THE UNIVERSITY OF AUCKLAND
FACULTY OF SCIENCE

Semiparametric Scale Mixtures of Normal Distributions

by

Keng-Hao(Danny) Chang

supervised by Senior Lecturer Yong Wang

A thesis submitted in partial fulfillment
of the requirements for the degree of Master of Science
in the Department of Statistics

February, 2010

Abstract

Consider heavy-tailed data (such as financial data) grouped by numbers of different sources or behavior (such as the stock market crash of 1987). Modeling these data are not easy, as many traditional heavy-tailed distributions (such t-distribution and Laplace distribution) are not adequate.

Scale mixtures of normal distributions is constructed by mixing of normal distribution with common means but different variances, forming a very flexible and rich family of heavy-tailed distributions. Many traditional heavy-tailed distributions are just the special cases of this form.

The aim of this research was to study scale mixtures of normal distributions and develop their possible applications, such as density estimation, modeling log returns and robust linear regression.

Results from simulation and comparison with selected benchmarks suggest that our scale mixtures of normal distributions approach has the best result. Thus we claim it is a very strong candidate in many heavy-tailed or robustness analysis.

The Constraint Newton method (CNM) has been developed to find nonparametric maximum likelihood estimate for mixture model. It is a fast and stable algorithm compared to existing algorithms. The maximization by modifying the support set algorithm enhanced from the CNM algorithm (CNM-MS) has also been developed to find semiparametric maximum likelihood estimate for mixture model. We use the CNM-MS algorithm during our analysis.

Acknowledgements

I would like to express the deepest appreciation to my supervisor, Dr Yong Wang for his encouragement, guidance and support enabling me to develop an understanding of the subject.

I would also like to show my gratitude to the Marsden Fund which provided the generous scholarship.

The analysis for this these was implemented by statistical programming language `R` version 2.10.0 (R Development Core Team, 2009). Simulations were performed on the Sun N1 Grid Engine provided by the Department of Statistics, for which I am very grateful.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The research for this thesis focused on semiparametric scale mixtures of normal distributions and possible applications. This chapter introduces the research aim and outline of this thesis.

1.1 Research aim

Mixture model

Mixture models are a probability model constructed by mixtures of distributions, forming a rich and flexible family. They have identified applications in many fields, such as astronomy, biology, medicine, economics, engineering and marketing. Mixture models play a major role of in their analysis, for example, survival (failure time) analysis, factor analysis, density estimation and cluster analysis.

Scale mixtures of normal distributions

Scale mixtures of normal distributions are probability density functions constructed by mixtures of normal distributions with common mean but different variances in some distribution (discrete or continuous).

It has been proved that many traditional heavy-tailed distributions are in this form (see Chapter 3). They form a very rich family of heavy-tailed distributions, and can be applied on many areas.

We study scale mixtures of univariate (and multivariate) normal distributions in this thesis, and develop possible applications.

For scale mixtures of univariate normal distributions, we perform density estimation on simulated heavy-tailed data for different distributions, and also model log returns and estimate Value-at-Risk on various financial data. We develop robust linear regression by replacing normal distribution error assumption with scale mixtures of univariate normal distributions. We then extend to multivariate case. We perform density estimation on simulated heavy-tailed data for different distributions and various dimensions. We also model multiple log returns and estimate Value-at-Risk on different financial data.

Algorithm

The Constraint Newton method (CNM) has been developed by Wang (2007a) to find nonparametric maximum likelihood estimate for mixture model. It is fast and stable compared to other existing algorithms.

Wang (2010) extended the CNM algorithm and developed three new CNM-based algorithms, maximization by alternating the parameters (CNM-AP), maximization by profiling the likelihood (CNM-PL) and maximization by modifying the support set (CNM-MS). The main aim of these CNM-based algorithms is to find semiparametric maximum likelihood estimates for mixture models. They are also very stable and fast due to the superiority of the CNM algorithm.

We use the CNM-MS algorithms to find maximum likelihood estimate of scale mixtures of normal distributions during analysis.

1.2 Thesis outline

This section describe the structure of this thesis. Chapter 2 introduce the background on the research topic and discuss the algorithm we implement. Chapter 3 introduce the scale mixtures of univariate normal distributions, describe the implementation of algorithm for finding semiparametric maximum likelihood estimate, and discuss its possible applications.

Chapter 4 extends scale mixtures of univariate normal distributions to multivariate cases. We describe the implementation and modification of the algorithm for finding semiparametric maximum likelihood estimates. We also develop possible applications. Finally, Chapter 5 concludes this thesis and provide some possible further research.

Appendix A presents the source code we use in this thesis for reference. The Bibliography provide a list of references.

Chapter 2

Background

This chapter provides background on the topic of this thesis. Section 2.1 introduces mixture models. Section 2.2 describes the constraint Newton method (CNM), which is the algorithm of computing nonparametric maximum likelihood estimates for mixture models. Section 2.3 describes three CNM-based algorithms for computing semiparametric maximum likelihood estimates for mixture models.

Section 2.4 describes some miscellanea for implementation of CNM-based algorithms. Section 2.5 briefly describes scale mixtures of normal distributions and our research aim. Finally, Section 2.6 gives some concluding remarks and leads to Chapter 3.

2.1 Mixture Model

Mixture models are a probability models constructed by mixtures of distributions. They provide a flexible family and have been used in many areas. Huber (1964) considered robust M-estimators of a location parameter for contaminated normal distributions (mixtures of two normal distributions), McLachlan and Basford (1988) studied clustering of data via mixture model and Peel et al. (2001) analyzed heterogeneous directional data for mixture models. Mixture models ap-

proach have also been used in survival (failure time) analysis, factor analysis, density estimation, and applied to hidden Markov model.

If the random variable θ has unknown distribution G, the mixture model density can be expressed in the form

$$f(x;G) = \int_{\Theta} f(x;\theta)\mathrm{d}G(\theta), \tag{2.1}$$

where $G(\theta) \geq 0$ is the mixing distribution function, $\theta \in \Theta$, and $f(x;\theta)$ is the component density.

Suppose the random variable X has probability density function described in above equation, and a random sample $x_1, \ldots, x_n$ with sample size n is taken. The likelihood function can be expressed by

$$L(G) = \prod_{i=1}^{n} f(x_i;G) = \prod_{i=1}^{n} \int_{\Theta} f(x_i;\theta)\mathrm{d}G(\theta) \tag{2.2}$$

and therefore log-likelihood function is

$$l(G) = \sum_{i=1}^{n} \log \left\{ \int_{\Theta} f(x_i;\theta)\mathrm{d}G(\theta) \right\}. \tag{2.3}$$

Finite mixture models

A finite mixture model is a mixture model that a has a discrete mixing distribution G_d with known dimension. Most of the literature for mixture models is about finite mixture models, however, the dimension is usually unknown in practice. The dimension of G_d is a critical issue when studying finite mixture model.

We study semiparmetric mixture model in this thesis, with parametric finite dimensional vector $\boldsymbol{\mu}$ and nonparametric mixing distribution G. We avoid the problem of deciding the dimension of G_d by using our semiparametric maximum likelihood estimate algorithm discussed in Section 2.2 and 2.3.

2.2 Nonparametric mixture models

The nonparametric problem can be setup here, with unknown G. There exists a distinct discrete G maximizing $l(G)$ with everything else fixed (see Laird (1978) and Lindsay (1983)), denoted by the nonparametric maximum likelihood estimate $\hat{G}$. Thus, it is sufficient to consider discrete G only when computing $\hat{G}$.

The discrete G consist of support set $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$ and corresponding weight $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$. We can rewrite the mixture density defined in Equation (2.1) as

$$f(x; \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k f(x; \theta_k) \tag{2.4}$$

and log-likelihood

$$l(G) = l(\boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \pi_k f(x_i; \theta_k) \right\}, \tag{2.5}$$

where $\boldsymbol{\pi} \in [0, 1]$ and sum to 1.

Computing nonparametric maximum likelihood estimate $\hat{G}$ is equivalent to find optimized $\hat{\boldsymbol{\pi}}$ and corresponding support set $\hat{\boldsymbol{\theta}}$ with unknown dimension K that maximized $l(G)$.

The Constrained Newton method (CNM) is a stable and fast algorithm for computing the nonparametric maximum likelihood estimate of a mixing distribution introduced by Wang (2007a). It compute iteratively by converting the nonparametric maximum likelihood estimate problem to least square problem in terms of $\boldsymbol{\pi}$, using the non-negative least squares (NNLS) algorithm provided by Lawson and Hanson (1974) to optimize $\hat{\boldsymbol{\pi}}$ and computing support set $\hat{\boldsymbol{\theta}}$ by using the method introduced by Lesperance and Kalbfleisch (1992). The CNM algorithm has guaranteed quadratic convergence, and it is very fast (in terms of number of iteration).

2.2.1 CNM - find mixing proportions π

The CNM algorithm update mixing proportions $\boldsymbol{\pi}$ by quadratic approximation of the log-likelihood function with fixed $\boldsymbol{\theta}$. Let $\boldsymbol{\pi}'$ be the new updated proportion of $\boldsymbol{\pi}$. Denote $\mathbf{S}^{\mathrm{T}} = (\mathbf{s_1}, \ldots, \mathbf{s_n})$ where

$$\begin{aligned} \mathbf{s_i} &= \frac{\partial \log f(x_i; \boldsymbol{\pi}, \boldsymbol{\theta})}{\partial \boldsymbol{\pi}} \\ &= \left(\frac{f(x_i; \theta_1)}{f(x_i; \boldsymbol{\pi}, \boldsymbol{\theta})}, \ldots, \frac{f(x_i; \theta_K)}{f(x_i; \boldsymbol{\pi}, \boldsymbol{\theta})}, \right)^{\mathrm{T}}. \end{aligned} \tag{2.6}$$

Thus, the first derivative of $l(\boldsymbol{\pi}, \boldsymbol{\theta})$, $\mathbf{J}$, in terms of $\mathbf{S}$ is

$$\mathbf{J} = \frac{\partial l(\boldsymbol{\pi}, \boldsymbol{\theta})}{\partial \boldsymbol{\pi}} = \mathbf{S}^{\mathrm{T}} \mathbf{1}, \tag{2.7}$$

where $\mathbf{1}$ is vector of ones. The second derivative of $l(\boldsymbol{\pi}, \boldsymbol{\theta})$ is the Hessian matrix, $\mathbf{H}$, in terms of $\mathbf{S}$ is

$$\mathbf{H} = \frac{\partial l(\boldsymbol{\pi}, \boldsymbol{\theta})}{\partial \boldsymbol{\pi} \partial \boldsymbol{\pi}^T} = -\mathbf{S}^{\mathrm{T}} \mathbf{S}. \tag{2.8}$$

Denote difference between $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ as $\boldsymbol{\eta}$, $\boldsymbol{\eta} = \boldsymbol{\pi}' - \boldsymbol{\pi}$. By using a Taylor expansion, a quadratic approximation of $l(\boldsymbol{\pi}', \boldsymbol{\theta})$ can be expressed as

$$l(\boldsymbol{\pi}', \boldsymbol{\theta}) \approx l(\boldsymbol{\pi}, \boldsymbol{\theta}) + \mathbf{J}^{\mathrm{T}} \boldsymbol{\eta} + \frac{1}{2} \boldsymbol{\eta}^{\mathrm{T}} \mathbf{H} \boldsymbol{\eta}. \tag{2.9}$$

Therefore, the difference for the current and updated log-likelihood, $Q(\boldsymbol{\pi}' | \boldsymbol{\pi})$, can be approximated by

$$\begin{aligned} Q(\boldsymbol{\pi}' | \boldsymbol{\pi}) &= l(\boldsymbol{\pi}, \boldsymbol{\theta}) - l(\boldsymbol{\pi}', \boldsymbol{\theta}) \\ &\approx -\mathbf{J}^{\mathrm{T}} \boldsymbol{\eta} - \frac{1}{2} \boldsymbol{\eta}^{\mathrm{T}} \mathbf{H} \boldsymbol{\eta} \\ &= -\mathbf{1}^{\mathrm{T}} \mathbf{S} \boldsymbol{\eta} + \frac{1}{2} \boldsymbol{\eta}^{\mathrm{T}} \mathbf{S}^{\mathrm{T}} \mathbf{S} \boldsymbol{\eta} \\ &= \frac{1}{2} \left(\boldsymbol{\eta}^{\mathrm{T}} \mathbf{S}^{\mathrm{T}} \mathbf{S} \boldsymbol{\eta} - \mathbf{2}^{\mathrm{T}} \mathbf{S} \boldsymbol{\eta} + n \right) - \frac{n}{2} \\ &= \frac{1}{2} ||\mathbf{S} \boldsymbol{\eta} - \mathbf{1}||^2 - \frac{n}{2} \\ &= \frac{1}{2} ||\mathbf{S} \boldsymbol{\pi}' - \mathbf{S} \boldsymbol{\pi} - \mathbf{1}||^2 - \frac{n}{2} \\ &= \frac{1}{2} ||\mathbf{S} \boldsymbol{\pi}' - \mathbf{2}||^2 - \frac{n}{2}, \end{aligned} \tag{2.10}$$

where $\mathbf{2}$ is vector of twos. Finding on updated $\boldsymbol{\pi}'$ which maximizes $l(\boldsymbol{\pi}', \boldsymbol{\theta})$ with fixed $\boldsymbol{\theta}'$ is now equivalent to minimize least square with constraints,

$$\underset{\boldsymbol{\pi}'}{\text{Minimize}} \, ||\mathbf{S}\boldsymbol{\pi}' - \mathbf{2}||^2, \tag{2.11}$$

where $\boldsymbol{\pi}' \geq \mathbf{0}$ and sums to one.

Minimization of least squares (2.11) can be solved in many ways, Wang (2007a) suggests using the method from Haskell and Hanson (1981). The updated proportion $\boldsymbol{\pi}'$ can be found by solving the least square problem with non-negativity constraints,

$$\underset{\boldsymbol{\pi}'}{\text{Minimize}} \, |\boldsymbol{\pi}'^{\mathrm{T}}\mathbf{1} - 1|^2 + \gamma||\mathbf{S}\boldsymbol{\pi}' - \mathbf{2}||^2, \quad \text{subject to } \boldsymbol{\pi}' \geq \mathbf{0}, \quad \boldsymbol{\pi}'^{\mathrm{T}}\mathbf{1} = 1, \tag{2.12}$$

where γ has to be carefully chosen for some small value, Wang (2007a) suggests $\gamma = n \times 10^{-6}$ since it works practically well.

Wang (2010) also suggests a new approach using Dax (1990) method to solve least squares (2.11) problem with non-negativity constraints only, avoiding choosing γ, in the form

$$\underset{\boldsymbol{\pi}'}{\text{Minimize}} \, |\boldsymbol{\pi}'^{\mathrm{T}}\mathbf{1} - 1|^2 + ||\mathbf{E}\boldsymbol{\pi}'||^2, \quad \text{subject to } \boldsymbol{\pi}' \geq \mathbf{0}, \tag{2.13}$$

where matrix $\mathbf{E} = (\mathbf{s}_1 - \mathbf{2}, \ldots, \mathbf{s}_K - \mathbf{2})$, and $\mathbf{s}_j$ is jth column of $\mathbf{S}$.

Both implementations can be solved numerically by using the non-negative least squares (NNLS) algorithm from Lawson and Hanson (1974). The NNLS algorithm solve least squares minimization problem with non-negativity constraints by tuning solutions in two index sets, $\digamma$ and Ω. Variables indexed in the set Ω are all set to zero. Variables indexed in set $\digamma$ are free to choose any value greater than 0. Negatively valued variables are forced to move into set $\digamma$ or Ω.

Thus, the solution $\boldsymbol{\pi}'$ can be found using the NNLS algorithm and some elements of $\boldsymbol{\pi}'$ may set to 0 during iterations. We discard any zero entry of $\boldsymbol{\pi}'$ before the next iteration. Wang (2007a) also implements the back-tracking line search strategy with the Armijo rule to check log-likelihood is increasing at each iteration.

Lawson and Hanson (1974) provided a Fortran implementation of the NNLS algorithm, and Wang (2007b) also provide the NNLS algorithm implementation in package `lsei` for statistical software `R`.

2.2.2 CNM - find support set θ

The CNM algorithm find nonparametric maximum likelihood estimate for new support set $\boldsymbol{\theta}'$ with fixed $\boldsymbol{\pi}$ by using directional derivatives of log-likelihood as implemented by Wang (2007a) and Lesperance and Kalbfleisch (1992). The directional derivative (gradient function) of the log-likelihood with fixed G is defined as

$$\begin{aligned} d(\theta; G) &= \left. \frac{\partial l\{(1-h)G + h\delta_\theta\}}{\partial h} \right|_{h=0} \\ &= \sum_{i=1}^{n} \frac{f(x_i;\theta)}{f(x_i;G)} - n, \end{aligned} \tag{2.14}$$

where δ_θ is the degenerate distribution at θ. The nonparametric maximum likelihood estimate $\hat{G}$ can be equivalently established by following properties with the gradient function $d(.)$,

1. $\hat{G}$ maximizes $l(G)$
2. $\hat{G}$ minimizes $\sup_\theta\{d(\theta; G)\}$
3. $\sup_\theta\{d(\theta; \hat{G})\} = 0$
4. $\sup_\theta\{d(\theta; G)\} \geq l(\hat{G}) - l(G)$

as described in Wang (2007a) and Lesperance and Kalbfleisch (1992), the theoretical results of these theorems can be found in Lindsay (1983) and Lindsay (1995). First three properties are also known as the *general equivalence theorem* (for example, see Kiefer and Wolfowitz (1956)).

Therefore, we can find new good support set $\boldsymbol{\theta}$ by computing local maxima of gradient function $d(\theta; G)$, Wang (2007a) uses a slightly modified method created

by Lesperance and Kalbfleisch (1992) to locate the solution. The univariate Newton method is used to find local maxima of gradient function, which is equivalent to the roots of the first derivative of gradient function $d(\theta; G)'$ with respect to θ. As we know, the Newton method may not converge depending on the shape of the function and the starting point, therefore we apply the bisection method if the Newton method does not converge. The combined Newton-bisection process is described as following steps,

1. generate a number of grid point of θ, say 100, equally spaced.
2. apply the Newton method on any interval $[a, b]$ where $d(a; G)' > 0$ and $d(b; G)' < 0$ to locate a local maximum.
3. if any solution of the Newton method is found outside the interval $[a, b]$ during Newton method iteration, stop, and apply bisection method on the interval $[a, b]$.

2.2.3 CNM - summary

By using techniques that we introduced in last two sections (Section 2.2.2 and 2.2.1), the Constrained Newton method (CNM) introduced by Wang (2007a) can be summarized as following steps,

1. set initial estimate of G_0 where $l(G_0) > -\infty$
2. compute all local maxima $\boldsymbol{\theta}^*$ of gradient function $d(\boldsymbol{\theta}; G)$,
   If $\max_j \{d(\boldsymbol{\theta}_j^*); G\} = 0$, stop.
3. set $\boldsymbol{\theta}^+ = (\boldsymbol{\theta}^{\mathrm{T}}, \boldsymbol{\theta}^{*\mathrm{T}})^{\mathrm{T}}$ and $\boldsymbol{\pi}^+ = (\boldsymbol{\theta}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}})^{\mathrm{T}}$, find $\boldsymbol{\pi}^-$ by minimizing $Q(\boldsymbol{\pi}'|\boldsymbol{\pi}^+)$ and conducting a line search.
4. discard all support point with zero entries in $\boldsymbol{\pi}^-$, denote $\boldsymbol{\pi}'$ and $\boldsymbol{\theta}'$.
5. update $(\boldsymbol{\pi}, \boldsymbol{\theta})$ with $(\boldsymbol{\pi}', \boldsymbol{\theta}')$, back to step 2

Using the CNM algorithm described above to compute nonparametric maximum likelihood estimate for mixing distribution guarantees convergence, and is very fast and stable compared with other existing algorithms. Detailed theoretical results for convergence and comparison can be found in Wang (2007a).

2.3 Semiparametric mixture models

A semiparametric mixture model is the extension of nonparametric mixture models that we discussed in last section. It consists of two separate parameters, nonparametric mixing distribution G and parametric finite dimensional vector $\boldsymbol{\mu}$. The finite dimensional vector $\boldsymbol{\mu}$ makes mixture model much more flexible and general, and can be used to solve many types of problems.

We can rewrite some equations in Sections 2.1 in addition to a finite and known dimensional parameter $\boldsymbol{\mu}$. The density of mixture model can be rewritten as

$$f(x; G, \boldsymbol{\mu}) = \int_{\Theta} f(x; \theta, \boldsymbol{\mu}) \mathrm{d}G(\theta), \tag{2.15}$$

with log-likelihood

$$l(\mathbf{x}; G, \boldsymbol{\mu}) = \sum_{i=1}^{n} \log \left\{ \int_{\Theta} f(x_i; \theta, \boldsymbol{\mu}) \mathrm{d}G(\theta) \right\}. \tag{2.16}$$

Denote a semiparametric maximum likelihood estimate $(\hat{G}, \hat{\boldsymbol{\mu}})$ maximizing $l(G, \boldsymbol{\mu})$. As we discussed in Section 2.2, it is sufficiently to consider discrete G only when computing semiparametric maximum likelihood estimate $(\hat{G}, \hat{\boldsymbol{\mu}})$. We can rewrite density and log-likelihood of the semiparametric mixture model as discrete G,

$$f(x; \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\mu}) = \sum_{k=1}^{K} \pi_k f(x; \theta_k, \boldsymbol{\mu}), \tag{2.17}$$

and

$$l(\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{\mu}) = \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \pi_k f(x_i; \theta_k, \boldsymbol{\mu}) \right\}. \tag{2.18}$$

The directional derivative (gradient function) of log-likelihood with fixed G is

$$d(\theta; G, \boldsymbol{\mu}) = \sum_{i=1}^{n} \frac{f(x_i; \theta, \boldsymbol{\mu})}{f(x_i; G, \boldsymbol{\mu})} - n, \tag{2.19}$$

We can compute maximum likelihood estimate of nonparametric mixing distribution G by using the CNM algorithm we discussed in Section 2.2, but it is not suitable for semiparametric mixture model. However, it still provide a very good possible starting point.

Gradient function (2.19) is the key component for computing semiparametric maximum likelihood estimate. Consider the profile likelihood

$$l(\boldsymbol{\mu}) = l(\hat{G}_{\boldsymbol{\mu}}, \boldsymbol{\mu}) \tag{2.20}$$

where $\hat{G}_{\boldsymbol{\mu}}$ is the nonparametric maximum likelihood estimate for every fixed $\boldsymbol{\mu}$. The parametric maximum likelihood estimate $\hat{\boldsymbol{\mu}}$ can be found on this profile likelihood. However, it may have multiple local maxima on the profile likelihood. Recall properties we discussed in Section 2.2.2, any local maximum on the profile likelihood must satisfy

- $\frac{\partial l(G,\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = 0$.
- $\frac{\partial^2 l(G,\boldsymbol{\mu})}{\partial \boldsymbol{\mu} \partial \boldsymbol{\mu}^{\mathrm{T}}}$ is negative definite.
- $\sup_\theta \{d(\theta; G, \boldsymbol{\mu})\} = 0$.

We use these properties with an unconstrained optimization method to find $\hat{\boldsymbol{\mu}}$.

Wang (2010) extend the CNM algorithm, and introduce another three CNM-based algorithms to compute semiparametric maximum likelihood estimate for mixture model. These new CNN-based algorithms are maximization by alternating the parameters (CNM-AP), maximization by profiling the likelihood (CNM-PL) and maximization by modifying the support set (CNM-MS). All three CNM-based algorithms are guaranteed to find the local maximum of profile likelihood (2.20) and avoid stopping at local maximum by trying different initial values to increase the chances of reaching global maximum.

We only use the CNM-MS algorithm in this thesis since it takes least time to compute in our research problem (note that all three algorithms have similar result). The CNM-MS algorithm has the as following steps,

1. set initial estimate of $(G_0, \boldsymbol{\mu}_0)$ where $l(G_0, \boldsymbol{\mu}_0) > -\infty$.
2. update $(\boldsymbol{\pi}, \boldsymbol{\theta})$ to $(\boldsymbol{\pi}^+, \boldsymbol{\theta}^+)$ by using the CNM algorithm with $\boldsymbol{\mu}$ fixed.
3. update $(\boldsymbol{\pi}^+, \boldsymbol{\theta}^+, \boldsymbol{\mu})$ to local maximum $(\boldsymbol{\pi}', \boldsymbol{\theta}', \boldsymbol{\mu}')$ by using unconstrained optimization method.
4. if converges, stop, otherwise go return to step 2.

Many unconstrained optimization method can be used in the CNM-MS algorithm step 3, Wang (2010) implement the well known Broyden-Fletcher-Goldfarb-Shanno (BFGS) method. The BFGS method is the well-known quasi-Newton method used to solve unconstrained nonlinear optimization problems (see Fletcher (1987)).

