Uncertainty in Risk Modelling

by

Harjas Singh

Bachelor of Science, Ryerson, 2017

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the program of

Applied Mathematics

Toronto, Ontario, Canada, 2018

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Uncertainty in Risk Modelling

Master of Science, 2018

Harjas Singh

Applied Mathematics

Ryerson University

Abstract

In this thesis, we explore the uncertainty issues in risk modelling arising from the different approaches proposed in the literature and currently being used in the industry.

The first type of methods that we discuss assume that the returns of the stocks follows a generalized hyperbolic distribution. Data is calibrated by the Expectation-Maximization (EM) algorithm in order to estimate the parameters in the underlying distribution. Once we have the parameters, we estimate the Value at Risk (VaR) and Expected Shortfall (ES) by using Monte Carlo simulations.

Furthermore, we calibrate data to different copulas, including the Gauss Copula, the t Copula and the Gumbel Copula for estimation of VaR and ES using these copula structures.

The results from both methods are then compared. It can be concluded that uncertainty issues in risk modelling are very significant and can be troublesome as the values of the same risk measure computed using different methods demonstrate great oscillations.

Acknowledgements

I would first like to thank my thesis advisor Dr. Niushan Gao of the Department of Mathematics at Ryerson University. The door to Dr. Gao's office was always open whenever I ran into a trouble spot or had a question about my research or writing. Our discussions were always fruitful and constructive feedback was constantly provided. I would also like to acknowledge Dr. Foivos Xanthos of the Department of Mathematics at Ryerson University as the co-supervisor of this thesis. I am gratefully indebted to his support.

I must express my very profound gratitude to my parents and my sister for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Finally, I would like to thank God, who has always showered me with His blessings.

Harjas Singh

Dedication

In every age
There exist quiet heroes
Who, in ther selfless devotion,
lay aside their own needs and comforts
and consecrate their energies and talents
to healing the wounds of a troubled humanity.
To these rare and often unhonored souls
this thesis is humbly dedicated.

Table of Contents

List of Tables

List of Figures

List of Appendices

Chapter 1

Introduction

According to the Cambridge dictionary, risk is"the possibility of something bad happening." In financial terms, risk is defined as "the potential for financial loss and uncertainty about its extent" [15]. The crucial part for companies and regulators is precisely the extent of risk. Let us first begin with some definitions in order to quantify this risk.

1.1 Risk Measure

1.1.1 Basic Properties

Mathematically speaking, a risk measure is defined as a mapping from a set of random variables to the real numbers. In practice, the random variables are generally representative of portfolio returns. We represent a risk measure associated with a random variable X as $\rho(X)$. A risk measure $\rho : \mathcal{L} \to \mathbb{R} \cup \{+\infty\}$ has the following properties each with its own financial interpretation [3]:

- Normalized, i.e.

$$\rho(0) = 0.$$

  From a financial perspective, normalization is equivalent to saying that a portfolio

with no holding has zero risk.

- Translation invariance, i.e.

$$\text{If } a \in \mathbb{R} \text{ and } X \in \mathcal{L}, \text{ then } \rho(X + a) = \rho(X) - a.$$

From a financial perspective, translation invariance says that adding a fixed amount of capital to the portfolio will decrease the risk of the portfolio by the same amount.

- Monotonicity, i.e.

$$\text{If } X_1, X_2 \in \mathcal{L} \text{ and } X_1 \leq X_2, \text{ then } \rho(X_2) \leq \rho(X_1).$$

Viewed from the lens of finance, monotonicity has the interpretation that if a portfolio performs better than another one almost surely, then the first portfolio has less risk.

- Law invariance under $\mathbb{P}$, i.e.

$$\text{If } X_1, X_2 \in \mathcal{L} \text{ have the same distribution with respect to P, then } \rho(X_1) = \rho(X_2).$$

Law-invariant risk measures allot the same level of risk to financial portfolios that have an identical distribution with respect to $\mathbb{P}$ a priori. In reality, it is the most pervasive type of risk measures that are used. Principally, any risk measure whose computation relies on statistical methods and thus depends on distributions of random variables must be law-invariant.

- Sub-additivity, i.e.

$$\text{If } Z_1, Z_2 \in \mathcal{L}, \text{ then } \rho(Z_1 + Z_2) \leq \rho(Z_1) + \rho(Z_2).$$

This alludes to the principle commonly known as the diversification effect in finance, i.e. a portfolio that are well diversified carry lower risk than their counterparts. The idea is that the bad events for different assets in the portfolio may not happen simultaneously.

- Positive homogeneity, i.e.

$$\text{If } \alpha \geq 0 \text{ and } Z \in \mathcal{L}, \text{ then } \rho(\alpha Z) = \alpha\rho(Z).$$

This implies that the risk scaling is linear, i.e. doubling your financial position doubles risk.

- Convexity, i.e.

$$\text{If } Z_1, Z_2 \in \mathcal{L} \text{ and } \lambda \in [0,1] \text{ then } \rho(\lambda Z_1 + (1-\lambda)Z_2) \leq \lambda\rho(Z_1) + (1-\lambda)\rho(Z_2).$$

Convexity incorporates both sub-additivity and positive homogeneity and states that a linear combination (with non-negative weights) of two financial positions has less risk than the corresponding linear combination of risks associated with the financial positions.

Not all properties are satisfied by all risk measures. In particular, a risk measure ρ that satisfies monotonicity, sub-additivity, homogeneity, and translation invariance is called a coherent risk measure [3].

1.1.2 Examples of Risk Measures

A commonly used example of a risk measure is Value-at-Risk or VaR. Widely believed to be developed by JP Morgan during the late 1980's [2], VaR estimates the loss of a set of investments, given a certain probability and time interval. Mathematically speaking,

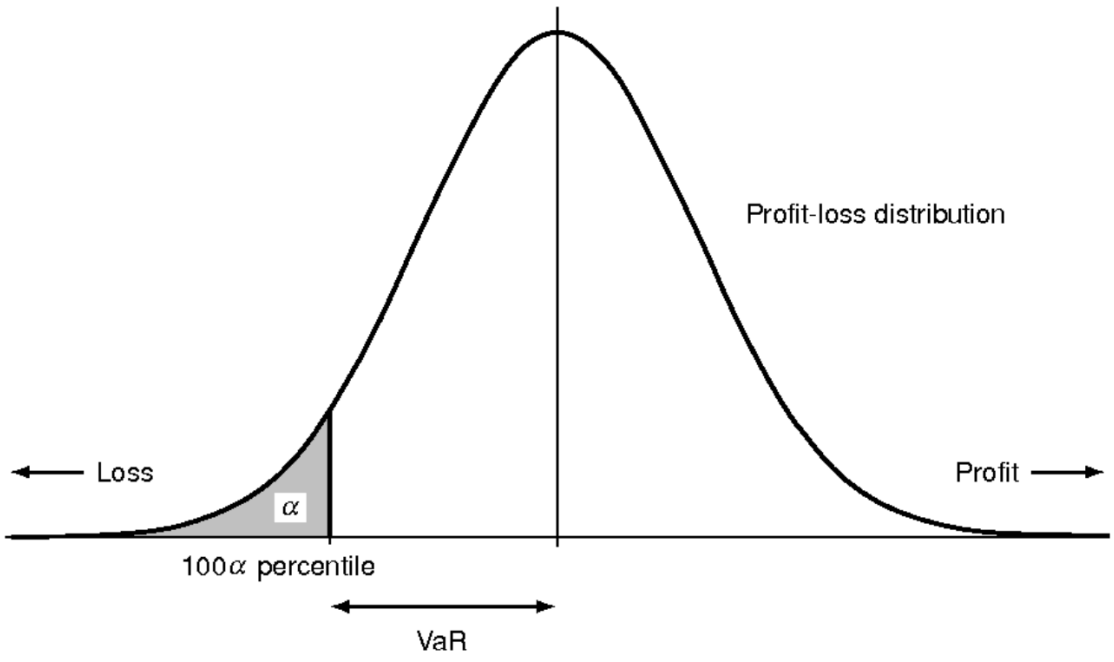

Figure 1.1: Graph highlighting Value-at-Risk.

if $\alpha \in (0,1)$ represents the level of confidence that we seek, then

$$\begin{aligned} \mathrm{VaR}_\alpha(X) &= -\inf\left\{x \in \mathbb{R} : F_X(x) > \alpha\right\} && (1.1)\\ &= F^{\leftarrow}_{-X}(1-\alpha), && (1.2)\\ &= \inf\{m \in \mathbb{R} : \mathbb{P}(X+m \leq 0) \leq \alpha\}, && (1.3) \end{aligned}$$

where F_X is the cumulative distribution function (cdf) associated with the random variable X and $F^{\leftarrow}_{-X}$ is the generalized inverse of the cumulative distribution function associated with -X. Hence, $\mathrm{VaR}_\alpha(X)$ is the minimal amount of cash that needs to be raised and added to the position X to make it "acceptable".

VaR has remained the industry standard for Basel regulators till recently, but it is

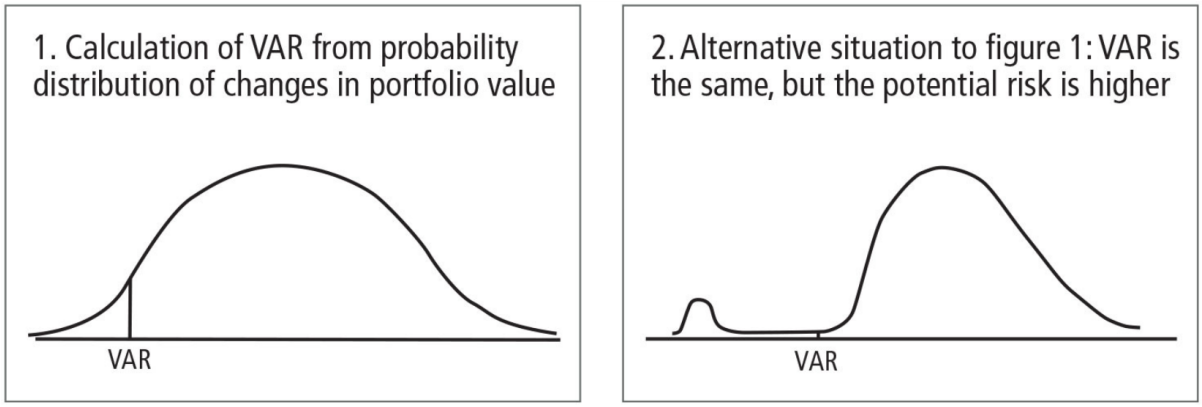

Figure 1.2: Graph highlighting shortcomings of VaR (does not capture tail-risk).

gradually being phased out in favour of Expected Shortfall (ES), which is another risk measure. Over the next few paragraphs, we discuss shortcomings of VaR and how ES elegantly takes care of those limitations.

We begin by observing that VaR is not a coherent risk measure. Indeed, it does not satisfy the sub-additivity property in some scenarios. In addition to this, it does not capture tail-risk, i.e. even if the distribution is heavy-tailed with a lot of large losses beyond the quantile level, the VaR does not change.

