Imperial College London

MEng Final Year Project

Distributionally Robust Risk Management: The Impact of Uncertainty about Uncertainty

Author:
Corina Alexandra Ciobanu

Supervisor:
Dr. Daniel Kuhn

June 2013

Abstract

Successful investments are a major challenge among investors. A rational investor aims to both maximise the expected return and minimise the expected risk of a portfolio. Since different risk measures have been developed over time, an interesting question is: How is it best to measure risk?

In this project, we are concerned with Worst-Case Value-at-Risk models as risk measures. Our approach is novel in the sense that it also considers European style options which expire beyond the investment horizon, by approximating their values as piecewise-linear functions. Thus, we derive an extension of the Worst-case Polyhedral Value-at-Risk (WPVaR) model and we provide tractable semidefinite and second-order cone program formulations.

We then extend the model by accounting for uncertainty in the distributional parameters: mean and covariance matrix. We follow two different directions: a box-type robust optimisation and a partitioned statistics approach, which can incorporate asymmetry in assets return distributions. Thus, the main novel contribution of our project is combining distributional asymmetry and options expiring beyond the investment horizon in the same portfolio optimisation problems. The robust WPVaR models are presented as tractable semidefinite programs.

Finally, we evaluate and compare our WPVaR and our box-type robust WPVaR models with other risk measures, VaR and CVaR. Our results show that the robust formulation provides significantly better results in terms of its efficient frontiers, variance of the expected actual return and stability of the optimal portfolio choice.

Acknowledgements

I would like to thank my project supervisor, Dr. Daniel Kuhn for his constant guidance throughout the project. His advice, comments and contributions have been of great value and help, while his enthusiasm has been a continuous motivation. I also wish to acknowledge Michael Hadjiyiannis and Vladimir Roitch for providing their insight into matlab optimisation frameworks.

I would also like to thank Dr. Panos Parpas for agreeing to be this project's second marker and for his helpful feedback.

Finally, I would like to thank my family for their unconditional support and encouragement throughout my studies.

Contents

List of Figures

Chapter 1

Introduction

Successful investments are a major challenge among investors – and distributing wealth over a set of assets (stocks, bonds, derivatives etc.) is a common part of them. The goal and main difficulty is to find investment decisions that provide the highest possible future wealth. A straightforward strategy is to identify the assets with highest expected return and invest as much as possible (or even the entire capital) in them. However, this may turn out to be not a very good strategy.

The challenge of successful investments lies in the fact that not only the return needs to be taken into account, but also the risk of the respective assets. Hence, investors need to identify those assets with good return-risk ratio, which provide high return and low risk at the same time. This in turn raises another challenging question — how is it best to measure risk?

A first model of the investment problem to consider both maximising the expected return and minimising the risk was formulated by Harry Markowitz in the 1950s [1]. His measure of the risk of a portfolio was the variance of its expected return. This has the advantage to reduce the problem formulation to a convex quadratic program, which can be solved efficiently. On the other hand, one of the main problems of the model is its sensitivity to distributional input parameters (mean and covariance of asset returns), amplifying any estimation errors. This is a major flaw, since these are impossible to be accurately estimated.

Another popular risk measure is the Value-at-Risk (VaR), defined as the worst loss over a given confidence level [2]. However, its computation requires complete knowledge of the probability distribution, which is almost never available in practice. Hence, different approximations have been researched, including worst-case values and robust optimisation. One of such risk measure is the Worst-Case Value-at-Risk proposed in [3], which maximises the VaR over all return distributions with given first and second moment (mean returns and covariance matrix of mean returns).

In this project we are concerned with portfolios which involve both basic assets and options (we mainly consider the most common examples, which are European call and put options). Options are commonly used financial instruments, mainly for speculation or hedging purposes.

In our work, we do not assume the exact assets return distributions are known, but rather that some statistical information about these can be inferred: the mean and covariance matrix or the mean and support information. We base our work on the Worst-Case Polyhedral Value-at-Risk (WPVaR) introduced in [4], which accounts for European style option which expire at the end of the investment horizon. Their return functions are convex piecewise-linear functions, which lead to a convex second-order cone program formulation of the WPVaR model.

The novelty of our approach is that we extend this model by taking into account European style option which expire beyond the investment horizon. To this purpose, we approximate their convex return function by an arbitrary number of linear functions, which yield a convex piecewise-linear approximation. Thus, we formulate a WPVaR model which insures over all possible distributions with given first and second moment information. This can be expressed a tractable semidefinite program (SDP).

This initial approach only accounts for uncertainty in the distribution of asset returns. Thus, it assumes known mean and covariance matrix of the expected returns. However, these are not available in practice and are also hard to accurately estimate. So, we go one step further and take into consideration uncertainty about uncertainty: what happens if we not only have uncertainty regarding the distribution itself, but also regarding its parameters? Thus, we extend our WPVaR model to account for uncertainty in the asset return distribution parameters, mean expected returns and covariance matrix. This is an important optimisation since it allows our model to incorporate asymmetry in asset return distributions, which is the case for real financial assets. We finally define and prove two new theorems about the resulting models, which can be formulated as tractable SDPs. These represent our project's *major novel contribution* — formulating portfolio optimisation problems which account for both distributional asymmetry and options which expire at or after the investment horizon.

1.1 Contributions

The main contributions of the project are summarised below:

Worst-case Polyhedral Value-at-Risk model considering the options with expiry beyond the investment horizon

The novelty of our project is that our WPVaR model also accounts for options which expire beyond the investment horizon. The derivation of the model is fully described in chapter 3. To achieve this, we approximate the options' return functions by an arbitrary number of linear functions, thus obtaining convex piecewise-linear portfolio return function. We then derive a formulation of the WPVaR which represents a tractable SDP and construct a mathematical proof that it is equivalent to the initial WPVaR model introduced in [4], for the basic case where only options at expiry are considered. Finally, we also derive a SOCP formulation of our WPVaR model.

Formulate proof for dualizing integral constraints

As part of our reformulation of the WPVaR model, we encounter the need to dualize an optimisation problem which contains integral constraints. This can be achieved by rewriting the integrals as infinite sums. We formulate a formal mathematical proof to confirm our dualization results.

Consider uncertainty regarding the parameters of the basic assets return distribution

Our formulation of the WPVaR model assumes knowledge of the mean and covariance matrix of basic asset returns distribution. This is not a realistic assumption, since in practice investors do not know these values. Hence, it is not enough to account for unknown distribution, but also for uncertainty regarding its first and second moment information. To this purpose, we extend our model using two different approaches, both detailed in chapter 4.

On one hand, we perform a box-type robust optimisation, assuming we know boundaries for the mean an covariance matrix, instead of the actual values. This leads to the formulation and proof of a new theorem, which presents the box-type robust WPVaR model as a tractable SDP.

On the other hand, we use partitioned statistics information. This allows us to model asymmetries in the return distributions of the basic assets that underly the options. This is an important contribution because real assets have skewed distributions and a good risk model should be able to capture this property. This work leads to the *main contribution and novel aspect* of our project: a distributionally robust risk model which provides a portfolio optimisation model that combines asymmetric return distributions and options expiring beyond the investment horizon. Again, we formulate and prove a new theorem which presents this result as a tractable SDP.

WPVaR models implementation

Initially, we provide implementation of the derived WPVaR model and check that its results match the ones obtained using the initial WPVaR model, when only options at expiry are considered. We also provide implementation of the new box-type robust model of our

WPVaR and of other risk measure models, such as VaR and CVaR. Finally, we provide the possibility to compare and validate these approaches by performing large scale simulations over generated and/or real historic data. All these are discussed in chapter 5.

Perform numerical tests

We perform a number of numerical tests, comparing our WPVaR and box-type robust WPVaR models with other risk measures, such as Value-at-Risk or Worst-case Conditional Value-at-Risk (CVaR). To this purpose, we use both simulated prices (to obtain results using the initially chosen distribution) and real data (with actual distribution unknown). We compare the models over large scale simulations, which indicates statistically significant results. The obtained results and comparisons are presented in chapter 6.

The main insights we gain from our experiments indicate that the WPVaR models perform better than the other risk measures. This is clearly shown by simulated tests, where the robust WPVaR formulation yields significantly better results: the estimated and actual frontier are closer to each other and to the true one and the variance in the expected actual return is much smaller. This is important since it indicates the robust WPVaR model is a better risk estimate than the others and yields more stable expected profit in reality. Another important observation is that the optimal portfolios obtained using the robust WPVaR model are much more stable compared to the others. This indicates a better stability from the robust model in terms of asset weights allocation and it is important because it saves transaction costs in reality. Although the out-of-sample tests do not indicate such a large difference between the different risk models, it shows that accounting for distributional ambiguity often yields portfolios which perform better than the others (in most cases the mean return is larger and the VaR is smaller).

1.2 Notation

The following notations are used throughout the project:

$\mathbb{R}$	the set of real numbers
$\mathbb{S}^n$	the set of symmetric matrices in $\mathbb{R}^{n*n}$
$\mathbf{v}, \mathbf{M}$	vector $\mathbf{v}$, matrix $\mathbf{M}$
$\mathbf{v}^T, \mathbf{M}^T$	transpose of vector $\mathbf{v}$, transpose of matrix $\mathbf{M}$
$\langle \mathbf{A}, \mathbf{B} \rangle$	trace of matrix $\mathbf{A} * \mathbf{B}$
$\mathbf{e}$	vector with all entries 1
$\mathbf{0}_{m,\ n}$	the m by n matrix with all elements zero
$[\cdot]_+$	denotes the max $\{\cdot\ ,\ 0\}$
$\\|\cdot\\|_n$	denotes the L^n norm
$\\|\cdot\\|$	denotes the Euclidean (L^2) norm
$1_{f(x)}$	the indicator function
$\mathbb{P}\{\cdot\}$	probability of input $\cdot$

Chapter 2

Background

Investment science represents the application of scientific tools or methods to investment — a commitment of resource with the purpose of achieving later benefits. Portfolio optimisation is a process of selecting an investment decision, such that the financial instruments acquired meet some conditions (mainly having a high possible profit and a low risk).

This chapter introduces some basic concepts related to portfolio optimisation, as well as different approaches and models for optimising a portfolio's return. A portfolio refers to a collection of financial instruments (stocks, bonds etc.) held by an investor, with the purpose of obtaining a payoff greater than the initial investment. Then, we introduce some basic mathematical background required throughout project.

2.1 Modern Portfolio Theory

Modern Portfolio Theory is a theory of investment science which seeks to maximise the expected return of a portfolio based on a given portfolio risk. It is assumed that investors are *risk averse* (between two assets with same expected return but different risks, a risk-averse investor always chooses the one with smallest risk) and that the *no-arbitrage* condition holds (it is not possible to earn money without investing anything — no risk-free profits). An asset represents an investment instrument which can be bought or sold.

The return of an asset is thus defined as:

$$R = \frac{X_1}{X_0}$$

where X_1 is the amount received from the asset and X_0 is the amount invested in the asset.

The rate of return of an asset is then:

$$r = \frac{X_1 - X_0}{X_0}$$

However, the rate of return is often referred to simply as the return of the asset. We shall also use this convention in our project. The context should make it clear if we refer to the rate of return.

A portfolio is a master asset, consisting of a number of n assets. Thus, the portfolio total return and rate of return are defined as:

$$R = \sum_{i=1}^{n} w_i R_i \quad , \qquad r = \sum_{i=1}^{n} w_i r_i$$

where w_i is the fraction of asset i in the portfolio (the sum invested in asset i, $X_{0i} = w_i X_0$). Note that this requires $\sum_{i=1}^{n} w_i = 1$.

Since the rate of return is usually unknown, this is considered to be a random variable $\tilde{r}$. Analytically, its expected value $\bar{r}$ is used.

Then, the expected return of a portfolio can be expressed in terms of the expected return of each asset (let $E(r_i)$ be the expected return of asset i):

$$E(r) = \sum_{i=1}^{n} w_i E(r_i)$$

It comes natural that investors wish to maximise this expected return, thus expecting as high profit as possible. However, there is another important factor which should not be overlooked — the risk of an asset. As mentioned in our earlier assumptions, there are no risk-free profits, so the question resumes to: how much risk should be taken for a higher possible profit?

The *Markowitz* model [1] was among the first mathematical models to address the problem of distributing wealth over assets in order to obtain highest possible profit. This included both maximising the expected return and minimising the risk of the investment. In this case, risk was measured by the variance of the expected return of the portfolio (which also lead to the alternative name of the *Mean-Variance Model*):

$$\sigma^2 = \sum_{i,j=1}^{n} w_i w_j \sigma_{ij}$$

where σ_{ij} is the covariance of asset i and asset j and σ^2 is the variance of the expected return, representing the risk. Thus, the Markowitz problem can be formulated as follows [5]:

$$\min \frac{1}{2} \sum_{i,j=1}^{n} w_i w_j \sigma_{ij}$$

$$s.\ t.\ \sum_{i=1}^{n} \omega_i \bar{r}_i = \bar{r}$$

$$\sum i = 1^n = 1$$

The feasible set of such problem is the set of points (which represent portfolios) which satisfy the requirements of the weights of the assets (in the simplest case: $\sum_{i=1}^{n} \omega_i = 1$). We can represent this set as the mean return in terms of the variance. Then, the left boundary of this set is called the minimum-variance set, since for any given return, the portfolio which yields that mean return with minimum variance is on the left boundary of the feasible set. The point on this set with smallest variance is called the minimum-variance point. The upper part of the minimum-variance set (from the minimum-variance point upwards) is called the *efficient frontier* of the feasible region [5]. This is the only part of the feasible set interesting to investors, since it contains the portfolios with lowest variance given a mean return and the portfolios with highest mean return, given a variance.

This risk model is a good approach when the returns are symmetrically distributed, but it does not perform as well as a risk measure on skewed portfolio returns. Figure 2.1 shows a comparison between a normal and a skewed distribution.

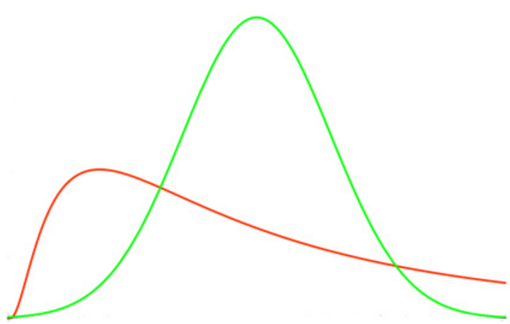


Figure 2.1: Comparison between a normal distribution (in green) and a skewed one (in red)

A popular and intuitive risk measure without this limitation is the Value-at-Risk (VaR) [6]. For a given portfolio, VaR at level ϵ is defined as the smallest number v, such that the probability that the loss of the portfolio is smaller than v is less than $1 - \epsilon$ (VaR is the least $(1 - \epsilon)$ quantile of the loss distribution of the portfolio. Figure 2.2 illustrates the 5% VaR point of a given distribution.

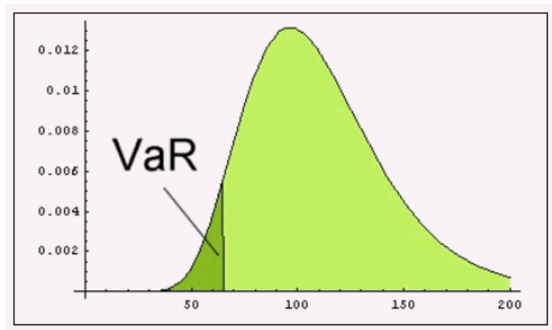


Figure 2.2: 5% Value-at-Risk represented by the dark green area (5% of the total area under the curve)

A different approach is to consider the Conditional-Value-at-Risk (CVaR) as a risk measure, instead of VaR. The β-CVaR [7] is defined as the conditional expectation of losses which are above a value α, where α is the β-VaR of the same portfolio. Due to this definition, the VaR is never larger than the CVaR — so a portfolio with low CVaR also has low VaR (for a certain probability level β). Due to its computational advantages over VaR (presented in [7], mainly the fact that in most contexts the CVaR can be expressed as a *remarkable minimisation formula*), more work has been done into generalising this approach for loss distributions which are not necessarily continuous [8].

A downside of the VaR approach presented above is that it assumes that the entire distribution of returns of the assets is known. Unfortunately, this is never the case in practice and the data uncertainty encountered in real life situations can affect any of the two risk models presented (variance and VaR).

To address this issue, [3] introduces the Worst-Case Value-at-Risk (WVar) – a pessimistic approach of the true VaR, obtained by maximising VaR over all possible return distributions consistent with known bounds on the mean and covariance. Their model of the WVaR is computed as a semidefinite program for many probability distribution sets, which is one of the main contributions.

An interesting part of the paper is represented by *Theorem 1*, which states:

Let $\mathcal{P}$ be the set of probability distributions with mean $\hat{x}$ and covariance matrix $\Gamma \succ 0$. Let $\epsilon \in (0, 1]$ and $\gamma \in \mathbb{R}$ be given. The following propositions are equivalent:

1. omitted

2. *We have:*

$$k(\epsilon)\|\Gamma^{1/2}\omega\|_2 - \hat{x}^T\omega \leq \gamma, \tag{2.1}$$

where $k(\epsilon) := \sqrt{\frac{1-\epsilon}{\epsilon}}$

3. *There exist a symmetric matrix $M \in \mathcal{S}_{n+1}$ and $\tau \in \mathbb{R}$ such that:*

$$\langle M, \Sigma \rangle \leq \tau\epsilon, \ M \succ 0, \ \tau \geq 0,$$
$$M + \begin{bmatrix} 0 & \omega \\ \omega^T & -\tau + 2\gamma \end{bmatrix} \tag{2.2}$$

There are two more propositions in the original paper, but we are only concerned with these two. The reason for that is that this equivalence allows us later to obtain a tractable SOCP (second-order cone program) from a semidefinite program.

[3] also indicates a connection to robust optimisation, by considering an uncertainty set against which the WVaR is computed.

2.2 Robust Optimisation

Robust optimisation is a paradigm for decision problems which depend on non-stochastic data uncertainty [9]. The uncertain values, although unknown, are assumed to belong to an *uncertainty set*. Thus, robust optimisation techniques find the best solution with regards to the worst-case values (of the uncertain parameters) within these sets. These techniques can be applied to different kinds of models, to different approaches of considering portfolio risk.

One approach to account for portfolio risk, besides the VaR models, relates to expected utility. Expected utility models decide between different risky investments by comparing their expected utility values. The expected utility value is a weighted sum considering utility values and their respective probability.

Natarajan et. al. [10] extend these models in portfolio optimisation by relaxing the assumption that the distribution of returns is fully known. Instead, they assume knowledge of only the mean, variance and support information. They obtain encouraging results for the robust approach based on the first two moments, for more complicated utility functions. They also provide extensions to the model, by using partitioned statistics or considering uncertainty in the provided mean and covariance.

A different approach to portfolio optimisation based on VaR models is presented in [11]. This separates the asset return distribution into two half-spaces (positive and negative) and minimises a new risk measure – Partitioned Value-at-Risk (PVar), using statistical data from the these two half-spaces. PVaR is an extension of the WVaR presented in [3], able to cope with ambiguity and asymmetry in the distributions. Hence, it provides empirical improvements over WVaR on skewed return distributions.

An interesting point regarding robustness is made in [12], where attention is drawn upon the "price of robustness". Bertsimas et. al. show concern that robust optimisation approaches may be too conservative. This occurs due to the requirement of finding a solution which remains feasible and close to optimal when data changes occur due to uncertainty. Their proposed solution is an approach which allows flexibility on the degree of conservatism for every constraint. Their model is guaranteed to provide a solution that remains feasible with respect to a certain constraint if less than a specified number of its uncertain parameters change. Moreover, their solution will remain feasible with high probability even if more parameters changes.

However, one of the main drawbacks of the WVaR (and PVaR implicitly) is that it requires first- and second-order moments information. This becomes even more of a difficulty when considering portfolios which contain derivatives – non-linear portfolios.

2.3 Non-linear Portfolios

Non-linear portfolios have a non-linear payoff structure. This is the case for portfolios which contain assets with (highly) non-linear payoffs, such as derivatives. Derivatives are financial instruments based on underlying assets. European options are the most common derivative considered for mathematical models for portfolio optimisation. Studying such portfolios is very important since investors often include derivatives in their portfolios, for different purposes (risk management, hedging etc.).

An option represents the right (not obligation) to buy or sell the underlying asset, within specified conditions [5]: the strike price (the price at which the underlying asset can be bought/sold), the expiration date etc. An European option allows this right to be exercised only at maturity (the expiration date), while an American option allows this at any time before this date (including the date itself). Depending on its type, an option can be a *call* option (gives the right to buy something) or a *put* option (gives the right to sell something).

One optimisation model for portfolios including European options is shown in [13]: the portfolio is enriched with derivatives (European options precisely), in order to obtain a deterministic lower bound on the return of the portfolio. This provides a *strong guarantee*, since it holds for all possible asset returns, which complements the *weak guarantee* of robust portfolios. Robust portfolios provide a weak guarantee only, since they do not insure against asset returns realised outside the uncertainty set. In this case, the realised portfolio return may actually be less than the computed worst-case portfolio return.

The model imposes this strong guarantee by using the fact that options prevent the portfolio's value from falling below a certain value. For example, consider a portfolio with one stock and one put option with strike price K (and with the underlying asset same as the stock in the portfolio). Then, the portfolio's payoff is:

$$V_{pf} = S_T + V_{put}(S_T) = max\{S_T,\ K\}$$

where S_T is the stock price at the investment horizon (at time T) and $V_{put}(S_T)$ is the value of the put option at time T.

This shows how the put option keeps the given portfolio's return value at expiration from dropping below K. Similarly, a call option keeps it from dropping below $-K$.

Another model which considers derivatives is shown in [4], which introduces two new approximations: Worst-case Polyhedral Value-at-Risk (WPVaR) and Worst-case Quadratic Value-at-Risk (WQVaR). This is achieved by computing the worst-case VaR over all return distribution of the derivative underliers (given the first- and second-order moments).

The WPVar is a conservative approximation of the VaR of a portfolio, with the addition of European options which expire at the end of the investment horizon. In this case, short-sales of options are not allowed (the process of short selling refers to the possibility of selling an asset without possessing it, by borrowing it from someone else). Also, the option returns are convex piecewise-linear function of the underlying asset return. Figure 2.3 shows the two return functions, for call and put options.

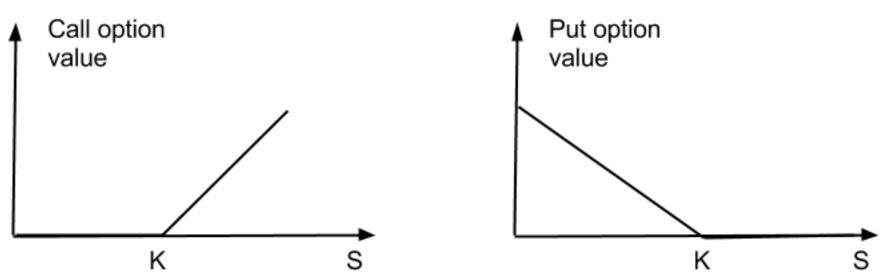


Figure 2.3: Value of options at expiration

To see how the values are computed, suppose we have an option with strike price K on an underlying stock with price S at expiration of the option. Then, the value of the option depends on its type and the relation between K and S. For a call option, if $S < K$ its value is 0 (since buying the stock at its current price S is cheaper than exercising the option and buying it at price K). Otherwise, its value is the difference $S - K$, since one could make this profit by exercising the option and buying the stock at price K and then selling it at its current higher price S. A very similar reasoning can be done for put options – since in this case we have the right to sale, the option has positive value when the current underlying stock price S is smaller than the strike price K.

The following steps for deriving WPVaR involve expressing the returns of the portfolio assets in terms of the basic asset returns only (this is possible since the derivatives' uncertainty is caused solely by the uncertainty of their underlying asset returns). Using the above options value function, the vector of asset returns can then be expressed as:

$$\tilde{\mathbf{r}} = f(\tilde{\boldsymbol{\xi}}) = \begin{pmatrix} \tilde{\xi} \\ \max\{-\mathbf{e},\ \mathbf{a} + \mathbf{B}\tilde{\boldsymbol{\xi}} - \mathbf{e}\} \end{pmatrix} \tag{2.3}$$

where $\tilde{\mathbf{r}}$ is the vector of asset returns, $\tilde{\boldsymbol{\xi}}$ is the vector of basic asset returns, $\mathbf{e}$ is the vector of $1s$ and max is component-wise maximisation. The vector $\mathbf{a}$ and matrix $\mathbf{B}$ are constants obtained by expressing the option return in terms of its underlying asset return.

For example, let asset j be a call option on basic asset i with initial price c_j and with strike price k_j. The price of the asset i at expiration (which is at the end of investment horizon) is $s_i(1+\tilde{\xi}_i)$, where s_i is the asset's initial price and $\tilde{\xi}_i$ is the asset's return. Then, by using the option's value at expiration, we can derive the following formula for f for the j_{th} asset:

$$f_j(\tilde{\boldsymbol{\xi}}) = \tfrac{1}{c_j} max\{-1, a_j + b_j\tilde{\xi}_i - 1\}$$

where $a_j = \frac{s_i - k_j}{c_j}$ and $b_j = \frac{s_i}{c_j}$.

Similar computation can be performed to obtain the values of $\mathbf{a}$ and $\mathbf{B}$ for the put options.

The Worst-case Polyhedral VaR is eventually defined as:

$$WPVaR_\epsilon(\boldsymbol{\omega}) = min\{\gamma : \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\{\gamma \leq -\boldsymbol{\omega}^T f(\tilde{\boldsymbol{\xi}})\} \leq \epsilon\} \tag{2.4}$$

where $\boldsymbol{\omega}$ is the vector of asset weights, $\mathcal{P}$ represents the set of all probability distributions of $\tilde{\boldsymbol{\xi}}$ with given mean vector and covariance matrix.

Replacing f by its value shown in (2.3), the WPVaR can be evaluated as the optimal value of a tractable SOCP (second-order cone program).

2.4 Mathematical Overview

An overview of the main concepts and results required throughout the project is given below. A basic mathematical background is assumed.