2.4 Miscellanea

In summary, implementation of the CNM-MS algorithm consists of some critical components for different mixture modes. We rewrite these critical functions in logarithm format since it is more convenient in practice, turning multiplication to summation, and suitable for object-oriented programming aspects. To find local maxima of gradient function (2.19) by using the univariate Newton method, we need the first and second directive of gradient function (2.19). Thus, the gradient

function (2.19), its first and second directive respect to θ can be expressed as

$$d(\theta; G) = \sum_{i=1}^{n} N_i - n, \tag{2.21}$$

$$\frac{\partial d(\theta; G)}{\partial \theta} = \sum_{i=1}^{n} N_i \frac{\partial \log f(x_i; \theta, \boldsymbol{\mu})}{\partial \theta}, \tag{2.22}$$

$$\frac{\partial^2 d(\theta; G)}{\partial \theta^2} = \sum_{i=1}^{n} N_i \left[\left(\frac{\partial \log f(x_i; \theta, \boldsymbol{\mu})}{\partial \theta} \right)^2 + \frac{\partial^2 \log f(x_i; \theta, \boldsymbol{\mu})}{\partial \theta^2} \right], \tag{2.23}$$

$$N_i = \frac{\exp\{\log f(x_i; \theta, \boldsymbol{\mu})\}}{\exp\{\log f(x_i; G), \boldsymbol{\mu}\}}.$$

We also need the first derivative of log-likelihood respect to $\boldsymbol{\mu}$ when apply the BFGS method. That is,

$$\frac{\partial l(\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = \sum_{i=1}^{n} \left\{ \frac{1}{\log(\sum_{k=1}^{K} \pi_k Z_{ik})} \sum_{k=1}^{K} \left[\pi_k Z_{ik} \frac{\partial \log f(x_i; \theta_k, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}} \right] \right\}, \tag{2.24}$$

$$Z_{ik} = \exp\{\log f(x_i; \theta_k, \boldsymbol{\mu})\}.$$

Therefore, every critical component in the CNM-MS algorithm are all in terms of differentiation of log-likelihood respect with θ or $\boldsymbol{\mu}$. Once identified these components for mixture model, the CNM-MS algorithm can be applied easily. We also use these log-likelihood expressions in this thesis and programming implementation.

2.5 Scale mixtures of normal distributions

The main aim of the research in this thesis is to use semiparametric mixture models to establish scale mixtures of normal distributions and its possible applications.

The scale mixtures of normal distributions are the probability distributions of the form $X = \frac{Z}{\sqrt{\Theta}}$, where Z is normally distributed with mean $\boldsymbol{\mu}$ and variance 1, and $\Theta \in (0, \infty)$ has continuous or discrete provability distribution G. Thus, we can treat scale mixtures of normal distribution as semiparametric mixture model with nonparametric mixing distribution G and parametric finite dimensional vector $\boldsymbol{\mu}$.

We implement the CNM-MS algorithm to find the semiparametric maximum likelihood estimate of scale mixtures of normal distributions, and present some possible applications. We do not have to worry about the number of components in the mixture model since we use the CNM-MS algorithm, as deciding the number of components is the critical issue in traditional mixture model study.

2.6 Summary

This chapter examined the background of this thesis. The aim of the research in this thesis is study scale mixtures of normal distributions, and their possible applications.

We introduced how to find nonparametric maximum likelihood estimates of mixture models by using the CNM algorithm, and extended them to locate semiparametric maximum likelihood estimates by using CNM-based algorithms. We only used CNM-MS in this thesis because of least computational time.

The advantage of CNM-based algorithms is avoiding judging the number of components for mixture models, which is critical issue for most of present literature about them.

Chapter 3

Scale Mixtures of Univariate Normal Distributions

This chapter describes how to compute semiparametric MLEs for scale mixtures of normal distributions (SMN) and their possible applications. We introduce SMN in Section 3.1, and Section 3.2 gives a basic definition of SMN. Section 3.3 demonstrates how semiparametirc MLE for SMN can be estimated by the CNM-MS algorithm. Section 3.4 discuss some computational issues.

Section 3.5 discusses possible application of SMN with no covariates. We perform model fitting with selected heavy-tailed simulated data, modeling log returns and estimating Value-at-Risk (VaR) which is widely used in financial risk management. Section 3.6 discusses another possible application with covariates, linear robust regression, and conduct a simulation to study the performance. Finally, Section 3.7 gives some concluding remarks.

3.1 Introduction

The scale mixture of normal distributions (SMN) is the probability distribution of the from $X = \frac{Z}{\sqrt{\Theta}}$, where Z is normally distributed with expected value μ and

variance 1, and $\Theta \in (0, \infty)$ has continuous or discrete probability distribution G.

It has been proved that many traditional, continuous and symmetric distributions can be expressed in this form, such as discrete mixtures of normal distributions, contaminated normal distribution, t-distribution, logistic distribution, Laplace distribution, stable family and normal distribution itself; see, for example, Andrews and Mallows (1974) and West (1987). Robustness and outlier models using scale mixtures of normal distributions are also studied by West (1984).

3.2 Basic Definition

Suppose $\Theta \in (0, \infty)$ has continuous or discrete probability distribution G, and X is a random variable with scale mixtures of normal distributions with density

$$f(x; \mu, G) = \int \phi(x; \mu, \theta) \mathrm{d}G(\theta), \tag{3.1}$$

where μ is the mean, and $\phi(.)$ is the probability density function for normal distribution. The cumulative distribution function of X is

$$F(x; \mu, G) = \int \Phi(x; \mu, \theta) \mathrm{d}G(\theta), \tag{3.2}$$

where $\Phi(.)$ is the cumulative distribution of normal distribution. The variance of X is the same as expected value of Θ,

$$\mathrm{VAR}(X) = \int \theta \mathrm{d}G(\theta). \tag{3.3}$$

Many univariate, continuous and symmetric distributions can also be expressed in scale mixtures of normal distributions, as we introduced in last section. We can obtain different distributions by changing G, the characterization result for some of these traditional distributions is described briefly as follows. Detailed proofs can be found in Andrews and Mallows (1974).

t-distribution

The t-distribution with degrees of freedom ν has density in Pearson Type VII:

$$f(x) = \frac{1}{\sqrt{\nu}B(\frac{\nu}{2}, \frac{1}{2})}\left[1 + \frac{x^2}{\nu}\right]^{-\frac{\nu+1}{2}}, \tag{3.4}$$

where B is the Beta function. It can be expressed as scale mixtures of normal distributions if G has density

$$G(\theta) = \frac{\nu}{2}\frac{(\frac{1}{2}\nu\theta)^{\frac{\nu}{2}-1}}{\Gamma(\frac{\nu}{2})}\exp\left(-\frac{1}{2}\nu\theta\right). \tag{3.5}$$

That is, $\frac{1}{2}\nu\Theta$ has a gamma distribution with shape parameter $\frac{\nu}{2}$.

Laplace distribution

The Laplace distribution has density

$$f(x) = \frac{1}{2}\exp(-|x|). \tag{3.6}$$

It can be expressed as scale mixtures of normal distributions if G has density

$$G(\theta) = \frac{1}{2\theta^2}\exp\left(-\frac{1}{2\theta}\right), \tag{3.7}$$

so that $\frac{1}{2\Theta}$ has an exponential distribution.

Logistic distribution

The logistic distribution has density

$$f(x) = \frac{\exp(-x)}{(1+\exp(-x))^2}. \tag{3.8}$$

It can be expressed as scale mixture of normal distributions if G has density

$$G(\theta) = 2\sum_{i=1}^{\infty}(-1)^{i-1}\frac{i}{\theta}^2\exp\left(-\frac{i^2}{2\theta}\right), \tag{3.9}$$

that is, $\frac{1}{2\sqrt{\Theta}}$ has the asymptotic distribution of Kolmogorov distance statistic.

3.3 Semiparametirc MLE

In this section, we introduce how to compute semiparametric MLE for scale mixtures of normal distributions.

3.3.1 Discrete G

As we discussed in Section 2.2, we can just consider discrete G when computing a semiparametric MLE for mixture models. Thus, we can rewrite scale mixtures of normal distribution density (3.1) in discrete form with unknown K-component normal mixtures

$$f(x;\mu,\boldsymbol{\pi},\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \phi(x;\mu,\theta_k) \tag{3.10}$$

where the discrete G consists of $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_K) \in (0,\infty)$ with mixing proportion $\boldsymbol{\pi} = (\pi_1,\ldots,\pi_K) \in (0,1)$ respectively. $\boldsymbol{\theta}$ is also the variance for each normal mixture, and μ is the common mean. The cumulative distribution function of X can also be written in discrete form

$$F(x;\mu,\boldsymbol{\pi},\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \Phi(x;\mu,\theta_k) \tag{3.11}$$

and the variance of X is

$$\mathrm{VAR}(X) = \sum_{k=1}^{K} \pi_k \theta_k \tag{3.12}$$

3.3.2 Formulation

Maximum likelihood estimation (MLE) of scale mixtures of normal distributions $(\mu,\boldsymbol{\pi},\boldsymbol{\theta})$ can be viewed as a semiparametric problem. It consist of a scalar parametric component μ and nonparametric components $(\boldsymbol{\pi},\boldsymbol{\theta})$ with unknown dimension K.

As already discussed in Section 2.3, an excellent, reliable, and fast algorithm for computing MLE in semiparametric mixture models was introduced by Wang (2010). We can use the algorithm, maximization by modifying the support set

with the CNM method (CNM-MS), to compute semiparametirc MLE for scale mixtures of normal distributions.

Firstly, we denote the probability density function of scale mixtures of normal distributions (3.10), that is

$$\begin{aligned} f(x;\mu,\boldsymbol{\pi},\boldsymbol{\theta}) &= \sum_{k=1}^{K} \pi_k \phi(x;\mu,\theta_k) \\ &= \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{2\pi\theta_k}} \exp\left(-\frac{(x-\mu)^2}{2\theta_k}\right) \end{aligned} \tag{3.13}$$

For computing semeparametric MLE for scale mixtures of normal distributions in the CNM-MS framework, we need to formulate our problem. We need the following expressions to compute the gradient function, find local maxima of the gradient function and update scalar parametric component μ. The density function of normal mixtures component is

$$\phi(x;\mu,\theta) = \frac{1}{\sqrt{2\pi\theta}} \exp\left(-\frac{(x-\mu)^2}{2\theta}\right), \tag{3.14}$$

and $\log\phi(.)$ is

$$\begin{aligned} l(x;\mu,\theta) &= \log\phi(x;\mu,\theta) \\ &= \log\left\{\frac{1}{\sqrt{2\pi\theta}} \exp\left(-\frac{(x-\mu)^2}{2\theta}\right)\right\} \\ &= -\frac{1}{2}\log 2\pi - \frac{1}{2}\log\theta - \frac{(x-\mu)^2}{2\theta}. \end{aligned} \tag{3.15}$$

The partial derivative of $l(.)$ with respect to μ is

$$\begin{aligned} \frac{\partial l}{\partial \mu} &= \frac{\partial}{\partial \mu}\left(-\frac{1}{2}\log 2\pi - \frac{1}{2}\log\theta - \frac{(x-\mu)^2}{2\theta}\right) \\ &= \frac{\partial}{\partial \mu}\left(-\frac{(x-\mu)^2}{2\theta}\right) \\ &= \frac{x-\mu}{\theta}. \end{aligned} \tag{3.16}$$

The partial derivative of $l(.)$ with respect to θ is

$$\begin{aligned}\frac{\partial l}{\partial \theta_k} &= \frac{\partial}{\partial \theta}\left(-\frac{1}{2}\log 2\pi - \frac{1}{2}\log\theta - \frac{(x-\mu)^2}{2\theta}\right) \\ &= \frac{\partial}{\partial \theta_k}\left(-\frac{1}{2}\log\theta - \frac{(x-\mu)^2}{2\theta}\right) \\ &= -\frac{1}{2\theta} + \frac{(x-\mu)^2}{2\theta^2}. \end{aligned} \tag{3.17}$$

The second partial derivative of $l(.)$ with respect to θ is

$$\begin{aligned}\frac{\partial^2 l}{\partial \theta^2} &= \frac{\partial}{\partial \theta}\left(-\frac{1}{2\theta} + \frac{(x-\mu)^2}{2\theta^2}\right) \\ &= \frac{1}{2\theta^2} - \frac{(x-\mu)^2}{\theta^3}. \end{aligned} \tag{3.18}$$

Using the CNM-MS algorithm with above expressions, the semiparametric MLE for scale mixtures of normal distributions can be computed with appropriate range for $\boldsymbol{\theta}$.

3.4 Miscellanea

3.4.1 Choice of θ

In scale mixtures of normal distributions, $\boldsymbol{\theta}$ is positive and independent, and it is also the variance for each normal mixture component. The constraint of $\boldsymbol{\theta}$ is $\boldsymbol{\theta} > 0$. Computationally, we need to specify the range of $\boldsymbol{\theta}$. It is very difficult to choose the range of $\boldsymbol{\theta}$ since it is unbounded. Hathaway (1985) suggests a linear inequality constraint

$$\frac{\theta_{\max}}{\theta_{\min}} \geq c > 0 \tag{3.19}$$

Unfortunately, there is no suitable parametric method to decide c at the present time. Therefore we choose an arbitrary value $c = 10^{16}$ in the CNM-MS implementation since it works practically well.

3.4.2 Quantile estimation

We may need to compute the quantile of scale mixtures of normal distributions in some applications, such as Value-at-Risk estimation that we will discuss later. Unfortunately, there is no closed-form expression of the quantile function of scale mixtures of normal distributions, $F^{-1}(.)$. Therefore we use the bisection method to approximate $F^{-1}(.)$.

Denote X as a random variable with scale mixtures of normal distributions. $F(.)$ is cumulative distribution function of X and q quantile of X can be computed in following steps,

1. choose initial value of (a, b) with some small and large value, say, $(a, b) = (-10\text{VAR}(X), 10\text{VAR}(X))$.
2. let $m = \frac{a+b}{2}$.
3. if $q < F(m)$, $a' = a$, $b' = m$.
   if $q > F(m)$, $a' = m$, $b' = b$.
   if $|q - F(m)| < 10^{-15}$, stop.
4. Update (a, b) with (a', b'), go to step 2.

Computation of quantile functions using the bisection method can approximate $F^{-1}(.)$ very quickly, and have extremely good performance.

3.5 Application with no covariates

In this section, we introduce two possible applications for scale mixtures of normal distributions with no covariates. Firstly we generate four simulated heavy-tailed datasets and study the performance of our approach. Secondly we study three real financial datasets, model log-returns and estimate Value-at-Risk (VaR).

3.5.1 Simulated Data

Four heavy-tailed simulated datasets have been generated by different distributions. They are t-distribution with degrees of freedom 6, standard logistic distribution, standard Laplace distribution, contaminated normal distribution with 30 percent contamination, in which 70 percent of data has $N(5, 1^2)$, and 30 percent has $N(5, 10^2)$. All of four datasets has sample size $5,000$ and mean 5.

Result

The computation results are given in Table 3.1, histogram with fitting density that can be found in Figure 3.1.

For data with $t_{\nu=6}$ distribution, it is the heavy-tailed data with sample kurtosis 2.223. We fit a scale mixtures of normal distributions with four components. There is one normal component has large variance (10.226) with proportion 0.008, which indicates the model fits the heavy-tailed part of data.

For the data with a standard logistic distribution, it is the heavy-tailed data with sample kurtosis 1.128. We fit scale mixtures of normal distributions with three components. There is one normal component has a large variance (11.634) with proportion 0.041, which also indicates the model fits the heavy-tailed part of data.

For the data with a standard Laplace distribution, it is the heavy-tailed data with highest sample kurtosis 2.762. We fit scale mixtures of normal distributions with three components. Since it has highest sample kurtosis (2.762), the proportion of normal mixing component indicate the heavy-tailed part of data (0.076) is higher than the $t_{\nu=6}$ distribution (0.008) and standard logistic distribution (0.041). They also have a mixing component with very small variance (0.338) and proportion 0.391, indicating the center peak of Laplace distribution.

For the data with 30% contaminated normal distribution, it is the heavy-tailed data with sample kurtosis 6.256. We fit a scale mixtures of normal distri-

			MLE			Sample		
Distribution	$\hat{\boldsymbol{\pi}}$	$\hat{\boldsymbol{\theta}}$	mean	var.	kurt.	mean	var.	kurt.
$t_{\nu=6}$	0.254	0.520	4.978	1.538	2.226	4.980	1.539	2.223
	0.576	1.220						
	0.162	3.813						
	0.008	10.226						
Logistic	0.299	1.262	5.013	3.315	1.131	5.025	3.315	1.128
	0.660	3.722						
	0.041	11.634						
Laplace	0.391	0.338	4.993	2.009	2.765	4.987	2.009	2.762
	0.533	2.400						
	0.076	7.900						
Con.Normal	0.704	1.02	5.004	33.074	6.256	4.965	33.079	6.248
	0.296	109.322						

Table 3.1: Model fitting

butions with two components. The fitted model indicates 70.4 percent of data has $N(5, 1.02)$, and 29.6 percent has $N(5, 109.32)$, which is very close to true value ($N(5, 1)$ and $N(5, 100)$).

In general, semiparametric MLE for scale mixtures of normal distributions using the CNM-MS algorithm has very good performance for different heavy-tailed distributions. Even if the data has really high kurtosis (6.248 or 2.762), it fits the data well. The estimated mean are all very close to 5, estimated variance and kurtosis are almost the same as sample variance and kurtosis. In Figure 3.1, fitted densities all fit datasets very well.

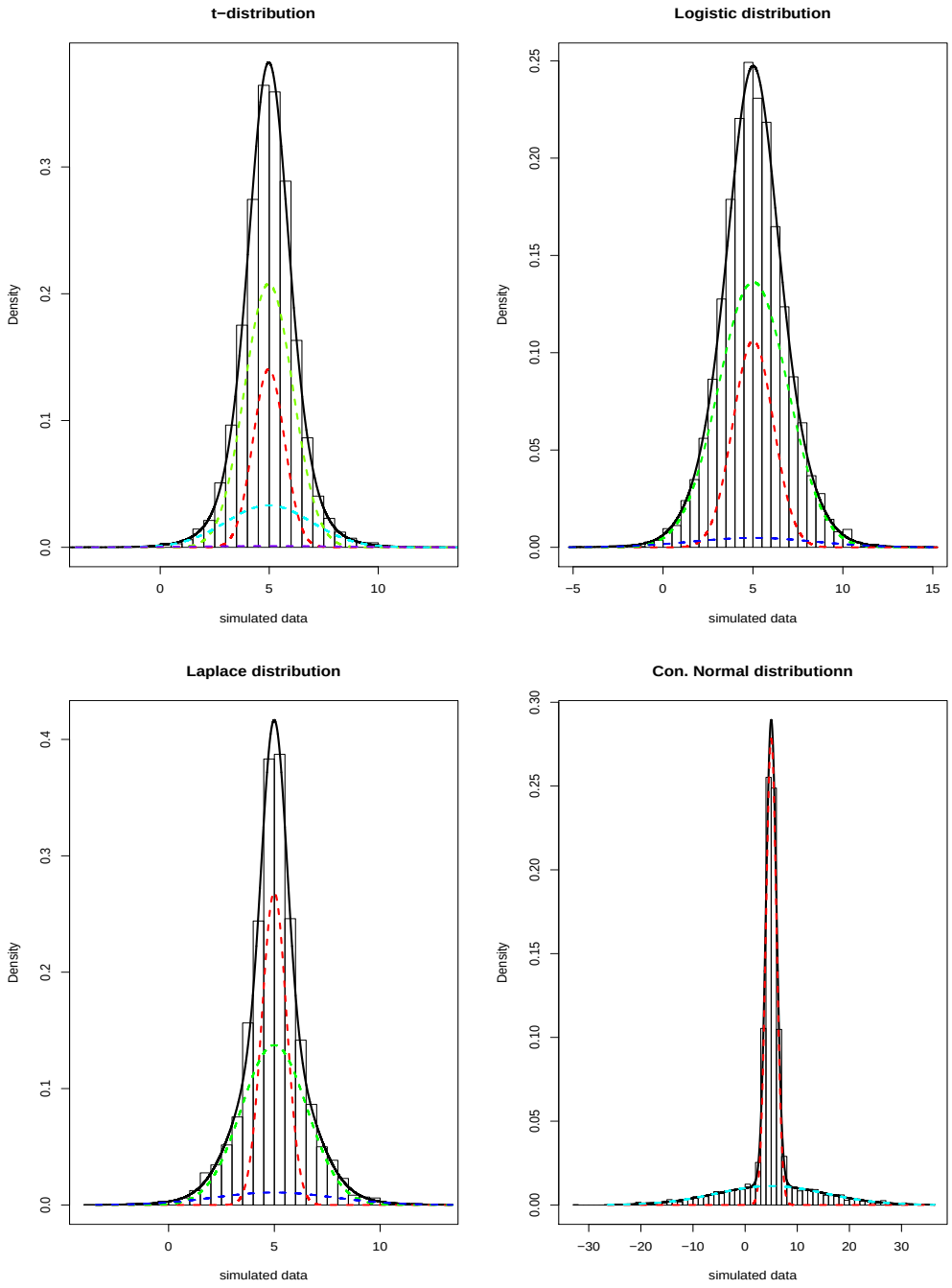

Figure 3.1: Fitted heavy-tailed simulated data

3.5.2 Modeling log returns and VaR Estimation

In this section, we consider a SMN VaR model, which assume that the loss-distribution has scale mixtures of normal distributions, and estimate Value-at-Risk (VaR) for three chosen datasets. The most common way to model the loss distribution is assume it is normal distribution, and alternative way is to assume it is noncentral t-distribution. Therefore we also fit a normal VaR model and t VaR model as benchmark. Three financial time series data have been considered:

- Nikkei 225, the stock market index for the Tokyo Stock Exchange
- CAC40, the stock market index for Paris Bourse
- GBP/JPY, British Pound to Japanese Yen exchange rate

Name	Observation	Time Period	Frequency	Source
Nikkei 225	4924	01/01/1989 - 01/01/2009	Daily	Yahoo! Finance
CAC40	4755	01/01/1989 - 01/01/2009	Daily	Yahoo! Finance
GBP/JPY	5048	01/01/1989 - 01/01/2009	Daily	Pacific Exchange Rate Service

Table 3.2: Financial data

These data can be found in Yahoo! Finance (`http://finance.yahoo.com`) and Pacific Exchange Rate Service (`http://fx.sauder.ubc.ca/data.html`).

Modeling log returns

In finance, returns is the ratio of money profit or loss, it measures the rate of change for an investment or an asset. Most of financial studies interest on returns analysis. Financial time series data to be investigated usually are the price of stock, a stock market index or an exchange rate. There are many types of returns, such as net returns, gross returns and internal rate of returns.

The most commonly used is logarithmic returns, also known as continuously compounded returns. Denote S_t be the price of an asset at time t with sample size n, log returns X_t at time t is defined as

$$X_t = \log \frac{S_t}{S_{t-1}}, \qquad t \in \{1, \ldots, n\}. \tag{3.20}$$

Profit-and-loss distribution for a portfolio (mix of investments) is usually evaluated in risk management. If our portfolio has only one asset X, the profit-and-loss distribution is X itself. Since risk management usually studies large losses, we focus on the loss distribution only, which is the distribution of random variable

$$L = -X. \tag{3.21}$$

Therefore we can focus on large losses, at the upper tail of the loss distribution.

Traditional techniques modeling the loss distribution assume it is normal distribution. Financial return data usually are leptokurtic (heavy-tailed), therefore the assumption of normality may not satisfied. We could assume it is a noncentral t-distribution to resolve the heavy-tailed problem. Our new approach is to assume loss distribution has scale mixtures of normal distributions and compare the performance.

Estimating VaR

Value-at-Risk (VaR) is probably the most widely used measurement in risk management. VaR is defined as a maximum possible loss of a portfolio that does not exceed the given probability $\alpha \in (0, 1)$. Denoting the loss distribution L, VaR is formally defined as

$$\mathrm{VaR}_\alpha = \inf\{u \in \mathbb{R} : P(L > u) \leq 1 - \alpha\}. \tag{3.22}$$

Therefore, VaR is just the quantile of the loss distribution L

$$\mathrm{VaR}_\alpha = F_L^{-1}(\alpha), \tag{3.23}$$

where $F_L^{-1}(.)$ is the quantile function of loss distribution L.

We estimate VaR for three chosen financial datasets at level $\alpha = (0.95, 0.99)$ in following steps,

1. calculate log returns X for the price of asset S.
2. calculate loss distribution $L = -X$.
3. fitting L by normal distribution, noncentral t-distribution and scale mixture of normal distribution.
4. estimate $\text{VaR}_{0.95}$ and $\text{VaR}_{0.99}$ for three model, which is also the 95% and 99% quantile of loss distribution.

Fitting scale mixtures of normal distributions is implemented by the CNM-MS algorithm, and fitting normal distributions and noncentral t-distributions by maximum likelihood.

Kupiec Test

Kupiec (1995) proportion of failure test that can measure the accuracy of the VaR estimation. It determines whether the proportion of observation below the estimated VaR is consistent with the given probability α. Denote the total number of observation N, number of observation below VaR n, the null hypothesis of Kupiec test is

$$\text{H}_0 : \alpha = \frac{n}{N}.$$

If the null hypothesis is true, n has a binomial distribution. The Kupiec test performs a likelihood ratio test, the test statistic is

$$\text{LR} = -2\log\left(\left(\frac{1-\alpha}{1-\frac{n}{N}}\right)^{N-n}\left(\frac{\alpha}{\frac{n}{N}}\right)^{n}\right). \tag{3.24}$$

LR is asymptotically χ^2 distributed with one degree of freedom. If LR is greater than the critical value, we will reject null hypothesis. Details can be found in Kupiec (1995).

Result

Table 3.3 shows a numerical summary of log returns of these three financial datasets, Figure 3.2 shows the time series plot for raw data and log returns. All three dataset are centered at 0, and they are all heavy-tailed (high kurtosis). The Nikkei 225 and CAC40 data are very similar, they have similar standard deviation (0.015 and 0.014), kurtosis (5.595 and 5.075) and skewness (0.027 and 0.047). GBP/JPY data has a very small standard deviation (0.008) compared to the other two, it also has the highest kurtosis (7.205) and it has slightly positive skew (skewness is 0.592).

