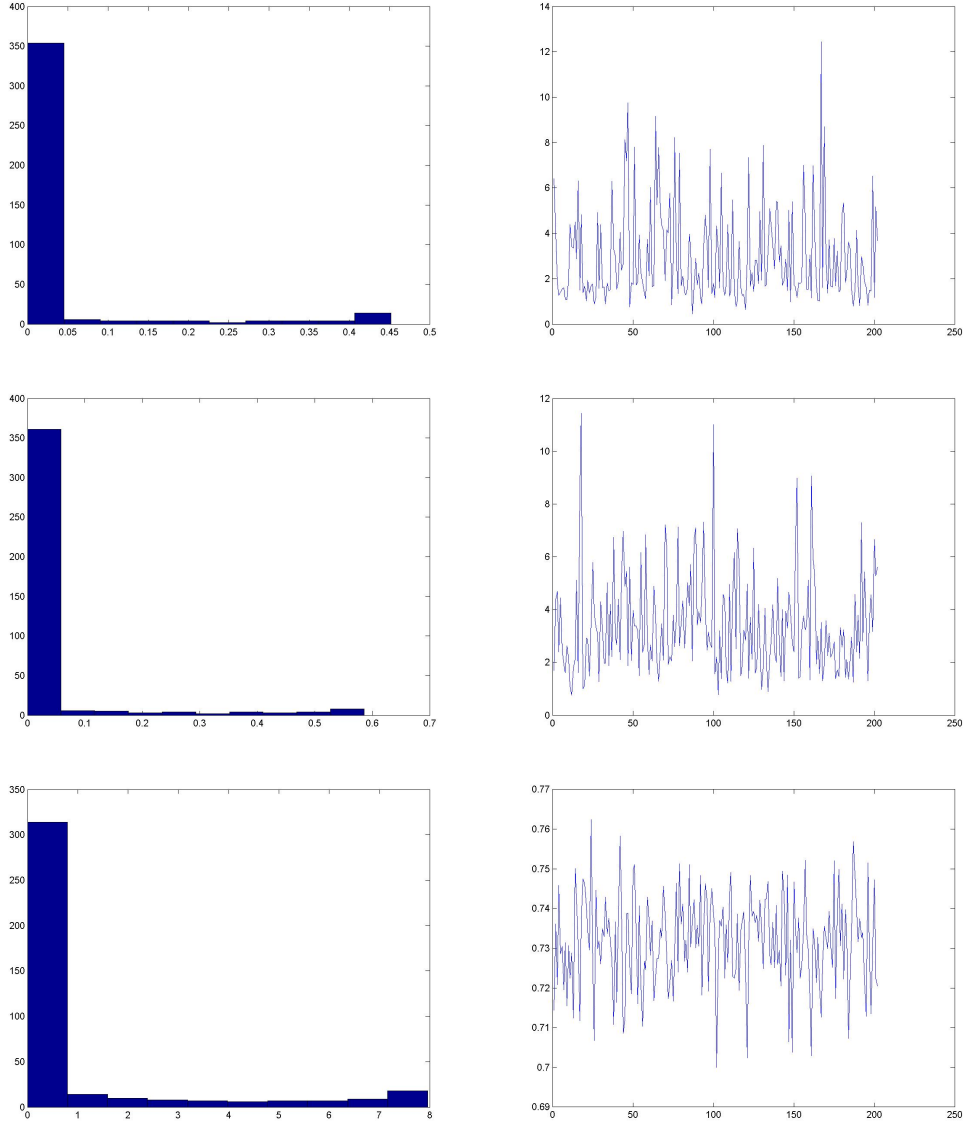


Figure 7: Histograms and progressive mean as in Figure 5, but for the case where parallel functions are not used.

For k , the jumps in the progressive mean are decreasing a bit, since the peaks are below 12 (except one), and in Figure 5, these values almost reaches 18. For n , we observe that jumps are higher than before. A small increase is noticed for the latter parameter.

3.1. TECHNICAL PART

As conclusion for this comparison we can say that the model provides similar results but the computing time differs. Therefore, when big datasets are used for analysis, parallel pooling is highly recommended because it will reduce the time drastically.

Case 3:

Changing the time window to 400 daily observation and considering 100 series give us some insights about how the model acts when we have only 4 estimates for the respective time window. Again in this case we use parallel pooling.

In this case we get 4 iterations:

Profile time 6123s
Iteration 1 CPU Time 1.404
Iteration 2 CPU Time 2.5428
Iteration 3 CPU Time 1.6068
Iteration 4 CPU Time 1.5912

Since we take into consideration more data, more estimates, I got the results in 1h 42 min, even if parallel pooling is used. This is due not only to the increase of daily observations number, but moreover to the decrease in the number of series. If the number of series decrease more, the computing time is even bigger because we have more estimates.

Next we will see how posterior distribution moves with iterations. From next picture I noticed that with more estimates the distribution will be different than before, we will see a big difference between this case and the case with only one estimate. Moreover in a later case, with even more estimates, not only the time is bigger, but also the distribution moves even more.

3.1. TECHNICAL PART

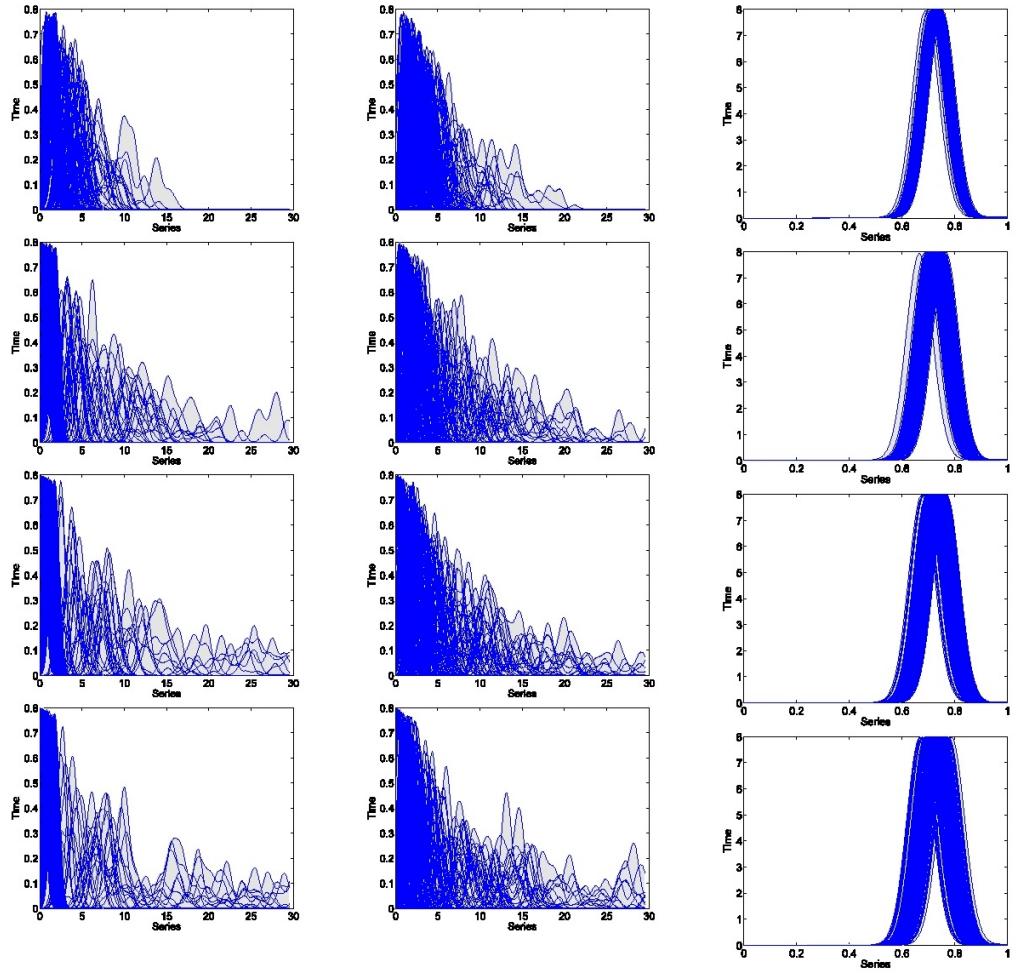
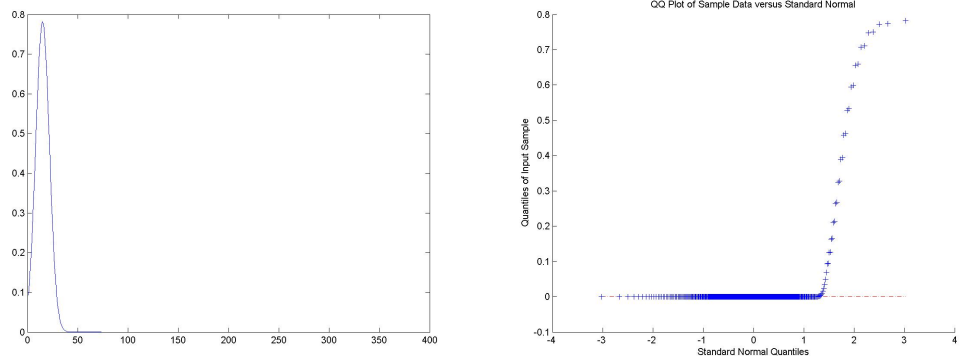