2.4.1 Matrices Results

A matrix is a rectangular array of numbers. A square matrix has equal number of rows and columns. The transpose of a matrix $\mathbf{A}_{m*n}$ is another matrix $\mathbf{A}^T_{n*m}$, which is obtained by writing the rows of $\mathbf{A}$ as columns of $\mathbf{A}^T$.

Below is a brief description of the mostly used properties of matrices:

Symmetric Matrix

A symmetric matrix is a square matrix which is equal to its transpose. We will write $\mathbb{S}^n$ for the set of symmetric matrices of size $n * n$.

Positive Semidefinite Matrix

A matrix $\mathbf{A}_{n*n}$ is positive semidefinite, written as $\mathbf{A} \succcurlyeq 0$, if $\forall \mathbf{v} \neq 0,\ \mathbf{v} \in \mathbb{R}^n :\ \mathbf{v}^T\mathbf{A}\mathbf{c} \geq 0$. If the inequality is strictly satisfied for all values of $\mathbf{v}$, then the matrix is positive definite ($\mathbf{A} \succ 0$).

Trace of a Matrix

The trace of a square matrix is the sum of the elements on the principal diagonal: $tr(\mathbf{A_{n*n}}) = a_{11} + a_{22} + \cdots + a_{nn}$.

The trace of a product of two matrices, say $\mathbf{A}$ and $\mathbf{B}$, will be represented in our project as: $\langle\mathbf{A},\ \mathbf{B}\rangle$. We will use the following properties of the trace of a matrix (where $\mathbf{A}$ and $\mathbf{B}$ are matrices and c is a constant):

$$tr(c\mathbf{A}) = c\ tr(\mathbf{A})$$
$$tr(\mathbf{AB}) = tr(\mathbf{BA})$$

2.4.2 Linear Programming

Linear programming is a class of optimisation problems. A mathematical optimisation problem can be described, in its simplest form [14]:

$$
\begin{aligned}
&\textit{minimise } f_0(\mathbf{x}) \\
&\textit{subject to } f_i(\mathbf{x}) \leq b_i \quad \forall i = 1, \ldots, m
\end{aligned}
$$

where $\mathbf{x} \in \mathbb{R}^n$ is the optimisation variable, f_0 is the *objective function* and f_i, b_i are the constraint functions and constraint bounds respectively.

A linear program (LP) has linear objective and linear constraint functions (they can be expressed as a linear combination of $\mathbf{x}$). Hence, it is of the form [14]:

$$
\begin{aligned}
&\textit{minimise } \mathbf{c}^T\mathbf{x} \\
&\textit{subject to } \mathbf{a_i}^T\mathbf{x} \leq b_i \quad \forall i = 1, \ldots, m
\end{aligned}
$$

The aim is to find a particular value for $\mathbf{x}$ which minimises the objective function. A common way to define a linear program is using its matrix form:

$$
\begin{aligned}
\min\ & \mathbf{c}^T\mathbf{x} \\
s.\ t.\ & \mathbf{A}\mathbf{x} \leq b \\
& \mathbf{x} \geq 0
\end{aligned}
\tag{2.5}
$$

The feasible set is the set of values of $\mathbf{x}$ which satisfy both conditions. The problem is *feasible* if the feasible set is not empty.

2.4.2.1 Duality Theory

If we consider the linear program defined in equation (2.5) as our *primal program*, then the *dual program* is defined as [15]:

$$
\begin{aligned}
\max\ & \mathbf{b}^T\mathbf{y} \\
s.\ t.\ & \mathbf{A}^T\mathbf{y} \geq \mathbf{c} \\
& \mathbf{y} \geq 0
\end{aligned}
\tag{2.6}
$$

Then, the *weak duality theorem* states that the dual optimal value is bounded above by the primal optimal value:

$$\min_{\mathbf{x}} \ \mathbf{c}^T\mathbf{x} \ \geq \ \max_{\mathbf{y}} \ \mathbf{b}^T\mathbf{y}$$

where $\mathbf{x}$ and $\mathbf{y}$ belong to the feasible set of the primal and dual problem respectively.

If the problems are feasible, then the *strong duality theorem* can be defined, which states that if one of the two problems is solvable, so is the other and the optimal values coincide:

$$\min_{\mathbf{x}} \ \mathbf{c}^T\mathbf{x} \ = \ \max_{\mathbf{y}} \ \mathbf{b}^T\mathbf{y}$$

For proofs of the above the reader is referred to section 1.7.3 of [15].

2.4.3 Convex Optimisation

A convex optimisation problem is of the form [14]:

$$\begin{aligned} &\textit{minimise } f_0(\mathbf{x}) \\ &\textit{subject to } f_i(\mathbf{x}) \leq 0 \ \ \forall i = 1, \ldots, m \\ &\qquad\qquad \mathbf{a_i}^T\mathbf{x} = b_i \ \ \forall i = 1, \ldots, p \end{aligned}$$

where the functions $f_0, \ldots, f_m$ are convex. Linear programs are also convex optimisation problems (since the objective and constraint functions are all affine).

A generalisation of convex optimisation problems allows the inequality constraint functions to be vector valued and uses generalised inequalities in the constraints. *Conic form problems* are a simple version of such generalised problems, with linear objective and one affine inequality constraint function (hence K-convex) [14]:

$$\begin{aligned} \textit{minimise} \ \ & \mathbf{c}^T\mathbf{x} \\ \textit{subject to} \ \ & \mathbf{F}\mathbf{x} + \mathbf{g} \preccurlyeq_K 0 \\ & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned}$$

2.4.3.1 Semidefinite Program (SDP)

A semidefinite program is a conic form problem with K being the cone of positive semidefinite $k * k$ matrices. This has the form:

$$\begin{aligned} &\textit{minimise} && \mathbf{c}^T\mathbf{x} \\ &\textit{subject to} && x_1F_1 + \cdots + x_nF_n + G \preccurlyeq 0 \\ & && \mathbf{Ax} = \mathbf{b} \end{aligned}$$

A *standard form SDP* has a matrix non negativity constraint on the variable $\mathbf{X} \in \mathbb{S}^n$ and linear equality constraints [14]:

$$\begin{aligned} &\textit{minimise} && tr(\mathbf{CX}) \\ &\textit{subject to} && tr(\mathbf{A_iX}) = b_i \quad \forall i = 1, \ldots, p \\ & && \mathbf{X} \succcurlyeq 0 \end{aligned}$$

2.4.4 Probability Theory

A random variable is a mapping from a sample space S to the real numbers. A probability measure P defined on S induces a probability distribution on a random variable X defined on S:

$$P_X(X \leq x) = P(S_x)$$

where S_x is the set containing those elements of S mapped by X to numbers smaller or equal to x.

The *cumulative distribution function* (cdf) F_X of a random variable X is:

$$F_X(x) = P_X(X \leq x)$$

A random variable is continuous if $\exists f_X : \mathbb{R} \to \mathbb{R}$ such that:

$$P_x(B) = \int_{x \in B} f_X(x)dx, \quad B \subseteq \mathbb{R}$$

The function f_X is called the *probability density function* (pdf) of the random variable X.

A *probability distribution* defines a probability for each outcome or subset of outcomes of a random variable. This can be defined by either a cumulative distribution function or a probability distribution function.

The *mean* or *expectation* of a random variable is defined as:

$$E(X) = \int_{-\infty}^{+\infty} x f_X(x) dx$$

The *variance* and *standard deviation* of a random variable are defined as:

$$\begin{aligned} Var(X) &= E([X - E(X)]^2) = E(X^2) - E(X)^2 \\ Std(X) &= \sqrt{Var(X)} \end{aligned}$$

The nth moment of a probability distribution defined by its cumulative distribution function F_X is defined as:

$$E(X^n) = \int_{-\infty}^{+\infty} x^n dF_X(x) dx$$

The *covariance* and correlation between two random variables X and Y are defined as:

$$\begin{aligned} Cov(X, Y) &= E[(X - E(X))(Y - E(Y))] \\ Cor(X, Y) &= \frac{Cov(X, Y)}{Std(X)Std(Y)} \end{aligned}$$

Chapter 3

WPVaR Model

The first part of the project constitutes of formulating a model for Worst-case Polyhedral Value-at-Risk, which considers options with expiration beyond the investment horizon. However, instead of using the actual return function of these options, this is approximated by an arbitrary number of linear functions. The model can then be formulated as a semidefinite program (SDP) or as a second-order cone program (SOCP).

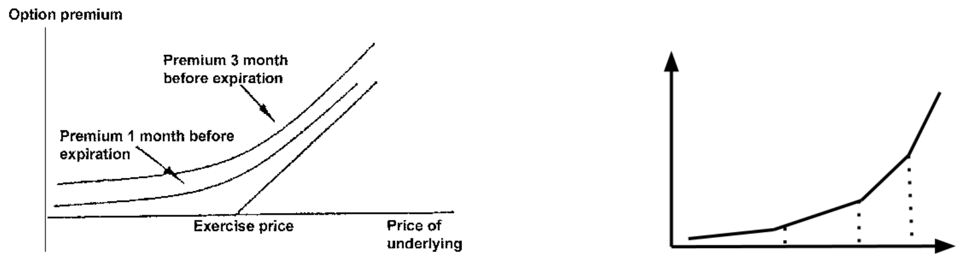
3.1 Piecewise Linear Portfolio Return

The novelty of our approach is to consider options whose expiry date is beyond the investment horizon when constructing our portfolio. As we will see, these have return which represents a convex function of the underlying asset. In order to incorporate them into our model, we use convex **p**iecewise linear approximations of these returns. We will allow any number of linear functions, which can provide an accurate representation of the true return function. This leads to a portfolio whose return is a convex piecewise linear function of the basic asset returns. Finally, we will derive a WPVaR model for such portfolio and show it can be evaluated efficiently as the optimal solution of a tractable semidefinite program or of a tractable second-order cone program.. The work generalises the result from [4] and uses a similar approach to [16] in deriving distributionally robust formulation of the WPVaR, presented in the next chapter.

3.1.1 Options Return Function

The goal is to consider option values with various times to expiration. As shown in figure 3.1a, the payoff function changes for different times to expiration and it is a smooth curve until the expiration date. The approach is to approximate the value function by an arbitrary number of linear functions. A basic example of such method is shown in figure 3.1b. Here, we can see how 4 linear functions are used to approximate a curve similar to the option value function.

To formulate this approximation analytically, assume g_k represents the functions used, with $k = 1, \ldots, G$, where G is the total number of functions used. If $V_o(S)$ is the options value



(a) Option value curve at various times (b) Piecewise linear approx.

Figure 3.1: Option value curve and piecewise linear approximation

in terms of the underlying asset's price S, then:

$$V_{option}(S) = \max_{k=1,\dots,G}\{g_k(S)\} \tag{3.1}$$

However, since we decided to use linear functions, we can say that:

$$g_k(S) = a_k S + b_k \tag{3.2}$$

Consider we want to determine the value of option j with basic underlying asset i, with initial price s_i. Then the price at expiration of the basic asset is $S = s_i(1 + \tilde{\xi}_i)$.

Using the above and (3.2) in equation (3.1), we obtain the value of the option:

$$V_j = \max_k\{a_k s_i(1 + \tilde{\xi}_i) + b_k\} \tag{3.3}$$

Next, we want to express the return $\tilde{r}_j$ of option j. As shown in [4] and described in section 2.3, $\tilde{r}_j$ can be expressed as a function f of $\tilde{\xi}$, the basic asset returns. Considering c_j the initial price of option j and using (3.3), $\tilde{r}_j$ can be expressed as the j_{th} component of f:

$$\tilde{r}_j = f_j(\tilde{\xi}) = \frac{1}{c_j}\max_k\{a_k s_i(1 + \tilde{\xi}_i) + b_k\} - 1 \tag{3.4}$$

$$f_j(\tilde{\xi}) = \max_k\{\frac{a_k s_i + b_k}{c_j} + \frac{a_k s_i}{c_j}\tilde{\xi}_i - 1\} \tag{3.5}$$

At this point we introduce $a_{jk} = \frac{a_k s_i + b_k}{c_j}$ and $b_{jk} = \frac{a_k s_i}{c_j}$, which allows the final formulation for f_j:

$$f_j(\tilde{\xi}) = \max_k\{a_{jk} + b_{jk}\tilde{\xi}_i - 1\} \tag{3.6}$$

3.1.2 Portfolio Return Formulation

From the derivation of the options' return function above, we can express the vector of expected asset returns $\tilde{\mathbf{r}}$ as:

$$\tilde{\mathbf{r}} = f(\tilde{\xi}) = \begin{pmatrix} \tilde{\xi} \\ \max_k\{\mathbf{a_k} + \mathbf{B_k}\tilde{\xi} - \mathbf{e}\} \end{pmatrix} \tag{3.7}$$

where $\mathbf{e}$ is the vector of 1s and $\mathbf{a_k}$ and $\mathbf{B_k}$ are constants determined by a_{jk} and b_{jk} from (3.6).

Finally, the expected return of the portfolio, with weight allocation into assets $\mathbf{w}$ can then be expressed as:

$$\begin{aligned} \tilde{r_p} &= \boldsymbol{\omega}^{\mathbf{T}}\tilde{\mathbf{r}} = \boldsymbol{\omega}^{\mathbf{T}} f(\tilde{\boldsymbol{\xi}}) \\ &= (\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\tilde{\boldsymbol{\xi}} + (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k} + \mathbf{B_k}\tilde{\boldsymbol{\xi}} - \mathbf{e}\} \end{aligned} \tag{3.8}$$

3.2 WPVaR Formulation

Our goal is now to use the above result to obtain a SDP formulation of the WPVaR. Replacing the $\boldsymbol{\omega}^T f(\tilde{\boldsymbol{\xi}})$ from above in the definition of WPVar from equation (2.4), we obtain the following expression for WPVaR:

$$\begin{aligned} &WPVaR_\epsilon(\boldsymbol{\omega}) \\ &= \min\{\gamma : \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\{\gamma \leq -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\tilde{\boldsymbol{\xi}} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k} + \mathbf{B_k}\tilde{\boldsymbol{\xi}} - \mathbf{e}\}\} \leq \epsilon\} \end{aligned} \tag{3.9}$$

where $\boldsymbol{\omega}$ is the vector of asset weights and $\mathcal{P}$ represents the set of all probability distributions of $\tilde{\boldsymbol{\xi}}$ with given mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. This is a distributionally robust formulation since it considers the result over all possible probability distributions.

Since $\mathbf{a_k}$ and $\mathbf{e}$ are both constants, we can combine them and reassign $\mathbf{a_k}$ as $\mathbf{a_k} - \mathbf{e}$. Then, our WPVaR formulation becomes:

$$
\begin{aligned}
&WPVaR_\epsilon(\boldsymbol{\omega}) \\
&= \min\{\gamma : \sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\{\gamma \leq -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\tilde{\boldsymbol{\xi}} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k} + \mathbf{B_k}\tilde{\boldsymbol{\xi}}\}\} \leq \epsilon\}
\end{aligned} \tag{3.10}
$$

Next, we will follow two main steps to obtain the desired result: first, we will reformulate the distributionally robust inner maximisation problem; second, we will reformulate the inner most piecewise maximisation problem, thus obtaining a semidefinite program. Finally, we will also reformulate this as a second-order cone program.

3.2.1 Dualizing Inner Maximisation Problem

Our first step to formulate this as a semidefinite program is to dualize the inner expression:

$$
\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{P}\{\gamma \leq -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\tilde{\boldsymbol{\xi}} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k} + \mathbf{B_k}\tilde{\boldsymbol{\xi}}\}\} \tag{3.11}
$$

To achieve this, we first reformulate this as the following maximisation problem, where $\boldsymbol{\mu}$ is the mean returns vector, $\boldsymbol{\Sigma}$ is the covariance matrix of the returns and 1_A is the indicator function for set A:

$$
\begin{aligned}
\sup_{\boldsymbol{\mu}\geq 0} \quad & \int_{\mathbb{R}^n} 1_{\gamma \leq -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\boldsymbol{\xi} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k}+\mathbf{B_k}\boldsymbol{\xi}\}} \ \boldsymbol{\mu} \ d\boldsymbol{\xi} \\
s.\ t. \quad & \int \boldsymbol{\mu} \ d\boldsymbol{\xi} = 1 \\
& \int \boldsymbol{\xi}\boldsymbol{\mu} \ d\boldsymbol{\xi} = \boldsymbol{\mu} \\
& \int \boldsymbol{\xi}\boldsymbol{\xi}^T \boldsymbol{\mu} \ d\boldsymbol{\xi} = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T
\end{aligned} \tag{3.12}
$$

We can dualize this problem by considering the integrals as infinite sums and dualize the new program. As a formal proof of such dualization has not been written, we present this below. Finally, we obtain the following minimisation problem:

$$\begin{aligned}
&\inf\ \alpha + \boldsymbol{\mu}^T\boldsymbol{\beta} + \langle \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T, \boldsymbol{\Gamma}\rangle \\
s.\ t.\ \ &\alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \boldsymbol{\Gamma} \in \mathbb{R}^{n*n} \\
&\alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{\Gamma}\boldsymbol{\xi} \geq 0\ \ \forall \boldsymbol{\xi} \in \mathbb{R}^n \\
&\alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{\Gamma}\boldsymbol{\xi} \geq 1\ \ \forall \boldsymbol{\xi} \in \mathbb{R}^n : \gamma \leq -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\boldsymbol{\xi} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k} + \mathbf{B_k}\boldsymbol{\xi}\}
\end{aligned} \tag{3.13}$$

Proof:

The problem from equation (3.12) can be reformulated by replacing the integrals with infinite sums. If we consider Δ to be the very small variation between neighbour values of $\boldsymbol{\xi}$, the problem becomes:

$$\begin{aligned}
\sup_{\boldsymbol{\mu}\geq 0}\ \ &\sum_i \Delta 1_{\gamma \leq -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\boldsymbol{\xi}_i - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k}+\mathbf{B_k}\boldsymbol{\xi}_i\}}\ \boldsymbol{\mu}\ d\boldsymbol{\xi}_i \\
s.\ t.\ \ &\sum_i \Delta\boldsymbol{\mu}\ d\boldsymbol{\xi}_i = 1 \\
&\sum_i \Delta\boldsymbol{\xi}_i\boldsymbol{\mu}\ d\boldsymbol{\xi}_i = \boldsymbol{\mu} \\
&\sum_i \Delta\boldsymbol{\xi}_i\boldsymbol{\xi}^T\boldsymbol{\mu}\ d\boldsymbol{\xi}_i = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T
\end{aligned} \tag{3.14}$$

Let $cond(\boldsymbol{\xi}) = \gamma \leq -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\boldsymbol{\xi}_i - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k} + \mathbf{B_k}\boldsymbol{\xi}_i\}$. Then, we can see this problem has the form $\max\{\mathbf{c}^T\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}\}$, where:

$$
\begin{aligned}
\mathbf{x}^T &= [\dots,\ \boldsymbol{\mu} d\boldsymbol{\xi}_i,\ \boldsymbol{\mu} d\boldsymbol{\xi}_{i+1},\ \dots] \\
\mathbf{b}^T &= [1,\ \mu_1,\ \mu_2,\ \dots,\ \Sigma_{11}+\mu_1^2,\ \Sigma_{12}+\mu_1\mu_2,\ \dots,\ \Sigma_{21}+\mu_2\mu_1,\ \dots] \\
\mathbf{c}^T &= \Delta\ [\dots,\ 1_{cond(\boldsymbol{\xi}_i)},\ 1_{cond(\boldsymbol{\xi}_{i+1})},\ \dots] \\
\mathbf{A} &= \Delta \begin{bmatrix}
\dots & 1 & 1 & \dots \\
\dots & \xi_{i1} & \xi_{(i+1)1} & \dots \\
\dots & \xi_{i2} & \xi_{(i+1)2} & \dots \\
\vdots & \vdots & \vdots & \vdots \\
\dots & \xi_{i1}^2 & \xi_{(i+1)1^2} & \dots \\
\dots & \xi_{i1}\xi_{i2} & \xi_{(i+1)1}\xi_{(i+1)2} & \dots \\
\dots & \xi_{i1}\xi_{i3} & \xi_{(i+1)1}\xi_{(i+1)3} & \dots \\
\vdots & \vdots & \vdots & \vdots \\
\dots & \xi_{i2}\xi_{i1} & \xi_{(i+1)2}\xi_{(i+1)1} & \dots \\
\vdots & \vdots & \vdots & \vdots
\end{bmatrix}
\end{aligned}
$$

We can now use the direct way for obtaining the dual problem (as mentioned in Chapter 2). We will add variable α corresponding to the first constraint, variables β_1, β_2, $\dots$, β_n for the following n constraints (which define the array $\boldsymbol{\beta}$) and variables Γ_{11}, Γ_{21}, $\dots$, Γ_{12}, $\dots$, Γ_{nn} for the rest of the $n * n$ constraints (thus defining the matrix $\boldsymbol{\Gamma}$.

Thus, we obtain the dual objective function:

$$
\begin{aligned}
&\inf \alpha + \mu_1\beta_1 + \mu_2\beta_2 + \dots + \mu_n\beta_n + (\Sigma_{11}+\mu_1\mu_1)\Gamma_{11} + (\Sigma_{12}+\mu_1\mu_2)\Gamma_{21} + \dots = \\
&\inf \alpha + \boldsymbol{\mu}^T\boldsymbol{\beta} + (\boldsymbol{\Sigma}+\boldsymbol{\mu}\boldsymbol{\mu}^T)_{1i}\Gamma_{i1} + (\boldsymbol{\Sigma}+\boldsymbol{\mu}\boldsymbol{\mu}^T)_{2i}\Gamma_{i2} + \dots = \\
&\inf \alpha + \boldsymbol{\mu}^T\boldsymbol{\beta} + \langle \boldsymbol{\Sigma}+\boldsymbol{\mu}\boldsymbol{\mu}^T,\ \boldsymbol{\Gamma}\rangle
\end{aligned} \tag{3.15}
$$

and the dual constraints:

$$
\mathbf{A}^T\ [\alpha,\ \beta_1,\ \beta_2,\ \dots,\ \Gamma_{11},\ \Gamma_{21},\ \dots,\ \Gamma_{12},\ \dots]^T \geq \mathbf{c}
$$

For an arbitrary row i of $\mathbf{A}^T$ we can express the constraint:

$$\Delta\ [1,\ \xi_{i1},\ \xi_{i2},\ \dots,\ \xi_{i1}^2,\ \xi_{i1}\xi_{i2},\ \dots,\ \xi_{i2}\xi_{i1}, \dots][\alpha, \boldsymbol{\beta}^T,\ \Gamma_{11}, \Gamma_{21}, \dots, \Gamma_{12}, \dots]^T \geq \Delta\ 1_{cond(\boldsymbol{\xi}_i)}$$
$$\alpha + \boldsymbol{\xi}_i^T \boldsymbol{\beta} + \boldsymbol{\xi}_i^T \Gamma \boldsymbol{\xi}_i \geq 1_{cond(\boldsymbol{\xi}_i)}$$

Since we want this to be true for all values of i, we can reformulate this set of constraints as the following two constraints:

$$\begin{aligned} &\alpha + \boldsymbol{\xi}^T \boldsymbol{\beta} + \boldsymbol{\xi}^T \Gamma \boldsymbol{\xi} \geq 0 \ \ \forall \boldsymbol{\xi} \\ &\alpha + \boldsymbol{\xi}^T \boldsymbol{\beta} + \boldsymbol{\xi}^T \Gamma \boldsymbol{\xi} \geq 1 \ \ \forall \boldsymbol{\xi} : \gamma \leq -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}} \boldsymbol{\xi} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_k} \boldsymbol{\xi}\} \end{aligned} \tag{3.16}$$

Finally, using equations (3.15) and (3.16) we obtain the dual program as written in equation (3.13). This completes the proof.