To account for these shortcomings, Expected Shortfall was introduced. Expected shortfall at α % level is the expected return on the portfolio in the worst α% of cases. Mathematically speaking, if X is the payoff of a portfolio at some future time and $0 < \alpha < 1$ then we define the expected shortfall as

$$ES_\alpha(X) = -\frac{1}{\alpha}\int_0^\alpha \mathrm{VaR}_\gamma(X)\mathrm{d}\gamma \tag{1.4}$$

Finally, we define the entropic risk measure, which uses the exponential utility func-

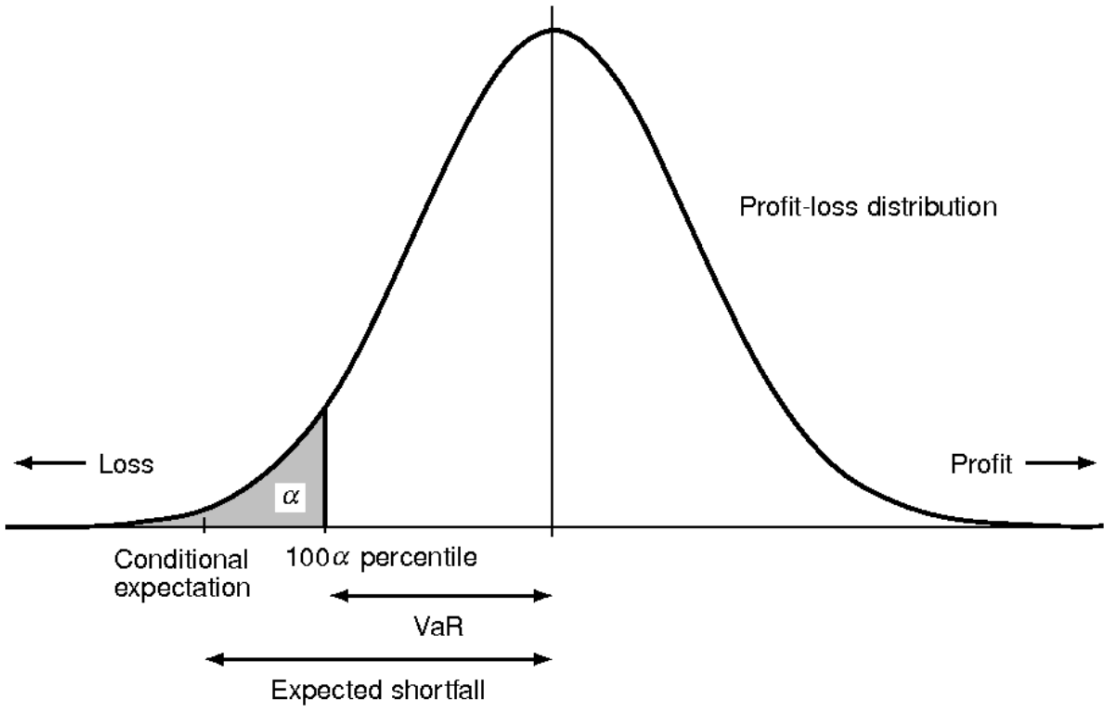

Figure 1.3: Graph highlighting Expected Shortfall.

tion in order to quantify risk. The entropic risk measure with the risk aversion parameter $\beta > 0$ is defined as

$$\rho^{ent}(X) = \frac{1}{\beta} \log \left(\mathbb{E}[e^{-\beta X}] \right). \tag{1.5}$$

Different from Expected Shortfall, the entropic risk measure is convex but not coherent.

We refer to [6] to a more comprehensive treatment of risk measures.

1.2 Calculating Risk Measures for Portfolios

There are several approaches to calculate risk measures. We highlight a couple of them in this section. In addition we discuss the data set that we will use to model returns and

subsequently estimate corresponding risk of the portfolio.

1.2.1 Methods to calculate Risk Measures

Of the few approaches that can be used to estimate risk measures, two are prominent in existing literature and practice. The first one is a direct approach, involving knowledge of the underlying joint distribution. The other one is an indirect approach, and models dependence using copulas. We discuss them one by one.

Direct approach: Modelling Joint Distribution

Generally speaking, the marginal distributions (probability density functions) of the constituents of the portfolio are known (or can be safely assumed) a priori and can be calibrated using well known univariate techniques. A simple example is the return distribution underlying the Black-Scholes option pricing model. In the Black-Scholes framework, an asset is assumed to have log returns that are normally distributed [4], thereby allowing pricing of options on the asset. In such a framework, once the return distribution is established beforehand, it is simply a question of calibrating the empirical data set for the unknown parameters, i.e. the mean and the variance. Following this, we can also check if our assumptions about the distribution were valid using, for example, tests of normality [8]. Finally, once we are convinced that the return distribution is indeed as expected, we may plug in the calibrated parameters to obtain the required answer which, in this case, is the option price.

On the other hand, the *joint* distribution of the portfolio may be sufficiently complex to preclude such an exercise. However, under certain assumptions, we can calibrate the joint distribution of the portfolio and subsequently estimate risk measures. This is the approach that we follow in Chapter 2, where we assume that the joint distribution is a Generalized Hyperbolic Distribution and is subsequently calibrated using the Expecta-

tion Maximization (EM) algorithm.

Once we know the underlying joint distribution, and have calibrated it, we may then begin calculating risk measures. For this, we simulate a return from the distribution n times, where n is a sufficiently large number (in practice, $n > 1,000,000$). These numbers are then stored in a vector, following which Value-at-Risk and Expected Shortfall are calculated using quantiles [7].

Indirect approach: Modelling Dependence via Copulas

Sometimes, it may be difficult to directly model the joint distribution of the portfolio. In this case, we rely on an implicit measure of dependence, i.e. the copula associated with that portfolio. Copulas capture the dependence structure of the joint distribution, and using this information we can evaluate risk measures. This is explained in greater detail in Chapter 3.

1.2.2 Data

The data that we use to perform analyses in this thesis consists of weekly returns of four stocks that trade on NASDAQ. These four stocks are Apple Inc. (ticker: AAPL), Ford Motor Company (ticker: F), Suncor Energy Inc. (ticker: SU) and Bank of America (ticker: BAC). The reason these particular stocks were chosen is that they represent four different sectors of the stock market, and thus their returns should, in general, be different over time. A portfolio consisting of these stocks will also be fairly representative of the entire stock market since the sectors are different. The reason weekly returns are chosen is to avoid the issue of volatility clustering that would be present in noisy daily return data.

We use the last 20 years of data (from May 1998 to May 2018), in order to have a sufficient number of weeks to conduct our analyses (1045 data points per stock). Instead

of prices, we employ log returns of closing prices to quantify movements allowing us to compare stocks at different price levels. Adjusted prices are used to account for stock splits and dividends.

We also assume an equal investment of $10,000 in each of the stocks at inception and track the movement of the portfolio with this as the reference point.

Chapter 2

Risk modelling via joint distribution

Generalized hyperbolic distributions are a large class of normal mixture distributions, containing as special and limiting cases many well-known classes of distributions, such as Student's t-distributions, variance-gamma distributions, normal-inverse Gaussian distributions, and hyperbolic distributions. More importantly, compared to normal distributions, generalized hyperbolic distributions generally have semi-heavy tails and can accommodate skewness. Thus it has been very appealing to use them to model various real-life phenomena. Their use in financial modelling started in the 1990s. See, for example, [11, 16]. As for the case of finite Gaussian mixture models, it is conventional to use the Expectation-Maximization Algorithm or its variants to estimate the parameters in generalized hyperbolic distributions. See, for example, [9, 15]. In this chapter, we fit a generalized hyperbolic distribution to return data of four stocks to model their joint distribution. After estimating the parameters using our modified MCECM algorithm, we use the Monte Carlo methods to evaluate various risk measures of a portfolio of these four stocks.

2.1 Algorithmic Setup

2.1.1 Generalized Hyperbolic Distributions

Earlier risk models were often based on multivariate normal distributions. In recent decades, more advanced distributions are demanded, among which normal mixture models have become increasingly popular. Pioneering mixture models introduce mixture to variance only, resulting in, e.g., a already very large classes of spherical and elliptical distributions. More generally, one can introduce mixtures to both mean and variance, which produces in particular generalized hyperbolic distributions.

We refer to [15] for basic properties of generalized hyperbolic distributions. A d-dimensional random vector X is said to follow a *generalized hyperbolic distribution* if

$$X \stackrel{\text{def}}{=} (\mu + W\gamma) + \sqrt{W}\Sigma^{\frac{1}{2}}Z, \tag{2.1}$$

where $\mu, \gamma \in \mathbb{R}^d$, Σ is a positive-definite $d \times d$ matrix, $Z \sim \mathrm{N}_d(0, I_d)$ follows the standard d-dimensional multivariate normal distribution, and

$$W \sim \mathrm{N}^-(\lambda, \chi, \psi)$$

follows a generalized inverse Gaussian (GIG) distribution and is independent of Z. Following the convention, we write

$$X \sim \mathrm{GH}_d(\lambda, \chi, \psi, \mu, \Sigma, \gamma).$$

To be specific, W has the following density:

$$f_W(w) = \frac{\chi^{-\frac{\lambda}{2}}\psi^{\frac{\lambda}{2}}}{2K_\lambda(\chi^{\frac{1}{2}}\psi^{\frac{1}{2}})}\ w^{\lambda-1}\exp\big(-\frac{\chi}{2}\frac{1}{w} - \frac{\psi}{2}w\big), \quad w > 0, \tag{2.2}$$

where K_λ is a modified Bessel function of the second kind with index λ. In some literature, K_λ is also called modified Bessel function of the third kind. It can be explicitly written for example as follows. For any $\lambda \in \mathbb{R}$ and $x > 0$,

$$\begin{aligned} K_\lambda(x) =& \frac{1}{2}\Big(\frac{x}{2}\Big)^\lambda \int_0^\infty \exp\Big(-t - \frac{x^2}{4t}\Big)\frac{\mathrm{d}t}{t^{\lambda+1}} \\ =& \frac{1}{2}\int_0^\infty t^{\lambda-1}\exp\Big(-\frac{x}{2}\big(t+t^{-1}\big)\Big)\,\mathrm{d}t. \end{aligned} \tag{2.3}$$

We refer to [1, 12] for basic facts on Bessel functions.

The appearance of K_λ in f_W is due to normalization: $\int_0^\infty f_W(w)\mathrm{d}w = 1$. In other words, it holds that

$$\int_0^\infty w^{\lambda-1}\exp\big(-\frac{\chi}{2}\frac{1}{w} - \frac{\psi}{2}w\big)\mathrm{d}w = 2\Big(\frac{\chi}{\psi}\Big)^{\frac{\lambda}{2}} K_\lambda(\chi^{\frac{1}{2}}\psi^{\frac{1}{2}}), \tag{2.4}$$

which follows easily from (2.3) by a change of variable. Thus, by a simple computation using (2.2) and (2.4), one easily sees that

$$\mathbb{E}[W^\alpha] = \frac{\chi^{-\frac{\lambda}{2}}\psi^{\frac{\lambda}{2}}}{2K_\lambda(\chi^{\frac{1}{2}}\psi^{\frac{1}{2}})}\int_0^\infty w^{\alpha+\lambda-1}\exp\big(-\frac{\chi}{2}\frac{1}{w} - \frac{\psi}{2}w\big)\mathrm{d}w \tag{2.5}$$

$$= \Big(\frac{\chi}{\psi}\Big)^{\frac{\alpha}{2}}\frac{K_{\lambda+\alpha}(\chi^{\frac{1}{2}}\psi^{\frac{1}{2}})}{K_\lambda(\chi^{\frac{1}{2}}\psi^{\frac{1}{2}})}. \tag{2.6}$$

By the definition of X, it is clear that $X|W$ is normally distributed with the following density:

$$\begin{aligned} & f_{X|W}(x|w) \\ =& \frac{1}{(2\pi)^{\frac{d}{2}}\det(w\Sigma)^{\frac{1}{2}}}\exp\Big(-\frac{(x-\mu-w\gamma)'(w\Sigma)^{-1}(x-\mu-w\gamma)}{2}\Big) \\ =& \frac{1}{(2\pi)^{\frac{d}{2}}\det(\Sigma)^{\frac{1}{2}}w^{\frac{d}{2}}}\exp\Big(-\frac{(x-\mu)'\Sigma^{-1}(x-\mu)}{2}\frac{1}{w} - \frac{\gamma'\Sigma^{-1}\gamma}{2}w + (x-\mu)'\Sigma^{-1}\gamma\Big). \end{aligned} \tag{2.7}$$

Therefore, by (2.4), X has the following density:

$$\begin{aligned} f_X(x) =& \int_0^\infty f_{X|W}(x|w) f_W(w)\,\mathrm{d}w \\ =& \frac{\exp((x-\mu)'\Sigma^{-1}\gamma)}{(2\pi)^{\frac{d}{2}}\det(\Sigma)^{\frac{1}{2}}} \cdot \frac{\chi^{-\frac{\lambda}{2}}\psi^{\frac{\lambda}{2}}}{2K_\lambda(\chi^{\frac{1}{2}}\psi^{\frac{1}{2}})} \\ & \times \frac{2K_{\lambda-\frac{d}{2}}\Big(\big(\chi+(x-\mu)'\Sigma^{-1}(x-\mu)\big)^{\frac{1}{2}}(\psi+\gamma'\Sigma^{-1}\gamma)^{\frac{1}{2}}\Big)}{\big(\chi+(x-\mu)'\Sigma^{-1}(x-\mu)\big)^{-\frac{\lambda-\frac{d}{2}}{2}}(\psi+\gamma'\Sigma^{-1}\gamma)^{\frac{\lambda-\frac{d}{2}}{2}}} \end{aligned} \tag{2.8}$$

