Non-Gaussian Quasi Maximum Likelihood Estimation (NGQMLE) for GARCH Model with Heavy-tailed Innovation

A PROJECT
SUBMITTED TO THE FACULTY OF
UNIVERSITY OF MINNESOTA
BY

Qiaozhi Li

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Advisors: Kang James, Barry James

May 2016

Acknowledgements

I would like to thank my advisors, Professor Kang James and Professor Barry James, for their excellent support in my study for the master's degree. And I would also like to thank Professor Douglas Dunham from the Computer Science Department for being the committee member.

Abstract

This project aims at discussing a new estimation method of Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model whose error term has a heavy-tailed distribution. The innovative three-step estimation procedure is studied, and we proposed a modified version to further improve the estimation accuracy under heavy-tailed distribution. Simulation study is provided, and the three-step Non-Gaussian Quasi Maximum Likelihood Estimation (NGQMLE) performs better than traditional GQMLE under several chosen heavy-tailed distributions. In the end, we apply this method to the datasets from Shanghai Stock Exchange Market for the empirical study.

Table of Contents

CHAPTER 1

Introduction

Volatility has been playing an essential role in financial risk management, and its clustering phenomenon has shown challenge to financial researchers. Engle(1982) designed the Autoregressive Conditional Heteroscedasticity(ARCH) model to capture the heteroscedasticity characteristic, which enabled the conditional volatility to depend on the past squared returns. Bollerslev(1986) generalized this idea and came up with the Generalized Autoregressive Conditional Heteroscedasticity(GARCH) model, so that the volatility not only depends on the past squared returns, it also depends on the past volatility. One advantage of the GARCH model is that: the order of the model can be much smaller than that of the ARCH model. The GARCH Model usually takes the following form:

$$x_t = \sigma_t \varepsilon_t$$
$$\sigma_t^{\ 2} = c + \sum_{i=1}^{p} \alpha_i x_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 .$$

For convenience of analysis, I will be adopting its equivalent form:

$$x_t = \sigma v_t \varepsilon_t$$
$$v_t^{\ 2} = 1 + \sum_{i=1}^{p} a_i x_{t-i}^2 + \sum_{j=1}^{q} b_j v_{t-j}^2 .$$

When it comes to the distribution of the error term of GARCH, the normal assumption has been taking an essential role. For example, Duan(1995) used a GARCH Model with Gaussian distribution innovation in the option pricing model. However, as shown by Mikosch and Starica(2000), the tails of the fitted GARCH models sometimes seem to be much thinner than the tails apparent in the data. Instead of the normality assumption,

heavier tailed distribution assumptions have been widely used these days on the innovation of GARCH.

Due to the fact that the true distribution of the data is not known, the Quasi Maximum Likelihood Estimation(QMLE) method has taken an essential role in the estimation of GARCH. A finite fourth moment is an important assumption for Gaussian quasi maximum likelihood estimation(GQMLE), as illustrated by Berkes, Horváth (2004). The GQMLE will be consistent and asymptotically normal if this moment assumption, along with other regular assumptions, is satisfied, even if the true innovation has a heavy-tailed distribution. However, the performances of GQMLE can still be unsatisfying in the heavy-tailed case. Intuitively, a non-Gaussian Quasi Maximum Likelihood Estimator (NGQMLE) could be considered. Nonetheless, as pointed out by Fan et al. (2014), the consistency of the NGQMLE cannot be guaranteed unless different model assumptions are placed, depending on the likelihood function used. As mentioned by Hill (2015), these alternative assumptions are not so pleasant when it comes to practical research in finance or economics.

A three-step NGQMLE was proposed by Fan et al. (2014) to fix this problem. The key idea of the three-step NGQMLE is to introduce an adjustment parameter, so that by the modification from this parameter, the identification condition for consistency can be satisfied. The steps proposed by Fan et al. (2014) are the following: in the preliminary estimation step, apply GQMLE to estimate the GARCH model and extract the residuals from the model; in the second step, estimate the adjustment parameter based on the residuals and the chosen likelihood function; in the last step, perform NGQMLE along with the adjustment parameter to finish the estimation of the parameters. In practice, once we retrieve the residuals from the first step with GQMLE, we can choose an appropriate

likelihood based on the retrieved residuals and further improve the efficiency as well as the performance of the estimator. In addition, the adjustment parameter is informative: as mentioned by Fan et al.(2014) in their rejoinder, it measures the difference between the likelihood we choose and the true distribution. However, the convenience of using GQMLE in the first step cannot totally remedy the mediocre performance under the situation when the true distribution is heavy-tailed, which is quite common in financial data. If the estimation in the first step is not satisfying, the retrieved estimated residuals can suffer from poor accuracy, which will influence our choice for the likelihood function as well as the accuracy of the estimated adjustment parameter in the second step. Here, I modify the first step to use a Rank estimator proposed by Andrews (2012). As shown by Andrews in her paper, the asymptotic relative efficiency for the Rank estimation with respect to GQMLE is always larger than one under several typical heavy-tailed distributions. This inspires me to use this robust and efficient nonparametric method in the first step of NGQMLE instead of the GQMLE.

In addition, Fan et al. (2014) stated three likelihood functions that satisfy the conditions that are needed to ensure the consistency property of the estimator. They are: Gaussian density functions, standard t density functions with degrees of freedom larger than 2 and generalized Gaussian density functions. And the likelihood function chosen for the simulation study in their research is a standard t density function with degrees of freedom 7. Besides the t distribution and Generalized Gaussian distribution, the hyperbolic distribution introduced by Barndorf (1977) also has a heavier tail, compared to the tail of the normal distribution. And using the hyperbolic distribution to fit financial data has already gained success in some research. This has been shown by E. Eberlein and U.

Keller(1995). In this paper, I verify that the standard symmetric hyperbolic density function satisfies the requirement placed on the likelihood functions to ensure the consistency of the estimator, which provides me one more choice for likelihood function to be used in the estimation.

The project is organized as following: Chapter 2 will discuss the basics of GARCH models, including the motivation for developing the GARCH model and the form of the model. In Chapter 3, estimation methods for GARCH models are introduced. They are GQMLE, rank estimation, three-step NGQMLE and the modified version of three-step NGQMLE. Chapter 4 will introduce the simulation results, where we draw RMSE (root mean square error) comparisons for GARCH coefficients estimated from GQMLE, three-step NGQMLE and our modified version of three-step NGQMLE. Also, we provide a table of RMSE of the estimated scale adjustment parameters obtained from GQMLE and rank estimation. Lastly, in Chapter 5, we estimated the scale adjustment parameters for 528 different stocks from the Shanghai Stock Exchange in China, separately with GQMLE and Rank estimation, and study their patterns using histograms. Moreover, a comparison of GARCH parameters from both GQMLE and three-step NGQMLE will be provided. And lastly, we will compare the difference between the estimated scale parameters given by two versions of NGQMLE.

CHAPTER 2

GARCH Model

Autoregressive Moving Average (ARMA) models have been used for the study of financial time series analysis. However, as pointed out by Cont (2001), there are some statistical facts that makes this classic model unsatisfying in financial research. Three of them hugely contribute to the motivation of this project:

1. Absence of significant autocorrelation for the return data.
2. Volatility clustering: the squared return data shows significant positive autocorrelation. There are periods when high volatility lasts and periods when low volatility lasts.
3. Heavy-tailed distribution: normality assumption has been used widely, but the practical data sets tend to have heavier tails.

The first two facts point out the limitation of ARMA models and the importance of introducing the autoregressive conditional heteroscedasticity (ARCH) model. Furthermore, evidence of these two facts can be found via figure 2.1 below. The Autocorrelation Function(ACF) and Partial Autocorrelation Function(PACF) plots do not show a very significant autocorrelation pattern of the log return of S&P 500. However, the squared data can show an obvious autocorrelation pattern(Figure 2.2).

The classic ARMA models can be considered as a combination of both the AR model and the MA model. If we consider the mean of current return as a linear function of the previous returns, then we can write out the form of AR (p) model:

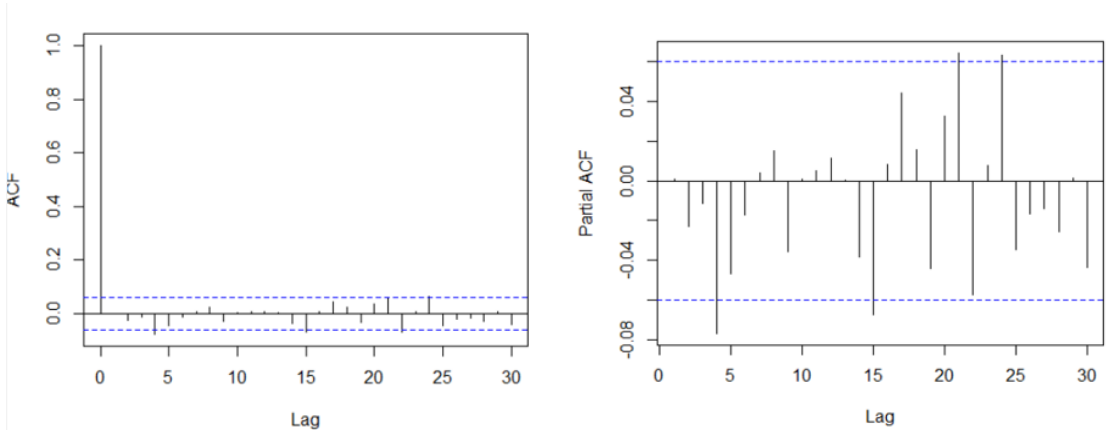


Figure2.1 ACF and PACF of the S&P 500 log return from 2012.1.1 to 2016.4.10

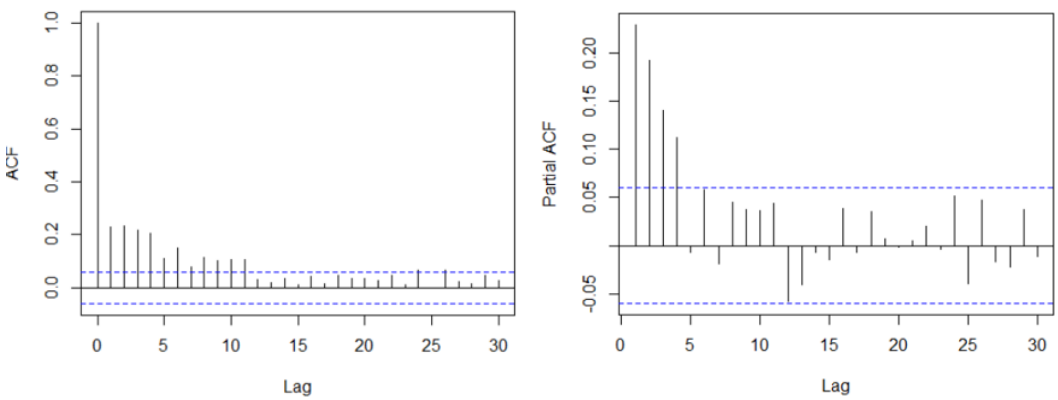


Figure2.2 ACF and PACF of the S&P 500 squared log return from 2012.1.1 to 2016.4.10

$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t$, which also indicates that $E[y_t \mid I_{t-1}] = \sum_{i=1}^{p} \phi_i y_{t-i}$.

Here $\varepsilon_t \sim N(0, \sigma^2)$, and I_t is the information up to time t; for example:

$E[y_t \mid I_{t-1}] = E[y_t \mid y_s, s \le t-1]$.

If we model the conditional mean of the current data as a linear combination of past errors, then we will have the MA(q) model:

$y_t = \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t$, which means that $E[y_t \mid I_{t-1}] = \sum_{i=1}^{q} \theta_i y_{t-i}$.

