MASTER'S THESIS

Regularised Estimation of the Precision Matrix for Financial Data
Regularisation Through Portfolio Optimisation

LINN ENGSTRÖM

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2019

Thesis for the Degree of Master of Science

Regularised Estimation of the Precision Matrix for Financial Data
Regularisation Through Portfolio Optimisation

Linn Engström

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology and University of Gothenburg
SE – 412 96 Gothenburg, Sweden
Gothenburg, May 2019

Regularised Estimation of the Precision Matrix for Financial Data
Regularisation Through Portfolio Optimisation
LINN ENGSTRÖM

Supervisor: Tobias Rydén, Lynx Asset Management
Examiner: Rebecka Jörnsten, Department of Mathematical Sciences

Master's Thesis 2019
Department of Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Department of Mathematical Sciences
Gothenburg, Sweden 2019

Regularised Estimation of the Precision Matrix for Financial Data
Regularisation Through Portfolio Optimisation
LINN ENGSTRÖM
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

A fundamental aspect of quantitative finance is portfolio optimisation, a field of mathematics that is very much governed by the Modern portfolio theory introduced by Harry Markowitz in 1952. The goal is to maximise the expected return for a given pre-determined level of risk. An optimal portfolio solving the problem is directly proportional to the inverse covariance matrix – the precision matrix – of the returns. Since the covariance structure in between markets is unknown, so is the precision matrix. Therefore, it must be estimated out of historical data, something that is not easily done due to the bad conditioning of the problem. There is thus a need for regularisation.

This Master's thesis proposes and derives a new estimator of the precision matrix, intending to minimise the expected distance between a pre-determined target level of risk and the actual risk of a Markowitz optimal portfolio. Since the proposed estimator belongs to the class of rotation-invariant estimators, minimisation is carried out by direct manipulation of its eigenvalues. Optimal parameters of a spectral mapping are found based on historical data. The mapping, defined by the computed optimal parameters, is then used for regularising the sample precision matrix of future data. The performance of the new estimator is compared with a simple l_2-penalised sample estimator and with two l_1- and l_2-penalised maximum likelihood estimators. An estimator is considered to perform well if the risk of its corresponding Markowitz portfolio is close to the target risk, given that the estimator doesn't underestimate the covariance out-of-sample. The results reveal that the choice of spectral mapping is of great importance for the strategy to be successful. For one of the investigated mappings the risk of the corresponding portfolio is indeed close to the target risk, even though the estimator seems to perform less well out-of-sample than some of the reference estimators. Further investigations of other mappings should be carried out.

Keywords: High-dimensional statistics, rotation-invariant estimators, Modern portfolio theory.

Acknowledgements

I would like to thank my supervisor Tobias Rydén, who came up with the idea of the risk-targeting estimator and helped derive it. Without his guidance this project would have become far less interesting.

Linn Engström, Stockholm, May 2019

Contents

List of Figures

List of Tables

1
Introduction

Lynx Asset Management is a Swedish hedge fund manager based in Stockholm. The company manages the funds Lynx, Lynx Dynamic and Lynx Active Balanced Fund and is one of the world's largest stakeholders within the field of algorithmic asset management. The investment decisions are made by mathematical models, aiming to predict trends in the markets. The company mainly invests in futures based on fixed income-products, equities, commodities and foreign exchange.[1] This Master's thesis project was carried out on behalf of Lynx.

1.1 Background: the precision matrix matters

One fundamental aspect of quantitative finance is portfolio optimisation, a field that is much governed by the Modern portfolio theory introduced by Harry Markowitz in 1952 – see [2]. The return of a portfolio $w_t = (w_{t,1}, ..., w_{t,N})'$ consisting of N instruments at time t is given by

$$R_t^w = \sum_{i=1}^{N} w_{t,i} R_{t,i} \quad ,$$

where $R_{t,i}$ is the return of the i^{th} instrument. The weight $w_{t,i}$ can be interpreted as the relative amount of risk put into the i^{th} instrument.
According to Modern portfolio theory, w_t should be chosen such that the total risk (that is, the standard deviation of R_t^w) is minimised subject to some constraint on the expected return $\mu = (\mu_1, ..., \mu_N)'$. It can be shown (see Appendix A) that the optimal portfolio w_t^* satisfies

$$w_t^* \propto \Sigma_R^{-1} \mu \quad , \tag{1.1}$$

where Σ_R is the covariance matrix of $R_{t,1}, ..., R_{t,N}$. More details regarding the mathematical framework will be given in section 2.2.
From eq. (1.1) it is clear that the inverse covariance matrix – *the precision matrix* – plays an important role when computing the optimal portfolio w_t^*. In reality the actual problem to be solved is more complicated than the one described above, taking additional constraints into consideration. The precision matrix Σ_R^{-1} is still crucial when computing the optimal portfolio though.

1.1.1 Regularised estimation of the precision matrix

One problem with the strategy described above is the fact that the covariance between the risk-adjusted returns $R_{t,1}, ..., R_{t,N}$ is unknown. Therefore, so is Σ_R^{-1}. One

must therefore construct an estimator of Σ_R^{-1} out of the market data available. For N instruments, this means that the number of parameters that need to be estimated is of order N^2 since $\Sigma_R^{-1} \in \mathbf{R}^{N \times N}$.
Let the number of observations be denoted T. It is clear that the quality of the estimator benefits from the fraction $\frac{N^2}{T}$ being small, since this would imply that the number of observations is larger than the number of unknowns. Since the dependencies in between markets might change over time one cannot use market data that is too old though, this fact putting an upper bound on T. Typically Lynx stores market data on a day- to day level (see section 3.4 for a full description of the data available) and only uses data that is less than one year old. Assuming that a trading year consists of approximately 250 trading days yields a fraction $\frac{N^2}{T} \approx 36 > 1$ when taking $N = 95$ instruments into account. This indicates that the problem of estimating Σ_R^{-1} is a an ill-posed one, for what reason it clearly needs to be regularised.

1.2 Purpose

The goal of this project was to construct a regularised estimator of the precision matrix, taking Lynx' market data as input. The estimator should mimic the investment process, in the sense that its corresponding Markowitz-portfolio should end up at a pre-determined level of risk. This would be an attractive property, since in reality one is typically restricted by the amount of risk one can accept prior to investing. A method for evaluating the performance of the derived estimator was also to be suggested.

1.3 Methods and working procedure

The project was initialised by the implementation of an l_p-penalised maximum likelihood estimator for $p = 1, 2$, derived under the assumption that $Z_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}_N(\mu, \Sigma)$ $\forall t$. See section 2.5.2. The maximum likelihood estimator was chosen as a simple reference estimator due to the fact that it is a very well-known standard one. Various types of penalties have been extensively studied and described by the literature during the past years. Regularisation through penalisation can thus be viewed as a standard procedure. A penalised maximum likelihood estimator therefore serves as a natural starting point and reference estimator.
The next step was to theoretically derive a new estimator with the desired properties. When doing so, some inspiration was found in [3]. A few important differences in strategy compared to [3] was made though.
Firstly, the ansatz from [3] has been made so as to

1. Choose a level of return λ.
2. Construct a portfolio that maximises the reward-to-risk ratio, given that the total return equals λ.
3. Minimise the out-of-sample risk for this portfolio.

In this project, the chosen strategy was instead to

1. Choose a level of risk ξ.

2. Construct a portfolio that maximises the expected return, given that the total risk equals ξ.
3. Minimise the expected mean-squared error between the actual risk and ξ for this portfolio.

It should be noted that the two strategies indirectly implies very different things about the investment procedure. The strategy described by [3] suggests that any level of risk could be accepted, given that the total return equals λ. The strategy used in this project however emphasises the importance of staying within the risk budget, something that might be more appropriate for a real-life setting.
Secondly, it was in [3] proven that their chosen loss function converges almost surely to a non-stochastic limit as $\frac{N}{T} \xrightarrow[T\to\infty]{} c$ for some $c \equiv const$, given that some assumptions on the covariance matrix and its estimator(s) hold. This asymptotic framework was neglected throughout this project, for what reason the expected value of the quantities in question was examined.

The analytic derivations of the estimators resulted in optimisation problems, where the optimal estimator would be the minimiser of the loss function in question. These optimisation problems needed to be solved numerically. Depending on the nature of the given problem, suitable algorithms were chosen and implemented in Matlab. Note that no packages or toolboxes were employed throughout the project – every algorithm was implemented from scratch in order to ensure maximal control and flexibility. More information will be given in section 3.2.

1.4 Delimitations

One important delimitation that had to be set up in an early stage of the project was the decision to focus only on static models, i.e. estimators that need to be updated manually when new market data is available. This decision was made based on the fact that the simple model currently used at Lynx is a static one, and on the aim to focus mainly on other aspects of the problem.

1.5 Structure of the report

The report starts off with an outline of the underlying theory in chapter 2. The notation used throughout the report is introduced and more details are given regarding the portfolio optimisation that was briefly described earlier. Some information regarding different ways of regularising a problem is also given. Then a rather long section follows, deriving analytic expressions for the estimators investigated.
Chapter 3 includes a description of the market data provided by Lynx, as well as information regarding the pre-processing of the data. It also includes information regarding the environment set up for training and testing the estimators based on historical data, as well as a description of the quantities chosen for evaluating the performance of the estimators. The chosen optimisation algorithms are described and derived for the given problems. The spectral mappings used for regularisation of the risk-targeting estimator are also listed in this chapter.

The results are presented in chapter 4. The chapter is partitioned into two main parts: the first part gives some intermediate results regarding the characteristics of the problems investigated. The final results – i.e. the performance of the estimators – are given in the second part of chapter 4. The results are then discussed in chapter 5.

2
Theory

In this chapter some notation will be introduced as well as a few important underlying assumptions. In section 2.3 some more details will be given regarding Modern portfolio theory, which will later be used in order to evaluate the performance of the estimators – see chapter 3. As mentioned in the introduction of this report, the cornerstone of this theory is that portfolio optimisation is a trade-off between maximising the profit and minimising the risk, and that the two aspects must be considered together and not separately. It is shown that the optimal portfolio is highly dependent of the inverse covariance matrix Σ^{-1}, which therefore needs to be estimated out of historical data. Some strategies for regularising the problem are then given. Finally, three regularised estimators are derived in section 2.5.2 and section 2.5.3. The first one is a very simple penalised sample estimator that is known to perform surprisingly well. The second one is the penalised maximum likelihood estimator, intended to work as a reference estimator for comparison of performance. The last one is tailor-made, constructed with the intention of mimicking the investment process. It is a rotation-invariant estimator where regularisation is applied by manipulation of the eigenvalues of Σ^{-1}.

2.1 Three basic derivatives

Before presenting the mathematical framework used during the project, a few words regarding the derivatives used are given. A derivative is a financial product consisting of a contract, where the parties agree to buy or sell an asset at a predetermined time in the future. Its value is derived from the value of the underlying asset. The terms of the contract can be very varied and depending on the type of contract, their value and return need to be modelled differently. Below follows a very brief overview of three main derivatives: options, futures and forwards. As mentioned in chapter 1 Lynx mainly invests in future contracts. A few instruments are of forward-type though. Lynx does not trade in options, but they are still mentioned for the sake of completeness.

2.1.1 Forwards

A forward contract is an agreement in between two parties to buy or sell an asset for a certain price at a certain time in the future. At the agreed time, maturity, the transaction must take place. Forward contracts are traded over-the-counter, meaning that each contract is tailor-made to suit the needs of the two parties.

The parties trade directly with each other, without any daily settlements: only one payment at maturity occurs. The value at time t_0 of of such a forward contract is

$$V_0 = P_0 e^{rT} \quad ,$$

where T is the time to maturity, r is the risk-free rate and P_0 is the spotprice of the underlying asset.[4] A few instruments used at Lynx are forwards.

2.1.2 Futures

Future contracts have the same main characteristics as forwards contracts but are traded at exchanges, meaning that each party trades against the exchange. This reduces the counterparty risk, i.e. the risk of the counterparty not being able to fulfil its obligations. Typically the exchange reduces the risk by having the parties successively adjust to the price developments of the underlying asset by daily settlement. The value of a future contract is thus difficult to evaluate and one is instead interested in the accumulated return.[5] Almost all instruments used at Lynx are futures.

2.1.3 Options

An option is a contract in between two parties that allows the buyer of the option to buy or sell the underlying asset at maturity for a predetermined price. However, the option need not be exercised in case the holder of the option would prefer not to. Options are both traded at the exchange and over-the-counter. There exists a wide range of different types of options but only a few of them has closed-form expressions that allows for easy computation of their respective value.[6] Lynx does not trade in options, for what reason no details regarding the value of different options will be given here.

2.2 About the setting

The underlying mathematical framework used throughout the report will be introduced below, some inspiration regarding the notation has been found in [6]. Let $\mathcal{M}$ be a market model where N assets $A_1, ..., A_N$ are traded, characterised by the probability space $(\Omega, \mathcal{F}, \mathbf{P})$. Let $P_{t,i}(\omega)$ be the price of asset A_i at time t under the market scenario $\omega \in \Omega$ and let $w_{t,i}(\omega)$ be the relative number of shares held in asset A_i at time t. In accordance with the general convention within financial mathematics, it is assumed that at time t_0 one's position w in the markets is set as one enters a number of contracts in $A_1, ..., A_N$. It is assumed that all the contracts expire at maturity at time $t = t_0 + T$ for some $T > 0$ and that the position w is kept throughout the period $t \in [t_0, t_0 + T]$.
When dealing with forwards and futures, the 1-day-return (from now on referred to as "the return") of the i^{th} asset A_i at day t is given by

$$R_{t,i}(\omega) = P_{t,i}(\omega) - P_{t-1,i}(\omega) \quad , \tag{2.1}$$

while the expected return is

$$\mathbf{E}[R_{t,i}(\omega)] = m_i \quad .$$

The risk of asset A_i is defined as the standard deviation of its return,

$$\sqrt{Var[R_{t,i}(\omega)]} = \sigma_{t,i} \quad .$$

Similarly, the return of the portfolio w at time t is given by

$$R_t^w(\omega) = \sum_{i=1}^{N} R_{t,i}(\omega) w_i \quad .$$

In order to be able to perform easy comparison in between instruments and markets, the return need to be normalised somehow.[5] Therefore, define the risk-adjusted return as

$$Z_{t,i}(\omega) = \frac{R_{t,i}(\omega)}{\sigma_{t,i}} \quad . \tag{2.2}$$

The total risk-adjusted return of the portfolio w is given by

$$Z_t^w(\omega) = \sum_{i=1}^{N} Z_{t,i}(\omega) w_i = \sum_{i=1}^{N} \frac{R_{t,i}(\omega)}{\sigma_{t,i}} w_i \quad .$$

The expected total risk-adjusted return and standard deviation of the total risk-adjusted return of w are

$$\mathbf{E}[Z_t^w(\omega)] = \mu' w \quad \text{and} \quad \sqrt{Var[Z_t^w(\omega)]} = \sqrt{w' \Sigma w}$$

respectively, where the elements of Σ are defined as the covariance in between the risk-adjusted returns. The elements μ_i of μ are the expected risk-adjusted returns:

$$[\Sigma]_{ij} = Cov(Z_{t,i}(\omega), Z_{t,j}(\omega)) \quad \text{and} \quad \mu_i = \mathbf{E}[Z_{t,i}(\omega)] \quad .$$

Note the following relation between the expected risk-adjusted return and the expected return:

$$\mu_i = \mathbf{E}[Z_{t,i}(\omega)] = \mathbf{E}\left[\frac{R_{t,i}(\omega)}{\sigma_{t,i}}\right] = \frac{1}{\sigma_{t,i}} \mathbf{E}[R_{t,i}(\omega)] = \frac{m_i}{\sigma_{t,i}} \quad .$$

For the variances of the risk-adjusted returns hold

$$Var(Z_{t,i}) = Var\left(\frac{R_{t,i}}{\sigma_{t,i}}\right) = \frac{1}{\sigma_{t,i}^2} Var(R_{t,i}) = 1 \quad ,$$

i.e. the risk-adjusted return has unit variance in theory. This is a characterising property of the risk-adjusted return.
As for the covariances, fact is that the covariance matrix Σ of the risk-adjusted returns is actually the (Pearson) correlation matrix of the returns:

$$\begin{aligned}[\Sigma]_{ij} &= Cov(Z_{t,i}(\omega), Z_{t,j}(\omega)) = Cov\left(\frac{R_{t,i}(\omega)}{\sigma_{t,i}}, \frac{R_{t,j}(\omega)}{\sigma_{t,j}}\right) = \\ &= \frac{Cov(R_{t,i}(\omega), R_{t,j}j(\omega))}{\sigma_{t,i}\sigma_{t,j}} = \frac{Cov(R_{t,i}(\omega), R_{t,j}(\omega))}{\sqrt{Var(R_{t,i}(\omega)) Var(R_{t,i}(\omega))}} \quad .\end{aligned}$$

Note that the volatility $\sigma_{t,i}$ used in the above expressions is an unknown quantity that must be estimated. Thus, the true risk-adjusted return $Z_{t,i}$ is unfeasible. Let any estimator of $\sigma_{t,i}$ be denoted $\hat{\sigma}_{t,i}$ and let $\hat{Z}_{t,i} = R_{t,i}/\hat{\sigma}_{t,i}$ be the corresponding estimator of $Z_{t,i}$. It is clear that the closer $\hat{\sigma}_{t,i}$ is to $\sigma_{t,i}$, the closer is the sample-variance of $\hat{Z}_{t,i}$ to unity.
It was the estimated risk-adjusted returns $\hat{Z}$ that served as the fundamental input data for any algorithm or computation used during the project – see chapter 3 for more information regarding the data preparation. From now on, the state ω will be dropped from the notation in order to not clutter the equations. It should still be kept in mind though that the returns (and any quantity derived from them) are always stochastic quantities.

2.3 Modern portfolio theory

As briefly mentioned previously, a fundamental aspect of quantitative finance is the trade-off between a large risk and a large expected return. This insight was first provided by Harry Markowitz in 1952 – see [2]. He emphasised that one should not seek to maximise the expected return on its own. Instead, for every level of expected return there is a portfolio w such that the total risk is minimised. Using the previously introduced notation, the problem can be formulated as an optimisation problem:

$$\begin{aligned} \underset{w}{\text{minimise}} \quad & w'\Sigma w \\ \text{subject to} \quad & w'\mu = \lambda \quad . \end{aligned} \tag{2.3}$$

Here, the parameter λ is the level of total expected risk-adjusted return.
An alternative approach is to maximise the expected risk-adjusted return, given that the level of risk is held fixed. The corresponding optimisation problem then reads

$$\begin{aligned} \underset{w}{\text{maximise}} \quad & w'\mu \\ \text{subject to} \quad & w'\Sigma w = \xi^2 \quad . \end{aligned} \tag{2.4}$$

Here, the parameter ξ is the risk. This is the form of the problem that was considered throughout the project, since this version allows for choosing the level of adjusted risk prior to investing. The optimal solution w^* of eq. (2.4) was used for evaluating the performance of any estimator of Σ^{-1} – see section 3.1 for information regarding this part.
The problems in eq. (2.3) and eq. (2.4) can easily be solved analytically – see Lemma A.0.1 and Lemma A.0.2 in appendix A. Their optimal solutions are $w^* = \frac{\lambda}{\mu'\Sigma^{-1}\mu}\Sigma^{-1}\mu$ and $w^* = \frac{\xi}{\sqrt{\mu'\Sigma^{-1}\mu}}\Sigma^{-1}\mu$ respectively. The solutions coincide when $\xi = \frac{\lambda}{\sqrt{\mu'\Sigma^{-1}\mu}}$. It is clear that the optimal solutions of both eq. (2.3) and eq. (2.4) satisfy

$$w^* \propto \Sigma^{-1}\mu \quad , \tag{2.5}$$

for what reason the inverse covariance matrix – the *precision matrix* – of the risk-adjusted returns plays a fundamental part in portfolio optimisation. From now on, let the precision matrix be denoted Θ, i.e. let

$$\Theta \coloneqq \Sigma^{-1} \quad .$$

Since the true covariance structure of the risk-adjusted returns is an unkown quantity, so is the precision matrix. It therefore needs to be estimated out of historical market data. Let any estimator of Σ^{-1} be denoted $\hat{\Theta}$.

2.4 Ill-conditioned problems need to be regularised

It was in section 2.3 argued that an estimator of the precision matrix Θ needs to be constructed out of historical market data. The most inuitive choice would perhaps be to try to invert the sample covariance matrix, that is to let

$$\hat{\Theta}^{\text{sample}} := \left(\frac{1}{T-1} \sum_{t=1}^{T} (Z_t - \bar{Z})'(Z_t - \bar{Z}) \right)^{-1}$$

where $\bar{Z} = \frac{1}{T} \sum_{t=1}^{T} Z_t$ and $Z_t = (Z_{t,1}, ..., Z_{t,N})'$. Unfortunately, the sample covariance matrix is typically very ill-conditioned when the number of observations T[1] is of the same order of magnitude as the number of variables N. As an example, the sample covariance matrix computed for the risk-adjusted returns of $N = 95$ instruments using $T = 250$ observations (corresponding to approximately one year of trading) results in the sample covariance matrix having a condition number of the order 10^4. Therefore, inversion of the sample covariance matrix is not a suitable strategy. Other estimators of Θ therefore have to be considered. Let any regularised estimator be denoted $\tilde{\Theta}$.

2.4.1 Regularisation by adding a penalty term

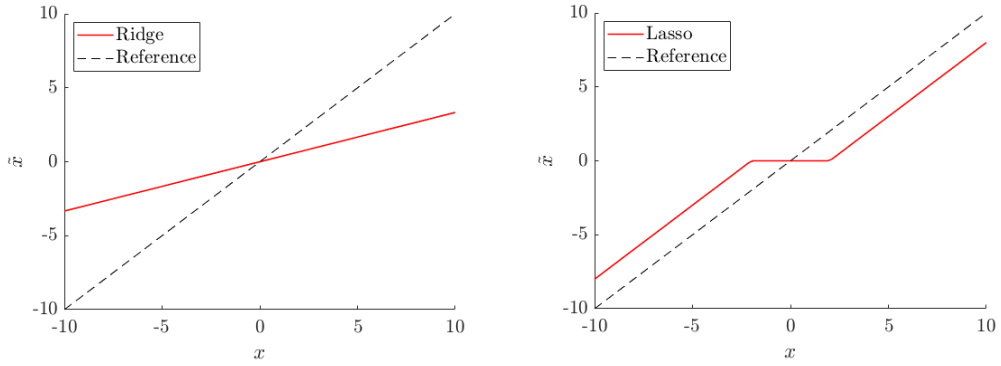
Common alternative estimators are usually either based on linear regression (for example, see [7]) or on maximisation of the assumed likelihood.[8] Both strategies result in a set-up where one is trying to estimate the order of N^2 unknowns out of T observations. Whenever $N^2 > T$, the problem is ill-posed[2]. In order to ensure the existence of a unique solution, the number of unknowns has to be shrunk. This can be achieved by the usage of a sparse model, i.e. a model where small components are shrunk towards zero. Shrinkage is obtained by adding a penalty term to the given objective function:

$$\tilde{\Theta} = \underset{\hat{\Theta}}{\text{argmin}} \quad f(\hat{\Theta}) + \gamma ||\hat{\Theta}||_p^p \quad .$$

In the above example the original objective function is f, the unregularised estimator is $\hat{\Theta}$ and the regularised estimator is $\tilde{\Theta}$. γ is the penalty parameter and p indicates what type of norm is being used. In a regression setting, f is typically the sum of squared errors. In case a maximum-likelihood strategy is being used, f is typically the negative likelihood of the data.[10]

[1] As mentioned briefly in chapter 1, one cannot use historical data that is "too old" since the covariance structure might change over time. This restriction puts an upper bound on the number of observations T that can be used.

