Markowitz Minimum Variance Portfolio Optimization using New Machine Learning Methods

by

Oluwatoyin Abimbola Awoye

Thesis

submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science

Department of Computer Science
University College London
Gower Street
United Kingdom
Email: t.alabi@cs.ucl.ac.uk

Declaration

I, Oluwatoyin Awoye, confirm that the work presented here is my own. When results have been derived from other sources this is indicated in the text.

Acknowledgements

First and foremost, I would like to thank God for continual guidance during this research. I would like to thank my main supervisor, Prof. John Shawe-Taylor for allowing me to undergo this research under his guidance. I would like to thank him for all his invaluable advice, pushing me and especially pointing me in the right direction to have contact with other researchers who have been beneficial to me.

I would like to thank Dr. Mark Herbster, who acted as my second supervisor. Mark has done more than is typical for that role, being extremely generous with his time especially during the early years of my research and I have benefited immensely from this. I would also like to thank Dr. Zakria Hussain, who directed me towards the required background on Markowitz portfolio optimization and its application in industry. I am grateful to my colleagues at the UCL Computer Science department for providing a supporting research atmosphere.

I would also like to thank my family for their encouragement throughout my research experience. To my husband, Yinka, thank you for being a great example to me, for your constant love, support and motivation throughout the years. Lastly, I would like to thank my parents for supporting and providing me with the necessary funding to undertake this research work. I am forever grateful and to them I dedicate this thesis.

Abstract

The use of improved covariance matrix estimators as an alternative to the sample covariance is considered an important approach for enhancing portfolio optimization. In this thesis, we propose the use of sparse inverse covariance estimation for Markowitz minimum variance portfolio optimization, using existing methodology known as *Graphical Lasso* [16], which is an algorithm used to estimate the inverse covariance matrix from observations from a multivariate Gaussian distribution.

We begin by benchmarking Graphical Lasso, showing the importance of regularization to control sparsity. Experimental results show that Graphical Lasso has a tendency to overestimate the diagonal elements of the estimated inverse covariance matrix as the regularization increases. To remedy this, we introduce a new method of setting the optimal regularization which shows performance that is at least as good as the original method by [16].

Next, we show the application of Graphical Lasso in a bioinformatics gene microarray tissue classification problem where we have a large number of genes relative to the number of samples. We perform dimensionality reduction by estimating graphical Gaussian models using Graphical Lasso, and using gene group average expression levels as opposed to individual expression levels to classify samples. We compare classification performance with the sample covariance, and show that the sample covariance performs better.

Finally, we use Graphical Lasso in combination with validation techniques that optimize portfolio criteria (risk, return etc.) and Gaussian likelihood to generate

new portfolio strategies to be used for portfolio optimization with and without short selling constraints. We compare performance on synthetic and real stock market data with existing covariance estimators in literature, and show that the newly developed portfolio strategies perform well, although performance of all methods depend on the ratio between the estimation period and number of stocks, and on the presence or absence of short selling constraints.

Contents

Contents

PART II: Graphical Lasso Application to Finance

List of Figures

List of Figures

List of Figures

List of Tables

List of Tables

List of Tables

List of Tables

List of Abbreviations

cDNA	complementary deoxyribonucleic acid
DNA	deoxyribonucleic acid
ERM	empirical risk minimization
GGM	graphical Gaussian model
GLasso	Graphical Lasso
GMVP	global minimum variance portfolio
i.i.d	independent and identically distributed
k-NN	*k*-Nearest Neighbours
Lasso	least absolute shrinkage and selection operator
LLS	linear least squares
LOESS	locally weighted scatter plot smoothing
loo	leave-one-out
MISE	mean integrated squared error
ML	maximum likelihood
MLE	maximum likelihood estimate
mRNA	messenger ribonucleic acid
MSE	mean squared error
MV	mean-variance
NSS	no short selling
NYSE	New York stock exchange
OLS	ordinary least squares
RMT	random matrix theory

List of Abbreviations

RNA	ribonucleic acid
SI	single index
S&P	Standard & Poor
SS	short selling
UIN	unique identification number

List of Symbols

$\mathbf{A}$	adjacency matrix
$\|\| \cdot \|\|_F$	Frobenius norm
$\| \cdot \|$	matrix determinant
C_0	capital to be invested
C_{end}	capital at the end of the investment period
$\boldsymbol{D}$	diagonal matrix of variances of ε_i
$\in$	belongs to
ϵ	noise (random error)
$\varepsilon_i(t)$	idiosyncratic stochastic term at time t
$f(t)$	index return at time t
f_x	probability density function of the random variable x
$f_{\boldsymbol{xy}}$	joint probability density function of the random variables x and
N	number of samples
i	number of assets
$\boldsymbol{I}$	identity matrix
$\boldsymbol{Q}$	shrinkage covariance estimate
R_{emp}	empirical risk
r_f	risk-free rate
$r_i(t)$	stock return for asset i at time t
R_i	return on asset i
R_P	total portfolio return
p	number of variables

List of Symbols

s	portfolio risk
$\boldsymbol{S}$	empirical (sample) covariance matrix
T	training set
$\boldsymbol{T}$	shrinkage target matrix
V	validation set
$\boldsymbol{w}$	weight vector
w_i	amount invested in asset i
$\mathbf{X}$	matrix of samples
$\mathbf{X}^C$	matrix of samples that have been centered
$\boldsymbol{x}$	vector of a single sample
$\boldsymbol{y}$	vector of output labels for all samples
y	single output label for a particular sample
$\boldsymbol{\mu}$	mean vector
$\boldsymbol{\Sigma}$	empirical (sample) covariance matrix
$\boldsymbol{\Sigma}^{-1}, \boldsymbol{\Omega}, \boldsymbol{X}, \boldsymbol{\Theta}$	inverse covariance (precision) matrix
$\boldsymbol{\beta}$	vector of estimated regression coefficients
λ, γ	ridge regression penalty parameter
$\perp\!\!\!\perp$	independence
$\forall$	for all
ρ	Graphical lasso regularization (penalty) parameter
σ	uniform noise magnitude
$\mu_P, \bar{r}$	portfolio expected return

List of Symbols

σ	portfolio variance
σ_P^2	variance of portfolio return
λ	portfolio risk-aversion parameter
Φ	range of Graphical Lasso regularization amounts
α	shrinkage intensity
σ_P	portfolio standard deviation
σ_{ij}	covariance between the returns of asset i and j
σ_{ii}	variance on the return of asset i