Both AR models and MA models can be considered as specific examples of ARMA models. ARMA(p,q) can be constructed by combing the ideas of AR(p) and MA(q):

$$y_t = \sum_{j=1}^{p} \phi_j y_{t-j} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t .$$

When the ϕ_j are all 0, it becomes a MA(q) model. Similarly, if θ_i are 0 for all i , then it is a AR(p) model.

The financial return data frequently shows a volatility clustering phenomenon, as illustrated by figure 2.3. Therefore, it is more reasonable to assume that the volatility is changing over time.

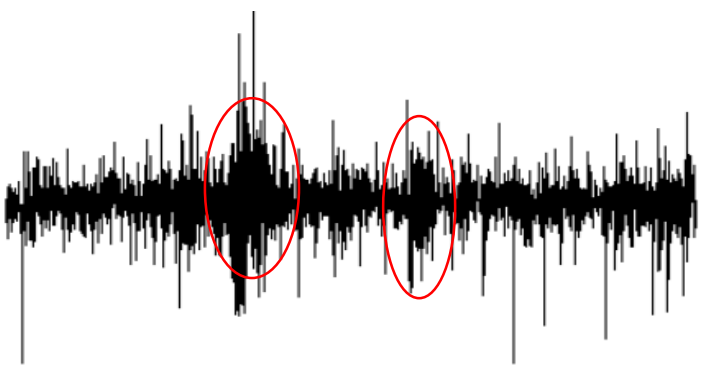


Figure 2.3 log return plot of IBM stock from 2012.1.1 to 2016.4.10

Borrowing the idea from the ARMA model, we express the conditional variance as a linear function of some past values. If we have centered the original data; let $x_t = y_t - E(y_t \mid I_{t-1})$. Then x_t will have conditional mean 0, which indicates that the conditional expectation of x_t^2 is actually the conditional variance of x_t. Therefore, x_t^2 will be a reasonable variable to be included in prediction of σ_t^2. If we express σ_t^2 as a linear function of the past values of x_t^2, then we will have an ARCH model:

$$\sigma_t^{\ 2} = c + \sum_{i=1}^{p} \alpha_i x_{t-i}^2 .$$

Borrowing the idea from the ARMA model, we can include the past values of $\sigma_t^{\ 2}$ itself on the right hand side as well, which brings us to the GARCH model:

$\sigma_t^{\ 2} = c + \sum_{i=1}^{p} \alpha_i x_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2$, where $c > 0, \alpha_i \geq 0, \beta_j \geq 0$ for all i and j.

Assuming that $x_t = y_t - E(y_t \mid I_{t-1})$, then we can write $x_t = \sigma_t \varepsilon_t$, where ε_t is the error term with mean 0 and variance 1. Therefore, we can see that

$$Var(x_t \mid I_{t-1}) = c + \sum_{i=1}^{p} \alpha_i x_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 .$$

Therefore, the conditional variance of x_t will be represented as a linear function of the history of $x_t^{\ 2}$ and σ_t^2.

The GARCH model can be written in an equivalent form:

$$x_t = \sigma v_t \varepsilon_t$$
$$v_t^{\ 2} = 1 + \sum_{i=1}^{p} a_i x_{t-i}^2 + \sum_{j=1}^{q} b_j v_{t-j}^2 .$$

In this form, the intercept term becomes 1, and there is a scale parameter σ in the expression of x_t. In this case, it can be shown that:

$$Var(x_t \mid I_{t-1}) = \sigma^2 v_t^{\ 2} = \sigma^2 (1 + \sum_{i=1}^{p} a_i x_{t-i}^2 + \sum_{j=1}^{q} b_j v_{t-j}^2) .$$

We will be using this form of GARCH model for analysis in the subsequent sections. The equivalence can be shown by a simple verification:

$$\sigma^2 v_t^{\ 2} = \sigma^2 (1 + \sum_{i=1}^{p} a_i x_{t-i}^2 + \sum_{j=1}^{q} b_j v_{t-j}^2)$$

$$= \sigma^2 + \sum_{i=1}^{p} \sigma^2 a_i x_{t-i}^2 + \sum_{j=1}^{q} b_j \sigma^2 v_{t-j}^2$$

$$= \sigma^2 + \sum_{i=1}^{p} \sigma^2 a_i x_{t-i}^2 + \sum_{j=1}^{q} b_j \sigma_{t-j}^2 .$$

Therefore, $c = \sigma^2, \alpha_i = \sigma^2 a_i$ and $\beta_j = b_j$.

CHAPTER 3
Estimation of GARCH Model

3.1 Quasi Maximum Likelihood Estimation(QMLE)

Let $f(x)$ be the likelihood function we choose and $g(x)$ be the density function of the random variable X. In Maximum Likelihood Estimation, we assume that $g(x) = f(x)$. However, the density function $f(x)$ behind the random variable is never known to us. Quasi Maximum Likelihood Estimation is built on this idea, which allows us to use a likelihood function that is different from the true density function.

The derivation of QMLE requires us to know the Kullback-Leibler Information Distance (KLID), which is defined as follows:

Definition (Kullback-Leibler Information Distance)

Let $g_t(x|\mathbf{x_{t-1}})$ be the true conditional density function of the random variable X_t. And $f_t(x|\mathbf{x_{t-1}};\boldsymbol{\theta})$ be the conditional quasi-likelihood function. Then the KLID of $g_t(x|\mathbf{x_{t-1}})$ relative to $f_t(x|\mathbf{x_{t-1}};\boldsymbol{\theta})$ is:

$$I(f_t;g_t) = \int \log\left(\frac{g_t(x|\mathbf{x_{t-1}})}{f_t(x|\mathbf{x_{t-1}};\boldsymbol{\theta})}\right) g_t(x|\mathbf{x_{t-1}})dx,$$

where $\mathbf{x_t} = (x_t, x_{t-1}, ..., x_1)$. And in our case, $\boldsymbol{\theta} = (\sigma, a_1...a_p, b_1...b_q)$.

Assuming that our sample size is T, then the average KLID for our sample is:

$$\frac{1}{T}\sum_{t=1}^{T} I(f_t;g_t) = \frac{1}{T}\sum_{t=1}^{T}\int \log\left(\frac{g_t(x|\mathbf{x_{t-1}})}{f_t(x|\mathbf{x_{t-1}};\boldsymbol{\theta})}\right) g_t(x|\mathbf{x_{t-1}})dx.$$

We obtain the quasi maximum likelihood estimation by minimizing the average KLID:

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T} I(f_t; g_t) = \arg\min_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T} \int \log(\frac{g_t(x \mid \mathbf{x_{t-1}})}{f_t(x \mid \mathbf{x_{t-1}}; \boldsymbol{\theta})}) g_t(x \mid \mathbf{x_{t-1}}) dx$$

$$= \arg\min_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T} (E_g[\log(g_t(x \mid \mathbf{x_{t-1}})) \mid \mathbf{x_{t-1}}] + E_g[\log(\frac{1}{f_t(x \mid \mathbf{x_{t-1}}; \boldsymbol{\theta})}) \mid \mathbf{x_{t-1}}])$$

$$= \arg\max_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T} E_g[\log(f_t(x \mid \mathbf{x_{t-1}}; \boldsymbol{\theta})) \mid \mathbf{x_{t-1}}].$$

The expectation operation above indicates the difference between our QMLE and the traditional MLE. However, in practice, we cannot do the expectation since the true density function is never known. Therefore, we use an approximation instead:

$$\arg\max_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T} \log(f_t(x_t \mid \mathbf{x_{t-1}}; \boldsymbol{\theta})).$$

The result is called the quasi maximum likelihood estimator. The whole procedure reveals the fact that we are actually minimizing the average KLIC of our sample when performing the QMLE method. If we choose the normal density function as the likelihood function then we obtain the Gaussian Quasi Maximum Likelihood Estimator (GQMLE). GQMLE has been widely used in the estimation of GARCH models, partially due to some good properties owned by the estimator as well as the relatively convenient calculation.

If we use the standard normal density for the function f above, then we can obtain the GQMLE by the following maximization:

$\arg\max_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T} (-\log(\sigma v_t) - \frac{x_t^2}{2\sigma^2 v_t^2})$. This can be verified by the following steps:

If we use the normal density function as the likelihood function, then we have

$$f_t(x_t \mid \mathbf{x_{t\text{-}1}}; \boldsymbol{\theta}) = \frac{1}{\sigma_t \sqrt{2\pi}} e^{-\frac{x_t^2}{2\sigma_t^2}} = \frac{1}{\sigma v_t \sqrt{2\pi}} e^{-\frac{x_t^2}{2\sigma^2 v_t^2}}.$$

Therefore:

$$\arg\max_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t=1}^{T} \log(f_t(x_t \mid \mathbf{x_{t\text{-}1}}; \boldsymbol{\theta}))$$

$$= \arg\max_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t=1}^{T} \log(\frac{1}{\sigma v_t \sqrt{2\pi}} e^{-\frac{x_t^2}{2\sigma^2 v_t^2}})$$

$$= \arg\max_{\boldsymbol{\theta}} \frac{1}{T} \sum_{t=1}^{T} (-\log(\sigma v_t) - \frac{x_t^2}{2\sigma^2 v_t^2}).$$

3.2 Rank Estimation

In the previous section, the QMLE is introduced as a method of estimating the GARCH model. In this section, a nonparametric estimation method will be introduced, which is called the rank estimation. Andrews (2012) proposed this method for the estimation of GARCH models with heavy-tailed innovations. And in the simulation study, she was able to show that this rank-based method outperformed the traditional GQMLE when the true distribution is heavy-tailed.

The construction of rank estimation mainly depends on minimizing the R-function, which is defined as:

$$D_T(\boldsymbol{\gamma}) = \sum_{t=1}^{T} \lambda\left(\frac{R_t(\boldsymbol{\gamma})}{n+1}\right)(\xi_t(\boldsymbol{\gamma}) - \overline{\xi(\boldsymbol{\gamma})}) \quad ,$$

where $\boldsymbol{\gamma} = (a_1 ... a_p, b_1 ... b_q)$, which means $\boldsymbol{\theta} = (\sigma, \boldsymbol{\gamma})$.

$\xi_t(\boldsymbol{\gamma}) = \ln(X_t^2) - \ln(v_t^2(\boldsymbol{\gamma}))$, $\overline{\xi(\boldsymbol{\gamma})} = \dfrac{\sum_{t=1}^{T} \xi_t(\boldsymbol{\gamma})}{T}$, $R_t(\boldsymbol{\gamma})$ is the rank of $\xi_t(\boldsymbol{\gamma})$.

$\lambda(\bullet)$ is a nondecreasing and nonconstant function that maps from (0,1) to $\mathbb{R}$.

The reason we use $\xi_t(\boldsymbol{\gamma}) = \ln(X_t^2) - \ln(v_t^2(\boldsymbol{\gamma}))$ as residuals instead of $\dfrac{x_t}{\sigma v_t(\boldsymbol{\gamma})}$ is that as the elements in $\boldsymbol{\gamma}$ increase, the $v_t(\boldsymbol{\gamma})$ will also increase and $\dfrac{x_t}{\sigma v_t(\boldsymbol{\gamma})}$ will decrease as a consequence. If this is the case, then no minimizer can be found for $D_T(\boldsymbol{\gamma})$. Therefore, we choose the transformation $\ln(X_t^2) - \ln(v_t^2(\boldsymbol{\gamma}))$.