Data	median	mean	sd	kurtosis	skewness
Nikkei 225	8.205×10^{-5}	2.494×10^{-4}	0.015	5.929	0.027
CAC40	-3.326×10^{-4}	-1.185×10^{-4}	0.014	5.075	0.047
GBP/JPY	-1.995×10^{-4}	1.049×10^{-4}	0.008	7.205	0.592

Table 3.3: Numerical summary of log returns

Model fitting results can be found in table 3.4, model AIC, VaR estimation and Kupiec test results are presented in table 3.5. Figure 3.2-3.8 also perform three plots for each dataset: Time series plot for raw data and log returns, density plot, Q-Q plot for each model and VaR Estimation.

Nikkei 225 For Nikkei 225 data, the normal distribution seems to have a poor performance. The Normal Q-Q plot shows it is clearly not normally distributed, and estimated VaR for both level 0.95 and 0.99 have extremely strong evidence (p-value=0.0039 and 0.0004) against null hypothesis at 5% significance level for Kupiec test.

	SMN			t		
Data	$\hat{\mu}$	$\hat{\boldsymbol{\pi}}$	$\sqrt{\hat{\boldsymbol{\theta}}}$	$\hat{\mu}$	$\hat{\nu}$	$\hat{\lambda}$
Nikkei 225	7.773×10^{-5}	0.211	0.074	8.391×10^{-5}	3.871	0.011
		0.647	0.115			
		0.131	0.159			
		0.010	0.238			
CAC40	-3.406×10^{-4}	0.258	0.087	3.661×10^{-4}	4.015	0.010
		0.474	0.104			
		0.225	0.133			
		0.043	0.191			
GBP/JPY	-3.267×10^{-5}	0.800	0.073	1.948×10^{-4}	3.870	0.005
		0.194	0.112			
		0.007	0.186			

	Normal	
Data	$\hat{\mu}$	$\hat{\sigma}$
Nikkei 225	2.494×10^{-4}	0.015
CAC40	-1.185×10^{-4}	0.014
GBP/JPY	1.049×10^{-4}	0.008

Table 3.4: Model fitting

The noncentral t-distribution model for Nikkei 225 has good performance. Estimated degrees of freedom is 3.871, and $\hat{\lambda} = 0.011$. Points of the Q-Q plot are approximately lie on the line $y = x$, the density plot also fits the data well. Estimated VaR for both level 0.95 and 0.99 have no evidence (p-value=1 and 0.9398) against null hypotheses at 5% significance level for the Kupiec test.

The scale mixtures of normal distributions model for Nikkei 225 also has very good performance. The model has four components of normal mixtures, there is a small standard deviation (0.074) with proportion 0.211 indicating a high peak at the center, and there is a large standard deviation (0.238) with proportion 0.01, which indicates heavy-tailed data. Points of the Q-Q plot are approximately lie on the line $y = x$, the density plot also fits the data very well, but the center peak is higher than the noncentral t-distribution model.

			$\alpha = 0.95$		$\alpha = 0.99$	
Data	Model	AIC	VaR	p-value	VaR	p-value
Nikkei 225	SMN	-27894.22	0.0239	1.0000	0.0409	0.6527
	t	-37868.49	0.0235	1.0000	0.0415	0.9398
	Normal	-27136.37	0.0255	0.0039	0.0360	0.0004
CAC40	SMN	-27797.10	0.0210	0.5132	0.0379	0.2119
	t	-27766.38	0.0217	1.0000	0.0378	0.2119
	Normal	-27082.37	0.0229	0.0197	0.0325	9.932×10^{-7}
GBP/JPY	SMN	-35641.21	0.0115	0.2985	0.0217	0.4944
	t	-35624.63	0.0119	1.0000	0.0210	0.1386
	Normal	-34760.12	0.0128	0.0231	0.0181	3.037×10^{-7}

Table 3.5: AIC, VaR Estimation and Kupiec Test p-value

The estimated VaR for both level 0.95 and 0.99 have no evidence (p-value=1 and 0.6527) against the null hypotheses at 5% significance level for the Kupiec test. Estimated SMN $\text{VaR}_{0.95}$ (0.0239) is higher than t $\text{VaR}_{0.95}$ (0.0235), but estimated $\text{VaR}_{0.99}$ (0.0409) is lower than t $\text{VaR}_{0.99}$ (0.0415).

CAC40 For CAC40 data, the normal distribution seems to have poor performance. Normal Q-Q plot shows it is clearly not normally distributed, and estimated VaR for both level 0.95 and 0.99 have extremely strong evidence (p-value=0.0197 and 9.932×10^{-7}) against null hypothesis at 5% significance level for the Kupiec test.

The noncentral t-distribution model for CAC40 has good performance. Estimated degrees of freedom is 4.015, and $\hat{\lambda} = 0.01$. Points of the Q-Q plot are approximately lie on the line $y = x$, density plot also fits the data well. Estimated VaR for both level 0.95 and 0.99 have no evidence (p-value=1 and 0.2119) against null hypotheses at 5% significance level for the Kupiec test.

The scale mixtures of normal distributions model for CAC40 also has very good performance. The model has four components of normal mixtures, there is a small standard deviation (0.087) with proportion of 0.258 indicating the high peak at center, and there is a large standard deviation (0.191) with proportion 0.043,

which indicates heavy-tailed data. Points of the Q-Q plot are approximately lie on the line $y = x$, density plot also fits the data very well, but the center peak is lower than the noncentral t-distribution model.

The estimated VaR for both level 0.95 and 0.99 have no evidence (p-value=0.5132 and 0.2119) against the null hypotheses at 5% significance level for Kupiec test. Estimated SMN $\text{VaR}_{0.95}$ (0.0210) is lower than estimated t $\text{VaR}_{0.95}$ (0.0217), but estimated SMN $\text{VaR}_{0.99}$ (0.0379) is almost the same as t $\text{VaR}_{0.99}$ (0.0378).

GBP/JPY For GBP/JPY data, normal distribution seems to have performed poorly. Normal Q-Q plot shows it is clearly not normally distributed, and estimated VaR for both level 0.95 and 0.99 have extremely strong evidence (p-value=0.0231 and 3.037×10^{-7}) against null hypothesis at 5% significance level for the Kupiec test.

The noncentral t-distribution model for GBP/JPY has good performance. Estimated degrees of freedom is 3.87 which is similar to Nikkei 225, but $\hat{\lambda} = 0.005$ is much smaller. Points of the Q-Q plot are approximately lie on the line $y = x$ except tails, since GBP/JPY is slightly positive skewed. Estimated VaR for both level 0.95 and 0.99 have no evidence (p-value=1 and 0.1836) against null hypotheses at 5% significance level for the Kupiec test.

The scale mixtures of normal distributions model for CAC40 also has very good performance. The model has three components of normal mixture, there is a large standard deviation (0.186) with proportion 0.007, indicating heavy-tailed part of data. Points of the Q-Q plot are approximately lie on the line $y = x$, density plot also fits the data very well except center peak. The center peak is a little bit lower than the noncentral t-distribution model.

The estimated VaR for both level 0.95 and 0.99 have no evidence (p-value=0.2985 and 0.4944) against the null hypotheses at 5% significance level for Kupiec test. Estimated SMN $\text{VaR}_{0.95}$ (0.0115) is lower than estimated t $\text{VaR}_{0.95}$ (0.0119), but estimated SMN $\text{VaR}_{0.99}$ (0.0217) is higher than t $\text{VaR}_{0.99}$ (0.0210).

In conclusion, the normal model is lacking, and estimated VaR are overestimated at level 0.95, underestimated at level 0.99. The noncentral t-distribution model and scale mixtures of normal distributions model both have good performance, they fit the data reasonably well, and both have similar VaR estimation, no evidence against null hypothesis at 5% significance level for Kupiec Test. However, SMN model has lowest AIC for all three datasets, thus we claim that SMN is the best model.