Chapter 1

Introduction

1.1 Background

The Markowitz mean-variance model (MV) has been used as a framework for optimal portfolio selection problems. In the Markowitz mean-variance portfolio theory, one models the rate of returns on assets in a portfolio as random variables. The goal is then to choose the portfolio weighting factors optimally. In the context of the Markowitz theory, an optimal set of weights is one in which the portfolio achieves an acceptable baseline expected rate of return with minimal volatility, which is measured by the variance of the rate of return on the portfolio. Under the classical mean-variance framework of Markowitz, optimal portfolio weights are a function of two parameters: the vector of the expected returns on the risky assets and the covariance matrix of asset returns. Past research has shown that the estimation of the mean return vector is a notoriously difficult task, and the estimation of the covariance is relatively easier. Small differences in the estimates of the mean return for example can result in large variations in the portfolio compositions. Also, the inversion of the actual sample covariance matrix of asset returns is usually a poor estimate especially in high dimensional cases and is rarely used since the portfolios it produces often are practically worthless, with extreme long and short positions [1]. The presence of the estimation risk in the MV model parameters makes its application

limited. Therefore it is of great importance to investigate ways of estimating the MV input parameters accurately.

1.2 Motivation

Markowitz portfolio selection requires estimates of (i) the vector of expected returns and (ii) the covariance matrix of returns [1, 2]. Finance literature in the past mostly focused on the estimation of the expected returns rather than the estimation of covariance. It was generally believed that in a mean-variance optimization process, compared to expected returns, the covariance is more stable and causes fewer problems, and therefore it is less important to have good estimations for it. Past research has shown that it is actually more difficult to estimate the expected returns vector than covariances of asset returns [3]. Also, the errors in estimates of expected returns have a larger impact on portfolio weights than errors in estimates of covariances [3]. For these reasons, recent academic research has focused on minimum-variance portfolios, which rely solely on estimates of covariances, and thus, are less vulnerable to estimation error than mean-variance portfolios [4]. Many successful proposals to address the first estimation problem (the estimation of the vector of expected returns) exist now. Over 300 papers have been listed [5] on the first estimation problem, so it is fair to say that a lot of research has been done on improving the estimation of the returns. In comparison, much less research has been done on the second estimation, the estimation of the covariance matrix of asset returns [1]. With new developments in technology and optimization, there has been growing interest in improving the estimation of the covariance matrix for portfolio

selection. Recent proposals by [6-14] among others, show that this topic is currently gathering significant amount of attention [1].

Despite all the research efforts, the problem of parameter uncertainty remains largely unsolved. It has been shown that the simple heuristic $1/N$ portfolio also known as the Naïve portfolio, outperforms the mean-variance portfolio and most of its extensions [15]. The finding has led to a new wave of research that seeks to develop portfolio strategies superior to the Naïve portfolio and to reaffirm the practical value of portfolio theory [15]. Despite the recent interests in improving the estimation of the covariance matrix of asset returns, there still remains many areas for further investigation. There has been very little work in literature on deriving estimators of the inverse covariance matrix directly, in the context of portfolio selection. Most research has focused on improving estimation of the covariance matrix and then inverting these estimates to compute portfolio weights, and it is well known that inversion of the covariance matrix produces very unstable estimates. In this respect, it is of great importance to develop efficient estimates of the inverse covariance matrix. The research in this thesis studies several issues related to obtaining a better estimate of the covariance matrix of asset returns in the context of the Markowitz portfolio optimization.

1.3 Contribution

The research in this thesis focuses on the application of machine learning methods for Markowitz mean-variance portfolio optimization. Using existing methodology known as *Graphical Lasso*, which is an algorithm used to estimate the precision matrix

(inverse covariance matrix) from observations from a multivariate Gaussian distribution, we first evaluate the performance of this sparse inverse covariance estimator on synthetic data. We perform experiments to show the importance of regularization and introduce a new method of setting the regularization parameter that results in performance that is at least as good as the original method by [16]. Next, we show the application of Graphical Lasso in a bioinformatics gene microarray tissue classification problem where we have a large number of genes relative to the number of samples. We present a method for dimensionality reduction by estimating graphical Gaussian models using Graphical Lasso, and using gene group average expression levels as opposed to individual expression levels to classify samples. We compare classification performance with the sample covariance, and show that Graphical Lasso does not appear to perform gene selection in a biologically meaningful way, though the sample covariance does.

Lastly, we create new Graphical Lasso portfolio strategies by applying validation methodology to optimize certain portfolio criteria (realized risk, portfolio return and portfolio Sharpe ratio) in the Markowitz mean-variance framework. By optimizing three different portfolio criteria, we come up with three new portfolio strategies and evaluate their performance using both real and synthetic data. We additionally come up with two new portfolio strategies based on Gaussian likelihood, which we use as a performance measure in the synthetic stock data experiment. We account for the presence of short-sale restrictions, or the lack thereof, on the optimization process and study their impact on the stability of the optimal portfolios. Using real and synthetic stock market data, we show that Gaussian likelihood is an effective way to

select optimal regularization based on results that show that the Graphical Lasso portfolio strategy that uses Gaussian likelihood to select the optimal regularization consistently performs the best compared to all other proposed Graphical Lasso strategies. From the real data results, we show that the new sparse portfolio strategies offer portfolios that are less risky than the methods that use the maximum likelihood estimate (MLE), the Naïve equally weighted portfolio method and other popular covariance estimation methods, even in the presence of short selling constraints. Lastly, we demonstrate the importance of regularization accuracy and show how this affects portfolio performance.

1.4 The organization of this Thesis

This thesis is divided into two parts:

i. Part I: Benchmarking the Graphical Lasso
ii. Part II: Applications to Finance

Appendix A provides the mathematical background knowledge necessary for this thesis and should be reviewed before the chapters. In the *Benchmarking the Graphical Lasso,* we focus on the evaluation of Graphical Lasso on synthetic data. To that end, we perform experiments showing the ability of Graphical Lasso to recover structure in chapter 2. We emphasize the importance of regularization and introduce a new way of setting the regularization parameter, which leads to performance at least as good as the original way of setting the regularization proposed by [16]. In chapter 3, we apply Graphical Lasso to Bioinformatics gene microarray data by generating

graphical Gaussian models which are used to perform supervised classification tasks of identifying tissue samples.