[2] According to Hadamard's definition, a problem is ill-posed if no unique solution exists that depends continuously on the initial data.[9]



(a) Ridge **(b)** Lasso

Figure 2.1: Shrinkage of the estimator by penalisation. The figure displays how unregularised coefficients x are mapped to regularised ones $\tilde{x}$ when adding an l_1- (lasso) or l_2- penalty (ridge) for the case when f is the mean-squared error. It is clear that while the ridge penalty imposes an even amount of shrinkage to the coefficients, the lasso specifically cuts off all coefficients whose absolute values are smaller than the parameter γ. The dashed line is intended to mark out the line $\tilde{x} = x$ as a comparison. The figure is inspired by [10] and was generated using $\gamma = 2$.

By altering the value of γ the amount of shrinkage can be modified: it is clear that in case γ is small, any optimisation algorithm would focus on minimising $f(\hat{\Theta})$ rather than $||\hat{\Theta}||_p^p$. Thus one would obtain $\tilde{\Theta} \approx \hat{\Theta}$, which indicates that the amount of regularisation is very slight. In case γ is big, the $||\hat{\Theta}||_p^p$-term would be prioritised over $f(\hat{\Theta})$ which would yield a large amount of regularisation.

There are two penalties that are the by far most commonly used: $p = 1$ and $p = 2$. The case when $p = 2$ is referred to as ridge regularisation and results in all coefficients $[\hat{\Theta}]_{ij}$ being proportionally shrunk. The case when $p = 1$ is referred to as "the lasso-penalty" when applied to the regression context. In the one-dimensional case the lasso performs a soft thresholding, i.e. all coefficients whose absolute value is smaller than the parameter γ are put to zero. Large coefficients are not shrunk at all.[10] See fig. 2.1 for a visualisation. In the multivariate context the situation is more complicated.

There exists a large number of variations of the penalty parameter, obtained by choosing or combining different values of p. A simple modification of the cases $p = 1, 2$ is to only penalise the off-diagonal elements. This choice is motivated by the fact that the elements on the main diagonal need not be shrunk. This is due to the fact that $[\Theta]_{ij} = 0$ if and only if the elements Z_i and Z_j are conditionally independent given all other elements – see for example the interpretation of the hedge-regression due to Stevens in [7]. From now on, let the off-diagonal l_p-penalty be denoted $||\cdot||_{p^*}^p$, as opposed to the "usual" l_p-penalty (including also the diagonal)

$||\cdot||_p^p$:

$$||X||_{p^*}^p \coloneqq \sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{M} |X|_{ij}^p \quad \text{as opposed to}$$

$$||X||_p^p \coloneqq \sum_{i=1}^{N}\sum_{j=1}^{M} |X|_{ij}^p \quad \text{for any matrix } X \in \mathbf{R}^{M\times N} \quad .$$

It was the off-diagonal l_1- and l_2 penalties that were considered during this project.

2.4.2 Regularisation by manipulation of the eigenvalues

It was in the preceding section explained how regularisation and model selection can be performed through penalisation of the off-diagonal elements of $\hat{\Theta}$. For a special class of matrices, regularisation can also be carried out through direct manipulation of the spectrum.
Any *normal* matrix $A \in \mathbf{R}^{N\times N}$ can be decomposed as

$$A = U'\Lambda U$$

for U unitary and Λ diagonal with the eigenvalues of A on its main diagonal. Let the eigenvalues of A be denoted $a_1, ..., a_N$. For such a matrix A, the condition number is given by the fraction

$$\kappa(A) = \left|\frac{\max_i\{a_i\}}{\min_i\{a_i\}}\right| \quad .$$

It is from the above expression clear that two things might contribute to a normal matrix having a large condition number: either is its largest eigenvalue very large, its smallest eigenvalue very small or a combination of the two. The condition number of such a matrix can thus be improved by reducing the largest eigenvalues and enlarging the smallest eigenvalues of A. Having this in mind, a regularised estimator can be constructed by simply replacing the original eigenvalues with regularised ones. The technique is referred to as "spectral regularisation".

Let $g: \quad \mathbf{R}^+ \to [a, b]$ for some $0 < a < b < \infty$ be a regularising mapping, defined by m parameters $p_1, ..., p_m$. Assume that g is differentiable with respect to $p_1, ..., p_m$ and decompose the normal matrix $\hat{\Theta}$ as

$$\hat{\Theta} = HDH' \tag{2.6}$$

where H is a unitary matrix and D is a diagonal matrix with the eigenvalues $d_1, ..., d_N$ of $\hat{\Theta}$ on the main diagonal. Now define a set of regularised eigenvalues $\{\tilde{d}_i\}_{i=1}^N$ as

$$\tilde{d}_i \coloneqq g(d_i) \quad \forall i = 1, ..., N$$

where d_i is the i^{th} eigenvalue of $\hat{\Theta}$. Out of these regularised eigenvalues $\tilde{d}_i$, construct a diagonal matrix $\tilde{D}$ such that

$$[\tilde{D}]_{ij} \coloneqq \begin{cases} \tilde{d}_i & \text{for } i = j \\ 0 & \text{else} \end{cases} \quad .$$

Out of $\tilde{D}$ as given above and $H: \quad \hat{\Theta} = HDH'$, define the regularised estimator $\tilde{\Theta}$ as

$$\tilde{\Theta} \coloneqq H\tilde{D}H' \quad .$$

It now remains to choose a suitable mapping g. When doing so, it should be kept in mind that the overall purpose of g is to enhance small arguments while shrinking large ones. Arguments of intermediate size should be kept approximately the way they are. Taking this into account, it seems that an S-shaped mapping might be a suitable choice for g. See chapter 3 for an overview of mappings that were tried out during the project.

2.5 Derivation of the estimators

In the following sections the estimators investigated are derived and explained. The first one is perhaps the simplest possible regularised estimator, constructed by penalising the sample covariance matrix before inverting it. The second one is a penalised maximum likelihood estimator, derived under the assumption that the observations are i.i.d. multivariate normally distributed. Two types of penalties are considered: the off-diagonal l_1- and l_2 penalties. The third estimator considered is perhaps the most interesting one, derived for the first time during this project. It is intended to minimise the distance between the target risk level and the actual risk level of a Markowitz optimal portfolio solving the problem of eq. (2.4). The two first estimators are intended to serve as references for comparison with the third one.

2.5.1 Simple reference estimator

It was in the introduction of section 2.4 noted that the most intuitive estimator of the precision matrix would be to simply invert the sample covariance matrix, but that this strategy cannot be used due to bad conditioning. This problem can be quite easily fixed though, by the addition of a small regularising extra term prior to inversion.
Let the sample covariance matrix be denoted $\hat{\Sigma}^{\text{sample}}$, i.e. let

$$\hat{\Sigma}^{\text{sample}} \coloneqq \frac{1}{T-1}\sum_{t=1}^{T}(Z_t - \bar{Z})'(Z_t - \bar{Z})$$

where as usual $\bar{Z} = \frac{1}{T}\sum_{t=1}^{T} Z_t$ and $Z_t = (Z_{t,1}, ..., Z_{t,N})'$. It was previously noted that $\hat{\Sigma}^{\text{sample}}$ typically has a condition number of the order 10^4 when $T = 250$ and $N = 95$. Now define a regularised sample covariance matrix $\tilde{\Sigma}_c$ as

$$\tilde{\Sigma}_c \coloneqq c\hat{\Sigma}^{\text{sample}} + (1-c)diag\left(\hat{\Sigma}^{\text{sample}}\right)$$

for some $0 < c < 1$. It is clear that the above is equivalent to

$$[\tilde{\Sigma}_c]_{ij} \coloneqq \begin{cases} [\hat{\Sigma}^{\text{sample}}]_{ii} & \text{if } i = j \\ c \cdot [\hat{\Sigma}^{\text{sample}}]_{ij} & \text{else,} \end{cases}$$

which implies that only the off-diagonal elements are shrunk by a factor c: the elements are shrunk proportionally. Recall from section 2.4.1 that l_2-regularisation also shrinks all coefficients proportionally. Thus, $\tilde{\Sigma}_c$ can be viewed as a very simple l_2-regularised version of the sample precision matrix. Since $\tilde{\Sigma}_c$ is much better conditioned than $\hat{\Sigma}^{\text{sample}}$ its inverse can be computed directly, resulting in a precision-matrix estimator

$$\tilde{\Theta}^{\text{simple}} = \tilde{\Sigma}_c^{-1} \quad . \tag{2.7}$$

It will be referred to as "the penalised sample estimator" from now on and was used for comparison when evaluating the results of the tailor-made estimator derived in section 2.5.3.

2.5.2 Penalised maximum likelihood estimator

As a simple starting point for trying out the methods of chapter 3, a classical estimator was considered: the penalised maximum likelihood estimator. Assuming that each observation at time t of the risk-adjusted returns $Z_t = (Z_{t,1}, ..., Z_{t,N})' \overset{\text{i.i.d.}}{\sim} \mathcal{N}_N(\mu, \Sigma)$, the joint likelihood L of independent realisations $Z = (Z_1, ..., Z_T)$ is given by

$$L(Z) = \frac{1}{(2\pi)^{TN/2}|\Sigma|^{T/2}} \prod_{t=1}^{T} \exp\left[-\frac{1}{2}(Z_t - \mu)'\Sigma^{-1}(Z_t - \mu)\right] \quad .$$

Since the exponential function is monotone, maximisation of the likelihood $L(Z)$ is equivalent to maximising its logarithm. Define the log-likelihood

$$l(Z) = NT\log(2\pi) - 2\log(L) = T\log|\Sigma| + \sum_{t=1}^{T}(Z_t - \mu)'\,\Sigma^{-1}\,(Z_t - \mu)$$

and note that

$$\log|\Sigma| = \log|\Theta^{-1}| = \log\left(|\Theta|^{-1}\right) = -\log|\Theta| \quad .$$

Let the sample covariance matrix be denoted S, i.e. let

$$S \coloneqq \frac{1}{T}\sum_{t=1}^{T}(Z_t - \mu)'(Z_t - \mu)$$

and rewrite the sum over t as

$$\frac{1}{T}\sum_{t=1}^{T}(Z_t - \mu)'\Sigma^{-1}(Z_t - \mu) = tr\left(\Sigma^{-1}\sum_{t=1}^{T}(Z_t - \mu)'(Z_t - \mu)\right) = Ttr\left(\Theta S\right) \quad ,$$

where as usual $\Theta = \Sigma^{-1}$. Plug the new expressions into l and obtain

$$l(Z) = Ttr(\Theta S) - T\log|\Theta| \quad .$$

Maximisation of L is equivalent to maximising $-l$. Thus the maximum likelihood estimator of Θ can be obtained from

$$\hat{\Theta}^{\text{MLE}} = \underset{\Theta}{\text{argmax}} \quad \log|\Theta| - tr(\Theta S) \quad . \tag{2.8}$$

In order to improve sparsity of the off-diagonal elements (see section 2.4.1), [11] proposed adding an l_1-penalty to the off-diagonal elements of the problem in eq. (2.8). The resulting quasi-maximum likelihood estimator is thus given by

$$\tilde{\Theta}^{\text{QML1}} = \underset{\Theta}{\text{argmax}} \quad \log|\Theta| - tr(\Theta S) - \gamma||\Theta||_{1^*} \quad , \tag{2.9}$$

where γ is the penalty parameter. The proposed QML-estimator of eq. (2.9) can easily be modified by replacement of the off-diagonal l_1-penalty to some other penalty. In this project, the off-diagonal l_2-penalty was also considered:

$$\tilde{\Theta}^{\text{QML2}} = \underset{\Theta}{\text{argmax}} \quad \log|\Theta| - tr(\Theta S) - \gamma||\Theta||^2_{2^*} \quad .$$

2.5.3 Tailormade risk-targeting estimator

The in section 2.5.2 described quasi-maximum likelihood estimator is a well-known estimator that serves as a good starting point and reference estimator. Its derivation does not captive any of the characteristics of the investment process, neither does it enhance any of the properties of a "good" estimator[3] though. The following section derives and describes the construction of a tailormade rotation-invariant estimator that is intended to mimic the investment process. The ansatz made by [3] serves as inspiration.
Since the derivation of the new estimator is rather long, it has been split into several sections. In section 2.5.3.1 the optimal weights of a Markowitz portfolio targeting a risk level ξ is recapitulated from section 2.3. In section 2.5.3.2 these weights are plugged into a loss function, which has been chosen so as to minimise the expected value of the distance between the actual risk and the target risk. Since the format of the resulting loss function is not very well suited for optimising over, the function has to be rewritten. This is performed in section 2.5.3.4, where the result is gradually built up over three lemmas. The full formula is given in Theorem 2.1. It is then clear that the resulting loss function is highly dependent of the eigenvalues of $\hat{\Theta}$. In section 2.5.3.5 it is explained how the loss function is to be minimised by regularisation of these eigenvalues. The optimal eigenvalues will give rise to a regularised, risk-targeting estimator of Θ.

2.5.3.1 Finding optimal weights targeting a risk level

As described in section 2.3, the cornerstone of Modern portfolio theory is to choose a portfolio w such that an optimal trade-off between minimal risk ($\sqrt{w'\Sigma w}$) and maximal expected return ($w'\mu$) is obtained. Two optimisation problems were presented – see eq. (2.3) and eq. (2.4) – the first minimising the risk for every level of expected return, the second maximising the expected return for every level of risk. In reality the investment process is typically carried out by initially choosing a level of risk that is considered as being acceptable. Only thereafter is the expected return maximised. Having this in mind, it is clear that it is the second version of

[3]See section 3.1 for a description of what is considered as being a "good" estimator.

the problem that mimics the investment procedure the best. It is repeated below for the reader's convenience.

$$\begin{aligned} \underset{w}{\text{maximise}} \quad & w'\mu \\ \text{subject to} \quad & w'\Sigma w = \xi^2 \end{aligned} \quad ,$$

where the level of risk is ξ. The optimal solution is

$$w_\xi^* = \frac{\xi}{\sqrt{\mu'\Sigma^{-1}\mu}}\Sigma^{-1}\mu \quad ,$$

see Appendix A for proof. The estimator of w_ξ^* is obtained through replacement of Σ^{-1} to its estimator $\hat{\Theta}$,

$$\hat{w}_\xi^* = \frac{\xi}{\sqrt{\mu'\hat{\Theta}\mu}}\hat{\Theta}\mu \quad .$$

This is the portfolio that will be used in the forthcoming computations.

2.5.3.2 Choosing a loss function

The next step is to construct a loss function that forces the total risk towards the target level ξ. Consider

$$\mathcal{L}(\Sigma, \hat{\Theta}) = \frac{1}{\xi^4}\mathbf{E}_\mu\left[\left(\hat{w}_\xi^*\Sigma\hat{w}_\xi^* - \xi^2\right)^2\right] \quad ,$$

where the square of the total risk $\sqrt{\hat{w}_\xi^*\Sigma\hat{w}_\xi^*}$ is considered in order to simplify later computations and the outer square has been added in order to obtain a positive quantity. The factor $1/\xi^4$ in front of the expectation will be made clear in a little while. Plugging in the expression for $\hat{w}_\xi^*$ yields a loss function

$$\begin{aligned} \mathcal{L}(\Sigma, \hat{\Theta}) &= \frac{1}{\xi^4}\mathbf{E}_\mu\left[\left(\xi^2\frac{\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu}{\mu'\hat{\Theta}\mu} - \xi^2\right)^2\right] \\ &= \mathbf{E}_\mu\left[\left(\frac{\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu}{\mu'\hat{\Theta}\mu} - 1\right)^2\right] \quad . \end{aligned} \tag{2.10}$$

The objective is now to find an optimal estimator $\hat{\Theta}^*$ such that

$$\hat{\Theta}^* = \underset{\hat{\Theta}}{\text{argmin}} \quad \mathcal{L}(\Sigma, \hat{\Theta}) \quad . \tag{2.11}$$

This estimator will be referred to as "the risk-targeting estimator". Since $\mathcal{L}$ is independent of the risk-level ξ, it is clear that the risk-targeting estimator only needs to be computed once. The appropriate level of risk is then included when forming the corresponding portfolio w_ξ^*.

2.5.3.3 Processing of the loss function

In order to be able to construct an algorithm that allows for solving eq. (2.11), the expression for the loss function needs to be simplified. Note that the vector of expected returns μ is an unknown quantity that would need to be estimated. Since nothing is known about the direction of μ, it seems most fair to choose μ spherically distributed. One such distribution is the multivariate normal distribution. Therefore, assume that $\mu \sim \mathcal{N}_N(0, I)$. This allows for avoiding having to estimate μ, thus being able to focus completely on the estimation of Σ^{-1}, without introducing any assumptions regarding its direction.

It is clear that the loss function can be written $\mathcal{L}(\Sigma, \hat{\Theta}) = E_1 - 2E_2 + 1$, where

$$E_1 = \mathbf{E}_\mu \left[\left(\frac{\mu' \hat{\Theta} \Sigma \hat{\Theta} \mu}{\mu' \hat{\Theta} \mu} \right)^2 \right] \quad \text{and} \quad E_2 = \mathbf{E}_\mu \left[\left(\frac{\mu' \hat{\Theta} \Sigma \hat{\Theta} \mu}{\mu' \hat{\Theta} \mu} \right) \right] \quad .$$

Thus, expressions need to be found for fractions – and the square of fractions – of quadratic forms of normal random vectors. In order to do so, some inspiration has been found in [12].

2.5.3.4 Lining up some theoretical results

Below follows a few theoretical results needed in order to find simpler expressions for the two expectations E_1 and E_2 of the loss function.

Lemma 2.0.1 (Expectations of fractions)**.** *Let X and Y be positive random variables with joint moment-generating function ϕ_{XY}. Then*

$$\mathbf{E}_{X,Y} \left[\frac{Y}{X} \right] = \int_0^\infty \lim_{t_2 \to 0} \left[\frac{\partial}{\partial t_2} \phi_{XY}(-t_1, t_2) \right] dt_1 \quad , \tag{2.12}$$

$$\mathbf{E}_{X,Y} \left[\left(\frac{Y}{X} \right)^2 \right] = \int_0^\infty \lim_{t_2 \to 0} \left[\frac{\partial^2}{\partial t_2^2} \phi_{XY}(-t_1, t_2) \right] dt_1 \quad . \tag{2.13}$$

Proof. See appendix A . □

In order for the integrals in lemma 2.0.1 to be evaluated, the derivatives of the moment-generating function in eq. (2.12) and eq. (2.13) need to be computed. Some assumptions regarding the distributions of X and Y must then be made.

Lemma 2.0.2 (Derivatives of the moment-generating function)**.** *Let X and Y be quadratic forms of a standard multivariate normal distributed random vector in $\mathbf{R}^n$, i.e. let $X = U'QU$ and $Y = U'RU$ for $U \sim \mathcal{N}_n(0, I)$ and $Q, R \in \mathbf{R}^{n \times n}$. Let ϕ_{XY} be their joint moment-generating function. Then*

$$\frac{\partial}{\partial t_2} \phi_{XY}(t_1, t_2) = |I - 2t_1 Q - 2t_2 R|^{-1/2} \mathbf{E}_U[U' L' R L U] \quad , \tag{2.14}$$

$$\frac{\partial^2}{\partial t_2^2} \phi_{XY}(t_1, t_2) = |I - 2t_1 Q - 2t_2 R|^{-1/2} \mathbf{E}_U[(U' L' R L U)^2] \quad . \tag{2.15}$$

where $L \in \mathbf{R}^{n \times n}$ is such that $LL' = (I - 2t_1 Q - 2t_2 R)^{-1}$.

Proof. See appendix A . □

Summing up the results from eq. (2.12), eq. (2.13), eq. (2.14) and eq. (2.15) reveals that for $X = U'RU$ and $Y = U'RU$ where $U \sim \mathcal{N}_n(0, I)$ holds

$$\mathbf{E}_{X,Y}\left[\frac{Y}{X}\right] = \int_0^\infty |I + 2t_1 Q|^{-1/2} \mathbf{E}_U[U'L'RLU]dt_1 \quad \text{and}$$
$$\mathbf{E}_{X,Y}\left[\left(\frac{Y}{X}\right)^2\right] = \int_0^\infty |I + 2t_1 Q|^{-1/2} \mathbf{E}_U[(U'L'RLU)^2]dt_1$$

where $L : LL' = (I - 2t_1 Q)^{-1}$ – see the full context in the proof of Lemma 2.0.2 in appendix A.

It now remains to find expressions for the first and second moments of the quadratic form $U'L'RLU$.

Lemma 2.0.3 (Moments of quadratic forms)**.** *Let the n-dimensional random vector $W \sim \mathcal{N}_n(0, I)$ and let $A \in \mathbf{R}^{n\times n}$ symmetric. Then*

$$\mathbf{E}_W\left[W'AW\right] = tr(A) \quad , \tag{2.16}$$
$$\mathbf{E}_W\left[(W'AW)^2\right] = 2tr(A^2) + tr(A)^2 \quad . \tag{2.17}$$

Proof. See appendix A . □

Summing up the results from Lemma 2.0.1, Lemma 2.0.2 and Lemma 2.0.3 finally yields the full expressions for E_1 and E_2.