Then we can obtain the rank estimator by :

$$\hat{\boldsymbol{\gamma}} = \arg\min_{\gamma} D_T(\boldsymbol{\gamma}),$$

$$\hat{\sigma} = \frac{1}{T}\sum_{t=1}^{T} \frac{X_t^2}{v_t^2(\hat{\boldsymbol{\gamma}})}.$$

There are lots of choices for the weight function; for example: $\lambda(x) = [\Phi^{-1}((x+1)/2)]^2 - 1$, $\lambda(x) = \dfrac{7(F_{t7}^{-1}((x+1)/2))^2 - 5}{(F_{t7}^{-1}((x+1)/2))^2 + 5}$ and the Wilcoxon weight function $\lambda(x) = 2x - 1$.

3.3 Three-step Non-Gaussian Quasi Maximum Likelihood Estimation

In section 3.1 we introduced the GQMLE that chooses the standard normal density function as the likelihood function. However, as mentioned in chapter 2, there is evidence that financial data show heavier tails than the tail from the normal distribution. Under certain assumptions, even if the true density function is not normal, the GQMLE will still be consistent. But the efficiency of this method will be relatively low if we believe that the true distribution is heavy-tailed. One intuitive remedy for such problem will be to use some density functions from heavy-tailed distributions. Everything comes with a price. Each time we choose a different density function, to ensure the consistency property of the estimator, we need to add new assumptions. In some cases, the assumptions we need to ensure are too strict in practice.

To solve this problem, Fan (2014) proposed a three-step estimation procedure, and introduced a scale adjustment parameter η to deal with the identification issue in NGQMLE. The scale adjustment parameter can be obtained by minimizing the Kullback-Leibler Information Distance between $g(\varepsilon)$ and $\frac{1}{\eta} f(\frac{\varepsilon}{\eta})$:

$$\arg\min_{\eta>0} \int \log(\frac{g(\varepsilon)}{\frac{1}{\eta} f(\frac{\varepsilon}{\eta})}) g(\varepsilon) d\varepsilon = \arg\max_{\eta>0} E_g[-\log\eta + \log f(\frac{\varepsilon}{\eta})].$$

Therefore, $\eta_f = \arg\max_{\eta>0} E_g[-\log\eta + \log f(\frac{\varepsilon}{\eta})]$.

This parameter will be the key in the three-step estimation method. In practice, since we do not know the true density $g(x)$, we can obtain an estimate of η_f in the following way:

$$\hat{\eta}_f = \arg\max_{\eta} \frac{1}{T}\sum_{t=1}^{T}(-\log(\eta) + \log f(\frac{\varepsilon_t}{\eta})).$$

ε_t is also unknown to us; to obtain an estimate, we need to perform a preliminary estimation.

A complete three-step non-Gaussian quasi maximum likelihood estimation procedure will be described as follows:

Optimization Step1: Use GQMLE for preliminary estimation:

$$\tilde{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T}(-\log(\sigma v_t) + \frac{x_t^{\ 2}}{2\sigma v_t^{\ 2}}).$$

We can get the estimate for ε_t by $\tilde{\boldsymbol{\theta}}$ in this step.

Optimization Step2: Estimate $\hat{\eta}_f$ from the estimated residuals $\tilde{\varepsilon}_t$ from the first step:

$$\hat{\eta}_f = \arg\max_{\eta} \frac{1}{T}\sum_{t=1}^{T}(-\log(\eta) + \log f(\frac{\tilde{\varepsilon}_t}{\eta})).$$

Optimization Step3: Perform NGQMLE with the chosen likelihood function together with $\hat{\eta}_f$:

$$\hat{\boldsymbol{\theta}}_T = \arg\max_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T}(-\log(\hat{\eta}_f\, \sigma v_t) + \log f(\frac{x_t}{\hat{\eta}_f\, \sigma v_t})).$$

Then $\hat{\boldsymbol{\theta}}_T$ is the final NGQMLE from our three-step procedure.

We need to choose a likelihood function in the three-step NGQMLE mentioned above. However, from a theoretical view, there are certain limitations for the likelihood function we plan to choose.

The following requirements for the likelihood function are provided in Qi et al.(2010) to ensure one of the conditions for consistency of the three-step NGQMLE:

Suppose $\{\varepsilon_t\} \sim \varepsilon$ is i.i.d. with mean 0, variance 1 and a finite pth moment:

1. $f(x)$ is continuously differentiable up the second order.
2. $h(x) \le 0$ where $h(x) = x\dfrac{\dot{f}(x)}{f(x)}$, and $\dot{f}(x)$ is the first derivative of $f(x)$.
3. $x\dot{h}(x) \le 0$,and the equality holds if and only if x=0.
4. $|h(x)| = \dfrac{x^2\alpha}{\sqrt{1+x^2}} \le \alpha x^2$ and $\left|x\dot{h}(x)\right| \le C|x|^p$ for some constant $C > 0$ and $p > 0$.
5. $\limsup_{x->+\infty} h(x) < -1$.

A few density functions have already been mentioned in Fan et al.(2014) that can be used as the likelihood function in the three-step estimation procedure: standard normal density function, standard t density function with degrees of freedom greater than 2 and the generalized Gaussian density function with $\log f(x) = -|x|^{\beta}(\Gamma(3/\beta))/\Gamma(1/\beta))^{\beta/2}$ + constant.

In this section, the density function from the standard symmetric hyperbolic distribution will also easily be shown to satisfy the above conditions as well.

Result: if $f(x)$ is the density function of the standard symmetric hyperbolic distribution, then the five requirements will be satisfied.

The symmetric hyperbolic distribution with scale parameter 1 and location parameter 0 has the following density function:

$f(x) = \frac{1}{2K_1(\alpha)} \exp(-\alpha\sqrt{1+x^2})$, where K_1 denotes the modified Bessel function of the third kind with index 1 and α is a parameter that takes a positive value.

The density function of the standardized symmetric hyperbolic distribution is continuously differentiable up to the second order.

And we have $h(x) = x\frac{\dot{f}(x)}{f(x)} = -\frac{x^2\alpha}{\sqrt{1+x^2}}$, therefore $h(x) \le 0$.

We also have $x\dot{h}(x) = -\frac{x^2(2+x^2)\alpha}{(1+x^2)^{3/2}}$, $x\dot{h}(x) = 0$ if and only if $x = 0$, otherwise $x\dot{h}(x) < 0$.

$|h(x)| = \frac{x^2\alpha}{\sqrt{1+x^2}} \le \alpha x^2$ and $\left|x\dot{h}(x)\right| = \frac{x^2(2+x^2)\alpha}{(1+x^2)^{3/2}} < 10\alpha|x|$.

Lastly, we can see that $\limsup_{x->+\infty} h(x) < -1$.

Therefore, the standard symmetric hyperbolic distribution satisfies the five requirements mentioned above, and can be chosen as the likelihood function in the three-step NGQMLE.

3.4 A Modified Version of Three-step NGQMLE

The three-step estimation procedure introduced in this section utilizes GQMLE for preliminary estimation. However, as we have mentioned, GQMLE is not a very efficient method when the data has a heavy-tailed distribution. Therefore, the accuracy of $\tilde{\varepsilon}_t$ given by GQMLE may be doubted. As a result, the accuracy of $\hat{\eta}_f$ as well as the estimated scale parameter will be influenced as well.

As mentioned in 3.2, rank estimation can provide good performance in GARCH coefficients estimation when the data is heavy-tailed. Moreover, its computation is not difficult to implement. Therefore, seeking a better result from the three-step estimation, we shift the first step from GQMLE to rank estimation:

Optimization Step1: Use rank method for preliminary estimation

$$\tilde{\boldsymbol{\gamma}} = \arg\min_{\gamma} D_T(\boldsymbol{\gamma}),$$

$$\tilde{\sigma} = \frac{1}{T}\sum_{t=1}^{T}\frac{X_t^2}{v_t^2(\tilde{\boldsymbol{\gamma}})},$$

$$\tilde{\boldsymbol{\theta}} = (\tilde{\sigma}, \tilde{\boldsymbol{\gamma}}).$$

Optimization Step2: Estimate $\hat{\eta}_f$ from the estimated residuals $\tilde{\varepsilon}_t$ from the first step:

$$\hat{\eta}_f = \arg\max_{\eta} \frac{1}{T}\sum_{t=1}^{T}(-\log(\eta) + \log f(\frac{\tilde{\varepsilon}_t}{\eta})).$$

Optimization Step3: Perform NGQMLE with the chosen likelihood function together with $\hat{\eta}_f$:

$$\hat{\boldsymbol{\theta}}_T = \arg\max_{\boldsymbol{\theta}} \frac{1}{T}\sum_{t=1}^{T}(-\log(\hat{\eta}_f\,\sigma v_t) + \log f(\frac{x_t}{\hat{\eta}_f\,\sigma v_t})).$$

Then $\hat{\boldsymbol{\theta}}_T$ is the final NGQMLE from our three-step procedure. From this modified version of the three-step NGQMLE, we should be able to obtain a more accurate estimation of the scale adjustment parameter as well as the scale parameter, when the true distribution is heavy-tailed. And the simulation results provided in the next chapter will provide verification of this statement.

CHAPTER 4
Simulation Study

In this section, I will draw the following comparisons based on root-mean-square-error (RMSE): 1.Comparisons of estimated a_i and b_j from GQMLE and three-step NQGMLE; 2. Comparisons of estimated adjustment parameters using GQMLE in the first step and using rank-based in the first step; 3. Comparisons for estimated scale parameter from GQMLE, three-step NGQMLE and the modified three-step NGQMLE. The simulation of the GARCH series is accomplished with fGarch package in R. The GARCH model we choose is GARCH(1,1), which is the most commonly used GARCH model in practical modelling. The simulation iterations is 500 and the true parameters for GARCH model are: $\sigma = 0.5, a = 0.6, b = 0.3$.

If without specification, the default likelihood function chosen here for NGQMLE is the standardized t distribution with 9 degrees of freedom. In this simulation, t3,t4 and t6 are abbreviations for standardized t distribution with degrees of freedom 3,4 and 6; gg0.4 and gg0.6 stand for generalized Gaussian distributions with shape parameters 0.4 and 0.6. For the rank estimation here, I will choose: $\lambda(x) = \dfrac{7(F_{t7}^{-1}((x+1)/2))^2 - 5}{(F_{t7}^{-1}((x+1)/2))^2 + 5}$.