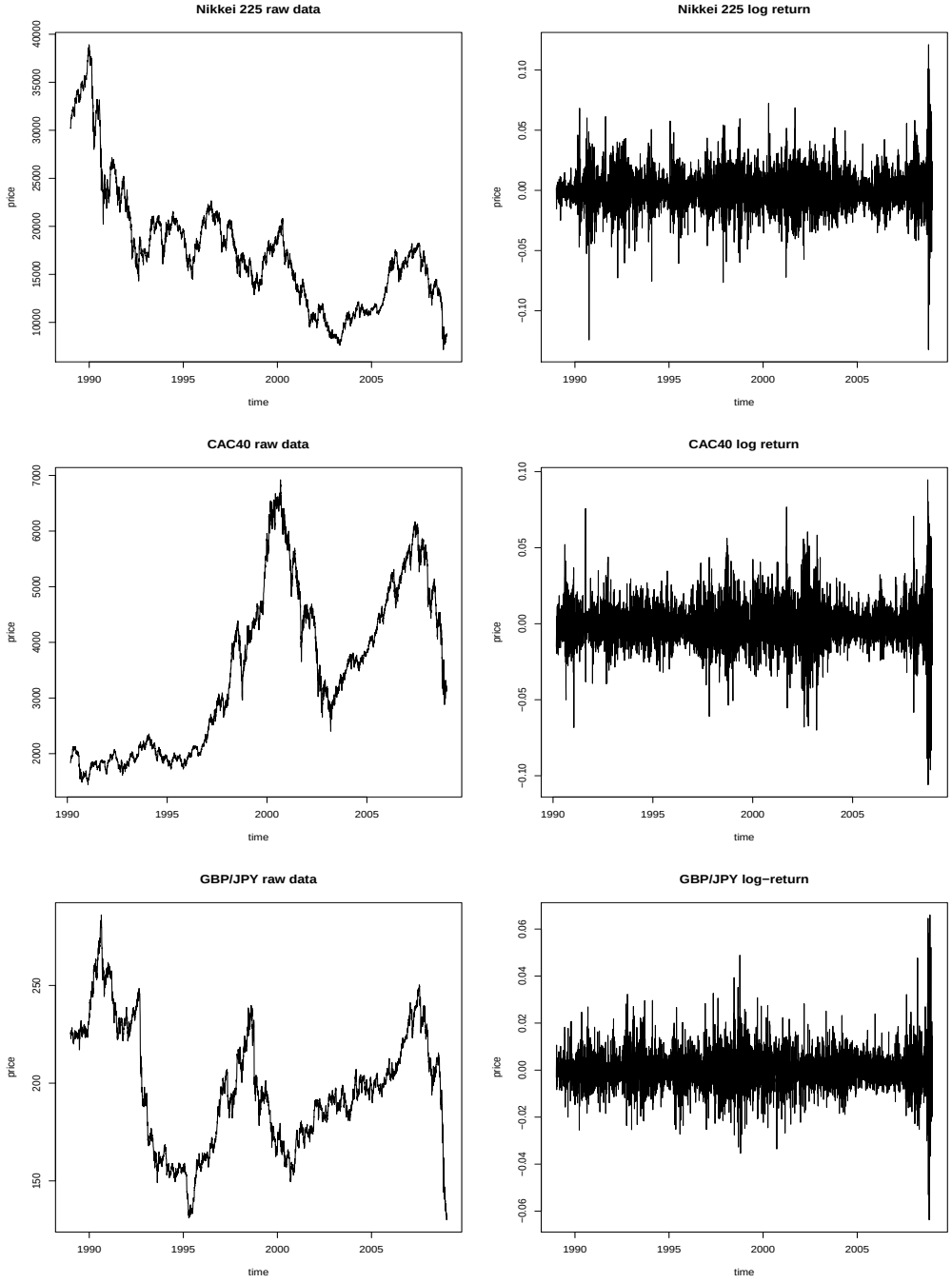

Figure 3.2: Financial data overview

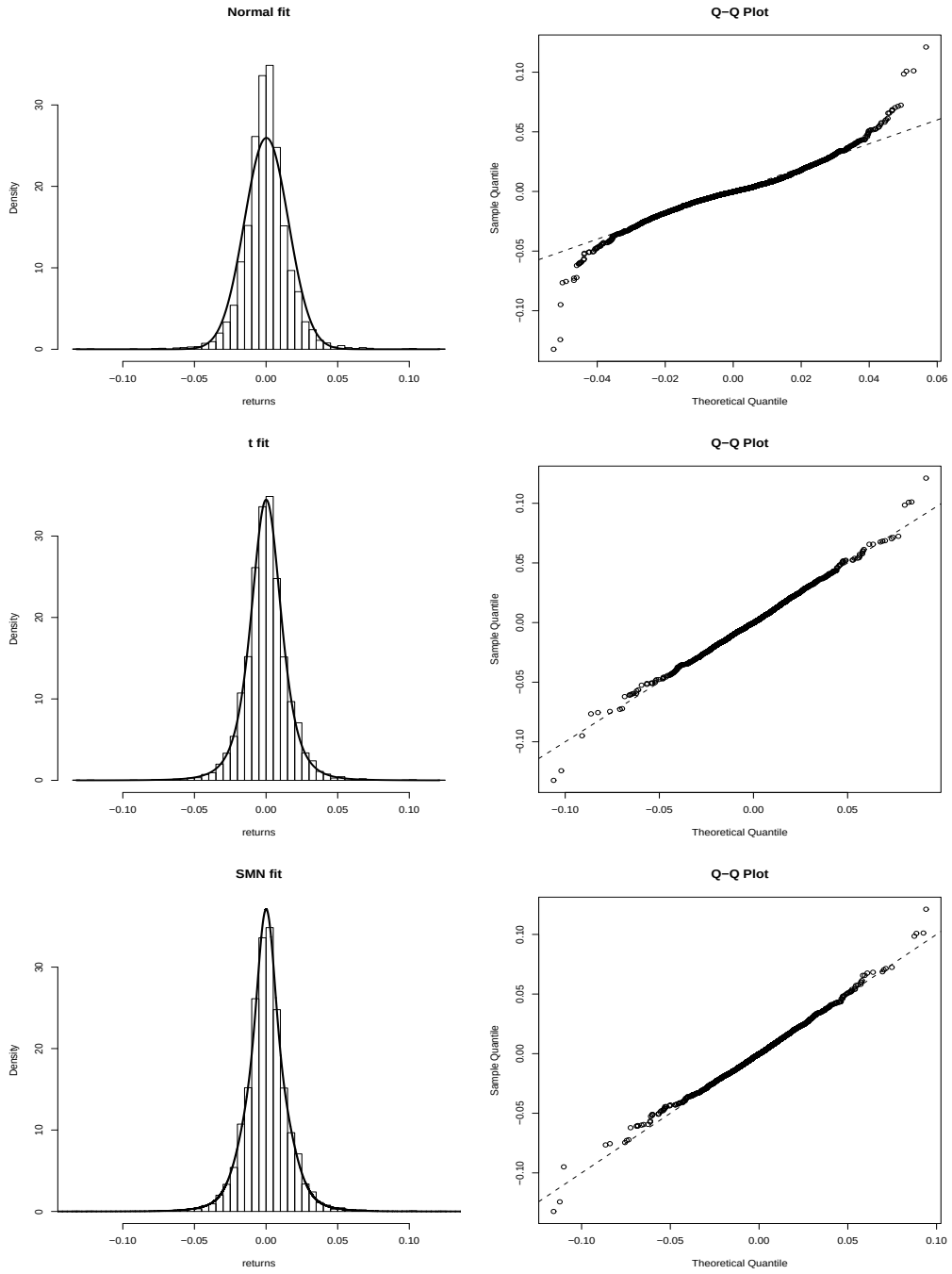

Figure 3.3: Nikkei 225 model Density and Q-Q plot

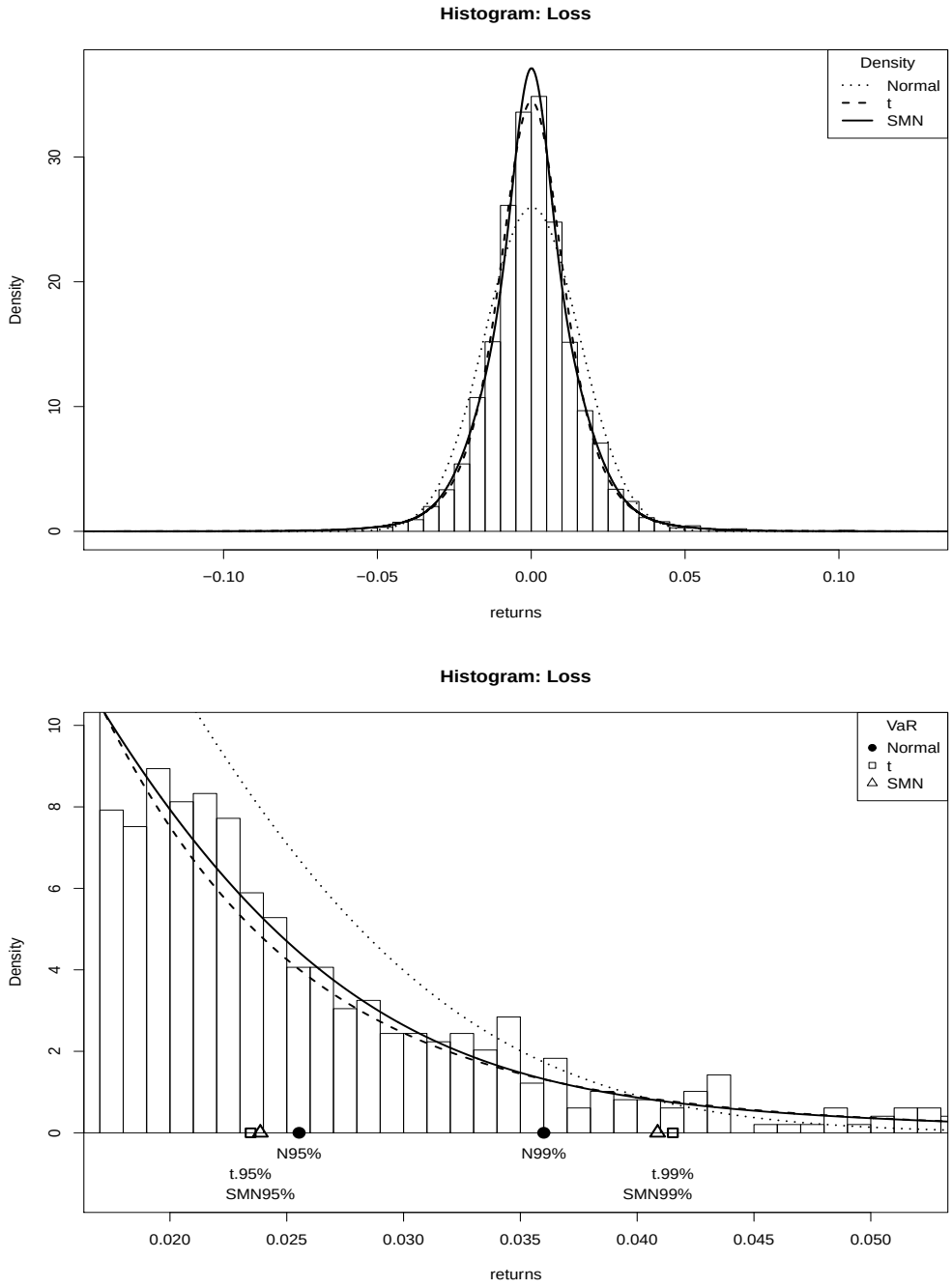

Figure 3.4: Nikkei 225 VaR estimation

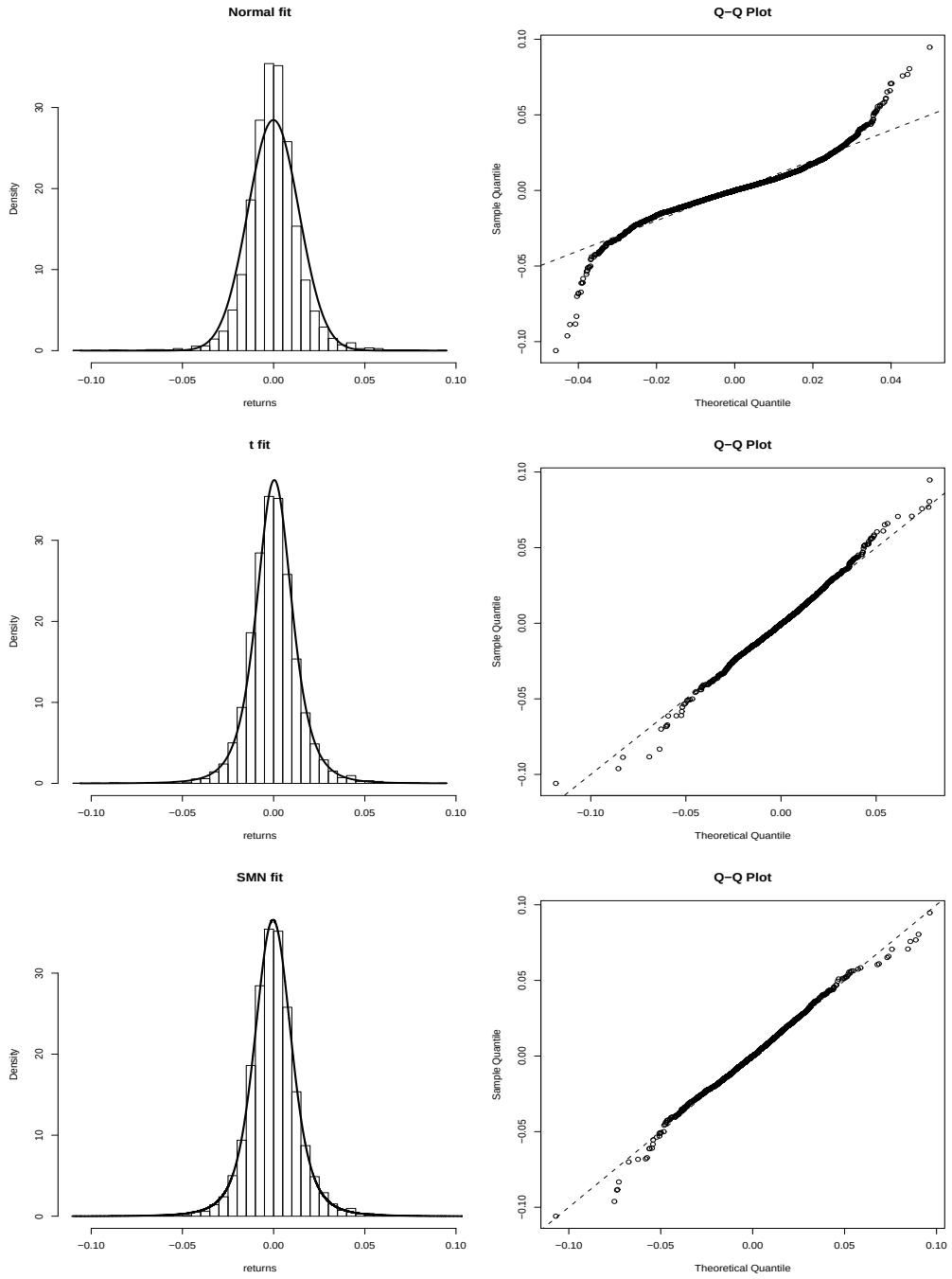

Figure 3.5: CAC40 model Density and Q-Q plot

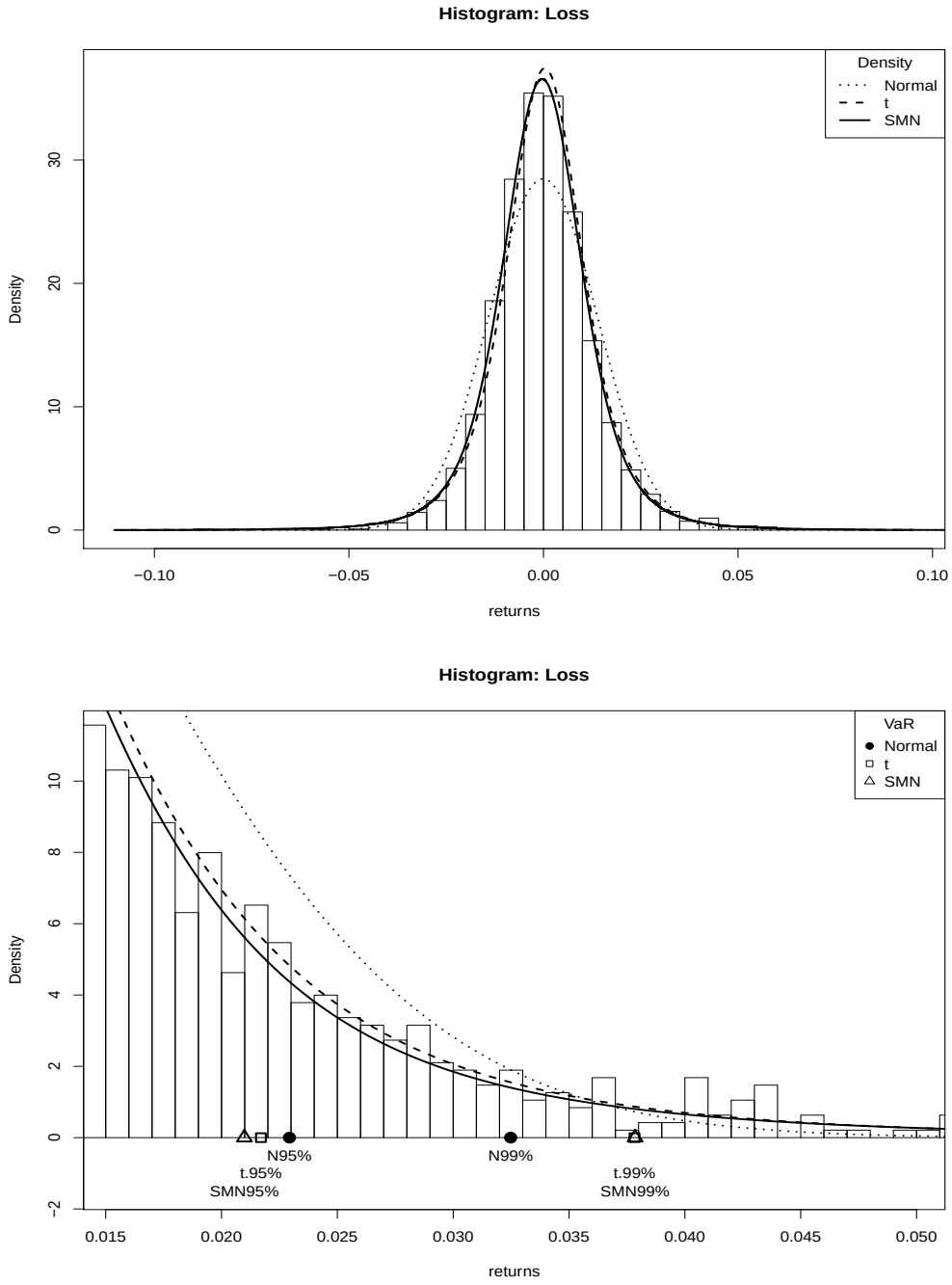

Figure 3.6: CAC40 VaR estimation

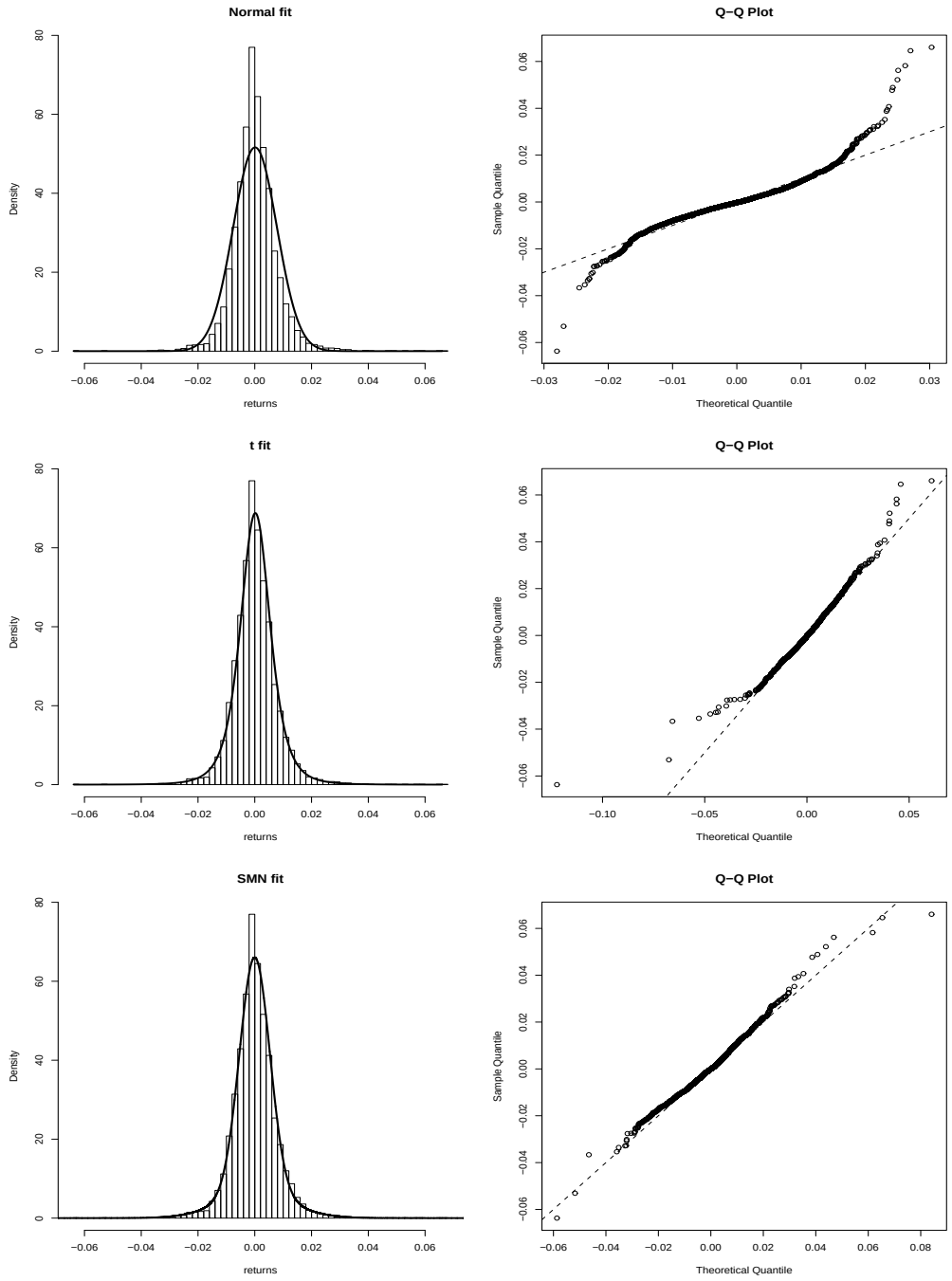

Figure 3.7: GBP/JPYmodel Density and Q-Q plot

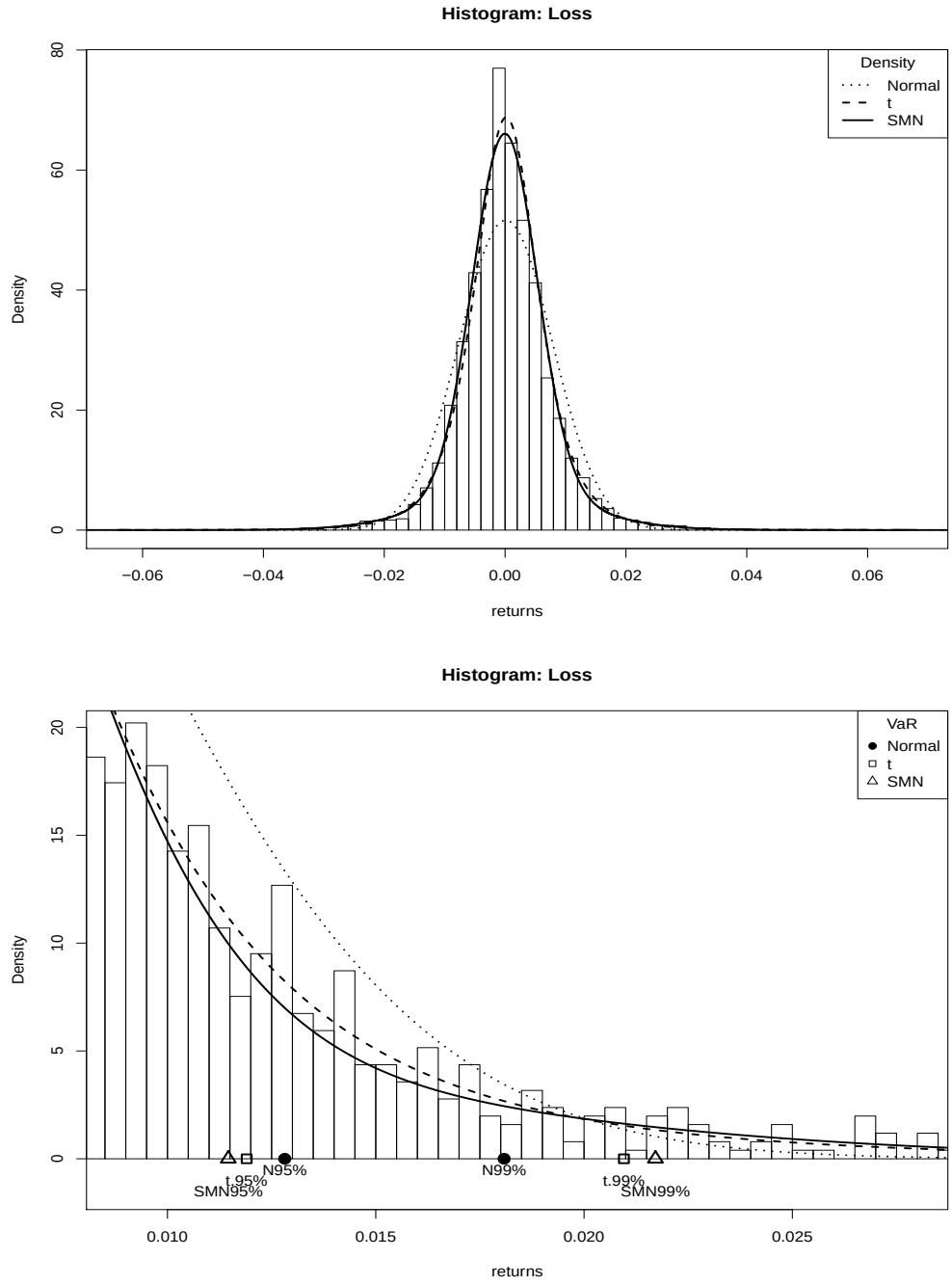

Figure 3.8: GBP/JPY VaR estimation

3.6 Linear robust semiparametric regression

3.6.1 Introduction

Regression is a statistical method to explore relationships between response variable $\mathbf{y}$ and explanatory variables $\mathbf{X}$. Linear regression can be performed if $\mathbf{y}$ and $\mathbf{X}$ have a linear relationship.

Assume p has unknown parameters $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)$ to be estimated, and response variable $\mathbf{y} = (y_1, \ldots, y_n)$ with size n is linearly related with $n \times p$ matrix $\mathbf{X}$ by

$$y_i = \sum_{j=1}^{p} x_{ij}\mu_j + \epsilon_i.$$

The above expression can also be expressed in matrix notation

$$\mathbf{y} = \mathbf{X}\boldsymbol{\mu} + \boldsymbol{\epsilon}, \tag{3.25}$$

where $\boldsymbol{\epsilon}$ is the error term.

The traditional way to estimate $\boldsymbol{\mu}$ is ordinary least squares (OLS), by minimizing the sum of squares

$$\underset{\boldsymbol{\mu}}{\text{Minimize}} \sum_{i=1}^{n} \epsilon_i^2. \tag{3.26}$$

It is also equivalent to maximum likelihood estimation with the normality assumption. Assume error term $\boldsymbol{\epsilon}$ is i.i.d. and has normal distribution with mean 0 and constant variance, that is

$$\boldsymbol{\epsilon} = \mathbf{y} - \mathbf{X}\boldsymbol{\mu} \sim N(0, \sigma^2). \tag{3.27}$$

Therefore, $\boldsymbol{\mu}$ is the maximum likelihood estimator obtained by maximizing the likelihood function

$$\begin{aligned} L(\boldsymbol{\epsilon}; \boldsymbol{\mu}, \sigma^2) &= \frac{1}{(\sqrt{2\pi\sigma^2})^n} \exp\left\{-\frac{\boldsymbol{\epsilon}^2}{2\sigma^2}\right\} \\ &= \frac{1}{(\sqrt{2\pi\sigma^2})^n} \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{y} - \mathbf{X}\boldsymbol{\mu}||^2\right\}. \end{aligned} \tag{3.28}$$

But this assumption may not necessarily hold in the real world, it can be badly fitted if error term $\boldsymbol{\epsilon}$ has heavier tails than the tails of a normal, or if the observations contain outliers.

The OLS estimator is very sensitive, it can be hugely biased by outliers or heavy-tailed data. Our new linear robust regression approach is replacing normal assumption by assuming error term $\boldsymbol{\epsilon}$ has scale mixtures of normal distributions with mean 0 to resolve the heavy-tailed problem.

3.6.2 Formulation

The estimators of linear robust semiprametric regression with scale mixtures of normal distributions approach can be also computed by the CNM-MS algorithm. Similar to Section 3.3, we need to formulate our problem to compute semiparametric MLE for $\boldsymbol{\mu}$ in the CNM-MS framework.

We make a new assumption that $\boldsymbol{\epsilon}$ has a scale mixture of normal distribution with mean 0. Refer to Section 2.2, we can discuss discrete G only when computing semiparametric MLE of mixture models. We estimate a parametric component p-vector $\boldsymbol{\mu}$ (instead of scalar we discussed in previous section), and nonparametric components $(\boldsymbol{\pi}, \boldsymbol{\theta})$ with unknown dimension K.

Refer to equation (3.13), $\boldsymbol{\epsilon} = \mathbf{y} - \mathbf{X}\boldsymbol{\mu}$ has probability density function

$$\begin{aligned} f(\epsilon; \boldsymbol{\pi}, \boldsymbol{\theta}) &= \sum_{k=1}^{K} \pi_k \phi(\epsilon; 0, \theta_k) \\ &= \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{2\pi\theta_k}} \exp\left(-\frac{\epsilon^2}{2\theta_k}\right) \\ &= \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{2\pi\theta_k}} \exp\left(-\frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta_k}\right). \end{aligned} \tag{3.29}$$

We need various expressions in the CNM-MS framework, which is very similar to Section 3.3 but with slight modification. The density function of normal

mixtures component is

$$\phi(\epsilon; 0, \theta) = \frac{1}{\sqrt{2\pi\theta}} \exp\left(-\frac{\epsilon^2}{2\theta}\right) \tag{3.30}$$

$$= \frac{1}{\sqrt{2\pi\theta_k}} \exp\left(-\frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta}\right), \tag{3.31}$$

and $\log \phi(.)$ is

$$\begin{aligned} l(\epsilon; 0, \theta) &= \log \phi(\epsilon; 0, \theta_k) \\ &= \log\left\{\frac{1}{\sqrt{2\pi\theta}} \exp\left(-\frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta}\right)\right\} \\ &= -\frac{1}{2}\log 2\pi - \frac{1}{2}\log\theta - \frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta}. \end{aligned} \tag{3.32}$$

The partial derivative of $l(.)$ with respect to p-vector $\boldsymbol{\mu}$ is

$$\begin{aligned} \frac{\partial l}{\partial \boldsymbol{\mu}} &= \frac{\partial}{\partial \boldsymbol{\mu}}\left(-\frac{1}{2}\log 2\pi - \frac{1}{2}\log\theta - \frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta}\right) \\ &= \frac{\partial}{\partial \mu}\left(-\frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta}\right) \\ &= \frac{1}{\theta}\mathbf{X}^T(y - \mathbf{X}\boldsymbol{\mu}). \end{aligned} \tag{3.33}$$

The partial derivative of $l(.)$ with respect to θ is

$$\begin{aligned} \frac{\partial l}{\partial \theta} &= \frac{\partial}{\partial \theta}\left(-\frac{1}{2}\log 2\pi - \frac{1}{2}\log\theta - \frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta}\right) \\ &= \frac{\partial}{\partial \theta}\left(-\frac{1}{2}\log\theta - \frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta}\right) \\ &= -\frac{1}{2\theta} + \frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta^2}. \end{aligned} \tag{3.34}$$

The second partial derivative of $l(.)$ with respect to θ is

$$\begin{aligned} \frac{\partial^2 l}{\partial \theta^2} &= \frac{\partial}{\partial \theta}\left(-\frac{1}{2\theta} + \frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{2\theta^2}\right) \\ &= \frac{1}{2\theta^2} - \frac{(y - \mathbf{X}\boldsymbol{\mu})^2}{\theta^3}. \end{aligned} \tag{3.35}$$

By using the above expressions with the CNM-MS algorithm discussed in Section 2.3, semiparametric MLE for linear robust regression with scale mixture of normal distribution can be computed with appropriate range for $\boldsymbol{\theta}$, which is discussed in Section 3.4.1.

3.6.3 Existing methods

This section describes some other existing methods for obtaining estimators of linear robust regression.

Least absolute deviation

Least absolute deviation (LAD) is very similar to ordinary least squares (OLS) that we described in Section 3.6.1. Instead of minimizing the sum of error squares, LAD obtains estimator of $\boldsymbol{\mu}$ by minimizing the sum of absolute errors,

$$\underset{\boldsymbol{\mu}}{\text{Minimize}} \sum_{i=1}^{n} |\epsilon_i|. \tag{3.36}$$

It is also equivalent to maximum likelihood estimation if we assume error term have a Laplace distribution.

LAD is robust to outliers. If there are outliers in the data, that is, they have unusual large errors, LAD has less of an effect of minimizing the sum of absolute errors than minimizing the sum of error squares. Therefore LAD is quite popular in other areas due to its robustness.

Least median of squares

Least median of squares (LMS) was introduced by Rousseeuw (1984), is similar to OLS and LAD. It obtains an estimator of $\boldsymbol{\mu}$ by minimizing the median of error squares,

$$\underset{\boldsymbol{\mu}}{\text{Minimize}} \quad \text{med}(\epsilon_i^2). \tag{3.37}$$

Least trimmed squares

Least trimmed squares (LTS) was introduced by Rousseeuw and Leroy (1987) to improve LMS estimators. It is very similar to OLS, instead of minimizing sum of error squares, LMS obtain an estimator of $\boldsymbol{\mu}$ by minimizing the quantile of error squares,

$$\underset{\boldsymbol{\mu}}{\text{Minimize}} \sum_{i=1}^{h} (\boldsymbol{\epsilon}^2)_{i:n} \tag{3.38}$$

Where $(\boldsymbol{\epsilon}^2)_{i:n}$ is ith order of $\boldsymbol{\epsilon}^2$, that is $(\boldsymbol{\epsilon}^2)_{1:n} \leq \cdots \leq (\boldsymbol{\epsilon}^2)_{n:n}$. The number of data points that going to trimmed is $n - h$, where $h = 1, \ldots, n$. If $h = n$, LTS is equivalent to OLS.

3.6.4 Simulation Study

In this section, we perform a simulation study to investigate the performance of linear robust semiparametric regression with scale mixtures of normal distributions, and compare various linear robust regression estimators as a benchmark. The simulation study is setup by following steps,

1. Generate random data $\mathbf{y} = \mathbf{X}\boldsymbol{\mu} + \boldsymbol{\epsilon}$ with sample size n, where
   - $\boldsymbol{\epsilon} \sim H$
   - $\mathbf{X} \sim U(0, 100)$
   - $\boldsymbol{\mu} = (1, 2, 3)$
2. Obtain estimator of $\boldsymbol{\mu}$ by various method,
   - Robust semiparametirc regression with SMN error (SMN)
   - Ordinary least squares (OLS)
   - Least absolute deviation (LAD)
   - Least median of squares (LMS)
   - Least trimmed squares (LTS)
3. Repeat 1-2 for $1,000$ times, calculate mean square error (MSE) for $\hat{\boldsymbol{\mu}}$ with known true value $\boldsymbol{\mu} = (1, 2, 3)$.

Possible error distribution H are chosen from various heavy-tailed distributions,

- t distribution with different degrees of freedom, 1.5, 3 and 6
- standard laplace distribution
- standard logistic distribution
- contaminated normal distribution with different proportions, 0%, 25% and 50%

We repeat these steps for different heavy-tailed distribution H and different sample size $n = (50, 100, 200, 300, 400, 500)$. We choose mean square error (MSE) for $\hat{\boldsymbol{\mu}}$ as statistics since MSE can be expressed as sum of bias square and variance, therefore MSE can measure variation and unbiasedness at the same time. The lower the MSE, the better the estimate.

We perform this simulation study in statistical software `R`, and use its built-in function `glm` for the OLS estimator. The LAD estimator is obtained by function `rq` in package `quantreg`. LMS and LTS estimator are obtained by function `lqs` in package `MASS`, we use the default trimmed percentage 50% in function `lqs` to perform LTS estimation in our simulation study.

Result

The computational result for mean square error (MSE) when sample size is 500 can be found in Table 3.6-3.8, and relevant MSE plots with different sample size are shown in Figure 3.9-3.12.

Generally, MSE is large, and not stable when sample size is small, such as 50 or 100, but it is getting closer to 0 and much more stable as sample size increases.

From these simulation results, SMN has almost the same performance to OLS when error term has single normal distribution (normal with 0% contaminated), since OLS assume error term has single normal distribution, and SMN performs as good as OLS.

SMN has almost the same performance to LAD when error term has a Laplace distribution, since LAD assume error term has a Laplace distribution, and SMN performs as well as LAD.

SMN and LAD both have similar, and very good performance as the error term has t-distribution. The performance of OLS is getting better as degrees of freedom increases, since t-distribution gets closer to normal distribution as degrees of freedom increases.

When the error term has contaminated normal distribution, SMN has the best performance of all, LAD has the second best performance. LTS has a reasonably good performance if we choose the correct percentage to trimmed, LTS performs reasonably well as LAD when error term has normal distribution with 50% contaminated since we choose trimmed 50% of data for LTS. But in practice, it is very difficult to determine the suitable percentage to trimmed. LMS generally has bad performance.

In conclusion, LAD has an acceptable good performance in many of our simulation study except contaminated normal error. Our new approach, linear robust semiparametirc regression with scale mixture of normal distribution is the only one has good consistent performance.

	Contaminated Normal			t			Laplace	Logistic
	0%	25%	50%	$df = 1.5$	$df = 3$	$df = 6$	$s = 1$	$s = 1$
SMN	0.0176	0.0270	0.0411	0.0289	0.0267	0.0276	0.0155	0.0507
OLS	0.0143	3.2281	5.6381	1.6499	0.0387	0.0230	0.0270	0.0468
LAD	0.0236	0.0384	0.0772	0.0298	0.0259	0.0294	0.0153	0.0495
LMS	0.1343	0.1227	0.1636	0.1022	0.1219	0.1204	0.0841	0.2569
LTS	0.1599	0.1022	0.0731	0.0787	0.1213	0.1287	0.0434	0.3110

Table 3.6: Mean square error for $\hat{\mu_0}$, $n = 500$

	Contaminated Normal			t			Laplace	Logistic
	0%	25%	50%	$df = 1.5$	$df = 3$	$df = 6$	$s = 1$	$s = 1$
SMN	2.7039	4.8005	6.9330	0.0537	0.0398	0.0378	0.0270	0.0768
OLS	2.3475	533.5697	1026.4241	2.2710	0.0715	0.0358	0.0451	0.0758
LAD	4.0338	6.3848	15.1277	0.0563	0.0416	0.0432	0.0259	0.0837
LMS	23.8003	21.6909	30.3138	0.1941	0.1821	0.2174	0.1416	0.4629
LTS	25.5576	18.6911	12.4121	0.1527	0.1946	0.2185	0.0780	0.5289

Table 3.7: Mean square error for $\hat{\mu_1}$, $n = 500$, unit=10^{-4}

	Contaminated Normal			t			Laplace	Logistic
	0%	25%	50%	$df = 1.5$	$df = 3$	$df = 6$	$s = 1$	$s = 1$
SMN	0.2441	0.4588	0.7763	0.0051	0.0043	0.0036	0.0029	0.0087
OLS	0.2229	52.5247	100.8656	0.3196	0.0073	0.0036	0.0047	0.0083
LAD	0.3485	0.6343	1.3997	0.0057	0.0048	0.0040	0.0029	0.0107
LMS	2.3129	2.1051	3.0712	0.0176	0.0185	0.0183	0.0140	0.0479
LTS	2.9032	1.7320	1.4246	0.0149	0.0177	0.0196	0.0080	0.0544