Figure 8: Block of posterior distributions of k, n, λ . Considering a bigger time window and 4 estimates we observe that posterior distribution diverges a bit, thus they seems to move more on the side where the tail is long.



3.1. TECHNICAL PART

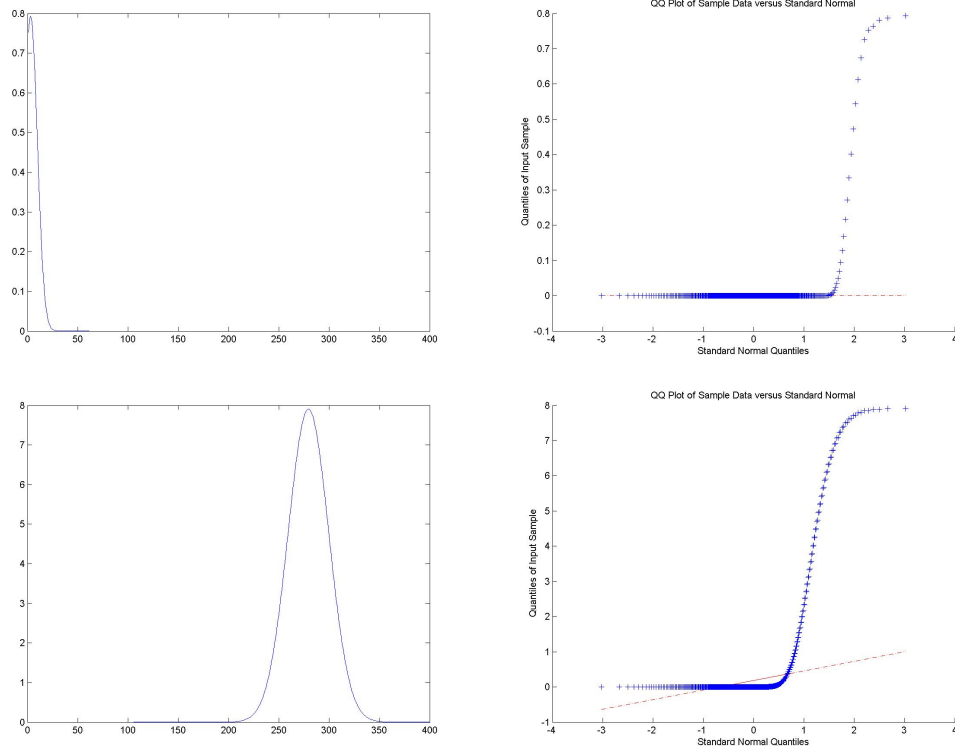


Figure 9: Distributions considering the first asset and QQ plots for each parameter. On the left of the figure are showed the distributions of each parameter considering only the first asset . The distributions of k and n seems to move to left but the distribution of λ is unchanged.

parameters	skewness	kurtosis
k	3.7564	16.1457
n	1.7079	4.3548
λ	2.1808	6.3867

Analysing these descriptive statistics, we observe that for n there is a huge decrease in them, in comparison with Case 1, Figure 4.

3.1. TECHNICAL PART

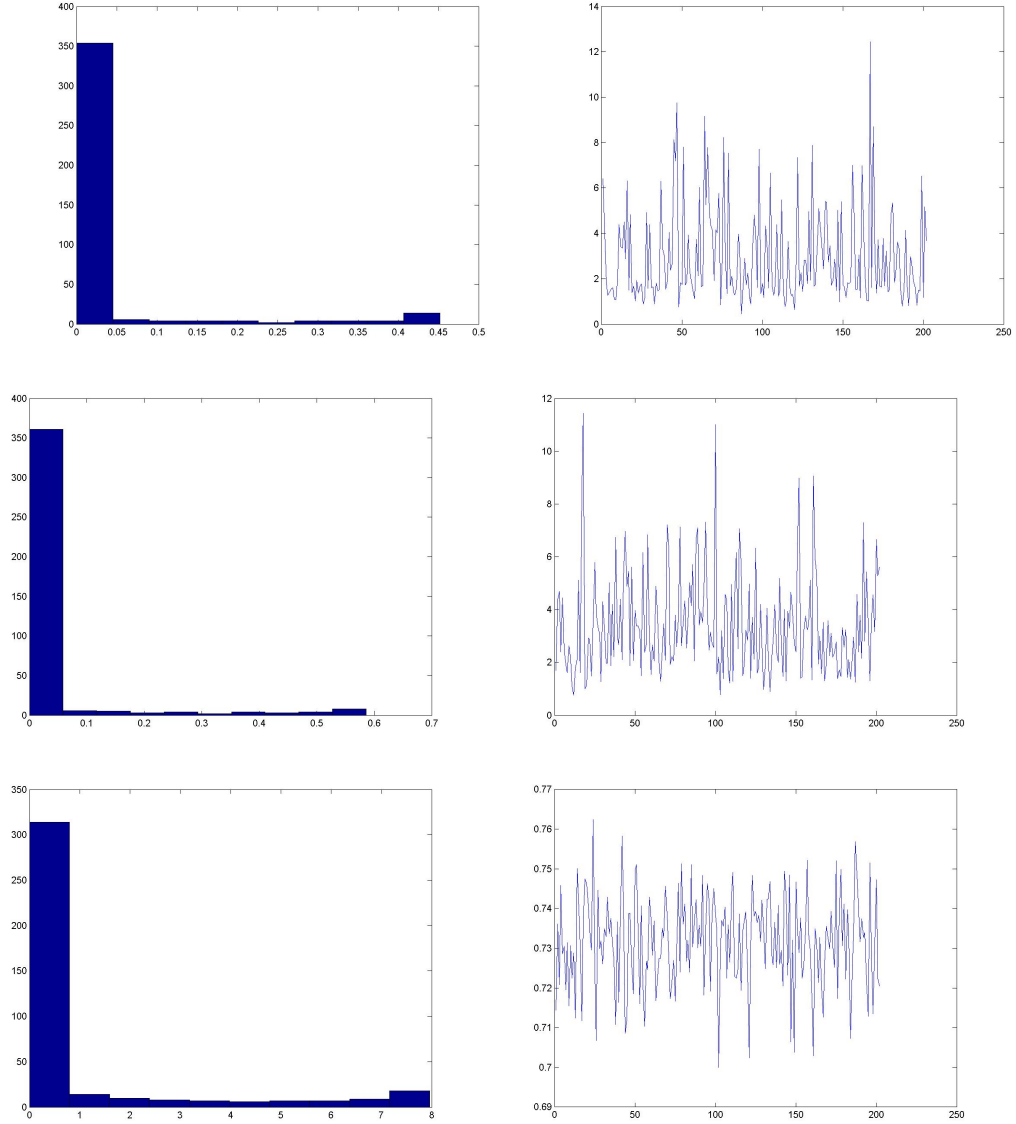


Figure 10: Histograms and progressive mean as in Figure 5.