The first constraint of our new program from equation (3.13), "$\alpha + \boldsymbol{\beta}^T \boldsymbol{\xi} + \boldsymbol{\xi}^T \boldsymbol{\Gamma} \boldsymbol{\xi} \geq 0 \ \ \forall \boldsymbol{\xi} \in \mathbb{R}^n$", can be reformulated using the definition of positive semi-definite matrix as (see Proof 1 from Appendix A):

$$\begin{bmatrix} \boldsymbol{\Gamma} & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta} & \alpha \end{bmatrix} \succcurlyeq 0 \tag{3.17}$$

All these lead to the following formulation:

$$\begin{aligned} \inf \ & \alpha + \boldsymbol{\mu}^T \boldsymbol{\beta} + \langle \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T, \boldsymbol{\Gamma} \rangle \\ s.\ t. \ & \alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \boldsymbol{\Gamma} \in \mathbb{R}^{n*n} \\ & \begin{bmatrix} \boldsymbol{\Gamma} & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta} & \alpha \end{bmatrix} \succcurlyeq 0 \\ & \alpha + \boldsymbol{\beta}^T \boldsymbol{\xi} + \boldsymbol{\xi}^T \boldsymbol{\Gamma} \boldsymbol{\xi} - 1 \geq 0 \ \ \forall \boldsymbol{\xi} \in \mathbb{R}^n : \gamma + (\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}} \boldsymbol{\xi} + (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_k} \boldsymbol{\xi}\} \leq 0 \end{aligned} \tag{3.18}$$

On the above problem, we can apply Farkas Theorem (Theorem 2.1, [17]), which leads to the addition of one more variable, $\lambda \geq 0$, but simplifies the final constraint. Hence, we obtain the equivalent minimisation problem:

$$
\begin{aligned}
&\inf\ \alpha + \boldsymbol{\mu}^T\boldsymbol{\beta} + \langle \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T, \boldsymbol{\Gamma}\rangle \\
&s.\ t.\ \ \alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \boldsymbol{\Gamma} \in \mathbb{R}^{n*n},\ \lambda \geq 0 \\
&\qquad \begin{bmatrix} \boldsymbol{\Gamma} & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta} & \alpha \end{bmatrix} \succcurlyeq 0 \\
&\qquad \alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{\Gamma}\boldsymbol{\xi} - 1 + \lambda\gamma + \lambda(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\boldsymbol{\xi} + \lambda(\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k} + \mathbf{B_k}\boldsymbol{\xi}\} \geq 0\ \ \forall \boldsymbol{\xi} \in \mathbb{R}^n
\end{aligned}
\tag{3.19}
$$

Since the final constraint is a 'greater or equal to' constraint and it must be true for all $\boldsymbol{\xi}$, it is equivalent to requiring the left-hand side of the expression to be greater or equal to zero for its minimum value over $\boldsymbol{\xi}$ only. Also, since in our case we need $\lambda > 0$, we can make the following substitutions without affecting the conditions: define new $\alpha = \frac{\alpha}{\lambda}$, $\boldsymbol{\beta} = \frac{\boldsymbol{\beta}}{\lambda}$, $\boldsymbol{\Gamma} = \frac{\boldsymbol{\Gamma}}{\lambda}$. This is possible since we have no constraints on the values of α, $\boldsymbol{\beta}$ and $\boldsymbol{\Gamma}$. So, we obtain the following problem:

$$
\begin{aligned}
&\inf\ \lambda(\alpha + \boldsymbol{\mu}^T\boldsymbol{\beta} + \langle \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T, \boldsymbol{\Gamma}\rangle) \\
&s.\ t.\ \ \alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \boldsymbol{\Gamma} \in \mathbb{R}^{n*n},\ \lambda > 0 \\
&\qquad \begin{bmatrix} \boldsymbol{\Gamma} & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta} & \alpha \end{bmatrix} \succcurlyeq 0 \\
&\qquad \lambda(\alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{\Gamma}\boldsymbol{\xi}) - 1 + \lambda\gamma + \lambda(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\boldsymbol{\xi} + \lambda(\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k} + \mathbf{B_k}\boldsymbol{\xi}\} \geq 0\ \ \forall \boldsymbol{\xi} \in \mathbb{R}^n
\end{aligned}
\tag{3.20}
$$

Additionally, we define $\tau = \frac{1}{\lambda}$ and replace this back into the initial WPVaR formulation, defined in equation (3.10). Hence, we obtain:

$$
\begin{aligned}
WPVaR_\epsilon(\boldsymbol{\omega}) = \min\{\gamma : \inf(&\alpha + \boldsymbol{\mu}^T\boldsymbol{\beta} + \langle \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T, \boldsymbol{\Gamma}\rangle) \leq \tau\epsilon\} \\
s.\ t.\ \ &\alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \boldsymbol{\Gamma} \in \mathbb{R}^{n*n},\ \tau \in \mathbb{R},\ \tau \geq 0 \\
&\begin{bmatrix} \boldsymbol{\Gamma} & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta} & \alpha \end{bmatrix} \succcurlyeq 0 \\
&\min_{\boldsymbol{\xi}}\ \alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{\Gamma}\boldsymbol{\xi} - \tau + \gamma + (\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\boldsymbol{\xi} + (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k\{\mathbf{a_k} + \mathbf{B_k}\boldsymbol{\xi}\} \geq 0
\end{aligned}
\tag{3.21}
$$

However, this formulation does not constitute a semidefinite program because of the inner piecewise maximisation function ($\max_k\{\mathbf{a_k} + \mathbf{B_k}\boldsymbol{\xi}\}$). So, we will reformulate this in order to obtain a SDP program.

3.2.2 Reformulation of Piecewise Maximisation Problem

The challenge is to reformulate the above maximisation expression in order to remove the optimisation constraint. To achieve this, we will use the following result in an appropriate manner.

In general, we can replace a maximisation problem of the type:

$$\max_k \ v_k$$

where k is a natural number with values from 1 to N, with the following maximisation problem:

$$\max_{\lambda\in\Delta} \sum_{i=1}^{N} \lambda_i f_i(x)$$

$$s.\ t.\ \ \lambda_i \in [0,1],\ \sum_{i=1}^{N} \lambda_i = 1$$

The reasoning behind this transformation is that if we want to maximise the sum, our choice of λ's will be as follows: $\lambda_j = 1$ iff j is the index of the maximum value v_i over all i and $\lambda_i = 0\ \forall i \neq j$. Hence, the two programs are equivalent.

However, in our case we cannot apply this result in a straightforward manner. The reason is that out max function is component-wise maximisation function. Hence, we apply this result on every component of our array.

For this we define a $\boldsymbol{\lambda}^i \in \Delta$ for every element i of the array $\max_k\{\mathbf{a}_k + \mathbf{B}_k\boldsymbol{\xi}\}$. So each component of the array will now be defined as:

$$\max_{\boldsymbol{\lambda}^i\in\Delta} \sum_{k=1}^{G} \lambda_k^i (a_{ki} + \mathbf{b}_{ki}^T\boldsymbol{\xi}) \tag{3.22}$$

where a_{ki} is the i-th element of $\mathbf{a}_k$, $\mathbf{b}_{ki}^T$ is the i-th line of $\mathbf{B}_k$ and G is the number of piecewise linear functions used to approximate the options return. Please note that even if the options aren't all expressed by the same number of linear functions, we can choose G as the largest number of functions used and add the function $f = 0$ as many times as necessary to obtain G functions for every option's return.

To replace this in our formula, we first need to express the product $(\boldsymbol{\omega^\eta})^\mathbf{T} \max_k\{\mathbf{a_k} + \mathbf{B_k}\boldsymbol{\xi}\}$ as a sum:

$$\textstyle\sum_{i=1}^{m-n} \omega_i^\eta \max_k\{a_{ki} + \mathbf{b_{ki}}^T\boldsymbol{\xi}\}$$

Now, since our inner maximisation problem refers to the i-th component of the array, we can replace equation (3.22) into the above and obtain an equivalent expression for our initial product:

$$\sum_{i=1}^{m-n} \omega_i^\eta \max_{\boldsymbol{\lambda}^i \in \Delta} \sum_{k=1}^{G} \lambda_k^i (a_{ki} + \mathbf{b}_{ki}^T \boldsymbol{\xi}) \tag{3.23}$$

We can now replace this back into our WPVaR formulation from equation (3.21) and obtain the following expression. Additionally, we can pull the inner maximisation problem to the front of the condition since no other part of the expression depends on $\boldsymbol{\lambda}$ or k:

$$\begin{aligned} WPVaR_\epsilon(\boldsymbol{\omega}) \; &= \min \; \{\gamma : \inf(\alpha + \boldsymbol{\mu}^T\boldsymbol{\beta} + \langle \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T, \boldsymbol{\Gamma}\rangle) \leq \tau\epsilon\} \\ s.\ t. \;\; & \alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \boldsymbol{\Gamma} \in \mathbb{R}^{n*n},\ \tau \in \mathbb{R},\ \tau \geq 0 \\ & \boldsymbol{\lambda}^i \in \Delta \;\; \forall i = 1, \ldots, m-n \\ & \begin{bmatrix} \boldsymbol{\Gamma} & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta} & \alpha \end{bmatrix} \succcurlyeq 0 \\ & \min_{\boldsymbol{\xi}} \max_{\boldsymbol{\lambda}^i \in \Delta} \alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{\Gamma}\boldsymbol{\xi} - \tau + \gamma + \\ & \qquad\qquad (\boldsymbol{\omega^\xi})^\mathbf{T}\boldsymbol{\xi} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i (a_{ki} + \mathbf{b}_{ki}^T\boldsymbol{\xi}) \geq 0 \end{aligned} \tag{3.24}$$

We can now change the order of the minimisation and maximisation problems of the last constraint and thus obtain a max-min problem instead of a min-max one:

$$\max_{\boldsymbol{\lambda}^i \in \Delta} \min_{\boldsymbol{\xi}} \ \alpha + \boldsymbol{\beta}^T \boldsymbol{\xi} + \boldsymbol{\xi}^T \boldsymbol{\Gamma} \boldsymbol{\xi} - \tau + \gamma + (\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}} \boldsymbol{\xi} + \sum_{i=1}^{m-n} \omega_i^{\eta} \sum_{k=1}^{G} \lambda_k^i (a_{ki} + \mathbf{b}_{ki}^T \boldsymbol{\xi}) \geq 0$$

Since we require the maximum over $\boldsymbol{\lambda}^i$ values to be greater than or equal to zero, we can replace the outer 'max' by an $\exists$ constraint over $\boldsymbol{\lambda}^i$ (it is enough for one value to be greater than or equal to zero, than the maximum will also oblige this condition. Then, we are left with the condition which requires the minimum over $\boldsymbol{\xi}$ to be greater than or equal to zero. We can replace this 'min' condition by an $\forall$ constraint over $\boldsymbol{\xi}$. Again, this is possible since if all the values are greater than or equal to zero, so will be the minimum of these values. Hence, we obtain the constraint:

$$\begin{aligned} &\exists \boldsymbol{\lambda}^i \in \Delta, \quad \forall i = 1, \ldots, m-n \\ &\forall \boldsymbol{\xi} : \quad \alpha + \boldsymbol{\beta}^T \boldsymbol{\xi} + \boldsymbol{\xi}^T \boldsymbol{\Gamma} \boldsymbol{\xi} - \tau + \gamma + (\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}} \boldsymbol{\xi} + \sum_{i=1}^{m-n} \omega_i^{\eta} \sum_{k=1}^{G} \lambda_k^i (a_{ki} + \mathbf{b}_{ki}^T \boldsymbol{\xi}) \geq 0 \end{aligned}$$

Now, we can again use the definition of positive semidefinite matrix to reformulate the above constraint as a positive semidefinite constraint (see Proof 1 in Appendix A):

$$\begin{aligned} &\exists \boldsymbol{\lambda}^i \in \Delta, \quad \forall i = 1, \ldots, m-n \\ &\begin{bmatrix} \Gamma & \frac{1}{2}(\boldsymbol{\beta} + \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^{\eta} \sum_{k=1}^{G} \lambda_k^i \mathbf{b_{ki}}) \\ \frac{1}{2}(\boldsymbol{\beta} + \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^{\eta} \sum_{k=1}^{G} \lambda_k^i \mathbf{b_{ki}})^T & \alpha - \tau + \gamma + \sum_{i=1}^{m-n} \omega_i^{\eta} \sum_{k=1}^{G} \lambda_k^i a_{ki} \end{bmatrix} \succcurlyeq 0 \end{aligned}$$

Replacing this into our WPVaR equation (3.24), we obtain the following SDP formulation for our WPVaR:

Theorem 3.1
$WPVaR_\epsilon = \inf \gamma,$

$$
\begin{aligned}
s.\ t.\ & \mathbf{\Gamma} \in \mathbb{S}^{n*n},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \alpha \in \mathbb{R},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R} \\
& \forall i = 1, \ldots, (m-n):\ \ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
& \tau \geq 0 \\
& \alpha + \boldsymbol{\beta}^T \boldsymbol{\mu} + \langle \mathbf{\Gamma}, \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T \rangle \leq \tau\epsilon \\
& \begin{bmatrix} \Gamma & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta}^T & \alpha \end{bmatrix} \succcurlyeq 0 \\
& \begin{bmatrix} \Gamma & \frac{1}{2}(\boldsymbol{\beta} + \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i \mathbf{b_{ki}}) \\ \frac{1}{2}(\boldsymbol{\beta} + \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i \mathbf{b_{ki}})^T & \alpha - \tau + \gamma + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i a_{ki} \end{bmatrix} \succcurlyeq 0
\end{aligned}
\tag{3.25}
$$

where $\boldsymbol{\mu}$ is the mean returns vector, $\mathbf{\Sigma}$ is the covariance matrix of $\tilde{\boldsymbol{\xi}}$, G is the number of linear functions used to approximate the options' value payoff functions and the vectors $\mathbf{a}_k$ are the same constants as in (3.10), with the modification described above (where we included the "$-\mathbf{e}$" into the $\mathbf{a}_k$). The vector $\mathbf{b_{ki}^T}$ is the i-th line of the matrix $\mathbf{B_k}$ as defined in (3.10).

This formulation of our WPVaR leads to a straightforward formulation of the portfolio optimisation problem. This is shown in equation (3.26) below. Please note short selling is not allowed due to the constraint that weights are positive.

$$
\begin{aligned}
&\min \gamma, \\
&s.\,t.\ \mathbf{\Gamma} \in \mathbb{S}^{n*n},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \alpha \in \mathbb{R},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R},\ \boldsymbol{\omega} \in \mathbb{R}^m \\
&\qquad \forall i = 1, \ldots, (m-n):\ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
&\qquad \tau \geq 0,\ \ \boldsymbol{\omega} \geq 0,\ \ \boldsymbol{\omega}^T \mathbf{e} = 1 \\
&\qquad \boldsymbol{\omega}^T \tilde{\mathbf{r}} \geq r_{min} \\
&\qquad \alpha + \boldsymbol{\beta}^T \boldsymbol{\mu} + \langle \mathbf{\Gamma}, \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T \rangle \leq \tau\epsilon \\
&\qquad \begin{bmatrix} \Gamma & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta}^T & \alpha \end{bmatrix} \succcurlyeq 0 \\
&\qquad \begin{bmatrix} \Gamma & \frac{1}{2}(\boldsymbol{\beta} + \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}}) \\ \frac{1}{2}(\boldsymbol{\beta} + \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}})^T & \alpha - \tau + \gamma + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i a_{ki} \end{bmatrix} \succcurlyeq 0
\end{aligned}
\tag{3.26}
$$

We can further reformulate the WPVaR expression from equation (3.25), to obtain a SOCP program. To this purpose, we let $\mathbf{M} = \begin{bmatrix} \Gamma & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta}^T & \alpha \end{bmatrix}$ and obtain:

$$
\begin{aligned}
&WPVaR_\epsilon = \inf \gamma, \\
&\qquad s.\,t.\ \mathbf{\Gamma} \in \mathbb{R}^{n*n},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \alpha \in \mathbb{R},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R},\ \mathbf{M} \in \mathbb{S}^{n+1} \\
&\qquad\quad \forall i = 1, \ldots, (m-n):\ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
&\qquad\quad \tau \geq 0 \\
&\qquad\quad \alpha + \boldsymbol{\beta}^T \boldsymbol{\mu} + \langle \mathbf{\Gamma}, \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T \rangle \leq \tau\epsilon \\
&\qquad\quad \mathbf{M} \succcurlyeq 0 \\
&\qquad\quad \mathbf{M} + \begin{bmatrix} \mathbf{0} & \frac{1}{2}(\boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}}) \\ \frac{1}{2}(\boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}})^T & -\tau + \gamma + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i a_{ki} \end{bmatrix} \succcurlyeq 0
\end{aligned}
\tag{3.27}
$$

We know the second-order moment matrix $\mathbf{\Omega} = \begin{bmatrix} \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T & \boldsymbol{\mu} \\ \boldsymbol{\mu}^T & 1 \end{bmatrix}$, so we can easily prove the following equivalence (see Proof 2 of appendix A):

$$\langle \mathbf{\Omega}, \mathbf{M} \rangle = \alpha + \boldsymbol{\beta}^T \boldsymbol{\mu} + \langle \mathbf{\Gamma}, \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T \rangle \tag{3.28}$$

Replacing equivalence (3.28) into the WPVaR expression, we obtain:

$$\begin{aligned}
WPVaR_\epsilon = &\inf \gamma, \\
&s.\ t.\ \mathbf{M} \in \mathbb{S}^{n+1},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R} \\
&\quad \forall i = 1, \dots, (m-n):\ \ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
&\quad \tau \geq 0,\ \mathbf{M} \succcurlyeq 0 \\
&\quad \langle \mathbf{\Omega}, \mathbf{M} \rangle \leq \tau\epsilon \\
&\quad \mathbf{M} + \begin{bmatrix} \mathbf{0} & \frac{1}{2}(\boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}}) \\ \frac{1}{2}(\boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}})^T & -\tau + \gamma + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i a_{ki} \end{bmatrix} \succcurlyeq 0
\end{aligned} \tag{3.29}$$

Then, we multiply all four constraint by 2 and let new $\mathbf{M} = 2\mathbf{M}$ and new $\tau = 2\tau$ and obtain the following WPVaR expression:

$$\begin{aligned}
WPVaR_\epsilon = &\inf \gamma, \\
&s.\ t.\ \mathbf{M} \in \mathbb{S}^{n+1},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R} \\
&\quad \forall i = 1, \dots, (m-n):\ \ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
&\quad \tau \geq 0,\ \mathbf{M} \succcurlyeq 0 \\
&\quad \langle \mathbf{\Omega}, \mathbf{M} \rangle \leq \tau\epsilon \\
&\quad \mathbf{M} + \begin{bmatrix} \mathbf{0} & \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}} \\ (\boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}})^T & -\tau + 2(\gamma + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i a_{ki}) \end{bmatrix} \succcurlyeq 0
\end{aligned} \tag{3.30}$$

We can see that we can now apply Theorem 1 from [3] and obtain the following SOCP formulation of our WPVaR:

Theorem 3.2

$$WPVaR_\epsilon(\boldsymbol{\omega}) = \min\{ -\boldsymbol{\mu}^T(\boldsymbol{\omega}^\xi + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i \mathbf{b_{ki}}) - \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i a_{ki}$$
$$+ k(\epsilon)\|\boldsymbol{\Sigma}^{1/2}(\boldsymbol{\omega}^\xi + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i \mathbf{b_{ki}})\| \} \tag{3.31}$$

3.3 WPVaR Equivalence for Options at Expiry

A first step in validating our WPVaR is to verify that in the basic case where we only use options at expiry (hence their return function is expressed by the maximum of two linear functions - see figure 2.3), this formulation is equivalent to the formulation of WPVaR presented as Theorem 4.1 in [4].

To achieve this, we need to show our final SDP formulation from equation (3.25), when the options are at expiry, is equivalent to the following program:

$$\begin{aligned} WPVaR_\epsilon(\boldsymbol{\omega}) = \inf \quad & \gamma \\ s.\ t. \quad & \mathbf{M} \in \mathbb{S}^{n+1},\ \mathbf{y} \in \mathbb{R}^{m-n},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R} \\ & \langle \boldsymbol{\Omega},\ \mathbf{M} \rangle \le \tau\epsilon,\ \mathbf{M} \succcurlyeq 0,\ \tau \ge 0,\ \mathbf{0} \le \mathbf{y} \le \boldsymbol{\omega}^\eta \\ & \mathbf{M} + \begin{bmatrix} \mathbf{0} & \boldsymbol{\omega}^\xi + \mathbf{B}^T\mathbf{y} \\ (\boldsymbol{\omega}^\xi + \mathbf{B}^T\mathbf{y})^T & -\tau + 2(\gamma + \mathbf{y}^T\mathbf{a} - \mathbf{e}^T\boldsymbol{\omega}^\eta) \end{bmatrix} \succcurlyeq 0 \end{aligned}$$

where $\boldsymbol{\Omega}$ is the second-order moment matrix and $\mathbf{a}$ and $\mathbf{B}$ are the constants used to express the options' return functions.

We will prove this equivalence by using the knowledge that all options are at expiry. Hence we will have the following constants:

Call options

For a call option, the return function will be $\max\{-1,\ \frac{s-k}{c} + \frac{s}{c}\xi - 1\}$, where s is the stock price (of the underlying asset), k is the strike price of the option, c is the price of the call option and ξ is the mean return of the underlying stock. Hence, it will have the following constants which we use in our model:

- first function: $a_1 = -1$, $b_1 = 0$
- second function: $a_2 = \frac{s-k}{c} - 1$, $b_2 = \frac{s}{c}$

Put options

For a put option, the return function will be $\max\{-1,\ \frac{k-s}{p} - \frac{s}{p}\xi - 1\}$, where s, k and ξ are as above and p is the price of the put option. Hence, it will have the following constants which we use in our model:

- first function: $a_1 = -1,\ b_1 = 0$
- second function: $a_2 = \frac{k-s}{p} - 1,\ b_2 = -\frac{s}{c}$

From this, we can define our $\mathbf{a}_k$ and $\mathbf{B}_k$ constants for $k \in \{1,\ 2\}$. Below we define the values for some option i, with strike price k_i and price P_i, defined on some basic asset j with stock price s_j. a_{2i} thus represents the i-th element of $\mathbf{a}_2$ and $\mathbf{b}_{2i}^T$ represents the ith row of $\mathbf{B}_2$.

$$\mathbf{a_1} = \mathbf{0}_{m-n}$$

$$a_{2i} = |\frac{s_j - k_i}{P_i}|$$

$$\mathbf{B_1} = \mathbf{0}_{m-n,\ n}$$

$$\mathbf{b}_{2i}^T = [0, \dots, 0, b_{2i}, 0, \dots, 0]$$

where the value b_{2i} is on position j and has value as defined above, according to the type of the option (call or put).

We can now compute the two expressions from our WPVaR formula which depend on these constants. We consider a_{ki} to be the k-th "a" constant of the i-th option (all option will have only 2 sets of constants, as defined above, since $G = 2$). Also, we use the fact that $\boldsymbol{\lambda}^i \in \Delta$, hence $\sum_{k=1}^{G} \lambda_k^i = 1\ \forall i$.

$$\begin{aligned}\sum_{i=1}^{m-n}\omega_i^{\eta}\sum_{k=1}^{G}\lambda_k^i a_{ki} &= \sum_{i=1}^{m-n}\omega_i^{\eta}(\lambda_1^i a_{1i}+\lambda_2^i a_{2i})\\ &= \sum_{i=1}^{m-n}\omega_i^{\eta}(-\lambda_1^i+\lambda_2^i a_{2i}-\lambda_2^i) = \sum_{i=1}^{m-n}\omega_i^{\eta}(\lambda_2^i-1)\\ &= \sum_{i=1}^{m-n}\omega_i^{\eta}\lambda_2^i a_{2i}-\mathbf{e}^T\boldsymbol{\omega}^{\eta} \qquad (3.32)\end{aligned}$$

$$\begin{aligned}\sum_{i=1}^{m-n}\omega_i^{\eta}\sum_{k=1}^{G}\lambda_k^i \mathbf{b}_{ki} &= \sum_{i=1}^{m-n}\omega_i^{\eta}(\lambda_1^i \mathbf{b}_{1i}+\lambda_2^i \mathbf{b}_{2i})\\ &= \sum_{i=1}^{m-n}\omega_i^{\eta}\lambda_2^i \mathbf{b}_{2i} \qquad (3.33)\end{aligned}$$

As a final step, we let $y_i = \lambda_2^i\omega_i^{\eta}\ \ \forall i = 1,\dots,m-n$. Since $\lambda_2^i \in [0,\ 1]$, then $y_i \in [0,\ \omega_i^{\eta}]$. If we define the array $\mathbf{y} \in \mathbb{R}^{m-n}$ with elements y_i and replace these new variables into equations (3.32) and (3.33) above, we obtain:

$$\sum_{i=1}^{m-n}\omega_i^{\eta}\sum_{k=1}^{G}\lambda_k^i a_{ki} = \sum_{i=1}^{m-n} y_i a_{2i}-\mathbf{e}^T\boldsymbol{\omega}^{\eta} = \mathbf{y}^T\mathbf{a}_2-\mathbf{e}^T\boldsymbol{\omega}^{\eta} \qquad (3.34)$$

$$\sum_{i=1}^{m-n}\omega_i^{\eta}\sum_{k=1}^{G}\lambda_k^i \mathbf{b}_{ki} = \sum_{i=1}^{m-n} y_i \mathbf{b}_{2i} = \mathbf{B}_2^T\mathbf{y} \qquad (3.35)$$

Replacing the above into the WPVaR formulation from equation (3.29), we obtain:

$$
\begin{aligned}
WPVaR_\epsilon = &\inf \gamma, \\
&s.\ t.\ \mathbf{M} \in \mathbb{S}^{n+1},\ \mathbf{y} \in \mathbb{R}^{m-n},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R} \\
&\quad \forall i = 1, \ldots, (m-n):\ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
&\quad \tau \geq 0,\ \mathbf{M} \succcurlyeq 0,\ \mathbf{0} \leq \mathbf{y} \leq \boldsymbol{\omega}^\eta \\
&\quad \langle \boldsymbol{\Omega}, \mathbf{M} \rangle \leq \tau\epsilon \\
&\quad \mathbf{M} + \begin{bmatrix} \mathbf{0} & \frac{1}{2}(\boldsymbol{\omega}^{\boldsymbol{\xi}} + \mathbf{B}_2^T\mathbf{y}) \\ \frac{1}{2}(\boldsymbol{\omega}^{\boldsymbol{\xi}} + \mathbf{B}_2^T\mathbf{y})^T & -\tau + \gamma + \mathbf{y}^T\mathbf{a}_2 - \mathbf{e}^T\boldsymbol{\omega}^\eta \end{bmatrix} \succcurlyeq 0
\end{aligned} \tag{3.36}
$$

A first observation is that the values of our $\mathbf{a}_2$ and $\mathbf{B_2}$ are equal to the values of $\mathbf{a}$ and $\mathbf{B}$ as defined in the paper, respectively. Second, we note that we can define a new matrix $M = 2M$, which will preserve all the properties and will lead to the same result as in [4].

To show this, we use that $\mathbf{2A} \succcurlyeq 0 \Longleftrightarrow \mathbf{A} \succcurlyeq 0$, which is trivially true for any matrix $\mathbf{A}$. To use this, we will multiply by 2 the positive semi-definite constraint on $\mathbf{M}$ ($\mathbf{M} \succcurlyeq 0$) and on the last constraint. Hence, we obtain:

$$
\begin{aligned}
\mathbf{2M} &\succcurlyeq 0 \\
\mathbf{2M} + \begin{bmatrix} \mathbf{0} & (\boldsymbol{\omega}^{\boldsymbol{\xi}} + \mathbf{B}_2^T\mathbf{y}) \\ (\boldsymbol{\omega}^{\boldsymbol{\xi}} + \mathbf{B}_2^T\mathbf{y})^T & -2\tau + 2(\gamma + \mathbf{y}^T\mathbf{a}_2 - \mathbf{e}^T\boldsymbol{\omega}^\eta) \end{bmatrix} &\succcurlyeq 0
\end{aligned}
$$

We will also use the fact that $\langle \boldsymbol{\Omega}, \mathbf{2M} \rangle = 2\langle \boldsymbol{\Omega}, \mathbf{M} \rangle$. This provides us the inequality:

$$
\langle \boldsymbol{\Omega}, \mathbf{2M} \rangle \leq 2\tau\epsilon
$$

Finally, we define a new $\tau = 2\tau$ and a new matrix $\mathbf{M} = \mathbf{2M}$. We can do this since $\tau \in \mathbb{R}$, $\tau \geq 0$ and $\mathbf{M} \in \mathbb{S}^{n+1}$ and the operation preserves these properties.