Table 3.8: Mean square error for $\hat{\mu_2}$, $n = 500$, unit=10^{-3}

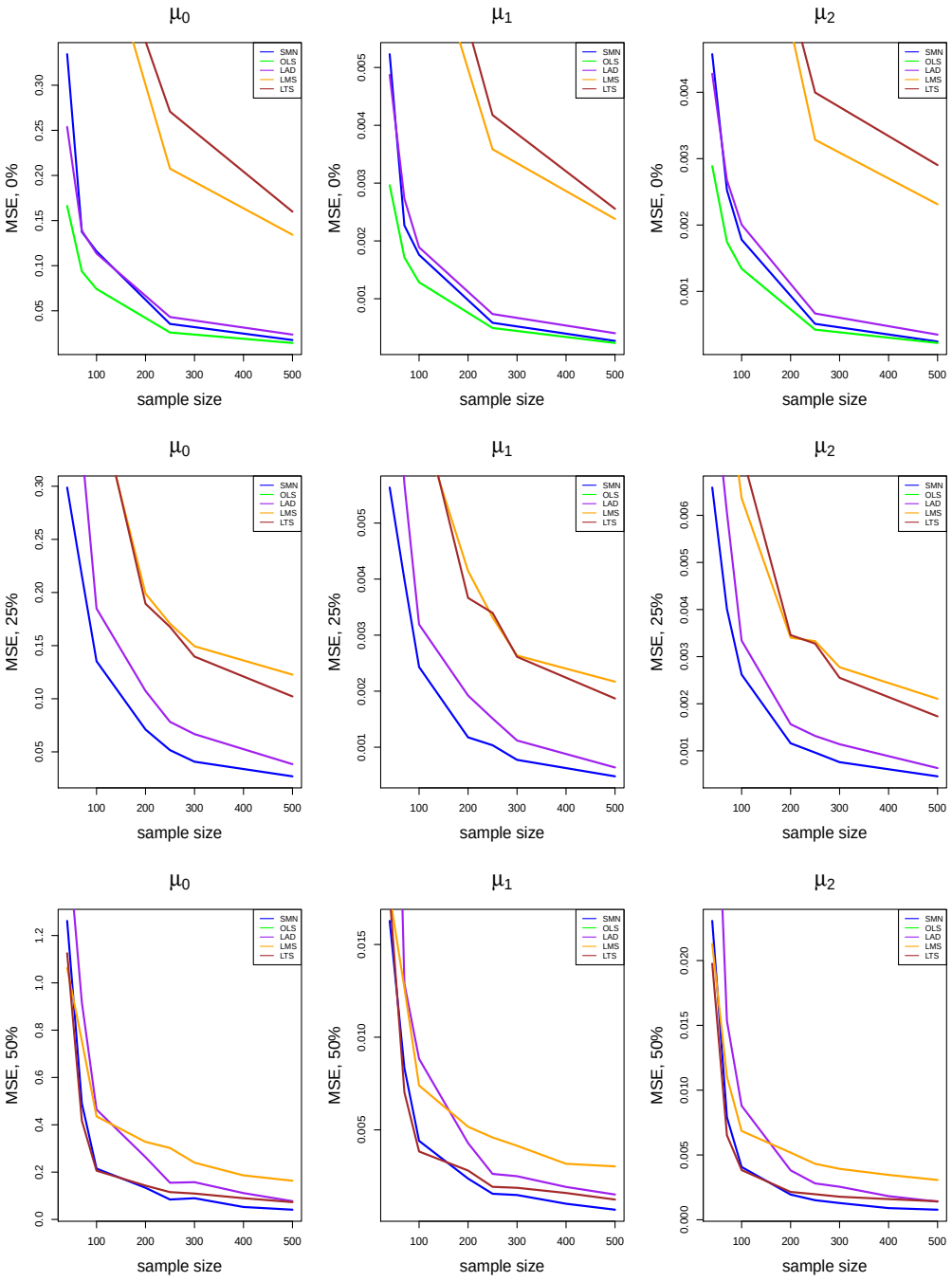

Figure 3.9: MSE plot of contaminated normal distribution for 0%, 25% and 50%

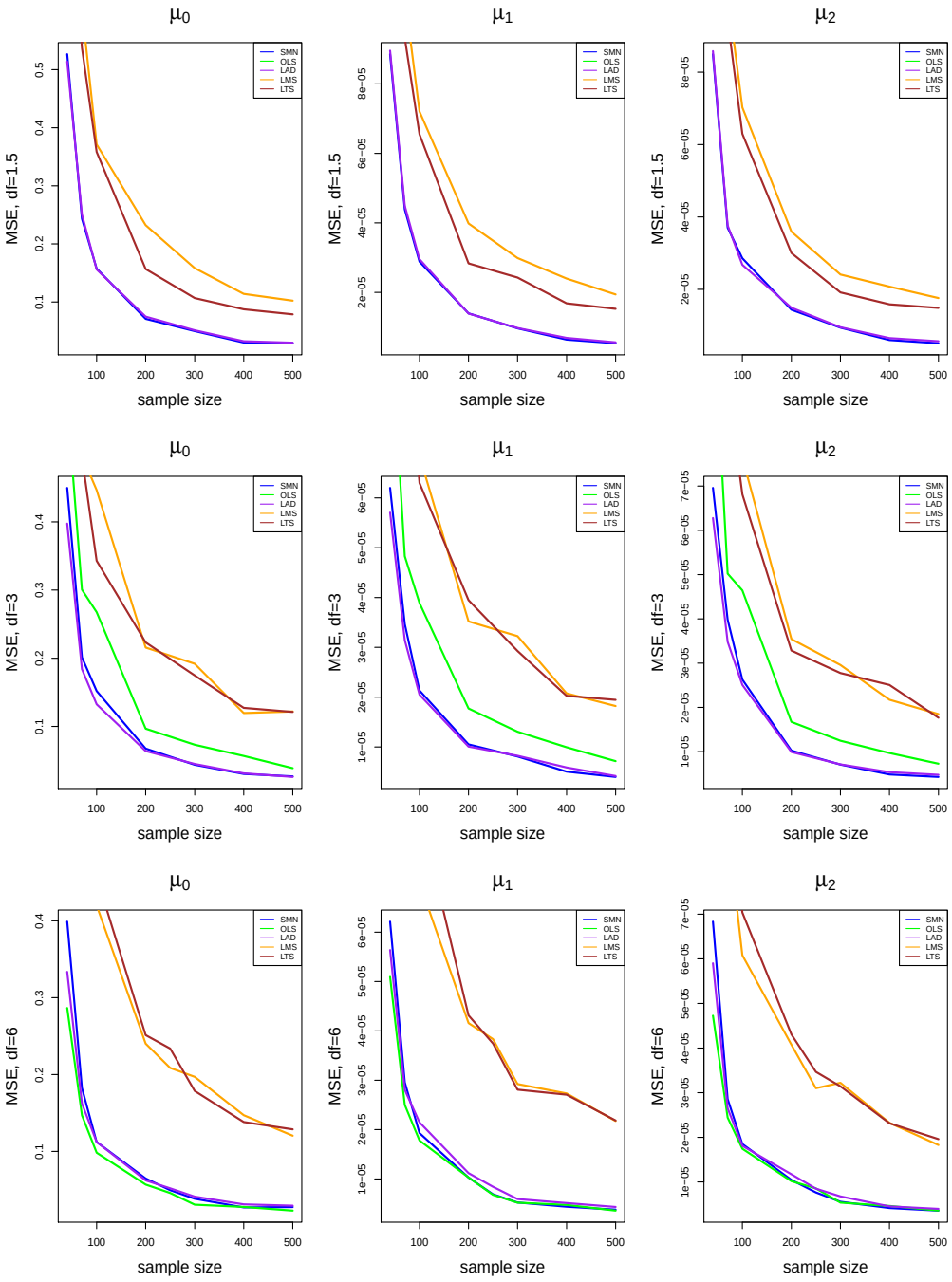

Figure 3.10: MSE plot of t distribution for d.f.=(1.5, 3, 6)

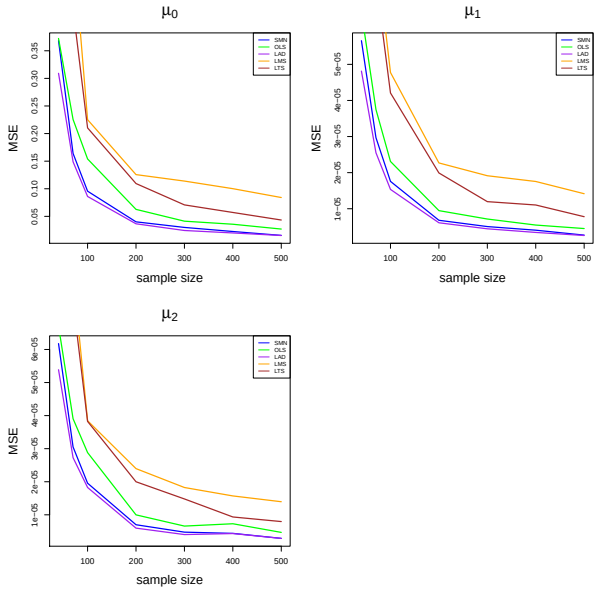

Figure 3.11: MSE plot of standard Laplace distribution

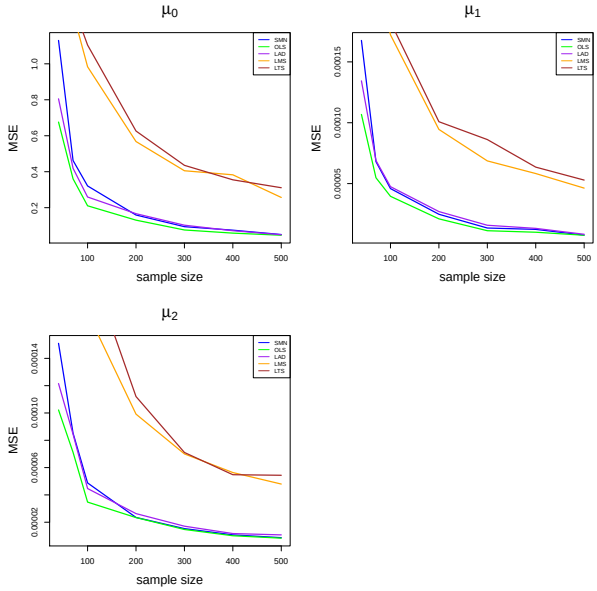

Figure 3.12: MSE plot of standard logistic distribution

3.7 Summary

In this chapter, we discussed scale mixtures of normal distributions (SMN). We introduced the way to obtain semiparametric MLE of SMN by the CNM-MS algorithm, and its possible applications without and with covariates (robust linear regression).

Semiparametric MLE of SMN obtained by CNM-MS algorithm fit simulated heavy-tailed data very well. Modeling log returns and Value-at-Risk (VaR) estimation also has very good performance, and it has similar result to t-distribution model. Both of them have much better results than the normal distribution model, the most common way. According to AIC statistic, we claim our SMN model is the best to model log returns and VaR estimation.

We also introduced how semiparametric MLE of linear robust regression with SMN error assumption can be computed by the CNM-MS algorithm. Our simulation study of linear robust regression shows that the SMN approach is the only one has good performance consistently.

Chapter 4

Scale Mixtures of Multivariate Normal Distributions

In this chapter, we introduce scale mixtures of multivariate normal distributions in Section 4.1 and its basic definition, relevant property in Section 4.2. Section 4.3.2 represents how to obtain the semeparametric MLEs for scale mixtures of multivariate normal distributions with slight modification from univariate case, and some other computation and presentation issue are described in Section 4.4

Section 4.5 presents possible applications of scale mixtures of multivariate normal distributions, fitting selected multivariate heavy-tailed data , modeling multiple log returns for portfolios and estimate Value-at-Risk. Finally, we summarized our result in Section 4.6.

4.1 Introduction

Scale mixtures of multivariate normal distributions (MSMN) is a generalization of the univariate scale mixtures of normal distributions (SMN) to higher dimensions. A random vector X is said be a p-dimensional scale mixtures of multivariate normal distribution if it has multivariate normal distribution with common mean

$\boldsymbol{\mu}$ and variance-covariance matrices in the form $\Theta\boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma}$ is $p \times p$ symmetirc matrix, and $\Theta \in (0, \infty)$ has univariate continuous or discrete probability distribution G.

4.2 Basic Definition

Suppose $\Theta \in (0, \infty)$ has univariate continuous or discrete probability distribution G, and X is a p-dimensional random vector with scale mixtures of multivariate normal distributions with density

$$f(x; \boldsymbol{\mu}, \boldsymbol{\Sigma}, G) = \int \phi_p(x; \boldsymbol{\mu}, \theta\boldsymbol{\Sigma}) \mathrm{d}G(\theta), \tag{4.1}$$

where $\boldsymbol{\mu}$ is the mean, $\boldsymbol{\Sigma}$ is $p \times p$ symmetric matrix and $\phi_p(.)$ is probability density function of p-variate normal distribution. If $p = 1$, it is equivalent to univariate scale mixtures of normal distributions that we discussed in Section 3. The cumulative distribution function of X is

$$F(x; \boldsymbol{\mu}, \boldsymbol{\Sigma}, G) = \int \Phi_p(x; \boldsymbol{\mu}, \theta\boldsymbol{\Sigma}) \mathrm{d}G(\theta), \tag{4.2}$$

where Φ_p is cumulative distribution function of p-variate normal distribution. The variance-covariance matrix, $\boldsymbol{\Psi}$, of X is the product of $\boldsymbol{\Sigma}$ and expected value of Θ,

$$\boldsymbol{\Psi} = \boldsymbol{\Sigma} \int \theta \mathrm{d}G(\theta). \tag{4.3}$$

It also shares some common properties with the multivariate normal distribution. The marginal distribution of scale mixtures of multivariate normal distributions is also univariate scale mixtures of normal distributions. The marginal distribution of jth component of X has density

$$f(x; \mu_j, \Sigma_{jj}, G) = \int \phi(x; \mu_j, \theta\Sigma_{jj}) \mathrm{d}G(\theta), \tag{4.4}$$

where μ_j is jth element of $\boldsymbol{\mu}$, Σ_{jj} is (j, j)th entry of symmetric matrix $\boldsymbol{\Sigma}$.

The linear combination of its components also has univariate scale mixtures of normal distributions. Let $\mathbf{a} = (a_1, \ldots, a_p)^T \in \mathbb{R}^p$, linear combination $\mathbf{Y} = \mathbf{a}^T\mathbf{X} = a_1\mathbf{X}_1 + \cdots + a_p\mathbf{X}_p$ have univariate scale mixtures of normal distributions with probability density function

$$f(y; \mathbf{a}^T\boldsymbol{\mu}, \mathbf{a}^T\boldsymbol{\Sigma}\mathbf{a}, G) = \int \phi(x; \mathbf{a}^T\boldsymbol{\mu}, \theta\mathbf{a}^T\boldsymbol{\Sigma}\mathbf{a})\mathrm{d}G(\theta). \tag{4.5}$$

4.3 Semiparametirc MLE

In this section, we introduce how to compute semiparametric MLE for scale mixtures of multivariate normal distributions.

4.3.1 Discrete G

As we discussed in Section 2.2, similar to 3.3.1, we can just regard G as a discrete distribution when computing semiparametirc MLE for mixture models. Again, we rewrite density function (4.2) of p-dimensional random vector X in discrete form with unknown K-components multivariate normal mixtures

$$f(x; \boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\Sigma}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \phi_p(x; \boldsymbol{\mu}, \theta_k\Sigma), \tag{4.6}$$

where the discrete G consists of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K) \in (0, \infty)$ with mixing proportion $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K) \in (0, 1)$ respectively. $\theta_k\boldsymbol{\Sigma}$ is variance-covariance matrix for each multivariate normal mixture component, where θ_k is multiplication coefficient of common base variance-covariance matrix $\boldsymbol{\Sigma}$. The cumulative distribution function of X can also be rewritten in discrete form

$$F(x; \boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\Sigma}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \Phi_p(x; \boldsymbol{\mu}, \theta_k\boldsymbol{\Sigma}). \tag{4.7}$$

The mean for scale of mixture multivariate normal distributions is its common mean, μ. The variance-covariance matrix for scale mixture of multivariate normal distributions, $\boldsymbol{\Psi}$ is

$$\boldsymbol{\Psi} = \boldsymbol{\Sigma} \sum_{k=1}^{K} \pi_k\theta_k. \tag{4.8}$$

The marginal distribution of jth component of X has probability density function

$$f(x;\mu_j,\boldsymbol{\pi},\Sigma_{jj},\boldsymbol{\theta})=\sum_{k=1}^{K}\pi_k\phi(x;\mu_j,\theta_k\Sigma_{jj}). \tag{4.9}$$

The linear combination of its components also has univariate scale mixture of normal distribution. Let $\mathbf{a}=(a_1,\ldots,a_p)^T\in\mathbb{R}^p$, linear combination $\mathbf{Y}=\mathbf{a}^T\mathbf{X}=a_1\mathbf{X}_1+\cdots+a_p\mathbf{X}_p$ has density

$$f(y;\mathbf{a}^T\boldsymbol{\mu},\boldsymbol{\pi},\mathbf{a}^T\boldsymbol{\Sigma}\mathbf{a},\boldsymbol{\theta})=\sum_{k=1}^{K}\pi_k\phi(y;\mathbf{a}^T\boldsymbol{\mu},\theta_k\mathbf{a}^T\boldsymbol{\Sigma}\mathbf{a}). \tag{4.10}$$

4.3.2 Formulation

We now compute parametric components (a p-vector $\boldsymbol{\mu}$ and a $p\times p$ symmetric matrix $\boldsymbol{\Sigma}$) and nonparametric components $(\boldsymbol{\pi},\boldsymbol{\theta})$ with unknown dimension K. Firstly, we define density (4.6) much details, that is

$$f(x;\boldsymbol{\mu},\boldsymbol{\pi},\boldsymbol{\Sigma},\boldsymbol{\theta})=\sum_{k=1}^{K}\pi_k\phi_p(x;\boldsymbol{\mu},\theta_k\Sigma) \tag{4.11}$$

$$=\sum_{k=1}^{K}\pi_k\frac{1}{(2\pi\theta_k)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2\theta_k}(x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})\right\} \tag{4.12}$$

and the log-likelihood for random variable X with sample size n is

$$l(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\pi},\boldsymbol{\Sigma},\boldsymbol{\theta})=\sum_{i=1}^{n}\log\left\{\sum_{k=1}^{K}\pi_k\phi_p(x_i;\boldsymbol{\mu},\theta_k\Sigma)\right\} \tag{4.13}$$

Similar to univariate case, as discussed in 3.3, we can compute semiparametric MLE for scale mixtures of normal distributions with the CNM-MS algorithm, but with some modifications for the multivariate case. We can combine the Expectation-Maximization (EM) algorithm and the CNM-MS algorithm in following steps,

1. Set initial value for $(\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\theta})$ such that $l(\mathbf{x};\boldsymbol{\mu},\boldsymbol{\pi},\boldsymbol{\Sigma},\boldsymbol{\theta})>-\infty$

2. Update $\boldsymbol{\Sigma}$ to $\boldsymbol{\Sigma}'$ by using the EM algorithm with fixed $(\boldsymbol{\mu},\boldsymbol{\theta})$

3. Update $(\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\theta})$ to $(\boldsymbol{\mu}', \boldsymbol{\pi}, \boldsymbol{\theta}')$ by using the CNM-MS algorithm with updated $\boldsymbol{\Sigma}'$

4. If $|l(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\Sigma}, \boldsymbol{\theta}) - l(\mathbf{x}; \boldsymbol{\mu}', \boldsymbol{\pi}', \boldsymbol{\Sigma}', \boldsymbol{\theta}')| < 10^{-10}$, stop, otherwise back to step 2

We compute semiparametric MLE in two steps, update $\boldsymbol{\Sigma}$ with fixed $(\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\theta})$ and update $(\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\theta})$ with updated $\boldsymbol{\Sigma}'$. The $p \times p$ symmetric matrix $\boldsymbol{\Sigma}$ is difficult, and takes a lot of computing resource to compute by differentiation approach, thus we update $\boldsymbol{\Sigma}$ by using the EM algorithm. Update $(\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\theta})$ with updated $\boldsymbol{\Sigma}'$ just reduce the problem to univariate case, and it is easy to implement by using the CNM-MS algorithm.

4.3.3 Update Σ with fixed (μ, π, θ)

The Expectation-Maximization (EM) algorithm is introduced by Dempster et al. (1977), and it is often used to obtain maximum likelihood estimator in statistics. It is an iterative method consisting of two steps, Expectation (E) steps and Maximization (M) steps. We update $\boldsymbol{\Sigma}$ to $\boldsymbol{\Sigma}'$ by using the EM algorithm with fixed $(\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\theta})$.

Briefly, in Expectation steps, we calculate $Q(\boldsymbol{\Sigma}'|\boldsymbol{\Sigma})$, the expected value of the log-likelihood function based on the current estimated parameter. In Maximization steps, we obtain a new parameter $\boldsymbol{\Sigma}'$ by maximizing $Q(\boldsymbol{\Sigma}'|\boldsymbol{\Sigma})$.

Expectation step

Estimate $\boldsymbol{\Sigma}'$ by given current estimation $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\theta})$, the proportional height of the normal density weighted by ρ expressed as

$$\rho'_{ik} = \frac{\pi_k \phi_p(\mathbf{x}_i, \boldsymbol{\mu}, \theta_k \boldsymbol{\Sigma})}{\sum_{j=1}^{K} \pi_j \phi_p(\mathbf{x}_i, \boldsymbol{\mu}, \theta_j \boldsymbol{\Sigma})}.$$

Thus, the E-step results in the function

$$Q(\boldsymbol{\Sigma}'|\boldsymbol{\Sigma}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \rho'_{ik} \log\left(\pi_k \phi_p(\mathbf{x}_i; \boldsymbol{\mu}, \theta_k \boldsymbol{\Sigma})\right). \tag{4.14}$$

Maximization step

Ignore any constant terms (since they will disappear after taking derivatives), rewrite $Q(\boldsymbol{\Sigma}'|\boldsymbol{\Sigma})$ as $Q'(\boldsymbol{\Sigma}'|\boldsymbol{\Sigma})$,

$$\begin{aligned} Q'(\boldsymbol{\Sigma}'|\boldsymbol{\Sigma}) &= \sum_{i=1}^{n}\sum_{k=1}^{K} \rho'_{ik} \left[-\frac{1}{2}\log(|\boldsymbol{\Sigma}|) - \frac{1}{2\theta_k}(\mathbf{x}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu})\right] \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \rho'_{ik} \left[\frac{1}{2}\log|\boldsymbol{\Sigma}^{-1}| - \frac{1}{2\theta_k}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1} N_i\right)\right], \end{aligned}$$

where $N_i = (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$.

Taking derivative of $Q'(\boldsymbol{\Sigma}'|\boldsymbol{\Sigma})$ respect to $\boldsymbol{\Sigma}^{-1}$,

$$\begin{aligned} \frac{\partial Q'(\boldsymbol{\Sigma}'|\boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}^{-1}} &= \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} \rho'_{ik}\left[2\boldsymbol{\Sigma} - \mathrm{diag}\left(\boldsymbol{\Sigma}\right)\right] - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} \frac{\rho'_{ik}}{\theta_k}\mathrm{tr}\left(2N_i - \mathrm{diag}(N_i)\right) \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} \rho'_{ik}(2M_i - \mathrm{diag}(M_i)) \\ &= 2S - \mathrm{diag}(S), \end{aligned}$$

where

$$\begin{aligned} M_i &= \boldsymbol{\Sigma} - \frac{N_i}{\theta_k}, \\ S &= \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} \rho'_{ik} M_i. \end{aligned}$$

Setting the derivative, $2S - \mathrm{diag}(S) = 0$, implies $S = 0$,

$$S = \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} \rho'_{ik}\left(\boldsymbol{\Sigma}' - \frac{1}{\theta_k}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T\right) = 0.$$

Therefore updated $\boldsymbol{\Sigma}'$ can be expressed as

$$\boldsymbol{\Sigma}' = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} \frac{\rho'_{ik}}{\theta_k}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T. \tag{4.15}$$

4.3.4 Update $(\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\theta})$ with updated $\boldsymbol{\Sigma}'$

After obtaining updated $\boldsymbol{\Sigma}'$ from Equation (4.15), similar to univariate SMN, we can compute semiparametric MLE for MSMN using the CNM-MS algorithm. Update $(\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\theta})$ to $(\boldsymbol{\mu}', \boldsymbol{\pi}', \boldsymbol{\theta}')$. Density function of multivariate normal components is

$$\phi_p(x; \boldsymbol{\mu}, \theta\boldsymbol{\Sigma}) = \frac{1}{(2\pi\theta)^{\frac{p}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\theta}(x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(x-\boldsymbol{\mu})\right\}, \tag{4.16}$$

and $\log \phi_p(.)$ is

$$l(x; \boldsymbol{\mu}, \boldsymbol{\Sigma}', \theta) = -\frac{p}{2}\log 2\pi - \frac{p}{2}\log\theta - \frac{1}{2}\log|\boldsymbol{\Sigma}'| - \frac{1}{2\theta}\left[(x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}'^{-1}(x-\boldsymbol{\mu})\right]. \tag{4.17}$$

The partial derivative of $l(.)$ with respect to $\boldsymbol{\mu}$ is

$$\begin{aligned}\frac{\partial l}{\partial \mu} &= \frac{\partial}{\partial \mu}\left(-\frac{1}{2\theta}\left[(x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}'^{-1}(x-\boldsymbol{\mu})\right]\right) \\ &= \frac{1}{\theta_k}\boldsymbol{\Sigma}'^{-1}(x-\boldsymbol{\mu}).\end{aligned} \tag{4.18}$$

The partial derivative of $l(.)$ with respect to θ is

$$\begin{aligned}\frac{\partial l}{\partial \theta} &= \frac{\partial}{\partial \theta}\left(-\frac{p}{2}\log\theta - \frac{1}{2\theta}\left[(x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}'^{-1}(x-\boldsymbol{\mu})\right]\right) \\ &= -\frac{p}{2\theta} + \frac{1}{2\theta^2}\left[(x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}'^{-1}(x-\boldsymbol{\mu})\right].\end{aligned} \tag{4.19}$$

The 2nd partial derivative of $l(.)$ with respect to θ is

$$\begin{aligned}\frac{\partial^2 l}{\partial \theta^2} &= \frac{\partial}{\partial \theta}\left(-\frac{p}{2\theta} + \frac{1}{2\theta^2}\left[(x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}'^{-1}(x-\boldsymbol{\mu})\right]\right) \\ &= \frac{p}{2\theta^2} - \frac{1}{\theta^3}\left[(x-\boldsymbol{\mu})^T\boldsymbol{\Sigma}'^{-1}(x-\boldsymbol{\mu})\right].\end{aligned} \tag{4.20}$$

Hence, we can compute the gradient function, find local maxima of the gradient function and update p-vector $\boldsymbol{\mu}$ by using the above expressions with the CNM-MS algorithm. We now can obtain semiparametric MLE $(\boldsymbol{\mu}', \boldsymbol{\pi}', \boldsymbol{\theta}')$, then update $(\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\theta})$ to $(\boldsymbol{\mu}', \boldsymbol{\pi}', \boldsymbol{\theta}')$.