Figure 9 and 10 show us that all distributions are right skewed (positive skewness). Thus there is more data in the right than expected in a normal distribution. Also we see that there are many jumps in the mean for each parameter, but this happens for all the considered cases. Also, all kurtosis are bigger than 3, and skewness bigger than 1.

3.1. TECHNICAL PART

Case 4:

If we turn back and consider 200 daily observations but only 10 series one can observe that the time is much bigger than in the other cases. A greater number of iterations is needed since with at this point we will have 40 estimates for the hole sample. Therefore I obtain the next figures after 12870 seconds (3 hours and 33 minutes) even if I used parpool function. The profile time from this case is 11 times bigger than in the first case (time window 200 daily observation and 400 series, 1 estimate). Furthermore, for those 40 estimates we get 20 iterations:

Iteration 1 CPU Time 0.7488
Iteration 2 CPU Time 0.5772
Iteration 3 CPU Time 0.3432
Iteration 4 CPU Time 0.312
Iteration 5 CPU Time 0.3432
Iteration 6 CPU Time 1.092
Iteration 7 CPU Time 0.5304
Iteration 8 CPU Time 0.7332
Iteration 9 CPU Time 0.1248
Iteration 10 CPU Time 0.4368

Iteration 11 CPU Time 0.2028
Iteration 12 CPU Time 0.2964
Iteration 13 CPU Time 0.2028
Iteration 14 CPU Time 0.1716
Iteration 15 CPU Time 0.2964
Iteration 16 CPU Time 1.4664
Iteration 17 CPU Time 16.8949
Iteration 18 CPU Time 0.546
Iteration 19 CPU Time 5.1168
Iteration 20 CPU Time 0.1716

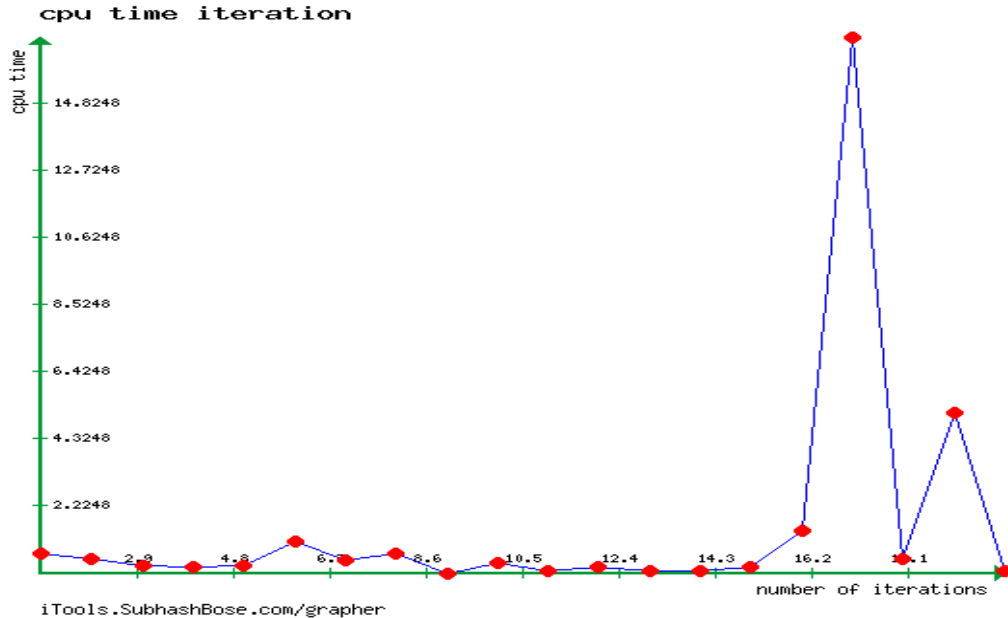
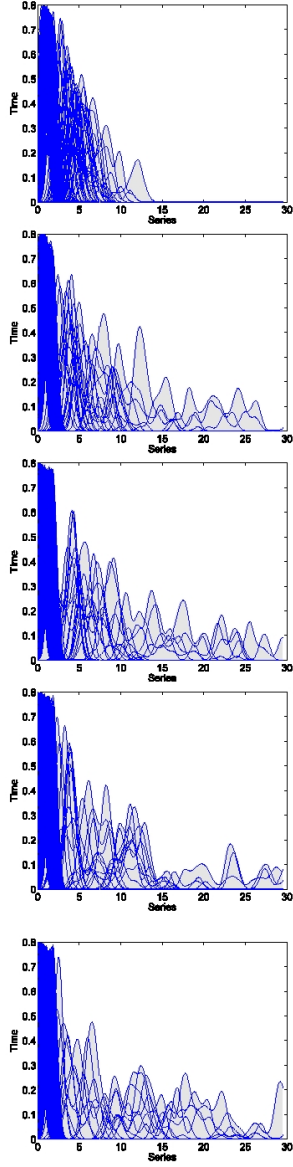


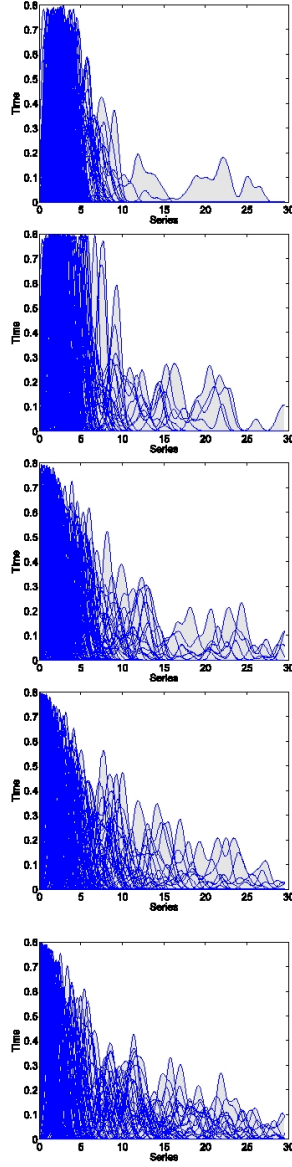
Figure 11: CPU time iteration when we have 40 estimates.