Table1 gives the RMSE of estimated a and b from GQMLE and three-step NGQMLE under several heavy-tailed distributions. As expected, the RMSE of the estimated parameters of the three-step NGQMLE is smaller than that of the GQMLE, which agrees with the simulation result shown by Fan et al.(2014). From the result, we can see that

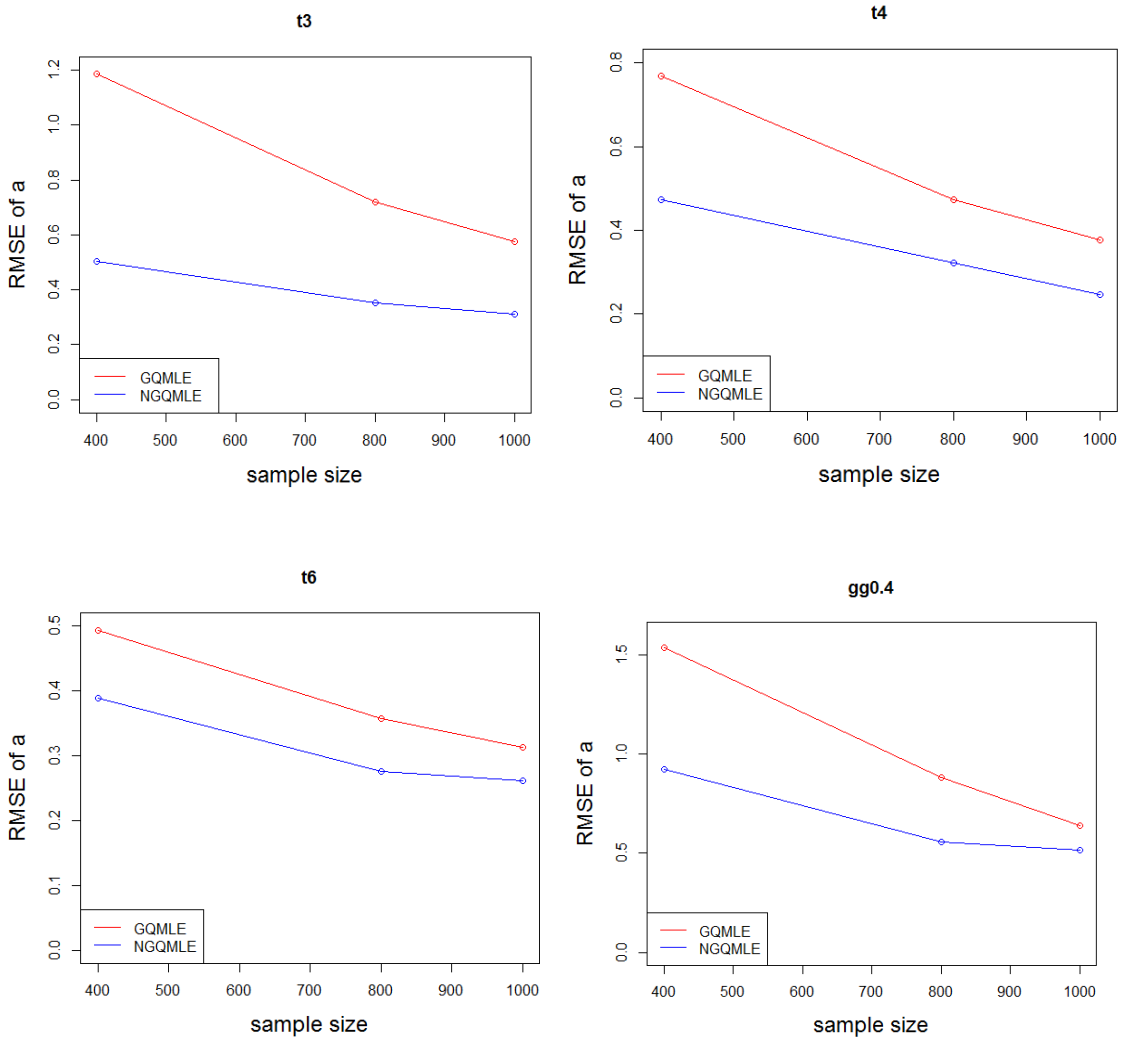
when the distributions tend to have heavier tails, the advantage of using three-step NGQMLE will become more obvious.

		RMSE of estimated α	
Distribution	Length	GQMLE	Three-step NGQMLE
t3	400	1.1882	0.5042
	800	0.7205	0.3533
	1000	0.5759	0.3118
t4	400	0.7692	0.4727
	800	0.4732	0.3218
	1000	0.3763	0.2471
t6	400	0.4935	0.3881
	800	0.3577	0.2755
	1000	0.3123	0.2609
gg0.4	400	1.5379	0.9252
	800	0.8828	0.558
	1000	0.6371	0.5155
gg0.6	400	1.0221	0.6531
	800	0.5255	0.3977
	1000	0.538	0.3726

Table4.1 RMSE comparison of estimated a from GQMLE and three-step NGQMLE

		RMSE of estimated b	
Distribution	Length	GQMLE	Three-step NGQMLE
t3	400	0.3329	0.2653
	800	0.2799	0.2179
	1000	0.2688	0.206
t4	400	0.3176	0.269
	800	0.2514	0.2086
	1000	0.2393	0.1966
t6	400	0.2887	0.2564
	800	0.2394	0.2164
	1000	0.2136	0.1959
gg0.4	400	0.3796	0.2974
	800	0.3318	0.2634
	1000	0.3191	0.2432
gg0.6	400	0.347	0.3045
	800	0.2938	0.245
	1000	0.2762	0.2301

Table4.2 RMSE comparison of estimated b from GQMLE and three-step NGQMLE



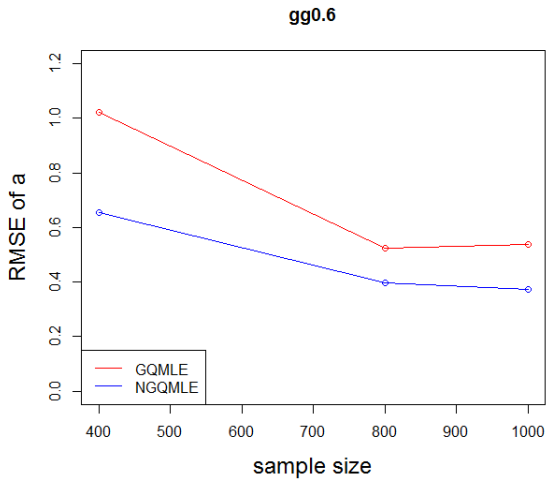
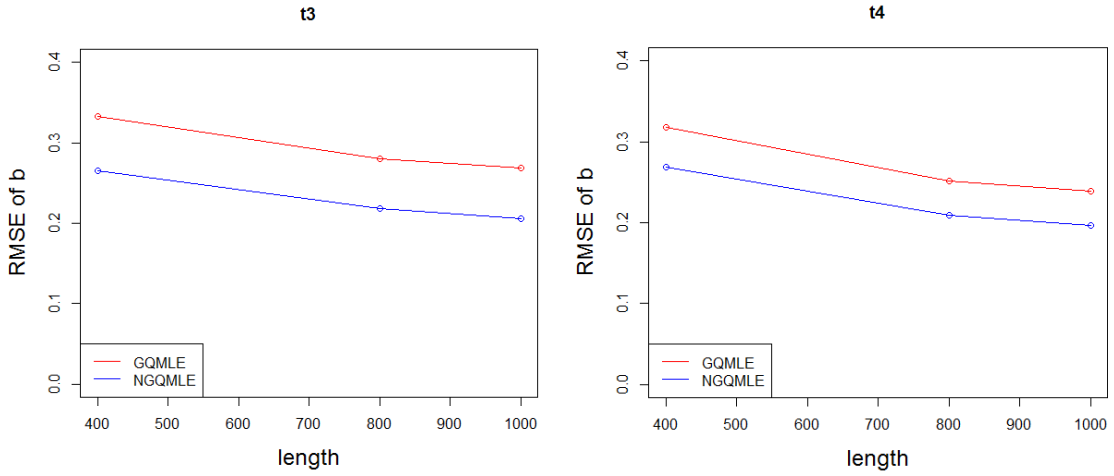


Figure 4.1 plot of RMSE of a under GQMLE and NGQMLE



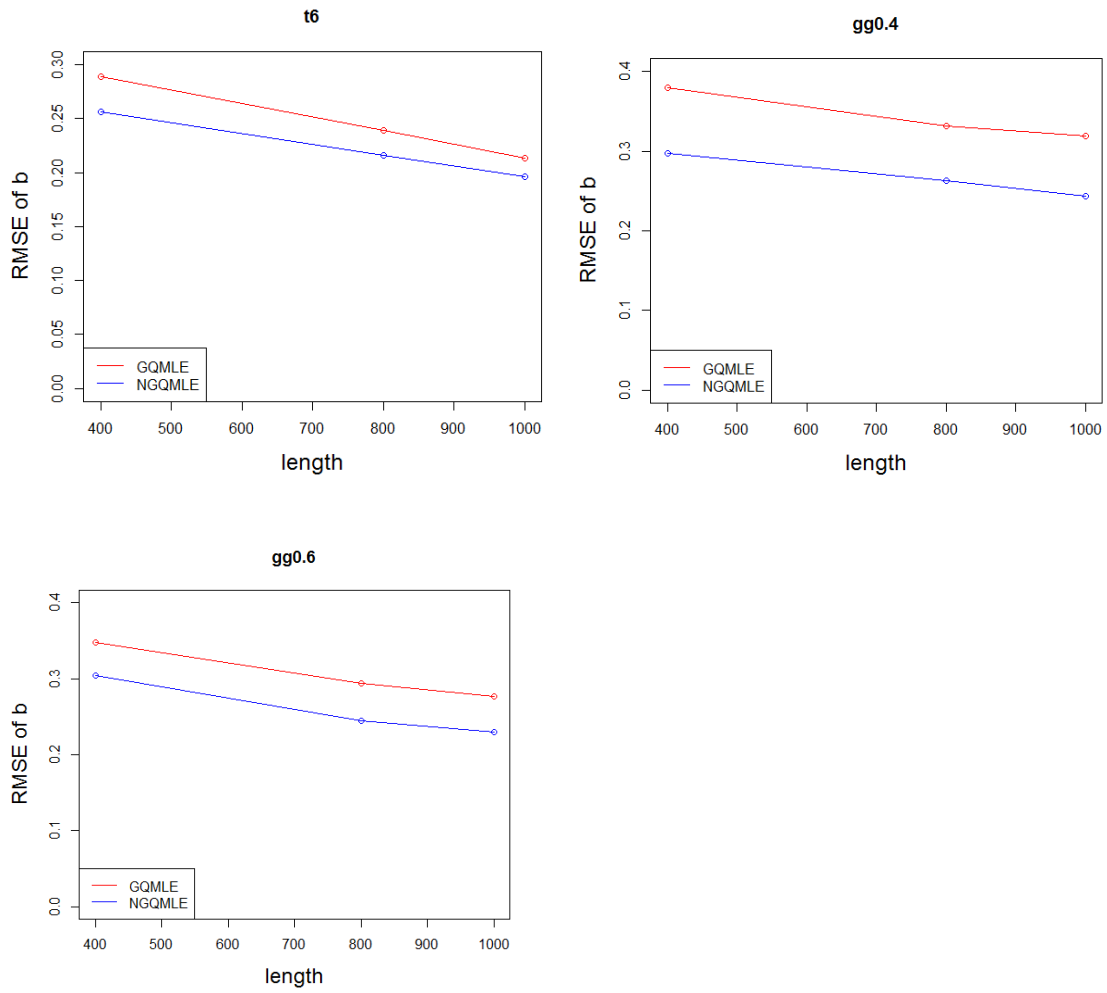


Figure4.2 plot of RMSE of b under GQMLE and NGQMLE

The accuracy of the estimated adjustment parameter is essential, since it will influence the performance of the estimated scale parameter. And the estimation of η_f depends on the estimated residuals we retrieve from the first step. The original first step applies GQMLE for preliminary estimation. Though GQMLE is easy to implement, its performance can be unsatisfying when the true distribution is heavy-tailed. In other words, the accuracy of the estimated η_f can be poor in this case. In this sense, applying

rank-based estimation in the first step will be better when the distribution is heavy-tailed, since compared to GQMLE, it can give us better estimates.

Table 2 will give us the RMSE of the estimated η_f, under the cases of GQMLE in the first step and rank-based estimation in the first step.

	RMSE(GQMLE)	RMSE(rank-based)
t3	0.4993	0.0895
t4	0.3168	0.0607
t6	0.1173	0.0249
gg0.4	0.3796	0.0667
gg0.6	0.2232	0.0415

Table4.3 RMSE for the adjustment parameter.