Subsequently, we proceed to the *Applications to Finance* part, where we present in chapter 4, the basics of Markowitz minimum-variance portfolio optimization and the formal definition of the Markowitz mean-variance (MV) model. We propose new portfolio strategies by using validation techniques that optimize certain portfolio criteria, and evaluate these new strategies on synthetic data. In chapter 5, we discuss the estimation risk of the MV model as well as several current methods used to address the estimation risk of the covariance of asset returns. We perform an in-depth comparative analysis of our newly proposed portfolio strategies against other existing methodology for the estimation of the covariance matrix for Markowitz minimum variance portfolio optimization on stock market data.

We conclude the thesis with chapter 6 which summarises results from all experiments and suggests future directions for research.

Part I: Benchmarking the Graphical Lasso

Chapter 2

Evaluation of the Graphical Lasso Algorithm

This chapter introduces the Graphical Lasso algorithm, which is the methodology that is used in this thesis to recover the sparse structure of multivariate Gaussian data. We begin by evaluating Graphical Lasso's performance on synthetic data, where the true inverse covariance matrix used to generate the data is known, and show the algorithm's ability to recover structure. We show the importance of regularization for optimal structure recovery, and demonstrate a way to select the penalty parameter for regularization, which is important for creating portfolio strategies in chapter 4 and 5 for Markowitz minimum variance portfolio optimization. Lastly, we introduce a new method of penalty selection for regularization and compare this new method's performance to the original method for penalty selection proposed by [16].

2.1 Introduction to the Sparse Inverse Covariance

The multivariate Gaussian distribution is used in many real life problems to represent data. As a consequence, recovering the parameters of this distribution from data is a central problem. Analyses of interactions and interrelationships in Gaussian data involve finding models, structures etc., which explain the data and allow for interpretation of the data [17]. For a small number of variables, the interpretation of resulting hypothesized models is relatively straightforward due to the small number

of possible interactions [18]. For higher dimensional data, problems arise due to the fact that the number of interactions increases as the dimensionality of the data increases. As a result of this, the ease of interpretation becomes an increasingly important aspect in the search for suitable models. In such situations it is therefore natural to think in terms of conditional independences between variables and subsets of variables [19].

Undirected graphical models offer a way to describe and explain the relationships among a set of variables, a central element of multivariate analysis [17]. The 'principle of parsimony' dictates that we should select the simplest graphical model that adequately explains the data [16,19,20,21,22]. The celebrated 'principle of parsimony' has long ago been summoned for large covariance or inverse covariance matrices because it allows for easier interpretation of data. Knowing the inverse covariance (precision) matrix of a multivariate Gaussian – denoted by both $\boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{X}$ in this thesis – provides all the structural information needed to construct a graphical model of the distribution. A non-zero entry in the precision matrix corresponds to an edge between two variables. The basic model for continuous data assumes that the observations have a multivariate Gaussian distribution with mean μ and covariance matrix $\boldsymbol{\Sigma}$. If the *ij*th component of $\boldsymbol{\Sigma}^{-1}$ is zero, then variables with indices *i* and *j* are conditionally independent, given the other variables. Thus, it makes sense to impose and L_1 penalty for the estimation of $\boldsymbol{\Sigma}^{-1}$ to increase its sparsity [16].

2.2 Undirected Graphical Models

A graph consists of a set of vertices (nodes), along with a set of edges joining some pairs of vertices [18]. In graphical models, each vertex represents a random variable, and the graph gives a visual way of understanding the joint distribution of the entire set of random variables [18]. They can be useful for either unsupervised or supervised learning. In an *undirected graph*, the edges have no directional arrows. We restrict our discussion to undirected graphical models, also known as *Markov random fields*. In these graphs, the absence of an edge between two vertices has a special meaning: the corresponding random variables are conditionally independent, given the other variables [18]. Figure A.5 shows an example of a graphical model for a flow-cytometry dataset with $p = 11$ proteins, measured on $N = 7466$ cells, from [23]. Each vertex of the graph corresponds to the real-valued expression level of a protein. The network structure was estimated assuming a multivariate Gaussian distribution, using the Graphical Lasso procedure discussed in the subsequent sections.

Sparse graphs have a relatively small number of edges, and are convenient for interpretation. They are useful in a variety of domains, including genomics and proteomics, where they provide rough models of cell pathways. The edges in a graph are parametrized by values or potentials that encode the strength of the conditional independence between the random variables at the corresponding vertices.

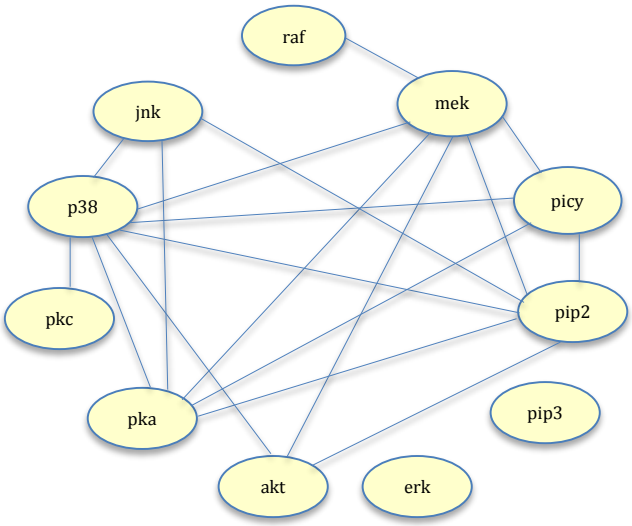

Figure 2.1 Example of a sparse undirected graph, estimated from a flow cytometry dataset, with p = 11 proteins measured on N = 7466 cells. The network structure was estimated using the Graphical Lasso.

The main challenges in working with graphical models are model selection (choosing the structure of the graph), estimation of the edge parameters from data, and computation of marginal vertex probabilities and expectations, from their joint distribution. The other major class of graphical models, the *directed graphical models*, known as *Bayesian networks*, in which the links have a particular directionality (indicated by arrows), will not be studied in this thesis. A brief overview of both directed and undirected graphs can be found in [24].

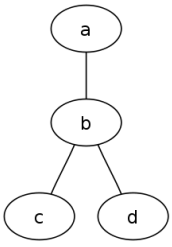

Figure 2.2 Example of an undirected graphical model. Each node represents a random variable, and the lack of an edge between nodes indicates conditional independence. For example, in the graph, c and d are conditionally independent, given b.