3.1. TECHNICAL PART

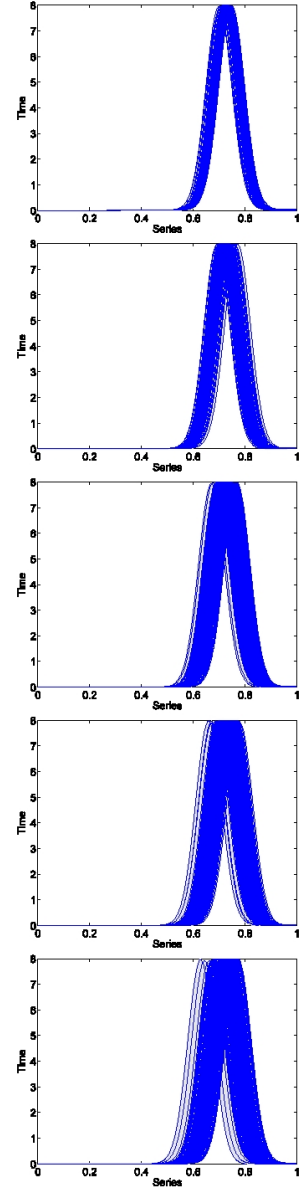
Blockparampost1 time1-5



Blockparampost2 time 1-5



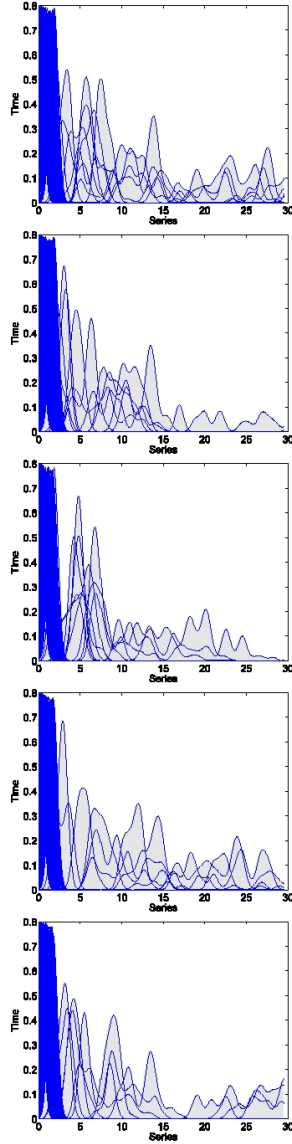
Blockparampost3 time1-5



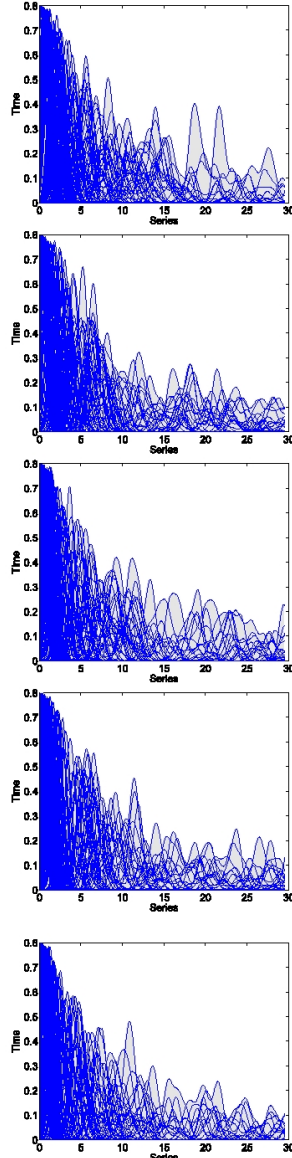
a) First five iterations.

3.1. TECHNICAL PART

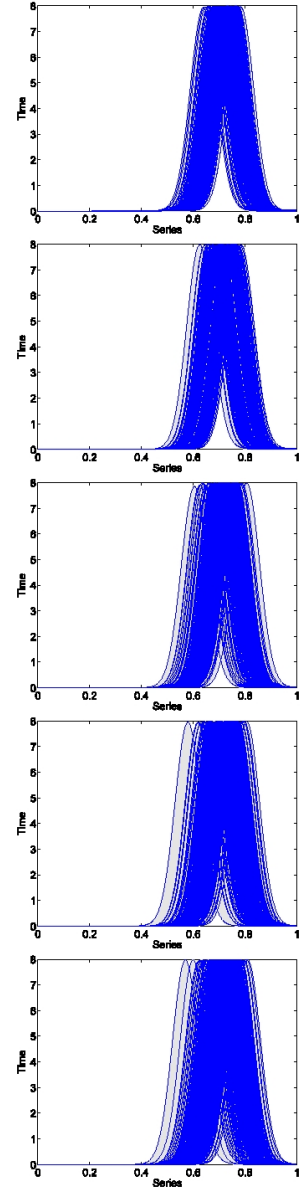
Blockparampost1 time 6-10



Blockparampost2 time 6-10



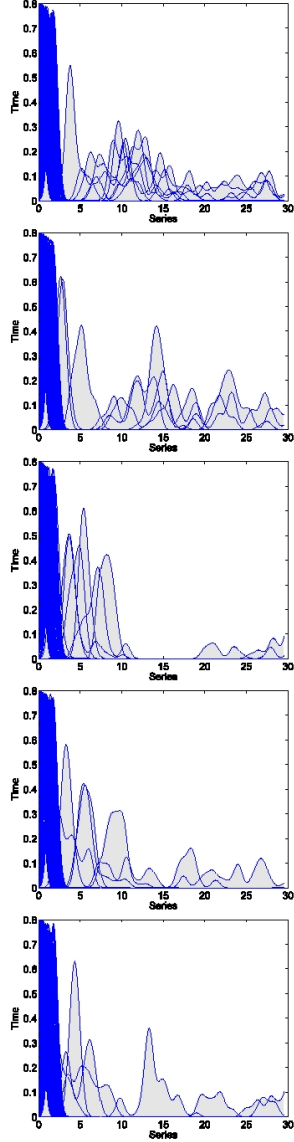
Blockparampost3 time 6-10



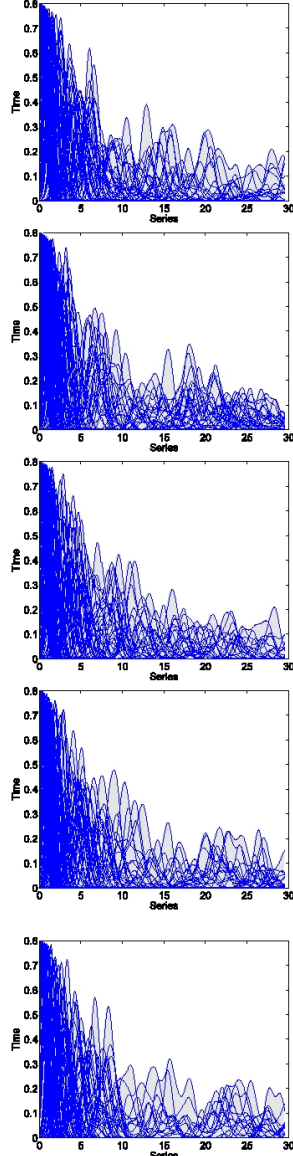
b) Iterations 6 to 10.