It can be shown from the table that when the tails of the true distribution gets lighter, the RMSE in both cases becomes smaller. However, the RMSE of the estimated adjustment parameter with GQMLE in the first step is always larger than the one under the rank-based method in the first step when the chosen distribution shows a heavy-tailed feature. As mentioned, the accuracy of the estimated adjustment parameter will influence the performance of the estimated scale parameter in the model. Based on the result given in table 2, we can conjecture that with the rank-based method in the first step, we can obtain a more accurate estimated σ under several typical heavy-tailed distributions. Table 3 give the result for RMSE of the estimated scale parameters in GARCH model, with GQMLE, three-step NGQMLE and the modified NGQMLE.

RMSE of estimated σ				
	T	GQMLE	NGQMLE	modified NGQMLE
t3	400	0.1805	0.1457	0.1427
	800	0.1778	0.1564	0.1103
	1000	0.161	0.1439	0.1004
t4	400	0.1586	0.1324	0.1252
	800	0.1292	0.109	0.0964
	1000	0.1236	0.1074	0.0841
t6	400	0.1335	0.1151	0.1145
	800	0.1053	0.093	0.0929
	1000	0.0931	0.0864	0.0849
gg0.4	400	0.2177	0.1745	0.1524
	800	0.2042	0.1722	0.1296
	1000	0.2015	0.1729	0.1173
gg0.6	400	0.1754	0.1519	0.1479
	800	0.1452	0.1227	0.117
	1000	0.1373	0.1124	0.1055

Table4.4 RMSE for the adjustment parameter, here we include the modified version of NGQMLE as well

From the table, we can find that under the chosen heavy-tailed distributions, the RMSE of the estimated scale parameter for the GARCH Model from the GQMLE method is always the largest, while the RMSE from the modified three-step NGQMLE is the smallest in all cases. And this advantage gets more obvious as the tails of the potential distribution get heavier. This indicates the advantage of using rank-based estimation in the first step instead of GQMLE, which can potentially further increase the performance of the estimators under heavy-tailed distributions. A similar result still holds from table 4.5, in which the symmetric standard hyperbolic distribution with shape parameter $a = 1$ is used as the likelihood function.

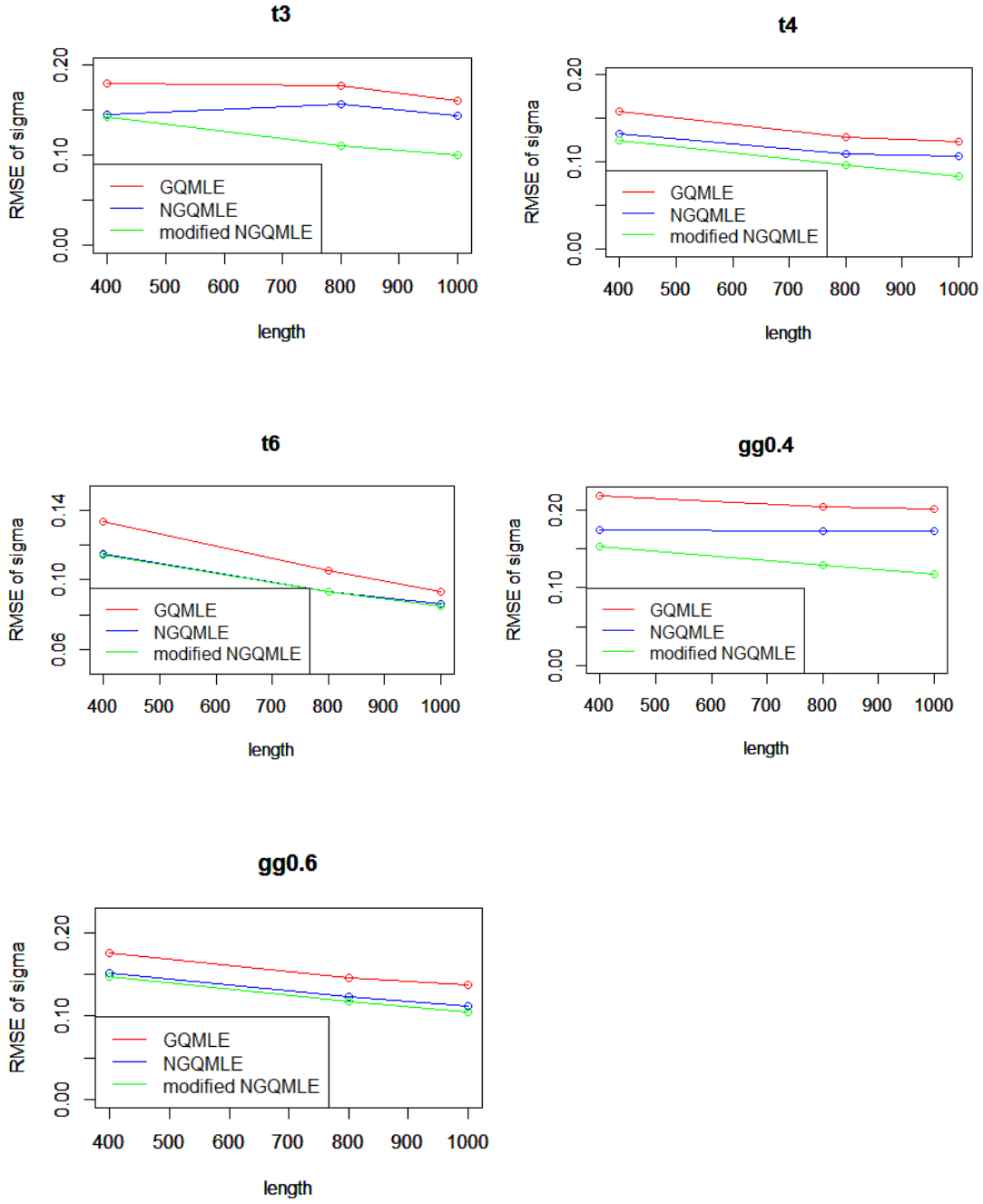
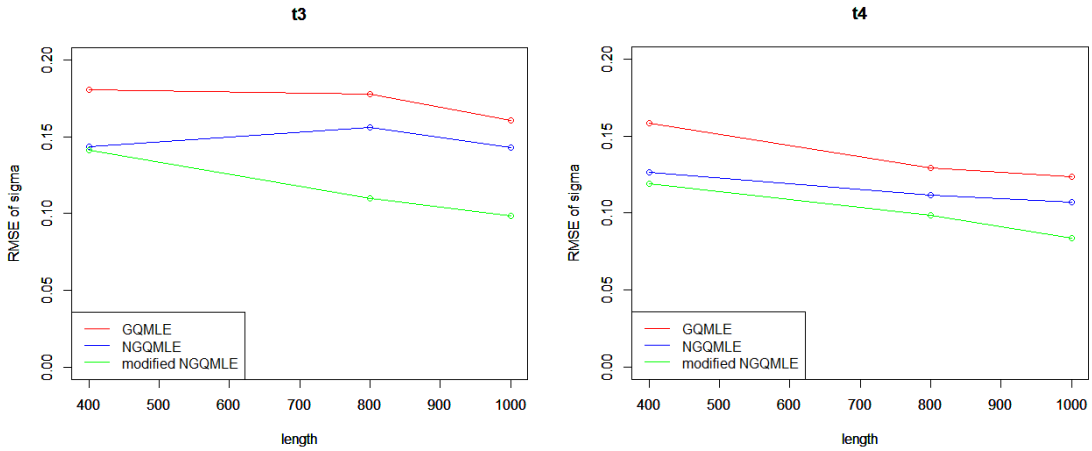


Figure4.3 plot of RMSE of sigma under GQMLE and two versions of NGQMLE

RMSE of estimated σ				
	T	GQMLE	NGQMLE	Modified NGQMLE
t3	400	0.1805	0.1437	0.1416
	800	0.1778	0.1562	0.1099
	1000	0.161	0.1433	0.0986
t4	400	0.1586	0.1267	0.1192
	800	0.1292	0.1118	0.0987
	1000	0.1236	0.1072	0.0836
t6	400	0.1335	0.1104	0.1098
	800	0.1053	0.0929	0.0927
	1000	0.0931	0.085	0.0835
gg0.4	400	0.2177	0.1737	0.1539
	800	0.2042	0.17	0.1269
	1000	0.2015	0.1705	0.1163
gg0.6	400	0.1754	0.1437	0.1399
	800	0.1452	0.115	0.1088
	1000	0.1373	0.1096	0.1027

Table4.5 RMSE for the scale parameter, using symmetric hyperbolic likelihood function



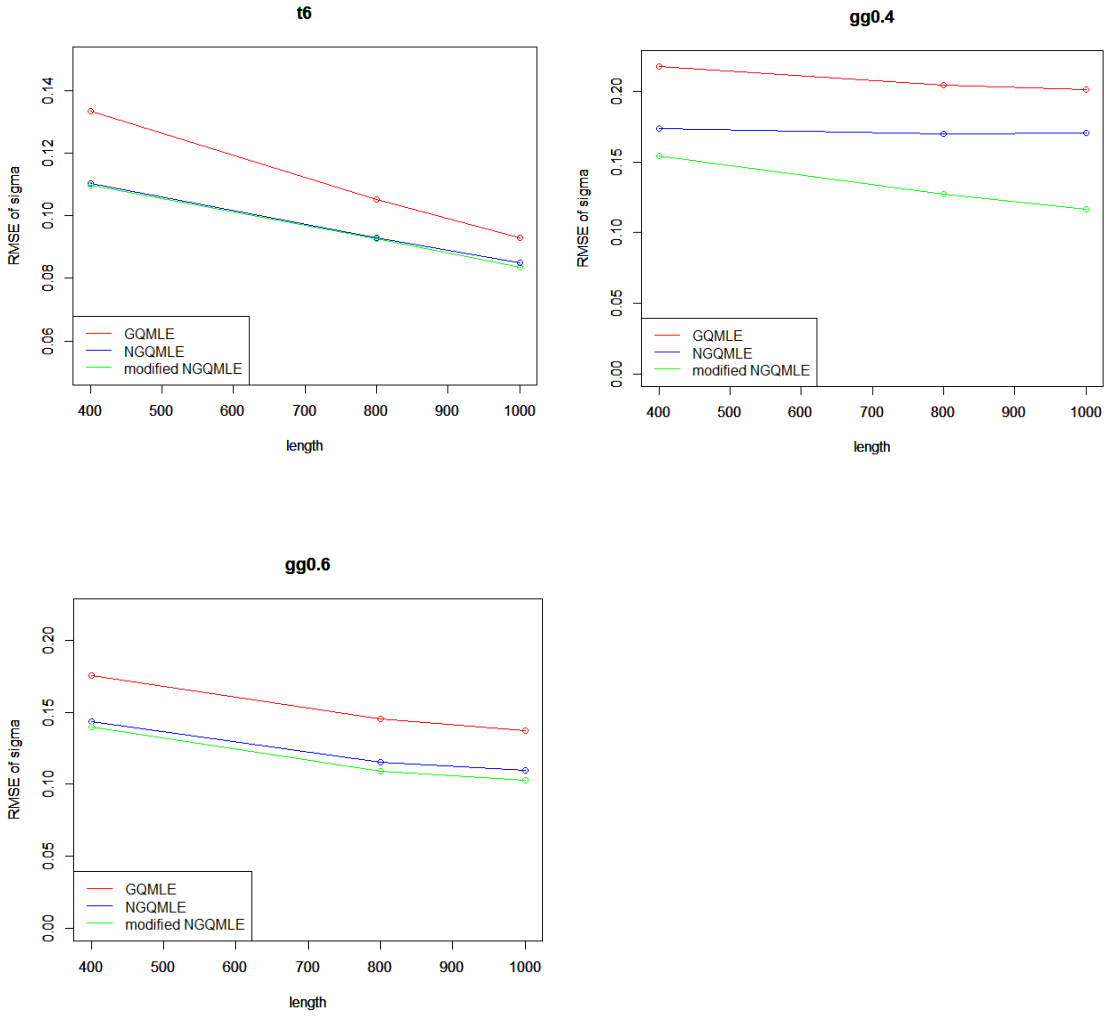


Figure4.4 plot of RMSE of sigma under GQMLE and two versions of NGQMLE, under symmetric hyperbolic likelihood function

CHAPTER 5
Empirical Study

The key idea in the three-step estimation method lies in using the scale adjustment parameter to improve the estimation accuracy of the scale parameter in NGQMLE, with the likelihood function we choose. The theoretical motivation as illustrated in the previous chapters is that if we choose a likelihood function from a non-normal density function, the estimator will not be consistent (it is actually the estimated scale parameter that will not be consistent) unless we change our model assumptions for each different likelihood function.