2.2.1 Graphical Gaussian Model

Here we consider Markov random fields where all the variables are continuous. The Gaussian distribution is almost always used for such graphical models, because of its convenient analytical properties. We assume that the observations have a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The Gaussian distribution has the property that all conditional distributions are also Gaussian. The inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ contains information about the partial covariances between the variables; that is, the covariances between pairs *i* and *j*, conditioned on all other variables. In particular the *ij*th component of $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ is zero, then variables *i* and *j* are conditionally independent, given the other variables. We now examine the conditional distribution of one variable versus the rest, where the role of $\boldsymbol{\Theta}$ is explicit. Suppose we partition $X = (Z, Y)$ where $Z = (X_1, \ldots, X_p)$ consists of the first $p - 1$ variables and $Y = X_p$ is the last. Then we have the conditional distribution of Y given Z [25]

$$Y|Z = z \sim N(\mu_Y + (z - \mu_Z)^T \boldsymbol{\Sigma}_{ZZ}^{-1} \sigma_{ZY}, \sigma_{YY} - \sigma_{ZY}^T \boldsymbol{\Sigma}_{ZZ}^{-1} \sigma_{ZY}), \tag{2.1}$$

where we have partitioned $\boldsymbol{\Sigma}$ as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{ZZ} & \sigma_{ZY} \\ \sigma_{ZY}^T & \sigma_{YY} \end{pmatrix} \tag{2.2}$$

The conditional mean in (2.1) has exactly same form as the population multiple linear regression of *Y* on *Z*, with regression coefficient $\beta = \boldsymbol{\Sigma}_{ZZ}^{-1} \sigma_{ZY}$. If we partition $\boldsymbol{\Theta}$ in the same way, since $\boldsymbol{\Sigma}\boldsymbol{\Theta} = \mathbf{I}$ standard formulas for partitioned inverses give

$$\theta_{ZY} = -\theta_{ZY} \cdot \mathbf{\Sigma}_{ZZ}^{-1}\sigma_{ZY}, \tag{2.3}$$

where $1/\theta_{YY} = \sigma_{YY} - \sigma_{ZY}^{T}\mathbf{\Sigma}_{ZZ}^{-1}\sigma_{ZY} > 0$. Hence

$$\begin{aligned}\beta &= \mathbf{\Sigma}_{ZZ}^{-1}\sigma_{ZY} \\ &= -\theta_{ZY}/\theta_{YY}\end{aligned}$$

Thus $\mathbf{\Theta}$ captures all the second-order information (both structural and quantitative) needed to describe the conditional distribution of each node given the rest, and is the so-called "natural" parameter for the graphical Gaussian model [1][18].

Another (different) kind of graphical model is the *covariance graph* or *relevant network*, in which vertices are connected by bidirectional edges if the covariance (rather than the partial covariance) between the corresponding variables is nonzero. The covariance graph however will not be studied in this thesis.

2.2.2 The Benefits of Sparsity

As pointed out in Appendix A section A.5.2.2, Lasso regularization is beneficial in that it provides an increase in prediction accuracy and allows for better interpretation. The *Graphical Lasso* is a method of sparse inverse covariance estimation, which offers the same benefits that lasso does for linear regression. This variable reduction capability is particularly important in portfolio optimization, where investors prefer

[1] The distribution arising from a Gaussian graphical model is a Wishwart distribution. This is a member of the exponential family, with canonical or "natural" parameter $\mathbf{\Theta} = \boldsymbol{\Sigma}^{-1}$.

to invest in a smaller basket of stocks, and also in bioinformatics, when visualizing gene microarray data.

2.2.3 Inexact Methods for the Inverse Covariance Estimation

2.2.3.1 Covariance Selection Discrete Optimization

Covariance selection was first introduced by [20] as a technique for reducing the number of parameters in the estimation of the covariance matrix of a multivariate Gaussian population [20]. The belief was that the covariance structure of a multivariate Gaussian population could be simplified by setting elements of the inverse covariance matrix to be zero. Covariance selection aims at discovering the conditional independence restrictions (the graph) from a set of independent and identically distributed observations [20]. One main current of thought that underlies Covariance selection is the 'principle of parsimony' in parametric model fitting. This principle suggests that parameters should only be introduced sparingly and only when the data indicate they are required [18]. Parameter reduction involves a tradeoff between benefits and costs. If a substantial number of parameters can be set to null values, the amount of noise in a fitted model due to errors of estimation is substantially reduced. On the other hand, errors of misspecification are introduced if the null values are incorrect. Every decision to fit a model involves an implicit balance between these two kinds of errors [18, 20].

Covariance selection relies on the discrete optimization of an objective function. Usually, greedy forward or backward search is used. In forward search, the initial estimate of the edge set is empty, and edges are added to the set until a time where

an edge addition does not appear to improve fit significantly. In backward search, the edge set consists of all off-diagonal elements, and then edge pairs and dropped from the set one at a time, as long as the decrease in fit is not significantly large. The selection (deletion) of a single edge in this search strategy requires an MLE fit for $O(p^2)$ different models [20].

The forward/backward search in covariance selection is not suitable for high-dimensional graphs in the multivariate Gaussian setting because if the number of variables p becomes moderate, the number of parameters $p(p+1)/2$ in the covariance structure becomes large. For a fixed sample size N, the number of parameters per data point increases $(p+1)/2$ as p increases. Using the technique proposed by [20], model selection and parameter estimation are done separately. The parameters in the precision matrix are typically estimated based on the model selected. Thus, parameter estimation and model selection in the Gaussian graphical model are equivalent to estimating parameters and identifying zeros in the precision matrix [26]. Applications of this sort of problem ranges from inferring gene networks, analyzing social interactions and portfolio optimization.

This sort of exhaustive search is computationally infeasible for all but very low-dimensional models and the existence of the MLE is not guaranteed in general if the number of observations is smaller than the number of nodes (variables).

2.2.3.2 Neighborhood Selection with the Lasso

To remedy the problems that arise from high dimensional graphs and computational complexity, [27] proposed a more computationally attractive method for covariance

selection for very large Gaussian graphs. They take the approach of estimating the conditional independence restrictions separately for each node in the graph. They show that the neighborhood selection can be cast into a standard regression problem and can be solved efficiently with the Lasso [16]. They fit a Lasso model to each variable, using the others as predictors.