Consequently, $W|X$ has the following density:

$$\begin{aligned} f_{W|X}(w|x) =& \frac{f_{X|W}(x|w)f_W(w)}{f_X(x)} \\ =& \frac{\big(\chi+(x-\mu)'\Sigma^{-1}(x-\mu)\big)^{-\frac{\lambda-\frac{d}{2}}{2}}(\psi+\gamma'\Sigma^{-1}\gamma)^{\frac{\lambda-\frac{d}{2}}{2}}}{2K_{\lambda-\frac{d}{2}}\Big(\big(\chi+(x-\mu)'\Sigma^{-1}(x-\mu)\big)^{\frac{1}{2}}(\psi+\gamma'\Sigma^{-1}\gamma)^{\frac{1}{2}}\Big)} \\ & \times w^{\lambda-\frac{d}{2}-1}\exp\Big(-\frac{\chi+(x-\mu)'\Sigma^{-1}(x-\mu)}{2}\frac{1}{w}-\frac{\psi+\gamma'\Sigma^{-1}\gamma}{2}w\Big), \end{aligned} \tag{2.9}$$

and therefore,

$$W|X=x \;\sim \mathrm{N}^-\Big(\lambda-\frac{d}{2},\chi+(x-\mu)'\Sigma^{-1}(x-\mu),\psi+\gamma'\Sigma^{-1}\gamma)\Big). \tag{2.10}$$

2.1.2 The Expectation-Maximization Algorithm

A comprehensive treatment in the EM algorithm can be found in [9, 14]. We now use it to fit a multivariate generalized hyperbolic distribution $\mathrm{GH}_d(\lambda,\chi,\psi,\mu,\Sigma,\gamma)$ to n observations

$$\mathbf{x}=(x_1,\ldots,x_n)$$

that are independently produced from an iid experiment. For this purpose, let X_i be the random outcome at the i-th observation, and write

$$\mathbf{X} = (X_1, \ldots, X_n).$$

Let

$$\mathbf{W} = (W_1, \ldots, W_n)$$

be the corresponding mixtures. Then

$$X_i \sim \mathrm{GH}_d(\lambda, \chi, \psi, \mu, \Sigma, \gamma), \quad W_i \sim \mathrm{N}^-(\lambda, \chi, \psi), \quad i = 1, \ldots, n.$$

Moreover, (X_i, W_i), $i = 1, \ldots, n$, are independent random pairs.

In the EM algorithm, we regard $(\mathbf{X}, \mathbf{W})$ as the complete information, $\mathbf{X}$ as the observable information, and $\mathbf{W}$ as the missing information. Put $\theta = (\lambda, \chi, \psi, \mu, \Sigma, \gamma)$, the parameter collection in the model. Given an estimate $\theta^{(m)}$ of θ at the m-th step, the estimate of θ at the $(m+1)$-th step is given by the argument in maximizing the following Q-function:

$$\begin{aligned}
& Q(\theta|\theta^{(m)}) \\
&= \mathbb{E}_{\mathbf{X},\mathbf{W}|\mathbf{x},\theta^{(m)}}[\ln f_{\mathbf{X},\mathbf{W}}(\mathbf{X}, \mathbf{W}|\theta)] \\
&= \sum_{i=1}^{n} \mathbb{E}_{X_i,W_i|x_i,\theta^{(m)}}\big[\ln f_{X_i,W_i}(X_i, W_i|\theta)\big] \qquad (2.11) \\
&= \sum_{i=1}^{n} \mathbb{E}_{X_i,W_i|x_i,\theta^{(m)}}\big[\ln f_{X_i|W_i}(X_i|W_i, \mu, \Sigma, \gamma)\big] + \sum_{i=1}^{n} \mathbb{E}_{X_i,W_i|x_i,\theta^{(m)}}\big[\ln f_{W_i}(W_i|\lambda, \chi, \psi)\big] \\
&= \sum_{i=1}^{n} \mathbb{E}_{W_i|x_i,\theta^{(m)}}\big[\ln f_{X_i|W_i}(x_i|W_i, \mu, \Sigma, \gamma)\big] + \sum_{i=1}^{n} \mathbb{E}_{W_i|x_i,\theta^{(m)}}\big[\ln f_{W_i}(W_i|\lambda, \chi, \psi)\big] \\
&:= Q_1(\mu, \Sigma, \gamma|\theta^{(m)}) + Q_2(\lambda, \chi, \psi|\theta^{(m)}),
\end{aligned}$$

where the second equality is due to independence of the pairs (X_i, W_i)'s and [9, Proposition 1.1].

The E-step. Here E stands for expectation. This step deals with computing Q.

Recall from (2.10) that

$$W_i|x_i, \theta^{(m)} \sim \mathrm{N}^-\Big(\lambda^{(m)}-\frac{d}{2}, \chi^{(m)}+(x_i-\mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i-\mu^{(m)}), \psi^{(m)}+(\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)}\Big).$$

Thus by (2.6), we have the following

$$\begin{aligned} a_i^{(m)} :=& \mathbb{E}_{W_i|x_i,\theta^{(m)}}\big[W_i\big] \qquad (2.12)\\ =& \Big(\frac{\chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)})}{\psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)}}\Big)^{\frac{1}{2}} \\ &\times \frac{K_{\lambda^{(m)}-\frac{d}{2}+1}(\sqrt{(\chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)}))(\psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)})})}{K_{\lambda^{(m)}-\frac{d}{2}}(\sqrt{(\chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)}))(\psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)})})} \end{aligned}$$

$$\begin{aligned} b_i^{(m)} :=& \mathbb{E}_{W_i|x_i,\theta^{(m)}}\big[W_i^{-1}\big] \qquad (2.13)\\ =& \Big(\frac{\chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)})}{\psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)}}\Big)^{-\frac{1}{2}} \\ &\times \frac{K_{\lambda^{(m)}-\frac{d}{2}-1}(\sqrt{(\chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)}))(\psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)})})}{K_{\lambda^{(m)}-\frac{d}{2}}(\sqrt{(\chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)}))(\psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)})})} \end{aligned}$$

Also, put

$$c_i^{(m)} = \mathbb{E}_{W_i|x_i,\theta^{(m)}}\big[\ln W_i\big].$$

The computation of $c_i^{(m)}$ is more sophisticated, and we will deal with it in the next

section. By (2.7), we have

$$\begin{aligned}
&Q_1(\mu, \Sigma, \gamma | \theta^{(m)}) \\
&= \sum_{i=1}^{n} \mathbb{E}_{W_i|x_i,\theta^{(m)}} \big[\ln f_{X_i|W_i}(x_i|W_i, \mu, \Sigma, \gamma) \big] \\
&= \sum_{i=1}^{n} (x_i - \mu)' \Sigma^{-1} \gamma - \frac{\gamma' \Sigma^{-1} \gamma}{2} \sum_{i=1}^{n} a_i^{(m)} - \sum_{i=1}^{n} \frac{(x_i - \mu)' \Sigma^{-1} (x_i - \mu)}{2} b_i^{(m)} - \frac{n}{2} \ln \det(\Sigma) + C,
\end{aligned}$$

where C is a constant independent of μ, Σ, γ.

By (2.2), we have

$$\begin{aligned}
&Q_2(\lambda, \chi, \psi | \theta^{(m)}) \\
&= \sum_{i=1}^{n} \mathbb{E}_{W_i|x_i,\theta^{(m)}} \big[\ln f_{W_i}(W_i|\lambda, \chi, \psi) \big] \\
&= -\frac{n}{2} \lambda \ln \chi + \frac{n}{2} \lambda \ln \psi - n \ln \Big(2K_\lambda(\chi^{\frac{1}{2}} \psi^{\frac{1}{2}}) \Big) - \frac{\psi}{2} \sum_{i=1}^{n} a_i^{(m)} - \frac{\chi}{2} \sum_{i=1}^{n} b_i^{(m)} + (\lambda - 1) \sum_{i=1}^{n} c_i^{(m)}.
\end{aligned}$$

The M-step. This step deals with finding the argument when maximizing Q. Namely,

$$\begin{aligned}
(\mu^{(m+1)}, \Sigma^{(m+1)}, \gamma^{(m+1)}) &= \arg \max_{\mu, \Sigma, \gamma} \; Q_1(\mu, \Sigma, \gamma | \theta^{(m)}) \\
(\lambda^{(m+1)}, \chi^{(m+1)}, \psi^{(m+1)}) &= \arg \max_{\lambda, \chi, \psi} \; Q_2(\lambda, \chi, \psi | \theta^{(m)}).
\end{aligned}$$

Monotonicity of Likelihood. Define the likelihood function as follows:

$$l(\theta) = \log f_{\mathbf{X}}(\mathbf{x}|\theta).$$

[9, Theorem 2.1] assets that each step in the EM algorithm increases the likelihood:

$$l(\theta) \geq l(\theta^{(m)}) \quad \text{whenever} \quad Q(\theta|\theta^{(m)}) \geq Q(\theta^{(m)}|\theta^{(m)}). \tag{2.14}$$

We refer to [10] and [14, Chapters 3 & 4] for convergence properties of the EM algorithm.

2.2 A Modified EM Algorithm

2.2.1 Modified MCECM Algorithm

Many variants of the EM algorithm have been introduced. See, e.g., [14, Chapters 5 & 6]. Since the maximization of Q is split into two components Q_1 and Q_2 which separate θ into two independent components (μ, Σ, γ) and (λ, χ, ψ), one sees that the EM algorithm coincides with the Expectation-Conditional Maximization algorithm (ECM), in which following each E-step the M-step (now called the CM-step) consists of a few consecutive maximizations subject to updated constraints on prescribed subsets of the parameter collection. If one also updates the E-step whenever a conditional maximization in the CM-step is conducted, we come to the MCECM algorithm (It should be alerted that MCECM also refers to Monte Carlo ECM in some literature). In what follows, we describe a modified version of the MCECM algorithm.

Suppose we have finished the m-th step and obtain $\theta^{(m)}$. After updating the E-step of computing $Q(\theta|\theta^{(m)})$, we have the option to first maximize Q_1 to obtain

$$(\mu^{(m+1,1)}, \Sigma^{(m+1,1)}, \gamma^{(m+1,1)})$$

, and then partially update the known parameters from $\theta^{(m)}$ to

$$\theta^{(m,0,1)} = (\lambda^{(m)}, \chi^{(m)}, \psi^{(m)}, \mu^{(m+1,1)}, \Sigma^{(m+1,1)}, \gamma^{(m+1,1)}).$$

Then we continue to update the E-step, namely, calculate $Q(\theta|\theta^{(m,0,1)})$, and then maxi-

mize Q_1 again to partially update the known parameters from $\theta^{(m,0,1)}$ to

$$\theta^{(m,0,2)} = (\lambda^{(m)}, \chi^{(m)}, \psi^{(m)}, \mu^{(m+1,2)}, \Sigma^{(m+1,2)}, \gamma^{(m+1,2)}).$$

Repeating this process, one obtains

$$\theta^{(m,0,k)} = (\lambda^{(m)}, \chi^{(m)}, \psi^{(m)}, \mu^{(m+1,k)}, \Sigma^{(m+1,k)}, \gamma^{(m+1,k)}), \quad k \in \mathbb{N}.$$

Intuitively,

$$\theta^{(m,0,\infty)} := \lim_{k\to\infty} \theta^{(m,0,k)} = (\lambda^{(m)}, \chi^{(m)}, \psi^{(m)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)})$$

gives the best fit of the model to the n observations, when $(\lambda^{(m)}, \chi^{(m)}, \psi^{(m)}$ is fixed, subject to the initial value $(\mu^{(m)}, \Sigma^{(m)}, \gamma^{(m)})$ of (μ, Σ, γ).

Next, we update the E-step, calculating $Q(\theta|\theta^{(m,0,\infty)})$, and maximize $Q_2(\lambda, \chi, \psi|\theta^{(m,0,\infty)})$ to update the parameters from $\theta^{(m,0,\infty)}$ to

$$\theta^{(m,1,\infty)} = (\lambda^{(m+1,1)}, \chi^{(m+1,1)}, \psi^{(m+1,1)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)}).$$

We continue to update the E-step, calculating $Q(\theta|\theta^{(m,1,\infty)})$, and maximize $Q_2(\lambda, \chi, \psi|\theta^{(m,1,\infty)})$ to update the parameters from $\theta^{(m,1,\infty)}$ to

$$\theta^{(m,2,\infty)} = (\lambda^{(m+1,2)}, \chi^{(m+1,2)}, \psi^{(m+1,2)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)}).$$

Repeating the process, we obtain

$$\theta^{(m,k,\infty)} = (\lambda^{(m+1,k)}, \chi^{(m+1,k)}, \psi^{(m+1,k)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)}), \quad k \in \mathbb{N},$$

and

$$\theta^{(m,\infty,\infty)} := \lim_{k\to\infty} \theta^{(m,k,\infty)} = (\lambda^{(m+1,\infty)}, \chi^{(m+1,\infty)}, \psi^{(m+1,\infty)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)}).$$

Finally, we complete the $(m+1)$-th step by setting

$$\theta^{(m+1)} := \theta^{(m,\infty,\infty)}.$$

In reality, it may be unfeasible or unnecessary to obtain $\theta^{(m,0,\infty)}$) and then $\theta^{(m,\infty,\infty)}$. Instead, one may stop first at $\theta^{(m,0,k_1)}$ and turn to work on Q_2 and stop at $\theta^{(m,k_2,k_1)}$. In practice, $k_1 = k_2 = 1$ is used. We will stick to $k_1 = k_2 = 3$. Repeatedly using (2.14), one sees that

$$l(\theta^{(m+1)}) \geq l(\theta^{(m)}).$$

It deserves mentioning that, alternatively, one can first work on Q_2 and then on the partially updated Q_1. However, we will not take this alternative.

