Abstract

Wu, Weiwei. Estimating Value at Risk and The Expected Shortfall for Heteroscedastic Financial Log Returns: a Two-stage Method. (Under the supervision of Dr. Peter Bloomfield)

Value at Risk and the Expected Shortfall are two measurements of market risks for financial assets. Statistically, they are extreme quantiles of the distribution of financial log returns. Though financial log return data are usually both heteroscedastic and fatter-tailed, most of the existing methods in literature only deal with one of the two properties.

Motivated by McNeil and Frey (2000), we propose a two-stage model, which is a combination of a tree-structured GARCH(1,1) and a revised version of the Generalized Pareto Distribution(GPD). In the first-stage model, both the number and the value of the tree nodes are chosen by maximizing some conditional reduction of negative log likelihood or the AIC criterion. In the second stage, the shape parameter of the GPD is defined as a linear function of the log estimated volatilities obtained from the first-stage model. This two-stage model not only considers both of the two data properties, but also allows the model to be different for different extents of market changes. Simulations show that our proposed model has advantages when the underlying model is classical GARCH(1,1) with t innovation, or the underlying model is tree-structured GARCH(1,1). We also applied the proposed method to historical log returns of the NASDAQ index and the MRK stock price.

ESTIMATING VALUE AT RISK AND THE EXPECTED SHORTFALL FOR HETEROSCEDASTIC FINANCIAL LOG RETURNS: A TWO-STAGE METHOD

by

WEIWEI WU

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

STATISTICS

Raleigh

2003

APPROVED BY:

Professor Peter Bloomfield Chair of Advisory Committee	Professor Jean-Pierre Fouque
Professor David A. Dickey	Professor Sujit K. Ghosh

Dedication

To my parents.

Biography

Weiwei Wu was born and grew up in Shaanxi Province, P. R. China. She graduated with a B.A. of Economics from the Department of Statistics, Renmin University of China, in 1996, and then a M.S. of Economics from the same department in 1999. She entered the Department of Statistics, North Carolina State University in August of 1999 for her doctoral study. And She expects to get her Ph.D. degree in Statistics in May 2004.

Acknowledgements

First, I truly thank my advisor, Dr. Peter Bloomfield, for consistently giving me patient guidance during the whole process of my research. Second, I thank my committee members, Dr. David A. Dickey, Dr. Jean-Pierre Fouque, and Dr. Sujit K. Ghosh for their enlightening comments on my research, and also for the very interesting courses I took from them.

Though I never met Dr. Peter Buhlmann, the author of "Tree-Structured Generalized Autoregressive Conditional Heteroscedastic Models", I thank him for answering my questions about his article via emails. I also want to express my thanks to Ms. Sandra Donaghy, the SAS consultant in the Department of Statistics. Whenever I had technical problems with SAS, she was always there for me to consult.

Finally, I thank all the current and past graduate students in the Department of Statistics, especially those who are (or once were) in the Hillsborough graduate office. They accompanied me during my whole study and research. And I also thank the department for supporting me financially all the time.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Market Risk, Value at Risk, and The Expected Shortfall

In finance, risk can be broadly defined as the degree of uncertainty about future net returns of some financial assets. Generally speaking, financial institutions mainly face four types of risk[1], namely: credit risk; operational risk; liquidity risk and market risk. Among these four types of risk, market risk is the most prominent one. It measures the unexpected changes caused by the market movements in the prices or rates of the underlying traded assets. Here by saying changes we usually mean losses, since investors are concerned about losses much more than gains for obvious reasons.

[1]We only list the most common types here. There are also other kinds of risk in addition to these four. For example, the legal risk.

Losses have different meanings for investors holding different positions. For investors holding a long position, a decrease of the asset prices means loss, whereas for those holding a short position, an increase of the asset prices means loss. Because of the large increase in the amount of traded assets, such as stocks, indices, and options and so on, market risk has become a primary concern for both the regulators and the financial institutions for risk management purposes.

To measure the market risk of financial assets, the most widely used tool is *Value at Risk* (henceforth, VaR). The definition of VaR can be stated as the following: For a given time horizon L and probability[2] p, the *Value at Risk* (VaR) is the loss in market value of a portfolio over the time horizon L that is exceeded with probability $1-p$. Analytically for a long position we have:

$$\Pr\left(P_t - P_{t+L} \geq \text{VaR}\right) = 1-p \text{ or } \Pr\left(P_{t+L} - P_t \leq -\text{VaR}\right) = 1-p, \tag{1.1}$$

where P_t is the price of that portfolio at time t. If we have the distribution of the price changes, or the so-called *Profit and Loss* distribution of a given portfolio, then VaR is just a quantile of it. Because p is usually at least 0.95, it is a high quantile. This can be seen clearly from Figure 1.1, which illustrates Equation 1.1.

Originally, VaR was used only as an internal risk management tool by a number of banks and financial institutions, including J. P. Morgan, which developed the famous RiskMetrics methodology and published it in 1994. Then the group charged with this task was spun off from J. P. Morgan and started their new business: using this

[2]In the literature on financial risk, p is also referred as a "confidence level". We avoid that usage because of the obvious conflict with statistical conventions.

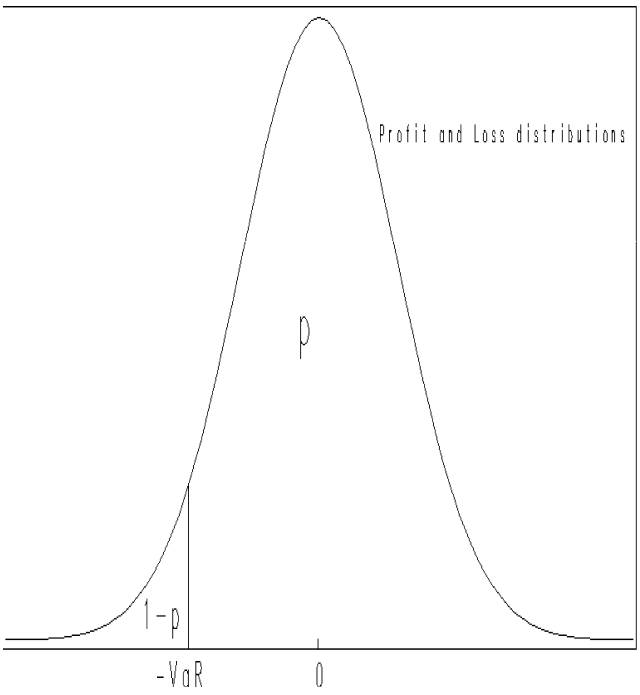


Figure 1.1: Definition of VaR

system to carry out VaR calculations for other banks, corporations, mutual funds and so on. Soon VaR became the most commonly used measure of market risk. It is used by investment institutions not only to manage their risk, to evaluate the performance of risk takers, but also to meet some regulatory requirements. For example, J. P. Morgan discloses its daily VaR at 95% level (that is, $L = 1$ day and $p = 0.95$), and Bankers Trust discloses its daily VaR at 99% level (i. e., $L = 1$ day and $p = 0.99$). In particular, the Basel Committee on Banking supervision (1996) at the Bank for International Settlements require financial institutions to set capital margins based

on VaR for $L = 10$ days and $p = 0.99$.

After 1994, many articles and books on VaR appeared in both the finance and the statistics literature. For a general exposition of VaR, we refer to Duffie and Pan (1997) [10], and Jorion (1997) [15].

Though VaR is the most popular measure of market risk, it has been criticized by many researchers as well. A commonly recognized shortcoming of VaR is that it only gives an upper bound on the losses that occur with a given probability. It tells nothing about the size of the potential loss given that a loss exceeding this bound has occurred. Furthermore, Artzner et al. (1997, 1999) [2, 3] point out that VaR is not a "coherent" measure of risk. They define *coherent measure of risk* to be a measure that has four properties: monotonicity, sub-additivity, positive homogeneity, and translation invariance. Here we use the mathematical definition of the four properties in Acerbi and Tasche (2002) [1]:

Consider a set V of real-valued random variables. A function $\rho : V \to R$ is called *coherent measure* if it is

1. monotonic: $X \in V, X \geq 0 \Longrightarrow \rho(X) \leq 0$,

2. sub-additive: $X, Y, X + Y \in V \Longrightarrow \rho(X + Y) \leq \rho(X) + \rho(Y)$,

3. positively homogeneous: $X \in V, h > 0, hX \in V \Longrightarrow \rho(hX) = h\rho(X)$, and

4. translation invariant: $X \in V, a \in R \Longrightarrow \rho(X + a) = \rho(X) - a$.

Among the four properties, sub-additivity means that the total risk of a portfolio

made of sub-portfolios is at most the sum of the risks of each sub-portfolio. For measures without this property, investment diversification may not reduce the total amount risked. This is somewhat contradictory to the common practices. Unfortunately, VaR is not sub-additive. To overcome this, Artzner et al. (1997, 1999) [2, 3] proposed a new measure called *the Expected Shortfall* . It is the conditional expected loss given that the loss exceeds VaR. That is, it is $E[\text{Loss}|\text{Loss} > \text{VaR}]$[3].

The introduction of the Expected Shortfall adds little difficulty to statistical modelling and the estimation. When the distribution is known, estimating the conditional expectation is no more difficult than estimating a quantile. So after the Profit and Loss distribution is set up, it is easy to get the Expected Shortfall in addition to VaR. Thus, it is ideal if a fitted model can give good estimates to both VaR and the Expected Shortfall.

1.2 Value at Risk Methodologies

1.2.1 Features of Financial Return Series

As Equation 1.1 indicates, VaR is usually defined in dollar amount: it is given as a quantile of the absolute change of some financial asset's price. But as most other financial studies, modelling and estimating VaR usually involves returns instead of prices. Let P_t be the price of an asset at time t; holding an asset for one period from

[3]Notice that "loss" is $P_t - P_{t+L}$ when $P_t < P_{t+L}$ in Equation 1.1.

date $t-1$ to date t would result in a *log return* :

$$r_t = \ln \frac{P_t}{P_{t-1}} \tag{1.2}$$

Notice that by Equation 1.2, a negative r_t means loss, and a positive one means gain. Also, r_t corresponds approximately to the percentage change of price, because P_t and P_{t-1} are usually very close to each other.

Using log return r_t instead of the price P_t in Equation 1.1, now VaR is defined as

$$\Pr\left(r_t \leq -\text{VaR}\right) = 1 - p. \tag{1.3}$$

There are two main reasons to use log returns rather than the original prices: one is that it is scale-free, and another is that it has some very attractive statistical properties. It has been well known for a long time, and has been proved by empirical research, that financial return series, especially stock return series, have the following features:

- they have fatter tails than the normal distribution;
- the return series themselves have almost zero-autocorrelations;
- they have positive serial correlations in the second moment, or volatility.

Thus, to estimate VaR is to estimate a high quantile of the distribution of log return series, which has the above features.

1.2.2 Existing Methods for VaR Estimation

In the past ten years or so, quite a few methods have been developed for VaR estimation in the literature. Some people classify them as conditional methods and unconditional methods, or parametric, semi-parametric, and nonparametric methods. Actually the model development is so fast that some complicated models appeared, which assume a hybrid way and are thus hard to classify. We enumerate the main methods here, mentioning the related more complicated models after each of them.

Historical Simulation (HS)

A common method for VaR assessment is the *historical simulation* (HS), in which the estimated distribution of returns is simply the empirical distribution of the past observations. The advantages of this methods is that it makes no distributional assumptions, it is nonparametric, and is therefore easy to implement. But it also has some very serious drawbacks: First, the estimated distribution is discrete. Second, it can not make out-of-sample estimation. Third, when there is a very large observation in the sample, the quantiles obtained by HS can be greatly affected by it. And finally, the choice of sample size can have large impact on the value predicted. To overcome the disadvantages of the historical simulation method, some variations of it are proposed in literature. For example, Boudoukh, Richardson and Whitelaw (1998) [5] propose a method which applies exponentially declining weights to the past return observations. And Butler and Schachter (1996) [6] propose a variation

which uses a kernel smoother to estimate the distribution of log returns.

RiskMetrics and GARCH Family

At the beginning of this section, we listed three features of the financial log return series. The last two of them actually indicate that the log return series are heteroscedastic, which means that the error term, or innovation, of the underlying model does not have a constant variance. There is a well-known family of models which deal with time series with heteroscedastic errors: the *generalized autoregressive conditional heteroscedasticity* (GARCH) models. So the last two features easily lead us to use the GARCH family models.

Generally, a GARCH(p,q) model with mean equation μ_t, an autoregressive parameter p, and a moving average parameter q is defined by,

$$\begin{aligned} r_t &= \mu_t + \sigma_t \epsilon_t \\ \sigma_t^2 &= \alpha_0 + \textstyle\sum_{i=1}^{p} \alpha_i r_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \\ \epsilon_t &\sim (0,1) \end{aligned} \tag{1.4}$$

where $\alpha_0 > 0$,$\alpha_i \geq 0$ for $i = 1, 2, \ldots, p$, and $\beta_j \geq 0$ for $j = 1, 2, \ldots, q$. The innovation ϵ_t is often assumed to have a standard Normal or standardized Student-t distribution. And the most commonly used mean equation for μ_t is the autoregressive (AR) model.

In particular, a classical GARCH(1,1) with zero mean and normal innovation has the following form:

$$\begin{aligned} r_t &= \mu_t + \sigma_t \epsilon_t \\ \mu_t &= 0 \\ \sigma_t^2 &= \alpha_0 + \alpha_1 r_{t-1}^2 + \beta \sigma_{t-1}^2 \\ \epsilon_t &\sim N(0,1) \end{aligned} \tag{1.5}$$

Furthermore, there exist several variants of the GARCH model: the Integrated GARCH model, the Exponential GARCH model, and the GARCH-in-mean model. They are also called IGARCH, EGARCH, and GARCH-M model respectively for short. Among them the IGARCH model is actually an unit-root GARCH model. In fact, the popular RiskMetrics approach[4] is, in its simple form, an Integrated GARCH (1,1) model with normal innovations. An IGARCH(1,1) model assumes the following form:

$$\begin{aligned} r_t &= \mu_t + \sigma_t \epsilon_t \\ \mu_t &= 0 \\ \sigma_t^2 &= \lambda \sigma_{t-1}^2 + (1-\lambda) r_{t-1}^2 \\ \epsilon_t &\sim N(0,1) \end{aligned} \tag{1.6}$$

where λ is always a fixed number in the interval $(0.9, 1)$. That is, the GARCH(1,1) model is specialized to the case $\alpha_0 = 0$ and $\alpha_1 + \beta = 1$. The choice to use an IGARCH(1,1) instead of a classical GARCH(1,1) is made based on practical experiences: when classical GARCH(1,1) models are fitted to stock return data, it is commonly observed that α_0 is close to 0 and the summation of the two parameters α_1 and β in the volatility model is very close to 1.

Extending the RiskMetrics approach to other members of the GARCH family is a natural step. Also, to adapt to the fatter tail feature of return series, we can consider using, say, t distribution as the innovation instead of the common Normal distribution. Another idea is to make the parameters such as λ variable with time.

[4]See http://www.riskmetrics.com/index.html.

Extreme Value Theory (EVT)

The common idea in both the HS and the GARCH family models is to aim at modelling the distribution of the return first, then to get the percentile needed. However, many researchers argue that this is not appropriate. Even if we get a statistically well-fitted model this way, we can only say that the model fits most of the data, or the central part of the data well. However, it is the extreme data, or the tail part of the data that VaR estimation mainly deals with. This has given rise to the recent use of the *Extreme Value Theory* (EVT) in VaR estimation.

The mathematical foundation of EVT is given by Gnedenko (1943) [13], who proved the so-called *Extreme Value Theorem*. The theorem can be restated as the following [5]:

Suppose $X_1,\ X_2,\ \ldots,X_n,\ldots$ are independent random variables with a common cdf $F(x)$. Let $M_n = \min\{X_1, X_2, \ldots, X_n\}$ [6]; then for suitable normalizing constants $a_n > 0$ and b_n, as n goes to infinity, the limiting distribution of $\frac{M_n - b_n}{a_n}$ can only assume one of the 3 types of extreme value laws, namely, the Gumbel distribution (type 1), the Frechet distribution (type 2), and the Weibull distribution (type 3). These three limiting distributions are defined by Equation 1.7, Equation 1.8 and Equation 1.9 respectively:

[5] Here we use a restatement similar to that used in Smith (1999) [22].