Theorem 2.1 (Full expectation formulas)**.** *Let E_1 and E_2 be as previously stated, i.e. let*

$$E_1 = \mathbf{E}_\mu\left[\left(\frac{\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu}{\mu'\hat{\Theta}\mu}\right)^2\right] \quad \textit{and} \quad E_2 = \mathbf{E}_\mu\left[\left(\frac{\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu}{\mu'\hat{\Theta}\mu}\right)\right]$$

for $\mu \sim \mathcal{N}_N(0, I)$, $\Sigma \in \mathbf{R}^{N\times N}$ and $\hat{\Theta} \in \mathbf{R}^{N\times N}$ where μ is the N-dimensional signal of expected risk-adjusted returns, Σ is the covariance matrix of the risk-adjusted returns and $\hat{\Theta}$ is an estimator of Σ^{-1}. Furthermore, let $\hat{\Theta} = HDH'$ where H is an orthonormal $N \times N$ matrix and D a diagonal $N \times N$ matrix with the (positive) eigenvalues $d_1, ..., d_N$ of $\hat{\Theta}$ on its main diagonal. Let L be such that $LL' = (I + 2t\hat{\Theta})^{-1}$. Then

$$E_1 = \int_0^\infty t|I + 2tD|^{-1/2}\left(2tr\Big((L'\hat{\Theta}\Sigma\hat{\Theta}L)^2\Big) + tr\Big(L'\hat{\Theta}\Sigma\hat{\Theta}L\Big)^2\right)dt = \tag{2.18}$$
$$= \int_0^\infty t\left(\prod_{i=1}^N \frac{1}{\sqrt{1+2td_i}}\right)\sum_{i=1}^N\sum_{j=1}^N \left(2[H'\Sigma H]_{ij} + [H'\Sigma H]_{ii}[H'\Sigma H]_{jj}\right)\frac{d_i^2 d_j^2}{(1+2td_i)(1+2td_j)}dt$$
$$E_2 = \int_0^\infty |I + 2tD|^{-1/2} tr\Big(L'\hat{\Theta}\Sigma\hat{\Theta}L\Big)dt = \tag{2.19}$$
$$= \int_0^\infty \left(\prod_{i=1}^N \frac{1}{\sqrt{1+2td_i}}\right)\sum_{i=1}^N [H'\Sigma H]_{ii}\frac{d_i^2}{(1+2td_i)}dt \quad .$$

Proof. See appendix A . □

This concludes the derivation of the function $\mathcal{L}$.

2.5.3.5 Regularisation by mapping of the eigenvalues

The proposed estimator $\hat{\Theta}^*$ from eq. (2.11) is defined as the minimiser of the loss function $\mathcal{L}(\Sigma, \hat{\Theta}) = E_1 - 2E_2 + 1$, where E_1 and E_2 are given by Theorem 2.1. No regularisation technique has yet been applied to $\hat{\Theta}^*$. It is from Theorem 2.1 clear that $\mathcal{L}$ is highly dependent of the eigenvalues of the estimator $\hat{\Theta}$: by altering the eigenvalues, the value of $\mathcal{L}$ can change dramatically. Recall from section 2.4.2 that an estimator $\hat{\Theta}$ can be regularised by mapping of its eigenvalues and that the regularised version can be written

$$\begin{aligned}&\tilde{\Theta} = H\tilde{D}H' \\ &\text{where } \hat{\Theta} = HDH' \text{ and } \tilde{D} = g_p(D)\end{aligned}$$

for some regularising function g_p, characterised by parameters $p_1, ..., p_m$. Let the optimal regularised estimator be

$$\tilde{\Theta}^* \coloneqq \underset{p}{\operatorname{argmin}} \quad \mathcal{L}(\Sigma, g_p(\hat{\Theta})) \quad , \tag{2.20}$$

where "$g_p(\hat{\Theta})$" is short for $Hg_p(D)H'$. $\hat{\Theta}$ is the initial, unregularised estimator that serves as input.

2.5.3.6 Extending the loss-function to multiple data sets

It has previously been remarked that one is restricted to estimating Θ out of data that is less than one year old. Still, it is desirable to make use of all data available. By partitioning the data into smaller training sets, each consisting of approximately 250 days, and computing an initial unregularised estimator for each such set a total loss function

$$\mathscr{L}\left(\Sigma, g_p(\hat{\Theta}_{(1)}), ..., g_p(\hat{\Theta}_{(K)})\right) = \sum_{k=1}^{K} \mathcal{L}(\Sigma, g_p(\hat{\Theta}_{(k)})) \quad .$$

Here $\hat{\Theta}_{(k)}$ is the unregularised estimator of Θ corresponding to the k^{th} training data set. Note that in practice Σ needs to be replaced by a sample covariance matrix $\hat{\Sigma}_{(k)}$ for the k^{th} data set. It is assumed that the number of such data sets is K. The optimal regularising parameters are computed as

$$p^* \coloneqq \underset{p}{\operatorname{argmin}} \quad \mathscr{L}\left(\Sigma, g_p(\hat{\Theta}_{(1)}), ..., g_p(\hat{\Theta}_{(K)})\right) \quad .$$

Then the risk-targeting estimator for the k^{th} data set is then given by

$$\tilde{\Theta}^{\text{RT}}_{(k)} \coloneqq g_{p^*}(\hat{\Theta}_{(k)}) \quad .$$

This concludes the derivation of the new tailormade, risk-targeting estimator of the precision matrix.

2.6 Summing up

This chapter was initiated by a very brief explanation of three basic financial derivatives: futures, forwards and options, of which Lynx mainly trades in futures. A section then followed, introducing some notation and the underlying mathematical framework. The risk-adjusted return was defined as the return divided by the corresponding risk, and was the main quantity of interest during this project. In the preceding section Modern portfolio theory was brought up. Two notions were pointed out: When optimising a portfolio, both the risk and the return must be considered simultaneously. Also, any optimal portfolio is directly proportional to the precision matrix of the risk-adjusted returns. Since the covariance structure in between markets is unknown, this matrix has to be estimated out of historical data.

The trivial choice of a precision matrix estimator would be to invert the sample covariance matrix. Unfortunately this strategy is unfeasible due to bad conditioning. Two main strategies of regularising a problem were then presented: regularisation by penalisation and regularisation by mapping of eigenvalues. The strategies were both employed in the forthcoming section, where three estimators of the precision matrix were considered. The first estimator was the simplest one, computed by adding a regularising extra term to the off-diagonal elements of the sample covariance matrix and then inverting the sum. This estimator served as a reference for comparison of results during the project. The second estimator was a standard one, being a penalised maximum likelihood-estimator. It was derived under the asumption that observations of the market data was i.i.d. multivariate normal distributed. The last estimator derived is the most exciting one, invented during the project. It was tailormade as to target a pre-determined level of risk ξ. This was achieved by finding the optimal parameters of a regularising function, optimal in the sense of minimising the distance between the actual risk and ξ for the all training data sets taken together.

3
Methodology

In this chapter some information regarding the more practical considerations of the project will be given. In section 3.1 it is explained of the performance of the estimators was compared against each other – the question "what properties are characterising for any good risk-targeting estimator of the precision matrix?" is answered here. Thereafter, schemes for solving the in section 2.5 given optimisation problems are given in section 3.2. Two types of optimisation algorithms are given, one being suitable for convex problems, the other one to be used on unconstrained problems. In section 3.3 the spectral mappings considered during the project are listed, along with explanations of why they are of interest. In the last section, section 3.4 a few words are given regarding the preparation of the market data provided by Lynx.

3.1 Evaluation of results

Before starting to construct any estimator, it is crucial to determine how to evaluate the results. What properties should be characterising for any "good" estimator? In the following sections the quantities chosen for evaluation of the performance of the estimators are summarised. It is argued that the main quantity of interest is the total loss function $\mathscr{L}$ defined in section 2.5.3: note that $\mathscr{L}$ was computed for all estimators, even for the reference estimators whose derivation did not build upon the minimisation of $\mathscr{L}$.
It is also of interest to check the performance of the estimators out-of-sample. One must make sure that the estimator doesn't underestimate the covariance to a large extent, since this would imply that the total risk is underestimated. A strategy of keeping track of the level of over- or underestimation of the precision matrix is suggested in section 3.1.2. As a final check, the total return and standard deviation are computed out-of-sample for the Markowitz portfolios corresponding to the estimators. The how-to's of this procedure are explained in section 3.1.3 below.

3.1.1 The total loss function should be small

Since the the main focus of the project was to provide a strategy for computing a *risk-targeting* estimator, the main concern is that the risk of the corresponding Markowitz portfolio is as close as possible to the target risk. Since the total loss function $\mathscr{L}$ from section 2.5.3 was chosen so as to minimise the expected value between the two it is clear that the lower the value of the total loss function, the

"more risk-targeting" the estimator is in-sample. That is, any good risk-targeting estimator should have a low value of $\mathscr{L}$. Therefore, the value of the total loss function $\mathscr{L}$ was the main quantity of interest when comparing the estimators from chapter 2: the lower the value of $\mathscr{L}$, the better the estimator.

3.1.2 Checking over- and underestimation

As argued above, it is important to keep track of whether the computed estimator $\hat{\Theta}$ tends to over- or underestimate Θ. This can be done by usage of the identity

$$\mathbf{E}\left[\left(Z_t - \bar{Z}\right)' \Sigma^{-1} \left(Z_t - \bar{Z}\right)\right] = N \quad \forall t = 1, ..., T \quad ,$$

where $Z_t = (Z_{t,1}, ..., Z_{t,N})'$, $\bar{Z} = \frac{1}{T}\sum_{t=1}^{T} Z_t$ and Σ is the covariance matrix of Z_t. For a proof of the identity, see Appendix A. The above is equivalent to

$$q \coloneqq \frac{1}{N}\mathbf{E}\left[\left(Z_t - \bar{Z}\right)' \Sigma^{-1} \left(Z_t - \bar{Z}\right)\right] \overset{!}{=} 1 \quad .$$

In practice, q is approximated by replacement of Σ^{-1} to its estimator $\hat{\Theta}$:

$$\hat{q} \coloneqq \frac{1}{N}\mathbf{E}\left[\left(Z_t - \bar{Z}\right)' \hat{\Theta} \left(Z_t - \bar{Z}\right)\right] \xrightarrow[\hat{\Theta}\to\Theta]{} 1 \quad . \tag{3.1}$$

The expectation is approximated by averaging first over the observations in each set of test data, then over all test sets. (For a description of how the available data was partitioned into train- and test sets, see section 3.4.) Whenever $\hat{q} > 1$, $\hat{\Theta}$ has been overestimated which is equivalent to underestimation of the covariance in between markets. Having this in mind, it is clear that $\hat{q}$ should preferrably be close to – and definetely not larger than – one. The closer the value of $\hat{q}$ is to one, the better the estimator.

3.1.3 Forming a Markowitz optimal portfolio

After checking what estimators $\hat{\Theta}$ minimise the total loss function $\mathscr{L}$ and whether they tend to over- or underestimate the precision matrix Θ, it might be interesting to evaluate their performance on a portfolio level out-of-sample. In order to do so, a portfolio w needs to be computed. The total return and standard deviation of w can then be examined out-of-sample.

Recall from section 2.3 that the problem of computing a risk-tagergeting optimal Markowitz portfolio reads

$$\begin{aligned} &\underset{w}{\text{maximise}} \quad w'\mu \\ &\text{subject to} \quad w'\Sigma w = \xi^2 \quad , \end{aligned} \tag{3.2}$$

where ξ is the pre-determined level of risk and $\mu = (\mu_1, ..., \mu_N)'$ is the vector of expected returns. The optimal solution of eq. (3.2) reads

$$w^* = \frac{\xi}{\sqrt{\mu'\Sigma^{-1}\mu}}\Sigma^{-1}\mu \quad .$$

It is thus clear that in order to compute the risk-targeting estimator w^*, one has to estimate not only Θ but also μ. Since also having to estimate μ would cloud the evaluation of the estimator $\hat{\Theta}$, it is desirable to employ a setting where estimation of μ can be avoided. In order to do so, a strategy similar to the one used for evaluation in [3] was applied, where the unknown μ was simply replaced by a strategically chosen vector.

Define $\hat{\mu}$ such that

$$\hat{\mu}_i \coloneqq \begin{cases} 1 & \text{if } A_i \text{ is either a commodity, an equity or a fixed-income product} \\ 0 & \text{if } A_i \text{ is a foreign exchange.} \end{cases}$$

This choice can be motivated as follows: due to inflation, the value of commodities will increase over time. Dividend and growth results in equities having a positive return over time. So will the fixed-income products. Thus, the return of such assets will be some positive number given that enough time passes. Therefore, the expected return of commodities, equities and fixed-income products is set to one. The same argument does not hold for foreign exchange, due to the symmetric character of currencies: over time, the value of one currency relative to another is not expected to increase or decrease. Therefore, the expected return of such assets is set to zero. This way of selecting $\hat{\mu}$ is the most general one, since no extra assumptions are introduced. Replacement of Σ^{-1} to $\hat{\Theta}$ and μ to $\hat{\mu}$ yields a portfolio

$$\hat{w}^* \coloneqq \frac{\xi}{\sqrt{\hat{\mu}\hat{\Theta}\hat{\mu}}}\hat{\Theta}\hat{\mu} \quad . \tag{3.3}$$

This is the portfolio used for evaluation of the estimator $\hat{\Theta}$ out-of-sample. The portfolio was formed at time t_0, then kept until time $t_0 + T$. The performance of $\hat{w}^*$ was measured in terms of its total risk-adjusted return during $[t_0, t_0 + T]$, as well as the standard deviation of the total risk-adjusted return over the same period. More details are given in section 3.1.3.2 and section 3.1.3.1 below.

3.1.3.1 Out-of-sample returns

The portfolio defined by eq. (3.3) was used for computing the positions in the markets at time t_0. For this portfolio, the total risk-adjusted return $Z_{\text{oos}}^{w^*}$ aggregated over the testing period $t \in [t_0, t_0 + T]$ was then computed as

$$Z_{\text{oos}}^{w^*} \coloneqq \sum_{t=t_0}^{t_0+T} \sum_{i=1}^{N} \hat{w}_i^* Z_{t,i} \quad .$$

Since the problem from eq. (3.2) aimed for maximising the total return, one wants to have $Z_{\text{oos}}^{w^*}$ as large as possible. An estimator $\hat{\Theta}$ giving rise to a portfolio $\hat{w}^*$ with a large value of $Z_{\text{oos}}^{w^*}$ was considered as being a "good" estimator out-of-sample.

3.1.3.2 Out-of-sample standard deviation

The portfolio defined by eq. (3.3) was used for computing the positions in the markets at time t_0. The standard deviation $\sigma_{[t_0,t_0+T]}$ of the total risk-adjusted returns of $\hat{w}^*$

was then computed using the testing data given by the period $t \in [t_0, t_0 + T]$:

$$\sigma_{[t_0,t_0+T]} \coloneqq \sqrt{Var\left(\sum_{t=t_0}^{t_0+T}\sum_{i=1}^{N} \hat{w}_i^* Z_{t,i}\right)} \quad .$$

An estimator $\hat{\Theta}$ giving rise to a portfolio $\hat{w}^*$ with a value of $\sigma_{[t_0,t_0+T]}$ close to the target risk ξ was considered as being a "good" estimator out-of-sample: the closer the value of $\sigma_{[t_0,t_0+T]}$ to ξ, the better the $\hat{\Theta}$.

3.2 Optimisation methods

As shown in chapter 2, its is clear that the derivation of estimators often results in an optimisation problem that needs to be solved. Depending on the character of the problem, different algorithms can be applied. In this project, the resulting optimisation problems were either convex – as in the case of the penalised maximum likelihood estimator – or unconstrained. Luckily enough, there are quite simple algorithms that can be applied to such problems. Below follows a brief description of the optimisation algorithms implemented as well as an explanation of how they were implemented to the actual problems.

3.2.1 Alternating direction method of multipliers (ADMM)

The following information is due to [13]. The Alternating direction method of multipliers – ADMM – is an optimisation algorithm intended for convex problems that are decomposable, i.e. problems that are of the form

$$\begin{aligned} &\underset{x,z}{\text{minimise}}\ f(x) + g(z) \\ &\text{subject to } Ax + Bz = c \end{aligned}$$

where $x \in \mathbf{R}^n$, $z \in \mathbf{R}^m$, $A \in \mathbf{R}^{p\times n}$, $B \in \mathbf{R}^{p\times m}$ and $c \in \mathbf{R}^p$. f and g are convex functions. The method is defined by forming the *augmented Lagrangian*,

$$L_\rho(x, z, y)) = f(x) + g(z) + y'(Ax + Bz - c) + \frac{\rho}{2}||Ax + Bz - c||_2^2 \quad ,$$

where $\rho > 0$ is the penalty parameter. ADMM is then performed by carrying out the iterations

$$\begin{aligned} x^{k+1} &\coloneqq \underset{x}{\text{argmin}}\ L_\rho(x, z^k, y^k) \\ z^{k+1} &\coloneqq \underset{z}{\text{argmin}}\ L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &\coloneqq y^k + \rho(Ax^{k+1} + Bz^{k+1} - c) \quad . \end{aligned}$$

By rescaling the dual variable, the algorithm can be stated on *scaled form*. Let $u = y/\rho$. The iterations then become

$$\begin{aligned} x^{k+1} &:= \underset{x}{\operatorname{argmin}} \left(f(x) + \frac{\rho}{2}||Ax + Bz^k - c + u^k||_2^2 \right) \\ z^{k+1} &:= \underset{z}{\operatorname{argmin}} \left(g(z) + \frac{\rho}{2}||Ax^{k+1} + Bz - c + u^k||_2^2 \right) \\ u^{k+1} &:= u^k + Ax^{k+1} + Bz^{k+1} - c \quad . \end{aligned}$$

3.2.1.1 ADMM algorithm for maximum likelihood estimator

Below the iteration scheme for application of the ADMM algorithm to eq. (2.9) is derived.
The problem in eq. (2.9) is equivalent to

$$\hat{\Theta}^{\text{QML1}} = \underset{\Theta}{\operatorname{argmin}} \ \{-\log|\Theta| + tr(\Theta S) + \gamma||\Theta||_{1^*}\} \quad . \tag{3.4}$$

By letting $f(x) := -\log|x| + tr(Sx)$ and $g(z) := \gamma||z||_{1^*}$, eq. (3.4) can be written as

$$\begin{aligned} &\underset{x,z}{\text{minimise}} \ f(x) + g(z) \\ &\text{subject to } x - z = 0 \quad . \end{aligned}$$

It is clear that here $A = B = I$ and $c = 0$. The iterative scheme becomes

$$x^{k+1} := \underset{x}{\operatorname{argmin}} \left(-\log|x| + tr(Sx) + \frac{\rho}{2}||x - z^k + u^k||_F^2 \right) \tag{3.5}$$

$$z^{k+1} := \underset{z}{\operatorname{argmin}} \left(\gamma||z||_{1^*} + \frac{\rho}{2}||x^{k+1} - z + u^k||_F^2 \right) \tag{3.6}$$

$$u^{k+1} := u^k + x^{k+1} + z^{k+1} \quad . \tag{3.7}$$

In order to be able to implement the algorithm, analytic expressions for the minimising xs and zs need to be found – see appendix A. The iterative scheme becomes

$$\begin{aligned} x^{k+1} &:= V^k \ \text{diag}(\tilde{x}_{11}^k, ..., \tilde{x}_{NN}^k) V^k \quad , \\ z_{ij}^{k+1} &:= \begin{cases} x_{ij}^{k+1} + u_{ij}^k & \text{when } i = j \\ x_{ij}^{k+1} + u_{ij}^k - \frac{\gamma}{\rho}\text{sign}(x_{ij}^{k+1} + u_{ij}^k) & \text{when } i \neq j \end{cases} \quad , \\ u^{k+1} &:= u^k + x^{k+1} + z^{k+1} \quad . \end{aligned}$$

where V^k is such that $\rho(z^k - u^k) - S = V^k \Lambda^k (V^k)'$ with $\Lambda^k = diag(\lambda_1^k, ..., \lambda_n^k)$ and $\tilde{x}_{ii}^k = \frac{\lambda_i^k + \sqrt{(\lambda_i^k)^2 + 4\rho}}{2\rho}$.

Replacing the off-diagonal l_1-penalty with an off-diagonal l_2-penalty is easily done and only affects the update of z. The new update then becomes

$$z_{ij}^{k+1} := \begin{cases} \frac{\rho}{\rho + 2\gamma}(x_{ij}^{k+1} + u_{ij}^k) & \text{when } i \neq j \\ x_{ij}^{k+1} + u_{ij}^k & \text{when } i = j. \end{cases}$$

3.2.2 Quasi-Newton BFGS method

In order to compute the risk-targeting estimator $\tilde{\Theta}^{\mathrm{RT}} = g_{p^*}(\hat{\Theta})$ for some regularising function g_p, the optimal parameters $p^* = (p_1^*, ..., p_N^*)'$ of g must be computed. p^* are said to be optimal when they minimise the total loss function $\mathscr{L}$, i.e. when

$$p^* \coloneqq \underset{p}{\operatorname{argmin}} \quad \mathscr{L}(\Sigma, g_p(\hat{\Theta}_{(1)}), ..., g_p(\hat{\Theta}_{(K)})) \quad , \tag{3.8}$$

where

$$\mathscr{L}(\Sigma, g_p(\hat{\Theta}_{(1)}), ..., g_p(\hat{\Theta}_{(K)})) = \sum_{k=1}^{K} \mathcal{L}(\Sigma, g_p(\hat{\Theta}_{(k)}))$$

and $\mathcal{L}$ is as given by Theorem 2.1. That is, in order to compute p^* an unconstrained optimisation problem has to be solved. Luckily enough, there exists a very simple algorithm for solving such problems: the Newton algorithm.

The Newton algorithm is an iterative method for solving problems of the form

$$\underset{x}{\text{minimise}} \quad f(x) \quad ,$$

where f is a twice differentiable function of $x \in \mathbf{R}^n$. The scheme reads

$$x^{k+1} = x^k - \alpha_k H_k^{-1} \nabla f(x_k)' \quad ,$$

where H is the Hessian of f and α_k the step size chosen so as to minimise x^{k+1}.[14] The scheme seems simple enough, but problems arise whenever $\nabla f(x)$ and H are difficult to evaluate analytically. This is the situation for $f = \mathscr{L}$: whilst the first derivatives of $\mathscr{L}$ can be computed with some difficulty, the expressions simply become too complicated when trying to find the second derivatives. The problem can be solved by approximation of the inverse Hessian. This strategy gives rise to a whole class of algorithms – the quasi-Newton methods.

In this project a quasi-Newton method called *the Broyden–Fletcher–Goldfarb–Shanno algorithm* ("quasi-Newton BFGS") was used. According to this method, the inverse Hessian is approximated iteratively as

$$H_{k+1}^{-1} \coloneqq H_k^{-1} + \frac{y_k y_k'}{y_k' s_k} - \frac{H_k^{-1} s_k s_k' (H_k^{-1})'}{s_k' H_k^{-1} s_k}$$

where y_k and s_k satisfy $y_k = \nabla f(x^{k+1}) - \nabla f(x^k)$ and $s_k = \alpha_k z_k$ respectively, for z_k such that $H_k^{-1} z_k = -\nabla f(x^k)$.[15] The full procedure is summarised in Algorithm 1 below.