The normality assumption for the stock return data has been challenged by lots of empirical studies, and one of the most popular choices for a non-normal likelihood function is a standard t density function. To illustrate the necessity of using the three-step method for estimating a GARCH model, it is meaningful to study the estimated adjustment parameter for various stock data sets, and in this project we choose the likelihood function as t4. If a large proportion of the scale adjustment is close enough to 1, then the ordinary NGQMLE with t4 likelihood is good enough and the scale adjustment is not essential, otherwise, three-step method should be considered important in practice.

582 stocks traded in the Shanghai Stock Exchange are selected. The sampling frequency for the stock price data is daily, and the data is chosen from 1/1/2010 to 5/27/2016. Before we start the analysis, it is worth pointing out that we have transformed the price data to the log return data with :

$\log return_t = \log(p_t) - \log(p_{t-1})$, where p_t is the stock price at time t.

The first part in this empirical analysis is to test the normality from the datasets using Anderson-Darling test. This part illustrates the fact that most of the stock log return data does not follow a normal distribution, and NGQMLE is needed in reality.

The plot below shows the p-values for the log return data from the first 200 stocks. None of the p-values are larger than 5%. And actually the normality assumption is rejected by all the datasets from the 528 stocks.

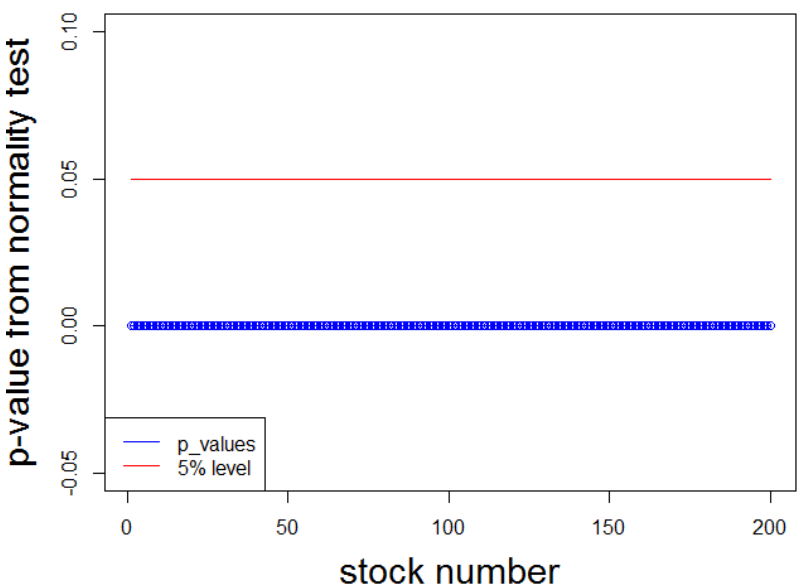


Figure5.1 p-values returned by the normality test for first 200 stock datasets

Next, each of the stock datasets is fitted using a GARCH(1,1) model using NGQMLE with likelihood function as standard t density function with 4 degrees of freedom. We obtain the estimated scale adjustment parameter in different ways: 1. use GQMLE in the

first step 2. use rank estimation in the first step. Therefore, we will ultimately have two different sets of scale adjustment parameters.

From the histograms, we can see that most of the estimated scale adjustment parameters are larger than 1 in both methods. And a large proportion of them are around 1.05~1.1, which is 5%~ 10% of deviation from 1.

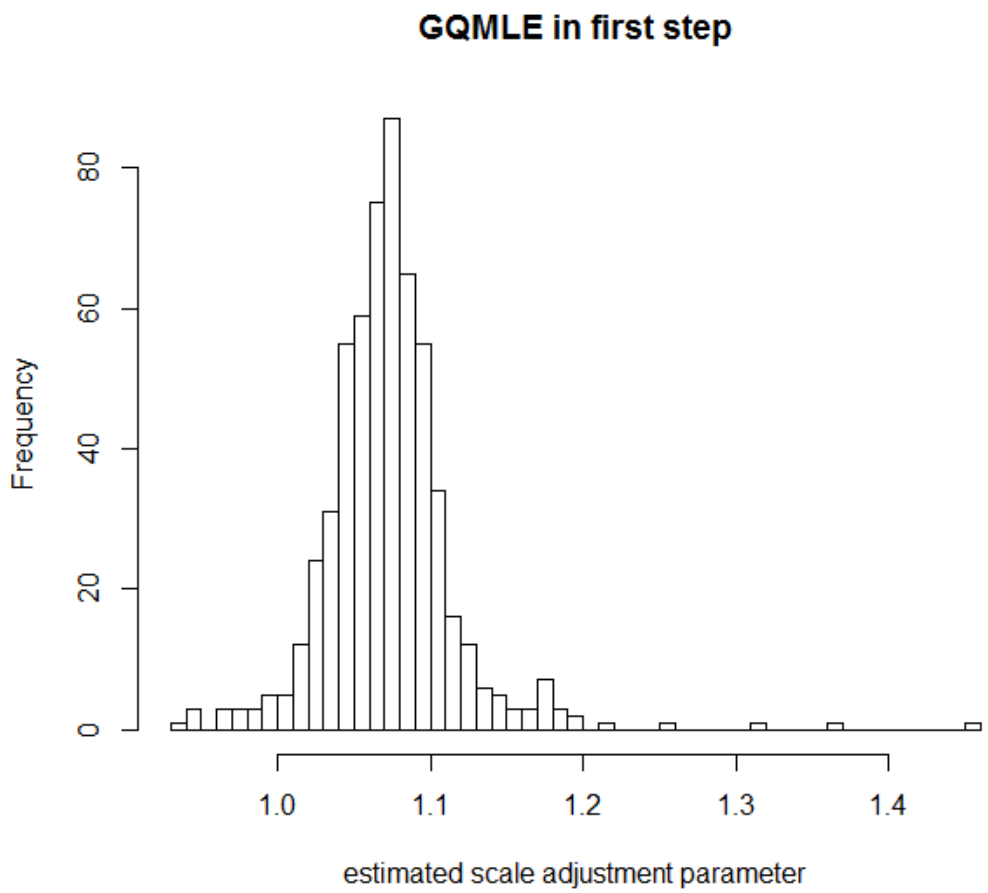


Figure5.2 histogram for estimated scale adjustment parameter from version1 NGQMLE

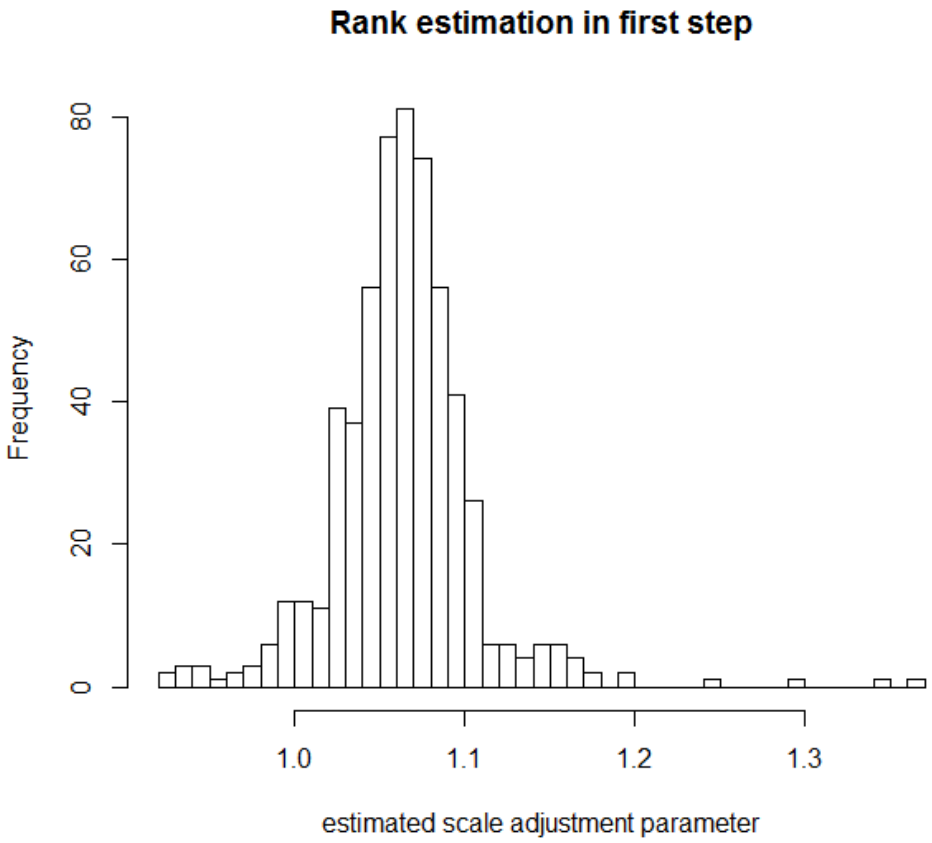
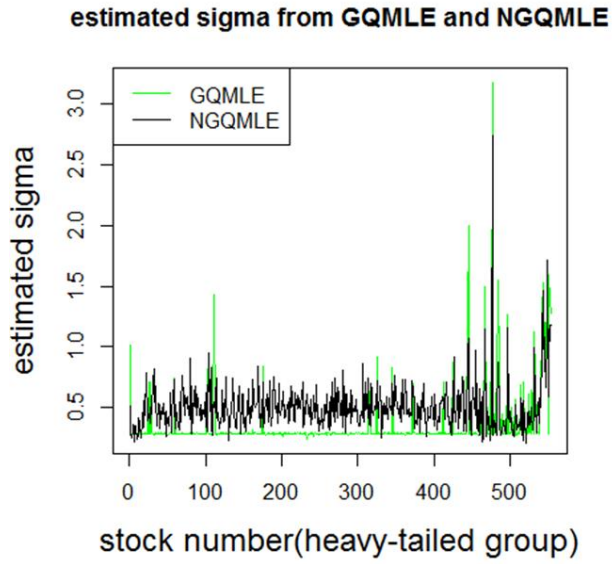
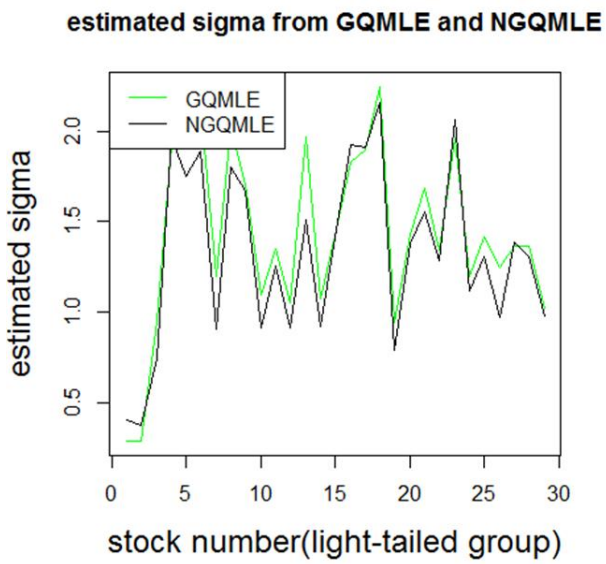
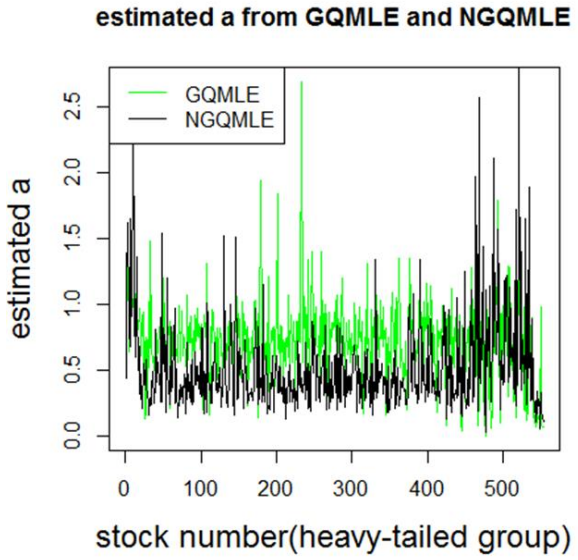
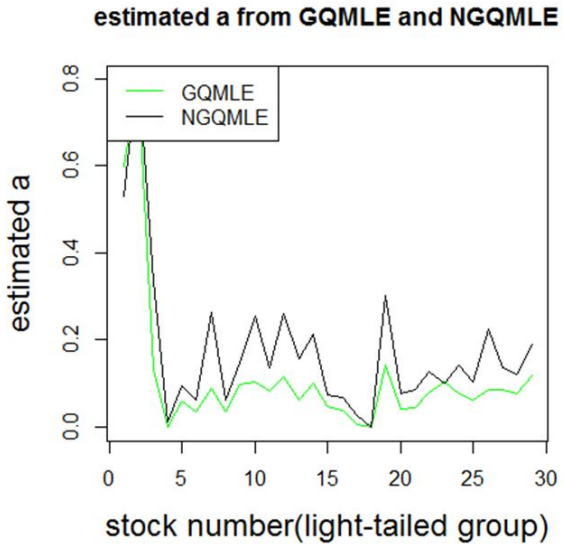