The component $\left(\boldsymbol{\Sigma}^{-1}\right)_{i,j}$ is then estimated to be non-zero if either the estimated coefficient of variable *i* on *j*, or the estimated coefficient of variable *j* on *i*, is non-zero (alternatively, they use an AND rule). They show that this approach consistently estimates the set of non-zero elements of the precision matrix. Neighborhood selection with the Lasso relies on optimization of a convex function, applied consecutively to each node in the graph. This method is more computationally efficient than the exhaustive search technique proposed by [20]. They show that the accuracy of the technique presented by [20] in covariance selection is comparable to the Lasso neighborhood selection if the number of nodes is much smaller than the number of observations. The accuracy of the covariance selection technique breaks down, however, if the number of nodes is approximately equal to the number of observations, in which case this method is only marginally better than random guessing. Neighborhood selection with the Lasso does model selection and parameter estimation separately. The parameters in the precision matrix are typically estimated based on the model selected.

The discrete nature of such procedures often leads to instability of the estimator because small changes in the data may result in very different estimates [28]. The

neighborhood selection with the lasso focuses on model selection and does not consider the problem of estimating the covariance or precision matrix.

2.2.4 Exact Methods for the Inverse Covariance Estimation

2.2.4.1 L_1- penalised methods

A penalized likelihood method that does model selection and parameter estimation simultaneously in the Gaussian graphical model was proposed by [22]. The authors employ an L_1 penalty on the off-diagonal elements of the precision matrix. The L_1 penalty, which is very similar to the Lasso in regression [29], encourages sparsity and at the same time gives shrinkage estimates. In addition, it is ensures that the estimate of the precision matrix is always positive definite. The method presented by [22] is said to be more efficient due to the incorporation of the positive definite constraint and the use of likelihood, though a little slower computationally than the neighborhood selection method proposed by [21] due to this same constraint. They show that because the approach of [27] does not incorporate the symmetry and positive-definiteness constraint in the estimation of the precision matrix, therefore an additional step is needed to estimate either the covariance or precision matrix. [22] show that their objective function is non-trivial but similar to the determinant-maximization problem [17, 22], and can be solved very efficiently with interior-point algorithms. One problem with their approach was the memory requirements and complexity of existing interior point methods at the time, which were prohibitive for problems with more than tens of nodes.

A new approach of discovering the pattern of zeros in the inverse covariance matrix by formulating a convex relaxation to the problem was proposed by [30]. The authors derive two first-order algorithms for solving the problem in large-scale and dense settings. They don't make any assumptions of known sparsity *a priori*, but instead try to discover structure (zero pattern) as they search for a regularized estimate. They present provably convergent algorithms that are efficient for large-scale instances, yielding a sparse, invertible estimate of the precision matrix even for $N < p$ [30]. These algorithms are the smooth optimization method and block coordinate descent method for solving the Lasso penalized Gaussian MLE problem. The smooth optimization method is based on Nesterov's first order algorithm [31] and yields a complexity estimate with a much better dependence on problem size than interior-point methods. The second method recursively solves and updates the Lasso problem. They show that these algorithms solve problems with greater than a thousand nodes efficiently but in experimental work, they choose to only use the smooth optimization method which is based on Nesterov's first-order algorithm and call their algorithm COVSEL. This method has an improved computational complexity of $O(p^{4.5})$ than previous methods.

A new method of estimating the inverse covariance was proposed by [16] using the block coordinate descent approach proposed by [30]. The authors go on to implement the block coordinate method pointed out by [30], and solve the Lasso-penalized Gaussian MLE problem using an algorithm called *Graphical Lasso*. Very fast existing coordinate descent algorithms enable them to solve the problem faster than previous methods, $O(p^3)$ [16]. Graphical Lasso maximizes the Gaussian log-likelihood

of the empirical covariance matrix $\boldsymbol{S}$, through L_1 (Lasso) regularization [16]. Through regularization, the method encourages sparsity in the inverse covariance matrix, and as a consequence, demonstrates a robustness to noise, a weakness that plagues the maximum likelihood approach [16]. In the context of portfolio selection, assuming *a priori* sparse dependence model may be interpreted as for example, assets belonging to a given class being related together while assets belonging to different classes are more likely to be independent. In other words, a sparse precision corresponds to covariates that are conditionally independent, so given the knowledge of a given subset, the remainder are uncorrelated.

2.3 Introduction to the Graphical Lasso

Suppose we have N multivariate Gaussian observations of dimension p, with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\sum}$. Following [67], Let $\boldsymbol{X} = \sum^{-1}$ be the estimated precision matrix and let $\boldsymbol{S}$ be the empirical covariance matrix, the problem is to maximize the log-likelihood by using a coordinate descent procedure to maximize the log-likelihood of the data [21]

$$\widehat{\boldsymbol{X}} = \arg\max_{\boldsymbol{X}} log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS}) - \rho\|\boldsymbol{X}\|_1 \tag{2.4}$$

where $\boldsymbol{S}$ is a $p \times p$ empirical (sample) covariance matrix computed from observed data and $\boldsymbol{X}$ is a $p \times p$ symmetric and positive semi-definite matrix, which is the estimated precision matrix. $\|\boldsymbol{X}\|_1$ is the sum of the absolute values of the elements in

$\boldsymbol{X}$. The term $\rho\|\boldsymbol{X}\|_1$ is known as a regularization term and ρ is known as the penalty parameter. $|\boldsymbol{X}|$ is the determinant of $\boldsymbol{X}$ and $Trace(\boldsymbol{XS})$ is the sum of the diagonal elements of the matrix product of $\boldsymbol{XS}$. Appendix B presents a detailed description of the Graphical Lasso algorithm.

The existence of the penalty term ρ is what encourages sparsity in the estimated precision matrix. The practical challenge is to decide how many and which non diagonal entries in the precision should be set to zero. The Graphical Lasso presents questions such as how to select the penalty term ρ? When does Graphical Lasso perform well? These questions will be addressed in this thesis as they relate to the Markowitz global minimum variance portfolio optimization problem.