3.1. TECHNICAL PART

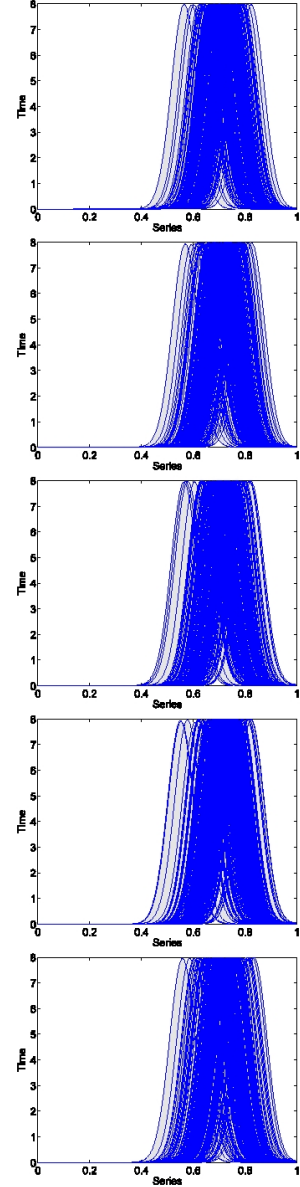
Blockparampost1 time 11-15



Blockparampost2 time 11-15

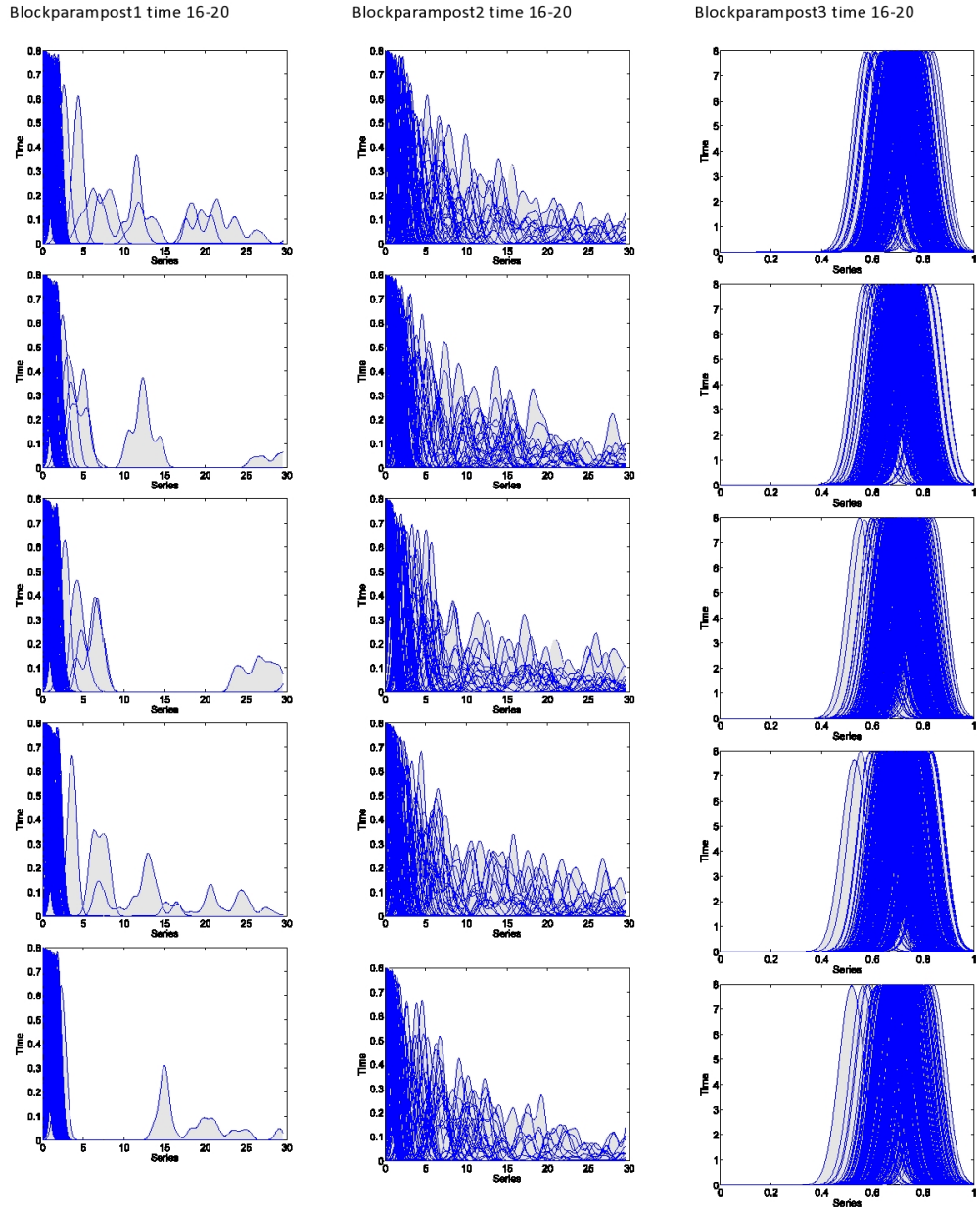


Blockparampost3 time 11-15



c) Iterations 11 to 15.

3.1. TECHNICAL PART



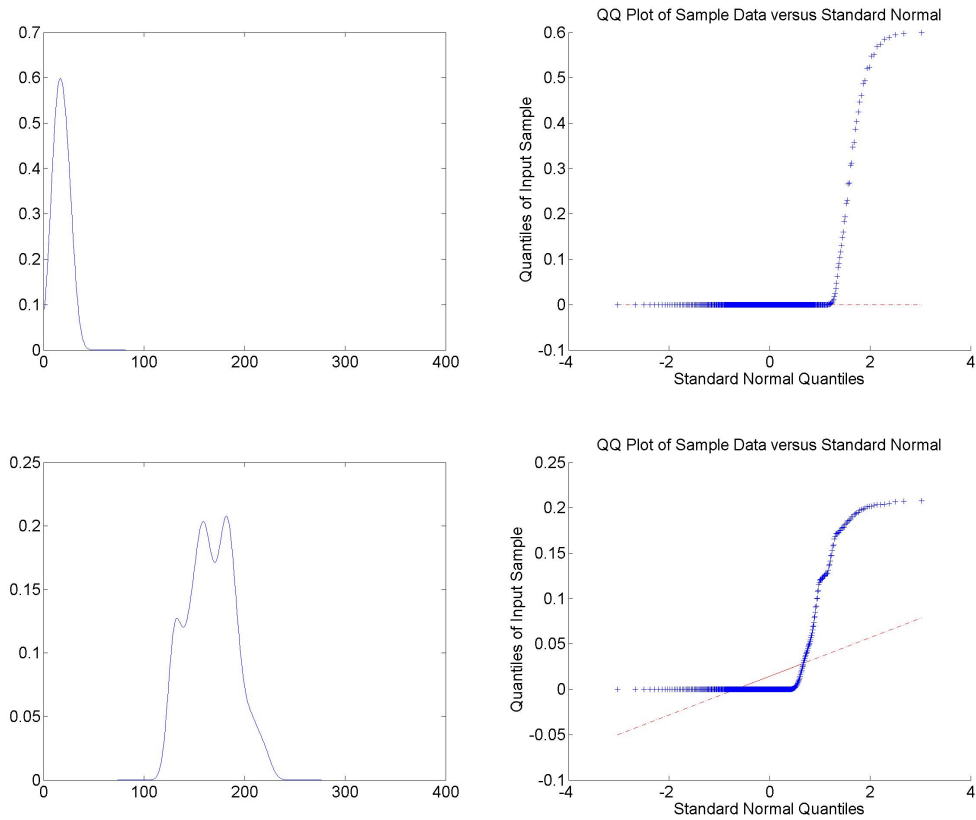
d) Last five iterations.

Figure 12: Block plots for the posterior of the parameters. Since we have 20

3.1. TECHNICAL PART

iterations and 40 estimates, as time increases we notice a change in distribution, thus an increase in skewness and kurtosis is observed. For k , the first parameter, the distribution becomes more concentrated on the left but we see also jumps in the tail. For the second parameter, n , the distribution seems to diverge more, also the angle of the right tail tends to increase with iterations.

What we can conclude from here is that using a smaller number of series the number of estimates increases proportional with the computing time (10 series \rightarrow 40 estimates \rightarrow 3 h 33 min; 400 series \rightarrow 1 estimate \rightarrow 21 min).



3.1. TECHNICAL PART

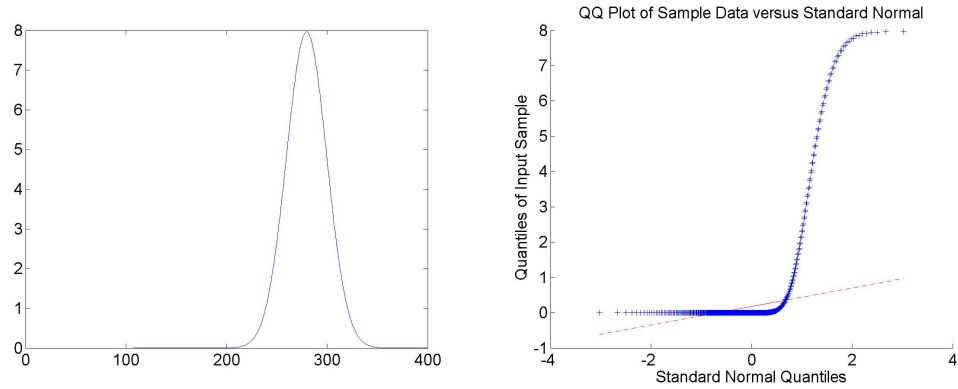
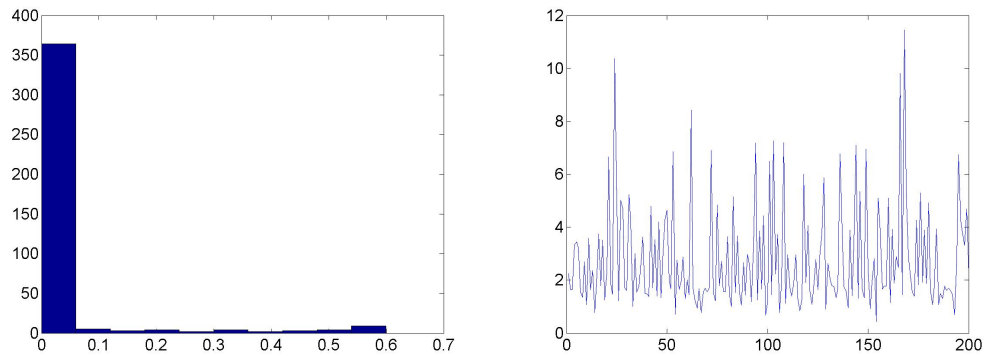


Figure 13: Parameter's distributions considering first asset only on the left and QQ plots on the right.

parameters	skewness	kurtosis
k	4.1711	19.7166
n	18.1996	345.9237
λ	2.1816	6.3902

Comparing this case with the first one (200 daily observations, 400 series, 1 estimate, 21 min), here, for k , a small decrease in skewness and kurtosis values is remarked. But the most significant change is for the second parameter: his skewness increase almost 5 times, and the kurtosis 21 times. That is way we get a distribution a bit change for n . Again a small increase in values for skewness and kurtosis for λ .