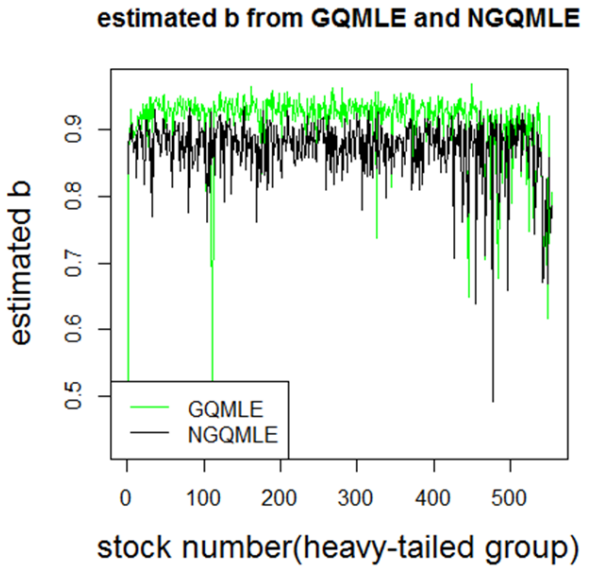
Figure5.3 histogram for estimated scale adjustment parameter from version2 NGQMLE


estimated sigma from GQMLE and NGQMLE
GQMLE
NGQMLE
estimated sigma
0.5
1.0
1.5
2.0
2.5
3.0
0
100
200
300
400
500
stock number(heavy-tailed group)


estimated sigma from GQMLE and NGQMLE
GQMLE
NGQMLE
estimated sigma
0.5
1.0
1.5
2.0
0
5
10
15
20
25
30
stock number(light-tailed group)


estimated a from GQMLE and NGQMLE
GQMLE
NGQMLE
estimated a
0.0
0.5
1.0
1.5
2.0
2.5
0
100
200
300
400
500
stock number(heavy-tailed group)


estimated a from GQMLE and NGQMLE
GQMLE
NGQMLE
estimated a
0.0
0.2
0.4
0.6
0.8
0
5
10
15
20
25
30
stock number(light-tailed group)



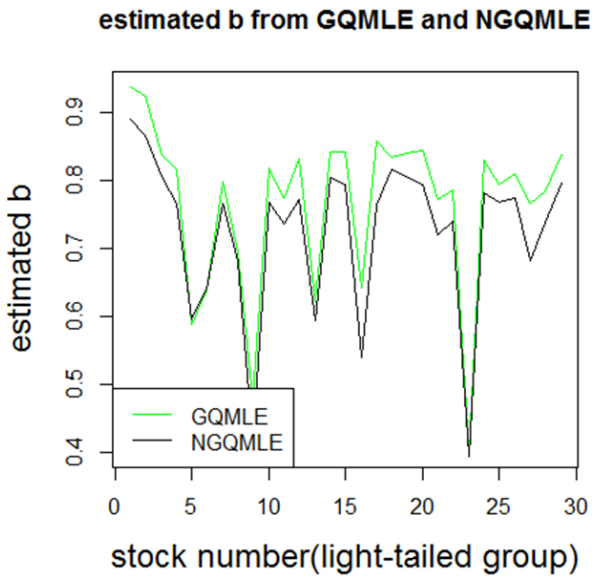


Figure5.4 Comparison of GARCH estimated parameters from GQMLE and NGQMLE under two different groups

We also draw comparisons for the coefficients estimated from GQMLE and the three-step NGQMLE(with GQMLE in the first step and t4 as likelihood function). From our simulation results in the previous chapter, we see that the three-step method provides

better performance when the datasets have heavy-tailed distribution. 95% of our datasets have kurtosis larger than 3, and we divide the datasets into two different groups based on their kurtosis. For the first group, the datasets have kurtosis larger than 3, which indicates that their distributions have heavier tails compared to normal. For the other group, the kurtosis of the datasets will be smaller than 3, which means that the tails of their distributions are lighter than normal. Then we draw comparisons separately under those two groups. From figure 5.4 we can see that GQMLE tends to underestimate sigma, overestimate a and b when the distributions of the datasets have heavier tails compared to normal.

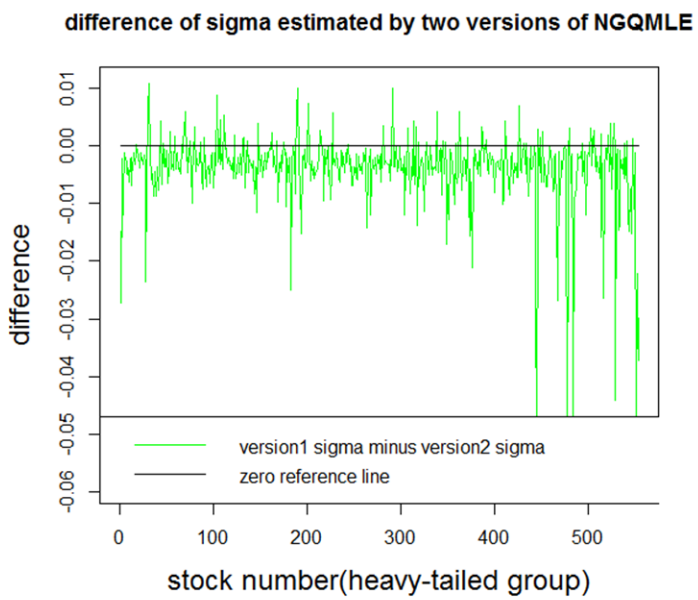


Figure5.5 difference of estimated sigma from two versions of NGQMLE

Due to the reason that the three-step estimation procedure aims at improving the accuracy of the estimated scale parameter. We also compare the estimated scale parameters

produced by those two versions of NGQMLE. From the graph above, we can actually discover that the original version of NGQMLE is more likely to produce smaller estimated scale parameters when the distributions of the datasets have heavier tails compared to normal.

Conclusion

In this project, we have introduced the GARCH model and its estimation methods. Due to the reason that the stock return data frequently has a heavy-tailed distribution, we point out the drawbacks of using GQMLE for GARCH estimation and introduce the three-step NGQMLE. Moreover, to obtain higher estimation accuracy under the heavy-tailed situations, we introduce a modified version of the three-step procedure, which uses the rank estimation in the first step instead of GQMLE. In our simulation study, the RMSE of the estimated coefficients given by the three-step method are lower, compared to the ones given by GQMLE. In addition, the RMSE of the estimated scale parameter are lower with our modified version of three-step method compared to the ones from the original version. In the empirical study, 528 stock return datasets from the Shanghai Stock Exchange Market are chosen, and none of them pass the normality test. Finally, comparisons of the coefficients estimated with three-step method and GQMLE are provided under two different groups. We find that GQMLE tends to underestimate sigma, overestimate a and b when the distributions of the datasets have heavier tails compared to normal. And the original version of NGQMLE is more likely to produce smaller estimated scale parameters compared to the ones provided by the modified version when the distributions of the datasets have heavier tails compared to normal.

References

Andrews, B. (2012), “Rank-Based Estimation for GARCH Processes”, Econometric Theory, 28, 1037–1064.

Bollerslev, T. (1986), “Generalized Autoregressive Conditional Heteroskedasticity”, Journal of Econometrics, 31, 307–327.

Duan, J.C. (1995) “The GARCH Option Pricing Model”, Mathematical Finance, 5,13–32.

Eberlein, E. and Keller, U. (1995) “Hyperbolic Distributions in Finance”, Bernoulli, 1(3), 281–299.

Engle, R. F. (1982), “Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation”, Econometrica, 50,987–1007

Fan, J., Qi, L. and Xiu, D. (2014), “Quasi-Maximum Likelihood Estimation of GARCH Models with Heavy-Tailed Likelihoods”, Journal of Business and Economic Statistics

Mikosch,T. and Starica, C. (2000), “Limit Theory for the Sample Autocorrelations and Extremes of a GARCH (1,1) Process”, Annals of Statistics, 1427–1451

Qi, L. , Xiu, D. and Fan,J. (2010), ”Non-Gaussian Quasi Maximum Likelihood Estimation of GARCH Models”, Working paper

Appendix
Basic Statistical Concepts

1.1 Kurtosis

The kurtosis is the fourth standardized moment, which is defined as:

$$Kurtosis[X] = \frac{E[(X-\mu)^4]}{(E[(X-\mu)^2])^2}.$$

Kurtosis can be used to see whether the distribution of the data has a heavier tail compared to the normal distribution. The kurtosis of a random variable X with a normal distribution is 3.

1.2 Anderson-Darling normality test

Test statistics:

$A^2 = -n - S$, $S = \sum_{i=1}^{n} \frac{2i-1}{n}[\ln(\Phi(X_i)) + \ln(1-\Phi(X_{n+1-i}))]$ and $\Phi(x)$ is the normal cumulative distribution function.

The null hypothesis is that the data follows a normal distribution.

1.3 Consistency

Let $\mathbf{x} = (x_1, ...x_n)$ be the sample and $T_n(\mathbf{x})$ be an estimator for the parameter θ. Then $T_n(\mathbf{x})$ is said to be consistent if and only if $T_n(\mathbf{x})$ converges to θ in probability.