2.3.1 A Synthetic Data Experiment

We begin with a small synthetic example showing the ability of Graphical Lasso to recover the sparse structure from a noisy matrix. Using the same data generation example for generating synthetic data as [30], the sparse structure is recovered at different regularizations ρ = (0.1, 0.4, 1, 2). The purpose of this is to see how the optimization solution changes as the regularization parameter, ρ, is increased. Starting with a sparse matrix $\boldsymbol{A}$, we obtain $\boldsymbol{S}$ by adding a uniform noise of magnitude σ = 0.1 to $\boldsymbol{A}^{-1}$. In Figure 2.3, Figure 2.4, Figure 2.5 and Figure 2.6, the sparsity pattern of $\boldsymbol{A}$ and the optimization solution are shown. The blue colour represents positive numbers while the red represents negative numbers. The magnitude of the number is also illustrated by the intensity of the colour, with darker colours representing higher magnitudes and vice versa.

The purpose of this experiment is to illustrate the very important problem the Graphical Lasso method presents, which is how to effectively select the correct regularization. From Figure 2.3, Figure 2.4, Figure 2.5 and Figure 2.6, it is evident that the selection of the right amount of regularization is crucial to getting the right solution. This issue of how to select the right penalty will be addressed in this thesis as it relates to the Markowitz portfolio optimization problem in chapters 4 and 5.

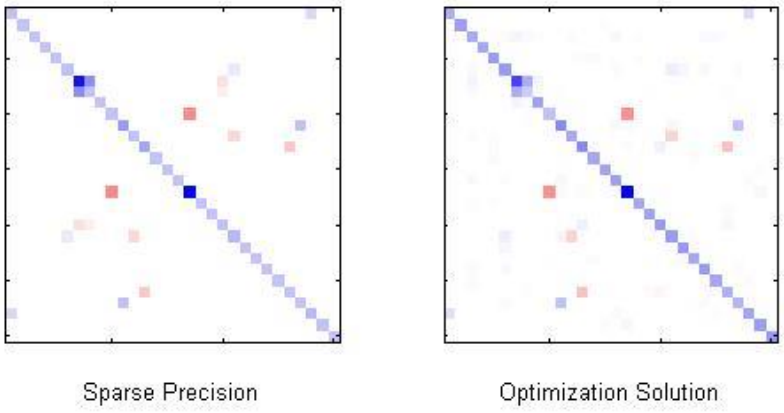

Figure 2.3 Original sparse precision versus Graphical Lasso optimization solution at regularization ($\rho = 0.1$).

Figure 2.3 shows the Graphical Lasso solution at a regularization of 0.1. This solution appears to be very close to the original sparse precision. The diagonal elements are estimated correctly and majority of the off-diagonal elements are also estimated correctly in the solution. In Figures 2.4, Figure 2.5 and Figure 2.6, we illustrate what happens as we increase the regularization amount.

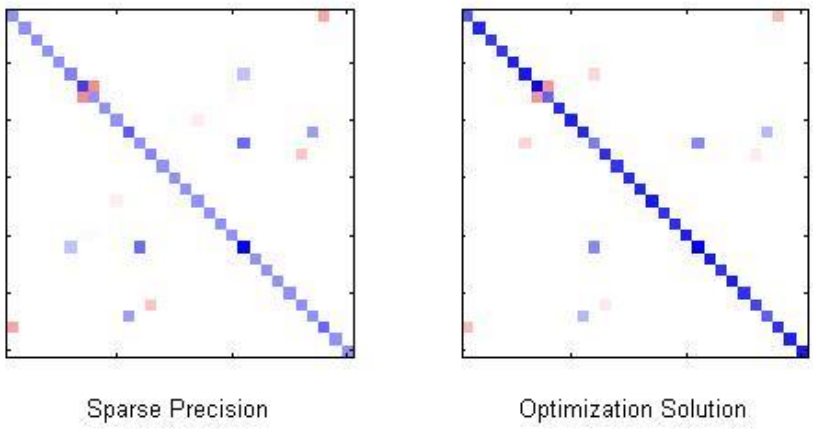

Figure 2.4 Original sparse precision versus Graphical Lasso optimization solution at regularization ($\rho = 0.4$).

Figure 2.4 shows the Graphical Lasso solution at a regularization of 0.4. We can see that the solution is worse. The diagonal elements are estimated larger, indicated by a darker diagonal, while a lot of off-diagonal elements disappear. By nature of the Graphical Lasso, as the regularization amount increases, the off-diagonals are shrunk closer to zero and this is evident in Figure 2.4. We now look at solutions for even higher regularization amounts in Figure 2.5 and Figure 2.6.

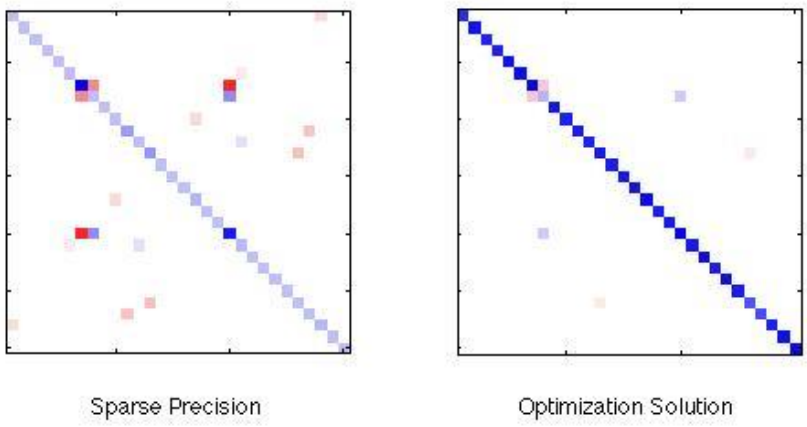

Figure 2.5 Original sparse precision versus Graphical Lasso optimization solution at regularization (ρ = 1).

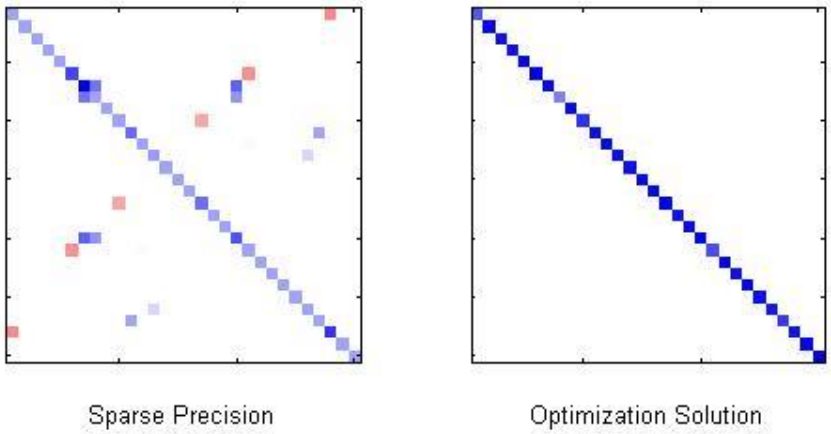

Figure 2.6 Original sparse precision versus Graphical Lasso optimization solution at regularization (ρ = 2).