Algorithm 1 Quasi-Newton BFGS

procedure ITERATE
top:
    Initialise all variables
loop:
    $z_k \leftarrow -H_k^{-1}\nabla f(x^k)$
    $\alpha_k \leftarrow \underset{\alpha}{\text{argmin}} \quad f(x^k + \alpha z_k)$
    $s_k \leftarrow \alpha_k z_k, \quad x^{k+1} \leftarrow x^k + s_k$
    $y_k \leftarrow \nabla f(x^{k+1}) - \nabla f(x^k)$
    $H_{k+1}^{-1} \leftarrow H_k^{-1} + \frac{y_k y_k'}{y_k' s_k} - \frac{H_k^{-1} s_k s_k' (H_k^{-1})'}{s_k' H_k^{-1} s_k}$

3.2.2.1 Quasi-Newton BFGS for risk-targeting estimator

The application of Algorithm 1 to the problem $\mathscr{L}$, $x = p$ is straightforward. Two remarks should be given though, regarding the computation of the gradients and regarding the one-dimensional line search.
Start by noting that $\nabla_p \mathscr{L}(\Sigma, g_p(\hat{\Theta}_{(1)}), ..., g_p(\hat{\Theta}), \mu) = \sum_{k=1}^{K} \nabla_p \mathcal{L}(\Sigma, g_p(\hat{\Theta}_{(k)}), \mu)$, where as in section 2.5.3 the sum is taken over K sets of training data, each with an initial unregularised estimator $\hat{\Theta}_{(k)}$ of its own. As for the gradient $\nabla_p \mathcal{L}$, it is defined as

$$\nabla_p \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial p_1}, ..., \frac{\partial \mathcal{L}}{\partial p_m} \right) \quad ,$$

where each $\partial \mathcal{L} / \partial p_i$ is given by

$$\frac{\partial \mathcal{L}}{\partial p_i} = \frac{\partial \mathcal{L}}{\partial \tilde{d}_1} \cdot \frac{\partial \tilde{d}_1}{\partial p_i} + ... + \frac{\partial \mathcal{L}}{\partial \tilde{d}_N} \cdot \frac{\partial \tilde{d}_N}{\partial p_i}$$

where $\tilde{d}_1, ..., \tilde{d}_N$ are the regularised eigenvalues of $\hat{\Theta}$, i.e. $\tilde{d}_i = g_p(d_i)\ \forall i = 1, ..., N$. Letting $d \coloneqq (d_1, ..., d_N)'$ and $g_p(d) \coloneqq (g_p(d_1), ..., g_p(d_N))'$ allows for writing the above on a somewhat more compact form,

$$\nabla_p \mathcal{L} = \left(\nabla_d \mathcal{L}' \frac{\partial g_p(d)}{\partial p_1}, ..., \nabla_d \mathcal{L}' \frac{\partial g_p(d)}{\partial p_m} \right)' \quad ,$$

where $\nabla_d \mathcal{L} = (\frac{\partial \mathcal{L}}{\partial d_1}, ..., \frac{\partial \mathcal{L}}{\partial d_N})'$. It remains to find expressions for the $\frac{\partial \mathcal{L}}{\partial p_i}$s and the $\frac{\partial g_p(d_i)}{\partial p_j}$s. Whilst the latter depends on the choice of regularising function g_p, the first can be evaluated using the formula for $\mathcal{L}$ from Theorem 2.1:

$$\begin{aligned}
&\mathcal{L}(\Sigma, \hat{\Theta}, \mu) = \\
&= \int_0^\infty t \left(\prod_{i=1}^{N} \frac{1}{\sqrt{1 + 2td_i}} \right) \sum_{i=1}^{N} \sum_{j=1}^{N} (2[H'\Sigma H]_{ij} + [H'\Sigma H]_{ii}[H'\Sigma H]_{jj}) \frac{d_i^2 d_j^2}{(1 + 2td_i)(1 + 2td_j)} dt + \\
&- 2 \int_0^\infty \left(\prod_{i=1}^{N} \frac{1}{\sqrt{1 + 2td_i}} \right) \sum_{i=1}^{N} [H'\Sigma H]_{ii} \frac{d_i^2}{(1 + 2td_i)} dt + 1 \quad ,
\end{aligned}$$

which results in partial derivatives according to

$$\frac{\partial \mathcal{L}}{\partial d_k} =$$
$$= \int_0^\infty t \left(\prod_{i=1}^{N} \frac{1}{\sqrt{1+2td_i}} \right) \left(-\frac{t}{(1+2td_k)} \sum_{i=1}^{N}\sum_{j=1}^{N} (2[H'\Sigma H]_{ij}^2 + [H'\Sigma H]_{ii}[H'\Sigma H]_{jj}) \cdot \right.$$
$$\left. \cdot \frac{d_i^2 d_j^2}{(1+2td_i)(1+2td_j)} + \frac{4d_k(1+d_k t)}{(1+2td_k)^2} \sum_{i=1}^{N} (2[H'\Sigma H]_{ik}^2 + [H'\Sigma H]_{ii}[H'\Sigma H]_{kk}) \frac{d_i^2}{(1+2td_i)} \right) dt +$$
$$- 2\int_0^\infty \left(\prod_{i=1}^{N} \frac{1}{\sqrt{1+2td_i}} \right) \left(-\frac{t}{(1+2td_k)} \sum_{i=1}^{N} [H'\Sigma H]_{ii}^2 \frac{d_i^2}{(1+2td_i)} + 2d_k [H'\Sigma H]_{kk} \frac{(1+td_k)}{(1+2td_k)^2} \right) dt \quad .$$

Here it has been assumed that differentiating under the integral signs is allowed. The assumption was checked numerically. The above formula holds for all eigenvalues d_k $k = 1, ..., N$ of $\hat{\Theta}$. Note that in practice, the integrals over t had to be evaluated numerically.

3.2.2.2 One-dimensional line search

Regarding the one-dimensional line search over α, it was carried out using a form of Armijo's Rule as described by [14]. The one-dimensional line search function

$$\phi(\alpha) \coloneqq \mathscr{L}(\Sigma, g_{p_k+\alpha z_k}(\hat{\Theta}_{(1)}), ..., g_{p_k+\alpha z_k}(\hat{\Theta}_{(N)}), \mu)$$

was defined. The goal is to choose α such that $\phi(\alpha)$ is minimised. In order to do so, two criteria are considered:

$$\phi(\alpha) \leq \phi(0) + \varepsilon\alpha\phi'(0) \quad \text{and} \tag{3.9}$$
$$\phi(\eta\alpha) > \phi(0) + \varepsilon\eta\alpha\phi'(0) \tag{3.10}$$

for some constants ε and η[1]. Some value α_0 of α was initially chosen, the conditions of eq. (3.9) and eq. (3.10) were checked one after another. In case eq. (3.9) did hold, α was repeatedly increased by a factor η until eq. (3.10) no longer holds. The largest α that satisfies both conditions is then chosen. In case eq. (3.9) would not hold, α would repeatedly be decreased by a factor η until the condition was met.
Depending on the eigenvalues $d_1, ..., d_N$ of $\hat{\Theta}$, the line-search function might be very ill-behaved, especially during the first few iterations of the full algorithm. The function sometimes has segments where its derivative is very large compared to the function itself. There might also be singularities in some cases. In order to ensure stability and convergence, α_0 must be chosen very small. Otherwise one might end up in a situation where one of the two conditions would never occur. Numerical experiments showed that $\alpha_0 = 2 \cdot 10^{-4}$ would yield good results. Taking such small steps in the optimal direction results in a rather slow convergence of the algorithm, but was necessary due to the bad behaviour of the function $\mathscr{L}$ (and thus of ϕ) – see chapter 4 for more information regarding the behaviour of the objective function.

[1] In this project $\varepsilon = 0.2$ and $\eta = 2$ were used.

3.2.2.3 Choosing an appropriate stopping criterion

A vital part in any optimisation routine is the choice of stopping criterion. Since each iteration of the quasi-Newton BFGS method applied to eq. (3.8) was quite time consuming, it was important to find a stopping criterion that did not allow for lots of extra iterations. Many different criteria were tried out initially. Some of them seemed to work well on low-dimensional and well-behaved objective functions, but failed to meet the requirements when applied to $\mathscr{L}$. Eventually a condition on the "typical length of the gradient" was found:

```
if ||∇_p L||_2^2 < 0.0001 · Γ
  then STOP
endif
```

where Γ is a number indicating the typical length of the gradient $\nabla_p \mathscr{L}$,

$$\Gamma \coloneqq \frac{||\mathscr{L}(\Sigma, g_{p+h}(\hat{\Theta}), ..., g_{p+h}(\hat{\Theta}), \mu) - \mathscr{L}(\Sigma, g_p(\hat{\Theta}), ..., g_p(\hat{\Theta}), \mu)||_2^2}{||h||_2^2} \quad ,$$

for some h small. This is the stopping criterion used when solving the optimisation problem from eq. (3.8).

3.3 About the regularising function

It was in chapter 2 argued that sometimes a matrix can be regularised by direct manipulation of its spectrum. The regularised version $\tilde{\Theta}$ of the estimator $\hat{\Theta}$ was defined as

$$\tilde{\Theta} \coloneqq H\tilde{D}H' \quad ,$$

for $\tilde{D} = g_p(D)$ where $H, D \in \mathbf{R}^{N \times N}$ are such that $\hat{\Theta} = HDH'$, $HH' = H'H = I$ and D is a diagonal matrix with the eigenvalues $d_1, ..., d_N$ of $\hat{\Theta}$ on the main diagonal. The original eigenvalues $d_1, ..., d_N$ are mapped to regularised ones $\tilde{d_1}, ..., \tilde{d_N}$ through some function g_p, characterised by the parameters $p_1, ..., p_m$. It was previously argued that an S-shaped mapping might be the most appropriate choice for g_p. This was taken into account when trying out a few different regularising functions.

3.3.1 Sigmoid regulariser

A natural first choice when considering S-shaped functions is the sigmoid function. It was parameterised in such a way that its slope would always end up being positive when running the optimisation routine described in section 3.2.2.1. A small constant a was added to the sigmoid, in order to ensure that the smallest eigenvalues wasn't mapped to a value close to zero.

$$g_p^{\text{sigmoid}}(d) \coloneqq a + \frac{e^b}{1 + e^{-\kappa(d - x_0)}} \quad . \tag{3.11}$$

Note that $g_p^{\text{sigmoid}} : \mathbf{R} \to [a, a + e^b]$. The fact that e^b is used instead of just b is that $e^b > 0 \ \forall b \in \mathbf{R}$, thus $a + e^b > a$ always. Here, the parameters to be optimised over were

$$\begin{aligned} p_1 &= b \quad , \\ p_2 &= \kappa \quad \text{and} \\ p_3 &= x_0 \quad . \end{aligned}$$

Note that a was held fixed and not optimised over: this strategy was chosen as a way of manually controlling the lowest regularised eigenvalue and thus the resulting condition number of $\tilde{\Theta}$. It was interesting to investigate the properties of regularised estimators of different conditioning, since the condition number can be seen as a measure of the amount of regularisation applied. The lower the condition number, the heavier the regularisation.
The partial derivatives of g_p^{sigmoid} with respect to p_1, p_2 and p_3 are

$$\begin{aligned} \frac{\partial g_p^{\text{sigmoid}}}{\partial b} &= \frac{e^b}{1 + e^{-\kappa(d - x_0)}} \quad , \\ \frac{\partial g_p^{\text{sigmoid}}}{\partial \kappa} &= \frac{e^b (x_0 - d)}{(1 + e^{-\kappa(d - x_0)})^2} \quad \text{and} \\ \frac{\partial g_p^{\text{sigmoid}}}{\partial x_0} &= \frac{\kappa e^b e^{-\kappa(d - x_0)}}{(1 + e^{-\kappa(d - x_0)})^2} \quad . \end{aligned}$$

These expressions were used for computing the full gradient of the total loss function $\mathscr{L}$ as described in section 3.2.2.1. Optimal values of b, κ and x_0 as found by the quasi-Newton BFGS routine are given in section 4.1.

3.3.2 Normal regulariser

Another example of an S-shaped function is the cumulative density function of the normal distribution. It was parametrised in a similar way as the sigmoid function was:

$$g_p^{\text{normal}}(d) \coloneqq a + b\Phi(d - e^{x_0}) \quad , \tag{3.12}$$

where Φ is the cumulative density function of a standard normal distribution,

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy \quad .$$

Note that $g_p^{\text{normal}} : \mathbf{R} \to [a, a + b]$. In this case the constant b was not parameterised exponentially, but the intersection-parameter x_0 was. This selection was made based on numerical experiments.
The parameters to be optimised over were

$$\begin{aligned} p_1 &= b \quad \text{and} \\ p_2 &= x_0 \quad , \end{aligned}$$

while a was held fixed just as for the sigmoid regulariser. The partial derivatives of $g_p^{\text{normal}}(d)$ with respect to the parameters are

$$\frac{\partial g_p^{\text{normal}}}{\partial b} = \Phi(d - e^{x_0}) \quad \text{and}$$
$$\frac{\partial g_p^{\text{normal}}}{\partial x_0} = -\frac{be^{x_0}}{\sqrt{2\pi}} e^{-(d-e^{x_0})^2/2} \quad .$$

Optimal values of b and x_0 as found by the quasi-Newton BFGS optimisation routine are given in section 4.1.

3.3.3 Log regulariser

The logarithmic function is not an S-shaped mapping. Investigation of the shape of the regularising function corresponding to the l_2-penalised maximum likelihood estimator from section 2.5.2 revealed that the l_2-penalty seems to transform the eigenvalues of the estimator logarithmically[2] – see chapter 4 for more information on this part. For this reason, a log-shaped regularising mapping might also be of interest. The function was parameterised in such a way that it resembled the mapping of the eigenvalues made by the l_2-penalised maximum likelihood estimator,

$$g_p^{\log}(d) \coloneqq a - bc + b\log(d + e^c) \quad . \tag{3.13}$$

Note that since the logarithmic function is unbounded, so is $g_p^{\log}$. Since the eigenvalues $d_1, ..., d_N$ are both positive and bounded, their images are restricted to $[a, a - bc + b\log(d_{\max} + e^c)]$ where $d_{\max} = \max\{d_i\}_{i=1}^N$. $g_p^{\log}(d)$ quickly flattens out, even for quite small d.
The parameter a was held fixed, while

$$p_1 = b \quad \text{and}$$
$$p_2 = c$$

were optimised over. The partial derivatives of $g_p^{\log}$ with respect to b and c are

$$\frac{\partial g_p^{\log}}{\partial b} = \log(d + e^c) - c \quad \text{and}$$
$$\frac{\partial g_p^{\log}}{\partial c} = \frac{be^c}{d + e^c} - b \quad .$$

The optimal values of b and c as found by the optimisation routine are given in section 4.1.

3.3.4 Linear regulariser

An attempt of applying a linear regularising function was also made, by choosing

$$g_p^{\text{linear}}(d) \coloneqq a + bd \quad , \tag{3.14}$$

[2]This argument makes sense, since the l_2-penalised maximum likelihood estimator is rotation invariant. Thus it suffices to look at the eigenvalues only.

where a was held fixed and

$$p_1 = b$$

was optimised over. The derivative of g_p^{linear} with respect to b is simply

$$\frac{\partial g_p^{\text{linear}}}{\partial b} = d \quad .$$

Unfortunately this strategy turned out to be unsuccessful – more information is given in chapter 4 and chapter 5.

3.3.5 Linear-sigmoid regulariser

The final regularising function tried out was a linear combination of the linear and the sigmoid functions:

$$g_p^{\text{ls}}(d) \coloneqq a + e^c d + \frac{e^d}{1 + e^{-\kappa(d - x_0)}} \quad . \tag{3.15}$$

This function was considered since (for small) e^c it maintains some of the nice properties of the sigmoid, such as the S-shape, even though this form is more general: while the sigmoid function has a very flat "floor" and "ceiling", the linear-sigmoid allows for a slightly tilting floor and ceiling.
The parameter a was kept fixed just as for the previous functions. The parameters optimised over were

$$\begin{aligned} p_1 &= d \quad , \\ p_2 &= c \quad , \\ p_3 &= \kappa \quad \text{and} \\ p_4 &= x_0 \quad , \end{aligned}$$

with corresponding partial derivatives

$$\begin{aligned} \frac{\partial g_p^{\text{ls}}}{\partial d} &= \frac{e^d}{1 + e^{-\kappa(d - x_0)}} \quad , \\ \frac{\partial g_p^{\text{ls}}}{\partial c} &= e^c d \quad , \\ \frac{\partial g_p^{\text{ls}}}{\partial \kappa} &= \frac{e^b (x_0 - d)}{(1 + e^{-\kappa(d - x_0)})^2} \quad \text{and} \\ \frac{\partial g_p^{\text{ls}}}{\partial x_0} &= \frac{\kappa e^b e^{-\kappa(d - x_0)}}{(1 + e^{-\kappa(d - x_0)})^2} \quad . \end{aligned}$$

The optimal parameter values, as computed by the optimisation routine, are given in section 4.1.

3.4 Data preparation

The first step when dealing with any kind of data is to set up a well-organised working environment for data extraction and preparation. The following section describes the structure of the data used, as well as a few standard strategies used at Lynx for preparing the data. These strategies was implemented without question throughout the project. Another section then follows, describing the working environment set up for training and testing the estimators.

3.4.1 Description of the market data

The database provided by Lynx contains the opening-, closing-, daily maximum and minimum prices of 95 instruments. It also contains information regarding the instruments themselves, such as their type (fixed-income, equities, foreign exchange and commodities). For some instruments, the oldest price data dates back to the 1980s. For other instruments, only newer price data is available. Since most time series starts on the 1$^{\text{st}}$ of January 2000, only data between this day and the 1$^{\text{st}}$ of January 2019 was used during the project.

3.4.1.1 Dealing with missing data

The markets are closed on holidays and weekends, resulting in missing values on such days. Depending on where the markets are located, they are closed on different days due to national holidays varying in between countries. At Lynx, missing values are handled by simply replicating the last non-missing value.

3.4.1.2 Computing the risk-adjusted returns

The risk-adjusted returns were computed from the closing prices according to eq. (2.1) and eq. (2.2). The volatility of the i^{th} market at time t $\sigma_{t,i}$ was estimated using an EMA-filtered *Yang-Zhang volatility estimator* – see [16]. Due to the EMA-component, the estimate of $\sigma_{t,i}$ is unstable for t small. In order to avoid this problem, a burn-in period of approximately six months after the start of the time series were excluded from the computations.

3.4.1.3 Avoiding underestimation of the covariance

The opening hours of the markets naturally varies due to differences in between time zones. For example, American markets are lagging behind European markets, which in turn are lagging behind Asian ones. This lag results in an decreased covariance in between markets belonging to different time zones. Unfortunately market data is only available on a daily level, for what reason the time lag is made somewhat invisible. In order not to risk underestimation of the covariance, the n-day returns were used instead of the 1-day returns (as was given by eq. (2.1)):

$$R_{t+n,i} = P_{t+n,i} - P_{t,i} \quad n \in \mathbf{N} \quad .$$

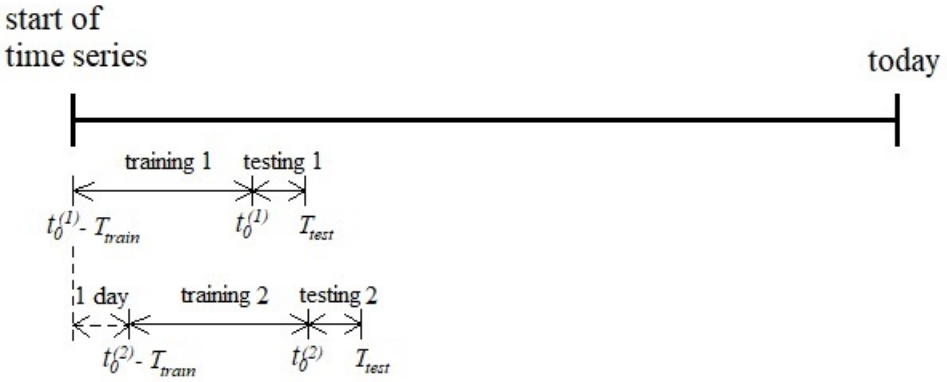


Figure 3.1: Illustration over how pairs of training- and testing data were formed in a sliding manner, going through all historical data available. For the k^{th} pair, the estimators were constructed using data from the days $t_0^k - T_{\text{train}}, ..., t_0^k$. They were then tested out-of-sample using "future" (as seen from day t_0^k) data from the days $t_0^k + 1, ..., t_0^k + T_{\text{test}}$. The next pair was then defined by letting $t_0^{k+1} = t_0^k + 1$ and repeating the procedure.

Here the notation from section 2.2 is used. $n > 1$ was chosen to be small, a typical choice was $n = 3$. The risk-adjusted return $Z_{t+n,i}$ was then computed as previously described.

3.4.2 Working environment for training and testing

In section 3.1 four quantities used for evaluation were lined up: the total loss function $\mathscr{L}$, the dimensional check $\hat{q}$, the average return $\bar{Z}_{\text{oos}}$ and the standard deviation $\sigma_{[t_0,t_0+T]}$. Out of these four quantities, only $\mathscr{L}$ was computed in-sample – that is, for training data. The other three, $\hat{q}$, $\bar{Z}_{\text{oos}}$ and $\sigma_{[t_0,t_0+T]}$, were computed out-of-sample – that is, for testing data. The following section describes how the available historical data was partitioned into pairs of training- and test sets and what parts of the training sets were used for computing what. T_{train} and T_{test} will be used to denote the number of training- and testing days respectively.

A training- and testing environment was set up, taking the following two aspects into consideration: Firstly, it is desirable that the testing period occurs just after the training period has ended. This is the situation in real life, where historical training data from approximately one year back is used at time t_0 in order to construct an estimator that is to be used until time $t_0 + T_{\text{test}}$. Secondly, one would like to have as many pairs of training- and testing data sets as possible. Therefore, an environment where the current day t_0 is successively moved one day forward at the time was set up. This resulted in a maximum number of pairs, thus making the most out of the available data. Implementation of the previously described strategy resulted in a total of 4569 pairs[3]. For every new pair, only instruments without any missing values were included into the train- and test sets. Note that this strategy resulted in

[3]That is, $K = 4569$ whith K as given in section 2.5.3.6.

some periods containing fewer than 95 instruments, something that had to be taken into account when computing $\hat{q}$.
For each new set of training data, an estimate $\hat{\Theta}$ of the inverse covariance matrix $\Theta = \Sigma^{-1}$ was computed. It was then used for computing a Markowitz portfolio $w^* \propto \hat{\Theta}\hat{\mu}$ (see section 2.3 and section 3.1). The performance of w^* was then tested out-of-sample on the corresponding set of testing data. The testing quantities were averaged over all data pairs in order to ensure stability and reliability of results. For an overview of the full procedure, see Algorithm 2 below.