Performing the above replacements yields the following program, which is identical to the one defined in Theorem 4.1 in [4].

$$
\begin{aligned}
WPVaR_\epsilon(\boldsymbol{\omega}) = \inf \quad & \gamma \\
s.\ t. \quad & \mathbf{M} \in \mathbb{S}^{n+1},\ \mathbf{y} \in \mathbb{R}^{m-n},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R} \\
& \langle \boldsymbol{\Omega}, \quad vectM \rangle \leq \tau\epsilon,\ \mathbf{M} \succcurlyeq 0,\ \tau \geq 0,\ \mathbf{0} \leq \mathbf{y} \leq \boldsymbol{\omega}^\eta \\
& \mathbf{M} + \begin{bmatrix} \mathbf{0} & \boldsymbol{\omega}^\xi + \mathbf{B}^T\mathbf{y} \\ (\boldsymbol{\omega}^\xi + \mathbf{B}^T\mathbf{y})^T & -\tau + 2(\gamma + \mathbf{y}^T\mathbf{a} - \mathbf{e}^T\boldsymbol{\omega}^\eta) \end{bmatrix} \succcurlyeq 0
\end{aligned}
$$

3.4 Summary

The WPVaR model we formulated represents a generalisation of the WPVaR presented in [4]. Its novelty consists of the fact that we have incorporated options with expiration beyond the investment horizon, in addition to options at expiry. This has been achieved by approximating the options return function by an arbitrary number of linear functions.

Finally, we have shown the WPVaR model can be reformulated as a tractable SDP or as a tractable SOCP. In addition, we provided a formal mathematical proof that our model is identical to the one described in [4], in the basic case where only options at expiry are considered.

Chapter 4

Robust Models

Our SDP formulation of WPVaR from theorem 3.1 is based on the assumption of knowledge of the mean return vector $\boldsymbol{\mu}$ and of the covariance matrix $\boldsymbol{\Sigma}$. However, this is not a realistic assumption. In practice investors do not know, nor can they exactly infer, these two values. Moreover, using only the first and second moments information does not provide a way of capturing asymmetry in the distribution — distributional skewness. This is undesirable in practice since in reality the distribution of financial assets returns is often skewed.

Hence, we define and prove two new theorems, by extending the WPVaR model using two strategies: first, we will assume a box-type uncertainty in both the mean and covariance matrix; second, we will use partitioned statistics information which captures asymmetry information in the distribution (a very common property of real assets returns distributions).

4.1 Box-type Uncertainty

In the WPVaR model developed so far, we have assumed the mean and the covariance matrix of the asset returns is known exactly. In this section, we specify box-type uncertainty sets for these as well, So, we assume the moments lie in a box region, specified by its lower and upper bounds.

The box-type uncertainty set for specifically the second-order moment matrix $\boldsymbol{\Omega}$ can be defined as:

$$\mathcal{O} = \{\boldsymbol{\Omega} \in \mathbb{S}^{n+1} \ : \ \boldsymbol{\Omega} \succcurlyeq 0, \ \underline{\boldsymbol{\Omega}} \leq \boldsymbol{\Omega} \leq \overline{\boldsymbol{\Omega}}\} \tag{4.1}$$

where $\underline{\boldsymbol{\Omega}}$ and $\overline{\boldsymbol{\Omega}}$ are the lower and upper bound respectively and $\underline{\boldsymbol{\Omega}}_{(n+1)(n+1)} = \overline{\boldsymbol{\Omega}}_{(n+1)(n+1)} = 1$.

To apply this constraint, we will use the WPVaR formulation from equation (3.30). We let $z = \gamma - \frac{\tau}{2}$ and obtain the following WPVaR expression:

$$
\begin{aligned}
WPVaR_\epsilon = &\inf z + \frac{\tau}{2}, \\
& s.\ t.\ \mathbf{M} \in \mathbb{S}^{n+1},\ \tau \in \mathbb{R},\ \gamma \in \mathbb{R},\ \ \forall i = 1, \ldots, (m-n):\ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
& \quad \tau \geq 0,\ \mathbf{M} \succcurlyeq 0,\ \langle \boldsymbol{\Omega}, \mathbf{M} \rangle \leq \tau\epsilon \\
& \quad \mathbf{M} + \begin{bmatrix} \mathbf{0} & \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}} \\ (\boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}})^T & 2(z + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i a_{ki}) \end{bmatrix} \succcurlyeq 0
\end{aligned}
$$

We now notice that at optimality $\tau = \frac{1}{\epsilon}\langle \boldsymbol{\Omega},\ \mathbf{M} \rangle$, so we can further simplify the above formulation:

$$
\begin{aligned}
WPVaR_\epsilon = &\inf z + \frac{1}{2\epsilon}\langle \boldsymbol{\Omega}, \mathbf{M} \rangle, \\
& s.\ t.\ \mathbf{M} \in \mathbb{S}^{n+1},\ \gamma \in \mathbb{R},\ \tau \in \mathbb{R},\ \ \forall i = 1, \ldots, (m-n):\ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
& \quad \mathbf{M} \succcurlyeq 0,\ \tau \geq 0 \\
& \quad \mathbf{M} + \begin{bmatrix} \mathbf{0} & \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}} \\ (\boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}})^T & 2(z + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i a_{ki}) \end{bmatrix} \succcurlyeq 0
\end{aligned}
$$

Finally, we use the box-type uncertainty set defined in equation (4.1) to define the robust risk measure, which is a worst-case value of our WPVaR:

$$
\begin{aligned}
\sup_{\boldsymbol{\Omega} \in \mathcal{O}} WPVaR_\epsilon = &\inf z + \frac{1}{2\epsilon} \sup_{\boldsymbol{\Omega} \in \mathcal{O}} \langle \boldsymbol{\Omega}, \mathbf{M} \rangle, \\
& s.\ t.\ \mathbf{M} \in \mathbb{S}^{n+1},\ \gamma \in \mathbb{R},\ \tau \in \mathbb{R},\ \ \forall i = 1, \ldots, (m-n):\ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
& \quad \mathbf{M} \succcurlyeq 0,\ \tau \geq 0 \\
& \quad \mathbf{M} + \begin{bmatrix} \mathbf{0} & \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}} \\ (\boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}})^T & 2(z + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i a_{ki}) \end{bmatrix} \succcurlyeq 0
\end{aligned}
$$

Using the definition of the uncertainty set, we can apply duality theory (see section 2) and results from [10] to obtain the following equivalence:

$$\sup_{\Omega \in \mathcal{O}} \langle \mathbf{\Omega}, \mathbf{M} \rangle = \inf \quad \langle \overline{\mathbf{\Omega}}, \mathbf{\Omega}^+ \rangle - \langle \underline{\mathbf{\Omega}}, \mathbf{\Omega}^- \rangle$$
$$s.\ t. \quad \mathbf{\Omega}^+,\ \mathbf{\Omega}^- \geq 0$$
$$\mathbf{\Omega}^+ - \mathbf{\Omega}^- \succcurlyeq \mathbf{M}$$

We can replace this into our worst-case WPVaR formulation above, pull out the minimisation problem and leave only one minimisation over all variables. Hence, we can define our robust WPVaR risk measure as the optimal value of the following tractable SDP:

Theorem 4.1

$$\begin{aligned}
\sup_{\Omega \in \mathcal{O}} WPVaR_\epsilon = \inf z + \frac{1}{2\epsilon}(\langle \overline{\mathbf{\Omega}}, \mathbf{\Omega}^+ \rangle - \langle \underline{\mathbf{\Omega}}, \mathbf{\Omega}^- \rangle) \\
s.\ t.\ \mathbf{M} \in \mathbb{S}^{n+1},\ \mathbf{\Omega}^+, \mathbf{\Omega}^- \in \mathbb{R}^{(n+1)*(n+1)},\ \gamma \in \mathbb{R},\ \tau \in \mathbb{R} \\
\forall i = 1, \dots, (m-n):\ \boldsymbol{\lambda}^{\mathbf{i}} \in \mathbb{R}^G, \boldsymbol{\lambda}^{\mathbf{i}} \in \Delta \\
\mathbf{M} \succcurlyeq 0,\quad \mathbf{\Omega}^+, \mathbf{\Omega}^- \geq 0,\ \tau \geq 0 \\
\mathbf{\Omega}^+ - \mathbf{\Omega}^- \succcurlyeq \mathbf{M} \\
\mathbf{M} + \begin{bmatrix} \mathbf{0} & \boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}} \\ (\boldsymbol{\omega}^{\boldsymbol{\xi}} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i \mathbf{b_{ki}})^T & 2(z + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^G \lambda_k^i a_{ki}) \end{bmatrix} \succcurlyeq 0
\end{aligned} \tag{4.2}$$

4.2 Partitioned Statistics Information

Since our initial WPVaR model defined in theorem 3.1 only uses first and second order information, it is unable to represent distributional skewness. Since this is a common characteristic of financial assets returns distribution, we will use the partitioned statistics approach (as described in [10]) to model distributional asymmetry in the asset returns.

As defined in section 3, the portfolio loss is represented by (from equation (3.8), including the $-\mathbf{e}$ inside the a_k as described later in section 3):

$$\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \boldsymbol{\xi}) = -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}} \tilde{\boldsymbol{\xi}} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_k} \tilde{\boldsymbol{\xi}}\}$$

We will follow three main steps in applying the partitioned statistics approach: we will formulate upper bounds on the VaR of the portfolio based on two different distributional assumptions; then we will partition the random variable, the vector of *basic* asset returns $\boldsymbol{\xi}$, into its positive and negative parts; finally we will define a partitioned statistics upper bound on VaR.

4.2.1 VaR Upper Bounds

Our first step is to define two upper bounds on VaR, based on two different distributional assumptions. First, we assume given mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ and use $\mathcal{P}_1$ to represent the set of all probability distributions of the asset returns $\boldsymbol{\xi}$ with such properties. Second, we assume given mean vector $\boldsymbol{\mu}$ and polyhedral support $\{\boldsymbol{\xi} \in \mathbb{R}^n : \mathbf{W}\boldsymbol{\xi} \geq \mathbf{h}\}$ and represent the set of probability distributions which satisfy these conditions by $\mathcal{P}_2$. We will use the notation:

$$\pi_i(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) = \sup_{\mathbb{P} \in \mathcal{P}_i} \mathbb{P}\text{-}VaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}})) \quad for\ i = 1, 2$$

From theorem 3.2 we can see that:

$$\begin{aligned}\pi_1(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) = \min\{ -\boldsymbol{\mu}^T(\boldsymbol{\omega}^\xi + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i \mathbf{b_{ki}}) - \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i a_{ki} \\ + k(\epsilon) \| \boldsymbol{\Sigma}^{1/2}(\boldsymbol{\omega}^\xi + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i \mathbf{b_{ki}}) \| \ \}\end{aligned} \tag{4.3}$$

We will now derive a LP-representation of the WPVaR under $\mathcal{P}_2$. To achieve this we will return to the initial definition of WPVaR given in section 2, equation (2.4). Hence we want first to dualize the problem:

$$\sup_{\mathbb{P} \in \mathcal{P}} \mathbb{P}\{\gamma \leq -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}} \tilde{\boldsymbol{\xi}} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_k} \tilde{\boldsymbol{\xi}}\}\}$$

We will as before rewrite this as an optimisation problem with integral constraints. However, since we now have no information on the covariance matrix, we obtain the following maximisation problem:

$$
\begin{aligned}
\sup_{\boldsymbol{\mu}\geq 0} \quad & \int_{\mathbb{R}^n} 1_{\gamma \leq -(\boldsymbol{\omega^\xi})^\mathbf{T}\boldsymbol{\xi} - (\boldsymbol{\omega^\eta})^\mathbf{T} \max_k\{\mathbf{a_k}+\mathbf{B_k}\boldsymbol{\xi}\}} \ \boldsymbol{\mu} \ d\boldsymbol{\xi} \\
s.\ t. \quad & \int \boldsymbol{\mu} \ d\boldsymbol{\xi} = 1 \\
& \int \boldsymbol{\xi}\boldsymbol{\mu} \ d\boldsymbol{\xi} = \boldsymbol{\mu}
\end{aligned}
$$

Following similar steps as in section 3.2.1, we obtain the following dual program:

$$
\begin{aligned}
\inf \ & \alpha + \boldsymbol{\mu}^T\boldsymbol{\beta} \\
s.\ t. \ & \alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \tau \in \mathbb{R},\ \tau \geq 0 \\
& \alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} \geq 0 \ \ \forall \boldsymbol{\xi} : \mathbf{W}\boldsymbol{\xi} \geq \mathbf{h} \\
& \alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} - \tau + \gamma + (\boldsymbol{\omega^\xi})^\mathbf{T}\boldsymbol{\xi} + (\boldsymbol{\omega^\eta})^\mathbf{T} \max_k\{\mathbf{a_k} + \mathbf{B_k}\boldsymbol{\xi}\} \geq 0 \ \ \forall \boldsymbol{\xi} : \mathbf{W}\boldsymbol{\xi} \geq \mathbf{h}
\end{aligned}
$$

We can eliminate the inner piecewise maximisation problem by applying a similar method as in section 3.2.2 and thus obtain the following expression for our second upper bound:

$$
\begin{aligned}
\pi_2(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) = \min \gamma & \\
s.\ t. \ & \alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \tau \in \mathbb{R},\ \tau \geq 0 \\
& \inf \ \alpha + \boldsymbol{\mu}^T\boldsymbol{\beta} \leq \tau\epsilon \\
& \alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} \geq 0 \ \ \forall \boldsymbol{\xi} : \mathbf{W}\boldsymbol{\xi} \geq \mathbf{h} \\
& \alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} - \tau + \gamma + (\boldsymbol{\omega^\xi})^\mathbf{T}\boldsymbol{\xi} + \sum_{i=1}^{m-n} \omega_i^\eta \sum_{k=1}^{G} \lambda_k^i (a_{ki} + \mathbf{b}_{ki}^T\boldsymbol{\xi}) \geq 0 \ \ \forall \boldsymbol{\xi} : \mathbf{W}\boldsymbol{\xi} \geq \mathbf{h}
\end{aligned}
$$

We now have two constraints of the form $f(\boldsymbol{\xi}) \geq 0 \ \forall \boldsymbol{\xi} : \mathbf{W}\boldsymbol{\xi} \geq \mathbf{h}$. Since we want all values of the function to be greater than or equal to zero for certain values of $\boldsymbol{\xi}$, we can express these constraints by a minimisation problem:

$$
f(\boldsymbol{\xi}) \geq 0 \ \forall \boldsymbol{\xi} : \mathbf{W}\boldsymbol{\xi} \geq \mathbf{h} \iff \min f(\boldsymbol{\xi}) \geq 0 \ s.\ t.\ \mathbf{W}\boldsymbol{\xi} \geq \mathbf{h}
$$

We apply this to both constraints and we dualize the obtained minimisation problems. We will then obtain maximisation problems of the form $\max \ g(\mathbf{x}) \geq 0 \ s.\ t.\ \mathbf{x} : \mathit{cond}(\mathbf{x})$, where $\mathbf{x}$ is the newly introduced variable and $cond(\mathbf{x})$ is the set of constraints of the dual program. These problems can in turn be expressed by an "exists" constraint:

$$\max\ g(\mathbf{x}) \geq 0\ s.\ t.\ \mathbf{x} : cond(\mathbf{x}) \iff \exists \mathbf{x} : cond(\mathbf{x}) \quad g(\mathbf{x}) \geq 0$$

We apply both these operations on our two final constraints and replace the result into our second upper bound problem from above. Then, we replace γ by its lower bound (derived from one of the dual constraints) and obtain the following LP-formulation:

$$\begin{aligned}
\pi_2(\boldsymbol{\omega}^{\xi}, \boldsymbol{\omega}^{\eta}) \ &= \ \min\ \tau - \alpha - \sum_{i=1}^{m-n} \omega_i^{\eta} \sum_{k=1}^{G} \lambda_k^i a_{ki} - \mathbf{H}^T \mathbf{y} \\
& s.\ t.\ \alpha \in \mathbb{R},\ \boldsymbol{\beta} \in \mathbb{R}^n,\ \tau \in \mathbb{R},\ \tau \geq 0,\ \mathbf{y} \geq 0, \mathbf{x} \geq 0 \\
& \quad \alpha + \boldsymbol{\beta}^T \boldsymbol{\mu} \leq \tau\epsilon \\
& \quad \alpha + \mathbf{h}^T \mathbf{x} \geq 0 \\
& \quad \mathbf{W}^T \mathbf{x} - \boldsymbol{\beta} = 0 \\
& \quad \mathbf{W}^T \mathbf{y} - \boldsymbol{\beta} - \sum_{i=1}^{m-n} \omega_i^{\eta} \sum_{k=1}^{G} \lambda_k^i \mathbf{b}_{ki} = \boldsymbol{\omega}^{\xi}
\end{aligned} \tag{4.4}$$

We will now apply the partitioned statistics approach and define a new upper bound on VaR.

4.2.2 Random Variables Partition

To apply the partitioned statistics approach we will partition the random variables — the *basic* asset returns in our case — into their positive and negative parts and compute the mean and covariance matrix of each of the two newly obtained vectors. This can be represented as:

$$\begin{aligned}
&\tilde{\boldsymbol{\xi}} = \tilde{\boldsymbol{\xi}}_+ - \tilde{\boldsymbol{\xi}}_- \\
&\tilde{\boldsymbol{\xi}}_+,\ \tilde{\boldsymbol{\xi}}_- \geq 0
\end{aligned}$$

where the two new vectors are defined as: $\tilde{\boldsymbol{\xi}}_{+i} = \max(0, \tilde{\boldsymbol{\xi}}_i)$ and $\tilde{\boldsymbol{\xi}}_{-i} = \max(0, -\tilde{\boldsymbol{\xi}}_i)$.

We then define the mean and the covariance matrix of the asset returns expressed by the new vectors:

$$\boldsymbol{\mu} = \boldsymbol{\mu}_+ - \boldsymbol{\mu}_-$$

$$\boldsymbol{\Sigma}_{2n} = \begin{bmatrix} \boldsymbol{\Sigma}_{++} & \boldsymbol{\Sigma}_{+-} \\ \boldsymbol{\Sigma}_{-+} & \boldsymbol{\Sigma}_{--} \end{bmatrix}$$

where $\boldsymbol{\mu}_+ = E(\tilde{\boldsymbol{\xi}}_+)$ and $\boldsymbol{\mu}_- = E(\tilde{\boldsymbol{\xi}}_-)$, $\boldsymbol{\Sigma}_{++}$ and $\boldsymbol{\Sigma}_{--}$ are $var(\tilde{\boldsymbol{\xi}}_+)$ and $var(\tilde{\boldsymbol{\xi}}_-)$ respectively and $\boldsymbol{\Sigma}_{+-} = \boldsymbol{\Sigma}_{-+}^T = cov(\tilde{\boldsymbol{\xi}}_+, \tilde{\boldsymbol{\xi}}_-)$.

We will also introduce a modified loss function, used in further calculations:

$$\mathcal{L}'(\boldsymbol{\omega}_+^\xi, \boldsymbol{\omega}_-^\xi, \boldsymbol{\omega}^\eta, \boldsymbol{\xi}_+, \boldsymbol{\xi}_-) = -\begin{pmatrix} \boldsymbol{\omega}_+^\xi \\ \boldsymbol{\omega}_-^\xi \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\xi}_+ \\ \boldsymbol{\xi}_- \end{pmatrix} - (\boldsymbol{\omega^\eta})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_{k+}}\boldsymbol{\xi}_+ - \mathbf{B_{k-}}\boldsymbol{\xi}_-\} \quad (4.5)$$

We can easily see that:

$$\mathcal{L}'(\boldsymbol{\omega}^\xi, -\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \boldsymbol{\xi}_+, \boldsymbol{\xi}_-) = \mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \boldsymbol{\xi})$$

Below we will use the defined partitions to compute a new upper bound on the VaR.

4.2.3 Partitioned Statistics Upper Bound on VaR

We will again define two sets of probabilities, for two different distribution assumptions. With slight abuse of notation, we define $\mathcal{P}_1$ as the set of probability distributions of $(\boldsymbol{\xi}_+^T, \boldsymbol{\xi}_-^T)^T$ with known mean vector $(\boldsymbol{\mu}_+^T, \boldsymbol{\mu}_-^T)^T$ and covariance matrix $\boldsymbol{\Sigma}_{2n}$ as defined above. Similarly, we let $\mathcal{P}_2$ be the set of distributions probability distributions of $(\boldsymbol{\xi}_+^T, \boldsymbol{\xi}_-^T)^T$ with given mean vector $(\boldsymbol{\mu}_+^T, \boldsymbol{\mu}_-^T)^T$ and support $\{(\boldsymbol{\xi}_+^T, \boldsymbol{\xi}_-^T)^T \in \mathbb{R}^{2n} : \boldsymbol{\xi}_+ \geq 0, \boldsymbol{\xi}_- \geq 0\}$.

Using these and the modified loss function defined in equation (4.5), we define:

$$\pi_i'(\boldsymbol{\omega}_+^\xi, \boldsymbol{\omega}_-^\xi, \boldsymbol{\omega}^\eta) = \sup_{\mathbb{P} \in \mathcal{P}_i} \mathbb{P}\text{-}VaR_\epsilon(\mathcal{L}'(\boldsymbol{\omega}_+^\xi, \boldsymbol{\omega}_-^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}}_+, \tilde{\boldsymbol{\xi}}_-) \quad for\ i = 1, 2$$

We will then use these two to define a new upper bound on VaR:

$$
\begin{aligned}
\pi_3(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) \;=\; \min \quad & \pi_1'(\boldsymbol{\omega}^\xi_{+1}, \boldsymbol{\omega}^\xi_{-1}, \boldsymbol{\omega}^\eta_1) + \pi_2'(\boldsymbol{\omega}^\xi_{+2}, \boldsymbol{\omega}^\xi_{-2}, \boldsymbol{\omega}^\eta_2) \\
s.\ t. \quad & \boldsymbol{\omega}^\xi_{+1}, \boldsymbol{\omega}^\xi_{-1}, \boldsymbol{\omega}^\xi_{+2}, \boldsymbol{\omega}^\xi_{-2} \in \mathbb{R}^n,\ \boldsymbol{\omega}^\eta_1, \boldsymbol{\omega}^\eta_2 \in \mathbb{R}^{m-n} \\
& \boldsymbol{\omega}^\xi_{+1} + \boldsymbol{\omega}^\xi_{+2} = \boldsymbol{\omega}^\xi \\
& \boldsymbol{\omega}^\xi_{-1} + \boldsymbol{\omega}^\xi_{-2} = -\boldsymbol{\omega}^\xi \\
& \boldsymbol{\omega}^\eta_1 + \boldsymbol{\omega}^\eta_2 = \boldsymbol{\omega}^\eta
\end{aligned} \tag{4.6}
$$

We will now show that this new upper bound is smaller or equal to the minimum of the first two upper bounds defined by equations (4.3) and (4.4) and that it is an upper bound on VaR over the intersection of the two sets of probability distributions $\mathcal{P}_1$ and $\mathcal{P}_2$.

For the first part, we will consider two feasible sets of values for our variables. First, consider the following values:

$$
\begin{aligned}
\boldsymbol{\omega}^\xi_{+1} &= \boldsymbol{\omega}^\xi & \boldsymbol{\omega}^\xi_{+2} &= 0 \\
\boldsymbol{\omega}^\xi_{-1} &= -\boldsymbol{\omega}^\xi & \boldsymbol{\omega}^\xi_{-2} &= 0 \\
\boldsymbol{\omega}^\eta_1 &= \boldsymbol{\omega}^\eta & \boldsymbol{\omega}^\eta_2 &= 0
\end{aligned}
$$

Then, we will get that $\pi_2'(0,0,0) = 0$ and $\pi_1'(\boldsymbol{\omega}^\xi, -\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\xi) = \pi_1(\boldsymbol{\omega}^\eta, \boldsymbol{\omega}^\xi)$. Hence we get:

$$\pi_3(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) = \min \pi_1(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta)$$

Second, we consider the values the other way around (so all $\boldsymbol{\omega}_1 = 0$ and all $\boldsymbol{\omega}_2$ have the initial values of $\boldsymbol{\omega}_1$). Then, we get a similar result, but for π_2:

$$\pi_3(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) = \min \pi_2(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta)$$

Hence, we can draw the straightforward conclusion that

$$\pi_3(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) \leq \min \{\ \pi_1(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta),\ \pi_2(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta)\ \} \tag{4.7}$$

To show the second part (that π_3 is an upper bound of VaR over the intersection of the two probability distributions sets), we will show that:

$$\pi_3(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) \geq \sup_{\mathbb{P}\in\mathcal{P}_1\cap\mathcal{P}_2} \mathbb{P}\text{-}VaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}}))$$

A first step is the observation that (see Theorem 6.1 from [4]):

$$\sup_{\mathbb{P}\in\mathcal{P}_i} \mathbb{P}\text{-}VaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}})) = \sup_{\mathbb{P}\in\mathcal{P}_i} \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}})) \tag{4.8}$$

Using the definition of π'_i and the observation above, we can reformulate the sum of the two. We make the notation $\mathcal{L}'_i = \mathcal{L}'(\boldsymbol{\omega}^\xi_{+i}, \boldsymbol{\omega}^\xi_{-i}, \boldsymbol{\omega}^\eta_i, \tilde{\boldsymbol{\xi}}_+, \tilde{\boldsymbol{\xi}}_-)$ for $i = 1, 2$ in order to improve readability.

$$\begin{aligned}
&\pi'_1(\boldsymbol{\omega}^\xi_{+1}, \boldsymbol{\omega}^\xi_{-1}, \boldsymbol{\omega}^\eta_1) + \pi'_2(\boldsymbol{\omega}^\xi_{+2}, \boldsymbol{\omega}^\xi_{-2}, \boldsymbol{\omega}^\eta_2) \\
= &\sup_{\mathbb{P}\in\mathcal{P}_1} \mathbb{P}\text{-}VaR_\epsilon(\mathcal{L}'_1) + \sup_{\mathbb{P}\in\mathcal{P}_2} \mathbb{P}\text{-}VaR_\epsilon(\mathcal{L}'_2) \\
= &\sup_{\mathbb{P}\in\mathcal{P}_1} \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}'_1) + \sup_{\mathbb{P}\in\mathcal{P}_2} \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}'_2) \\
\geq &\, \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}'_1) + \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}'_2) \quad \forall \mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2
\end{aligned} \tag{4.9}$$

We now use the result from Theorem 1 of [7], which defines $\mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}}))$ as:

$$\mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}})) = \inf_\beta \left\{ \beta + \frac{1}{\epsilon} E_{\mathbb{P}}([\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}}) - \beta]_+) \right\} \tag{4.10}$$

where $[x]_+ = x$ when $x > 0$ and $[x]_+ = 0$ otherwise.