3.1. TECHNICAL PART

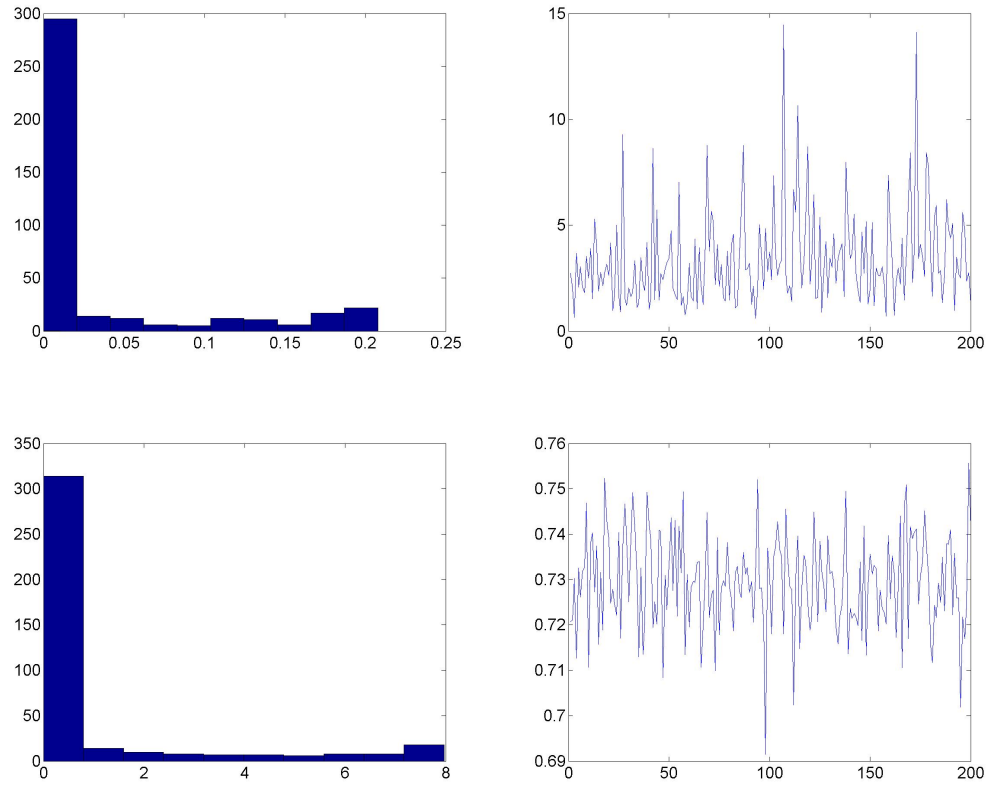


Figure 14: Histograms on the left column and progressive posterior mean over time of for each parameter. The jumps in means are higher than before.

As before, for the first parameter the maximums from posterior mean become smaller than in the first case. For n , we notice the opposite. As for λ , there is not a significant change.

Case 5:

Until now we saw how the model works when the dataset is liquid. But if we decrease the illiquidity rate condition then we will have more values around zero. For this purpose I consider the illiquidity rate bigger than 30%. With this case I want to show that it is not very appealing for economics analysis because here the dataset is not clean and with more insignificant values of daily observations we don't get accurate results.

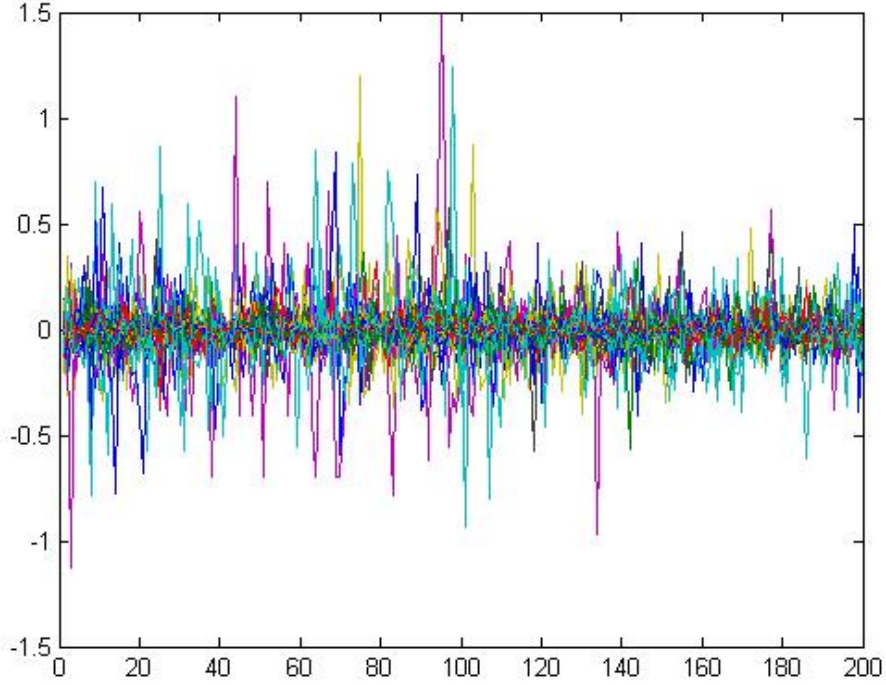
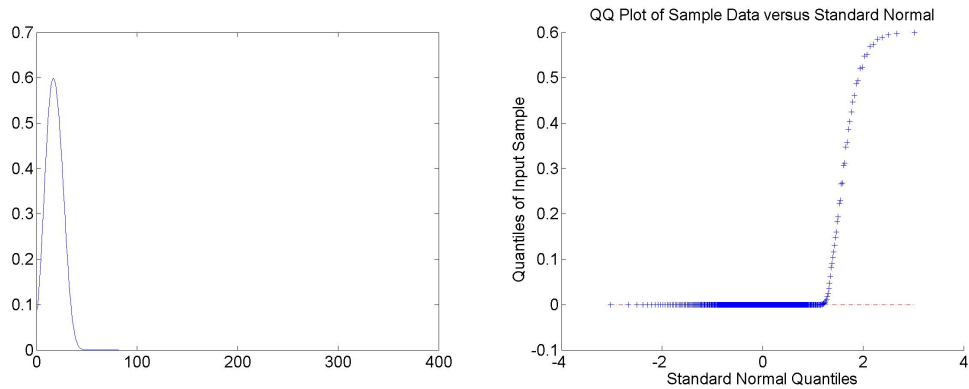


Figure 15: Daily returns when we consider more illiquid assets.

Comparing this figure with Figure 1, we see that most of the values of those returns lies between -0.5 and 0.5 thus very close to zero. In the first figure the most values are between -2 and 2. Moreover, there is a great difference between maximums and minimums in those two figure.

For this case I consider a time window of 200 daily observations and 200 series: Iteration 1 CPU Time 2.8548 and Profile time: 1287s.