Algorithm 2 Testing the performance of the estimators out-of-sample

procedure TRAINANDTEST
top:
    $T_{\text{test}} \leftarrow$ # days used to test out-of-sample
    $T_{\text{train}} \leftarrow$ # days used to train
    load and prepare *data*
    $K \leftarrow$ number of pairs
    $t_0^0 \leftarrow T_{\text{train}} + 1$
loop:
    for $1 \leq k \leq K$ **do**
        $t_0^k = t_0^{k-1} + 1$
        $dataTrain \leftarrow data(t_0^k - T_{\text{train}}, ..., t_0^k)$
        $dataTest \leftarrow data(t_0^k + 1, ..., t_0^k + T_{\text{train}})$
        compute $\hat{\Theta}_{(k)}$ using *dataTrain*
        compute testing quantity X^k using $\hat{\Theta}_{(k)}$ on *dataTest*
        close
        $X \leftarrow \frac{1}{K}\sum_{k=1}^{K} X^k$, average over all pairs

Remark 3.0.1. One could argue that the usage of market data ranging back to the year of 2000 should result in a large bias. That is not the case though: in case old data was used for training estimators to be used on new data, one would indeed risk ending up in a situation where the bias became large. The old data is only used as a method of evaluating the performance of the estimators though, thus no bias will be carried forward. The underlying assumption is here that a technique for estimating Θ that worked well a decade ago, should work well also today.

3.4.2.1 Implementation when computing the total loss function

A slight modification of the above described procedure was made when computing the total loss function $\mathcal{L}$. This was due to the fact that it depends on the covariance matrix Σ. Ideally one would like to know the true covariance matrix Σ of the testing period $t \in [t_0, t_0 + T_{\text{test}}]$, which of course is infeasible at $t = t_0$. Thus Σ need to be replaced by an estimator $\hat{\Sigma}$, based on the training data.
The chosen strategy goes as follows: the T_{train} days of training data were partitioned into two disjoint subsets, each consisting of $T_{\text{train}}^{\Sigma}$ and $T_{\text{train}} - T_{\text{train}}^{\Sigma}$ days respectively.

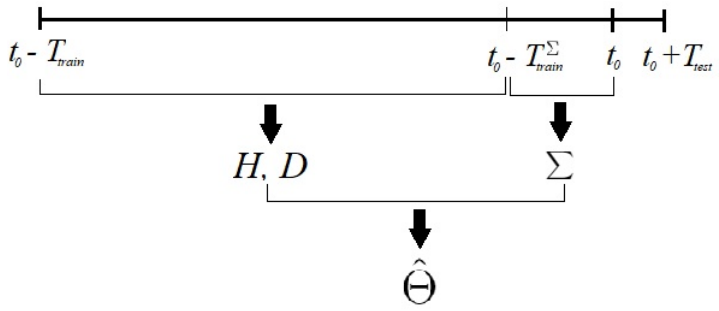


Figure 3.2: One pair of training- and testing data. The testing data has been partitioned into two subsets, each of length $T^{\Sigma}_{\text{train}}$ and $T_{\text{train}} - T^{\Sigma}_{\text{train}}$ respectively. The first part of the training data (of length $T_{\text{train}} - T^{\Sigma}_{\text{train}}$) is used for computing the normal decomposition matrices H and D of the sample precision matrix. The second part of the training data (of length $T^{\Sigma}_{\text{train}}$) is used for computing a sample covariance matrix Σ. H, D are Σ are then plugged into the machinery described in section 2.5.3.

See fig. 3.2 for a visualisation of the time line for each pair of training- and testing data. A typical choice for the number of days were $T_{\text{train}} = 250$ and $T^{\Sigma}_{\text{train}} = 20$.

3.5 Summing up

This chapter was initiated with a description of how the performance of the in chapter 2 derived estimators were evaluated. Four quantities of interest were listed in the order of significance. The most important quantity was considered to be the value of the total loss function $\mathscr{L}$, which measures how close the corresponding Markowitz portfolio is to the target risk in-sample. $\mathscr{L}$ should be as small as possible. The second most important quantity was $\hat{q}$, introduced as a means of checking the level of over- or underestimation of $\hat{\Theta}$. $\hat{q}$ was compued out-of-sample but without the formation of any portfolio. The last two quantities of interest were the average return $\bar{Z}Z_{\text{oos}}$ and the standard deviation (of returns) $\sigma_{[t_0,t_0+T]}$ computed for a portfolio $\hat{w}^* \propto \hat{\Theta}\hat{\mu}$ out-of-sample, where $\hat{\mu}$ is an N-dimensional column vector with all entries equal to zero or one depending on the type of underlying asset.
Then a section followed, deriving the schemes of the optimisation routines involved when computing the estimators. Two types of algorithms were presented: the ADMM algorithm and the quasi-Newton BFGS algortihm. The former method is to be applied to convex problems – such as the minimisation of the negative log-likelihood from section 2.5.2. The latter methods is to be applied to unconstrained problems – such as the minimisation of the loss function from section 2.5.3. Closed-form expressions for the steps involved were given.
The regularising mappings considered were listed in section 3.3, along with brief explanations on why these specific functions where of interest. Partial derivatives were given of all the mappings, with respect to the parameters that were to be optimised over.
The last section described the structure of the market data provided by Lynx, as

well as how it was initially prepared before being plugged into any estimator. An overview was given on how the available data was partitioned into pairs of training- and testing data, and of which parts of the training data was used for computing what quantities.

4

Results

This chapter has been partitioned into two main parts. The first part presents some intermediate results, intending to lead up to the final results given in the second part. In the first part, some results regarding the behaviour of the loss function $\mathcal{L}$ are given. In the second part, information regarding the performance of the estimators for various choices of parameters is included.

4.1 A few intermediate results

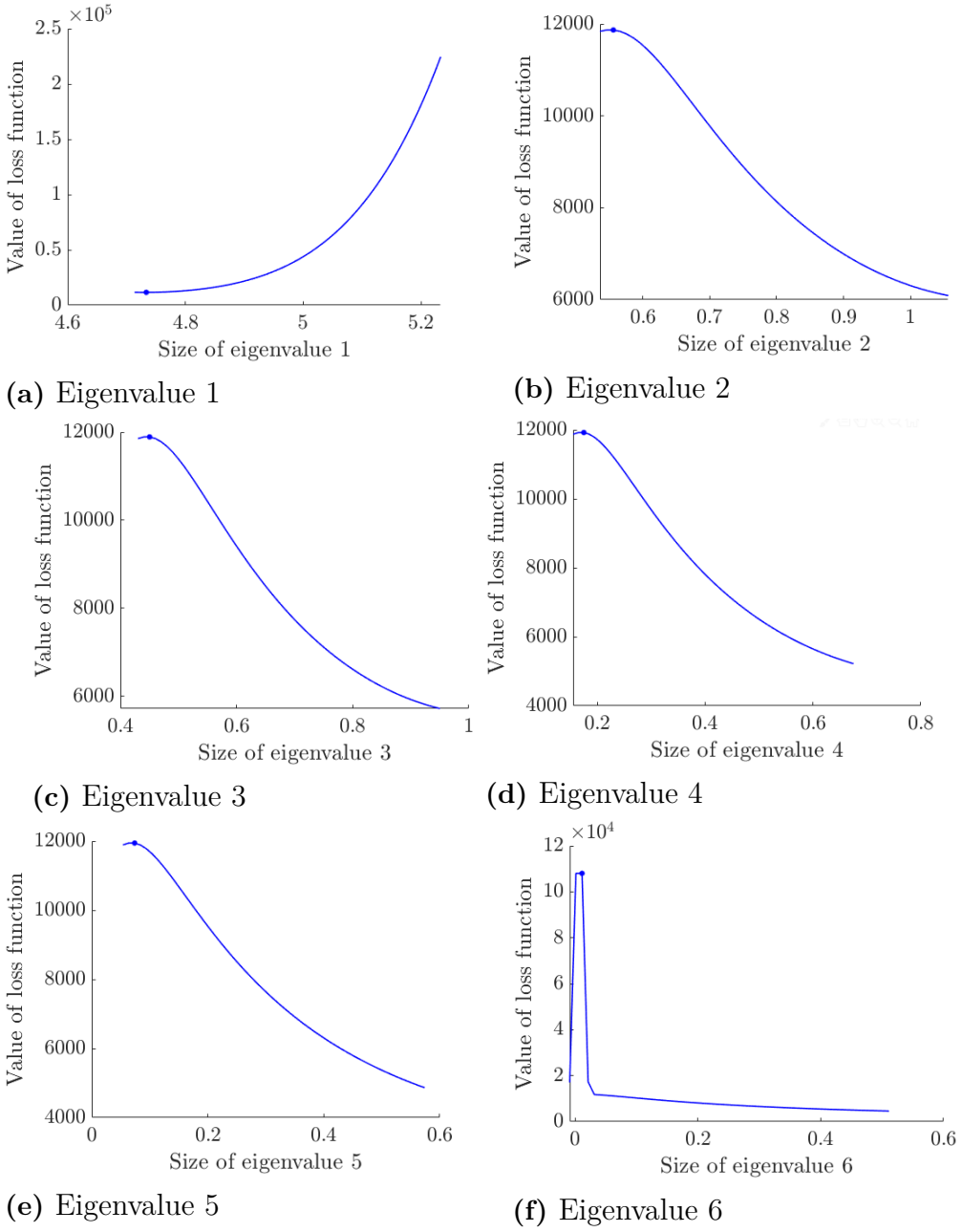
In this section a few intermediate results are given, intended to give the reader some intuition regarding the (numerical) nature of the problems of the project.

4.1.1 The loss function and the eigenvalues

A fundamental quantity of interest throughout the project is the total loss function, defined in section 2.5.3.6. It is given by

$$\mathscr{L}(\Sigma, g_p(\hat{\Theta}_{(1)}), ..., g_p(\hat{\Theta}_{(K)})) \coloneqq \sum_{k=1}^{K} \mathcal{L}(\Sigma, g_p(\hat{\Theta}_{(k)})) \quad ,$$

where the loss function $\mathcal{L} = E_1 - 2E_1 + 1$ with E_1 and E_2 as given by Theorem 2.1. The expressions for E_1 and E_2 reveal that the value of $\mathcal{L}$ is highly dependent of the eigenvalues $d_1, ..., d_N$ of $\hat{\Theta}$: hence the idea of minimising $\mathscr{L}$ by mapping of the K sets of eigenvalues. Before carrying out any such mapping of the eigenvalues, it is interesting to try to distinguish how the individual eigenvalues affect the magnitude of $\mathcal{L}$. The investigation was carried out as follows.
An artificial $N \times N$ covariance matrix similar to the real covariance matrix was generated, its eigendecomposition computed. Eigenvalues of the precision matrix were obtained through inversion of the eigenvalues of the covariance matrix. The eigenvalues were then altered one by one, the loss function $\mathcal{L}$ was re-computed after each such modification. This way, $\mathcal{L}$ was obtained as a function of each of the eigenvalues individually.
Since the numerical integrals in $\mathcal{L}$ are very sensitive to manipulation of the eigenvalues (especially to shrinkage of them), each eigenvalue could only be varied within a rather small range. The results from the case when $N = 6$ are displayed in fig. 4.1. The original eigenvalues are listed in table 4.1.



(a) Eigenvalue 1 **(b)** Eigenvalue 2

(c) Eigenvalue 3 **(d)** Eigenvalue 4

(e) Eigenvalue 5 **(f)** Eigenvalue 6

Figure 4.1: The loss function $\mathcal{L}$ as a function of the eigenvalues of an artificially generated precision matrix, for the case when $N = 6$. Each eigenvalue has been altered separately, then the loss function has been recomputed. The original eigenvalue is marked out with a dot. No regularisation was applied.

Index	Eigenvalue
1	4.7
2	0.56
3	0.45
4	0.17
5	0.07
6	0.01

Table 4.1: The original eigenvalues of the artificially generated precision matrix. They are marked out with blue dots in fig. 4.1.

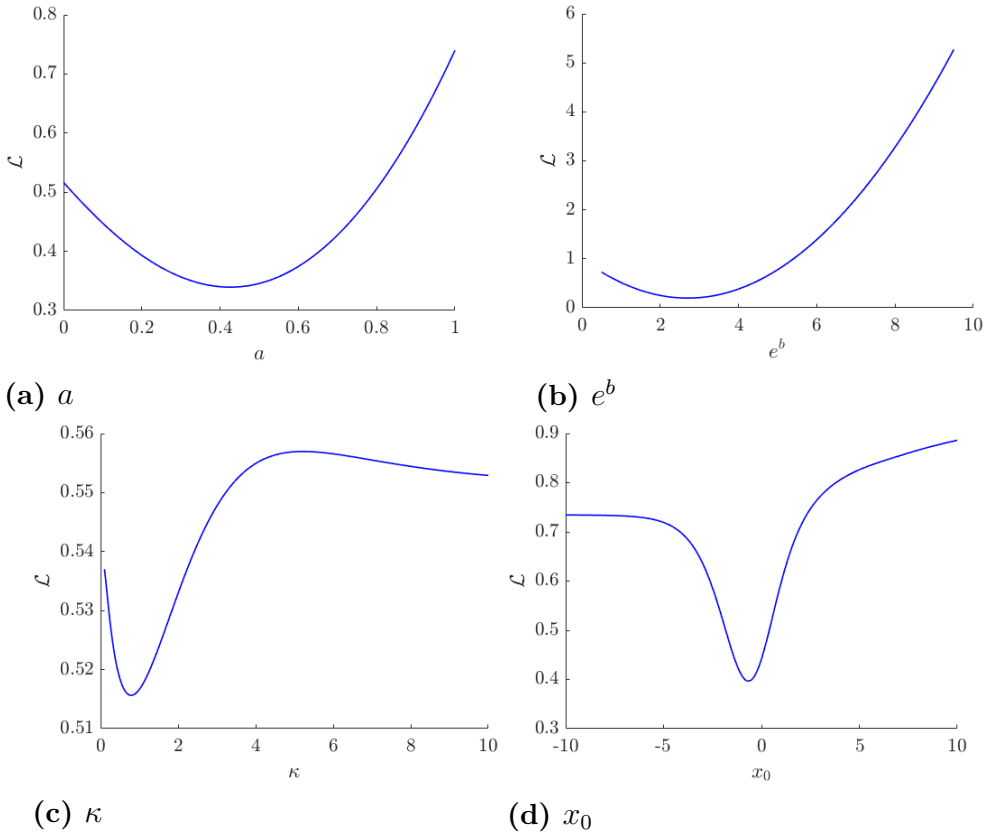
It is from fig. 4.1 clear that for the largest eigenvalue – "Eigenvalue 1" in fig. 4.1a – the loss function $\mathcal{L}$ grows exponentially as the eigenvalue is increased. This eigenvalue also has the largest impact on the magnitude of $\mathcal{L}$: just compare the values of the y-axes of fig. 4.1. For all other – smaller – eigenvalues, $\mathcal{L}$ instead decreases exponentially as the eigenvalue is increased. For larger Ns (not included here in order to save some space) the results are similar: for the (few) largest eigenvalues (typically of size 2 or larger) $\mathcal{L}$ increases when the eigenvalue increases, whilst for eigenvalues of intermediate or small size $\mathcal{L}$ decreases when the eigenvalue is increased. This indicates that $\mathcal{L}$ indeed favours S-shaped spectral mappings, as proposed already in chapter 2.

4.1.2 The loss function and the regularising parameters

In the previous section it was displayed how the loss function $\mathcal{L}$ depends upon the various eigenvalues of the precision matrix estimator. It was concluded that $\mathcal{L}$ grows when large eigenvalues are increased and when small eigenvalues are decreased. It was also noted that the numerical integrals of $\mathcal{L}$ are very sensitive to alterations of the eigenvalues. For this reason, it is of interest to see how the value of the loss function $\mathcal{L}$ varies along with the parameters p of the regularising spectral mappings, g_p. The investigation was carried out in a similar manner as in the previous section. An artificial $N \times N$ covariance matrix similar to the sample covariance matrix of the market data was generated, its normal decomposition was computed. The eigenvalues of the corresponding precision matrix was obtained by inversion of the eigenvalues of the artificial covariance matrix. They were then mapped by the regularising function g_p. Using these mapped eigenvalues, the loss function $\mathcal{L}$ was computed. The procedure was repeated while varying the regularising parameters p one by one, demonstrating $\mathcal{L}$ as a function of the regularising parameters. This was done for those of the in section 3.3 listed regularising functions that seemed to result in the numerical integrals converging: g_p^{sigmoid}, $g_p^{\log}$ and g_p^{ls}. The result of the investigations are given in figure 4.2 – 4.4.
The sigmoid regularising function is given by

$$g_p^{\text{sigmoid}}(d) \coloneqq a + \frac{e^b}{1 + e^{-\kappa(d - x_0)}} \quad .$$

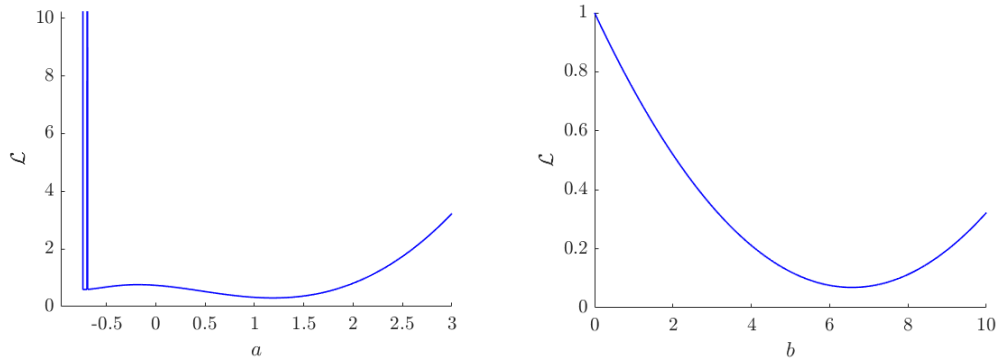
Its dependency of the various parameters is visualised in fig. 4.2. It is clear that the parameter c affects the value of the loss function $\mathcal{L}$ the most. The parameter κ



(a) a **(b)** e^b

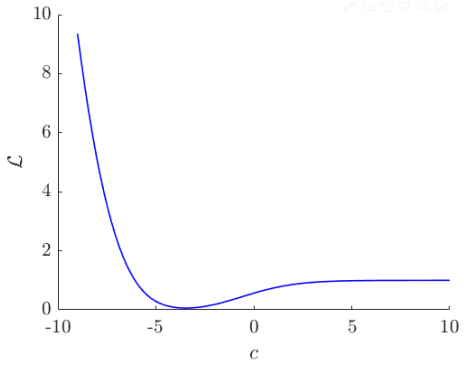
(c) κ **(d)** x_0

Figure 4.2: The loss function as a function of the regularising parameters, using the regularising function $g_p^{\text{sigmoid}}(d) \coloneqq a + \frac{e^b}{1+e^{-\kappa(d-x_0)}}$ for $a = 0$, $b = 0.5$, $\kappa = 1$ and $x_0 = 0.5$ as original input, then varying the parameters one at the time.



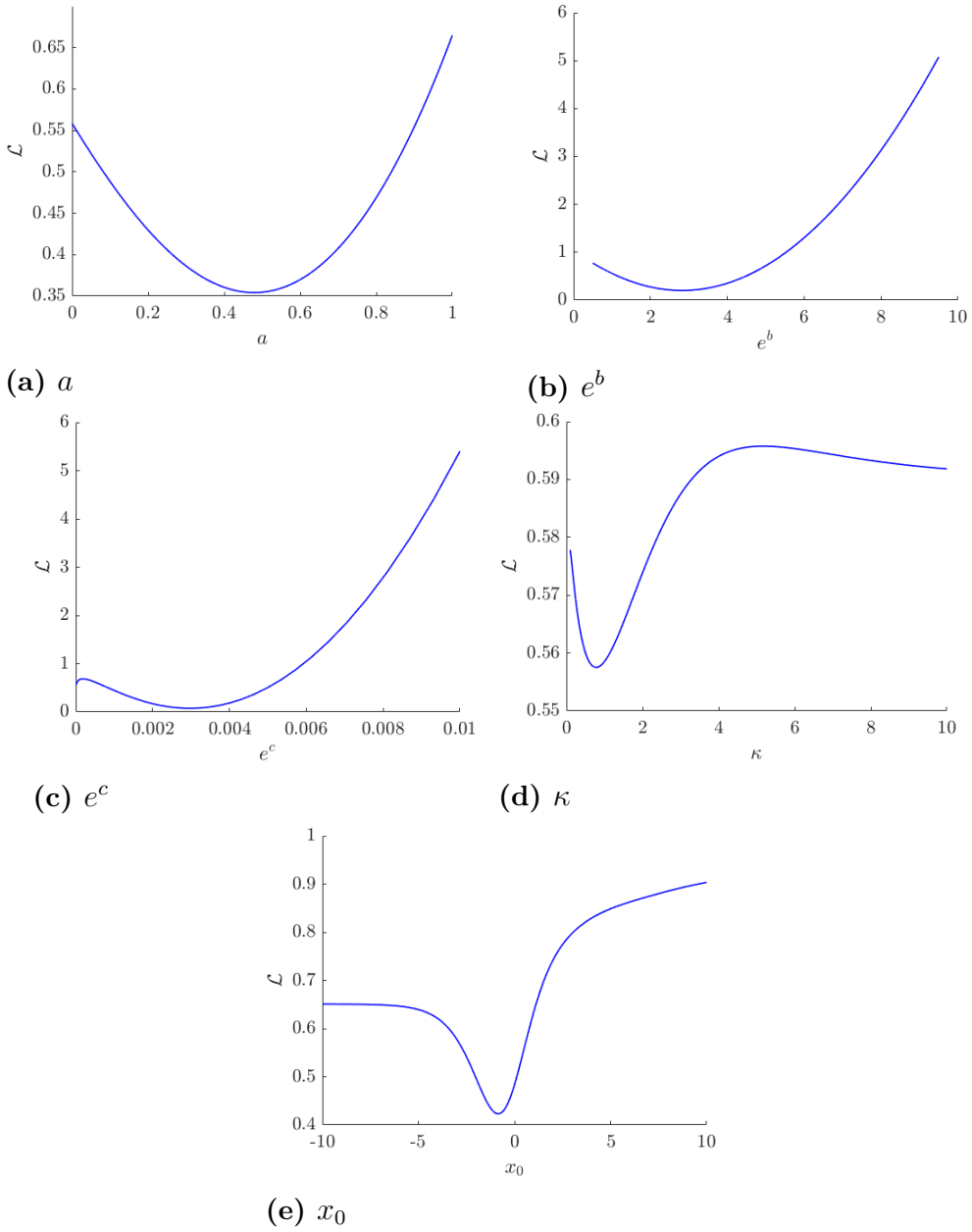
(a) a

(b) b



(c) c

Figure 4.3: The loss function as a function of the regularising parameters, using the regularising function $g_p^{\log}(d) \coloneqq a - bc + b\log(d + e^c)$ for $a = 0$, $b = 2$ and $c = 1$ as original input, then varying the parameters one at the time.