Figure 2.5 shows the Graphical Lasso solution at a regularization of 1 while Figure 2.6 shows the solution at a regularization of 2. We can see the solution gets worse as the regularization amount increases. In Figure 2.6, the off diagonal elements completely disappear, leaving very strong diagonal elements.

This experiment shows the importance of selecting the correct regularization amount, while also illustrating the issue of over-regularization, where the off-diagonal elements are shrunk too much while the diagonal elements are estimated too largely as the regularization amount is increased. This issue of over-regularization is addressed in section 2.4, where we introduce a new method of selecting the regularization by separately choosing regularization amounts for diagonal and off-diagonal elements.

2.3.2 A Method for Penalty Selection

Recall that Graphical Lasso solves the optimization problem given by

$$\max_{\boldsymbol{X}} \; log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS}) - \rho\|\boldsymbol{X}\|_1 \tag{2.5}$$

where the optimal value of the penalty term ρ must somehow be approximated. There are several ways that have been presented in literature on how to set the penalty ρ. One method called the "regression" approach fits Graphical Lasso to a portion of the data, and uses the penalized regression model for testing in the validation set [16]. Another method called the "likelihood" approach involves training the data with Graphical Lasso and evaluating the log-likelihood over the

validation set [16]. In chapter 4, we present new validation methods for selecting the penalty ρ to obtain estimates of the inverse covariance matrix for mean-variance portfolio optimization.

This section demonstrates a method for approximating the optimal value of the penalty ρ. The method involves drawing a portion of data points from the training set to form a validation set and assessing the solution produced by Graphical Lasso for a number of different values of ρ on this validation set. We let T denote the training set and V denote the validation set.

Algorithm 2.1 Penalty Selection

1: Select V $\subset$ T
2: T' $\coloneqq$ T \ V
3: $\Phi \coloneqq \{\rho_1, \rho_2, \dots, \rho_n\}$
4: **for all** $\rho_i \in \Phi$ **do**
5: $\widehat{\boldsymbol{X}}_i$ = Graphical Lasso (T', ρ_i)
6: Υ_i = Fitness of Solution (V, $\widehat{\boldsymbol{X}}_i$)
7: **end for**
8: Select $\rho^* = \rho_i$ corresponding to maximal Υ_i
9: $\widehat{\boldsymbol{X}}^*$ = Graphical Lasso (T, ρ^*)

In this thesis, the calculation of the fitness of each $\widehat{\boldsymbol{X}}_i$ is achieved through the "Likelihood" approach that is offered by [16]. In this approach, the fitness of the solution of $\widehat{\boldsymbol{X}}_i$ is equal to the log-likelihood given in (2.5) in which the empirical covariance matrix $\mathbf{S}_V$ is calculated from the validation set V, providing unseen (to Graphical Lasso) information.

$$\text{Fitness of Solution (V, } \widehat{\boldsymbol{X}}_i \text{)} = log\left(\left|\widehat{\boldsymbol{X}}\right|\right) - Trace\left(\widehat{\boldsymbol{X}}_i \, \mathbf{S}_V\right) - \rho\left\|\widehat{\boldsymbol{X}}_i\right\|_1 \qquad (2.6)$$

This thesis and the original Graphical Lasso paper [16] use a type of validation called k-fold cross-validation in which the training set T is randomly partitioned into k validation subsets $V_{1,\ldots,k}$ each of size $\frac{|T|}{k}$ leaving k corresponding training sets $T'_{1,\ldots,k}$ each of size $\frac{(k-1)|T|}{k}$. Algorithm 3.1 is then applied k times to each V_i and T'_i, and the final selected ρ^* is the average over the results of the k runs. Through k-fold cross-validation it is hoped to minimize overfitting of the training set T. In this thesis, we use k = 5 folds (while the original paper [16] uses k = 10 folds) as a compromise between running time and overfitting.

2.4 A New Method for Penalty Selection

From the experiment in section 2.3.1, it is evident that as the regularization amount ρ increases, the off-diagonal element values decrease, while the diagonal elements increase. There is evidence of overestimation of the diagonal elements as the solution becomes more sparse. To remedy this problem, the idea of using two different penalties is considered; one penalty for the diagonal elements, and the other for the off-diagonal elements.

2.4.1 The Model

Graphical Lasso maximizes the L_1- log likelihood equation,

$$log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS}) - \rho\|\boldsymbol{X}\|_1 \tag{2.7}$$

To correct the problem of overestimation of the diagonal, expression (2.7) is modified to the following

$$log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS}) - \rho 1\|\boldsymbol{X}\|_{1\{i\neq j\}} - \rho 2\|\boldsymbol{X}\|_{1\{i=j\}} \quad (2.8)$$

From (2.8) we can see that two different penalties will be chosen; one for the off-diagonal elements ($\rho 1$), and the other for the diagonal elements ($\rho 2$). This new algorithm will be referred to as the *Modified Graphical Lasso*.

To approximate the penalties ρ, $\rho 1$ and $\rho 2$, a method involving cross validation is used. This method, described in section 2.3.2, Algorithm 2.1 involves drawing a portion of data points from the training set to form a validation set and assessing the solution produced by both Graphical Lasso and the Modified Graphical Lasso for a number of different values of ρ, $\rho 1$ and $\rho 2$ on the validation set.

The possible regularizations amounts chosen for ρ, $\rho 1$ and $\rho 2$ are the same range of 20 different values from 0 to 1000. For the Modified Graphical Lasso, all possible combinations of $\rho 1$ and $\rho 2$ are considered. The optimal choice of ρ, $\rho 1$ and $\rho 2$ are picked based on maximum likelihood/largest fitness. The fitness of the solution $\widehat{\boldsymbol{X}}$ is equal to the log-likelihood given in (2.7) and (2.8), in which the empirical covariance matrix is calculated using the validation set.