3.1. TECHNICAL PART

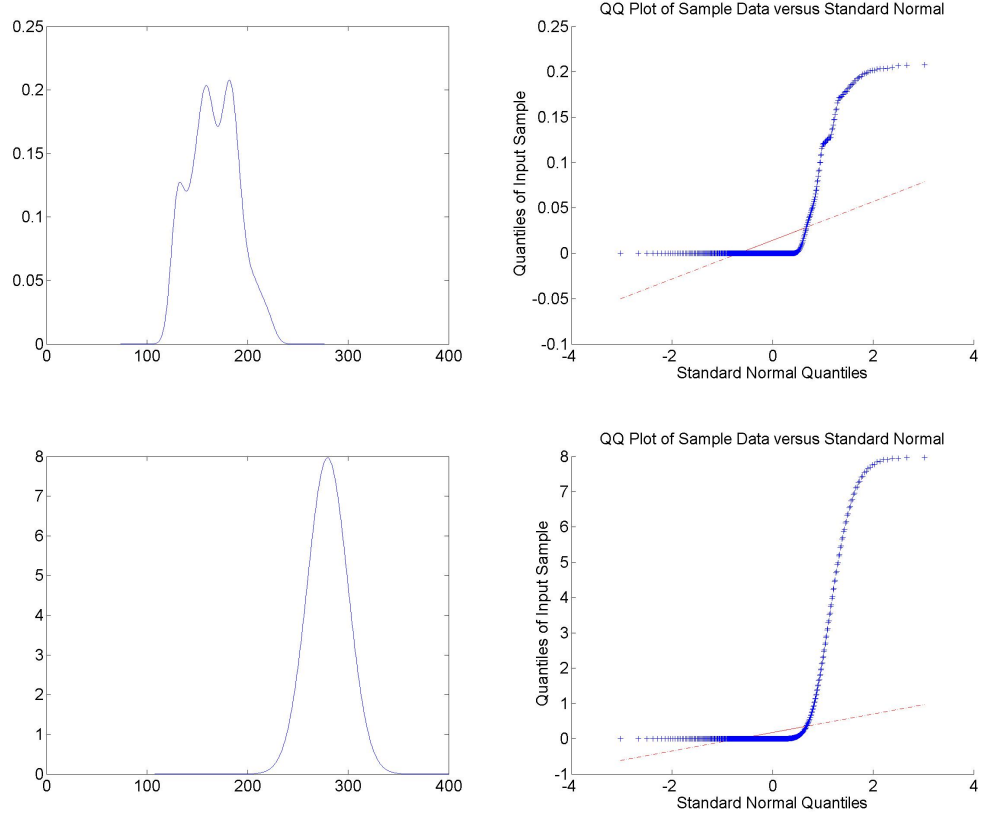


Figure 16: Parameter's distributions on the left and QQ plots on the right.

Again for λ the distribution doesn't change, but for the first two parameters we notice some changes.

parameters	skewness	kurtosis
k	4.3270	21.1840
n	4.2170	20.1883
λ	2.1805	6.3852

From these statistics, we see an increase for k and n , but a small decrease for the last parameter, if we compare them with the results from the first case.

3.1. TECHNICAL PART

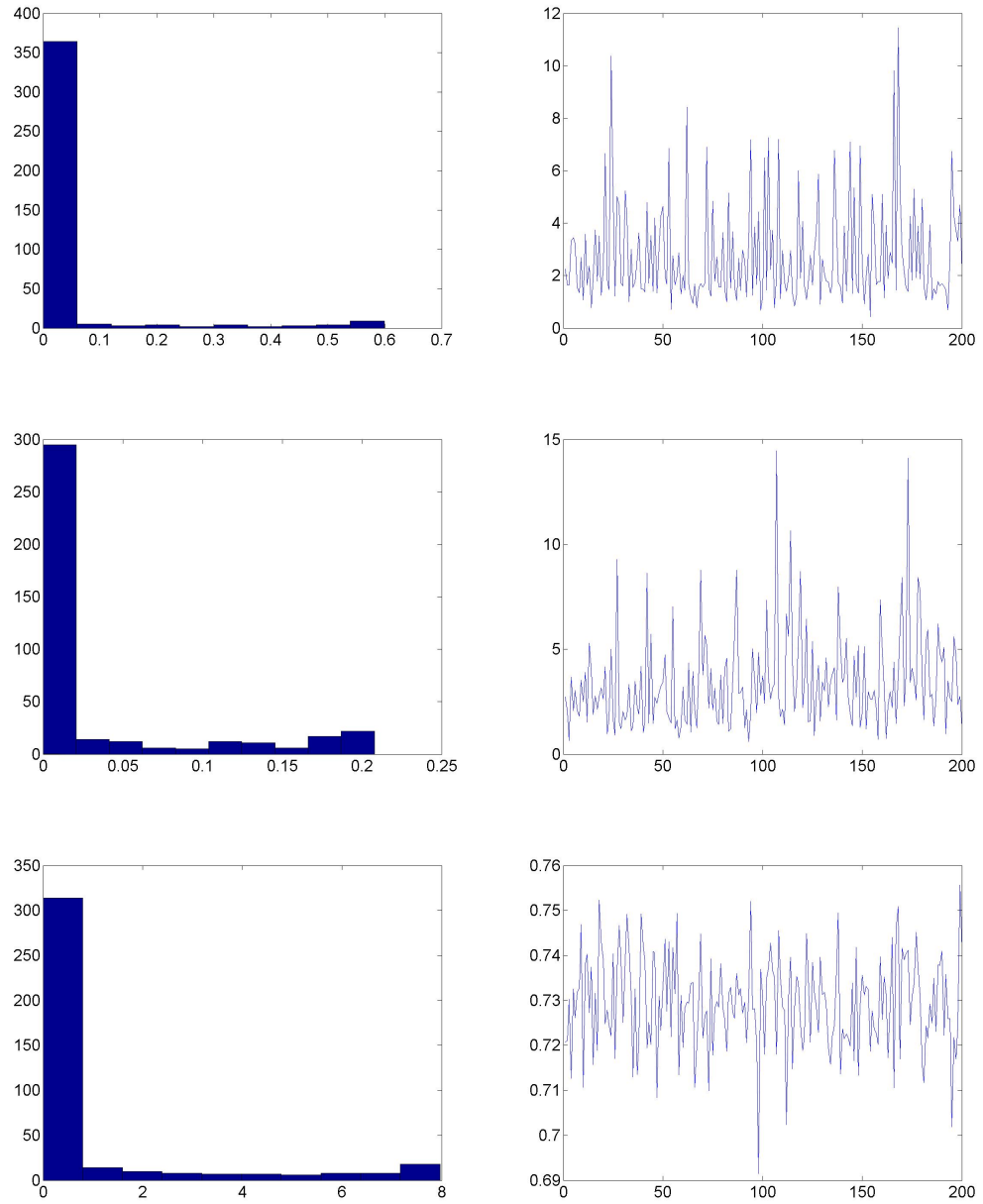


Figure 17: Histograms on the left column and progressive posterior mean over time of for each parameter. The jumps in means are higher than before.

I will put now all the means and variance of parameters for each case in order to see small difference between them. Analyzing them I see that the

3.1. TECHNICAL PART

difference is not significant, thus this model works well for all cases.

Case 1:

parameters	mean	variance
k	0.0331	0.0131
n	0.0332	0.0126
λ	1	4.6261

Case 2:

parameters	mean	variance
k	0.0331	0.0130
n	0.0332	0.0116
λ	1	4.6237

Case 3:

parameters	mean	variance
k	0.0297	0.0134
n	0.0298	0.0104
λ	1	4.6265

Case 4:

parameters	mean	variance
k	0.0228	0.0123
n	0.0218	0.0087
λ	1	4.6212

Case 5:

parameters	mean	variance
k	0.0331	0.0131
n	0.0332	0.0126
λ	1	4.6264

We notice that in second case, the third and the forth one there is a decrease in means and variances for the first two parameters, comparing with case 1. When the set is illiquid there is almost no difference.

3.2. CONCLUSIONS

In conclusion, for the above analysis, I employ a Metropolis within Gibbs sampler and draw all of the unknown quantities. To do this, I sample using the propositions from Chapter 2 in order to generate the draws. The main remarks are about the difference between parameters and time computing. When data sets are big it is required to use parallel pooling, obviously after a clean on the data is made.

Also, in all the previous case, I get positive skewness for all parameters, and that is why the distributions are right skewed (most values are concentrated on the left of the mean, with extreme values to the right). As for the indicator of flatterness or "peakedness" of a distribution, kurtosis is bigger than 3 in all the analyzed cases; thus we get leptokurtic distribution for all parameters. This means high probability for extreme values. Furthermore, the impact of the size of the sample plays a role in the change of these descriptive statistics.