2.2.2 A further CM split of λ, χ, ψ

We will use the MATLAB tool "fmincon" to optimize Q_2. In Section 2.1.2, one sees that Q_2 is highly nonlinear and has three variables λ, χ, ψ. These two factors add a significant level of difficulties, instability and inaccuracy into the optimization task. Indeed, if one assumes the true values of μ, Σ, γ are known (and thus never update them in the EM algorithm), then simulation studies often reveal that the estimates of λ, χ, ψ produced by the EM algorithms may be wrong, in particular, when the true values of λ, χ, ψ are somewhat extreme. We include the MATLAB codes for these simulation studies on EM estimates of W, i.e., estimates of λ, χ, ψ in Appendix A.2. One reason for such failures is that "fmincon" can only find local minimums and has reduced efficiency over high-dimensional regions.

On the other hand, if one assumes the true values of all parameters but λ (respectively, χ, ψ) are known (and thus never update them in the EM algorithm), then simulation studies show that the EM algorithm produces expected estimates of λ (respectively, χ, ψ). We include the MATLAB codes for these simulation studies on EM estimates of W, i.e., estimates of λ, χ, ψ in Appendix A.3.

Due to the above reasons, we modify the CM-step in the maximization of Q_2. Namely, we optimize $Q_2(\lambda, \theta_*^{(m,k)} | (\lambda^{(m+1,k)}, \theta_*^{(m,k)})$, where

$$\theta_*^{(m,k)} = (\chi^{(m+1,k)}, \psi^{(m+1,k)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)}),$$

over λ to update the parameters to the following:

$$(\lambda^{(m+1,k+1)}, \chi^{(m+1,k)}, \psi^{(m+1,k)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)}).$$

Then we optimize $Q_2(\chi, \theta_{**}^{(m,k)} | (\chi^{(m+1,k)}, \theta_{**}^{(m,k)})$, where

$$\theta_{**}^{(m,k)} = (\lambda^{(m+1,k+1)}, \psi^{(m+1,k)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)}),$$

over χ to update the parameters to the following:

$$(\lambda^{(m+1,k+1)}, \chi^{(m+1,k+1)}, \psi^{(m+1,k)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)})$$

Finally, we optimize $Q_2(\psi, \theta_{***}^{(m,k)} | (\psi^{(m+1,k)}, \theta_{***}^{(m,k)})$, where

$$\theta_{***}^{(m,k)} = (\lambda^{(m+1,k+1)}, \chi^{(m+1,k+1)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)}),$$

over ψ to update the parameters to the following:

$$(\lambda^{(m+1,k+1)}, \chi^{(m+1,k+1)}, \psi^{(m+1,k+1)}, \mu^{(m+1,\infty)}, \Sigma^{(m+1,\infty)}, \gamma^{(m+1,\infty)})$$

2.3 Computational considerations

2.3.1 Initial values

For $X \sim \mathrm{GH}_d(\lambda, \chi, \psi, \mu, \Sigma, \gamma)$ as defined in (2.1), we have

$$\mathbb{E}[X] = \mu + \mathbb{E}[W]\gamma, \tag{2.15}$$

$$\mathrm{V}[X] = \mathrm{V}[W]\gamma\gamma' + \mathbb{E}[W]\Sigma. \tag{2.16}$$

In practice, one usually starts with

$$\gamma^{(0)} = 0,$$

which corresponds to the special class of *symmetric* generalized hyperbolic distributions. Thus by (2.15), we set

$$\mu^{(0)} = \frac{1}{n}\sum_{i=1}^{n} x_i,$$

and by (2.16), we set

$$\Sigma^{(0)} = S,$$

where S is the sample covariance matrix. Now a second application of (2.16) inspires to restrict

$$\mathbb{E}[W] = \left(\frac{\chi}{\psi}\right)^{\frac{1}{2}} \frac{K_{\lambda+1}(\chi^{\frac{1}{2}}\psi^{\frac{1}{2}})}{K_\lambda(\chi^{\frac{1}{2}}\psi^{\frac{1}{2}})} = 1.$$

We set

$$\chi^{(0)} = \psi^{(0)} = 1,$$

and set $\lambda^{(0)}$ by solving

$$K_{\lambda^{(0)}+1}(1) = K_{\lambda^{(0)}}(1),$$

which by inspection has the solution $\lambda^{(0)} = -0.5$.

2.3.2 Computing $c_i^{(m)}$

The computation of $c_i^{(m)}$ involves evaluating $\mathbb{E}[\ln W]$, where $W \sim \mathrm{N}^-(\lambda, \chi, \psi)$. We have at least three methods to do this. The first one is to use Monte Carlo simulation, namely, simulating iid $W_i \sim \mathrm{N}^-(\lambda, \chi, \psi)$, $i = 1, \ldots, N$, and estimating

$$\mathbb{E}[\ln W] \;\approx\; \frac{1}{N}\sum_{i=1}^{N} W_i.$$

A drawback is that the convergence is quite slow, with rate $O(N^{-\frac{1}{2}})$.

Alternatively, by (2.5),

$$\mathbb{E}[\ln W] = \frac{\chi^{-\frac{\lambda}{2}}\psi^{\frac{\lambda}{2}}}{2K_\lambda(\chi^{\frac{1}{2}}\psi^{\frac{1}{2}})}\int_0^\infty (\ln w) w^{\lambda-1}\exp\big(-\frac{\chi}{2}\frac{1}{w} - \frac{\psi}{2}w\big)\mathrm{d}w \tag{2.17}$$

$$= \frac{\partial \mathbb{E}[W^\alpha]}{\partial \alpha}(0). \tag{2.18}$$

Thus one may compute $\mathbb{E}[\ln W]$ via differentiation in (2.18) or via integration via improper integration in (2.17). To avoid numerical instability of differentiation, we will use integration, which can be easily done by implementing the integral command in MATLAB.

Therefore, since

$$W_i|x_i, \theta^{(m)} \;\sim \mathrm{N}^-\Big(\lambda^{(m)} - \frac{d}{2}, \chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)}), \psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)}\Big),$$

we have

$$
\begin{aligned}
c_i^{(m)} =& \Big(\frac{\psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)}}{\chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)})}\Big)^{\frac{\lambda^{(m)} - \frac{d}{2}}{2}} \\
&\times \frac{1}{2K_{\lambda^{(m)} - \frac{d}{2}}(\sqrt{(\chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)}))(\psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)})})} \\
&\times \int_0^\infty (\ln w) w^{\lambda^{(m)} - \frac{d}{2} - 1} \exp\Big(- \frac{\chi^{(m)} + (x_i - \mu^{(m)})'(\Sigma^{(m)})^{-1}(x_i - \mu^{(m)})}{2}\frac{1}{w} \\
&- \frac{\psi^{(m)} + (\gamma^{(m)})'(\Sigma^{(m)})^{-1}\gamma^{(m)}}{2} w\Big) \mathrm{d}w
\end{aligned}
$$

2.3.3 Maximizing Q_1

We refer to [19] for basics on vector and matrix differentiation. Differentiating Q_1 we have

$$
\begin{aligned}
\frac{\partial Q_1}{\partial \mu} =& - n\Sigma^{-1}\gamma + \sum_{i=1}^n \Sigma^{-1}(x_i - \mu) b_i^{(m)}, \\
\frac{\partial Q_1}{\partial \gamma} =& \sum_{i=1}^n \Sigma^{-1}(x_i - \mu) - \Sigma^{-1}\gamma \sum_{i=1}^n a_i^{(m)}, \\
\frac{\partial Q_1}{\partial \Sigma^{-1}} =& \sum_{i=1}^n (x_i - \mu)\gamma' - \frac{1}{2}\gamma\gamma' \sum_{i=1}^n a_i^{(m)} - \frac{1}{2}\sum_{i=1}^n (x_i - \mu)(x_i - \mu)' b_i^{(m)} + \frac{n}{2}\Sigma.
\end{aligned}
$$

Setting the partial derivatives to 0 and solving the equations, we obtain

$$
\begin{aligned}
\gamma^{(m+1)} =& \frac{\frac{1}{n}\sum_{i=1}^n b_i^{(m)}(x_i - \overline{x})}{1 - \frac{1}{n}\sum_{i=1}^n b_i^{(m)} \cdot \frac{1}{n}\sum_{i=1}^n a_i^{(m)}}, \\
\mu^{(m+1)} =& \frac{\sum_{i=1}^n x_i}{n} - \frac{\sum_{i=1}^n a_i}{n}\gamma, \\
\Sigma =& \frac{1}{n}\sum_{i=1}^n (x_i - \mu^{(m+1)})(x_i - \mu^{(m+1)})' b_i^{(m)} - \frac{1}{n}\sum_{i=1}^n a_i^{(m)} \cdot \gamma^{(m+1)}(\gamma^{(m+1)})'
\end{aligned}
$$

2.3.4 Maximizing Q_2

In practice, one shall maximize $2Q_2/n$ to remove the magnitude n in case the data size is very large and unnecessary constants. Namely, we need to find the argument in minimizng the following function:

$$\begin{aligned} & Q_2'(\lambda, \chi, \psi | \theta^{(m)}) \\ &= \lambda \ln \chi - \lambda \ln \psi + 2 \ln \left(K_\lambda(\chi^{\frac{1}{2}} \psi^{\frac{1}{2}}) \right) + \psi \frac{\sum_{i=1}^n a_i^{(m)}}{n} + \chi \frac{\sum_{i=1}^n b_i^{(m)}}{n} - 2\lambda \frac{\sum_{i=1}^n c_i^{(m)}}{n}. \end{aligned}$$

We will use the MATLAB tool "`fmincon`" to solve this nonlinear optimization problem with constraints: $\chi \geq 0$ and $\psi \geq 0$. Note that the precise constraints of the parameters λ, χ, ψ are as follows: $\chi \geq 0$, $\psi > 0$ if $\lambda > 0$; $\chi > 0$, $\psi > 0$ if $\lambda = 0$; $\chi > 0, \psi \geq 0$ if $\lambda < 0$.

2.4 Numerical Results

In this section, we discuss the results from calibrating and simulating Generalized Hyberbolic Distributions with our sample data.

2.4.1 EM Estimates of Parameters

Running the code in Appendix A.1 enables us to calculate the best estimates of the parameters involved in the Generalized Hyperbolic Distribution. We list them here.

$\lambda = -1.8552$

$\chi = 2.2326$

$\psi = 5.1476 * 10^{-5}$

$$\gamma' = \begin{bmatrix} 3.5261 * 10^{-3} \\ 2.4177 * 10^{-3} \\ 1.0796 * 10^{-3} \\ 2.6350 * 10^{-3} \end{bmatrix}$$

$$\mu' = \begin{bmatrix} -1.0028 * 10^{-2} \\ -3.739 * 10^{-3} \\ -1.2105 * 10^{-3} \\ -6.1779 * 10^{-3} \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 2.9068 * 10^{-3} & 6.0378 * 10^{-4} & 6.8643 * 10^{-4} & 5.7658 * 10^{-4} \\ 6.0378 * 10^{-4} & 2.1498 * 10^{-3} & 9.8082 * 10^{-4} & 6.0630 * 10^{-4} \\ 6.8643 * 10^{-4} & 9.8082 * 10^{-4} & 2.4551 * 10^{-3} & 6.4051 * 10^{-4} \\ 5.7658 * 10^{-4} & 6.0630 * 10^{-4} & 6.4051 * 10^{-4} & 2.1930 * 10^{-3} \end{bmatrix}$$

2.4.2 Simulation Results

The parameters are then used in conjunction with Monte Carlo simulations to generate 1,000,000 simulations of the stock return values. Following this, we take the rowsums to evaluate the loss for the portfolio. For this, we assume that we have invested $10,000 in each of the four stocks. The results are listed in Table 2.1

Expected Shortfall	**Value at Risk**
$4,793	$4,352

Table 2.1: ES and VaR based on Generalized Hyperbolic Distribution

We will come back to these numbers in Chapter 4, where we will discuss results along with conclusions and inferences.