[6] In the rest of this article, we will not distinguish long position and short position. Also, because $\max\{X_1, X_2, \ldots, X_n\} = -\min\{-X_1, -X_2, \ldots, -X_n\}$, we will not rigorously distinguish the maximum and the minimum since they make little difference in modelling and analysis.

$$H(x) = \exp(-e^{-x}) \quad \text{for } -\infty < x < \infty, \tag{1.7}$$

$$H(x) = \begin{cases} 0 & \text{for } x \leq 0, \\ \exp(-x^{-\alpha}) & \text{for } x > 0, \end{cases} \tag{1.8}$$

$$H(x) = \begin{cases} \exp(-(-x)^{\alpha}) & \text{for } x \leq 0, \\ 1 & \text{for } x > 0. \end{cases} \tag{1.9}$$

In both Equation 1.8 and Equation 1.9, we have $\alpha > 0$.

If $F(x)$ is a thin-tailed distribution such as the normal or the lognormal distribution, then the limiting distribution is Gumbel (type 1). If $F(x)$ is fatter-tailed such as the t distribution or the stable Paretian distribution, then we get Frechet distribution (type 2). The Weibull distribution is obtained when $F(x)$ has no tail. If we let X be the log return series, since financial return series is usually fatter-tailed, Frechet distribution is the type that concerns us.

EVT can be applied directly to VaR estimation. Nonparametric estimators of α are given by Hill (1975) [14] and Pickands (1975) [20], both of which are based on the bootstrap method. For parametric estimator, suppose we have $N \times n$ daily log returns. Divide the data into N non-overlapped time intervals, each interval containing n data points. From each of the N intervals, get M_n, the minimum of the n log returns. Then we get N minima. When n is sufficiently large, we know that EVT applies to M_n; and if N is also large, we have sufficient number of M_n to do model fitting. That is, if both N and n are large enough, we can use the N minima

to fit the Frechet distribution in Equation 1.8 directly, getting the MLE estimates of a_n, b_n and α. Then VaR, as a high quantile, can be obtained by using the relation $F_{M_n}(x) = 1 - (1 - F(x))^n$. We call this kind of method *the classical EVT* . For the classical EVT, we refer to Longin (1996) [16] and (2000) [17].

In the classical EVT, since we only use the minimum of each of the N non-overlapped intervals, for N and n to be large enough, a lot of data points are needed. To overcome this disadvantage, an alternative approach which is often used in environmental statistics can be applied instead. It is based on *the exceedances over high thresholds* :

If for variable X, we fix a high threshold u and look at the exceedance of u, which is denoted by $Y = X - u$, the distribution of the excess value is:

$$F_u(y) = \Pr\{X \leq u + y | X > u\} = \frac{F(u+y) - F(u)}{1 - F(u)}, y > 0. \tag{1.10}$$

According to Pickands (1975) [20], if $F(x)$ is fatter-tailed and is such that a EVT distribution exists, then as $u \to \infty$, $F_u(y)$ has the following limiting distribution:

$$G(y) = 1 - (1 + \xi y/\beta)^{-1/\xi} \tag{1.11}$$

where $\beta > 0$, $\xi \geq 0$ and the support is $y \geq 0$. This distribution is the so-called *generalized Pareto distribution* (GPD).

Usually the threshold approach is applied by fitting the GPD to the observed excesses over the threshold u. Then the natural question here is how to choose the threshold. There are several articles discussing the selection of a suitable u, and the treatment of time series dependence. Again, these articles are mainly in

the environmental area. Smith (1999) [22] suggests to use it in VaR estimation and proposes a variation to it. The variation is a change-point model based on hierarchical Bayesian structure. It models both the exceedance times and the excess values as a two-dimensional point process instead of just a GPD.

Other Methods

There are two other methods which are not included in the above enumeration, both of which model the evolution of the quantiles directly. One is called *quantile regression* . For this method, see chapter 7.4.2 of Tsay (2002) [23], Chernozhukov (2002) [7] and the references therein. Another is the *Conditional Autoregressive Value at Risk* , or CAViaR. See Engle and Manganelli (1999) [11].

The GARCH model, in particular the RiskMetrics model, is the most popular one used in practice, whereas the EVT-based models are the commonest models seen in the theoretical literature. The GARCH model considers the heteroscedastic property of the log return series. By using such innovations as the t distribution, it also considers the fatter-tail feature to some extent. But as stated above, it aims at modelling the whole distribution of the return, thus gives good estimates of such usual statistics as the mean and the variance of the return, rather than a high quantile such as VaR. Especially, some researchers believe that if the goal is to get the VaR at $p = 0.95$, RiskMetrics may be enough, but if $p = 0.99$ or higher is needed, the RiskMetrics

is not so useful. In practice, $p = 0.95$ is far from enough for risk management purpose. For example, for daily data $p = 0.95$ means the unprotected events happen once in every 20 days on average. This is just too risky to be acceptable to some investors. The EVT based models solve this problem by applying the Extreme Value Theorem, but at some expense: The Extreme Value Theorem assumes independence, which is not true for financial returns according to the last two features listed in Chapter 1.2.1. That is, EVT does not consider the heteroscedasticity. But EVT handles the fatter tail in a better way than the GARCH model. In addition, even when a t distribution is used, GARCH actually assumes the right tail and the left tail are symmetric. But in practice, we sometimes have good reasons to believe that the market may respond differently to upward oscillations and downward oscillations. Using EVT, we can model the right tail and the left tail respectively, thus reflecting this idea.

Now it seems combining these two popular methods together may be a good idea. McNeil and Frey (2000) [18] describe one such combination. They propose a two-stage method: first, model the returns as a classical GARCH(1,1), assuming normal innovations and using pseudo MLE to get the parameter estimates; then, construct the residuals of the first-stage model and fit them to GPD. The high quantile, VaR, is estimated according to both of the fitted models and their relationship. Such a two-stage model considers both the heteroscedasticity and the extreme quantile property of VaR.

Chapter 2

METHOD

2.1 The Motivation and Outline

The method proposed here is motivated by McNeil and Frey (2000) [18]. We keep the idea of building a two-stage model and combine the GARCH and the EVT together. Furthermore, we want to add the following thought to our consideration: The market may respond differently not only to upward and downward movements, but also to different extent of the movements. That is, when the market is experiencing very large movements, the underlying model may be different from that when the market is experiencing small movements. To realize this idea, we use the following two-stage model: in the first-stage model, a tree-structured GARCH(1,1) similar to that proposed by Audrino and Buhlmann (2001) [4] will be used, and the residuals of the model are calculated; in the second stage, GPD is used to model the residuals of the first-stage model, with the shape parameter being a function of the estimated

volatilities in the tree-structured GARCH(1.1). Then the VaR, as a predicted quantile of the tree-structured GARCH(1,1) with innovation being the fitted covariate GPD, is estimated from the fitted two-stage model. Likewise, the Expected Shortfall is obtained as a function of the estimated parameters and quantiles.

The tree-structured GARCH is chosen such that the classical GARCH(1,1) is a special case of it. Generally, we get a partition of the $R \times R^+$ space of (r, σ^2). For all the partition cells, we adapt a single mean equation, which is the same as the first line in Equation 1.5, the classical GARCH(1,1) model. But for different partition cells, we use different volatility equations: we still assume that all the volatility equations have the same form as that in Equation 1.5, whereas the values of $(\alpha_0, \alpha_1, \beta)$ are now different. Furthermore, the innovation term ϵ_t in the mean equation is assumed to have a standardized t distribution with the degree of freedom ν. Here ν is greater than 2, and is to be estimated together with other unknown parameters. The nodes (splits) of this partition are decided by a selection strategy, which maximizes some reduction of negative log likelihood or some specified criterion (for example, the AIC). Maximum likelihood method is used to get the parameter estimates. By using such a tree-structured GARCH model, we actually fit different GARCH(1,1) models for each of the partition cells. In another word, we use different GARCH(1,1) models for different extents of market movements (i.e., different σ^2). Also, because r_t is also partitioned, we may be able to reflect the asymmetry of the upward and downward movements. We hope that this tree-structured GARCH will make the estimation of the volatilities more accurate.

In the second-stage model, to make it also changeable with the volatility level, we model the shape parameter of the GPD as a linear function of the log values of the estimated volatilities obtained in the first-stage model, keeping the scale parameter a constant. This time, we do not fit multiple GPD, but fit only one GPD model for all the data. This is because when it is possible, a smooth model is always preferred to a non-smooth one. For a non-smooth model like the one we use in the first stage, because of the use of nodes, the values which are very close to the nodes may not be estimated as accurately as the others. This is simply decided by the nature of non-smooth models.

In the following two sections, the details of the modelling process are given for each of the two stages.

2.2 The First-stage Model: Tree-structured GARCH(1,1)

Let's redefine the working model by the following equation:

$$\begin{aligned} r_t &= \mu_t + \sigma_t(\theta)\epsilon_t \\ \mu_t &= \phi r_{t-1} \\ \sigma_t^2(\theta) &= f_\theta\{r_{t-1}, \sigma_{t-1}^2(\theta)\} \\ \epsilon_t &\sim t(\nu), \nu > 2. \end{aligned} \tag{2.1}$$

Except for the form of f_θ and the distribution of ϵ_t, model 2.1 differs from both the classical GARCH(1,1) model (see Equation 1.5) and the IGARCH(1,1) model (see Equation 1.6) used in RiskMetrics in that it has an autoregressive term ϕr_{t-1} in the

mean equation. This term is used to adapt to the real data, which may not have an exactly zero autocorrelation. Considering the almost zero-autocorrelation property of log returns, the estimate of ϕ should be close to zero.

Suppose $\mathcal{P} = \{\mathcal{R}_1, \ldots, \mathcal{R}_k\}$ is a partition of the $R \times R^+$ space (r, σ^2), where k is the number of partition cells. For every partition cell $\mathcal{R}_i$, we apply a different volatility equation in the classical GARCH(1,1) model. That is, f_θ has the following form:

$$f_\theta(r, \sigma^2) = \sum_{j=1}^{k} (\alpha_{0,j} + \alpha_{1,j} r^2 + \beta_j \sigma^2) I_{[(r,\sigma^2) \in \mathcal{R}_j]} \tag{2.2}$$

where $\theta = \{\alpha_{0,j}, \alpha_{1,j}, \beta_j; j = 1, \ldots, k\}$ with $\alpha_{0,j}, \alpha_{1,j}, \beta_j \geq 0$. As a special case, when $k = 1$, Equation 2.1 and Equation 2.2 give the classical GARCH(1,1) model with standardized t innovation.

By the above definition, the negative log likelihood of our working model is:

$$-l(\phi, \theta) = -\sum_{t=2}^{n} \log \left[\sigma_t^{-1}(\theta) * f_\epsilon \left(\frac{r_t - \phi r_{t-1}}{\sigma_t(\theta)} \right) \right] \tag{2.3}$$

where f_ϵ is the probability density function (PDF) of the innovation ϵ_t. In our case, we have:

$$f_\epsilon(x|\nu) = \frac{\Gamma((\nu+1)/2)}{\nu/2} \frac{1}{\sqrt{\nu\pi}} \left(1 + x^2/\nu\right)^{-\frac{\nu+1}{2}} \tag{2.4}$$

Notice that this negative log likelihood is conditional on the first observation r_1 and some starting value $\sigma_1^2(\theta) = var(r_1)$.

In order to find the MLE estimates of ϕ, ν and θ, we need to minimize the negative log likelihood given in Equation 2.3. Here notice that Equation 2.3 depends on the partition, i.e., depends on the selection of partition nodes through $\sigma_t(\theta)$. Because

we plan to carry out the selection by maximizing some reduction of negative log likelihood, the nodes selecting and parameter estimating processes actually can be combined together by using an algorithm[1], which is to be described in details in the next two subsections.

Forward: Selecting the Nodes

We'll select the nodes of the partition by constructing a binary tree. Figure 2.1 is an example of a simple binary tree. It is constructed by the following way: First, select a node of r, call it d_1, which partitions the support of r, R, into two cells: $a_1 = \{r \leq d_1\}$ and $a_2 = \{r > d_1\}$. Second, select one of the cells from the previous partition, say, $a_1 = \{r \leq d_1\}$, and a second node d_2 of σ^2, then use this node to partition the selected cell into two cells, which are $b_1 = \{r \leq d_1, \sigma^2 \leq d_2\}$ and $b_2 = \{r \leq d_1, \sigma^2 > d_2\}$. Finally, repeat the second step once. That is, again select one of the cells from the previous partition and a third node. This time assume the selected cell is $a_2 = \{r > d_1\}$, and the node d_3 is a node of σ^2. Then use this node to partition the selected cell into two cells, which are $c_1 = \{r > d_1, \sigma^2 \leq d_3\}$ and $c_2 = \{r > d_1, \sigma^2 > d_3\}$.

If the second step is done repeatedly until we get $k-1$ nodes $d_1, d_2, \ldots, d_{k-1}$, we get a partition $\mathcal{P}$ which has k cells $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k$. Notice that each node divides a

[1]This algorithm is similar to that in Audrino and Buhlmann (2001) [4], but with some differences. For example, the maximization routine, the definition of AIC, and the realization of Step 1 etc. Also, this algorithm is more detailed.

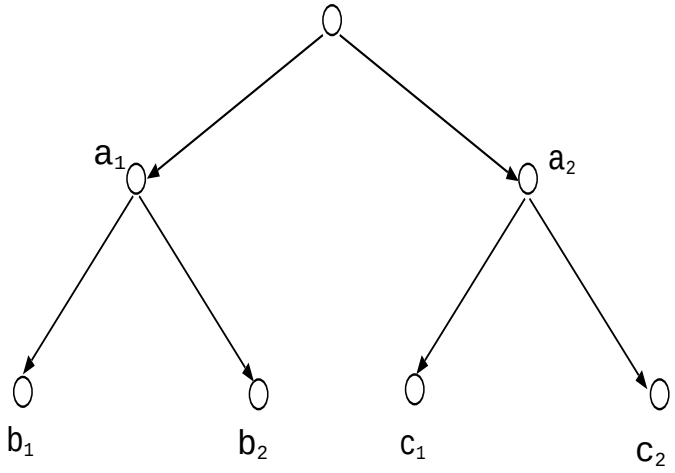


Figure 2.1: A Simple Binary Tree

selected cell from the previous partition into two new cells, one with a less-than-or-equal sign "$\leq$" and another with a greater-than sign ">". We denote the former new cell by $\mathcal{R}_{\text{left}}$ and the later one by $\mathcal{R}_{\text{right}}$.

For our problem, to partition the $R \times R^+$ space (r, σ^2), we use the following algorithm, which not only constructs a binary tree by optimizing the reduction of a conditional negative log likelihood, but also estimates the parameter vector (ϕ, ν, θ) at the same time.

Step 1: Let $\mathcal{P}_{opt}^{(0)} = R \times R^+$, i.e., $\mathcal{P}_{opt}^{(0)}$ is the whole space without any partition. Compute the negative log likelihood in Equation 2.3 with volatility function

$$f_{\theta^{(0)}}^{\mathcal{P}_{opt}^{(0)}}\left(r, \sigma^2\right) = \alpha_0 + \alpha_1 r^2 + \beta\sigma^2, \theta^{(0)} = (\alpha_0, \alpha_1, \beta) \in (R^+)^3. \tag{2.5}$$

Then minimize the negative log likelihood, get the estimates $\hat{\phi}^{(0)}$, $\hat{\nu}^{(0)}$ and $\hat{\theta}^{(0)}$. Set a counter $m = 0$.

To do the minimization, basically any nonlinear optimization method can be used. These methods can be easily found as build-in subroutines in popular statistical softwares. For example, SAS provides subroutines to realize the following nonlinear optimization methods: the Conjugate Gradient Method, the Double Dogleg Method, the Nelder-Mead Simplex Method, the Newton-Raphson (Ridge) Method, the (Dual) Quasi-Newton Method, the Quadratic Optimization Method, and the Trust-Region Method etc.[2] . Among them, the Quasi-Newton method[3] is the most popular one, and is the one used by McNeil and Frey (2000) [18]. Here in our method, we recommend and use the so-called Double Dogleg Method (see Dennis and Mei (1979)[9], and Gay (1983)[12]) in stead of the Quasi-Newton. This is because the Double Dogleg optimization technique works well for medium to moderately large optimization problems, and we find in reality it is more stable for our specified case.

The starting values for this step can be obtained by simply applying the SAS procedure PROC AUTOREG. We use this procedure to fit a simple GARCH(1,1) with standard Normal innovation. And by default, it calculates and uses the OLS (the Ordinary Least Square) estimates of the parameters as its own starting values.

[2]See the SAS online documentation at http://www.ncsu.edu/it/sas/help/ for details.

[3]Also called Quasi Gauss-Newton iteration method. It is an algorithm to get the quasi-likelihood estimates of parameters given the conditional mean and conditional variance equations containing the unknown parameters. See Nocedal and Wright (1999)[19] and Davidian (2001)[8].