4.4 Miscellanea

4.4.1 Choice of θ

Computationally, we need to specify the range of $\boldsymbol{\theta}$. It is difficult since $\boldsymbol{\theta}$ is unbound. By using the formulation in last section, we reduced the problem to the univariate case when using the CNM-MS algorithm to update $(\boldsymbol{\mu}, \boldsymbol{\pi}, \boldsymbol{\theta})$. Thus, we continue to use the same strategy in the univariate case, as we discussed in Section 3.4.1. Hathaway (1985) suggest a linear inequality constraint

$$\frac{\theta_{\max}}{\theta_{\min}} \geq c > 0. \tag{4.21}$$

We choose an arbitrary value $c = 10^{16}$ in the CNM-MS implementation since it works practically well, since there is no suitable parametric method to decide c at the present time.

4.4.2 Standardizing θ

To present semiparametric MLE result much more clearly and easy to understand, we standardized estimated $\boldsymbol{\theta}$. Denote standardized $\boldsymbol{\theta}$ as $\boldsymbol{\theta}^+$, and corresponding $\boldsymbol{\Sigma}$ is $\boldsymbol{\Sigma}^+$,

$$\boldsymbol{\theta}^+ = \frac{1}{\theta_{min}}\boldsymbol{\theta}, \tag{4.22}$$

$$\boldsymbol{\Sigma}^+ = \theta_{min}\boldsymbol{\Sigma}, \tag{4.23}$$

Thus, θ^+_{min} must be 1, and the rest of elements all greater or equal to 1. We all present results in this form in following sections.

4.5 Applications

In this section, we study the performance of semiparametric MLE for scale mixtures of multivariate normal distributions (MSMN) using the CNM-MS algorithm. Then we introduce a possible application for MSMN, which is modeling multiple log-returns and estimate the Value-at-Risk for portfolios.

4.5.1 Simulated Data

We generate four heavy-tailed multivariate datasets for different distribution and different dimensions with sample size $5,000$. Details of these dataset is described as follows,

1. 2-dimensional scale mixture of multivariate normal distributions, denote as N2, with following parameters,

$$\boldsymbol{\mu} = (15, 3)^T, \boldsymbol{\pi} = (0.3, 0.7)^T, \boldsymbol{\theta} = (1, 6)^T$$
$$\boldsymbol{\Sigma} = \begin{pmatrix} 5 & 0.2 \\ 0.2 & 2 \end{pmatrix}.$$

2. 4-dimensional scale mixture of multivariate normal distributions, denote as N4, with following parameters,

$$\boldsymbol{\mu} = (15, 3, 25, 50)^T, \boldsymbol{\pi} = (0.3, 0.5, 0.2)^T, \boldsymbol{\theta} = (1, 3, 15)^T$$
$$\boldsymbol{\Sigma} = \begin{pmatrix} 5 & 0.2 & 0.8 & -0.5 \\ 0.2 & 6 & -0.4 & 0.3 \\ 0.8 & -0.4 & 20 & 0.7 \\ -0.5 & 0.3 & 0.7 & 1 \end{pmatrix}.$$

3. 2-dimensional multivariate t-distribution, denote as $t2$, with following parameters,

$$\boldsymbol{\mu} = (2, 12)^T, \nu = 4, \boldsymbol{\Lambda} = \begin{pmatrix} 8 & 2 \\ 2 & 3 \end{pmatrix},$$

   where $\boldsymbol{\mu}$ is mean, ν is degrees of freedom, $\boldsymbol{\Lambda}$ is dispersion matrix.

4. 4-dimensional multivariate t-distribution, denote as $t4$, with following pa-

rameters,

$$\boldsymbol{\mu} = (2, 15, 21, 30)^T, \nu = 4, \boldsymbol{\Lambda} = \begin{pmatrix} 5 & 0.2 & 0.8 & -0.5 \\ 0.2 & 6 & -0.4 & 0.3 \\ 0.8 & -0.4 & 20 & 0.7 \\ -0.5 & 0.3 & 0.7 & 1 \end{pmatrix},$$

where $\boldsymbol{\mu}$ is mean, ν is degrees of freedom, $\boldsymbol{\Lambda}$ is dispersion matrix.

Results

In univariate data, we can assess goodness of model fit by comparing fitted density with a histogram or empirical distribution. But it is very difficult to assess goodness of model fit in multivariate data, therefore we list the MLE value and create marginal fitted density plots for giving limited understanding of the model. Computation results can be found in Table 4.1. Figure 4.1 shows the marginal plot of fitted density for N2 and N4, Figure 4.2 shows the marginal plot of fitted density for $t2$ and $t4$.

The fitted model for N2 and N4 has very good performance. For N2, we fit a 2-components scale mixtures of multivariate normal distributions model (MSMN), which is the same as the true model. For N4, we fit a 3-components MSMN model, which is the same as the true model again. Fitted MLE for both N2 and N4 are also very close to true value, and fitted marginal density fit the data very well.

The fitted model for $t2$ and $t4$ also has very good performance. For $t2$, we fit a 3-components MSMN model, fitted mean and variance-covariance matrix are very close to true value. For $t4$, we also fit a 3-components MSMN model, fitted mean and variance-covariance matrix are also very close to true value. Note that for multivariate t distribution, variance-covariance matrix $\boldsymbol{\Psi}$ is related to degrees of freedom ν and dispersion matrix $\boldsymbol{\Lambda}$ by

$$\boldsymbol{\Lambda} = \frac{\nu - 2}{\nu}\boldsymbol{\Psi},$$

and expression of variance-covariance matrix for MSMN model can be found in Equation (4.8).

MSMN models have very good performance in different dimensions of data, and different types of heavy tailed data.

Data	True	MLE
N2	$\boldsymbol{\mu}=(15,3)^T$ $\boldsymbol{\pi}=(0.3,0.7)^T$ $\boldsymbol{\theta}=(1,6)^T$ $\boldsymbol{\Sigma}=\begin{pmatrix}5 & 0.2\\ 0.2 & 2\end{pmatrix}$	$\hat{\boldsymbol{\mu}}=(14.942,2.992)^T$ $\hat{\boldsymbol{\pi}}=(0.265,0.735)^T$ $\hat{\boldsymbol{\theta}}=(1.000,6.281)^T$ $\hat{\boldsymbol{\Sigma}}=\begin{pmatrix}4.423 & 0.168\\ 0.168 & 1.872\end{pmatrix}$
N4	$\boldsymbol{\mu}=(15,3,25,50)^T$ $\boldsymbol{\pi}=(0.3,0.5,0.2)^T$ $\boldsymbol{\theta}=(1,3,15)^T$ $\boldsymbol{\Sigma}=\begin{pmatrix}5 & 0.2 & 0.8 & -0.5\\ 0.2 & 6 & -0.4 & 0.3\\ 0.8 & -0.4 & 20 & 0.7\\ -0.5 & 0.3 & 0.7 & 1\end{pmatrix}$	$\hat{\boldsymbol{\mu}}=(14.930,2.956,25.058,49.967)$ $\hat{\boldsymbol{\pi}}=(0.250,0.557,0.193)^T$ $\hat{\boldsymbol{\theta}}=(1.000,3.089,16.927)$ $\hat{\boldsymbol{\Sigma}}=\begin{pmatrix}4.579 & 0.218 & 0.814 & -0.460\\ 0.218 & 5.663 & -0.430 & 0.219\\ 0.814 & -0.430 & 17.981 & 0.539\\ -0.460 & 0.219 & 0.539 & 0.901\end{pmatrix}$
$t2$	$\boldsymbol{\mu}=(2,12)^T$ $\boldsymbol{\Psi}=\begin{pmatrix}16 & 4\\ 4 & 6\end{pmatrix}$	$\hat{\boldsymbol{\mu}}=(2.068,12.032)^T$ $\hat{\boldsymbol{\pi}}=(0.546,0.428,0.026)^T$ $\hat{\boldsymbol{\theta}}=(1.000,3.817,23.372)^T$ $\hat{\boldsymbol{\Sigma}}=\begin{pmatrix}5.461 & 1.306\\ 1.306 & 2.017\end{pmatrix}$ $\hat{\boldsymbol{\Psi}}=\begin{pmatrix}15.257 & 3.648\\ 3.648 & 5.636\end{pmatrix}$
$t4$	$\boldsymbol{\mu}=(2,15,21,30)^T$ $\boldsymbol{\Psi}=\begin{pmatrix}10 & 0.4 & 1.6 & -1\\ 0.4 & 12 & -0.8 & 0.6\\ 1.6 & -0.8 & 40 & 1.4\\ -1 & 0.6 & 1.4 & 2\end{pmatrix}$	$\hat{\boldsymbol{\mu}}=(1.982,15.038,20.959,29.988)^T$ $\hat{\boldsymbol{\pi}}=(0.559,0.384,0.057)^T$ $\hat{\boldsymbol{\theta}}=(1.000,3.360,17.134)^T$ $\hat{\boldsymbol{\Sigma}}=\begin{pmatrix}3.572 & 0.199 & 0.547 & -0.328\\ 0.199 & 4.296 & -0.421 & 0.233\\ 0.547 & -0.421 & 15.063 & 0.542\\ -0.328 & 0.233 & 0.542 & 0.704\end{pmatrix}$ $\hat{\boldsymbol{\Psi}}=\begin{pmatrix}10.072 & 0.562 & 1.542 & -0.926\\ 0.562 & 12.112 & -1.187 & 0.656\\ 1.542 & -1.187 & 42.473 & 1.528\\ -0.926 & 0.656 & 1.528 & 1.985\end{pmatrix}$

Table 4.1: MSMN fit for heavy-tailed multivariate data

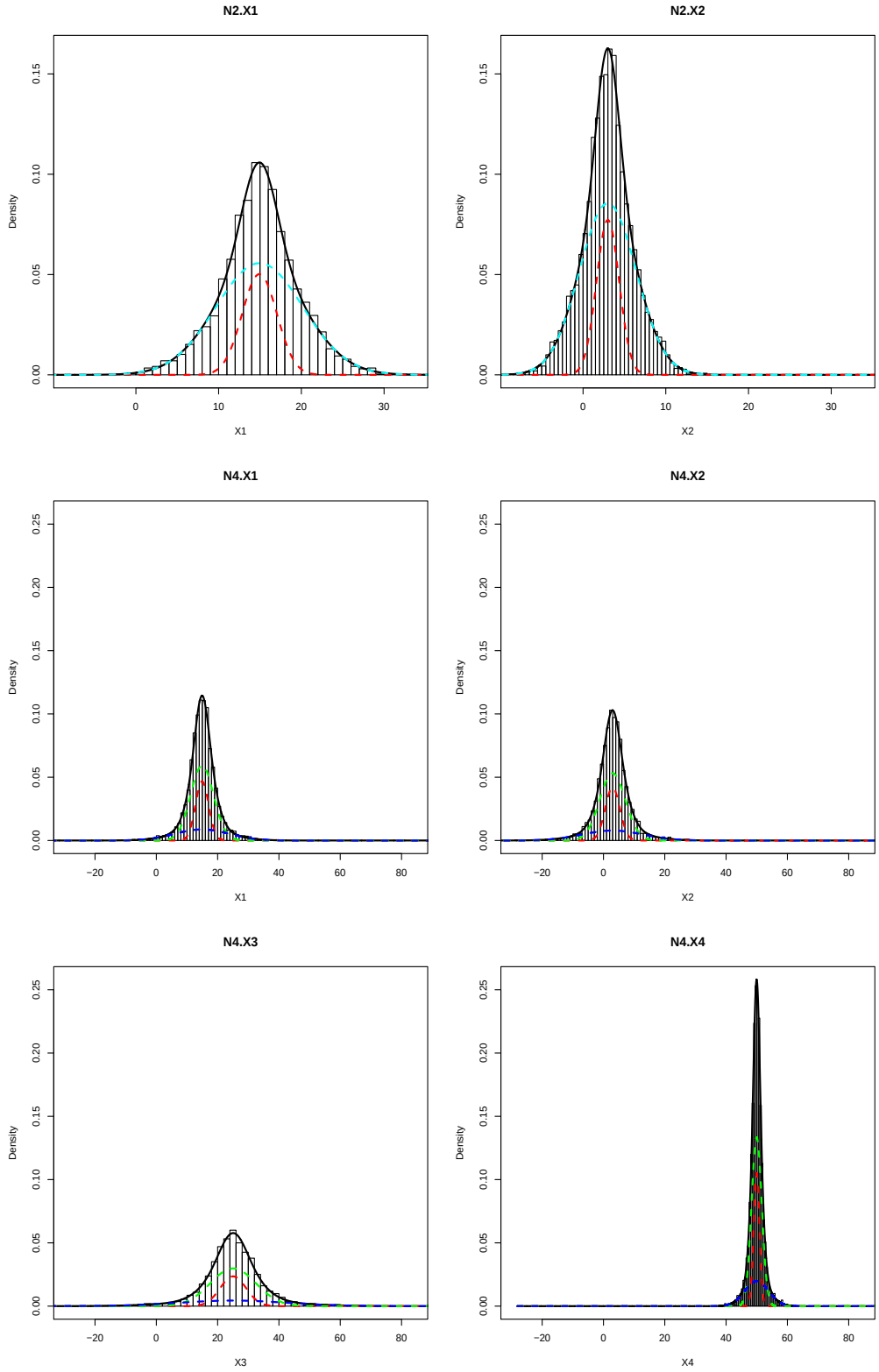

Figure 4.1: N2 and N4 marginal distribution and fitted density

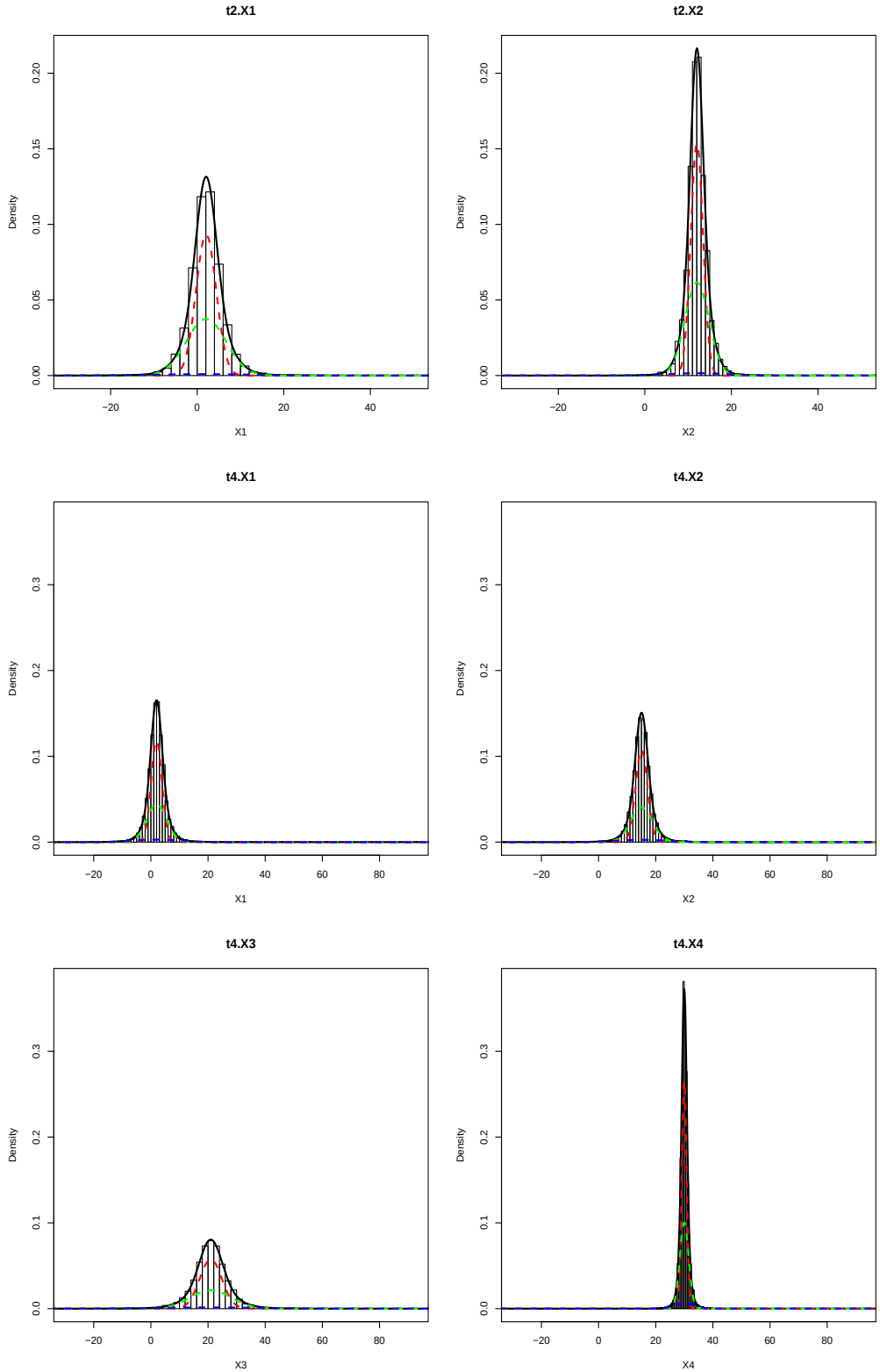

Figure 4.2: $t2$ and $t4$ marginal distribution and fitted density

4.5.2 Modeling multiple log returns and VaR estimation

In this section, we fit scale mixtures of multivariate normal distributions VaR model for multiple log-returns and estimate Value-at-Risk (VaR) of loss distribution.

Setup of portfolio and loss distribution

Suppose a British investor constructs a portfolio from period 01/01/1996 to 31/12/2003, which is consists of the following factors (assets),

- S_1, Financial Times 100 Shares Index (FTSE100), a share index of the 100 most highly capitalized UK companies listed on the London Stock Exchange.
- S_2, Standard and Poor's 500 (SP500), a capitalization index of the prices of 500 large-cap stocks traded in the United States.
- S_3, Swiss Market Index (SMI), Switzerland's twenty largest stock market index.

Since each of these assets are in different units: British pound (GBP), US dollars (USD) and Swiss francs (CHF), thus we still need another two factors,

- S_4, GBP/USD, exchange rate of GBP/USD
- S_5, GBP/CHF, exchange rate of GBP/CHF

to convert all of assets to GBP for our British investor.

Denote the number of shares of each asset by w_1, w_2 and w_3 for FTSE100, SP500 and SMI respectively. The value of the portfolio on time t, V_t, can be expressed by

$$V_t = w_1 S_{t,1} + w_2 S_{t,2} S_{t,4} + w_3 S_{t,3} S_{t,5}. \tag{4.24}$$

Thus, the loss distribution, L, is the random variable of daily loss of portfolio as a percentage. Loss of portfolio from day $t-1$ to day t as a percentage can be expressed by

$$L_t = \frac{-(V_t - V_{t-1})}{V_{t-1}}. \tag{4.25}$$

Suppose the portfolio weight (proportion of total value invested) for FTSE100, SP500 and SMI is 30 percent, 40 percent and 30 percent respectively, and it is fixed for all period. That is,

$$\begin{aligned} \frac{w_1 S_{t,1}}{V_t} &= 0.3, \\ \frac{w_2 S_{t,2} S_{t,4}}{V_t} &= 0.4, \\ \frac{w_3 S_{t,3} S_{t,5}}{V_t} &= 0.3 \end{aligned}$$

Linearized loss distribution

Linear approximation of loss distribution can express loss as a linear function of factor changes. It is very useful in the variance-covariance method, one of the VaR estimation methods that we will discuss later.

Refer to Equation 3.20, denote $X_{t,1}$, $X_{t,2}$, $X_{t,3}$, $X_{t,4}$ and $X_{t,5}$ as log returns at time t of FTSE100, SP500, SMI, GBP/USE and GBP/CHF respectively. The

loss from day $t-1$ to day t in percentage can be expressed by

$$\begin{aligned}
L_t =& \frac{-(V_t - V_{t-1})}{V_{t-1}} \\
=& -\frac{1}{V_{t-1}}\left[w_1(S_{t,1} - S_{t-1,1}) + w_2(S_{t,2}S_{t,4} - S_{t-1,2}S_{t-1,4}) + w_3(S_{t,3}S_{t,5} - S_{t-1,3}S_{t-1,5})\right] \\
=& -\frac{1}{V_{t-1}}[w_1(S_{t-1,1}\exp(X_{t,1}) - S_{t-1,1}) + w_2(S_{t-1,2}S_{t-1,4}\exp(X_{t,2})\exp(X_{t,4}) - S_{t-1,2}S_{t-1,4}) + \\
& \qquad w_3(S_{t-1,3}S_{t-1,5}\exp(X_{t,3})\exp(X_{t,5}) - S_{t-1,3}S_{t-1,5})] \\
=& -\frac{1}{V_{t-1}}[w_1 S_{t-1,1}(\exp(X_{t,1}) - 1) + w_2 S_{t-1,2}S_{t-1,4}(\exp(X_{t,2})\exp(X_{t,4}) - 1) + \\
& \qquad w_3 S_{t-1,3}S_{t-1,5}(\exp(X_{t,3})\exp(X_{t,5}) - 1)] \\
=& -\frac{w_1 S_{t-1,1}}{V_{t-1}}(\exp(X_{t,1}) - 1) - \frac{w_2 S_{t-1,2}S_{t-1,4}}{V_{t-1}}(\exp(X_{t,2} + X_{t,4}) - 1) \\
& \qquad - \frac{w_3 S_{t-1,3}S_{t-1,5}}{V_{t-1}}(\exp(X_{t,3} + X_{t,5}) - 1) \\
=& -0.3(\exp(X_{t,1}) - 1) - 0.4(\exp(X_{t,2} + X_{t,4}) - 1) - 0.3(\exp(X_{t,3} + X_{t,5}) - 1)
\end{aligned}$$

$$=1 - 0.3\exp(X_{t,1}) - 0.4\exp(X_{t,2} + X_{t,4}) - 0.3\exp(X_{t,3} + X_{t,5}). \tag{4.26}$$

By using a first order Taylor expansion, the exponential function can be approximated by

$$e^x \approx 1 + x. \tag{4.27}$$

Thus, daily loss in percentage, L_t, can be approximate by linearized loss L_t^*

$$\begin{aligned}
L_t \approx L_t^* &= 1 - 0.3(X_{t,1} + 1) - 0.4(X_{t,2} + X_{t,4} + 1) - 0.3(X_{t,3} + X_{t,5} + 1) \\
&= -0.3X_{t,1} - 0.4(X_{t,2} + X_{t,4}) - 0.3(X_{t,3} + X_{t,5}) \\
&= -0.3X_{t,1} - 0.4X_{t,2} - 0.3X_{t,3} - 0.4X_{t,4} - 0.3X_{t,5}.
\end{aligned} \tag{4.28}$$

Therefore, the linearized loss distribution is the random variable of L_t^*, which approximates the loss distribution L_t by a linear combination of log returns of each factor.

Variance-covariance method

In this section, we introduce a variance-covariance method, one of the methods for estimating VaR, which assume the linearized loss distribution is a good approximation for a loss distribution. The traditional variance-covariance method is using multivariate normal distribution and its important linear combination property.

The linearized loss distribution is the linear combination of log returns of each factor, which can be expressed by

$$L_t^* = \mathbf{a}^T \mathbf{X}_t. \tag{4.29}$$

If X_t has p-dimensional multivariate normal distribution $N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, any linear combination of its components has univariate normal distribution, thus

$$L_t^* \sim N(\mathbf{a}^T \boldsymbol{\mu}, \mathbf{a}^T \boldsymbol{\Sigma} \mathbf{a}), \tag{4.30}$$

where $\mathbf{a}^T = (-0.3, -0.4, -0.3, -0.4, -0.3)$ in this case. Therefore, referring to Equation (3.23), we can just estimate VaR in level α by computing the α quantile of the fitted univariate normal distribution of linearized loss distribution L_t^*.

By using the variance-covariance method, we reduce the multidimensional problem to one dimensional. It is easy to use, and calculation is also very simple.

There are some weakness of traditional variance-covariance method. Firstly, financial data usually has heavier tails than a normal distribution, so the assumption of multivariate normality may not hold. Secondly, it assumes the linearized loss distribution is a good approximation of the loss distribution, it may not hold for some particular data. We will show that it is a good approximation in this case later.

One alternative to the variance-covariance method is to replace the multivariate normal distribution with other heavy-tailed distributions, and these distributions must have a similar linear combination property as the multivariate normal distribution.

One option is a multivariate t-distribution. Since it also has the linear combination property as multivariate normal distribution. We now can assume log returns of each factor has multivariate t-distribution $t(\nu, \boldsymbol{\mu}, \boldsymbol{\Lambda})$, for any linear combination of its components, will also have univariate t-distribution, thus

$$L_t^* \sim t(\nu, \mathbf{w}^T\boldsymbol{\mu}, \mathbf{w}^T\boldsymbol{\Lambda}\mathbf{w}). \tag{4.31}$$

Therefore, again, we can estimate VaR in level α by computing the α quantile of the fitted univariate t-distribution of linearized loss distribution L_t^*.

The other option is our scale mixtures of multivariate normal distributions. Any linear combination of its components also have univariate scale mixture of normal distributions as we discussed in Section 4.2 and 4.3.1. Then we can estimate VaR by computing the α quantile of the fitted univariate scale mixture of normal distributions of linearized loss distribution L_t^*. Quantile estimation of univariate scale mixture of normal distributions by bisection method has been discussed in Section 3.4.2

Estimate VaR and assessment

In summary, the process of estimating VaR by using variance-covariance method can be listed in the following steps,

1. Fit multiple log returns by a particular model, such as multivariate normal distribution, multivariate t-distribution or scale mixture of multivariate normal distributions.