Chapter 3

Risk Modelling via Marginal Distributions

If we have a joint distribution function corresponding to a risk factor random vector, we have a dependence structure that is implicit in it. Copulas help us in the isolation of this dependence structure.

Dependence and correlation may seem synonymous at first glance, however, a deeper look reveals that linear correlations are inadequate to measure dependence. To this end, we can discuss alternative dependence measures including coefficients of tail dependence using copulas. The latter is of particular interest in the study of financial market crashes, since there may be a high degree of dependence in the tails which represent extreme events.

3.1 Basic Copula Theory

In this section we review the definition of a copula, some basic properties that a copula must satisfy and some examples of copulas. We refer to [18] for a comprehensive account of copulas. Throughout this chapter, all distributions are continuous, unless specified

otherwise.

3.1.1 Definitions

Definition 1. *A d-dimensional* copula *is a function* $C : [0,1]^d \to [0,1]$ *satisfying the following conditions:*

(i) $C(u_1, \ldots, u_d)$ *is increasing in each component* u_i*;*

(ii) $C(1, \ldots, 1, u_i, 1, \ldots, 1) = u_i$ *for all* $i \in \{1, \ \ldots \ , \ d\}$ *and* $u_i \in [0,1]$*;*

(iii) For any $(a_1, \ldots, a_d), (b_1, \ldots, b_d) \in [0,1]^d$ *with* $a_i \leq b_i$ *for all* $1 \leq i \leq d$*, it holds that*

$$\sum_{i_1=1}^{2} \cdots \sum_{i_d=1}^{2} (-1)^{i_1 + \ \cdots + i_d} C(u_{1i_1}, \ldots, u_{di_d}) \geq 0,$$

where $u_{j1} = a_j$ *and* $u_{j2} = b_j$ *for all* $j \in \{1, \ldots, \ d\}$.

As is well-known from distribution theory, copulas can be equivalently defined as follows.

Definition 2. *A function* $C : [0,1]^d \to [0,1]$ *is a* copula *if and only if there exists a random vector* $U = (U_1, \ldots, U_d)$ *such that* U_i *is a standard uniform distribution on* $[0,1]$ *for each* $1 \leq i \leq d$ *and that*

$$C(u_1, \ldots, u_d) = \mathbb{P}(U_1 \leq u_1, \ldots, U_d \leq u_d), \ \forall (u_1, \ldots, u_d) \in [0,1]^d.$$

We call this random vector U, although not unique, a *generating random vector* of C. It often plays a useful role in simulations.

3.1.2 Basic Properties

Next we proceed to discuss some properties of copulas. Before we do that, we observe the following properties for distribution functions.

Let F be a continuous distribution function and let F^- denote the left continuous quantile of F, i.e.,

$$F^-(y) = \inf\{x : F(x) \geq y\}.$$

Then the following hold:

(a) If U is a random variable following standard uniform distribution, then the distribution function of $F^-(U)$ is F;

(b) If X is a random variable with distribution function F, then $F(X)$ follows the standard uniform distribution.

We refer to [6] for detailed discussions of these properties.

The primary importance of copulas in isolating dependence structure from marginal distributions is illustrated in the celebrated theorem of Sklar.

Theorem 3 (Sklar)**.** *Let F be a joint distribution function with marginals $F_1, \ldots, F_d$. Then there exists a unique copula $C : [0,1]^d \to [0,1]$ such that, for all $x_1, \ldots, x_d \in \mathbb{R}$,*

$$F(x_1, \ldots, x_d) = C(F_1(x_1), \ \ldots \ , \ F_d(x_d)). \tag{3.1}$$

Clearly, by (3.1), for any $u_1, \ldots, u_d \in [0,1]$,

$$C(u_1, \ldots, u_d) = F(F_1^-(u_1), \ldots, F_d^-(u_d)). \tag{3.2}$$

Definition 4. *We call C in Theorem 3 the* copula of F*. If F is the distribution of a random vector $(X_1, \ldots, X_d)$, we also call C the copula of $(X_1, \ldots, X_d)$, or $(X_1, \ldots, X_d)$ an* associated random vector of C*.*

We now make two useful remarks regarding generating and associated random vectors, combining which gives the standard procedures to simulate random vectors with a given copula and given marginals.

Remark 5. *Let C be a copula with associated random vector $(X_1, \ldots, X_d)$. Let F_i be the distribution of X_i. Note that each $F_i(X_i)$ is a standard uniform distribution and that*

$$\begin{aligned}
&\mathbb{P}\big(F_1(X_1) \leq u_1, \ldots, F_d(X_d) \leq u_d\big) \\
=&\mathbb{P}\big(F_1^- \circ F_1(X_1) \leq F_1^-(u_1), \ldots, F_d^- \circ F_d(X_d) \leq F_d^-(u_d)\big) \\
=&\mathbb{P}(X_1 \leq F_1^-(u_1), \ldots, X_d \leq F_d^-(u_d)) \\
=&F(F_1^-(u_1), \ldots, F_d^-(u_d)) \\
=&C(u_1, \ldots, u_d).
\end{aligned}$$

Therefore, $(F_1(X_1), \ldots, F_d(X_d))$ is a generating random vector of C

Remark 6. *Let C be a copula with a generating random vector $(U_1, \ldots, U_d)$. Let $F_1, \ldots, F_d$ be given marginals. Note that each $F_i^-(U_i)$ has distribution function F_i and*

$$\begin{aligned}
&\mathbb{P}\big(F_1^-(U_1) \leq x_1, \ldots, F_d^-(U_d) \leq x_d\big) \\
=&\mathbb{P}\big(U_1 \leq F_1(x_1), \ldots, U_d \leq F_d(x_d)\big) \\
=&C(F_1(x_1), \ldots, F_d(x_d)).
\end{aligned}$$

Therefore, by (3.1)*, $(F_1^-(U_1), \ldots, F_d^-(U_d))$ is an associated random vector of C with prescribed marginals F_i's.*

We end this subsection with a useful result that allows us to simplify the associated random vector when determining its copula. This result demonstrates again the fact that copulas capture the essential dependence structure in a joint distribution.

Proposition 7. *Let $(X_1, \ldots, X_d)$ be a random vector with copula C and let $T_1, \ldots, T_d$ be strictly increasing functions on $\mathbb{R}$. Then $(T_1(X_1), \ldots, T_d(X_d))$ also has copula C.*

3.1.3 Examples of Copulas

We provide a number of examples of copulas in this section and these are subdivided into three categories: *fundamental* copulas represent a number of important special dependence structures; *implicit* copulas are extracted from well-known multivariate distributions using Sklar's Theorem, but do not necessarily possess simple closed- form expressions; *explicit* copulas have simple closed-form expressions and follow general mathematical constructions known to yield copulas.

Independence Copula

As a first example of fundamental copulas, the *independence copula* is defined by

$$\Pi(u, ., \ u) = \prod_{i=1}^{d} u_i. \tag{3.3}$$

By Sklar's Theorem, it is clear that a random vector has the independence copula if and only if its marginals are independent.

Comonotonicity Copula

The *comonotonicity copula* is another basic example of fundamental copulas. It is defined by

$$M(u_1, \ldots, u_d) = \min\{u_1, \ldots, u_d\}. \tag{3.4}$$

Observe that this copula has a generating random vector $(U, \ldots, U)$, where U follows a standard uniform distribution on $[0, 1]$.

Let X be a random variable and T_i, $i = 1, \ldots, d$, be strictly increasing functions

on $\mathbb{R}$. Then by Proposition 7, the copula of $(T_1(X), \ldots, T_d(X))$ is the same as that of $(X, \ldots, X)$, which in turn is the same as that of $(U, \ldots, U)$, where $U = F(X)$. Hence, the copula is in fact just the comonotonicity copula.

Gauss Copula

Gauss copula is one of the most important implicit copulas. By Proposition 7, $X \sim N_d(\mu, \Sigma)$ and $Y \sim N_d(0, P)$ have the same copula, where P is the corresponding correlation matrix of Σ. Their copula is called a Gauss copula. Specifically, by Remark 5, we have

$$\begin{aligned}
&C_P^{\mathrm{Ga}}(u_1, \ldots, u_d) \\
&= \mathbb{P}(\Phi(Y_1) \leq u_1, \ldots, \Phi(X_d) \leq u_d) \\
&= \Phi_P(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_d)) \\
&= \int_{-\infty}^{\Phi^{-1}(u_1)} \cdots \int_{-\infty}^{\Phi^{-1}(u_d)} \det(2\pi P)^{-\frac{1}{2}} \exp\Big\{ -\frac{1}{2} y' P^{-1} y \Big\} \mathrm{d}y.
\end{aligned}$$

Gauss copulas were most popular in risk modelling. But some academics and practitioners attribute the 2007-2008 crisis to its wide use in credit risk modelling and its weakness to capture dependence, and since then its use has been greatly reduced.

t Copula

In the same way that we can extract a copula from the multivariate normal distribution, we can extract an implicit copula from any other distribution with continuous marginal

dfs. For example, the d-dimensional t *copula* takes the form

$$\begin{aligned} & C_{v,P}^{t}(u_1,\ldots,u_d) \\ &= t_{v,P}(t_\nu^{-1}(u_1),\ \ldots\ ,\ t_\nu^{-1}(u_d)) \\ &= \int_{-\infty}^{\Phi^{-1}(u_1)} \cdots \int_{-\infty}^{\Phi^{-1}(u_d)} \frac{\Gamma\left[(\nu+d)/2\right]}{\Gamma(\nu/2)\nu^{d/2}\pi^{d/2}\left|\Sigma\right|^{1/2}} \left[1+\frac{1}{\nu}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right]^{-(\nu+d)/2} \mathrm{d}x. \end{aligned}$$

t copulas capture both upper and lower tail dependence. Their capture of dependence are stronger than Gauss copula but weaker than the following two copulas each of which only capture one-sided tail dependence for the upper and lower tail dependence.

Gumbel Copula

The *Gumbel copula* is a typical explicit copula, which captures upper tail-dependence.

$$C_\theta^{\mathrm{Gu}}(u_1,\ldots,u_d) = \exp\left\{-\Big(\sum_{i=1}^{d}(-\ln u_i)^\theta\Big)^{\frac{1}{\theta}}\right\}, \quad 1 \le \theta < \infty. \tag{3.5}$$

Clayton Copula

The last example is the *Clayton copula*, which is also an explicit copula but captures lower tail-dependence.

$$C_\theta^{\mathrm{Cl}}(u_1,\ldots,u_d) = (u_1^{-\theta}+\cdots+u_d^{-\theta}-d+1)^{-1/\theta},\ 0<\theta<\infty. \tag{3.6}$$

3.2 Marginal Distributions of stocks

We will work with 20 years of data for 4 stocks, which are Apple (AAPL), Bank of America (BAC), Ford (F) and Suncor Energy (SU). The reason for choosing these particular stocks was that they are among the largest in their respective sectors and hail from