Step 2: Increase m by 1. Find the best refined partition $\mathcal{P}_{opt}^{(m)}$ by binary dividing one of the cells from the previous partition $\mathcal{P}_{opt}^{(m-1)}$ into two new cells. The detailed process is listed as below:

(a) Given $\mathcal{P}_{opt}^{(m-1)} = \{\mathcal{R}_1, \ldots, \mathcal{R}_m\}$, consider a new partition $\mathcal{P}^{(m)}$, where only one partition cell $\mathcal{R}_{j^*} \in \mathcal{P}_{opt}^{(m-1)}$ is split into $\mathcal{R}_{j^*} = \mathcal{R}_{j^*_{left}} \cup \mathcal{R}_{j^*_{right}}$ as that described above. The new volatility function corresponding to $\mathcal{P}^{(m)}$ is

$$\begin{aligned} f^{\mathcal{P}^{(m)}}_{(\theta^{(m-1)\setminus *}, \theta^*)}(r, \sigma^2) = &\textstyle\sum_{j \neq j^*} (\alpha_{0,j} + \alpha_{1,j} r^2 + \beta_j \sigma^2) I_{[(r,\sigma^2) \in \mathcal{R}_j]} \\ &+ \textstyle\sum_{i \in \{j^*_{left}, j^*_{right}\}} (\alpha^*_{0,i} + \alpha^*_{1,i} r^2 + \beta^*_i \sigma^2) I_{[(r,\sigma^2) \in \mathcal{R}_i]}, \end{aligned} \tag{2.6}$$

where

$$\begin{aligned} \theta^{(m-1)\setminus *} &= \{\alpha_{0,j}, \alpha_{1,j}, \beta_j; j = 1, \ldots, m, j \neq j^*\} \in (R^+)^{3(m-1)}, \\ \theta^* &= \{\alpha^*_{0,i}, \alpha^*_{1,i}, \beta^*_i; i \in \{j^*_{left}, j^*_{right}\}\} \in (R^+)^6. \end{aligned}$$

(b) Using Double Dogleg method, minimize the negative conditional log likelihood in the refined partition $\mathcal{P}^{(m)}$ over θ^* and ν, holding the parameter vector $\hat{\phi}^{(m-1)}, \hat{\theta}^{(m-1)\setminus *}$ fixed. This time we have

$$\min_{\theta^*, \nu} \left[-l^{\mathcal{P}^{(m)}} \left\{ \hat{\phi}^{(0)}, \left(\hat{\theta}^{(m-1)\setminus *}, \theta^* \right) \right\} \right]. \tag{2.7}$$

Here $-l^{\mathcal{P}^{(m)}}$ is as in Equation 2.3 and the volatility function $f^{\mathcal{P}^{(m)}}$ is Equation 2.6. In this minimizing process, for the parameters in θ^*, use as starting values the components of $\hat{\theta}^{(m-1)}$ corresponding to the cell $\mathcal{R}_{j^*}$. And for ν, use $\hat{\nu}^{(m-1)}$ as the starting value.

(c) By varying $\mathcal{P}^{(m)}$ in (a) and re-computing (b), find the optimized Equation 2.7. Denote the optimal refined partition by $\mathcal{P}_{opt}^{(m)}$ and the corresponding

partition node by d_m. More details for this step will be given in the last paragraph of this subsection.

Step 3: Again, using Double Dogleg method, minimize the negative conditional log likelihood in Equation 2.3 corresponding to the partition $\mathcal{P}_{opt}^{(m)}$. This time use the volatility function $f^{\mathcal{P}_{opt}^{(m)}}$ from Equation 2.2 to get $(\hat{\phi}^{(m)}, \hat{\nu}^{(m)}, \hat{\theta}^{(m)})$. For this step, the starting values can be $\hat{\phi}^{(m-1)}$, $\hat{\nu}^{(m-1)}$, $\hat{\theta}^{(m-1)\backslash *}$ and the minimizer $\hat{\theta}^*$ in expression 2.7 that is obtained in Step 2. The resulting estimates in this step are denoted by $\hat{\phi}^m$,$\hat{(\nu)}$, and $\hat{\theta}^m$. Please notice that in this step we no longer fix any parameter in (ϕ, ν, θ). All the parameters are to be re-estimated simultaneously.

Step 4: Repeat Step 2 and Step 3 until $m = M$, where M is specified in advance. By doing this we get the partition $\mathcal{P}_{opt}^{(M)}$ which gives the final binary tree and the final parameter estimates $(\hat{\phi}^{(M)}, \hat{\nu}^{(M)}, \hat{\theta}^{(M)})$.

As to the value of M, it is pointed out by Audrino and Buhlmann (2001) [4] that for financial returns data, choosing M around six is appropriate. We will adapt this idea and use $M = 4$ unless specified otherwise. This means at the end of Step 4, we will have four nodes and five cells in the data-partition tree.

Now the only remaining issue is how to choose nodes to split cells in Step 2(a) and Step 2(c). Here a grid searching is proposed. We use the empirical α quantiles of r and those of the estimated σ^2 with $\alpha = 1/8, 2/8, \ldots, 7/8$. That is, for each cell of the previous partition, we search for $7 \times 2 = 14$ splits. And for each m, since we

have m cells, we actually go over Step 2(a) and 2(b) $m \times 14$ times in order to get the d_m and $\mathcal{P}_{opt}^{(m)}$ in Step 2(c).

Backward: Reducing the Number of Nodes

Because $M = 4$ is usually sufficiently large for financial data, the binary tree with five cells developed in the previous subsection may be too fine. Also, estimating too many parameters is always not encouraged in modelling, especially if the model is to be used in practice. So we consider using such measurement as the *Akaike Information Criterion* (AIC) instead of just the negative log likelihood to judge if one tree is better than the others. To do this, we apply the following method to reduce the number of nodes:

First, find the set T, which is the set of all the binary subtrees of $\mathcal{P}_{opt}^{(M)}$, the final tree we get from the previous subsection. Denote the elements of T by $\mathcal{P}_i$. Note that the "subtrees" are defined by dropping the decision nodes one by one, beginning from the end leaves of the tree. And the set T is usually larger than the set $\{\mathcal{P}_{opt}^{(0)}\mathcal{P}_{opt}^{(1)}, \ldots, \mathcal{P}_{opt}^{(M)}\}$. This is because there may be more than one node that can be dropped at some point during the dropping process, and it is not necessary to drop the nodes by the order they are constructed in the previous subsection.

To illustrate this, again we use the simple binary tree in Figure 2.1 as an example.

By the way it is constructed, we have

$$\mathcal{P}_{opt}^{(0)} = \text{no partition},$$
$$\mathcal{P}_{opt}^{(1)} = \{a_1, a_2\},$$
$$\mathcal{P}_{opt}^{(2)} = \{b_1, b_2, a_2\},$$
$$\mathcal{P}_{opt}^{(3)} = \{b_1, b_2, c_1, c_2\}.$$

These four subtrees correspond to the first four subtrees in Figure 2.2. Since they are subtrees of $\mathcal{P}_{opt}^{(3)}$, they are also the elements of set T. Though as a decision node, a_1 is split earlier than a_2 when we constructed the tree, we can choose to drop a_1 first instead of a_2 because they are both decision nodes giving end leaves of the tree. So except for the four elements listed above, the set T contains one more partition, which is $\mathcal{P}_5 = \{a_1, c_1, c_2\}$, and $\mathcal{P}_5 = \{a_1, c_1, c_2\}$ is the fifth subtree in Figure 2.2.

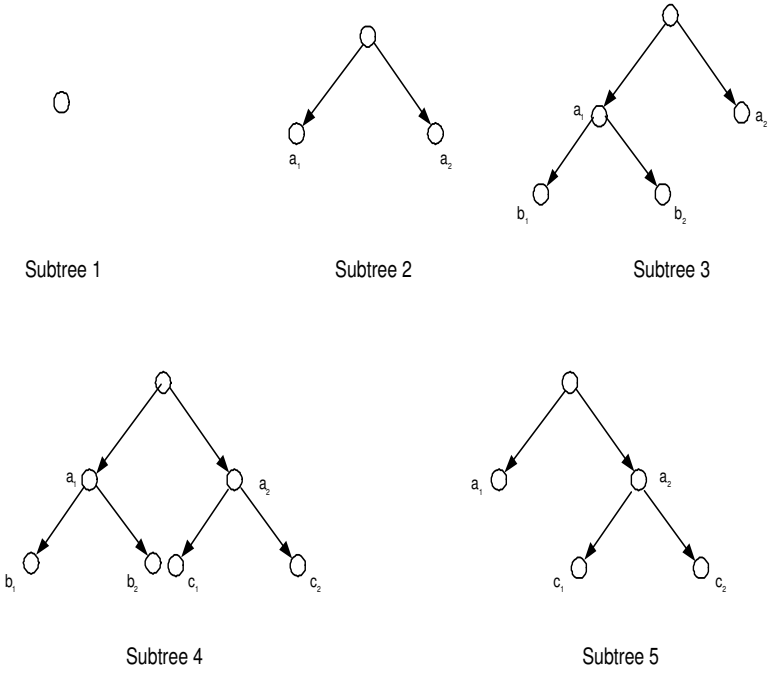


Figure 2.2: The Elements of T For The Simple Binary Tree

For a particular binary tree like that in Figure 2.1, finding all the elements of T is

very easy to do by simply looking at the graph. But in general, finding such elements as $\mathcal{P}_5$ computationally is not so easy. Please see Appendix A: Finding the Set of Subtrees for the strategy we use to do this.

Then after finding all the elements of T, we follow the following three steps to choose our final partition tree:

A. For every $\mathcal{P}_i$, use Double Dogleg method to maximize Equation 2.3 with volatility function 2.2, and get the parameter estimates $(\hat{\phi}^{\mathcal{P}_i}, \hat{\nu}^{\mathcal{P}_i}, \hat{\theta}^{\mathcal{P}_i})$. For starting values, we use the parameter estimates obtained in the previous subsection corresponding to the cells of $\mathcal{P}_i$.

B. For every $\mathcal{P}_i$, calculate the AIC:

$$AIC(\mathcal{P}_i) = -2l(\hat{\phi}^{\mathcal{P}_i}, \hat{\theta}^{\mathcal{P}_i}) + 2\{\dim(\hat{\theta}^{\mathcal{P}_i}) + 2\}. \tag{2.8}$$

The AIC penalizes the number of parameters, or nodes, to be estimated. Here notice that $\dim(\hat{\theta}^{\mathcal{P}_i})$ is the dimension of $\hat{\theta}^{\mathcal{P}_i}$, which equals the number of cells in $\theta^{\mathcal{P}_i}$ times 3, then plus the number of nodes in the tree, and the number "2" accounts for ϕ and ν.

C. Choose the binary tree, or the partition $\mathcal{P}_i$, which minimizes Equation 2.8 and call it $\mathcal{P}$.

The tree-structured GARCH(1,1) given by $\mathcal{P}$ is the final model for our first-stage modelling.

Note that at the end of the previous subsection, we develop a tree with four nodes and five cells/leaves. But after the reducing process, the final tree $\mathcal{P}$ may have any number of nodes from zero to four, and any number of leaves from one to five accordingly.

2.3 The Second-stage Model: GPD with Covariate σ_t^2

In the second-stage model, the dependent variable is no longer r_t. Instead, it is the residual of the first-stage model. Let's denote the residuals by z_t. That is

$$z_t = (r_t - \hat{\phi} r_{t-1})/\hat{\sigma}_t \tag{2.9}$$

where $\hat{\phi}$ and $\hat{\sigma}_t$ are the estimates of ϕ and σ_t obtained in the first-stage model. Notice that the residuals z_t are actually estimates of ϵ_t in Equation 2.1.

Hopefully, the tree-structured GARCH(1,1) model can give us good estimates of the volatilities σ_t^2, so that after the data filtration, the residuals we get no longer have the heteroscedastic property. This can be tested simply by drawing the autocorrelation graph of z_t and that of z_t^2. If both graphs show no significant autocorrelation, then we can say that the residuals do not have heteroscedasticity and thus are now at least plausibly independent.

Now treat z_t as they are independent. By the nature of financial return series, they are still supposed to be fatter-tailed. These properties indicate that the EVT-

based models apply. Here we use the model defined by Equation 1.11, the GPD, to model the exceedances of a high threshold of z_t. As stated in Chapter 2.1, to use a smooth model, this time we do not fit multiple GPD, but fit only one GPD model for all the exceedances, not considering which cell of the partition, or which classical GARCH(1,1) model they correspond to in the first-stage model. Also, to make the model changeable with different volatility levels, we'll use the estimated σ_t^2 obtained in the first stage as an explanatory variable for the shape parameter of the GPD.

Suppose u is a high threshold of z_t; for the exceedances of this threshold, $y_t = z_t - u$, the model is re-stated as the following:

$$\begin{aligned} G(y_t) &= 1 - (1 + \xi(\sigma_t) y_t/\gamma)^{-1/\xi(\sigma_t)} \\ \xi(\sigma_t) &= a + b \ln(\sigma_t^2) \end{aligned} \tag{2.10}$$

where $\gamma > 0$, $\xi(\sigma_t) \geq 0$ and the support is $y_t \geq 0$.

In practice, instead of fixing the threshold u, we take the highest 10% of z_t and treat them as the tail part that need to be modelled by Equation 2.10. We will also make sure that the data set of r_t is large enough to give sufficient number of y_t to fit the model. Again, the parameters will be estimated by MLE. As shown by Smith (1987) [21], in the case of a simple GPD model, the MLE estimates of the two parameters are consistent and asymptotically normal as the number of exceedances used goes to infinity.

This is the end of our two-stage modelling. And the last remained task is to forecast and estimate VaR and the Expected Shortfall.

2.4 Estimating VaR and The Expected Shortfall

Now after the two-stage model is set up, it finally comes to the point to estimate VaR and the Expected Shortfall. First, let's define the notations for these two statistics. For $0 < p < 1$, an unconditional quantile is denoted by

$$R_p = \inf \{r \in R : F_r(r) \geq p\}, \tag{2.11}$$

and a conditional quantile is denoted by

$$R_p^t(k) = \inf \left\{r \in R : F_{r_{t+1}+\ldots+r_{t+k}|\mathcal{G}_t}(r) \geq p\right\}, \tag{2.12}$$

where k means k steps ahead conditional on the information at time t, which is denoted by $\mathcal{G}_t$. If we define r_t to be the negative log returns rather than the log returns, then the VaR defined in Equation 1.1 is $R_p^t(k)$ for some high value p. Here we are only interested in the 1-step ahead prediction $R_p^t(1)$, and let's just call it R_p^t. Similarly, the unconditional Expected Shortfall is defined to be

$$S_p = E[r|r > R_p], \tag{2.13}$$

and the conditional Expected Shortfall is

$$S_p^t(k) = E\left[\sum_{j=1}^{k} r_{t+j} | \sum_{j=1}^{k} r_{t+j} > R_p^t(k), \mathcal{G}_t\right]. \tag{2.14}$$

Again, we are only interested in S_p^t, the 1-step ahead conditional Expected Shortfall. Please notice that our goal is to estimate the conditional VaR and the Expected Shortfall rather than the unconditional ones.

According to the first equation given in 2.1, it is easy to get:

$$\begin{aligned} F_{r_{t+1}|\mathcal{G}_t}(r) &= \Pr\left\{\sigma_{t+1}\epsilon_{t+1} + \mu_{t+1} \leq r|\mathcal{G}_t\right\} \\ &= F_\epsilon\left(\left(r - \mu_{t+1}\right)/\sigma_{t+1}\right). \end{aligned}$$

This leads to the following relation between R_p^t, S_p^t and the quantile of ϵ, which is denoted by z_p:

$$\begin{aligned} R_p^t &= \mu_{t+1} + \sigma_{t+1} z_p \\ S_p^t &= \mu_{t+1} + \sigma_{t+1} E\left[\epsilon|\epsilon > z_p\right] \end{aligned} \tag{2.15}$$

By setting up the two-stage model, we can obtain estimates of μ_{t+1}, σ_{t+1}, z_p, and $E[\epsilon|\epsilon > z_p]$. Plugging them in 2.15, we have the estimates of the wanted VaR and the Expected Shortfall.

It is clear that after fitting the first-stage model, we can calculate $\hat{\mu}_{t+1}$ and $\hat{\sigma}_{t+1}$ immediately. But it is not so easy to obtain estimates of z_p and $E[\epsilon|\epsilon > z_p]$. This is because in stage two, we only fit the GPD to the highest 10% of z_t, where z_t is the residual calculted from the stage-one model.