This allows us to reformulate the sum obtained as a lower bound for the sum of π'_1 and π'_2:

$$\begin{aligned}
&\mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}'_1) + \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}'_2) \quad \forall \mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2 \\
= &\inf_{\beta_1} \left\{ \beta_1 + \frac{1}{\epsilon} E_{\mathbb{P}}([\mathcal{L}'_1 - \beta_1]_+) \right\} + \inf_{\beta_2} \left\{ \beta_2 + \frac{1}{\epsilon} E_{\mathbb{P}}([\mathcal{L}'_2 - \beta_2]_+) \right\} \;\; \forall \mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2 \\
= &\inf_{\beta_1, \beta_2} \left\{ \beta_1 + \beta_2 + \frac{1}{\epsilon} E_{\mathbb{P}}([\mathcal{L}'_1 - \beta_1]_+ + [\mathcal{L}'_2 - \beta_2]_+) \right\} \;\; \forall \mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2
\end{aligned} \tag{4.11}$$

We now reformulate the expression inside the expectation function. To this purpose, we use the following result (where a and b are any number):

$$\max(a, 0) + \max(b, 0) \geq \max(a + b, 0)$$

Since $[x]_+ = max(x, 0)$ by definition, we conclude that:

$$\begin{aligned} & [\mathcal{L}'_1 - \beta_1]_+ + [\mathcal{L}'_2 - \beta_2]_+ \\ \geq\ & [\mathcal{L}'_1 - \beta_1 + \mathcal{L}'_2 - \beta_2]_+ \end{aligned}$$

From the definition of $\mathcal{L}'$ in equation (4.5) and the conditions on the variables from the definition of π_3 in equation (4.6), we can also compute the sum of $\mathcal{L}_1$ and $\mathcal{L}_2$.

$$\begin{aligned} \mathcal{L}'_1 + \mathcal{L}'_2 =\ & -\begin{pmatrix} \boldsymbol{\omega}^\xi_{+1} \\ \boldsymbol{\omega}^\xi_{-1} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\xi}_+ \\ \boldsymbol{\xi}_- \end{pmatrix} - (\boldsymbol{\omega}^{\boldsymbol{\eta}}_{\mathbf{1}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_{k+}}\boldsymbol{\xi}_+ - \mathbf{B_{k-}}\boldsymbol{\xi}_-\} \\ & -\begin{pmatrix} \boldsymbol{\omega}^\xi_{+2} \\ \boldsymbol{\omega}^\xi_{-2} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\xi}_+ \\ \boldsymbol{\xi}_- \end{pmatrix} - (\boldsymbol{\omega}^{\boldsymbol{\eta}}_{\mathbf{2}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_{k+}}\boldsymbol{\xi}_+ - \mathbf{B_{k-}}\boldsymbol{\xi}_-\} \\ =\ & -\begin{pmatrix} \boldsymbol{\omega}^\xi_{+1} + \boldsymbol{\omega}^\xi_{+2} \\ \boldsymbol{\omega}^\xi_{-1} + \boldsymbol{\omega}^\xi_{-2} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\xi}_+ \\ \boldsymbol{\xi}_- \end{pmatrix} - (\boldsymbol{\omega}^{\boldsymbol{\eta}}_{\mathbf{1}} + \boldsymbol{\omega}^{\boldsymbol{\eta}}_{\mathbf{2}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_{k+}}\boldsymbol{\xi}_+ - \mathbf{B_{k-}}\boldsymbol{\xi}_-\} \\ =\ & -\begin{pmatrix} \boldsymbol{\omega}^\xi \\ -\boldsymbol{\omega}^\xi \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\xi}_+ \\ \boldsymbol{\xi}_- \end{pmatrix} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_{k+}}\boldsymbol{\xi}_+ - \mathbf{B_{k-}}\boldsymbol{\xi}_-\} \\ =\ & -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\tilde{\boldsymbol{\xi}}_+ + (\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\tilde{\boldsymbol{\xi}}_- - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_k}\tilde{\boldsymbol{\xi}}\} \\ =\ & -(\boldsymbol{\omega}^{\boldsymbol{\xi}})^{\mathbf{T}}\tilde{\boldsymbol{\xi}} - (\boldsymbol{\omega}^{\boldsymbol{\eta}})^{\mathbf{T}} \max_k \{\mathbf{a_k} + \mathbf{B_k}\tilde{\boldsymbol{\xi}}\} \\ =\ & \mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \boldsymbol{\xi}) \end{aligned}$$

So we can replace this into the above reformulation of the expression inside the expectation function and obtain:

$$[\mathcal{L}'_1 - \beta_1]_+ + [\mathcal{L}'_2 - \beta_2]_+ \ \geq\ [\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \boldsymbol{\xi}) - \beta_1 - \beta_2]_+$$

Finally, we replace this back into our calculation of the sum of $\mathbb{P}$-$CVaR_\epsilon(\mathcal{L}'_1)$ and $\mathbb{P}$-$CVaR_\epsilon(\mathcal{L}'_2)$ from equation (4.11) and introduce the new variable $\beta = \beta_1 + \beta_2$.

$$
\begin{aligned}
&\mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L'}_1) + \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L'}_1) \quad \forall \mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2 \\
&\geq \inf_{\beta_1,\beta_2} \left\{ \beta_1 + \beta_2 + \frac{1}{\epsilon} E_{\mathbb{P}}([\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \boldsymbol{\xi}) - \beta_1 - \beta_2]_+) \right\} \quad \forall \mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2 \\
&\geq \inf_{\beta} \left\{ \beta + \frac{1}{\epsilon} E_{\mathbb{P}}([\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}}) - \beta]_+) \right\} \quad \forall \mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2 \\
&= \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}})) \quad \forall \mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2
\end{aligned}
$$

We can now replace this back into our equation (4.9) and obtain:

$$
\begin{aligned}
&\pi'_1(\boldsymbol{\omega}^\xi_{+1}, \boldsymbol{\omega}^\xi_{-1}, \boldsymbol{\omega}^\eta_1) + \pi'_2(\boldsymbol{\omega}^\xi_{+2}, \boldsymbol{\omega}^\xi_{-2}, \boldsymbol{\omega}^\eta_2) \\
&\geq \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}})) \quad \forall \mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2
\end{aligned}
$$

Using our earlier observation from equation (4.8), this is equivalent to:

$$
\begin{aligned}
&\pi'_1(\boldsymbol{\omega}^\xi_{+1}, \boldsymbol{\omega}^\xi_{-1}, \boldsymbol{\omega}^\eta_1) + \pi'_2(\boldsymbol{\omega}^\xi_{+2}, \boldsymbol{\omega}^\xi_{-2}, \boldsymbol{\omega}^\eta_2) \\
&\geq \sup_{\mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2} \mathbb{P}\text{-}CVaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}})) \\
&\geq \sup_{\mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2} \mathbb{P}\text{-}VaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}}))
\end{aligned}
$$

Finally, using the above relation and the definition of π_3 from equation (4.6), we obtain the relation we wanted to prove:

$$
\pi_3(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) \geq \sup_{\mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2} \mathbb{P}\text{-}VaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}})) \tag{4.12}
$$

From the two relations, defined in equations (4.7) and (4.12), we have obtained an upper bound and a lower bound for π_3. Hence, we have shown that the partitioned statistics bound $\pi_3(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta)$ is a tighter upper bound for VaR than both other bounds defined, since it considers information about the asymmetry of the returns distributions. This is all summarised in the following theorem:

Theorem 4.2

$$\begin{aligned}
\pi_3(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) \;=\; \min \quad & \pi_1'(\boldsymbol{\omega}^\xi_{+1}, \boldsymbol{\omega}^\xi_{-1}, \boldsymbol{\omega}^\eta_1) + \pi_2'(\boldsymbol{\omega}^\xi_{+2}, \boldsymbol{\omega}^\xi_{-2}, \boldsymbol{\omega}^\eta_2) \\
s.\ t. \quad & \boldsymbol{\omega}^\xi_{+1}, \boldsymbol{\omega}^\xi_{-1}, \boldsymbol{\omega}^\xi_{+2}, \boldsymbol{\omega}^\xi_{-2} \in \mathbb{R}^n,\ \boldsymbol{\omega}^\eta_1, \boldsymbol{\omega}^\eta_2 \in \mathbb{R}^{m-n} \\
& \boldsymbol{\omega}^\xi_{+1} + \boldsymbol{\omega}^\xi_{+2} = \boldsymbol{\omega}^\xi \\
& \boldsymbol{\omega}^\xi_{-1} + \boldsymbol{\omega}^\xi_{-2} = -\boldsymbol{\omega}^\xi \\
& \boldsymbol{\omega}^\eta_1 + \boldsymbol{\omega}^\eta_2 = \boldsymbol{\omega}^\eta
\end{aligned}$$

$$\sup_{\mathbb{P} \in \mathcal{P}_1 \cap \mathcal{P}_2} \mathbb{P}\text{-}VaR_\epsilon(\mathcal{L}(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta, \tilde{\boldsymbol{\xi}})) \leq \pi_3(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta) \leq \min\ \{\ \pi_1(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta), \pi_2(\boldsymbol{\omega}^\xi, \boldsymbol{\omega}^\eta)\ \}$$

4.3 Summary

In this chapter we have extended our WPVaR Model to cope not only with unknown distribution, but also with uncertainty regarding the distribution's parameters. To this purpose, we have used two techniques: first, we used a box-type uncertainty where the mean and covariance matrix are not exactly known, but within some given range; second, we have used partition statistics which allows to consider asymmetric distributions. Hence, we have defined and proved two new theorems, which present the two robust problems as tractable SDPs.

These represent the main novel contribution of our work, since they incorporate both asymmetry in asset returns and option expiring beyond the investment horizon in the same portfolio optimisation problem. They are realistic extensions, since in practice investors do not know the true moments of the distributions of asset returns, nor can they accurately infer them. Moreover, the distributions of asset returns are often skewed, which is another factor our initial WPVaR could not consider. Hence, such robust models could and should prove more stable in situations when large changes in the market occur, such as a sudden drop in the price of certain assets.

Chapter 5

Implementation

The next part of our project constitutes implementation of our derived models. The chosen programming language is matlab, due to its extensive built-in mathematical operations (e.g. matrices operations, which otherwise need additional implementation), to its easy to use graphical tools (e.g. plotting results is straightforward) and to the large availability of toolboxes related to our required functionality (e.g. frameworks for solving a SDP). Hence, this stage also involves research into available matlab tooloboxes or software packages (such as SDPT3, SeDuMi, yalmip).

Moreover, the implementation is required to perform numerical tests and provides a practical way to check our derived model yields similar results to the initial one in the particular case when only considering options at expiry. This can quickly identify an erroneous model, without the need of mathematical proofs.

Finally, we adapt our implementation to run on HTCondor, a high-throughput batch-processing system available in the Department of Computing of Imperial College London. This enables us to perform large data simulations in a relatively short amount of time.

5.1 SDP Solvers and WPVaR Models Implementation

Before the actual implementation of the models, we investigated some of the available matlab solvers for a SDP program. We mainly considered the Yalmip framework [18], along with the SeDuMi [19] and SDPT3 [20] solvers.

5.1.1 SDP Solvers

Yalmip [18] is a framework for modelling language for solving convex and non-convex optimisation problems. Its main advantages are that it is implemented as a matlab toolbox, it is consistent with the matlab syntax and it supports a very large number of optimisation problems (linear programming, quadratic programs, semidefinite programs, second order cone programs etc.). Its central idea is to provide a simple interface to the user, while relying on external solvers for the actual computations (although it also provides some actual computation for certain problems).

Yalmip is built to make it very easy to chose the desired solver, by setting the solver using the `sdpsettings` function, as in the following example (where we invoke the SDPT3 solver):

```
options = sdpsetting('solver', 'SDPT3');
```

The solvers we considered for our SDP programs are SeDuMi [19] and SDPT3 [20], [21]. However, using any of these has both benefits and disadvantages. One advantage is that they are both accessible through yalmip and they both provide access to their source code. But, error handling and debugging any errors proved to be rather hard and involved understanding of some of their implementation.

Hence, using Yalmip and SDPT3 we were able to implement our WPVaR models defined in theorems 3.1 and 4.1.

5.1.2 WPVaR Models Implementation

We implemented our initial WPVaR model from theorem 3.1 and the box-type robust optimisation from theorem 4.1. Also, we implemented the corresponding portfolio optimisation problems, which provide as result the weight allocation obtained using the different risk measures.

Using Yalmip and SDPT3, the obtained SDPs were implemented using a matlab-like syntax and an almost straightforward translation of the mathematical models. One of the main characteristics of all function is that they are fully parameterised, which ensures we could easily vary their values during our simulations.

We also considered the many similarities when computing the optimal portfolios, for different risk measures. Thus, we provide unique functions to obtain the desired results, which simply take as parameters the optimisation problems to use for computing the actual results. This proved very useful when adding new risk models and processing the new results. Also, this allowed us to reuse most of the code when performing different kinds of tests.

One of the earliest tests we performed involved the initial implementation of our WPVaR model. This allowed us to perform an early validation of our derived SDP.

5.2 Initial Validation

Since at this stage we were able to provide implementation for both our derived WPVaR model and the basic version presented in [4], we performed a practical initial validation that our model yields similar results to the one in the paper, on the basic case where only options at expiry are used. This provides a quick check, which can indicate a mistake without the need of additional mathematical proofs. However. this would not be enough to prove the models are equivalent, result for which a formal proof is required and provided.

This validation proved very useful, since we initially did find inconsistencies between the results of the two implementations. The mismatches indicated a mistake in our derivation. When double checking our derived model, we found indeed that a mistake has been made. We corrected the work and derived the new WPVaR model shown in theorem 3.1.

A new run of initial validation tests showed the two models now yield identical results for the basic case. Hence, we moved on to proving the equivalence of the two using a formal mathematical proof, shown in section 3.3. The proof confirmed the equivalence that the tests suggested.

After establishing this equivalence, we continued with the implementation of gathering input data and processing the results.

5.3 Data Flow

The main purpose of our simulations is to compare the results obtained using our WPVaR model and using our box-type robust WPVaR model with those obtained using other models — VaR or CVaR. We have performed two types of simulations: using generated data, with a known distribution; or using historical data obtained from yahoo finance[1]. These allow us to describe the efficient frontiers (real, estimated and actual) for all three portfolio optimisation approaches and to compare out-of-sample results from the four models. Figure 5.1 describes the general data flow of our simulations.

So, to perform the required experiments we have to implement all three models, generate input data or obtain it from yahoo finance and process the results. We first discuss the implementation necessary to acquire input data and then the models we implemented for the three portfolio optimisation problems.

5.3.1 Simulating Input Data

To generate sample data for our experiments, we assumed known mean vector and covariance matrix of the basic assets returns. We then used three different distributions to generate data samples for our problems: multivariate normal, multivariate T and multivariate Poisson distributions.

Matlab provides functions for the multivariate normal and t distributions. However, while the normal distributions requires the mean vector and covariance matrix as parameters, the

[1]http://uk.finance.yahoo.com/

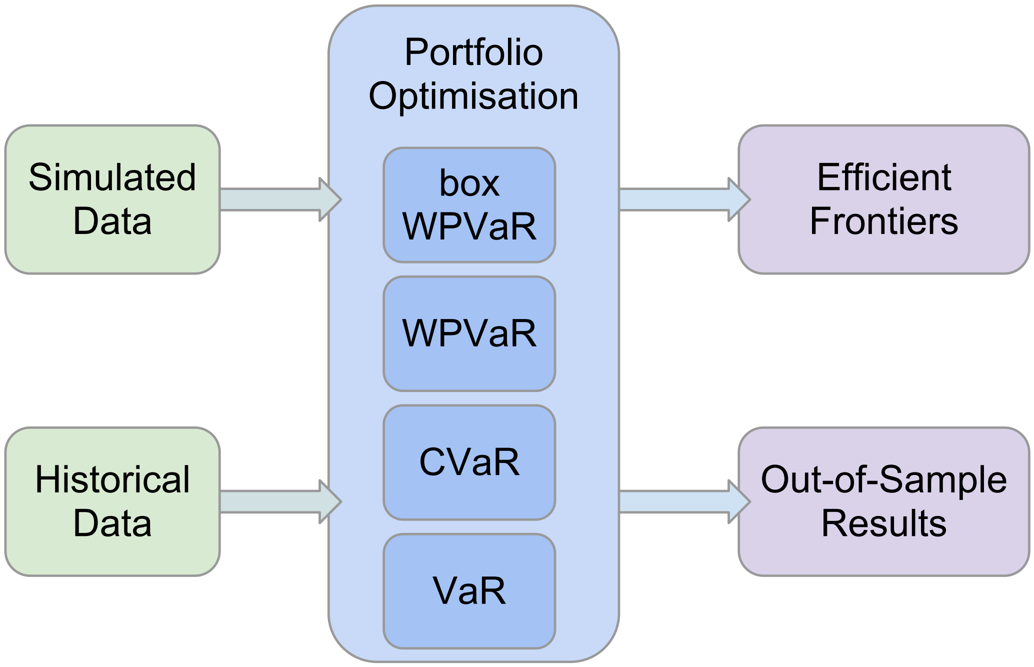


Figure 5.1: Data flow diagram, showing input obtained either through generated data (with known distribution) or historical data and comparison results for the three portfolio optimisations problems — minimising WPVaR, VaR or CVaR.

multivariate t requires the correlation matrix and the degrees of freedom. The correlation matrix can be obtained from the covariance matrix straightforward, from the definitions of the correlation and the relation between standard deviation and variance. To compute the degrees of freedom, we have used the straightforward formula $n - 1$, where n is the sample size. Since this yielded acceptable samples, we considered something more detailed is not required for the purpose of our experiments. Finally, we shifted the values to obtain samples with our desired mean, instead of 0, the default mean used by the matlab function.

To generate sample data from the multivariate Poisson distribution, we used the function `sampleCovPoisson`[2] available on the Matlab Central File Exchange website. Again, this

[2]http://www.mathworks.co.uk/matlabcentral/fileexchange/20591-sampling-from-multivariate-

was straightforward since it requires the mean and the covariance matrix. However, it operates on integer numbers only, so we had to scale our means for the function to provide a wide range of samples and scale the obtained data back to acceptable values for our returns.

For our experiments, we have repeatedly sampled a distribution, with a given time series length, used the sample data as input to our portfolio optimisations problems and computed the desired values (expected return, actual return, expected VaR, actual VaR etc.). Then, the results were averaged over the number of simulations, in order to obtain statistically significant results.

5.3.2 Accessing Historical Financial Data

We use yahoo finance to obtain historical data for the price of real stocks. Matlab provides a few easy to use functions which provide access to basic stock data, which is enough for the purpose of our experiments. The listing below shows a simple example which obtains the close prices of the stock with given `stockId` between the given two dates.

```
y = yahoo;
stockHistoricalClosePrices = fetch(y, stockId, closePriceId, ...
                                   startDate, endDate);
```

Similar function call can be used to obtain different historical information about the stocks. In general, we have chosen stocks from related markets (since these supposedly have some correlation to each other) and as much historical data as possible.

The next step was to implement the portfolio optimisation problems used to compare with our models — minimising the VaR or the CVaR of the portfolio.

5.3.3 VaR and CVaR Portfolio Optimisation Implementation

We followed the approach described in [7] to implement a computationally tractable program for computing the weights which yield a minimum CVaR portfolio. By definition, the VaR is never larger than the CVaR, so a portfolio with low CVaR automatically has a low VaR. The approach from the paper to describe the β-CVaR is to determine it from the formula:

correlated-binary-and-poisson-random-variables

$$\phi_\beta(x) = \min_{\alpha \in \mathbb{R}} F_\beta(x, \alpha)$$
$$F_\beta(x, \alpha) = \alpha + (1-\beta)^{-1} \int_{\mathbf{y} \in \mathbb{R}^m} [f(x, \mathbf{y}) - \alpha]^+ p(\mathbf{y}) d\mathbf{y}$$

where $x \in X$ represents an available portfolio, ϕ_β is the β-CVaR and F_β has some desirable features shown in Theorem 1 of [7]. $f(x, \mathbf{y})$ represents the loss function and $p(\mathbf{y})$ is the probability density function of $\mathbf{y} \in \mathbb{R}^m$, assumed for convenience to exist. This further leads to formulation of Theorem 2 of the same paper, which states that:

$$\min_{x \in X} \phi_\beta(x) = \min_{(x,\alpha) \in X \times \mathbb{R}} F_\beta(x, \alpha)$$

Applying this result to portfolio optimisation, we obtain the following minimisation problem. This provides the optimal portfolio in terms of minimising the CVaR. We use this when we have a set of historical samples $\boldsymbol{\xi}_k \quad \forall k = 1, \ldots, q$, where q is the number of historical samples provided.

$$\begin{aligned}
&\min \ \alpha + \frac{1}{q\epsilon} \sum_{k=1}^{q} u_k \\
&s.\ t.\ \boldsymbol{\omega} \in \mathbb{R}^n,\ \mathbf{u} \in \mathbb{R}^q,\ \alpha \in \mathbb{R} \\
&\qquad u_k \geq 0 \ \ \forall k = 1, \ldots, q \\
&\qquad \boldsymbol{\omega}^T \boldsymbol{\xi}_k + \alpha + u_k \geq 0 \ \ \forall k = 1, \ldots, q \\
&\qquad \omega_i \geq 0 \ \ \forall i = 1, \ldots, n,\ \sum_{i=1}^{n} \omega_i = 1 \\
&\qquad \boldsymbol{\omega}^T \bar{\boldsymbol{\xi}} \geq \bar{r}_{min}
\end{aligned} \tag{5.1}$$

where $\bar{r}_{min}$ is the minimum required expected return of the portfolio. Also note that this formulation does not allow short selling, since $\omega_i \geq 0$. We use this formulation throughout all our experiments, in comparing the out-of-sample results from our WPVaR formulation with these and the VaR portfolio optimisation problem formulation.

The next step is to formulate an optimisation problem which minimises VaR. This is done differently for data that follows a normal distribution and that does not. For normal distribution, we have the following straightforward VaR formulation [3]:

$$VaR_\epsilon = k(\epsilon)\sqrt{\boldsymbol{\omega}^T\boldsymbol{\Sigma}\boldsymbol{\omega}} - \mathbf{r}^T\boldsymbol{\omega}$$
$$k(\epsilon) = -\Phi^{-1}(\epsilon)$$

Consequently, the portfolio optimisation problem is (short selling is not allowed):

$$\begin{aligned}
&\min\ k(\epsilon)\sqrt{\boldsymbol{\omega}^T\boldsymbol{\Sigma}\boldsymbol{\omega}} - \mathbf{r}^T\boldsymbol{\omega}\\
&s.\ t.\ \boldsymbol{\omega} \in \mathbb{R}^n\\
&\qquad \omega_i \geq 0 \ \ \forall i = 1, \ldots, n,\ \sum_{i=1}^{n} \omega_i = 1\\
&\qquad \boldsymbol{\omega}^T\bar{\boldsymbol{\xi}} \geq \bar{r}_{min}
\end{aligned} \tag{5.2}$$

To formulate an optimisation problem which minimises the VaR of a set of data with unknown distribution is not as straightforward. To this purpose, we initially followed the approach described in [22]. This result is based on an approximation of the historical VaR, named smoothed VaR (SVaR). However, we decided this yielded a problem formulation which was more complicated than necessary for our experiments. So, we decided to use a mixed-integer programming formulation, for computing the historical VaR. This yields the following portfolio optimisation problem:

$$\begin{aligned}
&\min\ \gamma\\
&s.\ t.\ \gamma \in \mathbb{R},\ \boldsymbol{\omega} \in \mathbb{R}^n,\ \mathbf{b} \in \{0,\ 1\}^I\\
&\qquad b_i\gamma + (1 - b_i)M \geq -\boldsymbol{\omega}^T\boldsymbol{\xi}_i \ \ \forall i = 1, \ldots, I\\
&\qquad \sum_{i=1}^{I} b_i \geq (1 - \epsilon)I\\
&\qquad \omega_i \geq 0 \ \ \forall i = 1, \ldots, n,\ \sum_{i=1}^{n} \omega_i = 1\\
&\qquad \frac{\sum_{i=1}^{I} \boldsymbol{\omega}^T\boldsymbol{\xi}_i}{I} \geq \bar{r}_{min}
\end{aligned} \tag{5.3}$$

where $\boldsymbol{\xi}_i$ is the i-th historical sample (of I in total) and M is just a big constant (for instance we take $M = \max_i \|\boldsymbol{\xi}_i\|_1$).

Also, we decided that due to the nature of our models, it would be more fit to display the efficient frontiers in terms of the VaR of the portfolios (instead of the variance as mentioned in chapter 2). So we use a similar formulation to compute the VaR of a given portfolio (given $\boldsymbol{\omega}$).

5.4 HTCondor Simulation

HTCondor is a high-throughput computing framework, a batch-processing system that provides job queueing and scheduling mechanisms — users need only submit jobs to HTCondor, which then chooses when and where (on which machine) to run them [23].

The main advantage of using HTCondor to run our simulation is the much larger number of simulations we can run in a given time period. This is due to the nature of HTCondor, which uses as many machines as available at any given time. Moreover, implementing the required code and scripts is fairly straightforward.