1.4 Autocorrelation function(ACF)

The ACF is defined as:

$$\rho(t+h,t) = Corr(X_{t+h}, X_t) = \frac{E[(X_{t+h}-\mu_{t+h})(X_t-\mu_t)]}{std(X_{t+h})std(X_t)}.$$

R codes

```
#simulation part

library("fGarch")

library("rugarch")

library("GeneralizedHyperbolic")


sig=0.5

alpha1=0.6

beta1=0.3


a0=sig^2

a1=sig^2*alpha1

b1=beta1

##############################################################################

#####################

#preparation part: adjustment parameter and specification of garch models


# t likelihood

t_fdf=9

t_mlog=function(y,x){

  bb=sum(log(y)-log(dstd(x/y, mean = 0, sd = 1, nu = t_fdf, log = FALSE)))/length(x)

}
```

```
# try hyperbolic likelihood:
hy_mlog=function(y,x){
  bb2=sum(log(y)-log(dhyperb(x/y, mu = 0, delta = 1, alpha = 1, beta = 0)))/length(x)
}

#define the degree of freedom for t
t_truedf1=3
t_truedf2=4
t_truedf3=6
#shape parameter for generalized Gaussian(gg)
g_truedf1=0.4
g_truedf2=0.6
g_truedf3=1

#estiamte the scale adjustment parameter
findadj=function(obj,sample){
  out=nlm(obj,1,x=sample)
  adj=out$estimate
  return(adj)
}

#this part is to find the approximation to the true scale adjustent paraters
#case1: t distribution , t likelihood & hyperbolic likelihood
```

```
randomt1=rstd(10000, mean = 0, sd = 1, nu = t_truedf1)
randomt2=rstd(10000, mean = 0, sd = 1, nu = t_truedf2)
randomt3=rstd(10000, mean = 0, sd = 1, nu = t_truedf3)

t_adj1=findadj(t_mlog,randomt1)
t_adj2=findadj(t_mlog,randomt2)
t_adj3=findadj(t_mlog,randomt3)

t_adj1h=findadj(hy_mlog,randomt1)
t_adj2h=findadj(hy_mlog,randomt2)
t_adj3h=findadj(hy_mlog,randomt3)

cat("the adjustment parameter estimated when true distribution is t7, t11,
t20:",t_adj1,t_adj2,t_adj3,"\n")

#case2: gg distribution , t likelihood & hyperbolic likelihood
randomg1=rged(10000, mean = 0, sd = 1, nu = g_truedf1)
randomg2=rged(10000, mean = 0, sd = 1, nu = g_truedf2)
randomg3=rged(10000, mean = 0, sd = 1, nu = g_truedf3)

g_adj1=findadj(t_mlog,randomg1)
g_adj2=findadj(t_mlog,randomg2)
g_adj3=findadj(t_mlog,randomg3)
```

```
g_adj1h=findadj(hy_mlog,randomg1)
g_adj2h=findadj(hy_mlog,randomg2)
g_adj3h=findadj(hy_mlog,randomg3)

#specification of garch series with t distribution with df 7,11,20
t_spec1 = garchSpec(model = list(omega=a0, alpha = a1, beta = b1,
shape=t_truedf1),cond.dist = "std")
t_spec2 = garchSpec(model = list(omega=a0, alpha = a1, beta = b1,
shape=t_truedf2),cond.dist = "std")
t_spec3 = garchSpec(model = list(omega=a0, alpha = a1, beta = b1,
shape=t_truedf3),cond.dist = "std")

#specification of garch series with gg distribution with df 0.5,1,1.5
g_spec1 = garchSpec(model = list(omega=a0, alpha = a1, beta = b1,
shape=g_truedf1),cond.dist = "ged")
g_spec2 = garchSpec(model = list(omega=a0, alpha = a1, beta = b1,
shape=g_truedf2),cond.dist = "ged")
g_spec3 = garchSpec(model = list(omega=a0, alpha = a1, beta = b1,
shape=g_truedf3),cond.dist = "ged")
```

```
##########################################################################
###########################

#estimation part

true=c(sig,alpha1,beta1)

times=500
obs1=400
obs2=800
obs3=1000


#t_likelihood of NGQMLE with adjustment
ngfitgarch=function(the,x){
  v=vector(length=length(x))
  likeli=vector(length=length(x)-1)
  v[1]=var(x)
  for (t in 2:(length(x))){
    v[t]=1+the[2]*(x[t-1])^2+the[3]*(v[t-1])
    likeli[t-1]=log(sqrt(v[t])*the[1])-log(dstd(x[t]/(the[1]*sqrt(v[t])), mean = 0, sd = 1, nu
= t_fdf, log = FALSE))
  }
```

```
  return(sum(likeli/(length(x)-1)))
}
#hyperbolic_likelihood of NGQMLE with adjustment
ngfitgarch2=function(the,x){
  v=vector(length=length(x))
  likeli=vector(length=length(x)-1)
  v[1]=var(x)
  for (t in 2:(length(x))){
    v[t]=1+the[2]*(x[t-1])^2+the[3]*(v[t-1])
    likeli[t-1]=log(sqrt(v[t])*the[1])-log(dhyperb(x[t]/(the[1]*sqrt(v[t])), mu = 0, delta = 1,
alpha = 1, beta = 0))
  }
  return(sum(likeli/(length(x)-1)))
}

#GQMLE: normal likelihood
gfitgarch=function(the,x){
  v=vector(length=length(x))
  likeli=vector(length=length(x)-1)
  v[1]=var(x)
  for (t in 2:(length(x))){
    v[t]=1+the[2]*(x[t-1])^2+the[3]*(v[t-1])
```

```
  likeli[t-1]=log(sqrt(v[t])*the[1])-log(dnorm(x[t]/(the[1]*sqrt(v[t])), mean = 0, sd = 1,
log = FALSE))
  }
  return(sum(likeli/(length(x)-1)))
}

#estimation from GQMLE
gqmle=function(sample,obs,true){
  x=1:(obs+1)
  x[2:(obs+1)]=sample
  x[1]=mean(sample)
  out1 <- nlminb( start=true1,
                objective=gfitgarch,
                control = list(eval.max=3000,iter.max=3000, rel.tol=1.0e-14,abs.tol = 1.0e-
20,x.tol=1.0e-14,step.min = 2.2e-14),
                lower=c(-Inf,1.0e-8,1.0e-8),
                upper=c(Inf,Inf,1-(1.0e-8)),
                x = x )
  para=out1$par
  return(para)
}

#get the residual from fitted model
```

```
get_res=function(sample,para){
  h=vector(length=length(sample))
  res=vector(length=length(sample)-1)
  h[1]=(sample[1]/para[1])^2
  for(i in 2:length(sample)){
    h[i]=1+para[2]*(sample[i-1])^2+para[3]*h[i-1]
    res[i-1]=sample[i]/sqrt(h[i])/para[1]
  }
  return(res)
}

#error from NGQMLE with known adj
err_ng=function(sample,obs,adj,true,obj){
  x=1:(obs+1)
  x[2:(obs+1)]=sample
  x[1]=mean(sample)
  out1 <- nlminb( start=true1,
                objective=obj,
                control = list(eval.max=3000,iter.max=3000, rel.tol=1.0e-14,abs.tol = 1.0e-
20,x.tol=1.0e-14,step.min = 2.2e-14),
                lower=c(-Inf,1.0e-8,1.0e-8),
                upper=c(Inf,Inf,1-(1.0e-8)),
                x = x )
```

```
  para=out1$par
  para[1]=para[1]/adj
  err=(true-para)^2
  return(err)
}

#error from NGQMLE with unknown adj
err_ng2=function(likelihood,sample,residual,obs,true,obj){
  adj=findadj(likelihood,residual)
  x=1:(obs+1)
  x[2:(obs+1)]=sample
  x[1]=mean(sample)
  out1 <- nlminb( start=true1,
                objective=obj,
                control = list(eval.max=3000,iter.max=3000, rel.tol=1.0e-14,abs.tol = 1.0e-
20,x.tol=1.0e-14,step.min = 2.2e-14),
                lower=c(-Inf,1.0e-8,1.0e-8),
                upper=c(Inf,Inf,1-(1.0e-8)),
                x = x )
  para=out1$par
  para[1]=para[1]/adj
  err=(true-para)^2
  return(err)
```

```
}

# function helps to find RMSE for GQMLE and NGQMLE
rmse_gestimate=function(times,obs,spec,true,adj1,adj2){
  cat=matrix(, nrow = 1, ncol=24)
  cat1=matrix(rep(0,24), nrow = 1, ncol=24)
  for (i in 1:times){
    sample=as.vector(garchSim(spec, n=obs))
    gpara=gqmle(sample,obs,true)
    residual=get_res(sample,gpara)
    err1=(true-gpara)^2
    cat[1,1:3]=err1
    err2=err_ng(sample,obs,adj1,true,ngfitgarch)
    cat[1,4:6]=err2
    err3=err_ng2(t_mlog,sample,residual,obs,true,ngfitgarch)
    cat[1,7:9]=err3
    r_coef=rank_es(sample)
    rank_residual=rank_res(sample,r_coef)
    rsig=rank_sig(rank_residual)
    r_coef1=c(rsig,r_coef)
    rank_residual2=get_res(sample,r_coef1)
    err4=(true-r_coef1)^2
    cat[1,10:12]=err4
```

```
    err5=err_ng2(t_mlog,sample,rank_residual2,obs,true,ngfitgarch)
    cat[1,13:15]=err5
    err6=err_ng(sample,obs,adj2,true,ngfitgarch2)
    cat[1,16:18]=err6
    err7=err_ng2(hy_mlog,sample,residual,obs,true,ngfitgarch2)
    cat[1,19:21]=err7
    err8=err_ng2(hy_mlog,sample,rank_residual2,obs,true,ngfitgarch2)
    cat[1,22:24]=err8
    cat1=cat1+cat
  }
  rmse=sqrt(cat1/times)
  return(rmse)
}

result=function(spec,adj1,adj2){
  result_rmse=matrix(,nrow=3,ncol=24)
  result_rmse[1,]=rmse_gestimate(times,obs1,spec,true,adj1,adj2)
  result_rmse[2,]=rmse_gestimate(times,obs2,spec,true,adj1,adj2)
  result_rmse[3,]=rmse_gestimate(times,obs3,spec,true,adj1,adj2)

colnames(result_rmse)=c("Gsig","Galpha","Gbeta","NG1sig","NG1alpha","NG1beta","N
G2sig","NG2alpha","NG2beta","ranksig","rankalpha","rankbeta","rankNGsig","rankNGa
lpha","rankNGbeta","HY_NGsig","HY_NGalpha","HY_NGbeta","HY_NG2sig","HY_N
```

```
G2alpha","HY_NG2beta","HY_RANKNGsig","HY_RANKNGalpha","HY_RANKNGb
eta")
  return(result_rmse)
}



#t3distribution
t3_rmse=matrix(,nrow=3,ncol=24)
t3_rmse=result(t_spec1,t_adj1,t_adj1h)

#t4 distribution
t4_rmse=matrix(,nrow=3,ncol=24)
t4_rmse=result(t_spec2,t_adj2,t_adj2h)

#t6 distribution
t6_rmse=matrix(,nrow=3,ncol=24)
t6_rmse=result(t_spec3,t_adj3,t_adj3h)

#gg distribution with 0.4 shape
gg0.4_rmse=matrix(,nrow=3,ncol=24)
gg0.4_rmse=result(g_spec1,g_adj1,g_adj1h)

#gg distribution with 0.6 shape
```

```
gg0.6_rmse=matrix(,nrow=3,ncol=24)

gg0.6_rmse=result(g_spec2,g_adj2,g_adj2h)


#gg distribution with 1.5 shape

gg1.5_rmse=matrix(,nrow=3,ncol=24)

gg1.5_rmse=result(g_spec3,g_adj3,g_adj3h)
```