To estimate z_p, recall that in Chapter 1 we mentioned the following equation while introducing GPD:

$$F_u(y) = \Pr\{X \leq u + y|X > u\} = \frac{F(u+y) - F(u)}{1 - F(u)}, \qquad y > 0, \tag{2.16}$$

where $F_u(.)$ is the cumulated distribution function (CDF) of exceedances $y = z - u$ over threshold u and $F(.)$ is the CDF of X. In our case $F_u(.)$ is the GPD in Equation 2.10 and $F(.)$ is the CDF of z. From the above equation, we have

$$1 - F(z) = \left(1 - F(u)\right)\left(1 - F_u(z-u)\right). \tag{2.17}$$

The second part on the right-hand side can be estimated by the fitted GPD, and the first part is just 10%, because we only choose the highest 10% of z_t to fit the GPD model. Specifically, if we use m z_ts to get the GPD model, where $m \ll n$ and n is the total number of data points we have, and if $z_{(1)}, z_{(2)}, \ldots, z_{(n)}$ are the ordered residuals such that $z_{(1)} \geq z_{(2)} \geq z_{(3)} \ldots \geq z_{(n)}$, then

$$\hat{F}(z) = 1 - \frac{m}{n}\left(1 + \hat{\xi}\frac{z - z_{(m+1)}}{\hat{\gamma}}\right)^{-1/\hat{\xi}}. \tag{2.18}$$

By inverting this formula, for time $t+1$ we can write $\hat{z}_p$ as

$$\hat{z}_p = z_{(m+1)} + \frac{\hat{\gamma}}{\hat{\xi}_{t+1}}\left(\left(\frac{1-p}{m/n}\right)^{-\hat{\xi}_{t+1}} - 1\right). \tag{2.19}$$

Having this result in hand, by Equation 2.15, we calculate $\hat{R}_p^t$ by

$$\hat{R}_p^t = \hat{\mu}_{t+1} + \hat{\sigma}_{t+1}\hat{z}_p. \tag{2.20}$$

Now we need to estimate $E[\epsilon|\epsilon > z_p]$. First notice that what we have in the second-stage model, in the simple form, is $\epsilon - u|\epsilon > u \sim GPD(\xi, \gamma)$. For $z_p > u$, we can write

$$\mathcal{L}\left(\epsilon - z_p|\epsilon > z_p\right) = \mathcal{L}\left((\epsilon - u) - (z_p - u)|(\epsilon - u) > (z_p - u)\right). \tag{2.21}$$

Based on this expression, it can be easily shown that

$$\epsilon - z_p|\epsilon > z_p \sim GPD\left(\xi, \gamma + \xi(z_p - u)\right). \tag{2.22}$$

Then it follows from Equation 2.22 that

$$E\left[\epsilon|\epsilon > z_p\right] = z_p\left(\frac{1}{1-\xi} + \frac{\gamma - \xi u}{(1-\xi)z_p}\right). \tag{2.23}$$

Plugging it in Equation 2.15, for time $t+1$, we finally get

$$\hat{S}_p^t = \hat{\mu}_{t+1} + \hat{\sigma}_{t+1}\hat{z}_p \left(\frac{1}{1-\hat{\xi}_{t+1}} + \frac{\hat{\gamma} - \hat{\xi}_{t+1} z_{(m+1)}}{(1-\hat{\xi}_{t+1})\hat{z}_p} \right). \tag{2.24}$$

Chapter 3

DATA APPLICATION

In this chapter, we apply the two-stage model to some simulated data and some real data respectively. In both cases, we compare the results with those of several other popular models to see if the two-stage model performs better.

3.1 The Simulations

3.1.1 Simulation Structure

To perform simulations, first we need to choose data generators. Here we use three models with the following common form, which is very similar to Equation 2.1, to generate data:

$$\begin{aligned} r_t &= \mu_t + \sigma_t \epsilon_t \\ \mu_t &= \phi r_{t-1} \\ \sigma_t^2 &= f\{r_{t-1}, \sigma_{t-1}^2\} \end{aligned} \tag{3.1}$$

The three models are defined by different forms of $f(r, \sigma^2)$: The first model is a classical GARCH(1,1), that is, Equation 3.1 with

$$f(r, \sigma^2) = 0.05 + 0.1r^2 + 0.85\sigma^2. \tag{3.2}$$

The second model is a tree-structured GARCH(1,1) which has

$$f(r, \sigma^2) = \begin{cases} 0.1 + 0.5r^2 & \text{if } r \leq d_1 = 0, \\ 0.2 + 0.2r^2 + 0.75\sigma^2 & \text{if } r > d_1 = 0 \text{ and } \sigma^2 \leq d_2 = 0.5, \\ 0.8 + 0.5\sigma^2 & \text{if } r > d_1 = 0 \text{ and } \sigma^2 > d_2 = 0.5. \end{cases} \tag{3.3}$$

The third model is neither classical GARCH nor tree-structured GARCH. It is Equation 3.1 with $f(r, \sigma^2)$ defined to be

$$f(r, \sigma^2) = \left(0.1 + 0.2|r| + 0.9r^2\right) \left\{0.8 \exp\left(-1.5|r| \times |\sigma|\right)\right\} + \left(0.4r^2 + 0.5\sigma^2\right)^{3/4}. \tag{3.4}$$

In all the three models, we let $\phi = 0$ for convenience. These three models are the same as those used in Audrino and Buhlmann (2001)[4], who choose these parameters to mimic the time series of real log returns.

We use these three models to construct the following five data-generators:

Generator 1: model 3.2 with $\sqrt{(3/1)}\epsilon_t \sim t_3$;

Generator 2: model 3.3 with $\sqrt{(3/1)}\epsilon_t \sim t_3$;

Generator 3: model 3.4 with $\sqrt{(3/1)}\epsilon_t \sim t_3$;

Generator 4: model 3.2 with $\epsilon_t \sim N(0, 1)$;

Generator 5: model 3.3 with $\epsilon_t \sim N(0, 1)$.

Here we use 3 as the degree of freedom of the t distributions to obtain substantially fatter tails.

Using each of the five data-generators, we generate $N = 120$ samples with sample size $n = 1000$. Notice here that $n = 1000$ corresponds to about four year's of real daily data. Then we use the two-stage model proposed in the previous chapter to obtain the estimates of volatility (σ_t^2), VaR (R_p^t), and the Expected Shortfall (S_p^t). Also notice that for each sample, we have 1000 $\hat{\sigma}_t^2$, but have only one $\hat{R}_p^t$ and one $\hat{S}_p^t$ for a particular value of p.

For comparison purposes, we also fit four other models mentioned in Chapter 1 for the same generated data, and obtain estimates for the three measurements. These four models are:

1. RiskMetrics, i.e., the IGARCH(1,1) model defined by Equation 1.6[1];

2. classical GARCH(1,1) model defined by Equation 1.5;

3. threshold GPD fitted to the highest 10% data;

4. the MF model: the two-stage model proposed in McNeil and Frey (2000)[18].

Here notice that R_p^t and S_p^t can be calculated for all the four models. But the estimates of volatility, the $\hat{\sigma}_t^2$s, can't be obtained for the threshold GPD.

[1]Actually the model we use here is not exactly Equation 1.6. Instead, we use the IGARCH(1,1) model defined in SAS, which has an extra intercept term ω in the volatility Equation of 1.6. Notice that when $\omega = 0$, it is exactly Equation 1.6.

To compare the performance of our two-stage model with those of the four models mentioned above, and also to quantify the goodness of fit, we need to calculate some statistics. For simulated data, a big advantage we can take is that the true values of volatility (σ_t^2), VaR (R_p^t), and the Expected Shortfall (S_p^t) are all known. Keeping this in mind, we define the following statistics:

1. the AISRL (*Average In-Sample Relative Loss*) of $\hat{\sigma}_t^2$: $AISRL_{\hat{\sigma}_t^2} = \frac{1}{Nn}\sum_{k=1}^{N}\sum_{t=1}^{n}\frac{|\sigma_{t,k}^2-\hat{\sigma}_{t,k}^2|}{\sigma_{t,k}^2}$;
2. the sample mean of $\hat{R}_p^t$;
3. the sample standard deviation of $\hat{R}_p^t$;
4. the ARL (*Average Relative Loss*) of $\hat{R}_p^t$: $ARL_{\hat{R}_p^t} = \frac{1}{N}\sum_{k=1}^{N}\frac{|R_p^{t,k}-\hat{R}_p^{t,k}|}{R_p^{t,k}}$;
5. the sample mean of $\hat{S}_p^t$;
6. the sample standard deviation of $\hat{S}_p^t$;
7. the ARL (*Average Relative Loss*) of $\hat{S}_p^t$: $ARL_{\hat{S}_p^t} = \frac{1}{N}\sum_{k=1}^{N}\frac{|S_p^{t,k}-\hat{S}_p^{t,k}|}{S_p^{t,k}}$;

The first statistic is used to measure the performance of the first-stage model, i.e., the tree-structured GARCH(1,1), in estimating the volatility. Notice again that it can not be calculated for the threshold GPD, which does not estimate the volatility. The next three statistics are used to measure the models' ability to estimate VaR. And the last three are for the Expected Shortfall. Obviously, the smaller, the better are the relative loss statistics.

In addition to the statistics listed above, we also check how many nodes our two-stage mode find for each of the 1000 data sets simulated from the five data generators.

The true models defined by Equation 3.2 to Equation 3.4 show that Generator 1 and 4 have no node, whereas Generator 2 and 5 have two nodes. So we access whether, for each of the 1000 data sets generated, our model can tell the true number of nodes most of the time.

One more thing needs to be pointed out: In order to calculate the estimates for the four competing models, and also to obtain the real values of VaR and the Expected Shortfall, some formulas other than those mentioned in Chapter 2.4 should be used. These formulas are listed in Appendix B.

3.1.2 Simulation Results

In this subsection, we present the results relating to the first-stage model first. This includes the effect of the data filtration for heteroscedasticity, the number of nodes estimated by our tree-structured GARCH(1,1) model, and the AISRL statistics. Then we summarize the statistics about the estimation of VaR and the Expected Shortfall. And some conclusions are drawn at the end.

Results Related to the First-stage Model

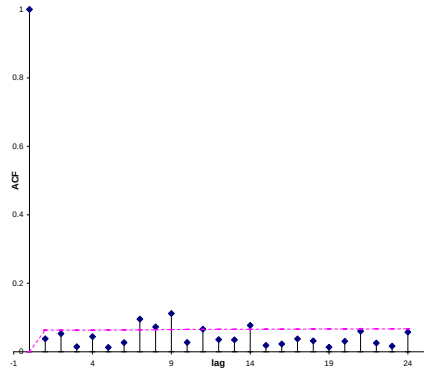
First, we would like to check whether the two-stage model succeeded in removing the heteroscedasticity by using the tree-structured GARCH(1,1) as the data filtration.

To do this, we calculate the autocorrelations of series r_t, r_t^2, e_t and e_t^2 respectively. Remember that e_t are the estimates of the true innovation series, ϵ_t. The ideal situation is that r_t has almost no significant autocorrelations, but r_t^2 has some,

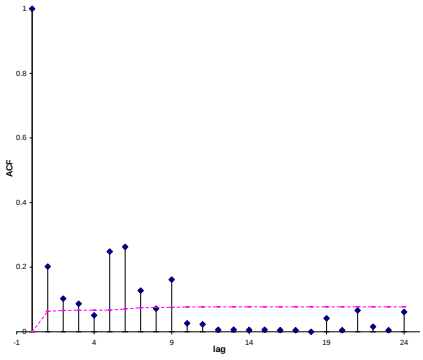
showing that the generated series r_t is heteroscedastic. Meanwhile, both e_t and e_t^2 have no significant autocorrelations. This means that we successfully remove the heteroscedasticity from the the generated data by using the tree-structured GARCH(1,1) filtration. In addition, we also get the QQ plots against $N(0,1)$ for both r_t and e_t to see if the data is fatter tailed, and if this property remains after the filtration. Obviously, we expect it does for the generators with t innovations.

The following three figures are produced respectively for the first data sets generated from Generator 1, Generator 2, and Generator 3. In each of the figures, the top four panels (panel (a), (b), (c) and (d)) are plots of the absolute value of autocorrelations for r_t, r_t^2, e_t and e_t^2. The two panels on the bottom (panel (e) and (f)) are QQ plots for r_t and e_t. For the autocorrelation plots, we use lag = 24, because in reality, there are usually 22 to 24 trading days within a month, though this has little meaning for simulated data. Also, we plot the two times' standard errors of the autocorrelations at the same time. Thus a prominent spike means a significant autocorrelation value of the corresponding lag.

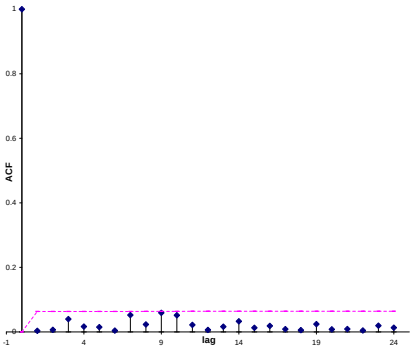
All the QQ plots in Figure 3.1, 3.2 and 3.3 show that both r_t and e_t have fatter tails than a Normal distribution. So the generated data are fatter-tailed and the use of the first-stage model does not change this property. This also indicates that the use of GPD in the second stage is appropriate. It's also noticed that in Figure 3.2 panel (e), the QQ plot for r_t is not a straight line. Actually this is true for almost all the data sets generated from Generator 2, indicating that the model used in generator 2 is far from a Normal distribution.



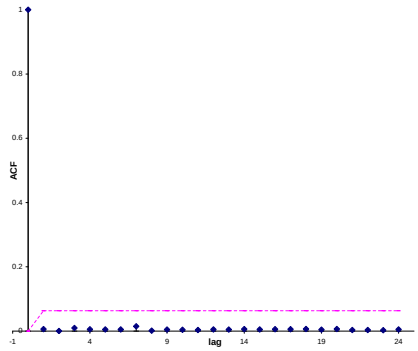
(a) Series r_t



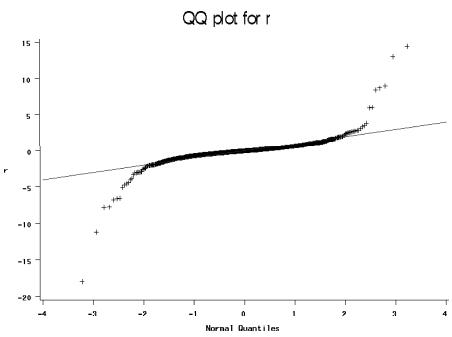
(b) Series r_t^2



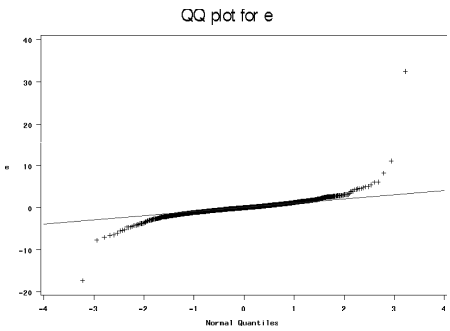
(c) Series e_t



(d) Series e_t^2

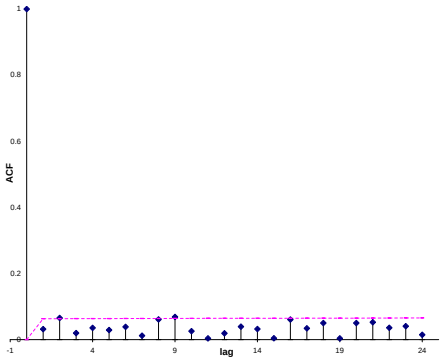


(e) QQ Plot for r_t

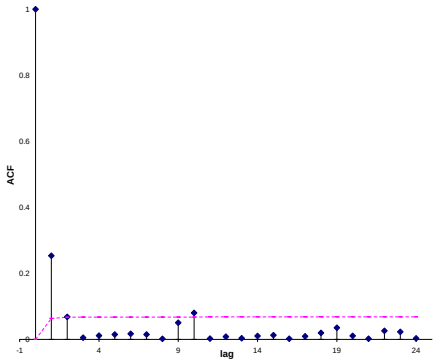


(f) QQ Plot for e_t

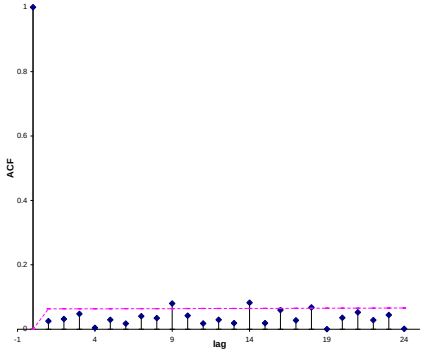
Figure 3.1: ACF and QQ Plots: Generator 1