The first step was to parallelise the part of the code which computes the simulations. To this purpose, we only had to perform relatively small changes and we were able to reuse most of our code. The main change was the removal of the outermost `if` which performed the provided number of simulations:

```
for i = 1 : noSimulations
    runOneEstimation(means, covarianceMatrix, m, n, ...);
end
```

Instead, we built a matlab script which simply calls the `runOneEstimation` function. Also, the way we process the results has changed. Initially we processed them at the end of the simulation, but for condor we simply print the results and have separate matlab scripts which compute the required values afterwards.

Second, a job specification file was required, which indicates the program to be run and several HTCondor options. An example of such file is attached in appendix B. As we can see, we can easily change the number of simulations by setting the “queue” parameter to the desired value. Each simulation yields its own output and error files, which makes it easy to process the results afterwards. Also, it is straightforward to redirect our output and error files (`.out` and `.err` respectively) to our desired location.

Once all the simulations outputs are obtained, we use a simple script to obtain the desired results. Since we want to be able to process these results in different ways to gather as much information as possible, we simply create one file for each parameter we are interested in, where we aggregate all the obtained results. We can then process these results using matlab, performing different tests and analysis, detailed in chapter 6 (we will mainly compute means and variances in the different values and plot appropriate graphs and figures to display the results).

5.5 Summary

We have chosen matlab to implement our models due to its readily available mathematical operations and extensive toolboxes in solving SDPs, which we use for our SDPs. We implemented our optimisation problems having in mind the large possibility for code reuse and the need to vary the different parameters during our simulations. The implementation of our initial WPVaR model proved useful even from the early stages of the project, when it helped identify a difference between our model and the initial WPVaR only considering options at expiry.

Since our simulations follow a clear data flow, we provide implementation for generating sample data which follow different distribution and for obtaining historical financial data. Then, we provide different ways to compare and analyse the results from the different portfolio optimisation problems, which use different risk measures. Finally, we provide the possibility to run our simulations on HTCondor to have the ability to run a very large number of numerical tests.

Chapter 6

Evaluation

To evaluate our WPVaR models we perform numerical tests to obtain an indication of the results they yields on the general case, by themselves and in comparison with other risk measures, such as Worst-case Conditional Value-at-Risk and Value-at-Risk.

For a broader observation, we will perform test with both simulated and real financial data. Simulated tests are based on simulated returns, obtained using a known distribution. In this case, we check if the model behaves as expected. On the other hand, we also perform real data tests, with actual input prices, whose distribution is unknown. We can then compute out-of-sample results and see how the different risk measures perform.

6.1 Numerical Tests

Numerical tests follow these two approaches, to obtain and analyse practical results of the model. They include test with simulated sample data (obtained using simulated returns, generated by a known distribution with given first two moments) and with real historical financial sample data (with unknown distribution). The results are compared with those of other models (such as CVaR and VaR).

6.1.1 Simulations Using Generated Data

First, we performed some test using data generated using the multivariate normal distribution. To sample this, we used the matlab provided function `mvnrnd`. We used these samples to compute the three efficient frontiers using different portfolio optimisation problems. These frontiers are computed as mean return in terms of Value-at-Risk.

Real frontier — obtained using the known mean vector and covariance matrix of the assets returns as input to the portfolio optimisation problem;

Estimated frontier — obtained using the sample mean and sample covariance matrix (estimated from sample data) as input to the portfolio optimisation problem;

Actual frontier — obtained computing the actual return and VaR of the portfolio found using the sample estimates.

We performed these tests using three different multivariate distribution for generating the samples — multivariate normal distribution, multivariate poisson distribution and multivariate t-distribution.

6.1.1.1 Multivariate Normal Distribution Results

We first performed experiments using samples generated with the multivariate normal distribution. We show below the results obtained after averaging the outcome of 200 runs obtained using HTCondor. Figure 6.1 shows the average estimated and average actual frontiers obtained using the three four measures. This shows that the frontiers obtained using the VaR and our WPVaR are rather close to each other (the estimated ones and the actual ones respectively).

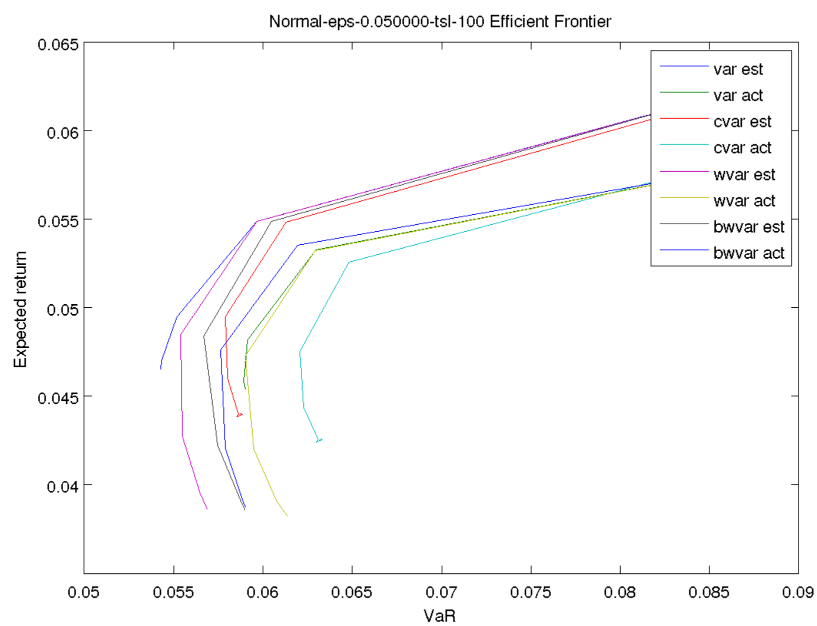


Figure 6.1: Average estimated and actual frontiers for the three portfolio optimisation problems, computed for samples of normal distribution, for a value of $\epsilon = 0.05$.

However, it is also a rather significant difference between the estimated and actual frontier obtained with the box-type robust WPVaR model and the other pairs. As we can see, these two frontiers are much closer to each other than the other pairs, which indicates the box-type robust WPVaR model might be a better risk estimate, in terms of the difference between the expected and the actual return.

A second analysis we perform is to see how the average estimated and actual frontiers compare to the true one. Figures 6.2, 6.3 and 6.4 below show all three frontiers for VaR, WPVaR and box-type robust WPVAR as risk measures respectively. Again, we can see that the average estimated and actual frontiers do not get significantly closer to the true frontier in the case of using WPVaR as a risk measure as opposed to VaR.

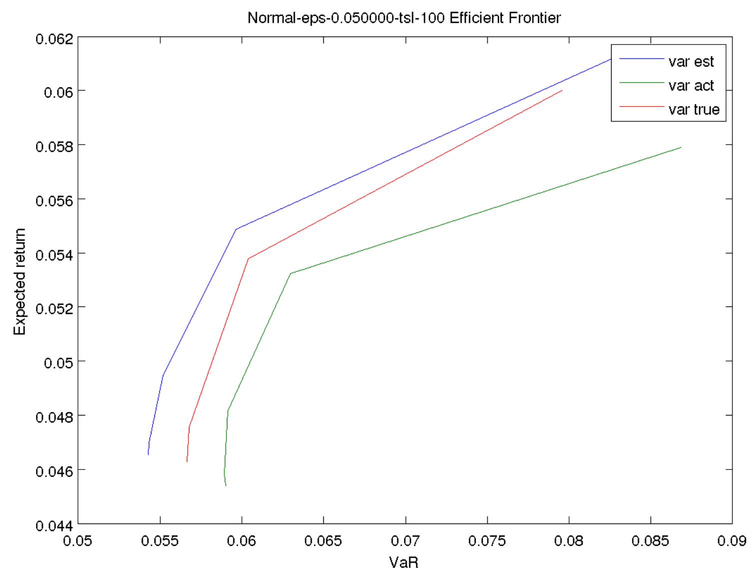


Figure 6.2: Estimated, actual and true frontiers for the portfolio optimisation problem using VaR as risk measure, computed for samples of normal distribution and $\epsilon = 0.05$.

However, the frontiers generated by the box-type robust model are significantly closer to the true frontier than in the case of the other risk models. Again, this indicates this robust model as a more desirable risk model, since the estimated and actual returns it yields are closer to the true best possible returns (represented by the true frontier), which are the portfolios every investor is interested in.

To further study the difference between the different risk measures, we repeat the experiments for different values of ϵ, which can be seen as an indication of the level of risk the investor is willing to take (higher values for ϵ correspond to riskier investments). The average estimated and actual efficient frontiers obtained for a smaller value of $\epsilon = 0.01$ and for larger values $\epsilon = 0.1$ and $\epsilon = 0.2$ are shown in appendix C section C.1. Comparisons between all three frontiers (true, average estimated an average actual) are also shown when minimising VaR, WPVaR and box-type robust WPVaR, for the more common values of $\epsilon = 0.1$ and $\epsilon = 0.01$.

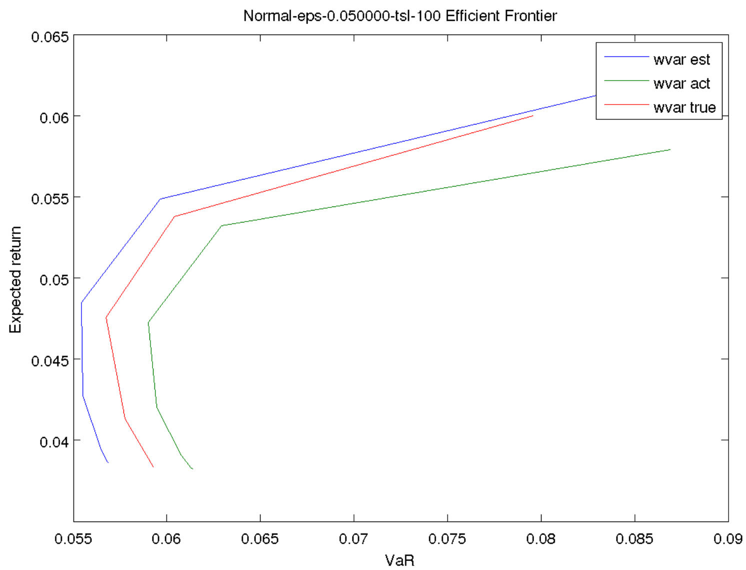


Figure 6.3: Estimated, actual and true frontiers for the portfolio optimisation problem using our WPVaR as risk measure, computed for samples from normal distribution and $\epsilon = 0.05$.

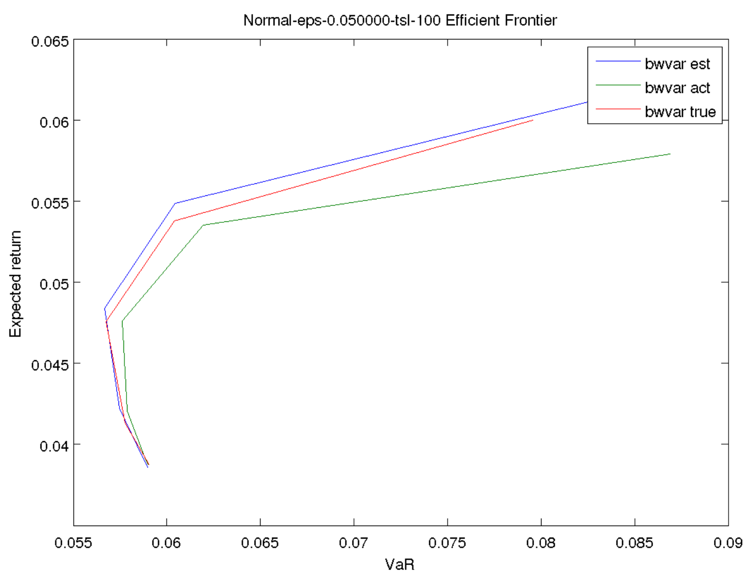


Figure 6.4: Estimated, actual and true frontiers for the portfolio optimisation problem using box-type robust WPVaR as risk measure, computed for samples form normal distribution and $\epsilon = 0.05$.

The first three figures show how corresponding pairs of efficient frontiers (average estimated and actual for VaR, average estimated and actual for CVaR etc.) get further apart from each other as ϵ grows (considering the different scales). However, the two frontiers corresponding to the box-type robust WPVaR remain considerably closer to each other than the other pairs of efficient frontiers. This supports our initial theory that the box-type robust WPVaR model may be a more reliable risk measure than the others. A similar conclusion can be drawn from the next figures. These show that the average estimated and actual frontiers obtained when using the robust WPVaR model are again closer to the true frontier than in the case of VaR and WPVaR.

A different approach we can take to comparing the portfolios obtained by minimising different risk measures is to compare how much the expected return varies between different runs for the four portfolio optimisation problems. To this purpose, figure 6.5 shows a whisker plot for the estimated returns of the three portfolios, for a value of $\epsilon = 0.05$. We have 10 different points since they correspond to the 10 different minimum expected portfolio returns we used to compute the efficient frontiers. Similarly, figure 6.6 shows whisker plots for the actual returns of the three portfolios.

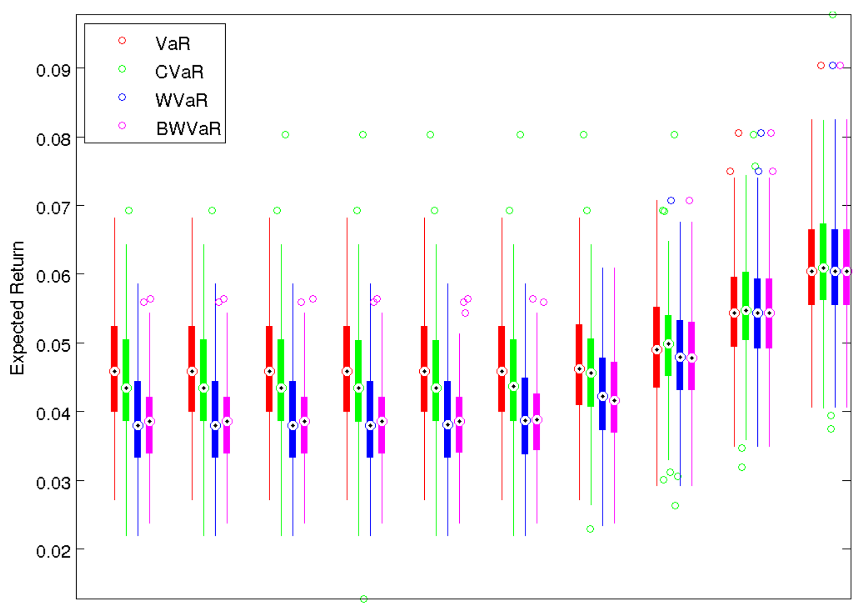


Figure 6.5: Whisker plot for the estimated returns computed for samples generated using the multivariate normal distribution and a value of $\epsilon = 0.05$.

In the case of estimated returns, we cannot see any significant difference between the four risk measures. But, things are different in the case of actual returns. Figure 6.6 indicates a

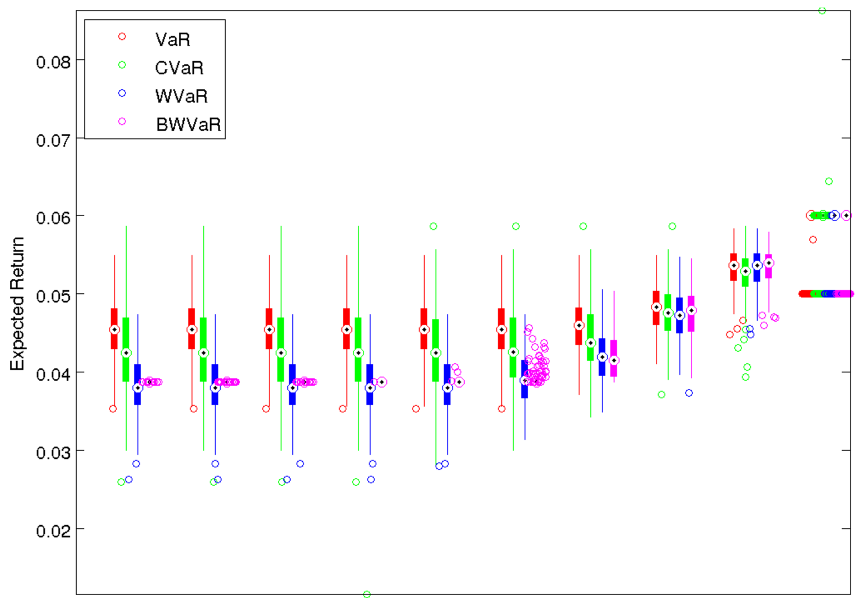


Figure 6.6: Whisker plot for the actual returns computed for samples generated using the multivariate normal distribution and a value of $\epsilon = 0.05$.

considerable difference between the variance of the actual return when using the box-type robust WPVaR model as opposed to the other three. However, this difference does decrease as the required minimum expected return increases.

Another way to compare the three models is not in terms of their expected return, but in terms of how much their portfolio choice varies over different runs. This can be done by looking at how much the weights for a particular asset vary across runs, for the same minimum portfolio return. Figures 6.7 and 6.8 show whisker plots for the weights of the five assets considered, for the four different portfolio optimisation problems and for two different expected minimum returns.

We can see that for most assets the variance in the weights obtained using our WPVaR model is smaller than the other two. This indicates the WPVaR model may be more robust, yielding more similar portfolios weight allocations between different simulation. Moreover, even for the other assets (e.g. figure 6.7, third asset), although the box size is larger for our WPVaR model, the number of outliers is much larger for the VaR and CVaR models. However, in the case of the box-type robust WPVaR model, the results are much more exciting. The difference in asset weights variance is much larger between the robust model and the other three: in general, this yields considerably more stable weights over different runs than all the other three models.

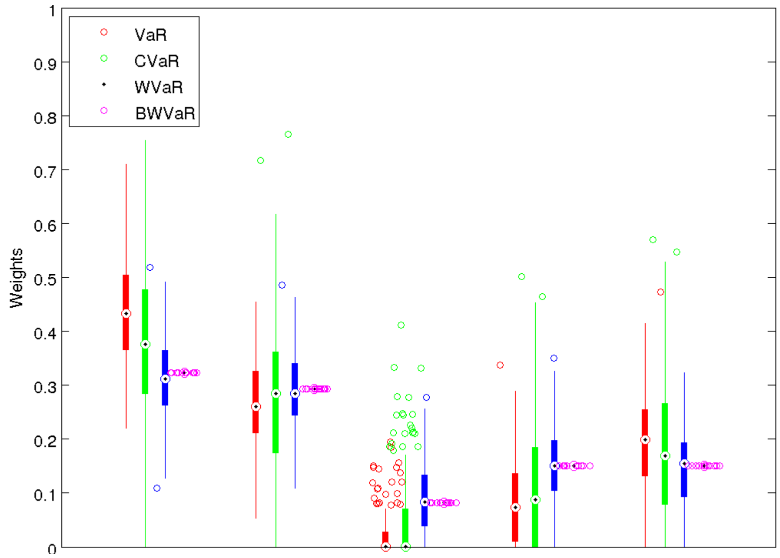


Figure 6.7: Whisker plot showing the variance in weights for the five assets over 200 different simulations. Results are shown for multivariate normally distributed sample data, for the smallest expected minimum return and $\epsilon = 0.05$.

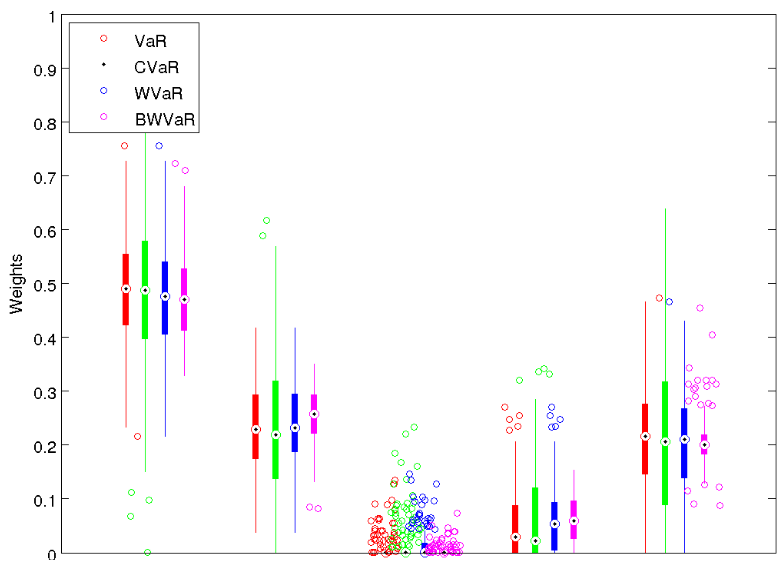


Figure 6.8: Whisker plot showing the variance in weights for the five assets over 200 different simulations. Results are shown for multivariate normally distributed sample data, for one of the largest expected minimum returns and $\epsilon = 0.05$.

6.1.1.2 Multivariate Skewed Distributions Results

Although the experiments with the multivariate normal distribution are a good start, they are not very realistic. This is the case since asset returns distributions are often skewed in practice, which is not true for the normal distribution. So, we also used samples generated with the multivariate Poisson and the multivariate t-distribution. Figure 6.9 and 6.10 show the averaged estimated and actual frontiers obtained for a value of $\epsilon = 0.05$ for the four risk measures.

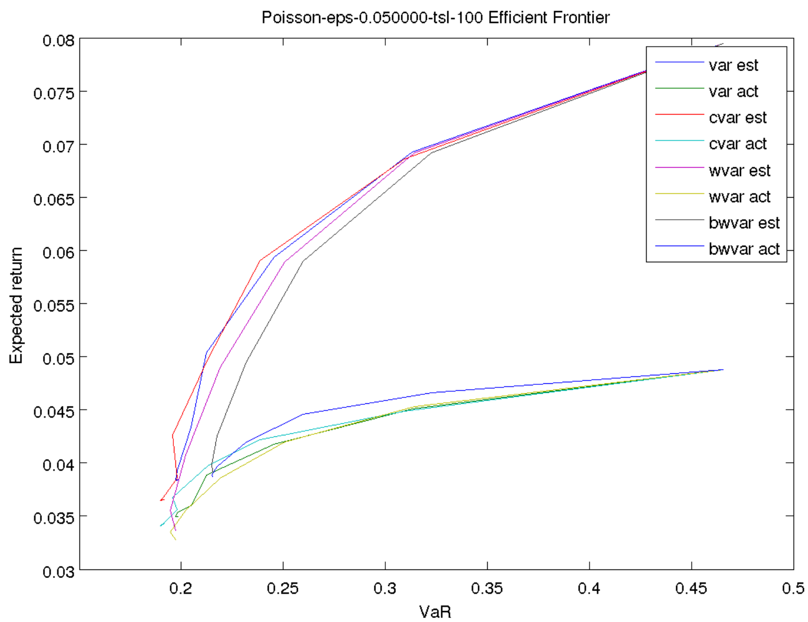


Figure 6.9: Average estimated and actual frontiers for the four portfolio optimisation problems, computed for samples generate using the multivariate Poisson distribution, a value of $\epsilon = 0.05$ and averaged over 200 condor runs.

A first observation is that the pairs of efficient frontiers (estimated and actual) are much further apart than in the case of normal distribution.

Comparing our risk models, we can see that in both cases, the average estimated frontier for the portfolio obtained using the box-type robust WPVaR as the risk measure is closer to its corresponding average actual frontier. This would confirm the results we obtained from our simulations over sample data using the multivariate normal distribution. For further insight, we show results for different values of ϵ for both distributions in appendix C section C.2.

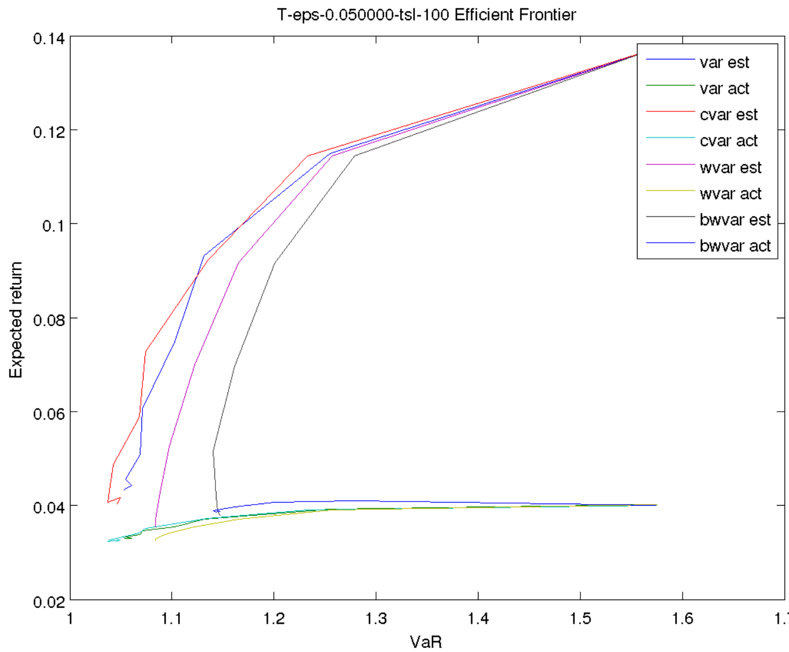


Figure 6.10: Average Estimated and Actual Frontiers for the four portfolio optimisation problems, computed for samples generated using the multivariate t-distribution, a value of $\epsilon = 0.05$ and averaged over 200 condor runs.

Again, these do confirm our observation that in general the averaged estimated and actual frontiers obtained from the portfolio which minimises the box-type robust WPVaR are closer to each other than the other two pairs of averaged frontiers.

As before, we will also look at the whisker plots for the estimated and actual returns of the four portfolios, to compare the variance in performance of the four. Figures 6.11 and 6.12 show whisker plots for the estimated returns of the four portfolios, for samples generated using the multivariate Poisson and multivariate t-distribution respectively, both for a value of $\epsilon = 0.05$.

As for the normal distribution, in this case, there is no significant difference between the four results.