(a) a **(b)** e^b **(c)** e^c **(d)** κ **(e)** x_0

Figure 4.4: The loss function as a function of the regularising parameters, using the regularising function $g_p^{\text{ls}}(d) \coloneqq a+e^c d+\frac{e^b}{1+e^{-\kappa(d-x_0)}}$ for $a = 0$, $b = 0.5$, $c = \log(0.00001)$, $\kappa = 1$ and $x_0 = 0.5$ as original input, then varying the parameters one at the time.

seems to influence $\mathcal{L}$ the least. It is clear that choosing a, b or c too small or too large might result in $\mathcal{L}$ – and thus $\mathscr{L}$ – becoming very large.
As for the log regularising function, it is given by

$$g_p^{\log}(d) \coloneqq a - bc + b\log(d + e^c) \quad .$$

Its dependency of the parameters a, b and c is displayed in fig. 4.3. It is clear that there exists a singularity slightly below $a = -0.7$: $\mathcal{L}$ quickly tends to infinity here, due to too many eigenvalues becoming too small for such choices of a. The behaviour of $\mathcal{L}$ seems nice enough when varying b: it decreases exponentially as b increases. Numerical experiments shows that $\mathcal{L}$ explodes to infinity as soon as $b < 0$ is fed into the function though. $\mathcal{L}$ grows rapidly when picking $c < -5$ but flattens out when $c \to \infty$.
The linear-sigmoid regulariser is given by

$$g_p^{\text{ls}}(d) \coloneqq a + e^c d + \frac{e^b}{1 + e^{-\kappa(d - x_0)}} \quad .$$

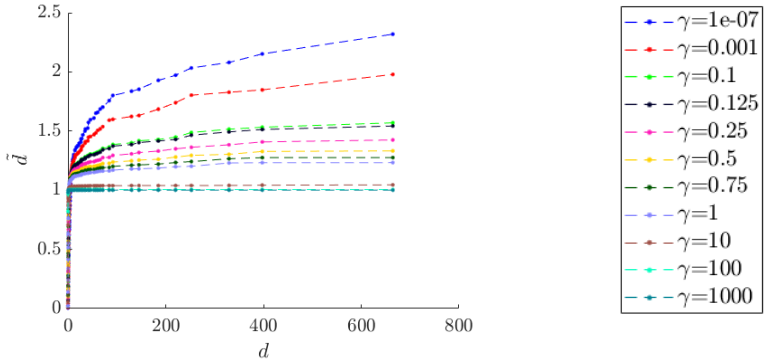
Its dependency of the parameters is displayed in fig. 4.4. Comparison with the plots of the sigmoid regulariser g_p^{sigmoid} in fig. 4.2 shows that g_p^{ls} depends upon the parameters a, b, κ and x_0 in the same way as g_p^{sigmoid} does. This is to be expected, since g_p^{ls} is simply the sigmoid regulariser with an extra linear term added to it. The plot in fig. 4.4c displays that $\mathcal{L}$ is extremely sensitive to changes in the coefficient e^c. The value of $\mathcal{L}$ explodes as $e^c > 0.004$.

4.1.3 Regarding the choice of spectral mapping

The attentive reader has already noticed that the regularising functions g_p^{normal} and g_p^{linear} listed in section 3.3 have not been mentioned in the preceding section. A few words should therefore be given regarding the choice of spectral mapping, and what its consequences are for the convergence of the numerical integrals of the loss function $\mathcal{L}$.
The normal regulariser g_p^{normal}, defined in section 3.3.4, has been excluded from this chapter since optimisation over its parameters would result in a function of identical shape as the sigmoid function. Including the full results obtained from application of g_p^{normal} is therefore completely redundant.
The linear regulariser g_p^{linear} defined in section 3.3.4 has been excluded from this chapter due to the resulting integrals not converging. (Remember from the preceding section the influence of the linear term of the linear-sigmoid regulariser on $\mathcal{L}$!) Fact is that this was the case for many other attempts of spectral mappings that were tried out during the project. For more information on this part, see chapter 5.

4.1.4 The eigenvalues of the penalised MLE

A main concern during the implementation of the risk-targeting estimator was the choice of the spectral mapping g_p. In order to find some inspiration on possible functions, the regularised eigenvalues of the l_2-penalised maximum likelihood estimator (section 2.5.2) were examined in relation to the eigenvalues of the unregularised



(a) Plot of eigenvalues

(b) Colour chart of the plot

Figure 4.5: The eigenvalues of the l_2-penalised maximum likelihood estimator as a function of the eigenvalues of the unregularised sample precision matrix estimator for a few different values of the penalty parameter.

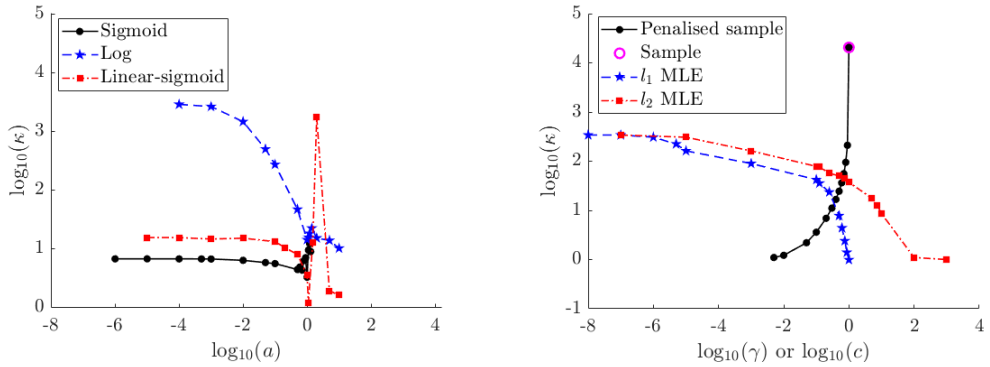
estimator. This choice was made due to the fact that the l_2-penalised maximum likelihood estimator is rotation invariant, which allows for comparison of the eigenvalues straight away. A sample covariance matrix S was therefore computed out of the market data and used as input for the l_2-penalised maximum likelihood estimator (see section 2.5.2) $\tilde{\Theta}^{\text{QML2}}$, whose eigenvalues are denoted $\tilde{d}$ in fig. 4.5a. The eigenvalues d of the unregularised estimator $\hat{\Theta} = S^{-1}$ were computed as $d = 1/s$ where s is an eigenvalue of S. The procedure was repeated for all penalty parameters γ used throughout the project. The result is displayed in fig. 4.5. It seems that d could be mapped to $\tilde{d}$ by some logarithmic function. This was the inspiration behind the spectral mapping $g_p^{\log}$ defined in section 3.3.3.

4.2 Performance of the estimators

In this section information regarding the performance of the various estimators is given. Quantities of interest are – apart from the total loss function $\mathscr{L}$ – $\hat{q}$, $\bar{Z}_{\text{oos}}$ and σ_{oos} as explained in section 3.1. All quantities are given in relation to the condition number κ of the estimator, since this number serves as an indication of "how much" regularisation that has been applied. The smaller the κ, the more regularised is the estimator. Full results are given for all sets of parameters in section 4.2.1. The results are summed up and compared in section 4.2.2, using the best set of parameters for each estimator.

4.2.1 Performance for all sets of parameters

The estimators from chapter 2 were computed using 250 days of training data. For the penalised estimators (maximum likelihood estimators and the simple sample estimator) the penalty parameters were varied. For the risk targeting estimator, the parameter a of the spectral mapping was varied. This was done in an attempt of trying to create estimators of different condition number, so as to be able to draw



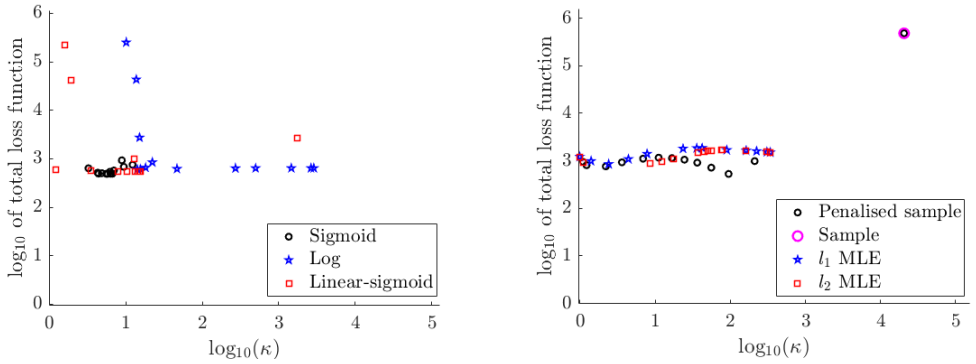
(a) Risk-targeting estimators **(b)** Reference estimators

Figure 4.6: The condition number of the estimators as a function of the parameter a (risk-targeting estimators), γ (maximum likelihood estimators) or c (penalised sample estimator) plotted in a log-log scale. To the left: risk-targeting estimators obtained from using different spectral mappings g_p. To the right: reference estimators.

conclusions about how the values of $\mathscr{L}$, $\hat{q}$, $\bar{Z}_{\text{oos}}$ and σ_{oos} varies when the amount of regularisation changes. The strategy was more successful for the penalised estimators, due to their simpler nature: since the parameters p defining the risk-targeting estimator were found automatically by the quasi-Newton algorithm it turned out to be difficult to manipulate the condition number. The choice of keeping the parameter a fixed, i.e. not to be optimised over, was made in order to try to regain some control. The parameters used for the different estimators are listed in appendix B along with the corresponding condition number.

The condition number κ is plotted as a function of the parameters γ (maximum likelihood estimator), c (simple penalised sample estimator) and a (risk-targeting estimator) respectively in fig. 4.6. It is from the figure clear that the condition number of the penalised estimators varies more than the condition number of the risk-targeting estimators. The maximum-likelihood estimators displays a similar behaviour despite being penalised somewhat differently, even though the condition number drops quicker for the l_1-penalised one when the penalty parameter γ is increased. As for the risk-targeting estimators, fig. 4.6a shows that the logarithmic spectral mapping allows for a smooth adjustment of the condition number. The sigmoid mapping regularises quite a lot, since the condition number of the resulting estimator is always quite small.

The total loss function $\mathscr{L}$ is plotted as a function of the condition number in fig. 4.7. The corresponding numbers are tabulated in Appendix B. Remember from section 3.1 that $\mathscr{L}$ should be as small as possible. It is from the figure difficult to draw any conclusions regarding the performance of the risk-targeting estimators in relation to the amount of regularisation applied. The sigmoid spectral mapping seems to yield the best results among the risk-targeting estimators, while the penalised sample estimator outperforms both maximum likelihood estimators among



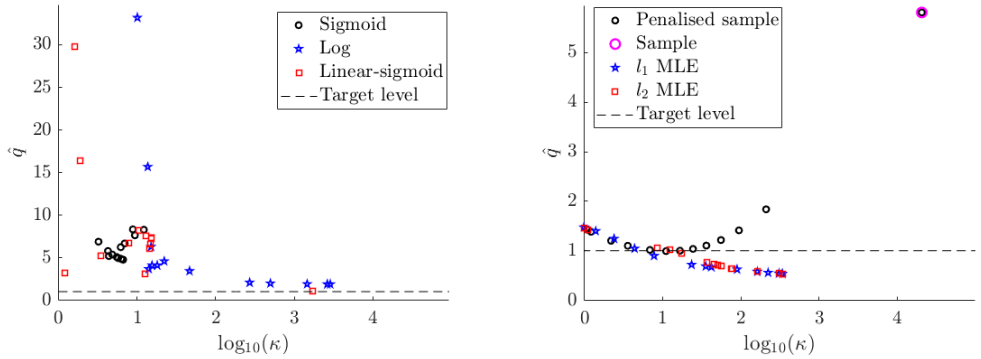
(a) Risk-targeting estimators **(b)** Reference estimators

Figure 4.7: The total loss function $\mathscr{L}$ as a function of the condition number of the corresponding estimator, plotted in a log-log scale. To the left: risk-targeting estimators obtained from using different spectral mappings g_p. To the right: reference estimators.

the penalised estimators. Close inspection of the numerical values (see Appendix B) reveals that the smallest value of $\mathscr{L}$ achieved is 489, obtained by the risk-targeting estimator using a sigmoid function with parameters $a = 0.05$, $e^b = 2.39$, $k = 9.66$ and $x_0 = 2.20$. The condition number of the estimator is 5.8 which indicates that the amount of regularisation is large. The second smallest value of $\mathscr{L}$ is 530, attained by the penalised sample estimator using a penalty parameter $c = 0.8$. The condition number of the estimator is 95, which indicates that the amount of regularisation applied is intermediate. The lowest values of $\mathscr{L}$ achieved by the rest of the estimators are 534 (risk-targeting estimator using the linear-sigmoid function), 608 (risk-targeting estimator using the log mapping), 856 (maximum likelihood estimator with l_1-penalty) and 898 (maximum likelihood estimator with l_2-penalty). The unregularised sample estimator results in $\mathscr{L}$ becoming $4.8 \cdot 10^4$.

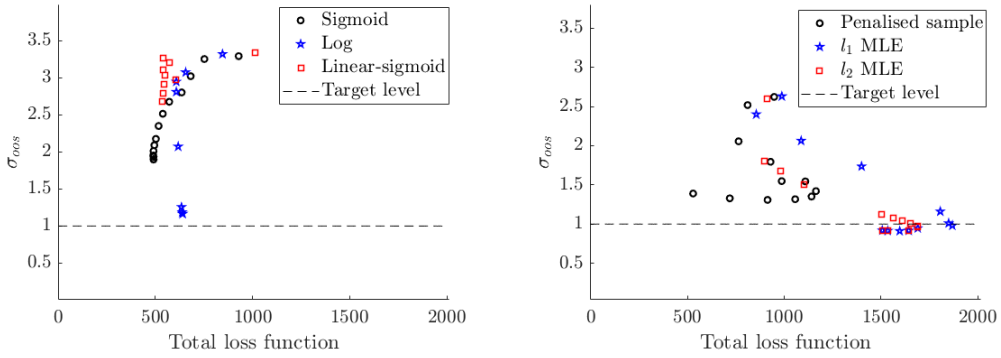
The tendency $\hat{q}$ of over- or underestimation of the risk is plotted as a function of the condition number in fig. 4.8. Remember from section 3.1 that $\hat{q}$ should be as close to one as possible. Having $\hat{q} < 1$ is preferred over having $\hat{q} > 1$, since $\hat{q} < 1$ indicates that the covariance (and thus the risk) in between markets is underestimated. It is from the figure clear that in general do the penalised estimators perform much better than the risk-targeting estimators – for details, see the tabulated values in Appendix B. The $\hat{q}$-values of the risk-targeting estimators are generally quite large, indicating that these estimators have a strong tendency of underestimating the market covariance.

The out-of-sample risk σ_{oos} of the Markowitz optimal portfolio w^* is plotted as a function of the total loss function $\mathscr{L}$ in fig. 4.9. The reason for displaying σ_{oos} against $\mathscr{L}$ – and not the condition number – is to check whether the chosen strategy of minimising $\mathscr{L}$ actually leads to $\sigma \xrightarrow{?} \xi$ out-of-sample, where σ is the risk. (Recall that minimising $\mathscr{L}$ is equivalent of letting $\sigma \to \xi$ in-sample.) In the numerical implementations the target risk level was chosen to be one, marked out using a



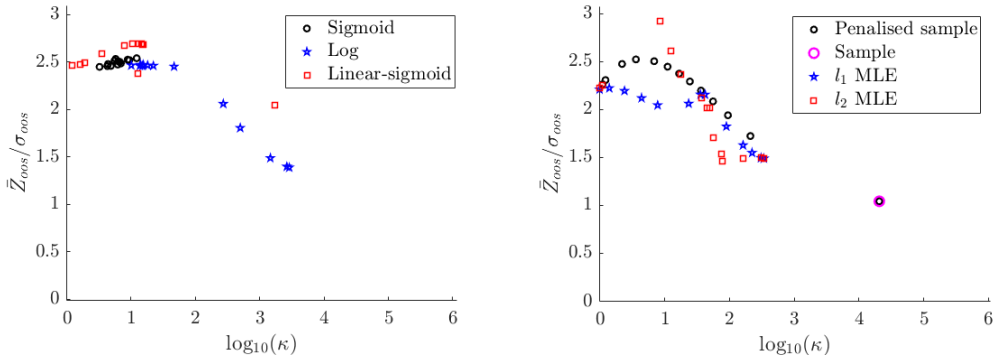
(a) Risk-targeting estimators **(b)** Reference estimators

Figure 4.8: The tendency $\hat{q}$ of over- or underestimation of the risk as a function of the condition number of the corresponding estimator, plotted in a lin-lin scale. To the left: risk-targeting estimators obtained from using different spectral mappings g_p. To the right: reference estimators. The target level of $\hat{q}$, 1, is marked out using a dashed line. Note that the y-axes are scaled differently in the two plots.



(a) Risk-targeting estimators **(b)** Reference estimators

Figure 4.9: The out-of-sample risk σ_{oos} as a function of the total loss function $\mathscr{L}$ plotted in a log-lin scale. To the left: risk-targeting estimators obtained from using different spectral mappings g_p. To the right: reference estimators. The target level ξ of σ_{oos}, chosen to be 1 in the computations, is marked out using a dashed line.



(a) Risk-targeting estimators **(b)** Reference estimators

Figure 4.10: The out-of-sample risk-adjusted average return $\bar{Z}_{\text{oos}}/\sigma_{\text{oos}}$ as a function of the condition number of the corresponding estimator plotted in a log-lin scale. To the left: risk-targeting estimators obtained from using different spectral mappings g_p. To the right: reference estimators.

dashed line in the figure. It is from the figure clear that the strategy works quite well for the risk-targeting estimators, especially for the one regularised by the sigmoid spectral mapping. No such tendency can be found among the penalised reference estimators. Unfortunately the out-of-sample risk σ_{oos} never reaches the target risk level for any of the risk-targeting estimators, something that is actually accomplished by several of the maximum likelihood estimators. The value of σ_{oos} is generally higher for the risk-targeting estimators than for the penalised ones. For detailed values of σ_{oos}, see Appendix B.

The out-of-sample risk-adjusted average return $\bar{Z}_{\text{oos}}/\sigma_{\text{oos}}$ for the Markowitz optimal portfolio $w^* = \frac{1}{\sqrt{\hat{\mu}'\tilde{\Theta}\hat{\mu}}}\tilde{\Theta}\hat{\mu}$, where the target risk has been chosen to be one ($\xi = 1$) and $\hat{\mu}$ is as defined in section 3.1.3, is plotted as a function of the condition number of the estimator $\tilde{\Theta}$ in fig. 4.10. The reason for inspecting $\bar{Z}_{\text{oos}}/\sigma_{\text{oos}}$ and not $\bar{Z}_{\text{oos}}$ is to truly inspect the *risk-adjusted* returns. From fig. 4.9 it is clear that generally $\sigma_{\text{oos}} \neq 1$, which indicates that $\bar{Z}_{\text{oos}}$ is not completely risk-adjusted.

Since w^* solves the problem of maximising the return Z for the risk level one, $\bar{Z}_{\text{oos}}/\sigma_{\text{oos}}$ should be as large as possible. It seems that for each type of estimator, a small condition number would result in a large $\bar{Z}_{\text{oos}}$. The relationship is more prominent for the penalised estimators than for the risk-targeting estimators. For the risk-targeting estimator obtained from using a sigmoid spectral mapping no such tendency is visible. For detailed values of $\bar{Z}_{\text{oos}}$ and σ_{oos}, see the tables in Appendix B.

4.2.2 Final result

In the preceding section scatter plots were displayed, showing how the performance of the estimators varies when the defining parameters are altered. Detailed numbers corresponding to the plots are tabulated in Appendix B. From these tables the performance of the best estimator of each type has been extracted, "the best estimator" is simply the estimator with the lowest value of the total loss function $\mathscr{L}$.

Estimator	$\mathscr{L}$	$\mathbf{q}$	$\bar{\mathbf{Z}}_{\text{oos}}$	σ_{oos}	κ
Sigmoid regularised	**489**	3.22	4.94	1.95	5.75
Log regularised	608	3.66	6.94	2.81	14
Linear-sigmoid regularised	534	6.08	**7.21**	2.68	15
MLE with l_1-penalty	856	1.24	5.27	2.4	2.4
MLE with l_2-penalty	898	**1.07**	4.65	1.8	8.6
Sample with simple l_2-penalty	530	1.41	2.70	**1.39**	95
Unregularised sample	$4.8 \cdot 10^5$	5.8	3.33	3.20	$2 \cdot 10^4$

Table 4.2: Comparison of the estimators, using the best set of parameters/ the best penalty parameter as found in section 4.2.2. The best value of each quantity $\mathscr{L}$, $\hat{q}$, $\bar{Z}_{\text{oos}}$, σ_{oos} is marked out using bold text.

The results are summed up in 4.2 for a simple comparison of the various estimators. It is from table 4.2 clear that in general it is the risk-targeting estimators that attain the lowest values of $\mathscr{L}$, even though the penalised sample estimator also performs very well in this sense. Somewhat worrying is the fact that all the risk-targeting estimators results in $\hat{q}$ being quite large, which indicates that they all tend to underestimate the covariance in between markets. The penalised estimators perform much better in this sense: they all have values of $\hat{q} \in [1, 2]$. The out-of-sample risk of their corresponding Markowitz portfolio are generally slightly closer to the target level, 1, than the out-of-sample risk of the Markowitz portfolios given by the risk-targeting estimators. As for the out-of-sample return, it is difficult to make any general comparisons between the two groups of estimators. Taking a look at the condition numbers of the estimators reveal that most estimators are quite heavily regularised: having condition numbers ranging from 2.4 to 15. The exception is the penalised sample precision matrix, whose condition number is 95.

4.3 Summing up

In this chapter the results of the project were presented. The chapter was initialised with a demonstration of how the loss function $\mathcal{L}$ depends upon the individual eigenvalues of the precision matrix estimator. It was concluded that $\mathcal{L}$ is minimised when the largest eigenvalue(s) is (are) shrunk and when eigenvalues of small to intermediate size are enlarged. It was also very clear that changes of the largest eigenvalue affects the value $\mathcal{L}$ the most – just look at the scale of then y-axes in fig. 4.1.
The second matter dealt with was somewhat similar: an investigation of the influence of the individual parameters of the spectral mapping upon the value of $\mathcal{L}$. It was concluded that the parameters had to be chosen carefully, since the numerical integrals of $\mathcal{L}$ might not converge otherwise. Especially delicate is the choice of constant in front of any linear term, which might explain why the (fully) linear regulariser defined in section 3.3.4 did not converge at all.
The presentation of the intermediate results was ended with a plot displaying the eigenvalues of the l_2-penalised maximum likelihood estimator as a function of the eigenvalues of the unregularised sample estimator, intended to serve as inspiration

for possible spectral mappings. The reason for using the l_2-penalised maximum likelihood estimator is that it is rotation invariant, i.e. it suffices to look at its eigenvalues only. The resulting plot revealed that it might be a good idea to investigate log-shaped spectral mappings, for what reason the mapping $g_p^{\log}$ (see section 3.3.3) was constructed.
In the final section, the performance of the estimators was displayed for various sets of defining parameters. The results were visualised as functions of the condition number of the estimator using scatter plots, since the condition number indicates how heavily the estimators have been regularised. The out-of sample risk was plotted as a function of the total loss function $\mathscr{L}$, so as to check whether the strategy of minimising $\mathscr{L}$ would actually result in the out-of-sample risk getting closer to the target risk level. It seems to work well for the risk-targeting estimators, but no such trend is visible among the reference estimators. On the other hand is the out-of-sample risk of the risk-targeting estimators much larger than the corresponding quantity of the reference estimators, despite giving rise to lower values of the total loss function $\mathscr{L}$. The risk-targeting estimators also show a greater tendency of underestimating the total risk compared with the reference estimators. The performance of a risk-targeting estimator is highly dependent of the choice of the spectral mapping, g_p.