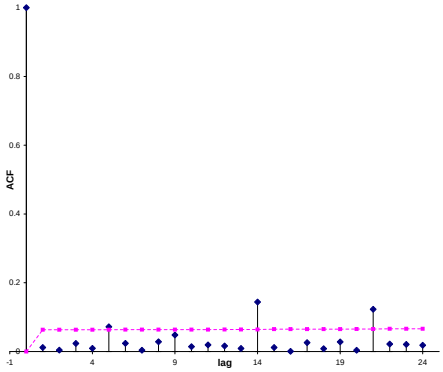
(a) Series r_t



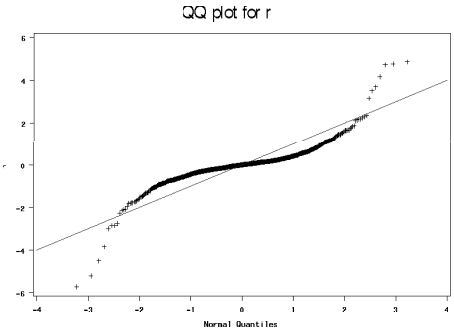
(b) Series r_t^2



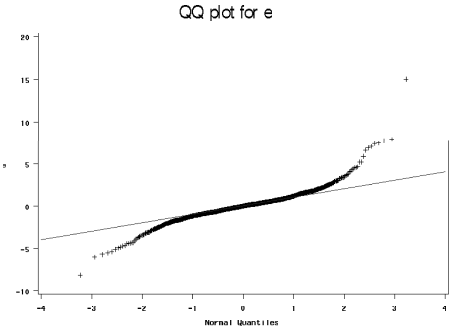
(c) Series e_t



(d) Series e_t^2

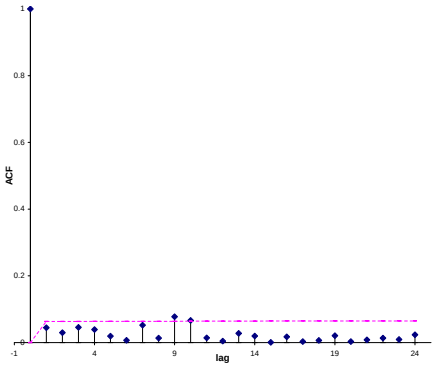


(e) QQ Plot for r_t

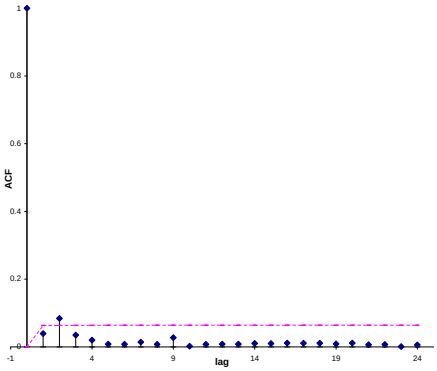


(f) QQ Plot for e_t

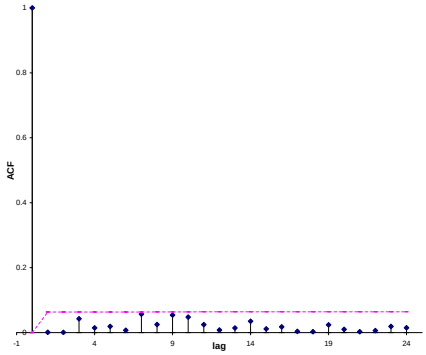
Figure 3.2: ACF and QQ Plots: Generator 2



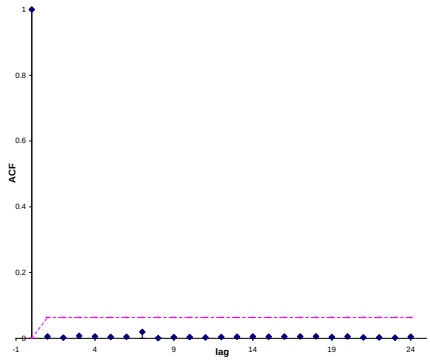
(a) Series r_t



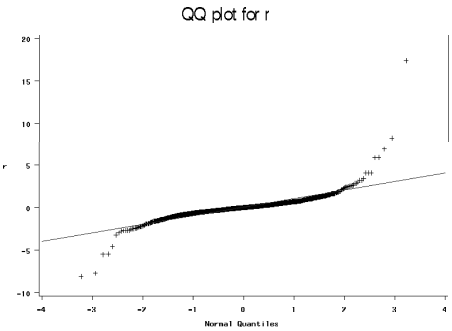
(b) Series r_t^2



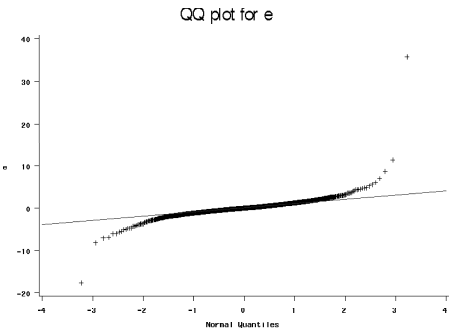
(c) Series e_t



(d) Series e_t^2



(e) QQ Plot for r_t



(f) QQ Plot for e_t

Figure 3.3: ACF and QQ Plots: Generator 3

As to the heteroscedasticity, we have the following observations: Panel (a) in all the three figures shows no significant autocorrelations except for lag $= 7, 8$ and 9 in Figure 3.1. Panel (b) has significant spikes in all the three figures, and this is especially obvious in Figure 3.1 and Figure 3.2, which are for data sets generated from GARCH models. Panel (c) shows e_t are not autocorrelated for all the three data sets. And panel (d) shows e_t^2 are not autocorrelated except for lag $= 14$ and 21 in Figure 3.2. It is also observed, though not from the above figures, the lag values we just mentioned do not always have significant autocorrelations in other data sets generated by the same data generating models. Note that 5% of the absolute ACF values should lie beyond the bounds because we use two times' standard errors as the limits. So the existence of the significant autocorrelations mentioned above is reasonable. We can conclude that r_t, e_t and e_t^2 do not have low-order autocorrelations, but r_t^2 has.

Though we present the ACF and QQ plots for only the first generated data sets of each of the three data generators, we need to point out that the majority of the other data sets have similar data characteristics as these three. This gives the fundamental reason to use our two-stage model. Meanwhile, there are a small proportion of data sets which do not have heteroscedasticity and/or a fatter tail, though they are supposed to have according to the models used to generate them.[2] We should say that it is something we must deal with while doing a simulation: extreme cases do occur. And we do not exclude them from our simulation because of this reason. That is, we

[2]The proportion is relatively larger for Generator 4 and 5, which use standard Normal innovations rather than t innovations. But in general, it is a small percentage.

still apply the two-stage model and all other models to them, though these may not be the appropriate models to these particular data sets.

Secondly, we explore how well our first-stage model performs in finding the nodes and estimating the volatilities. Table 3.1[3] summarizes the number of nodes found by our two-stage model for different data generating models:

Data Generator		0 node	1 node	2 nodes	3 nodes	4 nodes	Total
1	frequency	71	24	11	10	4	120
	percentage	59.2%	20.0%	9.2%	8.3%	3.3%	100%
2	frequency	0	23	32	33	32	120
	percentage	0.0%	19.2%	26.7%	27.5%	26.7%	100%
3	frequency	45	34	23	11	7	120
	percentage	37.5%	28.3%	19.2%	9.2%	5.8%	100%
4	frequency	110	7	2	1	0	120
	percentage	91.7%	5.8%	1.7%	0.8%	0.0%	100%
5	frequency	0	6	33	38	43	120
	percentage	0.0%	5.0%	27.5%	31.7%	35.8%	100%

Table 3.1: Frequency of Nodes: Simulated Data

For data Generator 1 and Generator 4, which are the simple GARCH(1,1) and have no node, our two-stage model gives quite satisfactory result: For Generator 1, for 71 (59.2%) out of the 120 data sets, it catches the true model. For another 20.0% of the data sets, it results in only one node. For Generator 4, it is even better. The two percentages are 91.7% and 5.8% respectively. This is acceptable because we use

[3]Note: The percentages for Data Generator 2 in this table do not sum up to exactly 100.0% due to rounding problems.

the AIC criteria in choosing the final tree structure, and it is commonly known that AIC tends to overfit the model but give more accurate forecasts. Generator 2 and Generator 5 have two nodes, $d_1 = 0$ in r and $d_2 = 0.5$ in σ^2. For the 120 data sets generated from Generator 2, the two-stage model results in almost the same number of trees which have two nodes, three nodes, or four nodes. For Generator 5, it results in even more data sets with three or four nodes. For these two tree-structured GARCH(1,1) generators, our model does not perform as well as it does for Generator 1 and 4, over-estimating the number of nodes most of the time. But the results do make it clear that the data do not come from a simple GARCH(1,1), because none of the resulting trees have zero node. For Generator 3, which is neither a simple GARCH(1,1) nor a tree-structured GARCH(1,1), the two-stage model tends to mimic it by a tree with zero or one node.

Table 3.2[4] lists the AISRL (the average in-sample relative loss) statistics we calculated for the five data generators:

Here notice that the AISRL statistics can not be calculated for Model 3, the threshold GPD, because Model 3 does not estimate the volatilities. Also, this statistic is the same for Model 2 and Model 4. This is because Model 2 is the classical GARCH(1,1), which is exactly the first-stage model for Model 4, the McNeil and Frey model. The volatilities are estimated only in the first-stage model.

Table 3.2 shows that for data generated from a simple GARCH(1,1) or the non-

[4]In this table we use TS to stand for our two-stage model. This abbreviation will also be used in other tables and the context of this chapter.

	Model 1 RiskMetrics	Model 2 and 4 GARCH(1,1) and MF	TS Model Two-stage
Generator 1	0.6394	0.4806	0.6240
Generator 2	1.8120	1.5173	0.6186
Generator 3	0.5648	0.4018	0.6190
Generator 4	0.1094	0.0464	0.3485
Generator 5	0.9615	0.8466	0.1963

Table 3.2: AISRL Statistics

GARCH model, our two-stage model is not the best one in estimating the volatilities. It has average biases of 62% or so for Generator 1 and 3, and 34.85% for Generator 4. The best model is the classical GARCH(1,1), which has an average relative loss of 48.06%, 40.18%, and 4.64% respectively. But if the data come from a tree-structured GARCH(1,1), our model is considerably better than the other three models: For Generator 2, it has an average relative loss of 62% or so, whereas the losses for other three models are above 151%. For Generator 5 its AISRL is 19.63% compared with 96.15% and 84.66% for others. Putting this into consideration, we conclude that our model is not much worse than the others when the data are really from GARCH(1,1), but much better when the true data generating model is a tree-GARCH.

Statistics Related to VaR and ES estimates

Finally we present the statistics about the estimateds of VaR and the Expected Shortfall. Table 3.3 to 3.7 summarize the real and estimated means and standard errors for each of the five data generators. ES in the column titles means the Expected

Shortfall. The bold numbers in the tables are the best among the five models.

Model		VaR $p = 0.95$	VaR $p = 0.99$	VaR $p = 0.995$	ES $p = 0.95$	ES $p = 0.99$	ES $p = 0.995$
True	mean	1.0923	2.1075	2.7110	1.7982	3.2504	4.1367
	std	0.2233	0.4308	0.5542	0.3676	0.6645	0.8456
TS	mean	**1.0879**	**2.1087**	**2.6976**	**1.7884**	3.2033	4.0463
	std	**0.2667**	0.5537	0.7728	0.4873	1.1345	1.6828
Model 1	mean	1.3831	1.9562	2.1660	1.7345	2.2411	2.4318
	std	0.4208	0.5951	0.6589	0.5276	**0.6818**	0.7398
Model 2	mean	1.3244	1.8731	2.0739	1.6608	2.1459	2.3284
	std	0.3456	0.4888	**0.5412**	0.4334	0.5600	**0.6076**
Model 3	mean	1.2108	2.5020	3.2777	2.1142	3.9972	5.1626
	std	0.1483	**0.4698**	0.7485	**0.4184**	1.2344	1.9292
Model 4	mean	1.1303	2.1973	2.7993	1.8523	**3.2804**	**4.1056**
	std	0.2899	0.5902	0.7880	0.4966	1.0150	1.4036

Table 3.3: Mean and Std for VaR and ES Estimates: Generator 1

For data Generator 1, for which the results are shown in Table 3.3, except for ES ($p = 0.99$) and ES ($p = 0.995$), the means of our TS model are the ones which are closest to those of the true model. And for the two exceptions, Model 4, which also has a two-stage structure, is the best. But both of these models are not good at the standard errors. Their estimates do not have better stds than the true values and the other models.

From Table 3.4, we can see that for data simulated from Generator 2, our two-stage model again has the most best means, but this time the exceptions are VaR ($p = 0.99$) and ES ($p = 0.95$). This is because both the true model and our TS model

Model		VaR $p = 0.95$	VaR $p = 0.99$	VaR $p = 0.995$	ES $p = 0.95$	ES $p = 0.99$	ES $p = 0.995$
True	mean	0.8476	1.6354	2.1037	1.3956	2.5223	3.2100
	std	0.2233	0.4308	0.5542	0.3676	0.6645	0.8456
TS	mean	**0.8831**	1.7325	**2.2372**	1.4817	**2.7310**	**3.5149**
	std	0.4570	0.9445	1.3155	0.8607	2.0843	3.1821
Model 1	mean	1.1840	**1.6746**	1.8541	1.4848	1.9185	2.0817
	std	0.4461	0.6309	0.6986	0.5594	**0.7228**	**0.7843**
Model 2	mean	1.1212	1.5858	1.7558	**1.4061**	1.8168	1.9713
	std	0.2536	0.3587	0.3972	0.3181	0.4109	0.4459
Model 3	mean	1.0264	2.2141	2.9189	1.8516	3.5534	4.5927
	std	0.0927	0.2734	0.4649	0.2442	0.8187	1.3265
Model 4	mean	0.9468	1.9737	2.5637	1.6488	3.0595	3.8932
	std	**0.1945**	**0.4219**	**0.5942**	**0.3583**	0.8451	1.2495

Table 3.4: Mean and Std for VaR and ES Estimates: Generator 2

assume a tree-structure with a t innovation term. So our TS model should be the best one. Model 4 has most best stds in this table. Again, our TS model has larger values than the stds should be.

Table 3.5 shows that for Generator 3, Model 4 is the best, which has five out of the six best means. Our TS model only has the one for ES ($p = 0.995$). But by taking a careful look at the first five columns of the table, we can see that the means of the two-stage model are always the second closest to those of the true model. The differences between TS model's means and Model 4's are not very much.

Generally, from Table 3.3, Table 3.4 and Table 3.5, we conclude that for data generated from GARCH-type models with t innovations, i.e., Generated 1 and 2, the

Model		VaR $p = 0.95$	VaR $p = 0.99$	VaR $p = 0.995$	ES $p = 0.95$	ES $p = 0.99$	ES $p = 0.995$
True	mean	1.1851	2.2866	2.9413	1.9510	3.5265	4.4881
	std	0.2448	0.4724	0.6077	0.4031	0.7286	0.9272
TS	mean	1.1547	2.2109	2.8169	1.8914	3.3964	**4.3231**
	std	**0.3267**	0.6449	0.9025	0.6230	1.7450	2.8917
Model 1	mean	1.5041	2.1273	2.3555	1.8663	2.4372	2.6446
	std	0.6888	0.9742	1.0786	0.8638	1.1161	1.2110
Model 2	mean	1.4474	2.0471	2.2667	1.8151	2.3453	2.5448
	std	0.6394	0.9043	1.0013	0.8018	1.0360	**1.1242**
Model 3	mean	1.2705	2.5507	3.3109	2.1604	3.9977	5.1210
	std	0.0933	0.2837	**0.4760**	0.2526	**0.8627**	1.4197
Model 4	mean	**1.2082**	**2.3241**	**2.9490**	**1.9614**	**3.4433**	4.2946
	std	0.3350	**0.6350**	0.8072	**0.5301**	0.9948	1.3468

Table 3.5: Mean and Std for VaR and ES Estimates: Generator 3

means of our TS estimates for VaR and ES are generally better than all of the other three models. But for data from Generator 3, which is not a GARCH-type model, our model performs a little worse than Model 4. Here we would also like to point out another observation about the means of estimates: It seems from all the three tables that when it comes to ES ($p = 0.99$) and ES ($p = 0.995$), the true means of ES tend to increase very fast. And all the models which incorporate the GPD (TS, Model 3 and Model 4) can reflect this increase by an increasing mean of estimates, but the other two models can't.