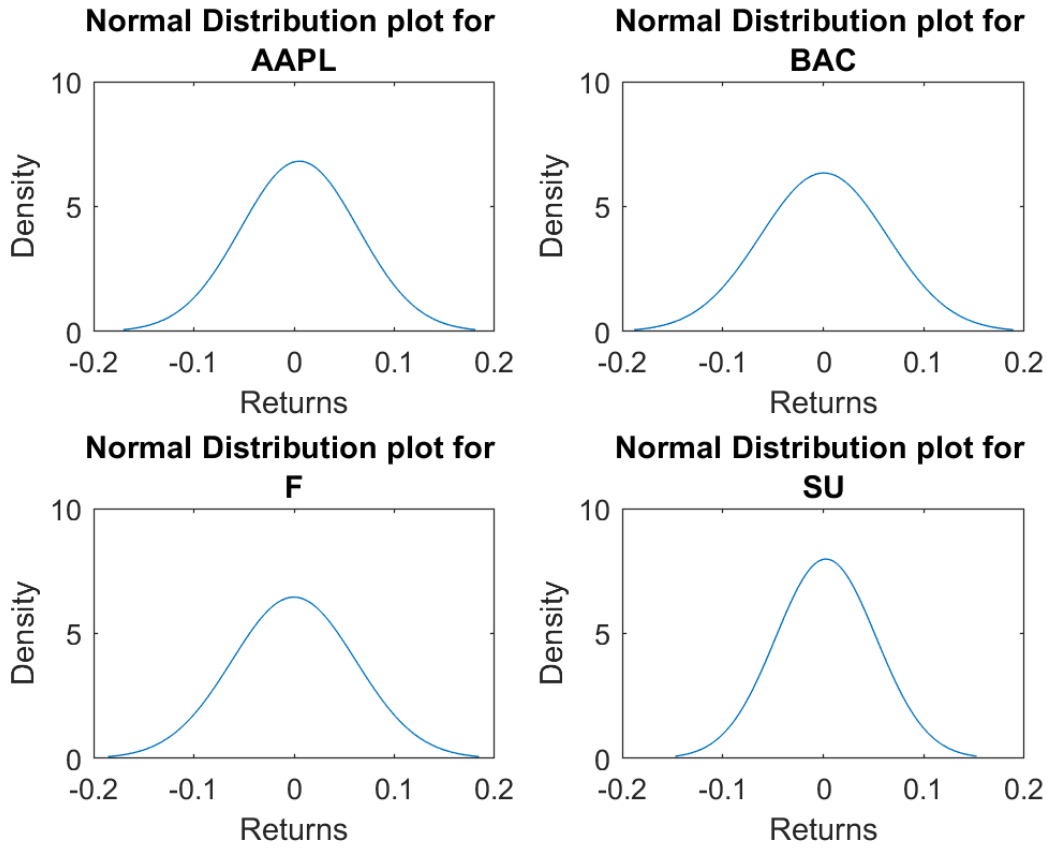

Figure 3.1: Marginal distributions of individual stocks.

different sectors, thereby allowing for some diversification. We assume that the marginal distributions of the stocks are normal. This assumption is chosen for simplicity, but may be relaxed to include any other distribution with no change in the procedure. Under this assumption, we calibrate the data set and obtain the distributions for the four stocks in Figure 3.1. The corresponding values are listed in Table 3.1.

Next, we assume that we have invested \$10,000 in each of the four stocks. In accordance with Basel regulations, we use the 99th percentile for VaR and 97.5th percentile

Stock	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}$
AAPL	-0.0053	0.0585
BAC	-0.0005	0.0629
F	0.0002	0.0618
SU	-0.0027	0.0500

Table 3.1: Mean and standard deviation of log returns of marginal distributions of stocks

onwards for ES. The ES and VaR values based on 1,000,000 simulations are tabulated in Table 3.2.

Stock	**Expected Shortfall**	**Value at Risk**
AAPL	\$1,312	\$1,311
BAC	\$1,415	\$1,310
F	\$1,378	\$1,323
SU	\$1,076	\$1,056
Sum	**\$5,178**	**\$4,999**

Table 3.2: ES and VaR based on marginal distributions of stocks

3.3 Empirical Results

For each of the four copulas defined in Section 2.1.3, and for the empirical marginal distributions determined in Section 2.2, we evaluate the results of plugging in the marginal distributions into the copulas.

3.3.1 Gauss Copula

We start with the Gauss Copula. For this, we need a correlation matrix. As suggested in [15], the following is a close approximation to the calibrated correlation matrix:

$$\rho_S(X_i,\ X_j) = (6/\pi)\arcsin\frac{1}{2}\rho_{ij} \approx \rho_{ij},$$

where ρ_S is the matrix formed by taking the pairwise Spearman's rank correlation coefficients between the stock return vectors. The matrix ρ then takes the following form based on the empirical data:

$$\rho = \begin{pmatrix} 1.0000 & 0.2713 & 0.2843 & 0.2329 \\ 0.2713 & 1.0000 & 0.4137 & 0.2439 \\ 0.2843 & 0.4137 & 1.0000 & 0.2650 \\ 0.2329 & 0.2439 & 0.2650 & 1.0000 \end{pmatrix}$$

We then simulate the copula by the following steps:

Algorithm 2.1 (simulation of Gauss copula).

(1) Generate $Z \sim N_d(0,\ P)$;

(2) Return $U = (\Phi(Z_1), \ldots, \Phi(Z_d))'$, where Φ is the standard normal df. The random vector U has df C_P^{Ga};

(3) Apply Remark 6 with the marginal distributions of the stocks as calculated in section 3.2 to obtain a random vector with C_P^{Ga}.

Once we have the copula, and plugged in the marginals, we take the rowsums to evaluate the loss for the portfolio. For this, we assume that we have invested \$10,000 in each of the four stocks. The results are listed in Table 3.3

Expected Shortfall	**Value at Risk**
\$3,388	\$3,303

Table 3.3: ES and VaR based on Gauss Copula

As expected, both VaR and ES decrease when the stocks are put in a portfolio. The number for Value at Risk is \$3,303 and for Expected Shortfall is \$3,388, down from \$4,999 and \$5,178 respectively.

3.3.2 t Copula

The next copula that we discuss is the t Copula. For this, we need a correlation matrix and degrees of freedom. We may choose any arbitrary correlation matrix, however, as suggested in [15] the following relation holds:

$$\rho_\tau(X_i,\ X_j) = (2/\pi)\sin^{-1}\rho_{ij},$$

where ρ_τ is the matrix formed by taking the pairwise Kendall's rank relation coefficients between the stock return vectors.
It then follows, that our matrix may be estimated by multiplying both sides by $\pi/2$ and subsequently taking the sine of both sides leading to the following for the purposes of this simulation:

$$\rho = \begin{pmatrix} 1.0000 & 0.2866 & 0.3007 & 0.2477 \\ 0.2866 & 1.0000 & 0.4415 & 0.2612 \\ 0.3007 & 0.4415 & 1.0000 & 0.2858 \\ 0.2477 & 0.2612 & 0.2858 & 1.0000 \end{pmatrix}$$

We also choose ν to be 5. The reasoning behind this is that we want to differentiate this from the Gauss Copula (since t Copula converges asymptotically to the Gauss Copula as degrees of freedom increase) and in order to observe the strong tail dependence present in the t Copula by design.

We then simulate the copula by the following steps:

Algorithm 2.2 (simulation of t copula).

(1) Generate $X \sim t_d(\nu,\ 0,\ P)$.

(2) Return $U = (t_\nu(X_1),\ \ldots\ ,\ t_\nu(X_d))'$, where t_ν denotes the df of a standard univariate t distribution. The random vector U has df $C^t_{v,P}$.

(3) Apply Remark 6 with the marginal distributions of the stocks as calculated in section 3.2 to obtain a random vector with $C^t_{v,P}$.

Once we have the copula, and plugged in the marginals, we take the rowsums to evaluate the loss for the portfolio. For this, we assume that we have invested \$10,000 in each of the four stocks. The results are listed in Table 3.4.

Expected Shortfall	**Value at Risk**
\$3,700	\$3,660

Table 3.4: ES and VaR based on t Copula

As expected, both VaR and ES decrease when the stocks are put in a portfolio even for the t Copula. The number for Value at Risk is \$3,660 and for Expected Shortfall is \$3,700, down from \$4,999 and \$5,178 respectively. Also note that while the Value at Risk increasing by using a t Copula as opposed to a Gauss Copula, the Expected Shortfall actually goes down. This is due to the fact that for the same correlation matrix ρ, the t Distribution has heavier tails than the Normal Distribution.

3.3.3 Gumbel Copula

Another copula that we discuss in greater detail is the Gumbel copula. The interesting feature about this copula is that it captures upper tail-dependence. A Gumbel copula is uniquely characterized by the parameter $\theta \geq 1$.

Calibrating θ in the two-dimensional case boils down to a one-to-one mapping between the Kendall's rank correlation coefficient and θ, however, it is not trivial to calibrate in the d-dimensional case. For the purpose of our simulations, we choose $\theta = 3$, and then run the following algorithm to simulate the Gumbel Copula:

Algorithm 2.3 (simulation of Gumbel copula).

(1) First, generate a variate V with df G such that $\hat{G}$, the Laplace-Stieltjes transform of G, will be the inverse of the generator ϕ of the required copula.

(2) Next, generate independent uniform variates $X_1, \ldots, X_d$.

(3) Return $U = (\hat{G}(-\ln(X_1)/V), \ldots, \hat{G}(-\ln(X_d)/V$ where V is a positive stable variate $V \sim \mathrm{St}(1/\theta,\ 1,\ \gamma,\ 0)$, where $\gamma = (\cos(\pi/(2\theta)))^\theta$ and $\theta > 1$. The resultant df has Laplace transform $\hat{G}(t) = \exp(-t^{1/\theta})$ which is what we wanted.

(4) Apply Remark 6 with the marginal distributions of the stocks as calculated in section 3.2 to obtain a random vector with C_θ^{Gu}.

Once we have the copula, and plugged in the marginals, we take the rowsums to evaluate the loss for the portfolio. For this, we assume that we have invested \$10,000 in each of the four stocks. The results are listed in Table 3.5.

Expected Shortfall	**Value at Risk**
\$5,613	\$5,544

Table 3.5: ES and VaR based on Gumbel Copula

Due to high dependence in the upper tails, we have that the Value at Risk and Expected Shortfall increase from \$4,999 and \$5,178 to \$5,544 and \$5,613 respectively. This can also partly be explained by the instability of the random number generator for Positive Stable Variates in MATLAB.
We will come back to these numbers in Chapter 4, where we will discuss results along with conclusions and inferences.

Chapter 4

Conclusions and Inferences

This chapter is dedicated to summarizing and dissecting the results obtained in Chapters 2 and 3, and attempting to explain them using both Mathematics and Finance. We will begin with numerical comparisons, and then proceed to inferences. Finally, we will finish by discussing potential for future research.

4.1 Numerical Comparisons

First, we summarize the results obtained by the various copulas in Table 4.1.

Copula Name	**Expected Shortfall**	**Value at Risk**
Gauss Copula	\$3,388	\$3,303
t Copula	\$3,700	\$3,660
Gumbel Copula	\$5,613	\$5,544

Table 4.1: ES and VaR based on various Copulas

It is evident from the table that both Gauss Copula and t Copula are able to capture the concept of risk reduction due to diversification [13], whereas the Gumbel Copula amplifies upper tail risk.

Next, we compare this with the numbers obtained by using the Generalized Hyperbolic Distribution. Recall that the values for the Value at Risk and Expected Shortfall were \$4,352 and \$4,793 respectively. While these are higher than those obtained by using the Gauss and t Copulas, they are significantly lower than the values of the Gumbel Copula. It is worth stressing again that this is the most generic version of the Generalized Hyperbolic Distribution, with all parameters being free.
Another comparison can be drawn between the values obtained by using the sum of the individual Value at Risk and Expected Shortfall for the various stocks versus the values obtained by putting them in a portfolio first and subsequently applying either Copulas or Generalized Hyperbolic Distributions to the complete portfolio. Recall that the sum of the individual values was \$5,178 and \$4,999 for the Expected Shortfall and Value at Risk respectively. While the Generalized Hyperbolic Distribution, and the Gauss and t Copulas result in a reduction of risk, the Gumbel Copula results in an *increase* in risk.

4.2 Inferences

The class of Generalized Hyperbolic Distributions encompasses a wide range of distributions due to the fact that there are six parameters in total. This inherently allows for tweaking and fitting according to real life needs. This customization is useful when a simple explanatory model (for example, the Multivariate Normal Model), may be insufficient to capture the intricacies of the data set.

Copulas are an elegant way of measuring and allowing for dependence in portfolios that may otherwise be difficult to capture. Different copulas have different properties, and we have to find one that suits our needs. Often, the most extreme case (in this case, the Gumbel Copula) can be taken from the point of view of a conservative risk-adverse investor. For other practical purposes (for example, internal and external reporting),

other copulas like the Gauss and the t Copulas may be better fits.

4.3 Future Research

The following are questions that can be considered extensions to the work done in this thesis and may be suitable for researching further:

1. Is it worth defining the Generalized Hyperbolic Distribution in its most generic form (using all six parameters), or is there merit to simplifying the problem by fixing one or more parameters, thereby specifying a subset of Generalized Hyperbolic Distributions (several of which are named distributions)?

2. How do these two methods, namely using Copulas or Generalized Hyperbolic Distributions, stack against other known methods of estimation in literature, for example, the Rearrangement Algorithm proposed in [17]?

3. How well do these methods capture risk in real life? This is a question that is quite open-ended and subjective, but having a metric that quantifies and compares the performance of different methods in the existing financial markets can be useful.