2. Construct linearized loss distribution model by linear combination property.

3. Estimate VaR in level α by computing the α quantile of linearized loss distribution model.

4. Compute Akaike information criterion (AIC) for each fitted model, and apply Kupiec Test (discussed in Section 3.5.2) on estimated VaR.

We fit multivariate normal distribution and multivariate t-distribution by maximum likelihood, and fit scale mixture of multivariate normal distributions by the CNM-MS algorithm.

Results

Table 4.2 shows the numerical summary of log returns for each factor, and Figure 4.4 displays the time series plot for these data. All of these data are almost centered at 0. Standard deviation for FSTE100, SP500 and SMI is larger than the other two exchange rate factors (GBP/USD and GBP/CHF). Almost all of the data are heavy-tailed, except GBP/CHF which is slightly short-tailed. SP500, SMI and GBP/USD are almost symmetric. FTSE100 is slightly negatively skewed, and GBP/CHF is slightly right skewed.

Data	median	mean	sd	kurtosis	skewness
FTSE100	0.00029	0.00001	0.01220	2.00399	-0.12819
SP500	0.00051	0.00043	0.01249	2.06430	-0.00494
SMI	0.00061	0.00022	0.01351	3.16519	-0.07410
GBP/USD	-0.00006	-0.00008	0.00476	1.51118	0.08228
GBP/CHF	-0.00038	-0.00018	0.00548	0.92255	0.22743

Table 4.2: Numerical summary of log returns for each factor

We first construct the linearized loss distribution to approximate the observed loss distribution. Table 4.3 shows the numerical summary for the observed loss distribution and the linearized loss distribution. It shows that both of them have almost the same median, mean, standard deviation and kurtosis, but different skewness. The linearized loss distribution has larger skewness than observed loss distribution, which determine the linearized loss distribution is slightly positive skewed after linearized.

Figure 4.3 shows both the histogram of linearized loss distribution and the loss distribution, and Q-Q plot of these two. Points of the Q-Q plot lies perfectly on the line $y = x$, determines that linearized loss distribution is extremely good

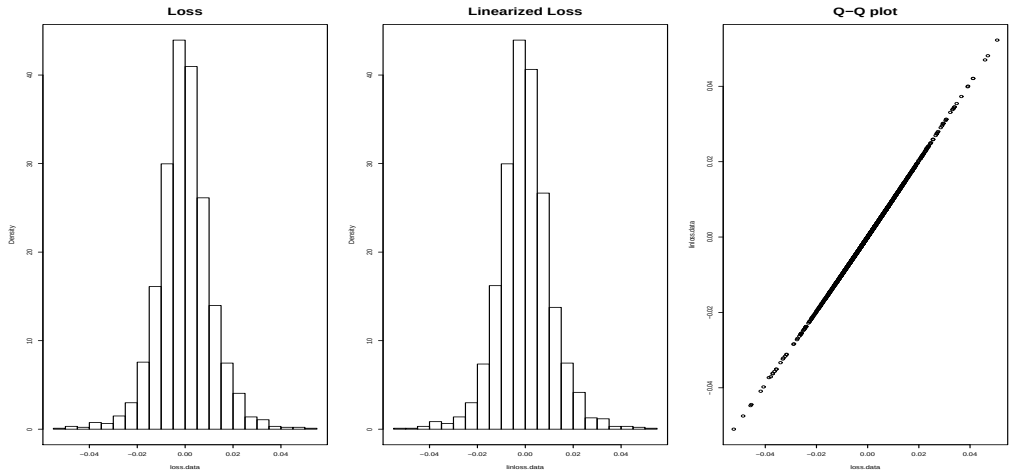

Figure 4.3: Loss and Linearized loss

approximation of loss distribution in this case. Variance-covariance method can be applied without concern.

	median	mean	sd	kurtosis	skewness
L	-0.00040	-0.00024	0.01099	2.13037	0.02688
L^*	-0.00037	-0.00015	0.01099	2.12964	0.10064

Table 4.3: Observed loss and linearized loss numerical summary

Table 4.4 shows MLE fitting result for each model. We fit a multivariate t-distribution with degrees of freedom 6.136, and fit a 3-components scale mixture of multivariate normal distributions. Figure 4.5-4.7 shows marginal plots for each multivariate model. Figure 4.5 shows these log returns are clearly not multivariate normally distributed since fitted density fit the data poorly. Figure 4.6 and 4.7 shows multivariate t-distribution and scale mixture of multivariate normal distribution have similar behavior and both fit the data very well.

Table 4.5 shows the Akaike information criterion (AIC) for each model, estimated VaR and Kupiec test p-value. Figure 4.8 performs the plot of fitted linearized loss distribution and estimated VaR.

For level 0.95, multivariate t-distribution model and scale mixture of multivariate normal distributions model both have similar estimated VaR, and esti-

Model	MLE
Mul.Normal	$\hat{\boldsymbol{\mu}}=(0.139, 4.250, 2.180, -0.806, -1.820) \times 10^{-4}$
	$\hat{\boldsymbol{\Sigma}}=\begin{pmatrix} 1.488 & 0.664 & 1.214 & 0.155 & -0.095 \\ 0.664 & 1.561 & 0.701 & 0.083 & -0.084 \\ 1.214 & 0.701 & 1.826 & 0.157 & -0.148 \\ 0.155 & 0.083 & 0.157 & 0.227 & 0.053 \\ -0.095 & -0.084 & -0.148 & 0.053 & 0.300 \end{pmatrix} \times 10^{-4}$
Mul.t	$\hat{\nu}$=6.136
	$\hat{\boldsymbol{\mu}}=(2.446, 5.885, 4.698, -0.698, -3.485) \times 10^{-4}$
	$\hat{\boldsymbol{\Lambda}}=\begin{pmatrix} 9.652 & 4.535 & 7.560 & 1.008 & -0.629 \\ 4.535 & 10.583 & 4.511 & 0.561 & -0.642 \\ 7.560 & 4.511 & 11.493 & 0.968 & -1.016 \\ 1.008 & 0.561 & 0.968 & 1.603 & 0.373 \\ -0.629 & -0.642 & -1.016 & 0.373 & 2.201 \end{pmatrix} \times 10^{-5}$
MSMN	$\hat{\boldsymbol{\mu}}=(2.535, 6.075, 4.968, -0.784, -3.486) \times 10^{-4}$
	$\hat{\boldsymbol{\pi}}=(0.252, 0.654, 0.093)$
	$\hat{\boldsymbol{\theta}}=(1.000, 3.094, 10.032)$
	$\hat{\boldsymbol{\Sigma}}=\begin{pmatrix} 4.362 & 2.021 & 3.409 & 0.457 & -0.275 \\ 2.021 & 4.797 & 2.001 & 0.252 & -0.280 \\ 3.409 & 2.001 & 5.197 & 0.440 & -0.455 \\ 0.457 & 0.252 & 0.440 & 0.730 & 0.172 \\ -0.275 & -0.280 & -0.455 & 0.172 & 0.994 \end{pmatrix} \times 10^{-5}$

Table 4.4: Model fitting

mated VaR for multivariate normal distribution model is higher than the other two. All of these three models have no evidence against null hypothesis for Kupiec test in 5% significant level.

For level 0.99, estimated VaR for multivariate normal distribution model (0.02541) has lowest value of all three models, and it has very strong evidence (p-value=0.0118) against null hypothesis for Kupiec Test. Estimated VaR for multivariate t-distribution model (0.02737) has highest value of all three models. Both of estimated VaR in level 0.99 for multivariate t-distribution model and scale mixtures of multivariate normal distributions have no evidence against the null hypothesis for Kupiec test.

Both multivariate t-distribution model and scale mixtures of multivariate nor-

		$\alpha = 0.95$		$\alpha = 0.99$	
Model	AIC	VaR	p-value	VaR	p-value
Mul.Normal	-64312.16	0.01792	1.00000	0.02541	0.01178
Mul.t	-65018.21	0.01686	0.37134	0.02737	0.26776
MSMN	-65022.34	0.01692	0.43379	0.02692	0.08394

Table 4.5: AIC, VaR Estimation and Kupiec Test p-value

mal distributions model have clearly better performance than multivariate normal distribution model. They both have no evidence against the null hypothesis for Kupiec Test, and also fit the data quite well. It is very difficult to judge which model is the best.

Since scale mixtures of multivariate normal distributions model has lowest AIC value (-65022.34), we claim that our scale mixtures of multivariate normal distributions model is the best model, and performs a very good estimated VaR.

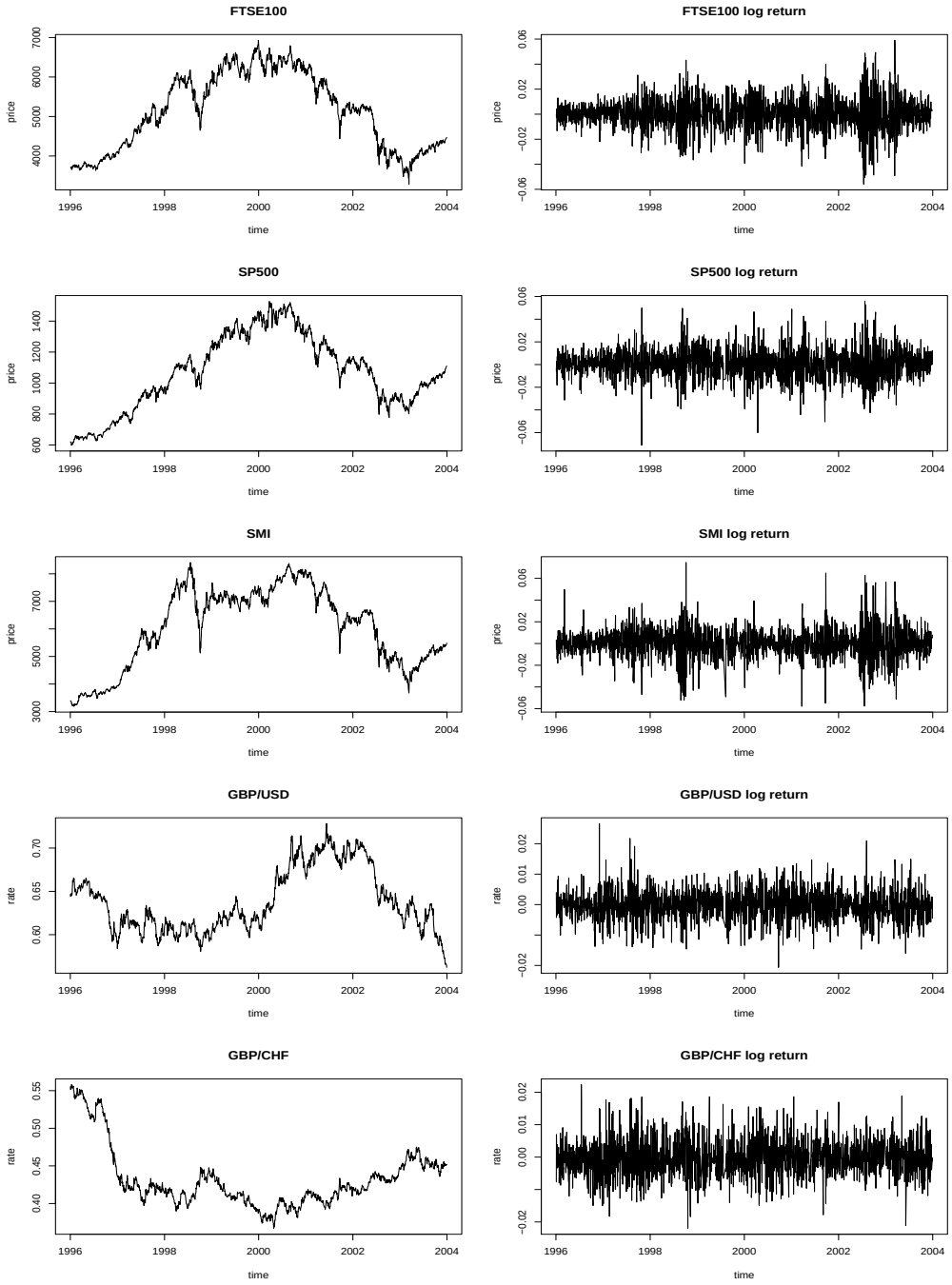

Figure 4.4: Raw data and log returns for each factor

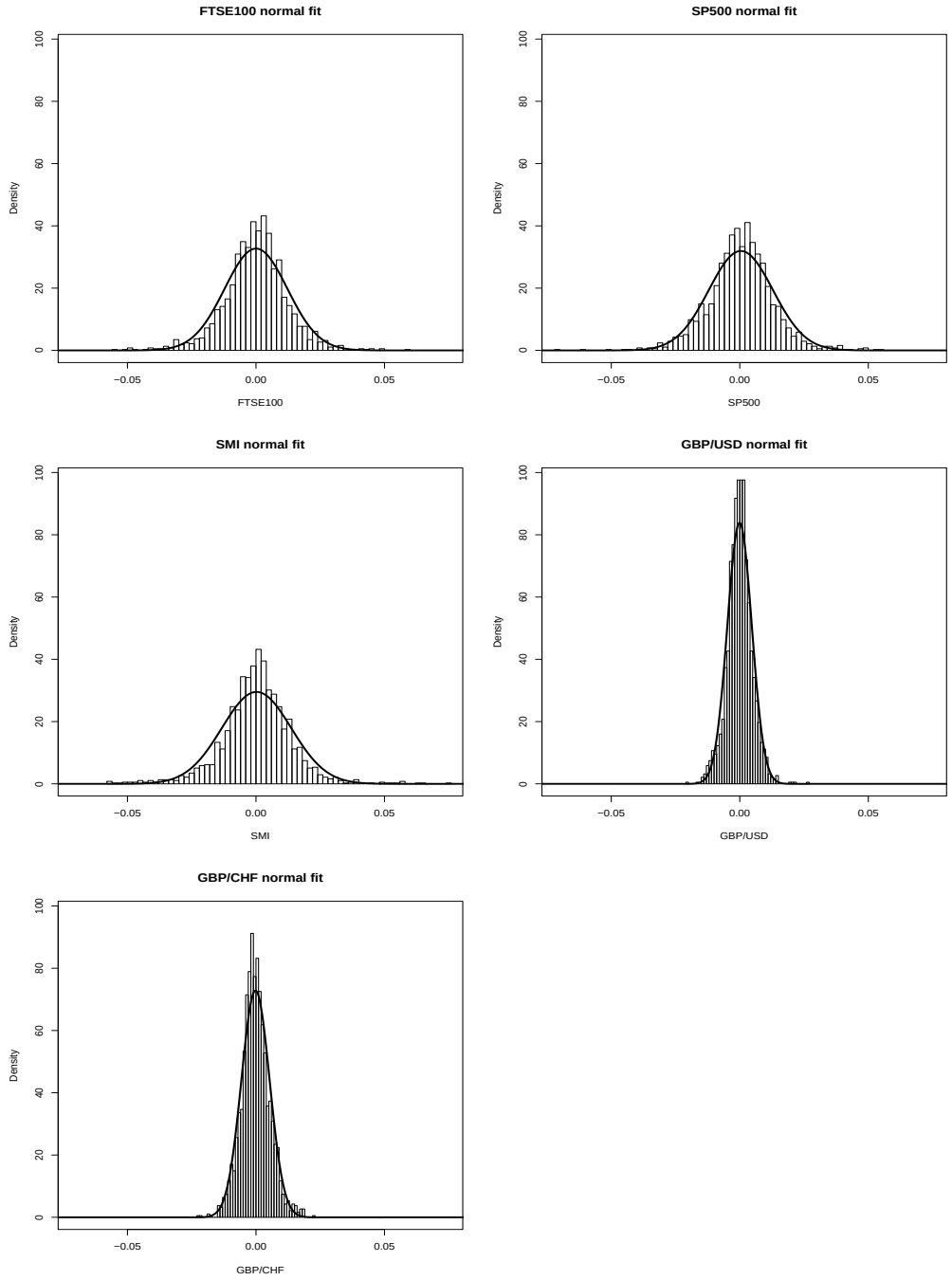

Figure 4.5: Marginal density for multivariate normal distribution model

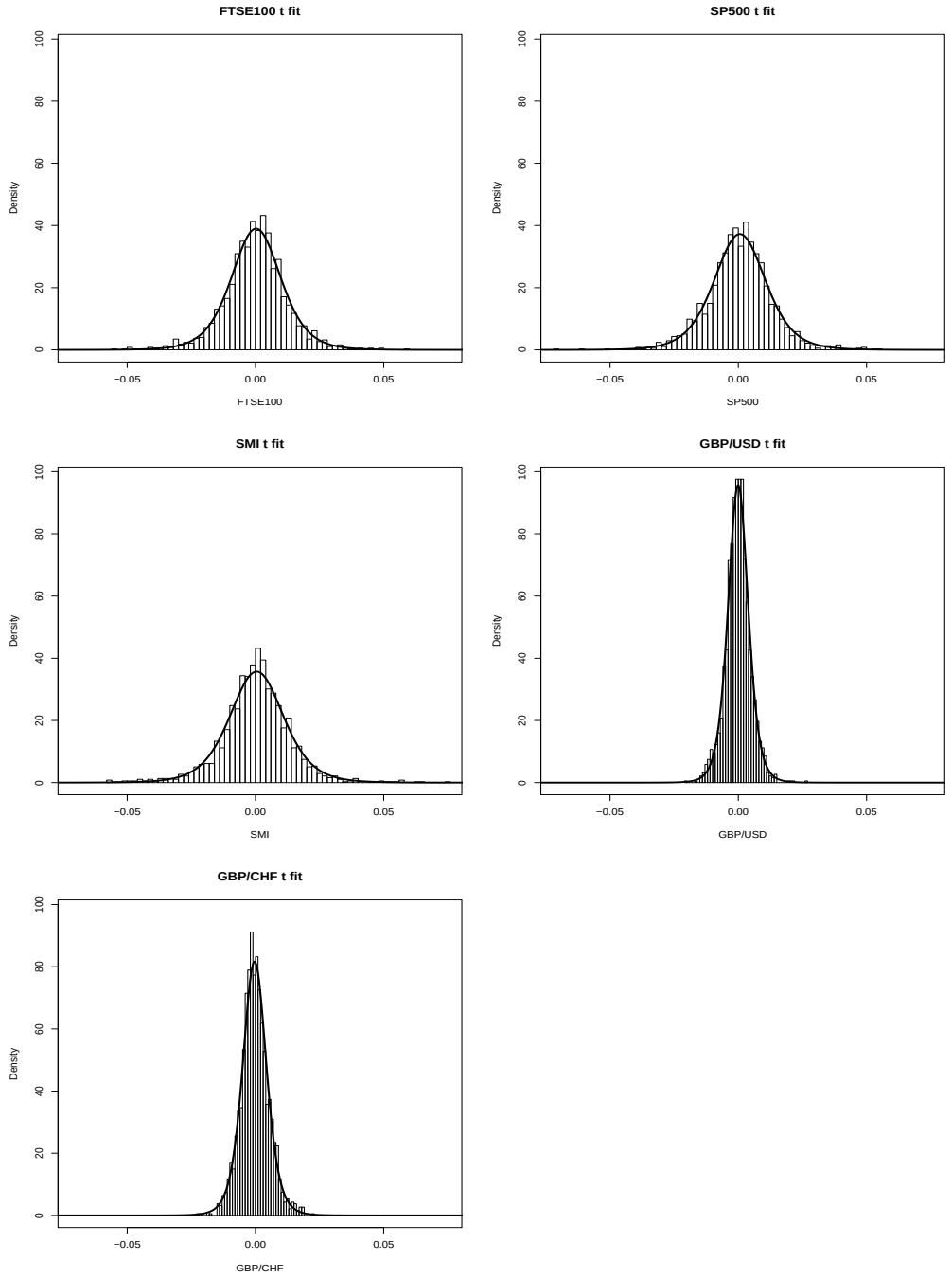

Figure 4.6: Marginal density for multivariate t distribution model

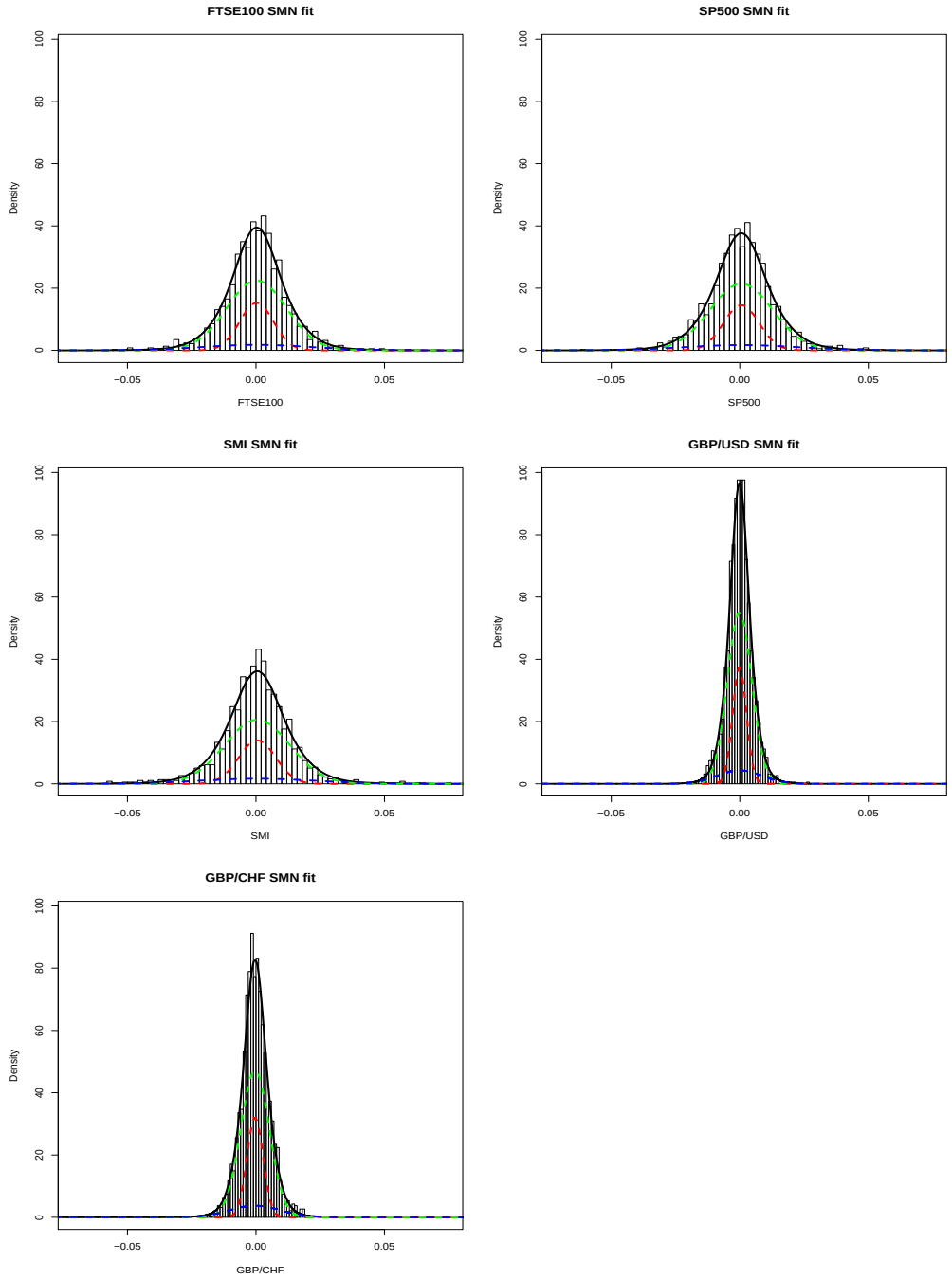

Figure 4.7: Marginal density for MSMN model

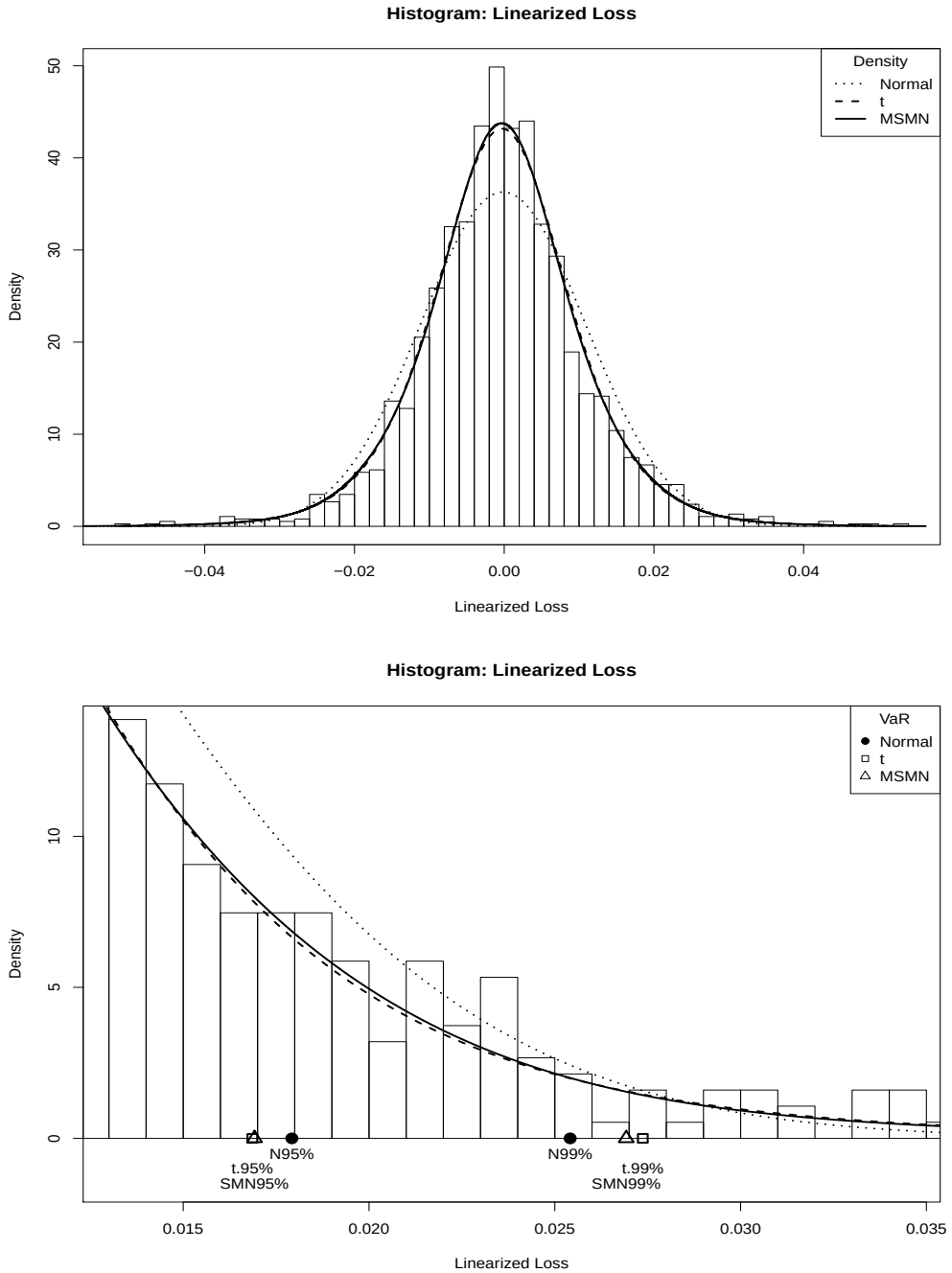

Figure 4.8: VaR estimation for each model

4.6 Summary

In this chapter, we discussed scale mixtures of multivariate normal distributions (MSMN), and represent using the CNM-MS algorithm with slight modification with the univariate case to get semiparametric MLE for MSMN. We also introduce some possible applications of MSMN, such as fitting simulated heavy-tailed data, modeling multiple log returns and Value-at-Risk(VaR) estimation.