It's also observed from these tables that though our TS model has good means of the estimates, the stds of the estimates are poorer in most of the cases. This is

especially true for ES ($p = 0.99$) and ES ($p = 0.995$). For these two cases our method gives estimates with considerably larger standard errors.

Generator 4 and Generator 5 differ from the first three data generators in that they use $N(0, 1)$ as the distribution of the innovation. Table 3.6 and Table 3.7 give the means and stds of these two generators:

Model		VaR $p = 0.95$	VaR $p = 0.99$	VaR $p = 0.995$	ES $p = 0.95$	ES $p = 0.99$	ES $p = 0.995$
True	mean	1.5750	2.2276	2.4665	1.9752	2.5521	2.7692
	std	0.3298	0.4665	0.5165	0.4136	0.5344	0.5799
TS	mean	1.5482	2.2859	2.6046	2.0068	2.7466	3.0662
	std	0.3476	0.5132	0.5856	0.4498	0.6175	0.6902
Model 1	mean	1.6317	2.3077	2.5552	2.0462	2.6439	2.8688
	std	0.4472	0.6325	0.7003	0.5608	0.7246	0.7863
Model 2	mean	**1.5799**	**2.2344**	**2.4741**	**1.9812**	**2.5599**	**2.7777**
	std	**0.3442**	**0.4868**	**0.5390**	**0.4317**	**0.5577**	**0.6052**
Model 3	mean	1.6053	2.4533	2.8258	2.1343	2.9979	3.3779
	std	0.1159	0.2282	0.2935	0.1848	0.3365	0.4215
Model 4	mean	1.5476	2.2827	2.5994	2.0044	2.7397	3.0564
	std	0.3448	0.5081	0.5798	0.4455	0.6117	0.6840

Table 3.6: Mean and Std for VaR and ES Estimates: Generator 4

For Generator 4, the true model is classical GARCH(1,1) with $N(0, 1)$ innovations. And Table 3.6 shows Model 2 has the best means and stds, because it uses exactly the true model and the true innovation distribution. Also, Model 4 is the second best one for this case. And our TS model is a little worse than Model 4.

For Generator 5, which is tree-structured GARCH(1,1) with $N(0, 1)$ innovations,

Model		VaR $p = 0.95$	VaR $p = 0.99$	VaR $p = 0.995$	ES $p = 0.95$	ES $p = 0.99$	ES $p = 0.995$
True	mean	1.2492	1.7668	1.9563	1.5666	2.0241	2.1964
	std	0.5548	0.7847	0.8688	0.6957	0.8990	0.9754
TS	mean	**1.2689**	**1.8823**	2.1488	**1.6506**	2.2687	2.5372
	std	0.5891	0.8758	1.0014	0.7673	1.0584	1.1858
Model 1	mean	1.4602	2.0652	2.2866	1.8311	2.3660	2.5673
	std	**0.5671**	**0.8020**	**0.8880**	**0.7111**	**0.9188**	**0.9970**
Model 2	mean	1.3390	1.8937	**2.0968**	1.6791	**2.1696**	**2.3541**
	std	0.3582	0.5066	0.5609	0.4492	0.5804	0.6298
Model 3	mean	1.4116	2.3945	2.8282	2.0254	3.0307	3.4755
	std	0.0902	0.1840	0.2368	0.1455	0.2716	0.3452
Model 4	mean	1.3184	2.1882	2.5723	1.8616	2.7517	3.1457
	std	0.3632	0.6238	0.7404	0.5252	0.7960	0.9190

Table 3.7: Mean and Std for VaR and ES Estimates: Generator 5

Table 3.7 shows for means, the TS model is the best one for small ps and the classical GARCH(1,1) is the best for other ps. The TS model is also the second best for large ps. Model 1 has all the best stds.

The last two tables for VaR and ES estimates are Table 3.8 and Table 3.9, which contain the results of the ARL statistics defined in Chapter 3.1.1:

Because ARL is the average relative loss, it is obvious that the smaller, the better. So except for ES ($p = 0.99$) and ES ($p = 0.995$), the TS model is better than the others for data Generator 1. And for data Generator 2, TS model is the best for both VaR and the ES for all the three p values. For data from Generator 3, Model 4, the MF model, is the best one. The TS model is the second best. For Generator 4, again

Generator	Model	VaR $p = 0.95$	VaR $p = 0.99$	VaR $p = 0.995$	ES $p = 0.95$	ES $p = 0.99$	ES $p = 0.995$
1	TS	0.0774	0.1016	0.1273	0.1046	0.1759	0.2169
	Model 1	0.2582	0.1429	0.2313	0.1306	0.3270	0.4203
	Model 2	0.2244	0.1575	0.2503	0.1366	0.3469	0.4382
	Model 3	0.2208	0.2933	0.3226	0.2860	0.3668	0.4102
	Model 4	0.0995	0.1196	0.1391	0.1174	0.1699	0.2024
2	TS	0.1343	0.1595	0.1860	0.1646	0.2393	0.2876
	Model 1	0.6908	0.3719	0.2945	0.3986	0.2641	0.2982
	Model 2	0.6861	0.3804	0.3183	0.4057	0.3033	0.3240
	Model 3	0.6554	0.8000	0.8349	0.7722	0.8635	0.8953
	Model 4	0.4911	0.5733	0.5850	0.5513	0.5847	0.5915
3	TS	0.0983	0.1133	0.1384	0.1225	0.2059	0.2626
	Model 1	0.2541	0.1691	0.2596	0.1514	0.3506	0.4409
	Model 2	0.2292	0.1899	0.2866	0.1686	0.3756	0.4622
	Model 3	0.1775	0.2170	0.2401	0.2154	0.2741	0.3077
	Model 4	0.0921	0.0958	0.1127	0.0957	0.1456	0.1786

Table 3.8: ARL Statistics: Generators with t Innovations

because it uses exactly the true model and the true innovation distribution, Model 2 is the one with the smallest ARLs. And Model 4 is the second best model, even Model 1 is better than the TS model for VaR ($p = 0.995$), ES ($p = 0.99$) and ES ($p = 0.995$). For Generator 5, the TS model is considerably better because it catches the tree structure, just as it does for Generator 2.

Generator	Model	VaR $p = 0.95$	VaR $p = 0.99$	VaR $p = 0.995$	ES $p = 0.95$	ES $p = 0.99$	ES $p = 0.995$
4	TS	0.0483	0.0564	0.0727	0.0509	0.0871	0.1129
	Model 1	0.0698	0.0698	0.0698	0.0698	0.0698	0.0698
	Model 2	0.0314	0.0314	0.0314	0.0314	0.0314	0.0314
	Model 3	0.1628	0.2096	0.2401	0.1951	0.2615	0.2982
	Model 4	0.0432	0.0506	0.0674	0.0451	0.0817	0.1076
5	TS	0.1104	0.1245	0.1416	0.1195	0.1558	0.1798
	Model 1	0.4268	0.4268	0.4268	0.4267	0.4268	0.4268
	Model 2	0.3873	0.3873	0.3873	0.3873	0.3873	0.3873
	Model 3	0.6073	0.7932	0.8746	0.7393	0.9256	1.0192
	Model 4	0.3836	0.5235	0.5925	0.4797	0.6325	0.6995

Table 3.9: ARL Statistics: Generators with Normal Innovations

Some Conclusions for Simulation

By reviewing all the results listed in the above tables and figures, we draw the following conclusions from the simulations:

- First, our two-stage model does remove the heteroscedasticity of the simulated data by using the tree-structured GARCH(1,1) as the data filtration;

- Second, our TS model shows undoubted advantage in estimating VaR and ES when the true underlying model is GARCH(1,1) and has a tree structure, no matter whether the innovation term is standard Normal or standardized $t(3)$, which has an obvious fatter tail than $N(0, 1)$;

- Third, when the data are simulated from a classical GARCH(1,1) with innova-

tions coming from $N(0,1)$, the existing RiskMetrics, GARCH, and MF methods are better than our TS model. But if the innovation is standardized $t(3)$, our TS model remains the best except for ES $p = 0.99$ and ES $p = 0.995$, for which cases the MF model is the second best choice. And both models are much better than the other three.

3.2 Application to Real Data

3.2.1 The Structure

In this section, we apply our two-stage model and the four competitive models to two real financial series of negative log returns. We choose one index, the NASDAQ, and one stock, the MRK for Merck & Co. , the famous pharmaceutical company. For the NASDAQ index, we use the daily closing NASDAQ 500 composite from January 2, 1990 to November 26, 1997. And for the MRK stock, we use the daily closing price starting from March 2, 1992. Both the NASDAQ index series and the MRK price series have a total of 2001 historical observations.

First, for both of the two financial instruments, we use the historical series to calculate daily negative log returns (in percentages) according to the following equation

$$r_t = -100 \ln \frac{P_t}{P_{t-1}}, \tag{3.5}$$

where P_t is the index or stock price at time point t for $t = 0, 1, 2, \ldots, 2000$. Notice that for a total of 2001 observations, we can get exactly 2000 historical log returns

for either of the two series.

Second, we fit our two-stage model and the four competing models to the log return series. As for the simulated data, we always use a window of $n = 1000$ to do model fitting and one-step forward estimating. We begin with the first 1000 observations of the historical log return series, fit the models and get the estimates of R_p^t and S_p^t for $p \in \{0.95, 0.99, 0.995\}$. Then we drop the first observation and add the 1001th one to form the second data set, and do the model fitting and estimating again. Then repeat this step until we exhaust the whole series. This means we fit each model 1000 times with sample size $n = 1000$ and get estimates of R_p^t and S_p^t for $t \in \{1001, 1002, \ldots, 2000\}$. Since there are 3 values of p, there are a total of 1000×3 estimates of R_p^t and S_p^t respectively.

For real data, we no longer assume $\phi = 0$ in the mean equations of all the five models except the competing Model 3, which does not have a mean equation. Instead, we estimate ϕ in each step as described in Chapter 2.2.

Finally, we again define and calculate some statistics to measure the performance of each model in estimating volatility, VaR and the Expected Shortfall. Then compare the values of these statistics and draw some conclusions about the model performances.

Again, we first give a table for the frequencies of the number of nodes estimated by our two-stage model. And to measure the goodness of fit of σ_t^2, we use the so-called

Average In-Sample Absolute Loss for Prediction (AISALP) defined by

$$AISALP = \frac{1}{N(n-1)} \sum_{k=1}^{N} \sum_{t=2}^{n} \left| \hat{\sigma}_t^2 - (r_t - \hat{\mu}_t)^2 \right|. \tag{3.6}$$

Here $N = 1000$ because of the way we exhaust the series with 2000 observations. And $\hat{\mu}_t = r_t - \hat{\phi} * r_{t-1}$. AISALP is the counterpart of the AISRL statistics we use for simulated data. But we choose to use absolute loss rather than relative loss here. This is because for real data, r_t and $\hat{\mu}_t$ can be so close that their differences can sometimes become very close to zero. If we divide the absolute difference in Equation 3.11 by this small number, the ratio can be extremely large. Thus it does not reflect the goodness-of-fit of $\hat{\sigma}_t^2$ very properly. But for simulated data, we use σ_t^2, which is known and not so close to zero, this problem does not exist.

When it comes to R_p^t and S_p^t, again we calculate the following statistics for the estimates:

1. the sample mean of $\hat{R}_p^t$;
2. the sample standard deviation of $\hat{R}_p^t$;
3. the sample mean of $\hat{S}_p^t$;
4. the sample standard deviation of $\hat{S}_p^t$.

Notice that the ARL statistics used for simulated data are not obtainable because of the fact that the true values for R_p^t and S_p^t are unknown for real data.

Though we no longer have the advantage of knowing the true values of the risk measurements, notice that we actually use $N = 1000$ instead of the $N = 100$ used

in the previous simulations. This makes backtesting applicable. For R_p^t, in addition to the mean and standard deviation of the estimates, for each value of p and the corresponding 1000 estimates, we can count the number of violations. A violation is said to occur whenever $r_{t+1} > \hat{R}_p^t$. By the definition of VaR, the number of violations should have a binomial distribution $B(1000, 1-p)$ under H_0 : *the model correctly estimates VaR* . Thus, we can compare the number of violations with its expectation $1000 \times (1-p)$. Obviously for $p = 0.95, p = 0.99$ and $p = 0.995$, the expected numbers of violations are 50, 10 and 1 respectively. Also, the binomial distribution and the null hypothesis give the following confidence interval for a significant level 0.95:

$$(1-p) - \sqrt{(1-p)p\frac{\chi^2_{0.95}(1)}{1000}} < \frac{n_v}{1000} < (1-p) + \sqrt{(1-p)p\frac{\chi^2_{0.95}(1)}{1000}}, \tag{3.7}$$

where n_v is the number of violations, and $\chi^2_{0.95}(1)$ is the inverse chi-square distribution function with degree of freedom 1.

For S_p^t, unfortunately there is no such beautiful result as the binomial distribution for R_p^t. When the underlying models are different, the distribution properties of S_p^t are also very different. So far it seems there is no good statistic to measure the goodness of fit for S_p^t. The only statistic we found in literature is based on the so-called *exceedance residuals* , which is denoted by r_{t+1}^* and defined by

$$r_{t+1}^* = \frac{r_{t+1} - \hat{S}_p^t}{\hat{\sigma}_{t+1}\sqrt{\hat{var}(\epsilon|\epsilon > z_p)}} \quad \text{for } r_{t+1} > \hat{S}_p^t. \tag{3.8}$$

It is said that the exceedance residuals should behave like an iid sample with mean zero and variance one. But it seems from the literature that the application of this

result is not satisfactory. We'll just list this statistics here without using it.

The last thing we do for real data application is to plot the 1000 estimates of R_p^t and S_p^t for each p and each model. From these plots we observe how fast the estimates change from time t to the adjacent time $t+1$, which can not be fully reflected by the statistics listed above. We hope that they do not change too quickly. This is because one important use of the market risk measurements is to help the investors set their margins. If the margins change dramatically from one day to another, it means the investors need to add or reduce their margins by a considerable amount daily. This is not preferable to the investors, even when the measures are statistically more accurate than the others. We will check our results from this non-statistical aspect.

3.2.2 The Results

In this subsection, we present the results about real data application by a similar way as that for simulated data. That is, first we check the autocorrelation and QQ plots for variable r_t and e_t, to see if our data filtration model, the tree-structured GARCH(1,1) succeeds in remove the heteroscedastic property of the log return series. Second, we give the table of frequencies of the number of nodes got by our two-stage model. Then we list the results of the AISALP statistics, which measures the performance of the first-stage model in estimating the volatilities. Third, we present the results of the means and standard deviations of our estimates. Fourth, we backtest

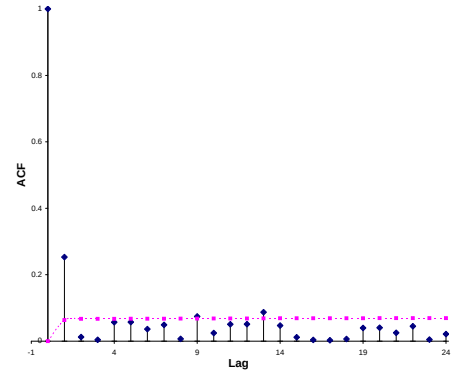
and count the number of violations we have for each model. And finally, we plot all the estimates of VaR and the Expected Shortfall for the NASDAQ index series, to see how volatile they are.

Results Related to the First-stage Model

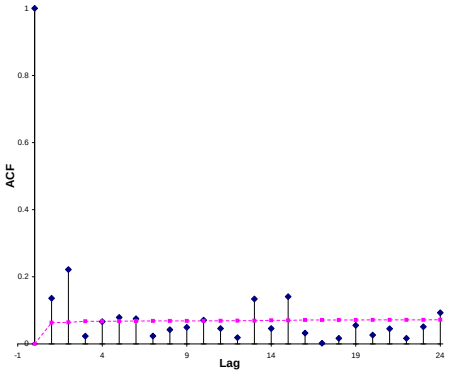
First, Figure 3.4 and Figure 3.5 are the ACF and QQ plots for NASDAQ and MRK respectively. For either of the figures, we use a data set of 1000 observations. We use the same number of lags and the same layout of the plots as before. Remember that a prominent spike means a significant autocorrelation value of the corresponding lag.