4. What other risk measures (for example, the Entropic Risk measure [5]) can be suitable for quantifying financial risk? What properties do these risk measures satisfy when applied to individual stocks? More importantly, what properties still hold (or come into existence) while discussing *portfolio* risk?

Appendix A

Codes for Chapter 2

A.1 EM algorithm: Calibration and Simulation

```
% clearvars -except returnmatrix

X = returnmatrix;
[n,d] = size(returnmatrix); % calculate dimensions of return
    matrix

nsimulations = 1000000; % number of simulations to run
confidence_level_VaR = 0.99; % confidence level for VaR
confidence_level_ES = 0.975; % confidence level for ES
dollarvalue = 10000; % amount invested in each stock

% initial values

gamma = zeros(1,d);
mu = mean(X);
```

```
15 Sigma = cov(X);
16 psi = 1;
17 chi = 1;
18 lambda = -0.5;
19 iterations = 100;
20 klambda = 5;
21 kchi = 5;
22 kpsi = 5;
23 kW = 5;
24 for iter = 1:iterations
25     % lambda optimization
26     chiwx = chi+sum(((-mu+X)/Sigma).*(-mu+X),2); % this is nx1
           matrix
27     psiwx = psi+gamma/Sigma*transpose(gamma);
28
29     for j=1:klambda
30
31         %compute the density of W_i|x_i
32         lambdawx = lambda-d/2;
33
34         %compute c
35         c = zeros(n,1);% this is nx1 matrix: E[ln W]
36         for i=1:1:n
37             fun1 = @(w) log(w).*w.^(lambdawx-1).*exp(-0.5*
                   psiwx*w).*exp(-0.5*chiwx(i,1)*w.^(-1));
38             c(i,1) = 0.5*(psiwx/chiwx(i,1))^(0.5*lambdawx)/
                   besselk(lambdawx,sqrt(chiwx(i,1)*psiwx))*
```

```
                integral(fun1,0,Inf);
39              clear fun1;
40          end
41
42          %update lambda, chi, psi
43          options = optimoptions('fmincon','Algorithm','sqp','
                Display','off');
44          fun2 = @(t) t*log(chi)-t*log(psi)+2*log(besselk(t,
                sqrt(chi*psi)))-2*t*mean(c);
45          A = [ ];
46          B = [ ];
47          Aeq = [];
48          beq = [];
49          lb = [];
50          ub = [];
51          nonlcon = [];
52          t0 = lambda;
53          t = fmincon(fun2,t0,A,B,Aeq,beq,lb,ub,nonlcon,
                options);
54          clear fun2;
55
56          %print out the result
57          step = j;
58          lambda = t;
59      end
60      % chi optimization
61      lambdawx = lambda-d/2;
```

```
62      psiwx = psi+gamma/Sigma*transpose(gamma);
63      for j=1:kchi %work on Q2 and partially update lambda,chi,psi
            for k times
64
65          %compute the density of W_i|x_i
66          chiwx = chi+sum(((-mu+X)/Sigma).*(-mu+X),2); % this is
                nx1 matrix
67
68          %compute a,b,c
69          b = sqrt(psiwx./chiwx).*besselk(lambdawx-1,sqrt(chiwx*
                psiwx))./besselk(lambdawx,sqrt(chiwx*psiwx));% this
                is nx1 matrix: E[W^(-1)]
70
71
72          %update lambda, chi, psi
73          options = optimoptions('fmincon','Algorithm','sqp','
                Display','off');
74          fun2 = @(t) lambda*log(t)+2*log(besselk(lambda,sqrt(t*
                psi)))+t*mean(b);
75          A = -1;
76          B = 0;
77          Aeq = [];
78          beq = [];
79          lb = [];
80          ub = [];
81          nonlcon = [];
82          t0 = chi;
```

```
83          t = fmincon(fun2,t0,A,B,Aeq,beq,lb,ub,nonlcon,options);
84
85          %print out the result
86          step = j;
87          chi = t;
88
89      end
90      % psi optimization
91      lambdawx = lambda-d/2;
92      chiwx = chi+sum(((-mu+X)/Sigma).*(-mu+X),2); % this is nx1
            matrix
93      for j=1:kpsi
94
95          %compute the density of W_i|x_i
96
97          psiwx = psi+gamma/Sigma*transpose(gamma);
98
99          %compute a,b,c
100         a = sqrt(chiwx/psiwx).*besselk(lambdawx+1,sqrt(chiwx*
                psiwx))./besselk(lambdawx,sqrt(chiwx*psiwx));% this
                is nx1 matrix: E[W]
101
102         %update lambda, chi, psi
103         options = optimoptions('fmincon','Algorithm','sqp','
                Display','off');
104         fun2 = @(t) -lambda*log(t)+2*log(besselk(lambda,sqrt(chi
                *t)))+t*mean(a);
```

```
105         A = -1;
106         B = 0;
107         Aeq = [];
108         beq = [];
109         lb = [];
110         ub = [];
111         nonlcon = [];
112         t0 = psi;
113         t = fmincon(fun2,t0,A,B,Aeq,beq,lb,ub,nonlcon,options);
114
115         %print out the result
116         step = j;
117         psi = t;
118     end
119     % mu, sigma, gamma optimization
120     for i = 1:kW
121         lambdawx = lambda-d/2;
122         chiwx = chi+sum(((-mu+X)/Sigma).*(-mu+X),2); % this is
              nx1 matrix
123         psiwx = psi+gamma/Sigma*transpose(gamma);
124         j;
125         %compute the density of W_i|x_i
126         chiwx = chi+sum(((-mu+X)/Sigma).*(-mu+X),2); % this is
              nx1 matrix
127         psiwx = psi+gamma/Sigma*transpose(gamma);
128
129         %compute a,b
```

```
130         a = sqrt(chiwx/psiwx).*besselk(lambdawx+1,sqrt(chiwx*
                psiwx))./besselk(lambdawx,sqrt(chiwx*psiwx));% this
                is nx1 matrix: E[W]
131         b = sqrt(psiwx./chiwx).*besselk(lambdawx-1,sqrt(chiwx*
                psiwx))./besselk(lambdawx,sqrt(chiwx*psiwx));% this
                is nx1 matrix: E[W^(-1)]
132
133         %update mu,Sigma,gammas
134         gamma = mean(b.*(X-mean(X)))/(1-mean(b)*mean(a));
135         mu = mean(X)-mean(a)*gamma;
136         Sigma = transpose(X-mu)*(b.*(X-mu))/n-mean(a)*transpose(
                gamma)*gamma;
137
138     end
139     if iter == iterations - 1
140         gamma_n_1 = gamma;
141         mu_n_1 = mu;
142         Sigma_n_1 = Sigma;
143         chi_n_1 = chi;
144         psi_n_1 = psi;
145         lambda_n_1 = lambda;
146     end
147     if iter == iterations
148         gamma_n = gamma;
149         mu_n = mu;
150         Sigma_n = Sigma;
151         chi_n = chi;
```

```
152             psi_n = psi;
153             lambda_n = lambda;
154         end
155     end
156     % error sanity checks
157     error_gamma = norm((gamma_n./gamma_n_1 - 1));
158     error_mu = norm((mu_n./mu_n_1 - 1));
159     error_Sigma = norm((Sigma_n./Sigma_n_1 - 1));
160     error_chi = norm((chi_n./chi_n_1 - 1));
161     error_psi = norm((psi_n./psi_n_1 - 1));
162     error_lambda = norm((lambda_n./lambda_n_1 - 1));
163
164     %simulation of final distribution
165
166     W = gigrnd(lambda,psi,chi,nsimulations); % generate W
167     Z = mvnrnd(zeros(1,d), Sigma, nsimulations); % generate Z with
        designed Sigma
168     Xsim = mu+W*gamma+sqrt(W).*Z; % combine to give X
169
170     Xsimsum = sum(Xsim,2); % find rowsums to calculate portfolio
        losses
171     sorted_lossesXsim = sort(Xsimsum,'descend'); % sort portfolio
        losses
172     num_losses = numel(sorted_lossesXsim); % count portfolio losses
173     VaR_index = floor((1-confidence_level_VaR)*num_losses)+1; %
        Calculate the index of the sorted losses that will be VaR
174     ES_index = floor((1-confidence_level_ES)*num_losses)+1; %
```

```
    Calculate the index of the sorted losses that will contribute
     to ES
175 VaRsumXsim = sorted_lossesXsim(VaR_index)*dollarvalue; % Use the
     index to extract VaR from sorted losses
176 ES1Xsim =(sorted_lossesXsim(1:ES_index-1)); % Extract the loss
    tail
177 ESsumXsim = mean(ES1Xsim)*dollarvalue; % average of loss tail to
     calculate ES
```

A.2 EM algorithm: testing on W

```
1 %This is the EM algorithm for estimating lambda, chi and psi
    while mu, Sigma, gamma are fixed.
2
3 format long;
4
5
6 %Simulating X for data
7
8 d = 4; %number of stocks
9 n = 1000; % number of data points
10 % Important: a larger n allows to see the estimated parameters
    closer to true values!
11
12 gamma = [0,1,4,-2];
13 mu = [2,1,-1,0];
14 A = [2 0 0 1; 0 1 0 0; 0 0 4 1; 1 0 1 4];
```

```
15 Sigma = A/nthroot(det(A),d); %for purpose of comparison, we
       adopt det(Sigma)=1
16 chi = 2;
17 psi = 0.5;
18 lambda = -1;
19 W = gigrnd(lambda,psi,chi,n); % generate W
20 Z = mvnrnd(zeros(1,d), Sigma, n); % generate Z with designed
       Sigma
21 X = mu+W*gamma+sqrt(W).*Z; % combine to give X
22 clear W;
23 clear Z;
24
25
26 %initial values
27
28 chi = 1;
29 psi = 1;
30 %settle initial value of lambda
31 fun = @(lambda) besselk(lambda+1,1)-besselk(lambda,1)*nthroot(
       det(cov(X)),d);
32 lambda = fzero(fun,-0.5);
33 clear fun;
34
35 %Excute the EM algorithm
36
37
38 k = 1000; % number of partial updates on Q2
```

```
39 % Important: a large k will allow us to see whether the
       algorithm converges!
40
41 for j=1:k %work on Q2 and partially update lambda,chi,psi for k
       times
42
43         %compute the density of W_i|x_i
44         lambdawx = lambda-d/2;
45         chiwx = chi+sum(((-mu+X)/Sigma).*(-mu+X),2); % this is
              nx1 matrix
46         psiwx = psi+gamma/Sigma*transpose(gamma);
47
48         %compute a,b,c
49         a = sqrt(chiwx/psiwx).*besselk(lambdawx+1,sqrt(chiwx*
              psiwx))./besselk(lambdawx,sqrt(chiwx*psiwx));% this
              is nx1 matrix: E[W]
50         b = sqrt(psiwx./chiwx).*besselk(lambdawx-1,sqrt(chiwx*
              psiwx))./besselk(lambdawx,sqrt(chiwx*psiwx));% this
              is nx1 matrix: E[W^(-1)]
51         c = zeros(n,1);% this is nx1 matrix: E[ln W]
52         for i=1:1:n
53             fun1 = @(w) log(w).*w.^(lambdawx-1).*exp(-0.5*psiwx*
                  w).*exp(-0.5*chiwx(i,1)*w.^(-1));
54             c(i,1) = 0.5*(psiwx/chiwx(i,1))^(0.5*lambdawx)/
                  besselk(lambdawx,sqrt(chiwx(i,1)*psiwx))*integral
                  (fun1,0,Inf);
55             clear fun1;
```

```
56          end
57
58          %update lambda, chi, psi
59          options = optimoptions('fmincon','Algorithm','sqp');
60          fun2 = @(t) t(1)*log(t(2))-t(1)*log(t(3))+2*log(besselk(
                t(1),sqrt(t(2)*t(3))))+t(3)*mean(a)+t(2)*mean(b)-2*t
                (1)*mean(c);
61          A = [0,-1,0;0,0,-1];
62          B = [0;0];
63          Aeq = [];
64          beq = [];
65          lb = [];
66          ub = [];
67          nonlcon = [];
68          t0 = [lambda,chi,psi];
69          t = fmincon(fun2,t0,A,B,Aeq,beq,lb,ub,nonlcon,options);
70          clear fun2;
71          clear a;
72          clear b;
73          clear c;
74          clear lambdawx;
75          clear chiwx;
76          clear psiwx;
77
78          %print out the result
79          fprintf(2,'\nstep\n')
80          step = j
```

```
81          lambda = t(1)
82          chi = t(2)
83          psi = t(3)
84  end
85
86
87  %print true values of parameters for comparison
88  %fprintf(2,'\ntrue values\n')
89  %chi=3
90  %psi=2
91  %lambda =-4
```