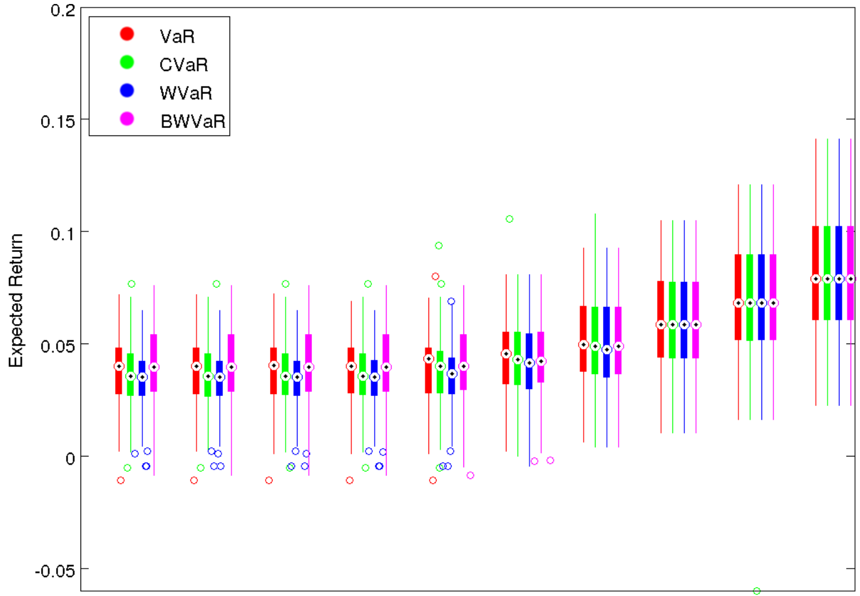


Figure 6.11: Whisker plot for the estimated returns computed for samples generated using the multivariate Poisson distribution and a value of $\epsilon = 0.05$.

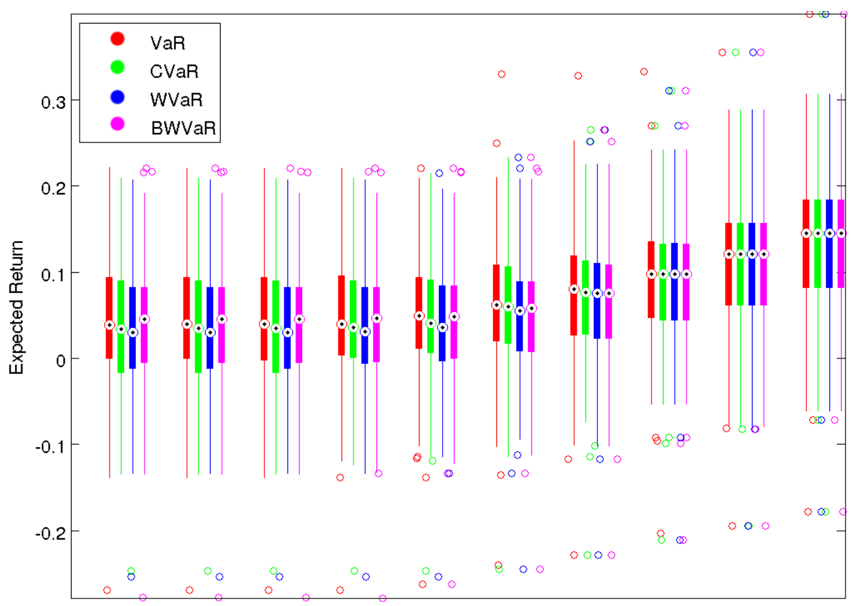


Figure 6.12: Whisker plot for the estimated returns computed for samples generated using the multivariate t-distribution and a value of $\epsilon = 0.05$.

However, figures 6.13 and 6.14, which show the same results for the actual returns, indicate quite a difference between the four portfolios. As we can see, the size of the box for the portfolio minimising WPVaR is somewhat smaller than the ones from minimising VaR or CVaR, for most of the minimum return values. The same can be observed about the distance between the outliers and the median. However, there is a significant difference between the variance in expected actual return for the box-type robust WPVaR model and all other three. This validates our conclusion that it is a more reliable risk estimate for skewed distributions as well. Although these differences decrease as the minimum return grows, neither the WPVaR nor the robust WPVaR models yield larger variance than the other two.

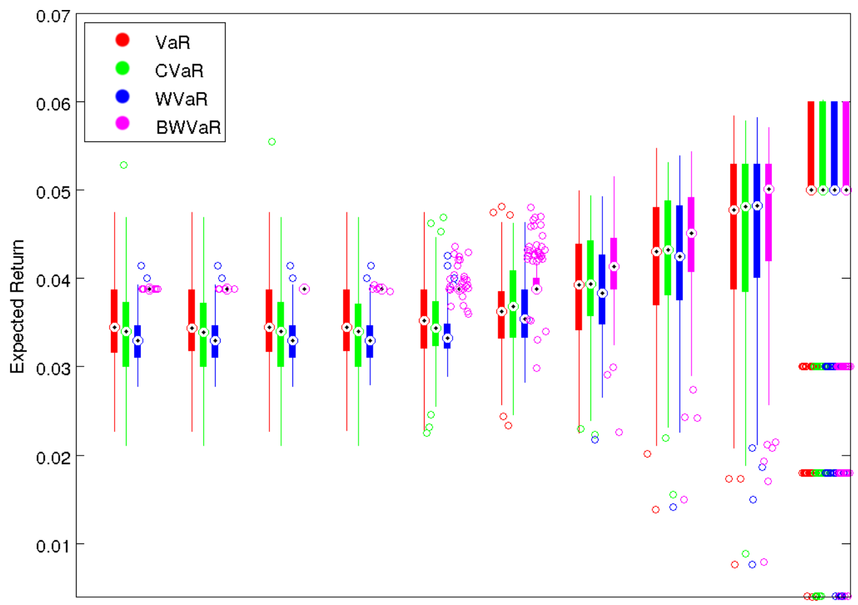


Figure 6.13: Whisker plot for the actual returns computed for samples generated using the multivariate Poisson distribution and a value of $\epsilon = 0.05$.

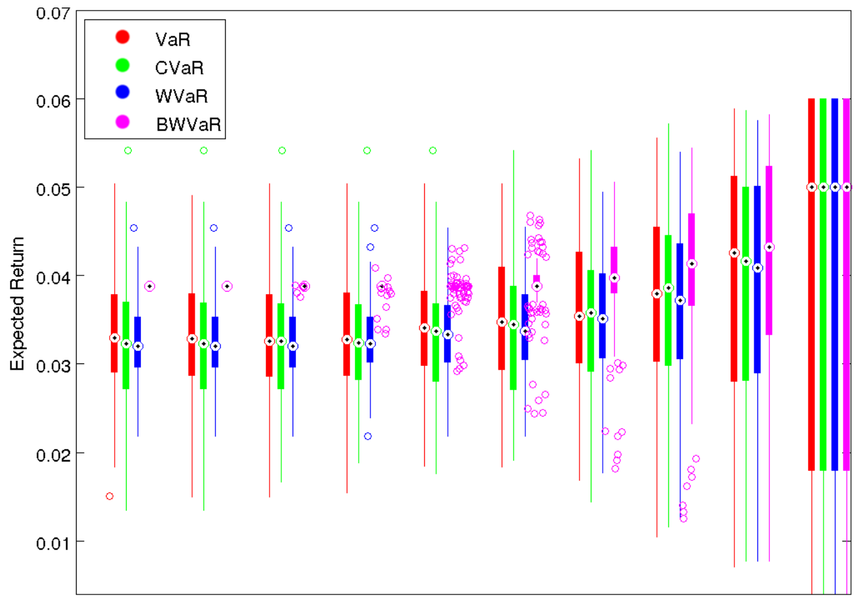


Figure 6.14: Whisker plot for the actual returns computed for samples generated using the multivariate t-distribution and a value of $\epsilon = 0.05$.

As for the normal distribution, another way to compare the four models is in terms of how much their portfolio choice varies over the different runs. Figures 6.15 and 6.16 show whisker plots for the weights of the five assets considered in our experiments, for the three different portfolio optimisation problems. The plots are created from experiments with data generated with the multivariate Poisson and t-distribution respectively, and for weights of the optimal portfolios obtained for the smallest expected minimum return.

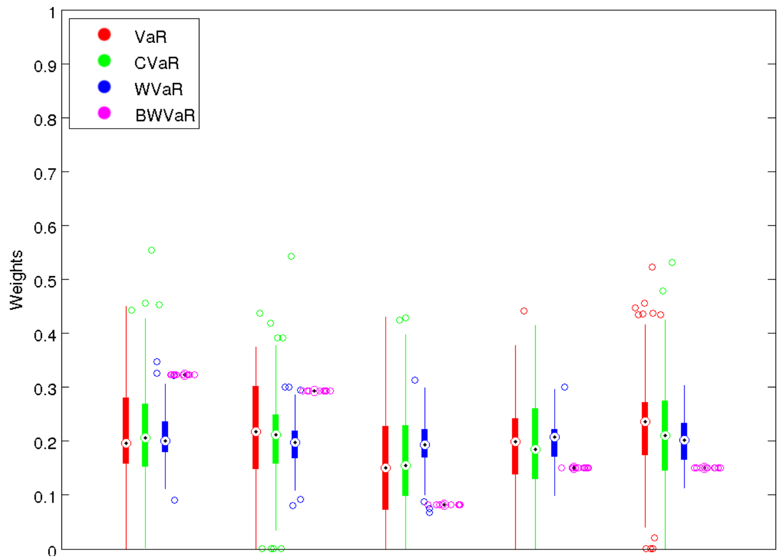


Figure 6.15: Whisker plot showing the variance in weights for the five assets over 200 different simulations. Results are shown for sample data generated using the multivariate Poisson distribution, for the smallest expected minimum return and for a value of $\epsilon = 0.05$.

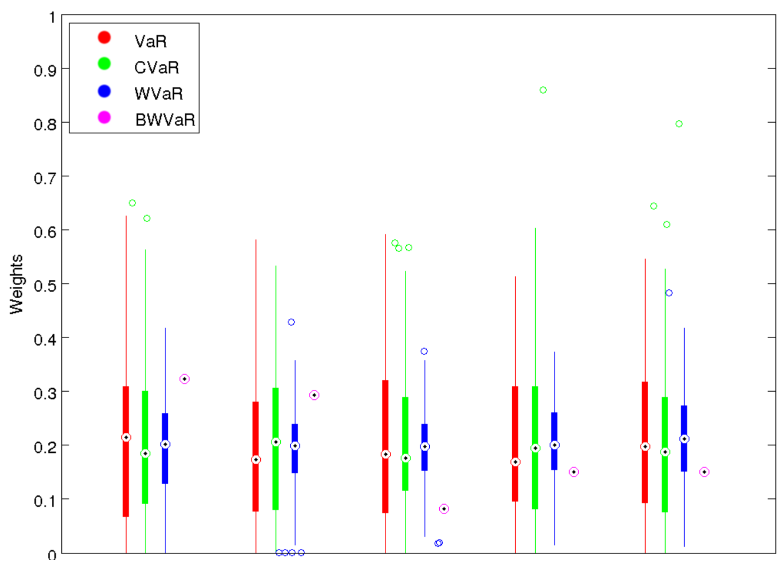


Figure 6.16: Whisker plot showing the variance in weights for the five assets over 200 different simulations. Results are shown for sample data generated using the multivariate t-distribution, for the smallest expected minimum return and for a value of $\epsilon = 0.05$.

In the case of the skewed distributions, the WPVaR model clearly yields optimal portfolios which vary a lot less than the optimal portfolios obtained using the VaR or CVaR risk measures. This suggests that our model is more robust than these two models on data following skewed distributions, choosing more similar portfolios between different simulation.

However, the box-type robust WPVaR model yields very stable results, for both distributions and compared to all other three risk models. This indicates the robust model is significantly more stable, yielding very similar optimal portfolios every run.

For a larger set of examples, we also show a pair of figures which display whisker plots for the weights for a larger expected minimum return. The results are shown in figures 6.17 and 6.18. Although the difference in size between boxes becomes smaller for a given asset, the portfolio obtained using the WPVaR model continues to be more stable than those from VaR or CVaR. also, the portfolio obtained using the box-type robust WPVaR continues to vary significantly less than all other three.

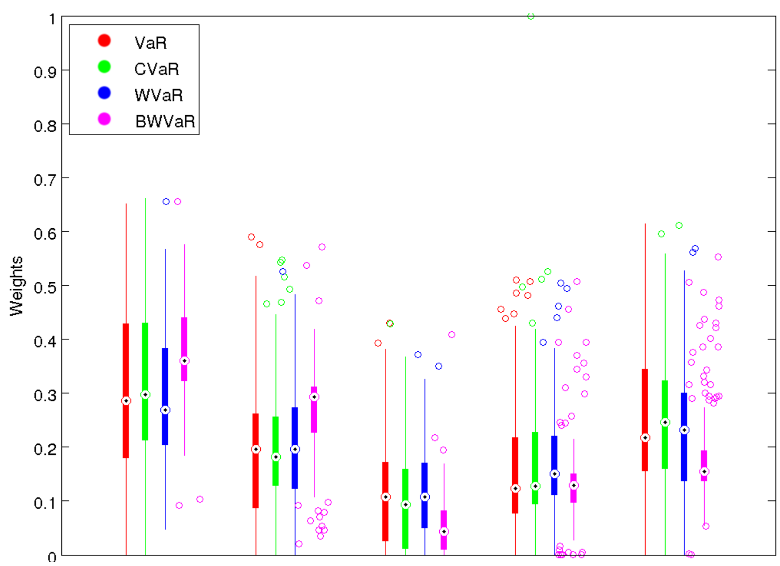


Figure 6.17: Whisker plot showing the variance in weights for the five assets over 200 different simulations. Results are shown for sample data generated using the multivariate Poisson distribution, for one of the larger expected minimum return and for a value of $\epsilon = 0.05$.

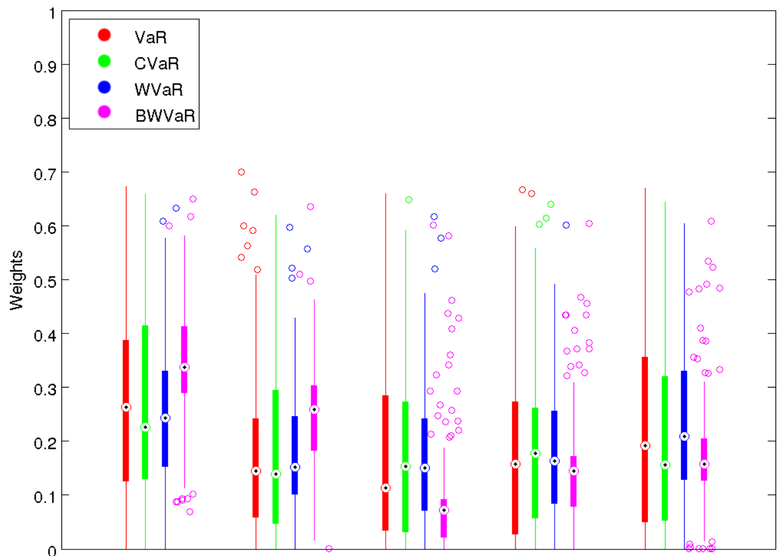


Figure 6.18: Whisker plot showing the variance in weights for the five assets over 200 different simulations. Results are shown for sample data generated using the multivariate t-distribution, for one of the larger expected minimum returns and for a value of $\epsilon = 0.05$.

As a final stage of our evaluation, we perform out-of-sample tests using historical financial data.

6.1.2 Out-of-Sample Statistics Using Historical Data

Although the experiments using sampled data provide some insight into how the WPVaR and the robust WPVaR perform as risk measures, it is also interesting to see how they perform on real historical financial data. To obtain historical data we used yahoo finance, as described in chapter 5. For our experiments, we chose to use a relatively small number of stocks, all from the same domain, since this gives a higher chance of correlation between their returns.

For each experiment, we use a set length of the time for which we consider historical returns. In all cases, we split the samples into training data and test data. This can be done in a number of ways, but in most experiments, we simply split the data into half, use the earliest half as training sample data and the latter one as test sample data. The "In-sample" results are performed on the training data, which is also used as input to the portfolio optimisation problems. The *"Out-of-sample"* results are however the ones we are interested in. These are obtained as follows: the portfolio is obtained using the training samples as input to the portfolio optimisation problems; these portfolios are then given the test sample data to obtain the required statistics.

Below we show experiments performed on four stocks related to the financial domain (since they are related to the same domain, they are likely to be correlated to each other). Different time periods are considered.

Using historical data between 1 January 2013 and 30 May 2013 we obtain the results shown in table 6.1. This is a rather short period of time and we can see there are very small differences between the results of the four approaches.

Statistic	Out-of-sample results			
	VaR	CVaR	WPVaR	box WPVaR
Mean	-0.002040	-0.002040	-0.002276	-0.002278
VaR	0.028412	0.028412	0.028597	0.028602
Std	0.011869	0.011869	0.012022	0.012030

Table 6.1: Table showing the out-of-sample statistics obtained using the four risk measures for a historical data sample from 1 January 2013 to 30 May 2013.

We then use longer and longer periods of time for our historical data. Tables 6.2, 6.3 and 6.4 show the results obtained for these periods. We can see that as the period gets bigger, we have some difference between the out-of-sample statistics of the different risk measures. Table 6.2 shows that the two WPVaR models yield a slightly higher mean return than the VaR model, while table 6.3 shows a 40% mean return increase compared to the VaR model and a 6% mean return increase compared to the CVaR one.

Statistic	Out-of-sample results			
	VaR	CVaR	WPVaR	box WPVaR
Mean	-0.000740	-0.000825	-0.000817	-0.000817
VaR	0.029687	0.031259	0.030045	0.030044
Std	0.019644	0.018562	0.018985	0.018985

Table 6.2: Table showing the out-of-sample statistics obtained using the four risk measures for a historical data sample from 1 January 2012 to 30 May 2013.

Statistic	Out-of-sample results			
	VaR	CVaR	WPVaR	box WPVaR
Mean	0.000624	0.000823	0.000874	0.000874
VaR	0.037727	0.038057	0.039560	0.039526
Std	0.023584	0.023425	0.023411	0.023412

Table 6.3: Table showing the out-of-sample statistics obtained using the four risk measures for a historical data sample from 1 January 2011 to 30 May 2013.

Statistic	Out-of-sample results			
	VaR	CVaR	WPVaR	box WPVaR
Mean	0.000691	0.000544	0.000615	0.000615
VaR	0.038306	0.039239	0.038220	0.038235
Std	0.031990	0.031231	0.030137	0.030138

Table 6.4: Table showing the out-of-sample statistics obtained using the four risk measures for a historical data sample from 1 January 2005 to 30 May 2013.

Although the results from the four portfolios are rather close, we can see the out-of-sample mean is in general slightly larger for the two WPVaR models. On the other hand, the VaR and Std tend to get smaller when using these portfolio optimisation problems. However, the results are unfortunately not clear enough to indicate the same significant difference between models as in the tests with simulated data.

To further compare the four portfolio optimisation problems, we also computed the empirical cdf of the four portfolios return. Figures 6.19 and 6.20 show these for the cases when we used historical data between 1 January 2013 and 30 May 2013 and between 1 January 2005 and 30 May 2013 respectively. However, we cannot see any clear stochastic dominance of one model over another, in either of the two figures. In fact, the cdfs get closer together as the time window increases.

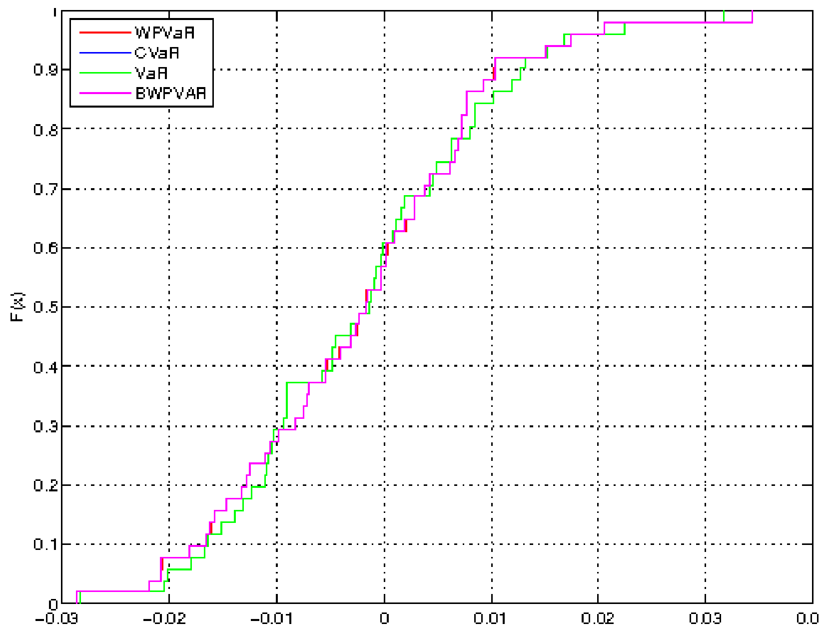


Figure 6.19: Empirical cdf for the out-of-sample portfolios return over the period between 1 January 2013 and 30 May 2013.

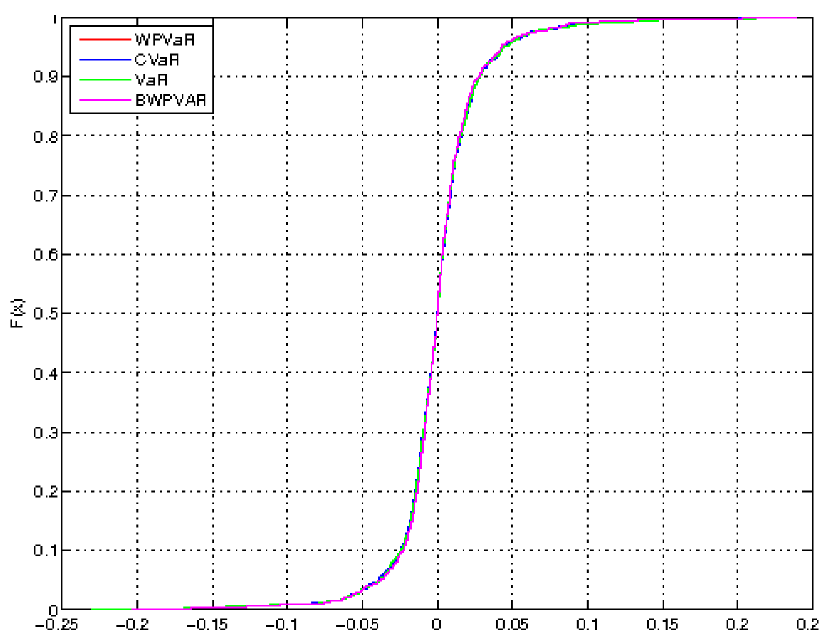


Figure 6.20: Empirical cdf for the out-of-sample portfolios return over the period between 1 January 2005 and 30 May 2013.

6.2 Results Summary

We have evaluated our WPVaR models by performing numerical tests, using both simulated and real historical sample data. We have also compared the results obtained from these models with results obtained from other risk measures: VaR and CVaR.

For simulated sample data tests, we used three different distributions: multivariate normal, multivariate Poisson and multivariate t. The first results, using the normally distributed samples, indicated a clear difference between the box-type robust WPVaR model and the other three. We have observed estimated and actual efficient frontiers closer to each other and to the true frontier and a small variance both in the expected actual return and asset weights over different runs. These observations were confirmed in the case of tests using skewed sample data, which are more realistic (since asset returns are often skewed in practice). These results indicate clearly that the box-type robust WPVaR model is more reliable than the others, providing a smaller expected actual return variance and very similar optimal portfolios between different simulations. This is important since in reality it may result in smaller transaction costs.

On the other hand, the tests we performed with historical financial data did not indicate such a significant difference between the four models. There was indeed a tendency that the WPVaR and the robust WPVaR models yield better out-of-sample results, but the results were not as clear. However, these results may change if the tests included complex financial instruments, namely options. This is expected since our WPVaR models specifically account for the dependency between an option's value and its underlier's return.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The initial part of this project is to derive a Worst-case Value-at-Risk formulation which incorporates the non-linear relationship between the options' return and their underlying asset. This was achieved for European options expiring beyond the investment horizon, by approximating the options' payoff function by an arbitrary number of linear functions. This lead to the SDP and SOCP formulations of the WPVaR model, presented in theorems 3.1 and 3.2. As part of our derivations, we also developed a formal proof for dualizing integral constraints of an optimisation problem.

The next step was to extend this model, in order to account not only for uncertainty in the actual distribution of asset returns, but also for uncertainty in the distribution parameters — mean and covariance matrix of asset returns. This is a very important optimisation since in practice investors do not know the real distribution parameters, but can merely estimate them from historical data. However, these estimates are often not accurate enough and errors in their estimated values can result in large errors of risk measure.

Hence, we developed two extensions to our model, using two different approaches. First, we applied a box-type robust optimisation, which assumes the mean and the covariance matrix are known to be between some boundaries. Second, we used the partitioned statistics approach which has the capability to account for asymmetry in the asset returns distribution. This is an important contribution since real assets returns have skewed distributions. Thus, we formulated and proved two new theorems 4.1 and 4.2, which defined the extended models as tractable SDPs. These represented the *main novelty* of our project — providing a distributionally robust portfolio optimisation model which combines both asymmetric asset returns distributions and options which expire beyond the investment horizon.

To validate, test and compare our models, among them and with other risk measures — VaR and CVaR, we performed a number of numerical tests, using both simulated and real historical sample data.

For simulated sample data tests, we used the multivariate normal, multivariate Poisson and multivariate t-distribution. In each case, we performed a large number of simulations and used the results to perform different analysis: we generated the average estimated and actual efficient frontiers and compared these for the four models and with the true one; we studied the variance in expected estimated and actual return for the different risk measures; we looked at the variance in asset weights for the four models.