The ACF plot for r_t in Figure 3.4 is very different from the counterparts in the figures for simulated data: it has a significant autocorrelation value for lag = 1. This is the reason why we do not assume $\phi = 0$ for real log returns. The spikes in panel (b) of the same figure shows that the NASDAQ log return series is heteroscedastic, as we expect. And panel (c) and (d) show that the data filtration is successful, the estimates of innovations, e_t, no longer has this property. The QQ plot in panel (e) is somewhat similar to that of Figure 3.2, showing both skewness and a fatter tail. And the QQ plot in panel (f) remain fatter-tailed. Figure 3.5 has the same characteristics as Figure 3.4, except that the r_t is not autocorrelated and the QQ plot for it show little skewness.

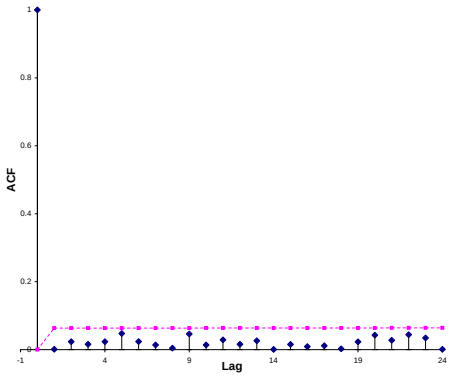
Again these figures just show part of the NASDAQ and MRK series. Most of the other parts not shown here are similar to these two. So both the series are het-



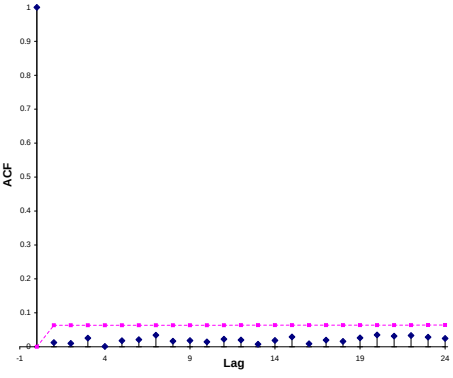
(a) Series r_t



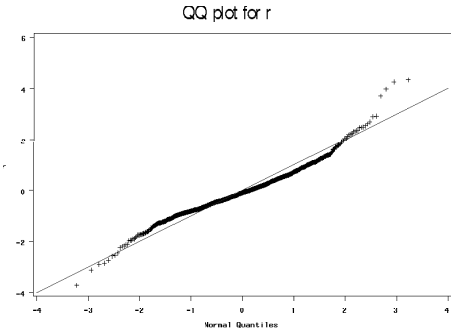
(b) Series r_t^2



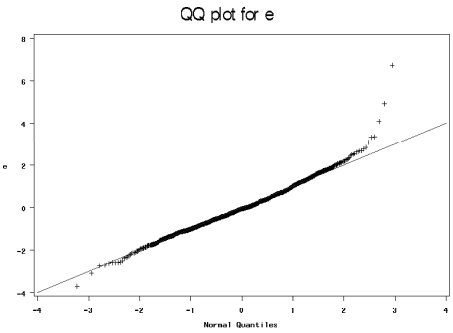
(c) Series e_t



(d) Series e_t^2

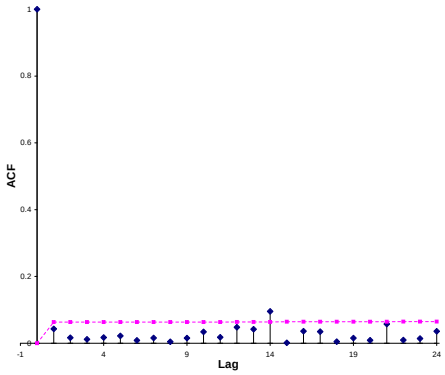


(e) QQ Plot for r_t

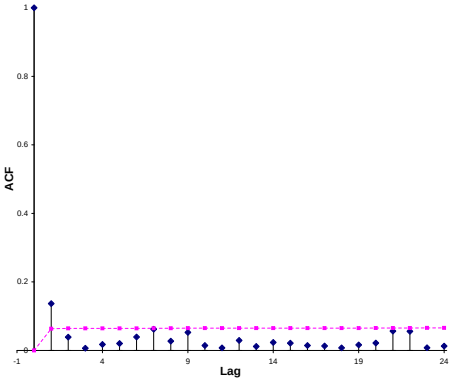


(f) QQ Plot for e_t

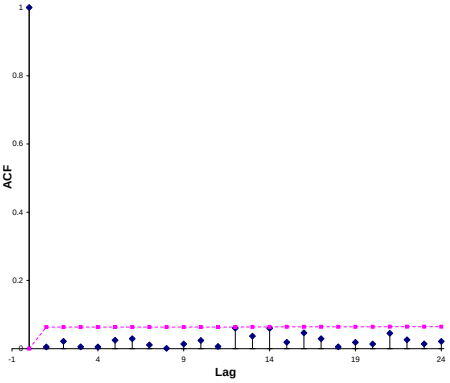
Figure 3.4: ACF and QQ Plots: NASDAQ



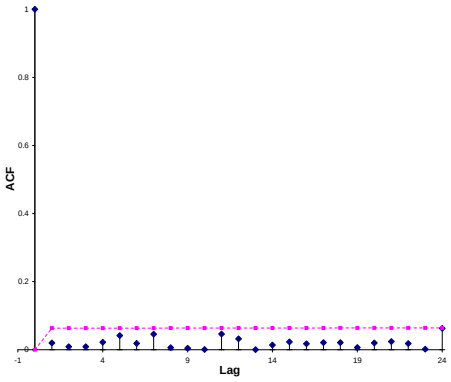
(a) Series r_t



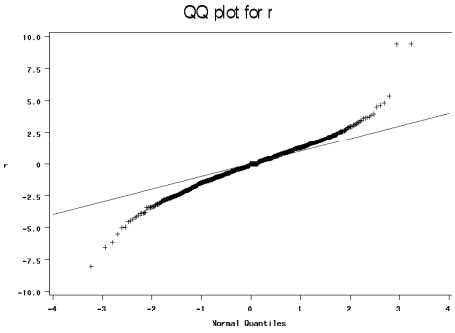
(b) Series r_t^2



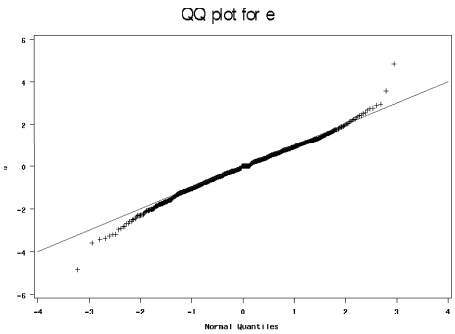
(c) Series e_t



(d) Series e_t^2



(e) QQ Plot for r_t



(f) QQ Plot for e_t

Figure 3.5: ACF and QQ Plots: MRK

eroscedastic and fatter-tailed. And our two-stage model can remove the heteroscedasticity from them.

Next, we examine how many nodes our two-stage model find for the real log returns. Table 3.10 summarizes the frequencies: For the NASDAQ log return series,

Data Series		0 node	1 node	2 nodes	3 nodes	4 nodes	Total
NASDAQ	frequency	0	3	136	316	545	1000
	percentage	0.0%	0.3%	13.6%	31.6%	54.5%	100%
MRK	frequency	225	52	122	168	433	1000
	percentage	22.5%	5.2%	12.2%	16.8%	43.3%	100%

Table 3.10: Frequency of Nodes: Real Data

Table 3.10 shows that our two-stage model regards the underlying model as a tree-structured GARCH(1,1) rather than a classical GARCH(1,1). And it has at least two nodes. For the MRK series, our two-stage model finds that 22.5% of the data sets have no node, but at the same time more than 60% of the data sets have at least two nodes. Competing the results for MRK with those of the simulated data, we see that MRK does not seem like any of the five generators. This suggests that the underlying model for MRK may not be any one of the five.

Finally, we list the AISALP statistics in Table 3.11. For the NASDAQ index, it is notable that our two-stage model is the best in estimating the volatilities. It has the smallest average in-sample absolute loss for prediction. For the MRK price series, all the AISALP statistics has much large values than those of the NASDAQ. But our TS model is still the one which has the smallest value.

	Model 1 RiskMetrics	Model 2 and 4 GARCH(1,1) and MF	TS Model Two-stage
NASDAQ	0.7294	0.6861	0.6283
MRK	12.1251	17.0263	7.0408

Table 3.11: AISALP Statistics

Statistics Related to VaR and ES estimates

For the VaR and ES, remember that we estimated them for $p = 0.95$, $p = 0.99$ and $p = 0.995$. The following two tables give the resulting means and standard deviations for the estimates of each model. Because we do not have the true values of VaR and ES any more, we can only evaluate the table values by their magnitudes.

Model		VaR $p = 0.95$	VaR $p = 0.99$	VaR $p = 0.995$	ES $p = 0.95$	ES $p = 0.99$	ES $p = 0.995$
TS	mean	1.3152	2.1232	2.4960	1.8251	2.6882	3.0896
	std	0.6645	0.9520	1.0799	0.8419	1.1400	1.2769
Model 1	mean	0.7794	1.0210	1.1094	0.9275	1.1411	1.2214
	std	0.5603	0.7262	0.7874	0.6618	0.8094	0.8652
Model 2	mean	1.5809	2.1627	2.3757	1.9377	2.4520	2.6456
	std	0.5470	0.7176	0.7802	0.6515	0.8026	0.8595
Model 3	mean	1.3292	2.3143	2.7703	1.9507	3.0042	3.4940
	std	0.0690	0.1440	0.1860	0.1165	0.2223	0.2945
Model 4	mean	1.3799	2.3532	2.8006	1.9932	3.0279	3.5060
	std	0.4806	0.7558	0.8701	0.6472	0.9141	1.0256

Table 3.12: Mean and Std for VaR and ES Estimates: NASDAQ

From Table 3.12, we can see that for the NASDAQ return series, Model 1, the

most often used RiskMetrics model in practice, is the one that always get the smallest estimates for VaR and ES. And the differences between the results of this model and the others are relatively large. This means that if this model is used to set the risk margins for investors, the money amount required is much less than those using other models. Except for Model 1, our TS model has the smallest mean estimates except for ES ($p = 0.99$) and ES ($p = 0.995$). For these two cases, Model 2, the classical GARCH(1,1), has the second smallest mean estimates. But our TS model estimates always have a large standard deviation than the others.

Model		VaR $p = 0.95$	VaR $p = 0.99$	VaR $p = 0.995$	ES $p = 0.95$	ES $p = 0.99$	ES $p = 0.995$
TS	mean	2.7015	4.8957	6.2630	4.5164	8.8498	12.2271
	std	5.2235	10.8237	14.3447	9.5524	21.9484	26.3786
Model 1	mean	3.7390	5.2748	5.8370	4.6807	6.0385	6.5495
	std	7.4391	10.1559	11.1526	9.1036	11.5099	12.4165
Model 2	mean	3.8276	5.4447	6.0368	4.8091	6.2489	6.7869
	std	6.9074	9.7733	10.8226	8.6646	11.1986	12.1523
Model 3	mean	2.4704	4.5898	5.9381	4.0060	7.3600	9.5611
	std	0.2414	0.6002	1.0143	0.5900	1.8135	2.9052
Model 4	mean	2.7757	5.1914	6.7001	4.4974	8.1908	10.5515
	std	5.1799	10.6936	14.0969	9.0296	17.1550	22.1914

Table 3.13: Mean and Std for VaR and ES Estimates: MRK

Table 3.13 shows that the means of the VaR and ES estimates for the single stock price series are much more larger than those for the index series. Furthermore, though all the models give similar means for most of the estimates (ES ($p = 0.99$) and ES

($p = 0.995$) are exceptions), the standard deviations are very different. For the competing models, except for the GPD model, which does not assume any distribution structure for the original log return series r_t, the other three models have estimates with surprisingly large standard deviations, especially Model 4, which is also a two-stage one. Actually, by examining the original data and the VaR and ES estimates carefully, we find that this is because a few of the original log returns and the corresponding estimates obtained by those models are extremely large. For example, the closing price of MRK on February 16, 1999 is 150.87, but that for the next day is only 74.67; this gives a r_t of 70.3340 for February 17, 1999, comparing with -2.2590 of the previous day. For the data set in which this day is the last observation, Model 1 gives an estimate for VaR (p=0.95) to be as large as 149.0255, but that obtained by Model 3 for the same data set is only 2.4358. And the extreme estimates do not always happen for the same data sets for all the four models, though most of them do. Compared with the three models, our TS model again has the largest stds.

Next, we do the backtesting for VaR and count the number of violations for the three p values. The results are summarized in Table 3.14. The frist row of this table is the expected number of violations out of the 1000 fits. The second row is the confidence interval calculated according to Equation 3.7.

Table 3.14 shows that both Model 3 and Model 4, i.e., the GPD model and the MF model are doing well for the NASDAQ series. All the actual counts of violations for these two models fall into the 95% confidence intervals. Our TS model does well for $p = 0.99$ and $p = 0.995$, but has more violations for $p = 0.95$ than it

	MODEL	$p = 0.95$	$p = 0.99$	$p = 0.995$
	Expected Violations	50	10	5
	Confidence Interval	36.5-63.5	3.8-16.2	0.6-9.4
NASDAQ	TS	67	10	6
	Model 1	174	131	118
	Model 2	47	17	11
	Model 3	58	13	9
	Model 4	61	11	6
MRK	TS	67	14	6
	Model 1	140	111	98
	Model 2	36	10	8
	Model 3	73	17	4
	Model 4	69	17	4

Table 3.14: Backtesting Results for VaR

should have. The RiskMetrics model used in practice obviously under-estimates all the three VaRs, resulting much more violations than the reasonable values given by the confidence intervals. As to the MRK series, Unlike the means and stds in Table 3.13, the counts of violations are not affected so much by the extreme estimates. To be specific, only Model 2 still has all the three counts falling in the confidence intervals. Our TS model remains the same as it does for the NASDAQ, but Model 3 and Model 4 have more violations than the confidence intervals indicate for $p = 0.95$ and $p = 0.99$. The RiskMetrics model (Model 2) again underestimates the VaR to such an extent that the actual violation counts are much larger than any of the other models. Putting both the NASDAQ and the MRK into consideration, we see that

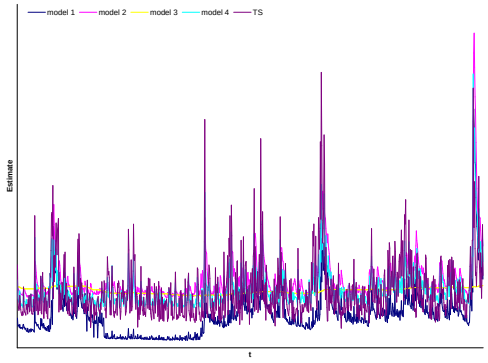
except for the RiskMetrics, all the other models have relatively small differences in violations counts, our TS model is a little worse than Model 2, but is a little better than Model 3 and Model 4.

And finally, to see how volatile the estimates are, we plot all the estimates of VaR and the Expected Shortfall for the NASDAQ index series in Figure 3.6. As the titles for the sub-figures indicate, the first three sub-figures are for VaR estimates, and the next three are for ES estimates.

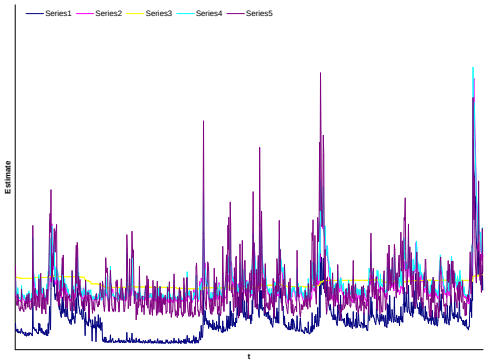
All the six sub-figures look similar: Model 1 and Model 3 give quite stable estimates; the estimates for the other three models change to quite some extent from day to day; and among the three, our two-stage model is the one that gives the most volatile estimates. From this figure, we can also see that the estimates of Model 1 are usually much smaller than those of the other models. Combining this with the numbers in Table 3.14, we know that it is because that Model 1 actually underestimates the risk measurements.