5

Discussion and Conclusion

In this thesis a new estimator of the precision matrix was proposed, intended to target a pre-determined level of risk for a Markowitz optimal portfolio. It is rotation-invariant, which implies that regularisation can be performed by direct manipulation of its eigenvalues. A number of different spectral mappings were tested for this purpose, a few of them have been listed in this report. A simple penalised sample estimator and two maximum likelihood estimators were used as references.

5.1 The spectral mapping is a sensitive matter

It turned out that the choice of spectral mapping is of great importance for the strategy to be successful. Unfortunately the integrals of the loss function are very sensitive for manipulation of the eigenvalues, for what reason many of the mappings tried out resulted in the loss function not converging. Examples of mappings that were tried out unsuccessfully are other parameterisations of the sigmoid function, other parameterisations of logarithmic functions and the exponential function of various polynomials. They have not been included in this report.
It was noted that extra care must be taken to ensure that the small eigenvalues aren't mapped to very small values, since this would typically result in singularities occurring. It is also of importance to check the asymptotics (i.e. the behaviour of $g_p(d)$ as $d \to \infty$) of any proposed mapping, since unfortunate combinations of the parameters p might result in large eigenvalues blowing up. A lot of time was spent on manually trying to patch suitable functions together, and on trying to find good combinations of the parameters that could be used as initial values for the quasi-Newton optimisation routine. Remember from the plots in section 4.1 that there might exist several local minima where the routine might get stuck, for what reason it is important to try to start in a good place.

5.2 Generalises out-of-sample to some extent

The underlying idea when defining the total loss function was to minimise the expected distance between the target risk and the actual risk of a Markowitz optimal portfolio, then hope that this property generalises out-of-sample. Then and only then would the estimator be truly risk-targeting. It is from fig. 4.9a clear that the strategy does indeed seem to work out as planned: for the risk-targeting estimators, there is clearly a connection between an estimator having a low value of the total loss

function and having an out-of-sample risk closer to the target level. This connection does not exist for any of the reference estimators (see fig. 4.9b).

5.3 Future work

Unfortunately, most of the risk-targeting estimators do underestimate the out-of-sample risk quite severely. This is indicated by them taking on values $\sigma_{\text{oos}} > \xi$, where ξ is the target risk level, and $\hat{q} > 1$. Since the performance of the risk-targeting estimators are highly dependent of the choice of spectral mapping, other spectral mappings should be investigated. An attempt of constructing a better spectral mapping in a more sophisticated manner was made very late in the project. The projection of the eigenvalues of the out-of-sample precision matrix onto the eigendirections of the in-sample precision matrix was examined. Let the unregularised out-of-sample precision matrix be denoted $\hat{\Theta}^{\text{oos}}$ and let the unregularised in-sample precision matrix be denoted $\hat{\Theta}^{\text{is}}$. Perform an eigendecomposition of the latter,

$$\hat{\Theta}^{\text{is}} = H^{\text{is}} D (H^{\text{is}})' \quad ,$$

then project the out-of-sample eigenvalues onto the in-sample eigendirections $H_1^{\text{is}}, ..., H_N^{\text{is}}$:

$$r_i \coloneqq (H_i^{\text{is}})' \hat{\Theta}^{\text{oos}} H_i^{\text{is}} \quad .$$

The spectral mapping g_p was then to be chosen such that $g_p(r_i) = d_i$, where d_i is the i^{th} eigenvalue of $\hat{\Theta}^{\text{is}}$. An attempt of finding a suitable g_p was made by fitting a function g_p^{-1} to a plot displaying $\{r_i\}_1^N$ versus $\{d_i\}_1^N$, then obtaining the spectral mapping g_p as the inverse of g_p^{-1}. Due to shortage of time, no mapping g_p that resulted in the loss function converging could be found. It is an interesting strategy though and should be investigated further.

5.4 Conclusion

Out of all the estimators that were tried out, it is the risk-targeting estimator obtained from using the sigmoid spectral mapping that minimises the total loss function. The sigmoid mapping has some nice properties related to the convergence of the numerical integrals: as long as the images of the smallest eigenvalues aren't too small, the resulting integrand is quite well-behaved. Unfortunately, this estimator does generalise the risk-targeting property quite poorly out-of-sample. Since the risk-targeting estimators perform very differently depending on the spectral mapping used when regularising them, it is likely that better performance out-of-sample could be achieved by another spectral mapping. A strategy for finding such a mapping has been briefly tried out, but should be looked into further.

Bibliography

[1] http://www.lynxhedge.se (visited on 04/11/2019).

[2] H. Markowitz, "Portfolio selection", The Journal of Finance **7**, 77–91 (1952).

[3] O. Ledoit and M. Wolf, "Nonlinear shrinkage of the covariance matrix for portfolio selection: Markowitz meets Goldilocks", Working Paper **137** (2017).

[4] J. Hull, *Fundamentals of futures and options markets*, 1st ed. (Pearson, Edinburgh Gate, Harlow, England, 2014).

[5] T. Rydén, Private communication with supervisor, 2019.

[6] P. Cirillo, "Lecture notes for the course in financial mathematics", Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Feb. 2018.

[7] G. Stevens, "On the inverse of the covariance matrix in portfolio analysis", The Journal of Finance **LIII**, 1821–1827 (1998).

[8] S. Goto and Y. Xu, "Improving mean variance optimization through sparse hedging restrictions", Journal of Financial and Quantitative Analysis **50**, 1415–1441 (2015).

[9] L. Rosasco, *Spectral regularization*, http://www.mit.edu/~9.520/spring09/Classes/class07_spectral.pdf (visited on 04/17/2019).

[10] R. T. T.Hastie and M. Wainwright, *The elements of statistical learning - data mining, inference and prediction*, 2nd ed. (Springer, Berlin, Germany, 2009).

[11] J. Friedman, T. Hastie and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso", Biostatistics **9**, 432–441 (2008).

[12] N. Cressie, A.S. Davis and J. Leroy Folks, "The moment-generating function and negative integer moments", The American Statistician **35**, 148–150 (1981).

[13] S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers", Foundations and Trends in Machine Learning **3**, 1–122 (2010).

[14] D. G. Luenberger, *Linear and nonlinear programming*, 2nd ed. (Addison-Wesley, Reading, Massachusetts, 1984).

[15] *Broyden–fletcher–goldfarb–shanno algorithm*, https://en.wikipedia.org/wiki/Broyden%5C%E2%5C%80%5C%93Fletcher%5C%E2%5C%80%5C%93Goldfarb%5C%E2%5C%80%5C%93Shanno_algorithm (visited on 04/29/2019).

[16] D. Yang and Q. Zhang, "Drift-independent volatility estiamtion based on high, low, open, and close prices", The Journal of Business **73**, 477–492 (2000).

A

Proofs and derivations

Below follows proofs and derivations of some of the results referred to throughout the report. Unless otherwise stated, the notation and the mathematical framework is as given in section 2.2.

Lemma A.0.1 (Markowitz optimal portfolio 1)**.** *The optimal solution of the problem in eq.* (2.3) *is given by*

$$w^* \propto \Sigma^{-1}\mu \quad .$$

Proof. The problem from eq. (2.3) is a minimisation problem with one constraint. It is solved by application of the method of Lagrange multipliers. The Lagrangian corresponding to eq. (2.3) reads

$$L(w, \alpha) = w'\Sigma w - \alpha(w'\mu - \lambda) \quad ,$$

where α is a Lagrange multiplier. It is clear that eq. (2.3) has an optimal solution when $\nabla_{(w,\alpha)} L(w, \alpha) = 0$, which is equivalent to

$$\nabla_w L(w, \alpha) = 0 \tag{A.1a}$$
$$w'\mu - \lambda = 0 \quad . \tag{A.1b}$$

Expanding and reordering of eq. (A.1a) yields

$$\begin{aligned} & \nabla_w L(w, \alpha) = 2\Sigma w - \alpha(w'\mu - \lambda) = 0 \\ \iff \quad & w^* = \frac{\alpha}{2}\Sigma^{-1}\mu \quad . \end{aligned}$$

Plugging the above expression of w^* into the constraint equation eq. (A.1b) and solving for α yields

$$\alpha = \frac{2\lambda}{u'\Sigma^{-1}\mu} \quad ,$$

which finally gives

$$w^* = \frac{\lambda}{u'\Sigma^{-1}\mu}\Sigma^{-1}\mu \propto \Sigma^{-1}\mu \quad .$$

□

Lemma A.0.2 (Markowitz optimal portfolio 2)**.** *The optimal solution of the problem in eq.* (2.4) *is given by*

$$w^* \propto \Sigma^{-1}\mu \quad .$$

Proof. The problem from eq. (2.4) is a maximisation problem with one constraint. The corresponding Lagrangian is

$$L(\alpha, w) = w'\mu - \alpha(w'\Sigma w - \xi^2) \quad ,$$

where α is a Lagrange multiplier. Eq. 2.4 has an optimal solution when

$$\nabla_w L(w, \alpha) = 0 \tag{A.2a}$$
$$w'\Sigma w - \xi^2 = 0 \quad . \tag{A.2b}$$

Expanding and reordering of eq. (A.2a) yields

$$\begin{aligned} & \nabla_w L(w, \alpha) = \mu - 2\alpha\Sigma w = 0 \\ \iff \quad & w^* = \frac{1}{2\alpha}\Sigma^{-1}\mu \quad . \end{aligned}$$

Plugging the above expression for w^* into the constraint equation eq. (A.2b) and solving for α yields

$$\alpha = \frac{1}{2\xi}\sqrt{\mu'\Sigma^{-1}\mu} \quad ,$$

which finally gives

$$w^* = \frac{\xi}{\sqrt{\mu'\Sigma^{-1}\mu}}\Sigma^{-1}\mu \propto \Sigma^{-1}\mu \quad .$$

□

Lemma A.0.3 (Expectations of fractions)**.** *Let X and Y be positive random variables with joint moment-generating function ϕ_{XY}. Then*

$$\mathbf{E}\left[\frac{Y}{X}\right] = \int_0^\infty \lim_{t_2 \to 0}\left[\frac{\partial}{\partial t_2}\phi_{XY}(-t_1, t_2)\right] dt_1 \quad , \tag{A.3}$$
$$\mathbf{E}\left[\left(\frac{Y}{X}\right)^2\right] = \int_0^\infty \lim_{t_2 \to 0}\left[\frac{\partial^2}{\partial t_2^2}\phi_{XY}(-t_1, t_2)\right] dt_1 \quad . \tag{A.4}$$

Proof. Let the distributions of X and Y be denoted F_X and F_Y respectively. Start by showing the identity in eq. (A.3).

By definition of the expected value,

$$\begin{aligned} \mathbf{E}\left[\frac{Y}{X}\right] &= \int_0^\infty \int_0^\infty \frac{y}{x} dF_X(x) dF_Y(y) = \left\{\text{note that } \frac{1}{x} = \int_0^\infty e^{-t_1 x} dt_1, \text{ swap order of integration}\right\} = \\ &= \int_0^\infty \int_0^\infty \int_0^\infty y e^{-t_1 x} dF_X(x) dF_Y(y) dt_1 = \int_0^\infty \mathbf{E}[Y e^{-t_1 X}] dt_1 = \\ &= \int_0^\infty \lim_{t_2 \to 0} \mathbf{E}[Y e^{-t_1 X + t_2 Y}] dt_1 = \int_0^\infty \lim_{t_2 \to 0} \frac{\partial}{\partial t_2} \mathbf{E}[e^{-t_1 X + t_2 Y}] dt_1 = \\ &= \int_0^\infty \lim_{t_2 \to 0} \frac{\partial}{\partial t_2} \phi_{XY}(-t_1, t_2) dt_1 \quad , \end{aligned}$$

which proves eq. (A.3).

The identity in eq. (A.4) is proven similarly:

$$\begin{aligned}\mathbf{E}\left[\left(\frac{Y}{X}\right)^2\right] &= \int_0^\infty \int_0^\infty \frac{y^2}{x^2} dF_X(x) dF_Y(y) = \left\{\frac{1}{x^2} = \int_0^\infty t_1 e^{-t_1 x} dt_1\right\} = \\ &= \int_0^\infty t_1 \mathbf{E}\left[Y^2 e^{-t_1 X}\right] dt_1 = \int_0^\infty t_1 \lim_{t_2 \to 0} \frac{\partial^2}{\partial t_2^2} \mathbf{E}\left[e^{-t_1 X + t_2 Y}\right] dt_1 = \\ &= \int_0^\infty t_1 \lim_{t_2 \to 0} \frac{\partial^2}{\partial t_2^2} \phi_{XY}(-t_1, t_2) dt_1 \quad .\end{aligned}$$

□

Lemma A.0.4 (Derivatives of the moment-generating function)**.** *Let X and Y be the quadratic forms of a standard multivariate normal distributed random vector in $\mathbf{R}^n$, i.e. let $X = U'QU$ and $Y = U'RU$ for $U \sim \mathcal{N}_n(0, I)$ and $Q, R \in \mathbf{R}^{n\times n}$. Let ϕ_{XY} be their joint moment-generating function. Then*

$$\frac{\partial}{\partial t_2}\phi_{XY}(t_1, t_2) = |I - 2t_1Q - 2t_2R|^{-1/2}\mathbf{E}_U[U'L'RLU] \quad , \tag{A.5}$$

$$\frac{\partial^2}{\partial t_2^2}\phi_{XY}(t_1, t_2) = |I - 2t_1Q - 2t_2R|^{-1/2}\mathbf{E}_U[(U'L'RLU)^2] \quad . \tag{A.6}$$

where $L \in \mathbf{R}^{n\times n}$ such that $LL' = (I - 2t_1Q - 2t_2R)^{-1}$.

Proof. By definition of the moment-generating function,

$$\begin{aligned}\frac{\partial}{\partial t_2}\phi_{XY}(t_1, t_2) &= \frac{\partial}{\partial t_2}\mathbf{E}_{X,Y}[e^{t_1X+t_2Y}] = \mathbf{E}_{X,Y}[Ye^{t_1X+t_2Y}] = \mathbf{E}_U[U'RU \cdot e^{U'(t_1Q+t_2R)U}] = \\ &= \int_{\mathbf{R}^n} u'Ru \cdot e^{u'(t_1Q+t_2R)u}dF_U(u) = \frac{1}{\sqrt{2\pi|I|}}\int_{\mathbf{R}^n} u'Ru \cdot e^{-\frac{1}{2}u'(I-2t_1Q-2t_2R)u}du \quad .\end{aligned}$$

Now take L such that

$$LL' \coloneqq (I - 2t_1Q - 2t_2R)^{-1} \quad ,$$

and perform a change of variables $W \coloneqq L^{-1}U$. Then

$$\begin{aligned}\frac{\partial}{\partial t_2}\phi_{XY}(t_1, t_2) &= |L|\int_{\mathbf{R}^n} \frac{1}{\sqrt{2\pi}}w'L'RLw \cdot e^{-\frac{1}{2}w'w}dw = \{\text{For } W \sim \mathcal{N}_n(0, I)\} = \\ &= |L|\mathbf{E}_W[W'L'RLW] = |L|\mathbf{E}_U[U'L'RLU] \quad .\end{aligned}$$

Noting that

$$\begin{cases} |LL'| = |I - 2t_1Q - 2t_2R|^{-1} & \text{and} \\ |LL'| = |L| \cdot |L'| = |L|^2 \end{cases}$$

yields $|L| = |I - 2t_1Q - 2t_2R|^{-1/2}$. This concludes eq. (A.5). Eq. (A.6) is proven in the same manner. □

Lemma A.0.5 (Moments of quadratic forms)**.** *Let the n-dimensional random vector $W \sim \mathcal{N}_n(0, I)$ and let $A \in \mathbf{R}^{n\times n}$ symmetric. Then*

$$\mathbf{E}_W[W'AW] = tr(A) \quad , \tag{A.7}$$

$$\mathbf{E}_W\left[(W'AW)^2\right] = 2tr(A^2) + tr(A)^2 \quad . \tag{A.8}$$

Proof. The proof is carried out by a applying the definition of the expected value and noting that the moments of a (univariate) standard normal variable W_i are

$$\mathbf{E}[W_i] = 0, \quad \mathbf{E}[W_i^2] = 1, \quad \mathbf{E}[W_i^3] = 0 \quad \text{and} \quad \mathbf{E}[W_i^4] = 3 \quad .$$

Starting off with eq. (A.7),

$$\mathbf{E}_W[W'AW] = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\mathbf{E}_W[W_iW_j] = \sum_{i=1}^{n}\left(A_{ii}\mathbf{E}_W[W_i^2] + \sum_{\substack{j=1\\j\neq i}}^{n} A_{ij}\mathbf{E}_W[W_iW_j]\right) =$$

$$= \sum_{i=1}^{n}\left(A_{ii} + \sum_{\substack{j=1\\j\neq i}}^{n} A_{ij}\cdot 0\right) = \sum_{i=1}^{n} A_{ii} = tr(A) \quad .$$

Moving on to eq. (A.8) requires more work:

$$\mathbf{E}_W\left[(W'AW)^2\right] = \mathbf{E}_W\left[(W'AW)(W'AW)\right] = \mathbf{E}_W\left[\left(\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}W_iW_j\right)\left(\sum_{k=1}^{n}\sum_{l=1}^{n} A_{kl}W_kW_l\right)\right] =$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} A_{ij}A_{kl}\mathbf{E}_W[W_iW_jW_kW_l] =$$

$$= \sum_{i=1}^{n} A_{ii}^2\mathbf{E}_W[W_i^4] + \sum_{i=1}^{n}\sum_{\substack{l=1\\l\neq i}}^{n} A_{ii}A_{il}\mathbf{E}_W[W_i^3]\mathbf{E}_W[W_l] +$$

$$+ \sum_{i=1}^{n}\sum_{\substack{k=1\\k\neq i}}^{n} A_{ii}A_{ki}\mathbf{E}_W[W_i^3]\mathbf{E}_W[W_k] + \sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq i}}^{n} A_{ij}A_{ii}\mathbf{E}_W[W_i^3]\mathbf{E}_W[W_j] +$$

$$+ \sum_{i=1}^{n}\sum_{\substack{k=1\\k\neq i}}^{n}\sum_{\substack{l=1\\l\neq i}}^{n} A_{ii}A_{kl}\mathbf{E}_W[W_i^2]\mathbf{E}_W[W_kW_l] + \sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq 1}}^{n}\sum_{\substack{l=1\\l\neq i}}^{n} A_{ij}A_{il}\mathbf{E}_W[W_i^2]\mathbf{E}_W[W_jW_l] +$$

$$+ \sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq i}}^{n}\sum_{\substack{k=1\\k\neq i}}^{n} A_{ij}A_{ki}\mathbf{E}_W[W_i^2]\mathbf{E}_W[W_jW_k] + \sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq i}}^{n}\sum_{\substack{k=1\\k\neq i}}^{n}\sum_{\substack{l=1\\l\neq i}}^{n} A_{ij}A_{kl}\mathbf{E}_W[W_i]\mathbf{E}_W[W_jW_kW_l] =$$

$$= 3\sum_{i=1}^{n} A_{ii}^2 + 0 + 0 + 0 + \sum_{i=1}^{n}\sum_{\substack{k=1\\k\neq i}}^{n} A_{ii}A_{kk} + 2\sum_{i=1}^{n}\sum_{\substack{j=1\\j\neq i}}^{n} A_{ij}^2 + 0 =$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ii}A_{jj} + 2\sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}^2 = tr(A)^2 + 2tr(A^2) \quad .$$

□

Theorem A.1 (Full expectation formulas)**.** *Let E_1 and E_2 be as previously stated, i.e. let*

$$E_1 = \mathbf{E}_\mu \left[\left(\frac{\mu' \hat{\Theta} \Sigma \hat{\Theta} \mu}{\mu' \hat{\Theta} \mu} \right)^2 \right] \quad \textit{and} \quad E_2 = \mathbf{E}_\mu \left[\left(\frac{\mu' \hat{\Theta} \Sigma \hat{\Theta} \mu}{\mu' \hat{\Theta} \mu} \right) \right]$$

for $\mu \sim \mathcal{N}_N(0, I)$, $\Sigma \in \mathbf{R}^{N \times N}$ and $\hat{\Theta} \in \mathbf{R}^{N \times N}$ where μ is the N-dimensional signal of expected risk-adjusted returns, Σ is the covariance matrix of the risk-adjusted returns and $\hat{\Theta}$ is an estimator of Σ^{-1}. Furthermore, let $\hat{\Theta} = HDH'$ where H is an orthonormal $N \times N$ matrix and D a diagonal $N \times N$ matrix with the (positive) eigenvalues $d_1, ..., d_N$ of $\hat{\Theta}$ on its main diagonal. Let L be such that $LL' = (I + 2t\hat{\Theta})^{-1}$. Then

$$E_1 = \int_0^\infty t|I + 2tD|^{-1/2} \left(2tr\Big((L'\hat{\Theta}\Sigma\hat{\Theta}L)^2 \Big) + tr\Big(L'\hat{\Theta}\Sigma\hat{\Theta}L \Big)^2 \right) dt = \tag{A.9}$$

$$= \int_0^\infty t \left(\prod_{i=1}^N \frac{1}{\sqrt{1 + 2td_i}} \right) \sum_{i=1}^N \sum_{j=1}^N \left(2[H'\Sigma H]_{ij} + [H'\Sigma H]_{ii}[H'\Sigma H]_{jj} \right) \frac{d_i^2 d_j^2}{(1 + 2td_i)(1 + 2td_j)} dt$$

$$E_2 = \int_0^\infty |I + 2tD|^{-1/2} tr\Big(L'\hat{\Theta}\Sigma\hat{\Theta}L \Big) dt = \tag{A.10}$$

$$= \int_0^\infty \left(\prod_{i=1}^N \frac{1}{\sqrt{1 + 2td_i}} \right) \sum_{i=1}^N [H'\Sigma H]_{ii} \frac{d_i^2}{(1 + 2td_i)} dt \quad .$$

Proof. Write $\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu = Y$ and $\mu'\Sigma\mu = X$. Then

$$E_1 = \mathbf{E}_{X,Y} \left[\left(\frac{Y}{X} \right)^2 \right] \quad \text{and} \quad E_2 = \mathbf{E}_{X,Y} \left[\frac{Y}{X} \right] \quad ,$$

a format that easily allows for application of Lemma A.0.3. Then letting

$$\hat{\Theta} = Q, \quad \hat{\Theta}\Sigma\hat{\Theta} = R \quad \text{and} \quad \mu = U$$

allows for using Lemma A.0.4. Thus

$$E_1 = \int_0^\infty t|I + 2t\hat{\Theta}|^{-1/2} \mathbf{E}_\mu \left[(\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu)^2 \right] dt \quad \text{and}$$
$$E_2 = \int_0^\infty |I + 2t\hat{\Theta}|^{-1/2} \mathbf{E}_\mu [\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu] dt \quad .$$