The results using the normally distributed samples indicated a clear difference between the box-type robust WPVaR model and the other three. We have observed estimated and actual efficient frontiers closer to each other and to the true frontier than for the other three risk measures. Although the robust model did not exhibit a smaller expected estimated return variance, it did show significant smaller variance for the expected actual return. These results indicate a better approximation of the expected return when using the box-type robust WPVaR model as opposed to the other three. This is important since it indicates the robust WPVaR model as a more stable risk measure. Finally, the robust model also showed a significantly smaller variance in assets weights over different runs. Although the difference between this and the variance obtained from the other models decreases as the minimum expected return increases, the robust WPVaR model never yields optimal portfolios with larger variance.

The above observations were confirmed when performing tests using skewed sample data. This is more realistic, since asset returns are often skewed in practice. Although the pairs of average estimated and actual frontiers are much further apart than in the case of normal distribution, the results show clearly that the ones obtained using the box-type robust WPVaR model are considerably closer to each other than the other pairs. Also, the variance in expected actual returns is very small in the case for skewed distributions for the robust model, which indicates a much better chance of good performance. These results clearly indicate that the box-type robust WPVaR model is more reliable than the others, providing a smaller expected actual return variance and very similar optimal portfolios between different simulations.

These results show the effect that accounting for uncertainty has on the results obtained using a certain risk measure. In this work, we did not consider uncertainty in the distribution only, but also in its parameters. This lead to our box-type robust formulation of our WPVaR model, which turned out to perform considerably better than our initial WPVaR model and the other risk measures considered.

On the other hand, the tests we performed with historical financial data did not indicate such a significant difference between the four models. Although there was a tendency that the WPVaR and the box-type robust WPVaR models provide better out-of-sample results, these were not as clear as the previous ones. However, these results may change if the tests included complex financial instruments, namely options. This is expected since our WPVaR models specifically account for the dependency between an option's value and its underlier's return.

We now present some of the future work that can be carried on in relation to our project:

7.2 Future Work

- implementation of the second robust WPVaR model defined in theorem 4.2, which uses the partitioned statistics approach to take into account distributional asymmetry. This can be then evaluated together with the other risk measures considered and comparison results can be obtained;

- more extensive numerical experiments with simulated sample data can be carried out, studying the behaviour of the WPVaR models in the presence of options (whose price can be computed using the Black Scholes formula) expiring at or after the investment horizon. Furthermore, different restrictions can be imposed, to study the behaviour of our models (e.g. every stock must be accompanied by a call/put option etc.);

- using a cross-validation technique for performing the tests using historical data, when computing the out-of-sample statistics. This would provide a better evaluation of the performance of the models given the sample data;

- investigate whether the minimum volume ellipsoid as described in [24] can be used to describe uncertainty in the distribution parameters; this could be incorporated into the work presented in [25], where a framework for distributionally robust optimisation is presented in order to generalise several approaches;

- investigate the application of methods similar to the ones presented in [26] for trying to bring the estimated and actual frontiers closer.

Bibliography

[1] H. Markowitz, "Portfolio selection*," *The journal of finance*, vol. 7, no. 1, pp. 77–91, 1952.

[2] P. Jorion, *Value at Risk: The New Benchmark for Managing Financial Risck*. McGraw-Hill, 2001.

[3] L. El Ghaoui, M. Oks, and F. Oustry, "Worst-case value-at-risk and robust portfolio optimization: A conic programming approach," *Operations Research*, vol. 51, no. 4, pp. 543–556, 2003.

[4] S. Zymler, D. Kuhn, and B. Rustem, "Worst-case value at risk of nonlinear portfolios," *Management Science*, vol. 59, no. 1, pp. 172–188, 2013.

[5] D. Luenberger, *Investment science.* New York: Oxford Univ. Press, 1998.

[6] T. J. Linsmeier and N. D. Pearson, "Risk measurement: An introduction to value at risk," 1996.

[7] R. T. Rockafellar and S. Uryasev, "Optimization of conditional value-at-risk," *Journal of risk*, vol. 2, pp. 21–42, 2000.

[8] ——, "Conditional value-at-risk for general loss distributions," *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[9] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust optimization.* Princeton University Press, 2009.

[10] K. Natarajan, M. Sim, and J. Uichanco, "Tractable robust expected utility and risk models for portfolio optimization," *Mathematical Finance*, vol. 20, no. 4, pp. 695–731, 2010.

[11] J. Goh, K. G. Lim, M. Sim, and W. Zhang, "Portfolio value-at-risk optimization for asymmetrically distributed asset returns," *European Journal of Operational Research*, 2012.

[12] D. Bertsimas and M. Sim, "The price of robustness," *Operations research*, vol. 52, no. 1, pp. 35–53, 2004.

[13] S. Zymler, B. Rustem, and D. Kuhn, "Robust portfolio optimization with derivative insurance guarantees," *European Journal of Operational Research*, vol. 210, no. 2, pp. 410–424, 2011.

[14] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[15] P. Kall and S. W. Wallace, *Stochastic programming.* John Wiley and Sons Ltd, 1994.

[16] S. Zymler, D. Kuhn, and B. Rustem, "Distributionally robust joint chance constraints with second-order moment information," *Mathematical Programming*, vol. 137, no. 1-2, pp. 167–198, 2013.

[17] I. Pólik and T. Terlaky, "A survey of the s-lemma," *SIAM review*, vol. 49, no. 3, pp. 371–418, 2007.

[18] J. Lfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://users.isy.liu.se/johanl/yalmip

[19] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[20] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "Sdpt3a matlab software package for semidefinite programming, version 1.3," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 545–581, 1999.

[21] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathematical programming*, vol. 95, no. 2, pp. 189–217, 2003.

[22] A. A. Gaivoronski and G. Pflug, "Value-at-risk in portfolio optimization: properties and computational approach," *Journal of Risk*, vol. 7, no. 2, pp. 1–31, 2005.

[23] "HTCondor website." [Online]. Available: http://research.cs.wisc.edu/htcondor/description.html

[24] L. Davies, "The asymptotics of rousseeuw's minimum volume ellipsoid estimator," *The Annals of Statistics*, pp. 1828–1843, 1992.

[25] W. Wiesemann, D. Kuhn, and M. Sim, "Distributionally robust convex optimization," 2013.

[26] S. Ceria and R. A. Stubbs, "Incorporating estimation errors into portfolio selection: Robust portfolio construction," *Journal of Asset Management*, vol. 7, no. 2, pp. 109–127, 2006.

Appendix A

Proofs

Proof 1

We will show that the constraint $\alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{\Gamma}\boldsymbol{\xi} \geq 0 \quad \forall\boldsymbol{\xi} \in \mathbb{R}^n$ is equivalent to $\begin{bmatrix} \Gamma & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta}^T & \alpha \end{bmatrix} \succcurlyeq 0$.

Starting from the definition of positive semidefinite matrix, we obtain:

$$
\begin{aligned}
& \begin{bmatrix} \Gamma & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta}^T & \alpha \end{bmatrix} \succcurlyeq 0 \\
\iff \quad & \begin{pmatrix} \boldsymbol{\xi} \\ \psi \end{pmatrix}^T \begin{bmatrix} \Gamma & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta}^T & \alpha \end{bmatrix} \begin{pmatrix} \boldsymbol{\xi} \\ \psi \end{pmatrix} \geq 0 \;\; \forall\boldsymbol{\xi} \in \mathbb{R}^n \;\; \forall\psi \in \mathbb{R} \\
\iff \quad & \begin{pmatrix} \boldsymbol{\xi} \\ 1 \end{pmatrix}^T \begin{bmatrix} \Gamma & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta}^T & \alpha \end{bmatrix} \begin{pmatrix} \boldsymbol{\xi} \\ 1 \end{pmatrix} \geq 0 \;\; \forall\boldsymbol{\xi} \in \mathbb{R}^n \\
\iff \quad & \alpha + \boldsymbol{\beta}^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{\Gamma}\boldsymbol{\xi} \geq 0 \;\; \forall\boldsymbol{\xi} \in \mathbb{R}^n
\end{aligned}
$$

Proof 2

We will prove the following equivalence:

$$\langle \boldsymbol{\Omega}, \mathbf{M} \rangle = \alpha + \boldsymbol{\beta}^T\boldsymbol{\mu} + \langle \boldsymbol{\Gamma}, \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T \rangle$$

where $\boldsymbol{\Omega}$ is the second-order moment matrix and $\mathbf{M} = \begin{bmatrix} \Gamma & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta}^T & \alpha \end{bmatrix}$.

Starting from the definition of $\mathbf{\Omega}$ and $\mathbf{M}$ and then using the definition of $\langle\cdot\rangle$:

$$\begin{aligned}\langle \mathbf{\Omega}, \mathbf{M}\rangle &= \langle \begin{bmatrix} \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T & \boldsymbol{\mu} \\ \boldsymbol{\mu}^T & 1 \end{bmatrix}, \begin{bmatrix} \Gamma & \frac{1}{2}\boldsymbol{\beta} \\ \frac{1}{2}\boldsymbol{\beta}^T & \alpha \end{bmatrix}\rangle = \langle \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T\rangle + \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\mu} + \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\beta} + \alpha \\ &= \alpha + \boldsymbol{\beta}^T\boldsymbol{\mu} + \langle \mathbf{\Gamma}, \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T\rangle\end{aligned}$$

Appendix B

HTCondor

Job specification file:

```
universe = vanilla

executable = /homes/cac409/workspace/rpo/condor/runCondorSim.sh

output = out/sim-$(Process).out
error  = out/sim-$(Process).err

log = sim.log

arguments = -n -m -p

queue 500
```

Appendix C

Efficient Frontiers

C.1 Multivariate Normal Distribution

Figures C.1, C.2 and C.3 show the average estimated and actual efficient frontiers obtained from 200 HTCondor simulations, for all four portfolio optimisation problems: minimising VaR, CVaR, WPVaR or box-type robust WPVaR. They each show the results obtained for a different value of ϵ, 0.01, 0.1 and 0.2 respectively.

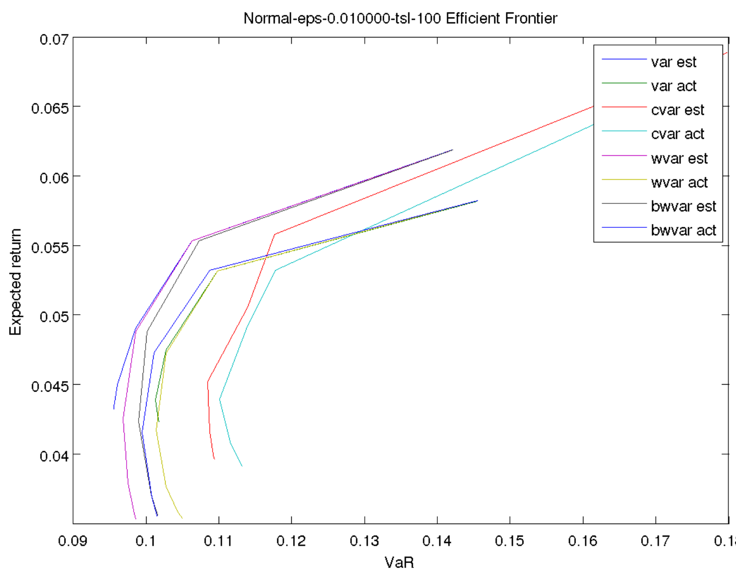


Figure C.1: Average estimated and actual frontiers for the four portfolio optimisation problems, computed for a value of $\epsilon = 0.01$, averaged over 200 condor runs.

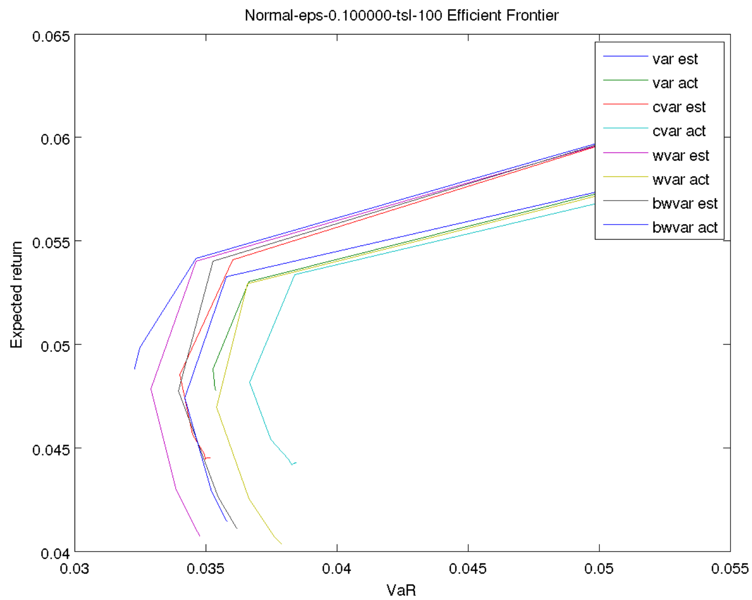


Figure C.2: Average estimated and actual frontiers for the four portfolio optimisation problems, computed for a value of $\epsilon = 0.1$, averaged over 200 condor runs.

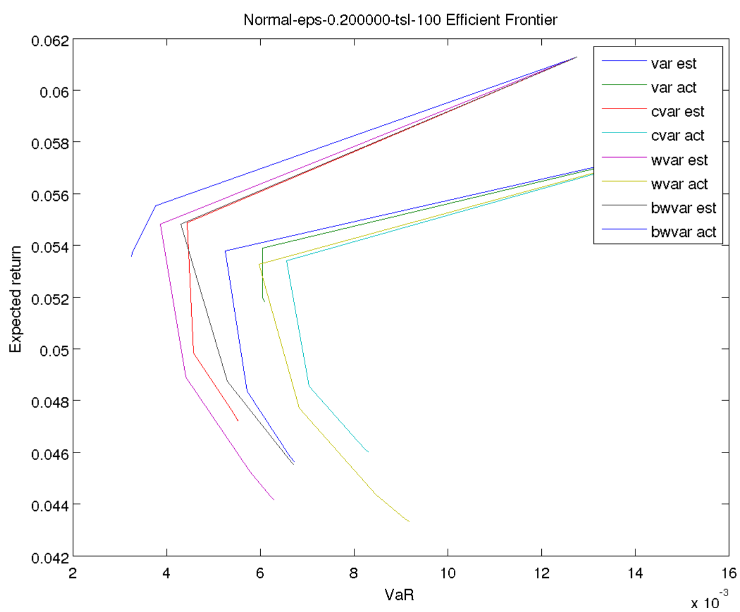
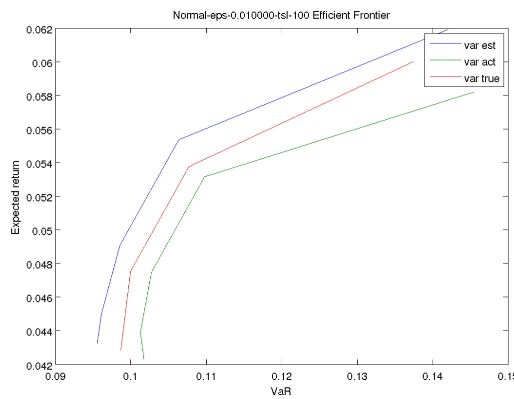
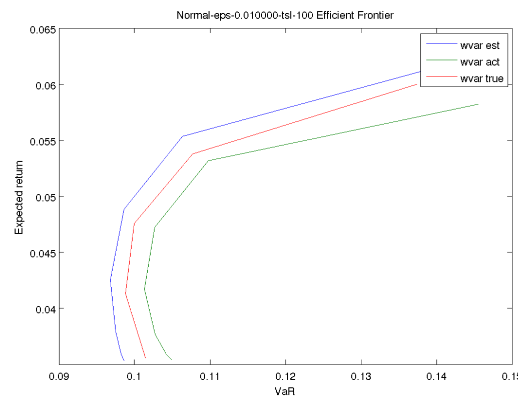


Figure C.3: Average estimated and actual frontiers for the four portfolio optimisation problems, computed for a value of $\epsilon = 0.2$, averaged over 200 condor runs.

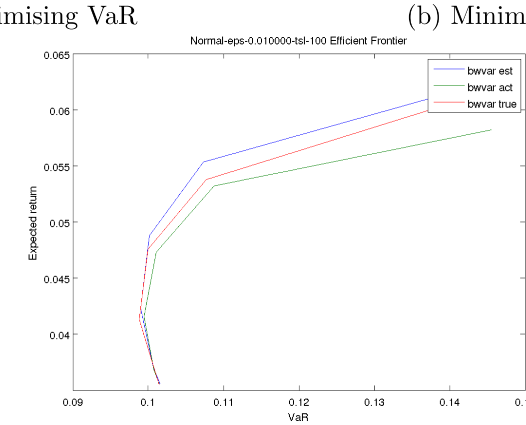
Figures C.4 and C.5 show comparisons between the true and the average estimated and actual frontiers. They each show results for the three portfolios obtained minimising VaR, WPVaR or box-type robust WPVaR, for the two more common values of ϵ, 0.01 and 0.1 respectively.



(a) Minimising VaR

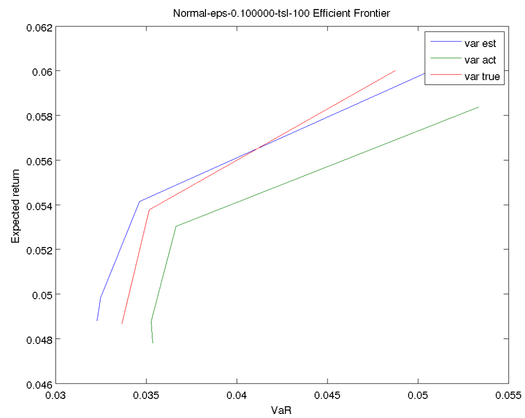


(b) Minimising WPVaR

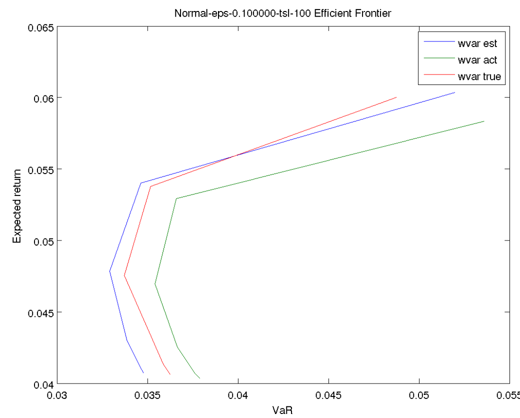


(c) Minimising box-type robust WPVaR

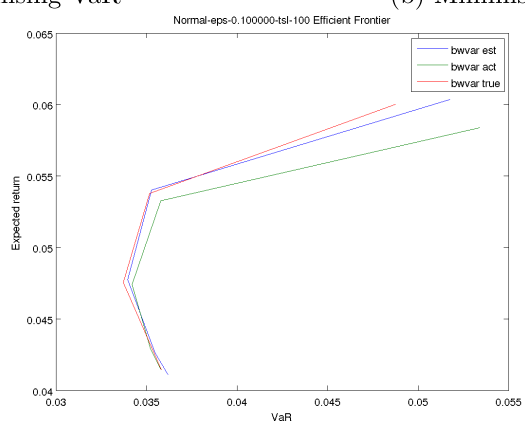
Figure C.4: The three frontiers obtained by minimising VaR, WP-VaR or box-type robust WPVaR with $\epsilon = 0.01$ on multivariate normal generated samples.



(a) Minimising VaR



(b) Minimising WPVaR



(c) Minimising box-type robust WPVaR

Figure C.5: The three frontiers obtained by minimising VaR, WVaR and box-type robust WPVaR with $\epsilon = 0.1$ on multivariate normal generated samples.

C.2 Multivariate Skewed Distributions

Figures C.6 and C.7 show the average estimated and actual efficient frontiers obtained from 200 HTCondor simulations, using sample data generated using the multivariate Poisson distribution and the multivariate t-distribution respectively. They display results for a value of $\epsilon = 0.01$, for all four portfolio optimisation problems: minimising VaR, CVaR, WPVaR or box-type robust WPVaR.

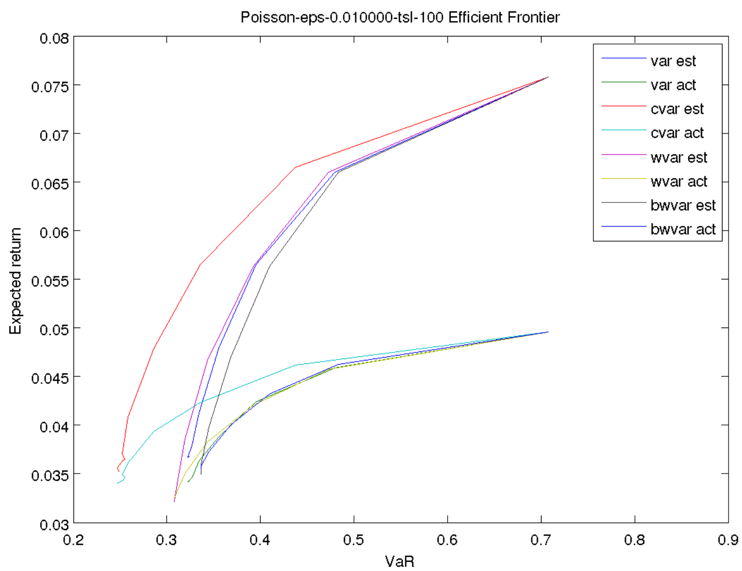


Figure C.6: Average estimated and actual frontiers, computed on sample data generated using the multivariate Poisson distribution, over 200 condor runs. Results are shown for a value of $\epsilon = 0.01$.

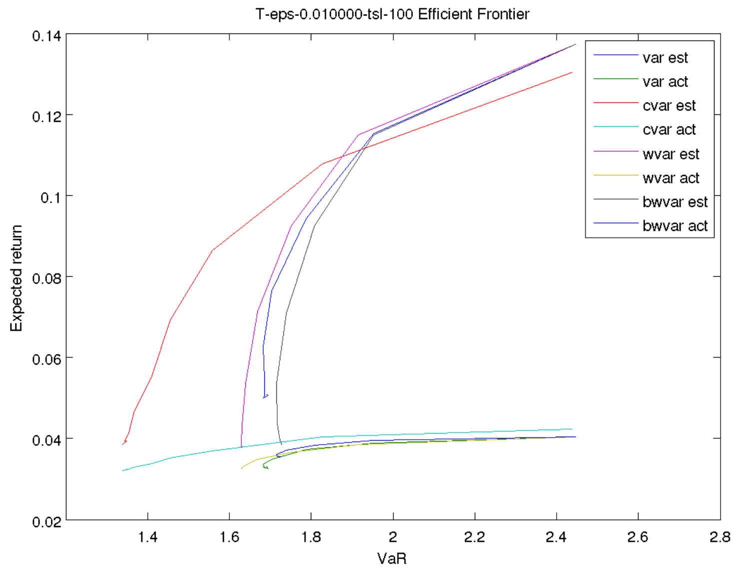


Figure C.7: Average Estimated and Actual Frontiers, computed on sample data generated using the multivariate t-distribution, over 200 condor runs. Results are shown for a value of $\epsilon = 0.01$.

Figures C.8 and C.9 show the average estimated and actual efficient frontiers obtained using the sample data from the same two distributions, but for a larger value of $\epsilon = 0.1$.

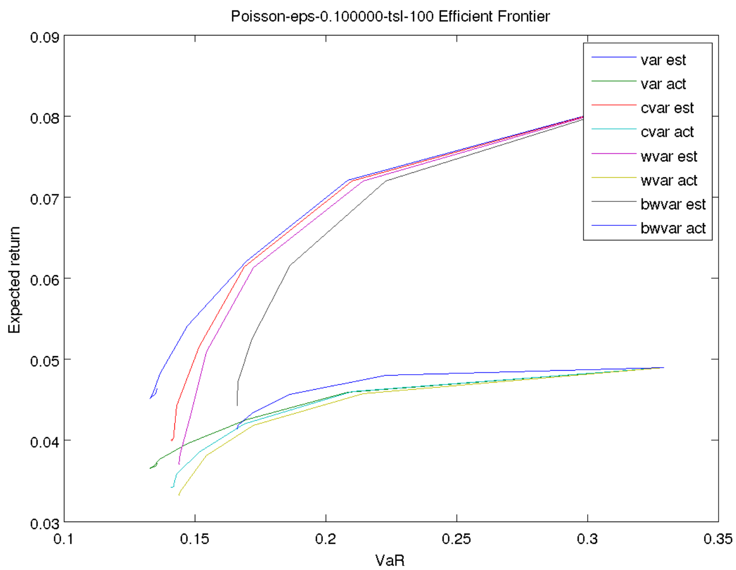


Figure C.8: Average estimated and actual frontiers, computed on sample data generated using the multivariate Poisson distribution, over 200 condor runs. Results are shown for a value of $\epsilon = 0.1$.

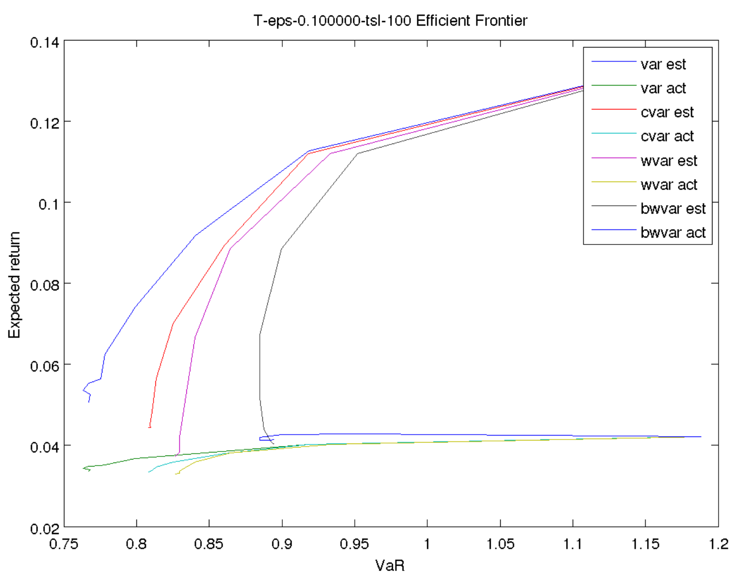


Figure C.9: Average estimated and actual frontiers, computed on sample data generated using the multivariate t-distribution, over 200 condor runs. Results are shown for a value of $\epsilon = 0.1$.