Conclusions and Comments for Real Data Application

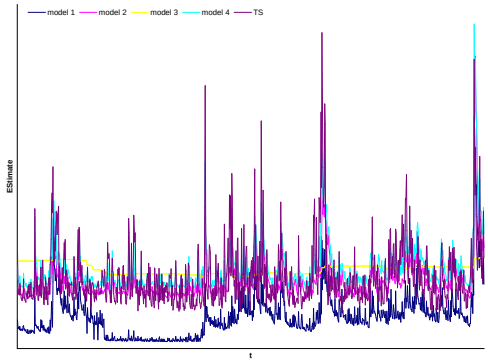
First, for the NASDAQ index, it seems that our TS model tends to believe it has a tree structure. But for the MRK price series, the TS model give complicated results in the frequency table. It seems more like that the series actually have different structures for different parts of it. Some parts of the data have a tree structure but some do not. Second, no matter what structure the underlying models have, for both the NASDAQ and the MRK, our TS model estimates the volatilities better



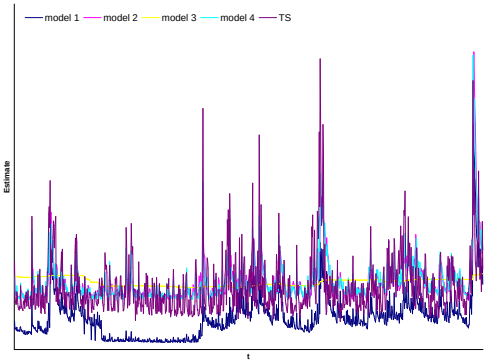
(a) VaR: $p = 0.95$



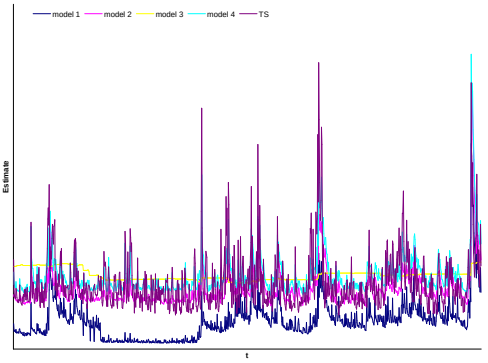
(b) VaR: $p = 0.99$



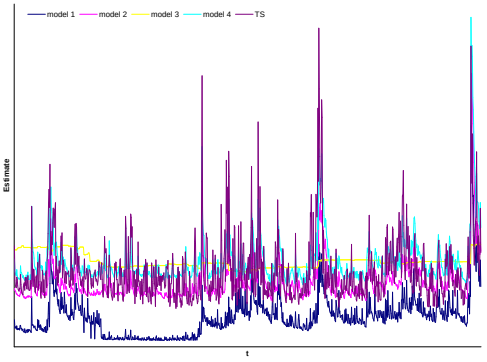
(c) VaR: $p = 0.995$



(d) ES: $p = 0.95$



(e) ES: $p = 0.99$



(f) ES: $p = 0.995$

Figure 3.6: Plots of Estimates: NASDAQ

than the other three models that do the same estimations according to the AISALP statistics. Third, the RiskMetrics model underestimates all the risk measurements, which is observed from the violation counts for both of the negative log return series. And this is why it gives smaller means of estimates for the NASDAQ. For the same NASDAQ series, except for the RiskMetrics, our TS model has the second smallest means of estimates most of the time. But it does not do well for the case $p = 0.995$. Also, the estimates of our TS model usually have larger standard variations than the other models. The MRK series have extreme observations, which result in larger means of estimates from all the models. The standard deviations of the estimates given by the GPD model are not affected very much by the extreme observations, but all the other models do not have this advantage, including our TS model, which again gives larger stds. Also, By plotting all the estimates of the NASDAQ series, we observe that the estimates of our TS model are more volatile than those of the other models. And both the RiskMtrics and the GPD model are rather stable. Finally, except for the RiskMetrics, all the other models show only small differences in the violation counts for the VaR estimates. Our TS model has only one count falling out of the confidence intervals for both the NASDAQ and the MRK series. It is better than all the other models other than Model 2, the classical GARCH(1,1) model.

We conclude this section with a few comments: Firstly, here we only applied the methods to the NASDAQ and the MRK stock, and from the application we can see that it is much more difficult to model the price series of a single stock. The price for a single stock can be easily affected by a lot of things, such as the company's

annual report, the amount of dividend given out, the amount of new stocks issued and so on. It is more changeable than an index series. Secondly, in order to facilitate the calculations and make the comparisons, we use an unique TS model structure for a large number of data sets. In reality, the model structure can actually be data dependent. For example, we can change the form of μ_t according to the characteristics of the data in hand. It is not necessarily to be an AR(1) used here. The sample size n can also be increased. Using these means may improve our estimates. Finally, we want to point out that modelling an index is not only easier, but also more meaningful, because much more financial instruments are based on the change of indices rather than that of a single stock.

3.3 Conclusions

In this chapter, we applied our two-stage method to both simulated data and two real negative log return series. Estimates of both Value at Risk and the Expected Shortfall are obtained for $p = 0.95$, $p = 0.99$ and $p = 0.995$. And we compared the results with those of four other models: the RiskMetrics, the classical GARCH(1,1) model, the threshold GPD model, and the McNeil and Frey (MF) model.

The simulated data are generated from five different data generators, including the classical GARCH(1,1) model with standard Normal innovation, the classical GARCH(1,1) model with standardized $t(3)$ innovation, the tree-structured GARCH(1,1) model with two nodes and standard Normal innovation, the tree-structured GARCH(1,1)

model with two nodes and standardized $t(3)$ innovation, and a non-GARCH model. We examined the number of nodes found by our two-stage model, and such statistics of the estimates as the Average In-Sample Relative Loss (AISRL) for $\hat{\sigma}_t^2$ and Average Relative Loss (ARL) of $\hat{R}_p^t$ and $\hat{S}_p^t$ are calculated.

The results show that our two-stage model estimates the risk measurements much better than the other models when the underlying model has a tree structure. When the underlying model is the classical GARCH(1,1) with standardized $t(3)$ innovation, our model also performs better for estimating ES ($p = 0.95$) and all the three VaRs. For the classical GARCH(1,1) model with standard Normal innovation and the non-GARCH data, RiskMetrics, classical GARCH, and the MF are better choices. And for the non-GARCH data, our model is only worse than the MF model. The threshold GPD model seems to be the one which gives the largest relative losses. But in general, our model gives estimates with larger standard deviations than they should have.

The two real negative log return series we used for real data application are the NASDAQ index series and the MRK stock price series. The NASDAQ index series are rather stable, but the MRK series contain some extreme observations. The Average In-Sample Absolute Loss for Prediction (AISALP) instead of the AISRL statistics is calculated. We also did backtestings and counted the actual number of violations for the three VaR estimates. Plots of all the estimates for the NASDAQ series are drawn to reflect the volatility.

It is observed from the results of real data application that the popular RiskMetrics model always underestimates both the risk measurements to quite some extent.

Except for this model, our two-stage model has smaller means of estimates for the NASDAQ series. For the MRK price series, the performances of our two-stage model, together with the other three models are greatly affected by the extreme value. Only the threshold GPD model, which does not assume any structure for the log return, remains stable. For both series, our two-stage model again gives estimates with larger standard deviations than the other models. And it has the most volatile estimates according to the plots. As to the violation counts, our two-stage model is satisfactory, only having larger number of violations for VaR ($p = 0.95$).

Generally speaking, the two-stage model we proposed here has advantages when the underlying model is tree-structured or has obvious fatter tail. And for other usual cases, this method is not much worse than the other existing methods. The real data application also shows reasonable results. Our model gives satisfactory number of violations for VaR estimates. The disadvantages are that it's estimates tend to have larger variations, and it does better for the Expected Shortfall with lower p and VaR than for the Expected Shortfall with higher p values. We suggest to use it to estimate these measurements for indices and stable stock price series, adding another perspective of view in addition to the current methods in literature.

Appendix A

Finding the Set of Subtrees

In the subsection of 2.2, "Backward: Reducing the Number of Nodes", we mentioned that it is easier to describe the concept of T, which is the set of all the binary subtrees of $\mathcal{P}_{opt}^{(M)}$, than to find it computationally. Here we describe the strategy we use to do this in our SAS program:

To find every element of T correctly, the key is to store the tree structure in an appropriate way. To be specific, the tree should be stored in such a way that it is very easy to tell which cell is an intermediate node or a leaf, and which one is not. We should be able to do this any time during the pruning process.

For illustration purpose, we still use the tree in Figure 2.1. Remember that during the growing process, we get $\{a_1, a_2\}$ first by using the splitting node d_1 in r, then get $\{b_1, b_2\}$ by splitting a_1 with d_2 in σ^2, and finally split a_2 with d_3 in σ^2 to get $\{c_1, c_2\}$. We store the tree in a data set which contains the first three columns of the following table:

dimension	node	left cell	subtree 3	subtree 5
r	d_1	2	2	2
σ^2	d_2	4	4	0
σ^2	d_3	6	0	6
0	0	0	0	-1
0	0	0	0	-1
0	0	0	-1	0
0	0	0	-1	0

Table A.1: The Subtrees of the Simple Binary Tree

The first row of the data set says that with the node d_1 in r, we split the data into two cells, and the left one, i.e., the one with the less-than-or-equal sign "$\leq$", is stored in the second row and the right one goes to the third row by default. So the first row is the root of the tree, the second row is cell a_1 and the third row is cell a_2. Since the values of "dimension" and "node" for the second row are not zero, cell a_1 is also split into two cells: the left one is the fourth row (cell b_1), and the right one is the fifth (cell b_2). These two cells are leaves because all the values for the first three columns are zero. The same thing happens to the third row (cell a_2), and it is split into two leaves: the sixth row (cell c_1) and the seventh (cell c_2).

After identifying the leaves and the nodes which produce the leaves, pruning the tree becomes rather easy: Notice that the third column is the full tree (or Subtree 4 in Figure 2.2), and only row four and five, or row six and seven are the end leaves that can be pruned. To get rid of row six and seven ($\{c_1, c_2\}$), just replace the zeros in the third column of these two rows with -1, and replace that of row three, the

node which produce them, with zero. The resulting subtree is stored in column four. It is Subtree 3 in Figure 2.2. Similarly, by pruning row four and five, we get column five which is Subtree 5 in Figure 2.2. We can see from both of the columns that now the subtrees have three leaves (zeros) and two nodes. Column four has already appeared in the growing process, since the nodes has the consecutive even numbers $\{2, 4\}$. Column five is a new subtree since the numbers are $\{2, 6\}$, which are not consecutive.

By further pruning column four and five, we always get Subtree 2 in Figure 2.2, and then Subtree 1. Thus we get all the subtrees in the set of subtrees T.

Appendix B

Additional Formulas

In order to calculate the statistics $ARL_{\hat{R}_p^t}$ and $ARL_{\hat{S}_p^t}$ proposed in Chapter 3.1, we need the true values of R_p^t and S_p^t for model 3.1. When r_t and σ_t are known for $t = 1000$, no matter which volatility equation we use to generate the data, Equation 3.2, Equation 3.3 or Equation 3.4, it is obvious that we can always get the true σ_{t+1} according to it. Then by the definition of VaR, for our data generating models with standardized t innovations, the true VaR is just

$$R_p^t = \mu_{t+1} + \sigma_{t+1} z_p = \sigma_{t+1} \frac{CDF_\epsilon^{-1}(p)}{\sqrt{3}}. \tag{B.1}$$

Here $CDF_\epsilon^{-1}(.)$ is the inverse CDF of ϵ_t. And remember that we let $\phi = 0$ and $\sqrt{(3/1)}\epsilon_t \sim t_3$ in Equation 3.1.

Similarly for the same data generators, the true value of the Expected Shortfall can be calculated according to the following formula:

$$S_p^t = \mu_{t+1} + \sigma_{t+1} E\left[\epsilon | \epsilon > z_p\right] = \sigma_{t+1} \times \frac{3}{[3 + (CDF_\epsilon^{-1}(p))^2]\pi} \times \frac{1}{(1-p)}. \tag{B.2}$$

Except for the formulas for true VaR and the Expected Shortfall, to compare the different models, we also need to get the estimated R_p^t and S_p^t for the four models mention in Chapter 3.1.

For IGARCH(1,1) and GARCH(1,1), $\hat{\sigma}_{t+1}$ can be obtained according to their volatility equations, provided the estimated parameters and σ_t. Then because we assume $\epsilon_t \sim N(0,1)$ in these models, by a little inference, the formulas for $\hat{R}_p^t$ and $\hat{S}_p^t$ are:

$$\hat{R}_p^t = \hat{\sigma}_{t+1} \times CDF_{N(0,1)}^{-1}(p) \tag{B.3}$$

and

$$\hat{S}_p^t = \hat{\sigma}_{t+1} \times \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(CDF_{N(0,1)}^{-1}(p))^2}{2}\right) \times \frac{1}{(1-p)}. \tag{B.4}$$

Please notice that these two formulas can also be used to get the true VaR and the Expected Shortfall for our data generators with the standard Normal innovations. The only change to be made is using σ_{t+1} instead of $\hat{\sigma}_{t+1}$ in both of the formulas.

For the threshold GPD model, since it simply fits a GPD model to the top 10% of r_t, Equation 2.19 and Equation 2.23 can be used directly to get $\hat{R}_p^t$ and $\hat{S}_p^t$. The only revision is that we need to use the corresponding symbols of r_t instead of those of the residuals. So the formulas are

$$\hat{R}_p^t = r_{(m+1)} + \frac{\hat{\gamma}}{\hat{\xi}}\left(\left(\frac{1-p}{m/n}\right)^{-\hat{\xi}} - 1\right) \tag{B.5}$$

and

$$\hat{S}_p^t = \hat{R}_p^t\left(\frac{1}{1-\hat{\xi}} + \frac{\hat{\gamma} - \hat{\xi} r_{(m+1)}}{(1-\hat{\xi})\hat{R}_p^t}\right). \tag{B.6}$$

With all these formulas and the ones proposed in Chapter 2.4 in hand, we are able to get all the true values and estimates of VaR and the Expected Shortfall. Hence are able to calculate the $ARL_{\hat{R}_p^t}$ and $ARL_{\hat{S}_p^t}$ statistics to do the comparisons.

Bibliography

[1] C. Acerbi and D. Tasche. Expected shortfall: A natural coherent alternative to value at risk. *Economic Notes*, 31(2):379–388, 2002.

[2] P. Artzner, J. Delbaen, and D. Heath. Thinking coherently. *Risk*, 10(11):68–71, 1997.

[3] P. Artzner, J. Delbaen, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[4] F. Audrino and P. Buhlmann. Tree-structured generalized autoregressive conditional heteroscedastic models. *Journal of Royal Statistical Society B*, 63 Part 4:727–744, 2001.

[5] J. Boudoukh, M. Richardson, and R. Whitelaw. The best of both worlds. *Risk*, 11:64–67, 1998.

[6] J. S. Butler and B. Schachter. Improving value-at-risk estimates by combining kernel estimation with historical simulation. mimeo, Vanderbilt University and Comptroller of the Currency, 1996.

[7] V. Chernozhukov. Extremal quantile regression. Preprint, 2002.

[8] M. Davidian. Nonlinear models for univariate and multivariate response. Class Notes Chapter 3-8, 2001.

[9] J. E. Dennis and H. H. W. Mei. Two new unconstrained optimization algorithms which use function and gradient values. *Journal of Optimization Theory and Applications*, 28:453–483, 1979.

[10] D. Duffie and J. Pan. An overview of value at risk. *Journal of Derivatives*, Spring 1997:7–49, 1997.

[11] R. F. Engle and S. Manganelli. Caviar: Conditional value at risk by quantile regression. Working Paper 7341, 1999.

[12] D. M. Gay. Subroutines for unconstrained minimization. *AMC Trans. Math. Software*, 9:503–524, 1983.

[13] B. V. Gnedenko. Sur la distribution limite du terme maximum dune serie aleatoire. *Annals of Mathematics*, 44:423–453, 1943.

[14] B. M. Hill. A simple general approach to inference about the tail of a distribution. *Annals of Statistics*, 46:1163–1173, 1975.

[15] P. Jorion. *Value at Risk: The New Benchmark for Controlling Market Risk.* Irwin, Chicago, 1997.

[16] F. M. Longin. The asymptotic distribution of extreme stock market returns. *Journal of Business*, 69 no.3, 1996.

[17] F. M. Longin. From value at risk to stress testing: The extreme value approach. *Journal of Banking and Finance*, 24:1097–1130, 2000.

[18] A. J. McNeil and R. Frey. Estimation of tail-related risk measures for heteroscedastic financial time series: an extreme value approach. *Journal of Empirical Finance*, 7:271–300, 2000.

[19] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 1999.

[20] J. Pickands. Statistical inference using extreme order statistics. *Annals of Statistics*, 3:119–131, 1975.

[21] R. Smith. Estimating tails of probability distributions. *The Annals of Statistics*, 15:1174–1207, 1987.

[22] R. L. Smith. Measuring risk with extreme value theory. Preprint, 1999.

[23] R. S. Tsay. *Analysis of Financial Time Series: Financial Econometrics*. Wiley, 2002.