The expectations of the quadratic forms are evaluated using Lemma A.0.5, for $\mu = W$ and $L'\hat{\Theta}\Sigma\hat{\Theta}L = A$:

$$\mathbf{E}_\mu [(\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu)^2] = 2tr\Big((L'\hat{\Theta}\Sigma\hat{\Theta}L)^2 \Big) + tr\Big(L'\hat{\Theta}\Sigma\hat{\Theta}L \Big)^2$$

where

$$\begin{aligned}
tr\Big(L'\hat{\Theta}\Sigma\hat{\Theta}L\Big) &= tr\Big(LL'\hat{\Theta}\Sigma\hat{\Theta}\Big) = \{LL' = (I+2t\hat{\Theta})^{-1}, \quad \hat{\Theta} = HDH'\} = \\
&= tr\Big(H(I+2tD)^{-1}H'HDH'\Sigma HDH'\Big) = \{I = H'H = HH'\} = \\
&= tr\Big(D(I+2tD)^{-1}DH'\Sigma H\Big) = \sum_{i=1}^{N}\sum_{j=1}^{N}\Big[D(I+2tD)^{-1}D\Big]_{ij}\Big[H'\Sigma H\Big]_{ji} = \\
&= \{D, (I+2tD)^{-1} \text{ are diagonal } \Rightarrow \Big[D(I+2tD)^{-1}D\Big]_{ij} = 0 \text{ for } i \neq j\} = \\
&= \sum_{i=1}^{N}\Big[D(I+2tD)^{-1}D\Big]_{ii}\Big[H'\Sigma H\Big]_{ii} = \sum_{i=1}^{N}\frac{d_i^2}{(1+2td_i)}\Big[H'\Sigma H\Big]_{ii}
\end{aligned}$$

and similarly, by usage of the same tricks

$$\begin{aligned}
tr\Big((L'\hat{\Theta}\Sigma\hat{\Theta}L)^2\Big) &= ... = tr\Big((D(I+2tD)^{-1}DH'\Sigma H)^2\Big) = \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\Big[D(I+2tD)^{-1}DH'\Sigma H\Big]_{ij}\Big[D(I+2tD)^{-1}DH'\Sigma H\Big]_{ji} = \\
&= \{\text{diagonal matrices}\} = \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N}\Big[D(I+2tD)^{-1}D\Big]_{ii}\Big[D(I+2tD)^{-1}D\Big]_{jj}\Big[H'\Sigma H\Big]_{ij}\Big[H'\Sigma H\Big]_{ji} = \\
&= \{(H'\Sigma H)' = H'\Sigma H\} = \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{d_i^2 d_j^2}{(1+2td_i)(1+2td_j)}\Big[H'\Sigma H\Big]_{ij}^2 \quad .
\end{aligned}$$

Thus

$$\begin{aligned}
\mathbf{E}_\mu[(\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu)^2] &= \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{j=1}^{N}\Big(2[H'\Sigma H]_{ij}^2 + [H'\Sigma H]_{ii}[H'\Sigma H]_{jj}\Big)\frac{d_i^2 d_j^2}{(1+2td_i)(1+2td_j)} \quad , \\
\mathbf{E}_\mu[\mu'\hat{\Theta}\Sigma\hat{\Theta}\mu] &= \sum_{i=1}^{N}\frac{d_i^2}{(1+2td_i)}\Big[H'\Sigma H\Big]_{ii} \quad .
\end{aligned}$$

It remains to argue that $|I+2tD|^{-1/2} \stackrel{?}{=} \prod_{i=1}^{N}\frac{1}{\sqrt{1+2td_i}}$. Start by noting that $(I+2tD)$ is a diagonal matrix, each non-zero element being on the form $1+2td_i$. The inverse of a diagonal matrix is another diagonal matrix, whose nonzero elements are the inverted nonzero elements of the first matrix, i.e. $1/(1+2td_i)$). Then note that $|A^{-1}| = |A|^{-1}$ for any invertible matrix A. Finally, note that the determinant of a diagonal matrix is the product of the diagonal elements. This concludes the identity. □

Lemma A.1.1 (Expectation over a quadratic form including the precision matrix)**.** *Let $X = (X_1, ..., X_d)'$ be a d-dimensional random vector with zero mean. Let $\Sigma \in \mathbf{R}^{d\times d}$ be its covariance matrix, i.e. $[\Sigma]_{ij} \coloneqq Cov(X_i, X_j) \quad \forall i, j = 1, ..., d$. Then*

$$\mathbf{E}_X\Big[X'\Sigma^{-1}X\Big] = d \quad .$$

Proof.

$$\begin{aligned}\mathbf{E}_X\left[X'\Sigma^{-1}X\right] &= \left\{X'\Sigma^{-1}X \text{ is a scalar}\right\} = \mathbf{E}_X\left[tr(X'\Sigma^{-1}X)\right] = \\ &= \mathbf{E}_X\left[tr(\Sigma^{-1}XX')\right] = tr\left(\mathbf{E}_X\left[\Sigma^{-1}XX'\right]\right) = \\ &= tr\left(\Sigma^{-1}\mathbf{E}_X\left[XX'\right]\right) = \{ \text{ zero mean } \} = \\ &= tr\left(\Sigma^{-1}\Sigma\right) = tr(I) = \sum_{i=1}^{d} 1 = d \quad .\end{aligned}$$

□

Lemma A.1.2 (Derivation of ADMM algorithm for MLE)**.** *The iterative scheme of eq.* (3.5)*, eq.* (3.6) *and eq.* (3.7) *corresponds to*

$$\begin{aligned}x^{k+1} &:= V^k \ diag(\tilde{x}_{11}^k, ..., \tilde{x}_{NN}^k)(V^k)' \quad , \\ z_{ij}^{k+1} &:= \begin{cases} x_{ij}^{k+1} + u_{ij}^k & i = j \\ x_{ij}^{k+1} + u_{ij}^k - \frac{\gamma}{\rho} sign(x_{ij}^{k+1} + u_{ij}^k) & i \neq j \end{cases} \quad , \\ u^{k+1} &:= u^k + x^{k+1} + z^{k+1} \quad ,\end{aligned}$$

where V^k *such that* $\rho(z^k - u^k) - S = V^k\Lambda^k(V^k)'$ *with* $\Lambda = diag(\lambda_1, ..., \lambda_n)$ *and* $\tilde{x}_{ii} = \frac{\lambda_i + \sqrt{\lambda_i^2 + 4\rho}}{2\rho}$.

Proof. Starting off with the minimisation problem in eq. (3.5), the minimising x is found by differentiating the objective function with respect to x, then setting all partial derivatives equal to zero. Note that here $x \in \mathbf{R}^{N\times N}$. Assume that $x_{ij} \neq 0$[1] in order for $|x_{ij}|$ to have a well-defined derivative.

$$\begin{aligned}&\frac{\partial}{\partial x_{ij}}\Big[-\log|x|\Big] = [x^{-1}]_{ij} \quad \Longrightarrow \quad \frac{\partial}{\partial x}\Big[-\log|x|\Big] = x^{-1} \quad , \\ &\frac{\partial}{\partial x_{ij}}\Big[tr(Sx)\Big] = S_{ji} \quad \Longrightarrow \quad \frac{\partial}{\partial x}\Big[tr(Sx)\Big] = S' = S \quad \text{and} \\ &\frac{\partial}{\partial x_{ij}}\Big[||x - z^k + u^k||_F^2\Big] = 2(x_{ij} - z_{ij}^k + u_{ij}^k) \quad \Longrightarrow \quad \frac{\partial}{\partial x}\Big[||x - z^k + u^k||_F^2\Big] = 2(x - z^k + u^k) \quad .\end{aligned}$$

Plugging the above expressions into eq. (3.5) reveals that one must try to find x such that

$$\begin{aligned}& S - x^{-1} + \rho(x - z^k + u^k) = 0 \\ \Longleftrightarrow \quad & \rho x - x^{-1} = \rho(z^k - u^k) - S \quad .\end{aligned}$$

The above equation is solved by orthogonal decomposition of the right-hand side; let $\rho(z^k - u^k) - S = V^k\Lambda^k(V^k)'$ for V^k such that $V^k(V^k)' = (V^k)'V^k = I$ and Λ^k

[1] This condition is satisfied whenever running the resulting algorithm.

being a diagonal matrix with the eigenvalues $\lambda_1^k, ..., \lambda_N^k$ of $\rho(z^k - u^k) - S$ on the main diagonal. Then

$$\begin{aligned} & \rho x - x^{-1} = V^k \Lambda^k (V^k)' \\ \iff \quad & \rho (V^k)' x V^k - (V^k)' x^{-1} V^k = \Lambda^k \\ \iff \quad & \rho \tilde{x} - \tilde{x}^{-1} = \Lambda^k \quad \text{for } \tilde{x} = (V^k)' x V^k \\ \iff \quad & \rho \tilde{x}_{ii} - [\tilde{x}^{-1}]_{ii} = \lambda_i^k \quad \forall i = 1, ..., N \end{aligned}$$

a second-order polynomial equation that is solved by

$$\tilde{x}_{ii} = \frac{\lambda_i^k + \sqrt{(\lambda_i^k)^2 + 4\rho}}{2\rho} \quad .$$

Note that $\tilde{x}$ is a diagonal matrix. Finally, the solution x is obtained as

$$x = V^k \begin{bmatrix} \tilde{x}_{11} & ... & \emptyset \\ \vdots & \ddots & \vdots \\ \emptyset & ... & \tilde{x}_{NN} \end{bmatrix} (V^k)' \quad ,$$

with the $\tilde{x}_{ii}$s given as above.

Moving on to the minimisation problem in eq. (3.6) it is clear that this problem is slightly more complicated that the previous one, the target function now being $\gamma ||z||_{1^*} + \frac{\rho}{2} ||x^{k+1} - z + u^k||_F^2$. First of all, remember from chapter 2 that $|| \cdot ||_{1^*}$ denotes the l_1-norm applied to all off-diagonal elements, i.e.

$$||z||_{1^*} = \sum_i^N \sum_{\substack{j=1 \\ j \neq i}}^N |z_{ij}| \quad .$$

Thus, the computation of the minimising z need to be performed separately for off-diagonal and diagonal elements. Secondly, note that the l_1-norm is not a differentiable function at zero for what reason the gradient of $||z||_{1^*}$ cannot be computed straight away. Starting off with the main diagonal elements,

$$\begin{aligned} & \frac{\partial}{\partial z_{ii}} \left[\gamma ||z||_{1^*} + \frac{\rho}{2} ||x^{k+1} - z + u^k||_F \right] = 2(x_{ii}^{k+1} - z_{ii} + u_{ii}^k) = 0 \\ \iff \quad & z_{ii} = x_{ii}^{k+1} + u_{ii}^k \quad . \end{aligned}$$

As for the off-diagonal elements, use the fact that the sub-gradient of $|\cdot|$ is the sign-function:

$$\partial_{zij} \left[\gamma ||z||_1 + \frac{\rho}{2} ||x^{k+1} - z + u^k||_F \right] = \text{sign}(z_{ij}) + \rho(z_{ij} - x_{ij}^{k+1} - u_{ij}^k) = 0 \quad ,$$

then look at the two cases $z_{ij} > 0$ and $z_{ij} < 0$ separately:

$$\begin{aligned}
z_{ij} > 0: \quad & \gamma + \rho(z_{ij} - x_{ij}^{k+1} - u_{ij}^k) = 0 \\
\iff \quad & \frac{\gamma + \rho z_{ij}}{\rho} = x_{ij}^{k+1} - u_{ij}^k \\
\implies \quad & \operatorname{sign}(z_{ij}) = \operatorname{sign}(x_{ij}^{k+1} - u_{ij}^k) \quad \text{and} \\
z_{ij} < 0: \quad & -\gamma + \rho(z_{ij} - x_{ij}^{k+1} - u_{ij}^k) = 0 \\
\iff \quad & \frac{-\gamma + \rho z_{ij}}{\rho} = x_{ij}^{k+1} - u_{ij}^k \\
\implies \quad & \operatorname{sign}(z_{ij}) = \operatorname{sign}(x_{ij}^{k+1} - u_{ij}^k) \quad ,
\end{aligned}$$

i.e. $\operatorname{sign}(z_{ij}) = \operatorname{sign}(x_{ij}^{k+1} - u_{ij}^k)$ whenever $z_{ij} \neq 0$. Note that this conclusion has been drawn under the assumption that $\gamma, \rho > 0$. Thus the defining relation for the minimising z is

$$\begin{aligned}
& \gamma \cdot \operatorname{sign}(x_{ij}^{k+1} + u_{ij}^k) + \rho(z_{ij} - x_{ij}^{k+1} - u_{ij}^k) = 0 \quad \text{for } i \neq j \\
\iff \quad & z_{ij} = x_{ij}^{k+1} + u_{ij}^k - \frac{\gamma}{\rho} \operatorname{sign}(x_{ij}^{k+1} + u_{ij}^k) \quad .
\end{aligned}$$

Summing up, the iterative scheme is

$$\begin{aligned}
x^{k+1} &:= V^k \operatorname{diag}(x_1 1^{k+1}, ..., x_{NN}^{k+1}) V^k \quad , \\
z_{ij}^{k+1} &:= \begin{cases} x_{ij}^{k+1} + u_{ij}^k & i = j \\ x_{ij}^{k+1} + u_{ij}^k - \frac{\gamma}{\rho} \operatorname{sign}(x_{ij}^{k+1} + u_{ij}^k) & i \neq j \end{cases} \quad , \\
u^{k+1} &:= u^k + x^{k+1} + z^{k+1} \quad .
\end{aligned}$$

□

B

Full results

In this chapter are the full numerical results are listed for reference.

B.1 Performance of the estimators

The various estimators were trained and tested as described in chapter 3. Their performance are listed in table B.1 - B.7 below. As usual, κ indicates the condition number of the estimator, $\mathscr{L}$ is the value of the total loss function, q is the quantity used for dimensional checking, $\bar{Z}_{\text{oos}}$ is the out-of-sample return and σ_{oos} is the out-of-sample risk. The best estimatro of each type is marked out using bold text.

Table B.1: Performance of the risk-targeting estimator, using the sigmoid regulariser.

κ	6.7	6.7	6.7	6.6	6.6	6.3	**5.8**	5.5	4.4	4.8	4.3	6.	7.0	3.2	9.4	12	8.9
$\mathscr{L}$	491	491	491	491	491	491	**489**	491	495	503	517	538	572	636	682	753	929
q	3.19	3.19	3.19	3.19	3.19	3.21	**3.22**	3.25	3.20	1.79	2.06	2.74	3.40	3.52	5.41	7.36	6.40
$\bar{Z}_{\text{oos}}$	4.75	4.75	4.75	4.76	4.76	4.85	**4.94**	5.06	5.18	5.35	5.78	6.22	6.66	6.87	7.62	8.27	8.32
σ_{oos}	1.90	1.90	1.90	1.90	1.90	1.93	**1.95**	2.01	2.09	2.18	2.35	2.52	2.68	2.80	3.02	3.26	3.29

Table B.2: Performance of the risk-targeting estimator, using the log regulariser.

κ	2878	2638	1461	498	271	47	**14**	16	18	22	15	14	10
$\mathscr{L}$	4641	640	640	638	634	618	**608**	608	656	846	2732	43141	250299
q	1.87	1.87	1.89	1.97	2.06	3.42	**3.66**	4.08	4.09	4.57	6.28	15.67	33.19
$\bar{Z}_{\text{oos}}$	1.62	1.63	1.73	2.14	2.59	5.08	**6.94**	7.28	7.58	8.18	9.77	15.45	21.90
σ_{oos}	1.17	1.17	1.16	1.19	1.26	2.07	**2.81**	2.95	3.08	3.32	3.97	6.28	8.88

Table B.3: Performance of the risk-targeting estimator, using the linear-sigmoid regulariser.

κ	15	15	**14**	15	13	10	7.9	3.5	1.2	12.7	1734	1.9	1.6
$\mathscr{L}$	545	545	**534**	541	548	539	539	574	606	1014	2658	41755	218352
q	7.29	7.29	**6.08**	6.65	7.60	8.23	6.72	5.19	3.16	3.07	1.03	16.38	29.76
$\bar{Z}_{\text{oos}}$	7.81	7.81	**7.21**	7.49	8.16	8.82	8.31	8.28	7.34	7.94	7.89	15.98	22.09
σ_{oos}	2.91	2.91	**2.68**	2.79	3.03	3.27	3.10	3.20	2.98	3.34	3.85	6.40	8.94

Table B.4: Performance of the MLE-l_1 estimator. γ is the penalty parameter used.

γ	10^{-8}	10^{-7}	10^{-6}	$5 \cdot 10^{-6}$	10^{-5}	0.001	0.1	0.125	0.25	0.5	0.6125	**0.75**	0.8625	1
κ	340	340	307	224	162	90	42	36	24	7.8	4.4	**2.4**	1.4	0.99
$\mathscr{L}$	1506	1506	1534	1596	1643	1689	1868	1848	1806	1399	1087	**856**	988	1217
q	0.54	0.54	0.54	0.56	0.58	0.62	0.67	0.69	0.72	0.90	1.04	**1.24**	1.40	1.47
$\bar{Z}_{\text{oos}}$	1.37	1.37	1.37	1.41	1.49	1.73	2.12	2.18	2.39	3.55	4.37	**5.27**	5.85	6.06
σ_{oos}	0.92	0.92	0.91	0.91	0.92	0.95	0.98	1.01	1.16	1.74	2.06	**2.40**	2.63	2.75

Table B.5: Performance of the MLE-l_2 estimator. γ is the penalty parameter used.

γ	10^{-7}	10^{-5}	0.001	0.1	0.125	0.25	0.5	0.75	1	5	7.5	**10**	100	1000
κ	340	307	162	78	76	57	50	46	37	18	12	**8.6**	1.1	1
$\mathscr{L}$	1506	1534	1644	1688	1687	1654	1609	1563	1503	1102	981	**898**	910	1180
q	0.54	0.54	0.58	0.64	0.65	0.68	0.71	0.74	0.77	0.95	1.01	**1.07**	1.42	1.47
$\bar{Z}_{\text{oos}}$	1.37	1.37	1.49	1.80	1.82	2.00	2.14	2.25	2.43	3.69	4.26	**4.65**	5.93	6.05
σ_{oos}	0.92	0.91	0.92	0.97	0.97	1.01	1.05	1.08	1.13	1.50	1.67	**1.80**	2.60	2.73

Table B.6: Performance of the sample estimator with simple l_2-penalty. c is the penalty parameter used.

c	0.005	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	**0.8**	0.9	1
κ	1.1	1.2	2.2	3.6	6.9	11	17	25	36	56	**95**	211	20541
$\mathscr{L}$	948	811	765	929	1109	1164	1141	1057	914	719	**530**	987	481640
q	1.43	1.38	1.20	1.10	1.02	0.99	1.00	1.04	1.10	1.22	**1.41**	1.84	5.81
$\bar{Z}_{\text{oos}}$	5.93	5.81	5.09	4.53	3.87	3.47	3.21	3.02	2.87	2.77	**2.70**	2.67	3.33
σ_{oos}	2.62	2.52	2.06	1.79	1.54	1.42	1.35	1.32	1.31	1.33	**1.39**	1.55	3.20

Table B.7: Performance of the inverted sample covariance without regularisation.

κ	20541
$\mathscr{L}$	481640
q	5.81
$\bar{Z}_{\text{oos}}$	3.33
σ_{oos}	3.20

B.2 Optimal parameters of the spectral mapping

The optimal parameters p^* of the regularising functions g_p were computed using the quasi-Newton BFGS method applied to the total loss funciton $\mathscr{L}$. For some functions g_p, the algorithm did not converge. The optimal parameters are listed in tables B.8 - B.10 for the cases when convergence was obtained.

Table B.8: Optimal parameters of the sigmoid regularising function, $g_p^{\text{sigmoid}}(d) = a + \frac{e^b}{1+e^{-k(d-x_0)}}$. Note that a was held fixed and not optimised over. κ is the condition number of the resulting estimator.

κ	6.7	6.7	6.7	6.6	6.6	6.3	5.8	5.5	4.4	4.8	4.3	6.2	7.0	3.2	9.4	12.2	8.9
a	10^{-6}	10^{-5}	10^{-4}	$5 \cdot 10^{-4}$	0.001	0.01	0.05	0.1	0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2	1.3
e^b	2.63	2.63	2.63	2.63	2.63	2.61	2.39	2.50	1.83	1.99	1.98	2.42	2.77	2.24	4.70	9.16	5.32
k	1.91	1.91	1.91	1.90	6.67	6.33	9.66	6.19	86.7	3.35	3.35	1.93	1.33	4.70	0.45	0.14	0.27
x_0	0.95	0.95	0.95	0.96	2.60	2.61	2.20	2.73	2.59	1.42	1.77	3.26	5.10	124.13	17.38	51.54	22.77

Table B.9: Optimal parameters of the log regularising function, $g_p^{\log}(d) = a - bc + b\log(d + e^c)$. Note that a was held fixed and not optimised over. κ is the condition number of the resulting estimator.

κ	2878	2638	1461	498	271	47	14	16	18	22	15	13	10
a	10^{-4}	0.001	0.01	0.05	0.1	0.5	1.0	1.1	1.2	1.4	2.0	5.0	10.0
b	9.90	9.90	9.91	9.92	9.86	6.16	3.86	4.85	9.12	21.46	16.65	34.20	28.13
c	3.65	3.65	3.65	3.68	3.70	2.88	3.29	3.41	4.54	5.59	5.20	5.00	3.46

Table B.10: Optimal parameters of the linear-sigmoid regularising function, $g_p^{\text{ls}}(d) = a + e^c d + \frac{e^b}{1+e^{-k(d-x_0)}}$. Note that a was held fixed and not optimised over. κ is the condition number of the resulting estimator.

κ	15.4	15.4	14.5	15.0	13.1	10.4	7.9	3.5	1.2	12.7	1733.9	1.9	1.6
a	10^{-5}	10^{-4}	0.001	0.01	0.1	0.2	0.5	1.0	1.1	1.5	2.0	5.0	10.0
e^b	2.85	2.85	2.34	2.58	2.83	2.76	2.14	1.41	1.03	0.87	0.51	1.78	2.07
e^c	0.05	0.05	0.04	0.05	0.04	0.04	0.04	0.04	0.04	0.04	0.06	0.12	0.26
k	1.17	1.17	1.61	2.26	1.72	1.30	2.23	3.31	3.19	1.39	1.80	1.73	3.06
x_0	1.31	1.31	0.65	0.89	1.58	1.29	1.45	2.24	2.45	6.98	5.71	7.17	6.56