3.2 Conclusions

I have an interest in big data problems and research that I think they are more important than anything, since we can understand many cases, for instance in finance, economics etc.. An important one is filtering out uncorrelated data because plays an important role in the quality of final analytics that are designed. Some of the main characteristics of the financial time series are time varying and clustering phenomena in volatility. Stochastic volatility models are widely used in finance in order to account for this issues.

I choose this model from Carvalho-Windle and Casarin paper because it provides a tractable model that can be used not only in finance but also in other fields. Moreover, I want to thank the authors for their contribution and for showing us how realized covariance matrices are used as latent states in a state-space model and how one can do a Bayesian analysis on it. Considering these papers as a model, it was easier to compute posterior distributions of the latent states and of the system's parameters since the observations and the latent states take values on the manifold of symmetric positive-definite matrices.

Applying this model, the univariate version of it, within the context of finance, it shows that covariance-valued state-space models can be considered as a good estimate of the conditional covariance matrix of a vector of

3.2. CONCLUSIONS

daily observations. Financial return distribution are leptokurtic and exhibits asymmetry. These features are showed in histograms and QQ-plots and represent a deviation from the normality assumption. The model's estimates and one-step ahead predictions are exponentially weighted averages which are known to give simple and robust estimates and forecasts in many cases.

Doing the Bayesian analysis in the second chapter, I got that the posterior samples are likely to display much more autocorrelation than if I were able to use known distributions. The Metropolis-Hasting step compounds this issue because as the dimension of the covariance matrices grows the acceptance rate of MH step will decrease. The current model approach, which is based on Uhlig's work also, can be forward and backward sampled. Therefore, it is much more effective.

In chapter one, I presented a few models related to stochastic volatility. What I can summarize from it is that it is better to use this model for short term predictions than to use factor stochastic volatility with daily returns only. The latter model uses opening and closing prices while UE model uses all the prices observed throughout the day.

In the cases from the previous section, I didn't count for uncertainty in the values of n , k and λ (smoothing parameter). I presented some cases to see how these parameters change over iterations and time window/daily observations. This algorithm can be used to achieve results in a better time, since I recommend using parallel pooling when we have to deal with big amount of data. Moreover, this model has few parameters which makes things easier to compute. Even if we cannot use usual methods in order to get distributions and predictions, I suggest that one can use Gibbs sample and MH algorithm. I run the model on a data set where I allowed more zero values to appear, thus to have a more illiquid set. With this comparison, I understood how the distributions change and how much time it is needed to obtain the result. From time point of view there is not a significant difference, only in distributions, they become more inaccurate. Also when we use more estimates for the whole sample the time increases proportionally with this number, even if parallel pooling is used. When we delete it the time increases drastically.

Also, this model outperforms better if we consider one-step ahead predictions. Although this doesn't give us a good view about forecasts. More informations about multiple step ahead prediction can be seen in Casarin's 2015 paper. More comments about the original paper, Windle and Carvalho 2015, can be found in Enrique ter Horst and German Molina and Catherine

3.2. *CONCLUSIONS*

Scipione Forbes papers. The first one focus more on frequency of data, the last one is about the model's assumptions.

Bibliografy

- [1] Aguilar, O. and West, M. (2000). Bayesian Dynamic Factor Models and Portfolio Allocation. *Journal of Business and Economic Statistics*, 18(3): 338-357.
- [2] Asai, M. and McAleer, M. (2009). The Structure of Dynamic Correlations in Multivariate Stochastic Volatility Models. *Journal of Econometrics*, 150: 182-192.
- [3] Bayesian analysis (2014), Volume 9, Number 4, pg 759-792
- [4] Carter, C. K. and Kohn, R. (1994). On Gibbs Sampling for State Space Models. *Biometrika*, 81: 541-533.
- [5] Casarin, R. (2015). Comment on a tractable state-space model for symmetric positive matrices, *Bayesian Analysis*, 9(4), 793-804.
- [6] Cochrane, J. H. (2005). *Asset Pricing*. Princeton University Press.
- [7] Fama, E. F. and French, K. R. (1993). Common Risk Factors in the Returns on Stocks and Bonds. *Journal of Financial Economics*, 33: 3-56.
- [8] Fame, (2012) Fame. Bureau van Dijk. [Online]. Available at: <http://fame2.bvdep.com/>
- [9] Fox, E. B. and West, M. (2011). Autoregressive Models for Variance Matrices: Stationary Inverse Wishart Processes. Technical report, Duke University.
- [10] Fruwirth-Schnatter, S. (1994). Data Augmentation and Dynamic Linear Models. *Journal of Time Series Analysis*, 15: 183-202.
- [11] Gouriéroux, C., Jasiak, J., and Sufana, R. (2009). The Wishart Autoregressive Process of Multivariate Stochastic Volatility.” *Journal of Econometrics*, 150: 167-181.

- [12] Harrison, J. and West, M. 1997. Bayesian Forecasting and Dynamic Models, Second Edition, Springer.
- [13] Hauberg, S., Lauze, F., and Pedersen, K. S. (2013). Unscented Kalman Filtering on Riemannian Manifolds. *Journal of Mathematical Imaging and Vision*, 46: 103-120.
- [14] Harvey, A., Ruiz, E., and Shephard, N. (1994). Multivariate Stochastic Volatility Models. *The Review of Economic Studies*, 61: 247-264.
- [15] Jacquier, E., Polson, N. G., and Rossi, P. E. (2004). Bayesian Analysis of Stochastic Volatility Models with Fat-Tails and Correlated Errors. *Journal of Econometrics*, 122: 185-212.
- [16] Julier, S. J. and Uhlmann, J. K. (1997). New Extensions of the Kalman Filter to Nonlinear Systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*, volume 3068.
- [17] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82 (Series D): 35-45.
- [18] Muirhead, R. J. (1982). *Aspects of Multivariate Statistical Theory*. Wiley.
- [19] Prado, R. and West, M. (2010). Time Series: Modeling, Computation, and Inference, chapter Multivariate DLMS and Covariance Models, 263-319. Chapman and Hall/CRC.
- [20] Ruppert David. (2011). *Statistics and data analysis for financial engineering*. Springer-Verlag.
- [21] Quintana, J. M. and West, M. (1987). An Analysis of International Exchange Rates Using Multivariate DLMS. *The Statistician*, 36: 275-281.
- [22] Shephard, N. (1994). Local Scale Models: State Space Alternative to Integrated GARCH Processes. *Journal of Econometrics*, 60: 181-202.
- [23] Shepard (2005). *Stochastic Volatility: Selected Readings*. Oxford University Press.
- [24] Sylvia Frhwirth-Schnatter. 2006. *Finite Mixture and Markov Switching Models*. Springer-Verlag.
- [25] Triantafyllopoulos, K. (2008). Multivariate stochastic volatility with Bayesian dynamic linear models. *Journal of Statistical Planning and Inference*, 138: 1021-1037.

BIBLIOGRAPHY

- [26] Tyagi, A. and Davis, J. W. (2008). A Recursive Filter For Linear Systems on Riemannian Manifolds. In IEEE Conference on Computer Vision and Pattern Recognition.
- [27] Uhlig, H. (1994). On Singular Wishart and Singular Multivariate Beta Distributions. *The Annals of Statistics*, 22(1): 395-495.
- [28] Uhlig (1997). Bayesian Vector Autoregressions with Stochastic Volatility. *Econometrica*, 65(1): 59-73.
- [29] Wang, H. and West, M. (2009). "Bayesian analysis of matrix normal graphical models." *Biometrika*, 96(4): 821-834.
- [30] West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. Springer Verlag.
- [31] Windle, J. and Carvalho, C. (2015). A tractable state-space model for symmetric positive-definite matrices. *Bayesian Analysis*.
- [32] <http://www.mathworks.com/help/matlab/ref/parfor.html>
- [33] <http://www.mathworks.com/help/distcomp/parpool.html>
- [34] https://en.wikipedia.org/wiki/Kalman_filter