Semiparametric MLE of MSMN by using the CNM-MS algorithm have extremely good performance for simulated data. Modeling multiple log returns and VaR estimation also have very good result. We also fit multivariate normal distribution model and multivariate t-distribution model as benchmark. We claim that our MSMN model is the best model since it has the lowest AIC.

Chapter 5

Conclusions

The research for this thesis focused on scale mixtures of normal distributions and their possible applications. This Chapter outlines the results of our research in Section 5.1, gives several possible further studies in Section 5.2 and conclusions in Section 5.3.

5.1 Summary

CNM

The constraint Newton method (CNM) is a very strong candidate to estimate nonparametric mixing distribution G, especially in time aspect (in terms of number of iteration); see, e.g., Wang (2007a). The advantage of the CNM algorithm is not only less computation time, but also is the nonparametric algorithm. By using the CNM algorithm, we avoid critical issues about the number of mixing components, which is the main issue in most of the present literature about mixture models.

The CNM algorithm uses a quadratic approximation to the log-likelihood function by using the Taylor series expansion on first and second derivatives (Hessian matrix). The problem for finding nonparametric mixing distribution

G is then converted to least square problem. There are many methods to solve this least square problem, Wang (2007a) suggests to use the method from Haskell and Hanson (1981), and Wang (2010) also suggests to use the Dax (1990) method to solve this least square problem for improvement. Both of these methods can be computed numerically by using the non-negative least squares (NNLS) algorithm from Lawson and Hanson (1974). Wang (2007b) provide the NNLS algorithm implementation in statistical software `R`.

The CNM algorithm finds relevant support sets by locating local maxima of the directional derivatives of the log-likelihood. It implements the univariate Newton method combined with the bisection method to locate local maxima. The CNM algorithm is a very fast and stable algorithm.

CNM-MS

Wang (2010) extended the CNM algorithm and developed three CNM-based algorithms (CNM-AP, CNM-PL and CNM-MS) to find semiparametric maximum likelihood for mixture models. We only use the CNM-MS in this thesis since it has quickest computation time. The CNM-MS algorithm is a very fast and stable algorithm. It involved the CNM algorithm for locate nonparametric mixing distribution G, and update all parameters by the BFGS method.

Scale mixtures of univariate normal distributions

Chapter 3 introduced, and explored scale mixtures of univariate normal distributions. We also discussed how to find semiparametric maximum likelihood of scale mixtures of univariate normal distributions by using the CNM-MS algorithm. We define relevant expressions in terms of log-likelihood, and mention some computational issues.

We explored possible applications of scale mixtures of normal distribution with or without covariates.

We first generate a few simulated datasets in different heavy-tailed distributions, and fit these data by using scale mixtures of normal distributions model with the CNM-MS algorithm. The result performs very well, consistent with the true model (contaminated normal distribution) and the fitting density also fits the data very well.

Modeling log returns and Value-at-Risk is another possible application without covariates. Three financial datasets had been chosen, and fitting the scale mixtures of normal distributions model with other models as benchmark. In general, the normal distribution model has poor performance, and the scale mixtures of normal distributions model has very good performance, as well as t-distribution model. We claim our scale mixtures of normal distributions model is the best model according to least AIC value. Note that all models for exchange rate dataset GBP/JPY still have some concerns at long-tail since it has slightly right skewed, but all our model are symmetric distributions.

We also discussed one possible application with covariates, the linear robust semiparametric regression. Our new approach is replacing traditional OLS normality error assumption with scale mixtures of normal distributions. The traditional OLS technique is very sensitive to outliers or heavy-tailed data. We resolve the heavy-tailed problem by replacing the linear regression error term to the scale mixtures of normal distributions.

The implementation of the linear robust semiparametric regression by using the CNM-MS algorithm is very similar to scale mixtures of normal distributions without covariates but with slight modification. We performed a simulation study with other existing linear robust regression methods as a benchmark. In general, every method perform well in certain cases, but our scale mixtures of normal distributions approach is the only one performs consistently well.

Scale mixtures of univariate normal distributions are very interesting and has very wide range possible heavy-tailed applications.

Scale mixtures of multivariate normal distributions

We extended univariate scale mixtures of normal distributions to multivariate case in Chapter 4. We introduced the basic definition of scale mixtures of multivariate normal distributions and discussed the implementation of the CNM-MS algorithm with some modification to find semiparametric maximum likelihood estimates.

The combination of the EM algorithm and the CNM-MS algorithm is used. We used the EM algorithm to update the common base variance-covariance matrix, and used the CNM-MS algorithm to update other parameters. The advantage of this modification is the EM algorithm update the common base variance-covariance matrix very fast and in fixed form. This reduce the multivariate problem to univariate, and it is easy to implement the CNM-MS algorithm.

We introduced two possible applications for scale mixtures of multivariate normal distributions. We first generated four heavy-tailed multivariate datasets with different distributions and dimensions, and fitting scale mixtures of multivariate normal distributions model. The result performed very good, all estimates are very close to true value, and marginal densities are all fit the datasets very well. We only performed marginal densities plot since it is very difficult to assess goodness of the model in multivariate case.

Modeling multiple log returns and VaR estimation are also been studied. We chose five financial datasets and build up a portfolio. We approximated loss distribution by using first a order Taylor expansion on the exponential function, and so called the linearized loss distribution. The variance-covariance method is used to estimate VaRs, by using the special linear combination property of multivariate normal distribution, multivariate t-distribution and scale mixtures of multivariate normal distributions.

For our portfolio, the linearized loss distribution is a very good approximation of the observed loss distribution. Multivariate normal distribution model fit

data very poorly, multivariate t-distribution model and scale mixtures of multivariate normal distributions both fit datasets very well. Multivariate normal distribution model has very poor performance at 99% level VaR, and multivariate t-distribution model and multivariate scale mixtures of normal distributions both have very good performance at all levels. We claim our scale mixtures of multivariate normal distributions is the best model according to least AIC value.

Scale mixtures of multivariate normal distributions have very wide range possible heavy-tailed applications, especially in financial areas (since most of financial data are heavy-tailed).

5.2 Possible further research

This section presents some ideas for possible further research that we identified during the study.

Skewed data

Scale mixtures of normal distributions is symmetric distribution. It may have bad performance if data is skewed. In the sense of semiparametric robust linear regression, that is skewed error. One possible further research is replace mixtures of normal distribution with other skewed distributions, such as log-normal distribution.

Log-normal distribution is a probability distribution of a random variable if its logarithm is normally distributed. It may have very high skewness by tunning the mean and the standard deviation of the model. Therefore it is reasonably to replace symmetrical normal mixtures to unsymmetrical log-normal mixtures for skewed data.

Nonlinear robust regression

We only studied semiaprametric linear robust regression with scale mixtures of normal distributions error in this thesis. Nonlinear robust regression is another possible further research.

Nonlinear regression is similar to linear regression, but response variable is modeled by nonlinear relationship of model parameters instead of linear combination. In practice, data may not have linear combinations with model parameters, thus linear regression technique is not suitable anymore.

Therefore it is reasonable to assume the error term of nonlinear regression is some heavy-tailed distributions in robustness study. Lange et al. (1989) study linear and nonlinear robust regression using t-distribution. We can replace t-distribution with scale mixtures of normal distributions and build another rich family of nonlinear regression.

Conditional method

We only studied unconditional method in financial-related applications. Unconditional method is assume there is no time-effect in the data, such as log returns time series is the stationary time series and portfolio weight is fixed for all time period. We all used unconditional method in this thesis.

Conditional method is opposite to unconditional method. The time-effect of data is critical importance in the analysis. The log returns time series is not the stationary time series anymore, and portfolio weight may vary by time. Thus, it is possible to add the time-effect to the model.

5.3 Conclusions

Scale mixtures of normal distributions model forms a rich and flexible family for heavy-tailed distribution. Great CNM-related algorithms reducing the difficulty of analysis, it is unnecessary to concern about the number of components in

mixture model, and they are very fast and stable (compared with several other algorithms).

Through a number of simulation and case studies, scale mixtures of normal distributions model have very good performance consistently, and even better than t-distribution model in most of case. We claim that scale mixtures of normal distributions model is very useful, and is a very strong candidate in many heavy-tailed or robustness analysis.

Appendix A - source code

This appendix provides source code for implementation of scale mixtures of multivariate normal distributions using the CNM algorithm, and the CNM-based (CNM-AP, CNM-PL and CNM-MS) algorithms. Note that scale mixtures of univariate normal distributions is just the special case if we set dimension to 1.

The implementation is using statistical programming language `R` version 2.10.0 (R Development Core Team, 2009). The source code of CNM algorithm, and the CNM-based (CNM-AP, CNM-PL and CNM-MS) algorithms are provided by Dr Yong Wang.

Scale mixtures of multivariate normal distributions, `msmn`

The `R` class `msmn` depends on the CNM algorithm, and the CNM-based (CNM-AP, CNM-PL and CNM-MS) algorithms provide by Dr Yong Wang.

Code

```
# ======================================== #
# Multivariate Scale Mixture Normal        #
# ======================================== #

library(mvtnorm)

## constructor
```

```
msmn = function(x, sigma=NULL){
  if(!is.numeric(x)) stop("x must be numeric")
  if(is.vector(x)) x = matrix(x)
  if(class(x)!="matrix") stop("x must be vector or matirx")
  if(ncol(x)==1) sigma=matrix(1)
  else sigma=cov(x)
  data = list(x=x, sigma=sigma, sigma.inv=solve(sigma))
  class(data) = "msmn"
  data
}

## update sigma
setSigma = function(x, sigma){
  if(ncol(sigma)!=nrow(sigma)) stop("sigma must be squared matrix")
  if(ncol(sigma)!=dim(x)) stop("dimension of sigma must be same as dim(x)")
  if(class(try(solve(sigma)))=="try-error") stop("sigma is singular")

  x$sigma = sigma
  x$sigma.inv = solve(sigma)
  x
}

'[.msmn' = function(obj, i) msmn(x=obj$x[i,], sigma=obj$sigma)
mean.msmn = function(obj) apply(obj$x, 2, mean)
dim.msmn = function(obj) ncol(obj$x)
length.msmn = function(obj) nrow(obj$x)
cov.msmn = function(obj) cov(obj$x)

## information of 1-D msmn object,
## include sample mean, sample variance, fitted mean, fitted variance
info.msmn = function(x, beta, mix){
  if(dim.msmn(x)==1){
    tab = matrix(ncol=2, nrow=2)
    dimnames(tab) = list(c("Sample", "Estimate"), c("mean", "var"))
    tab[1,] = c(mean.msmn(x), cov.msmn(x))
    tab[2,] = c(beta, sum(mix$pt*mix$pr))
  } else{
    stop("dimension not 1")
  }
  tab
}
```

```
## generate random number of msmn object
rmsmn = function(n=100, mu=3,
                 mix=dden(pr=c(.7,.3), pt=c(3, 8)), sigma=matrix(1)){
  data = numeric(0);
  for(i in 1:length(mix$pr)){
    ni = ceiling((n*mix$pr[i]))
    data = rbind(data, rmvnorm(n=ni, mean=mu, sigma=mix$pt[i]*sigma))
  }
  data = data[sample(1:n),]
  msmn(data, sigma)
}

## cumulative distribution function of msmn
pmsmn = function(x, beta, mix){
  n = length(x)
  y = numeric(n)
  for(i in 1:n)
    y[i] = sum(mix$pr*pnorm(x[i], beta, sqrt(mix$pt)))
  y
}

## quantile function of msmn object
qmsmn = function(p, beta, mix){
  n = length(p)
  y = numeric(n)
  SD = sqrt(sum(mix$pr*mix$pt))
  for(i in 1:n){
    r = beta + c(-100*SD, 100*SD)
    while(TRUE){
      m = pmsmn(mean(r), beta, mix)
      if(abs(m-p[i])<1e-15) break
      if(p[i]<m) r = c(r[1], mean(r))
      else r = c(mean(r), r[2])
    }
    y[i] = mean(r)
  }
  y
}

## density of msmn object
```

```
dmsmn = function(x, mu, mix, sigma=x$sigma){
  d = numeric(length(x))
  for(i in 1:length(mix$pr)){
    temp = mix$pr[i]*dmvnorm(x$x, mean=mu, sigma=mix$pt[i]*sigma)
    if(all(temp==0)){
print(paste("(pr, pt) =(", mix$pr[i], ",", mix$pt[i], ")"))
warning("precision error?")
}
    d = d + temp
  }
  d
}

## range of mixing variances
range.msmn = function(x, ...){
  if(dim.msmn(x)==1){
    lower = cov.msmn(x)*1e-8
    upper = cov.msmn(x)*1e8
  }
  else{
    lower = 1e-8
    upper = 1e8
  }
  c(lower, upper)
}

## initial value
initial.msmn = function(x, beta=NULL, mix=NULL, kmax=NULL) {
  if(is.null(beta)) beta = mean(x)
  if(is.null(kmax)) kmax = 10
  if(is.null(mix) || is.null(mix$pt)){
    pt.rep = seq(range(x, beta)[1], range(x, beta)[2], length.out=10)
    mix = dden(unique(quantile(pt.rep, p=seq(0,1,len=kmax), type=1)))
  }
  list(beta=beta, mix=mix)
}

## validation
valid.msmn = function(x, beta, mix){
  all(mix$pt > 0) &&
  all(mix$pt >= range(x)[1]) &&
```

```
  all(mix$pt <= range(x)[2])
}

## log likelihood expression
logd.msmn = function(x, beta, pt, which=c(1,0,0,0)){
  p = length(beta)
  dl = vector("list", 4)
  d = mahalanobis(x$x, beta, x$sigma)

  names(dl) = c("ld","db1","dt1","dt2")
  if(which[1] == 1){
    dl$ld = outer(rep(-1/2, nrow(x$x)), log((2*pi*pt)^p*det(x$sigma)), "*") -
            outer(d, 1/(2*pt), "*")
  }
  if(which[2] == 1) {
    dl$db1 = array(dim=c(nrow(x$x), length(pt), length(beta)))

    x.temp = t(x$sigma.inv%*%t(sweep(x$x, 2, beta)))

    for(i in 1:length(pt))
      dl$db1[,i,] = x.temp/pt[i]
  }
  if(which[3] == 1)
    dl$dt1 = outer(rep(p, nrow(x$x)), -1/(2*pt), "*") +
             outer(d, 1/(2*pt^2), "*")
  if(which[4] == 1)
    dl$dt2 = outer(rep(p, nrow(x$x)), 1/(2*pt^2), "*") -
             outer(d, 1/(pt^3), "*")
  dl
}

## find new sigma during the EM algorithm
newSigma = function(x, beta, mix, sigma){
  p = matrix(nrow=length(x), ncol=length(mix$pt))
  z = dmsmn(x, beta, mix, sigma)

  for(k in 1:ncol(p))
    p[,k] = mix$pr[k]*dmvnorm(x$x, mean=beta, sigma=mix$pt[k]*sigma)

  p = p/z
  sigma.new = 0
```

```
  p.new = sweep(p, 2, mix$pt, "/")
  if(is.vector(p.new)) p.new = matrix(p.new)
  else p.new = t(p.new)

  x.new = t(sweep(x$x, 2, beta))

  for(i in 1:nrow(p)){
    sigma.new = sigma.new + x.new[,i]%*%t(x.new[,i])*sum(p.new[,i])
  }

  sigma.new/length(x)
}

## update sigma using the EM algorithm
updateSigma = function(x, beta, mix, sigma, tol=1e-6){
  while(TRUE){
    sigma.new = newSigma(x, beta, mix, sigma)
    z = abs(sum(log(dmsmn(x, beta, mix, sigma.new))) -
            sum(log(dmsmn(x, beta, mix, sigma))))
    sigma = sigma.new
    if(z<tol) break
  }
  sigma
}

## find semiparametric maximum likelihood,
## using the combination of the EM algorithm and the CNM-based algorithm
msmnFit = function(x, phi=initial(x), cnmFUN=cnmms, plot="g", tol=1e-10){
  if(dim.msmn(x)==1){
    phi = cnmFUN(x, plot=plot)
    phi$mix = truncMix(x, phi$beta, phi$mix)
  } else{
    phi$sigma = x$sigma
    while(TRUE){
      print(phi)
      sigma.new = updateSigma(x, phi$beta, phi$mix, phi$sigma, tol=tol)
      x = setSigma(x, sigma.new)
      print("Sigma Updated :")
      print(x$sigma)

      fit = cnmFUN(x, init=list(beta=phi$beta, mix=phi$mix), plot=plot)
```

```
      #fit = cnmFUN(x, plot=plot)

      phi.new = list(beta=fit$beta, mix=fit$mix, sigma=sigma.new)
      print("(beta, mix) Upated")

      z = abs(sum(log(dmsmn(x, phi$beta, phi$mix, phi$sigma))) -
              sum(log(dmsmn(x, phi.new$beta, phi.new$mix, phi.new$sigma))))

      print(cat("difference of ll is", z, "\n"))
      phi = phi.new
      phi$mix = truncMix.msmn(x, phi$beta, phi$mix)

  if(max(phi$mix$pt)==max(range(x)) || min(phi$mix$pt)==min(range(x)))
      if(max(phi$mix$pt)/min(phi$mix$pt)<max(range(x))/min(range(x))){
        phi$sigma = phi$sigma*min(phi$mix$pt)
        phi$mix$pt = phi$mix$pt/min(phi$mix$pt)
      }
      print(phi)
      if(z<tol) break
    }
    phi$sigma = phi$sigma*min(phi$mix$pt)
    phi$mix$pt = phi$mix$pt/min(phi$mix$pt)
  }
  phi$ll.max = sum(log(dmsmn(x, phi$beta, phi$mix, phi$sigma)))
  phi
}


## d=mahalanobis distance
## plot mahalanobis distance for bivariate normal distribution
plot.ellipse = function(d, mu, sigma, col="red", lwd=3, add=FALSE, ...){
n=1e3
xy = matrix(ncol=length(mu), nrow=n)
theta = seq(0, 2*pi, length.out=n)
eg = eigen(sigma)
for(i in 1:n)
xy[i,] = d * eg$vectors %*% (sqrt(eg$values)*c(cos(theta[i]),
                                  sin(theta[i]))) + mu

if(add) lines(xy[,1], xy[,2], col=col, lwd=lwd, ...)
else plot(xy, type="l", col=col, lwd=lwd, ...)
```

```
}

## plot for msmn object, only works for 1-D and 2-D
## generate density plot in 1-D case,
## generate mahalanobis distance plot in 2-D case
msmn.plot = function(mu, mix, d=1, sigma=NULL, add=FALSE, each=TRUE,
            lwd=2, leg=TRUE, ...){
  if(d==1){
    if(each && lwd==1) stop("lwd cannot be 1 when each=TRUE")

    if(length(mu)!=1) stop("mu is not 1-D")
x.range = mu + c(-1, 1)*sqrt(max(mix$pt))*20
x = seq(x.range[1], x.range[2], length.out=1e5)
k = length(mix$pt)
y = matrix(nrow=length(x), ncol=k)
for(i in 1:k)
y[,i] = mix$pr[i]*dnorm(x, mu, sqrt(mix$pt[i]))

y = cbind(y, apply(y, 1, sum))

if(!add){
plot.new(); plot.window(xlim=x.range, ylim=c(0, max(y)))
box(); axis(1); axis(2)
}

lines(x, y[,k+1], lwd=lwd, ...)

    if(each){
color = rainbow(k)
      for(i in 1:k)
        lines(x, y[,i], lty=2, lwd=2, col=color[i])

      if(leg){
      legend("topright", legend=paste("(",signif(mix$pt,4),", ",
              signif(mix$pr, 4),")",sep=""),
              title=expression(theta),
              lwd=lwd-1, col=color, lty=2)
      }
    }

   }else if(d==2){
```

```
    if(length(mu)!=2) stop("mu is not 2-D")
    k = length(mix$pt)
    color = rainbow(k)
    for(i in k:1){
      plot.ellipse(d=1, mu, mix$pt[i]*sigma, col=color[i], lty=2, add=add)
      add = TRUE
    }

    if(leg){
      legend("topright", legend=paste("(",signif(mix$pt,4),", ",
              signif(mix$pr, 4),")",sep=""),
              title=expression(theta),
              lwd=2, col=color, lty=2)
    }
   }
}
```

Bibliography

D. F. Andrews and C. L. Mallows. Scale mixtures of normal distributions. *Journal of the Royal Statistical Society, Series B*, 36:99–102, 1974.

A. Dax. The smallest point of a polytope. *Journal of Optimization Theory and Applications*, 64(2):429–432, 1990.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1): 1–38, 1977.

R. Fletcher. *Practical Methods of Optimization*. Wiley, New York, 2nd edition, 1987.

K. H. Haskell and R. J. Hanson. An algorithm for linear least squares problems with equality and nonnegativity constraints. *Mathematical Programming*, 21(1):98–118, 1981.

R. J. Hathaway. A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *Annals of Statistics*, 13(2):795–800, 1985.

P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

J. Kiefer and J. Wolfowitz. Consistency of the maximum-likelihood estimator in the presence of infinitely many incidental parameters. *Annals of Mathematical Statistics*, 27(4):887–906, 1956.

P. H. Kupiec. Techniques for verifying the accuracy of risk measurement models. *Journal of Derivatives*, 3:73–84, 1995.

N. Laird. Nonparametric maximum likelihood estimation of a mixing distribution. *Journal of the American Statistical Association*, 73(364):805–811, 1978.

K. L. Lange, R. J. A. Little, and J. M. G. Taylor. Robust statistical modeling using the t distribution. *Journal of the American Statistical Association*, 84(408):881–896, 1989.

C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Prentice Hall, Englewood Cliffs, N.J., 1974.

M. L. Lesperance and J. D. Kalbfleisch. An algorithm for computing the nonparametric mle of a mixing distribution. *Journal of the American Statistical Association*, 87(417): 120–126, 1992.

B. Lindsay. *Mixture Models: Theory, Geometry and Applications*. Institute of Mathematical Statistic, 1995.

B. G. Lindsay. The geometry of mixture likelihoods: A general theory. *Annals of Statistics*, 11(1):86–94, 1983.

G. J. McLachlan and K. E. Basford. *Mixture models: Inference and Applications to Clustering*. New York: Marcel Dekker, 1988.

D. Peel, W. J. Whiten, and G. J. Mclachlan. Fitting mixtures of kent distributions to aid in joint set identification. *Journal of the American Statistical Association*, 96:56–63, 2001.

P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley, 1987.

Y. Wang. On fast computation of the non-parametric maximum likelihood estimate of a mixing distribution. *Journal of the Royal Statistical Society, Series B*, 69:185–198, 2007a.

Y. Wang. lsei: Least squares solution under equality and inequality constraints, `R` package version 1.02, `http://www.stat.auckland.ac.nz/~yongwang/`. 2007b.

Y. Wang. Maximum likelihood computation for fitting semiparametric mixture models. *Statistics and Computing*, 20(1):75–86, 2010.

M. West. Outlier models and prior distributions in bayesian linear regression. *Journal of the Royal Statistical Society, Series B*, 46:431–439, 1984.

M. West. On scale mixtures of normal distributions. *Biometrika*, 74:646–648, 1987.