A.3 EM algorithm: testing with W as fixed

```
1  %This is the EM algorithm for estimating mu, Sigma, gamma while
       lambda, chi and psi are fixed.
2
3  format long
4
5  %Simulating X for data
6
7  d = 4; %number of stocks
8  n = 100000; % number of data points.
9  % Important: a larger n allows to see the estimated parameters
       closer to true values!
10
11 gamma = [0,4,0,4];
```

```
12 mu = [2,10,-1,0];
13 A = [2 0 0 1; 0 1 0 0; 0 0 4 1; 1 0 1 4];
14 Sigma = A/nthroot(det(A),d); %for purpose of comparison, we
       adopt det(Sigma)=1
15 chi=2;
16 psi=5;
17 lambda =-3;
18 %It seems some combination of values of lambda, chi, psi can
       cause problems for the algorithm.
19 W = gigrnd(lambda,psi,chi,n); % generate W
20 Z = mvnrnd(zeros(1,d), Sigma, n); % generate Z with designed
       Sigma
21 X = mu+W*gamma+sqrt(W).*Z; % combine to give X
22
23
24 % initial values
25
26 gamma = zeros(1,d);
27 mu = mean(X);
28 Sigma = cov(X)/nthroot(det(cov(X)),d);
29 %small influence on initial values
30
31 %Excute the EM algorithm
32
33          %compute the density of W_i|x_i
34          lambdawx = lambda-d/2;
35
```

```
36 k = 10000; % number of partial updates on Q1;
37 % Important: a large k will allow us to see whether the
       algorithm converges!
38
39 for j = 1:k % work on Q_1 and partially update mu,Sigma,gamma
       for k times
40   fprintf(2,'\nstep\n')
41         j
42         %compute the density of W_i|x_i
43         chiwx = chi+sum(((-mu+X)/Sigma).*(-mu+X),2); % this is
              nx1 matrix
44         psiwx = psi+gamma/Sigma*transpose(gamma);
45
46         %compute a,b
47         a = sqrt(chiwx/psiwx).*besselk(lambdawx+1,sqrt(chiwx*
              psiwx))./besselk(lambdawx,sqrt(chiwx*psiwx));% this
              is nx1 matrix: E[W]
48         b = sqrt(psiwx./chiwx).*besselk(lambdawx-1,sqrt(chiwx*
              psiwx))./besselk(lambdawx,sqrt(chiwx*psiwx));% this
              is nx1 matrix: E[W^(-1)]
49
50         %update mu,Sigma,gammas
51         gamma = mean(b.*(X-mean(X)))/(1-mean(b)*mean(a))
52         mu = mean(X)-mean(a)*gamma
53         Sigma = transpose(X-mu)*(b.*(X-mu))/n-mean(a)*transpose(
              gamma)*gamma;
54         Sigma = Sigma/nthroot(det(Sigma),d)
```

```
55
56 end
57
58 %print true values of parameters for comparison
59 fprintf(2,'\ntrue values\n')
60 gamma = [0,4,0,4]
61 mu = [2,10,-1,0]
62 A = [2 0 0 1; 0 1 0 0; 0 0 4 1; 1 0 1 4];
63 Sigma = A/nthroot(det(A),d)  %for purpose of comparison, we
       adopt det(Sigma)=1s
```

Appendix B

Codes for Chapter 3

B.1 Code for Marginals and Copulas

```
close all
pricematrix = [AAPL.AdjClose BAC.AdjClose F.AdjClose SU.AdjClose
    ]; % load price matrix
[n,d] = size(pricematrix); % calculate dimensions of price
    matrix
returnmatrix = zeros(n-1,d); % initializing return matrix
names = {'AAPL','BAC','F','SU'}; % names of stocks
nsimulations = 10000; % number of simulations to run
confidence_level_VaR = 0.99; % confidence level for VaR
confidence_level_ES = 0.975; % confidence level for ES
dollaramount = 10000; % dollar amount for each stock
theta = 3; % for Gumbel Copula

for i = 1:n-1 % loop over time
    returnmatrix(i,:) = -log(pricematrix(i+1,:)./pricematrix(i
```

```
        ,:)); % calculate losses
14 end
15 meanvec = mean(returnmatrix)*dollaramount; % find mean of dollar
       losses
16 stdvec = std(returnmatrix)*dollaramount; % find standard
      deviation of dollar losses
17 figure
18 counter = 0; % initialized counter
19 for i = 1:d % loop over stocks
20     x = [-3*stdvec(i)+meanvec(i):0.001:3*stdvec(i)+meanvec(i)];
          % marginals
21     norm = normpdf(x,meanvec(i),stdvec(i)); % calculate density
22     counter = counter+1; % increase counter
23     subplot(2,2,counter) % initialize subplot
24     plot(x,norm) % plot marginal
25     xlabel('Returns');
26     ylabel('Density');
27     mytitleText = ['Normal Distribution plot for ',names(i)];
28     title(mytitleText,'Interpreter','tex');
29     y = meanvec(i) + randn(nsimulations,1)*stdvec(i); % simulate
           marginals
30     sorted_losses = sort(y,'descend'); % sort marginal losses
31     num_losses = numel(sorted_losses); % count marginal losses
32     VaR_index = floor((1-confidence_level_VaR)*num_losses)+1; %
          Calculate the index of the sorted losses that will be VaR
33     ES_index = floor((1-confidence_level_ES)*num_losses)+1; %
          Calculate the index of the sorted losses that will
```

```
        contribute to ES
34      VaR(i) = sorted_losses(VaR_index); % Use the index to
        extract VaR from sorted losses
35      ES1 =(sorted_losses(1:ES_index-1)); % Extract the loss tail
36      ES(i) = mean(ES1); % average of loss tail to calculate ES
37  end
38
39  correlationmatrixZ = corr(returnmatrix,'type','Spearman'); %
    correlation matrix for Gauss Copula
40  correlationmatrixT = sin(0.5*pi*corr(returnmatrix,'type','
    Kendall')); % correlation matrix for t Copula
41  gamma = (cos(pi/(2*theta)))^theta; % gamma value for Gumbel
    Copula
42
43  Z = mvnrnd(zeros(1,d),correlationmatrixZ,nsimulations); %
    simulate multivariate random normal for Gauss copula
44  T = mvtrnd(correlationmatrixT,5,nsimulations); % simulate
    multivariate random t for t copula
45  U = rand(nsimulations,4);
46  Gu = [];
47  while length(Gu) < nsimulations
48      Gurand = random('stable',1/theta,1,gamma,0,1); %simulate
        stable random variates for Gumbel copula
49      if Gurand > 0
50          Gu = [Gu ;Gurand];
51      end
52  end
```

```
53
54 simulatedmatrixZ = normcdf(Z); % evaluate normal cumulative
       distribution function to generate Gauss copula
55 simulatedmatrixT = tcdf(T,5); % evaluate t cumulative
       distribution function to generate t copula
56 simulatedmatrixGu = exp(-((-log(U)./Gu).^(1/theta))); % get
       generating vector for Gumbel copula
57
58 Xz = []; % initialize loss matrix for Gauss copula
59 Xt = []; % initialize loss matrix for t copula
60 Xgu = []; % initialize loss matrix for Gumbel copula
61 for i = 1:d
62     Xz = [Xz norminv(simulatedmatrixZ(:,i),meanvec(i),stdvec(i))
          ]; % plug in marginals to get losses for Gauss copula
63     Xt = [Xt norminv(simulatedmatrixT(:,i),meanvec(i),stdvec(i))
          ]; % plug in marginals to get losses for t copula
64     Xgu = [Xgu norminv(simulatedmatrixGu(:,i),meanvec(i),stdvec(
          i))]; % plug in marginals to get losses for Gumbel copula
65 end
66 Xzsum = sum(Xz,2); % find rowsums to calculate portfolio losses
       for Gauss copula
67 Xtsum = sum(Xt,2); % find rowsums to calculate portfolio losses
       for t copula
68 Xgusum = sum(Xgu,2); % find rowsums to calculate portfolio
       losses for Gumbel copula
69 sorted_lossesz = sort(Xzsum,'descend'); % sort portfolio losses
       for Gauss copula
```

```
70 sorted_lossest = sort(Xtsum,'descend'); % sort portfolio losses
   for t copula
71 sorted_lossesgu = sort(Xgusum,'descend'); % sort portfolio
   losses for Gumbel copula
72 num_losses = numel(sorted_lossesz); % count portfolio losses
73 VaR_index = floor((1-confidence_level_VaR)*num_losses)+1; %
   Calculate the index of the sorted losses that will be VaR
74 ES_index = floor((1-confidence_level_ES)*num_losses)+1; %
   Calculate the index of the sorted losses that will contribute
    to ES
75 VaRsumz = sorted_lossesz(VaR_index); % Use the index to extract
   VaR from sorted losses for Gauss copula
76 ES1z =(sorted_lossesz(1:ES_index-1)); % Extract the loss tail
   for Gauss copula
77 ESsumz = mean(ES1z); % average of loss tail to calculate ES for
   Gauss copula
78 VaRsumt = sorted_lossest(VaR_index); % Use the index to extract
   VaR from sorted losses for t copula
79 ES1t =(sorted_lossest(1:ES_index-1)); % Extract the loss tail
   for t copula
80 ESsumt = mean(ES1t); % average of loss tail to calculate ES for
   t copula
81 VaRsumgu = sorted_lossesgu(VaR_index); % Use the index to
   extract VaR from sorted losses for Gumbel copula
82 ES1gu =(sorted_lossesgu(1:ES_index-1)); % Extract the loss tail
   for Gumbel copula
83 ESsumgu = mean(ES1gu); % average of loss tail to calculate ES
```

```
for Gumbel copula
```

References

[1] Abramowitz, M., and Stegun, I. *Handbook of Mathematical Functions.* Dover Publications INC., 1972.

[2] Adamko, P., Spuchl'áková, E., and Valáškováb, K. The history and ideas behind VaR. *Procedia Economics and Finance 24* (2015), 218–24.

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

[4] Black, F., and Scholes, M. The pricing of options and corporate liabilities. *The Journal of Political Economy 81(3)* (1973), 637–654.

[5] Brandtner, M., Kürsten, W., and Rischau, R. Entropic risk measures and their comparative statics in portfolio selection: Coherence vs. convexity. *European Journal of Operational Research 264(2)* (2018), 707–716.

[6] Föllmer, H., and Schied, A. *Stochastic Finance An Introduction in Discrete Time.* Walter de Gruyter GmbH & Co. KG, Berlin/New York, 2011.

[7] Gaglianone, W. P., Lima, L. R., Linton, O., and Smith, D. R. Evaluating value-at-risk models via quantile regression. *Journal of Business & Economic Statistics 29(1)* (2011), 150–160.

[8] Ghasemi, A., and Zahediasl, S. Normality tests for statistical analysis: A guide for non-statisticians. *International Journal of Endocrinology & Metabolism 10(2)* (2012), 486–489.

[9] Gupta, M., and Chen, Y. Theory and use of the em algorithm. *Foundations and Trends in Signal Processing 4(3)* (2010), 223–296.

[10] Jeff Wu, C. On the convergence properties of the em algorithm. *Annals of Statistics 11(1)* (1983), 95–103.

[11] Keller, U., and Eberlein, E. Hyperbolic distributions in finance. *Bernoulli 1 (3)* (1995), 281–299.

[12] Koreney, B. *Bessel Functions and Their Applications*. CRC Press, 2002.

[13] Lintner, J. Security prices, risk, and maximal gains from diversification. *The Journal of Finance 20(4)* (1965), 587–615.

[14] McLachlan, G., and Krishnan, T. *The EM Algorithm and Extensions*, 2 ed. Wiley-Interscience, 2008.

[15] McNeil, A. J., Frey, R., and Embrechts, P. *Quantitative Risk Management*. Princeton University Press, 2005.

[16] Prause, K. *The Generalized Hyperbolic Model: Estimation Financial Derivatives and Risk Measures*. PhD thesis, Albert-Ludwigs-Universität Freiburg i. Br., Germany, 1999.

[17] Puccetti, G., and Rüschendorf, L. Computation of sharp bounds on the distribution of a function of dependent risks. *Journal of Computational and Applied Mathematics 236(7)* (2012), 1833–1840.

[18] Rachev, S., Stoyanov, S., and Fabozzi, F. *A Probability Metrics Approach to Financial Risk Measures.* Wiley-Blackwell, 2011.

[19] Schott, J. *Matrix Analyis for Statistics.* Wiley, 1996.