Fitness of the Graphical Lasso solution is given by

$$log\left(\left|\widehat{\boldsymbol{X}}\right|\right) - Trace\left(\widehat{\boldsymbol{X}}\mathbf{S}_V\right) - \rho\left\|\widehat{\boldsymbol{X}}\right\|_1 \quad (2.9)$$

Fitness of the Modified Graphical Lasso solution is given by

$$log(|\widehat{\boldsymbol{X}}|) - Trace(\widehat{\boldsymbol{X}}\mathbf{S}_V) - \rho 1\|\widehat{\boldsymbol{X}}\|_{1\{i\neq j\}} - \rho 2\|\widehat{\boldsymbol{X}}\|_{1\{i=j\}} \quad (2.10)$$

Synthetic data experiments are performed to show how the Modified Graphical Lasso performs compared to the Graphical Lasso, and the results are shown in the subsequent sections. 5-fold cross validation is used to train and test the data and calculate the likelihood and the optimal regularizations. We use the Moore-Penrose pseudoinverse (introduced in Appendix A) as a baseline method for comparison.

2.4.2 Generating Synthetic Data

To implement the new penalty selection method, synthetic data is generated according to the following models:

- Model 1: A sparse model taken from [22]

  $(\boldsymbol{X})_{i,i} = 1$, $(\boldsymbol{X})_{i,i-1} = (\boldsymbol{X})_{i-1,i} = 0.5$, and 0 otherwise.

  The diagonal elements of $\boldsymbol{X}$ are equal to 1 and the elements next to each diagonal entry equal to 0.5. All other elements are equal to 0.

- Model 2: A dense model taken from [22]

  $(\boldsymbol{X})_{i,i} = 2$, $(\boldsymbol{X})_{i,i'} = 1$ otherwise.

  The diagonal entries of $\boldsymbol{X}$ are equal to 2 and all other elements are equal to 1.

For each model, we simulated i.i.d Gaussian samples of sizes ($N = 5, 10, 20, \ldots, 100$) and for different variables sizes (p =10, 30, 50, 70) according to Algorithm 4.1.

2.4.3 Performance Measure

The performance of the original Graphical Lasso is compared to the Modified Graphical Lasso across different variable (p =10, 50, 70) and sample sizes (N = 5 to 100). For both the sparse and dense models, we know the true precision, $\boldsymbol{X}$. Algorithm 2.1 is used to approximate the optimal ρ^*, $\rho 1^*$ and $\rho 2^*$, and the corresponding optimal estimated precision, $\widehat{\boldsymbol{X}}$. The difference between $\widehat{\boldsymbol{X}}$ and $\boldsymbol{X}$ is then quantified using the Frobenius norm $\left\|\widehat{\boldsymbol{X}} - \boldsymbol{X}\right\|_F$, defined for an M x N matrix $\boldsymbol{A}$ as

$$||\boldsymbol{A}||_F = \sqrt{\sum_{m=1}^{M}\sum_{n=1}^{N} \boldsymbol{A}_{mn}^2}$$

Using the pseudoinverse as a baseline method for comparison, we expect that the Graphical Lasso and Modified Graphical Lasso will both perform better than the baseline pseudoinverse method for the sparse model. This is due to the fact that by nature, Graphical Lasso assumes a sparse model, and in such situations is expected to be a better approximation of the actual inverse covariance matrix than ordinarily inverting the covariance matrix or using its pseudoinverse. For the dense model, we expect the pseudoinverse to perform well especially at very high sample sizes relative to the number of variables.

2.5 Experiment Results

2.5.1 Model 1 Results

A sparse model: $(\mathbf{X})_{i,i} = 1$, $(\mathbf{X})_{i,i-1} = (\mathbf{X})_{i-1,i} = 0.5$, and 0 otherwise.

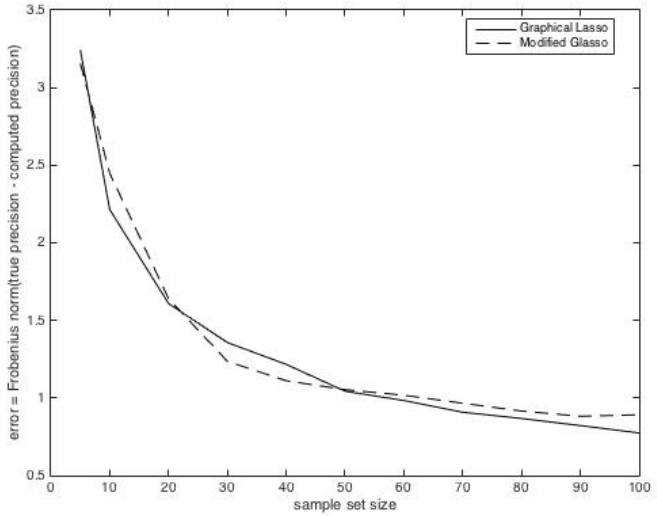

Figure 2.7 Model 1 error between Graphical Lasso and Modified Graphical Lasso (p =10).

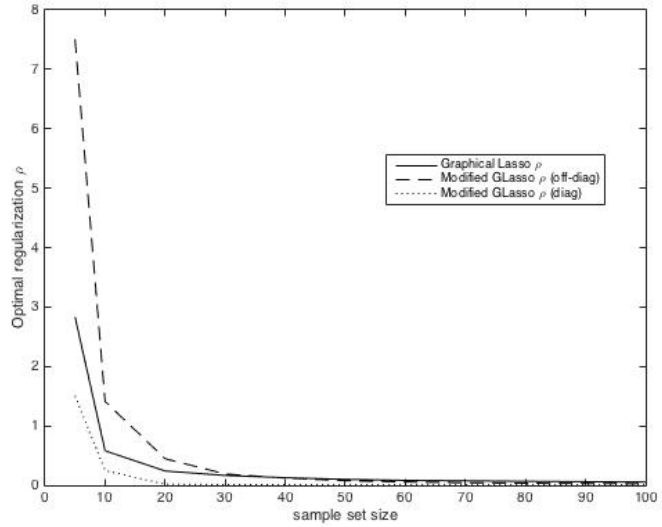

Figure 2.8 Model 1 optimal penalty for Graphical Lasso and Modified Graphical Lasso (p =10).

From Figure 2.7, it appears that at certain sample sizes, the Modified Graphical Lasso performs better than the original Graphical Lasso, having lower error. Statistical hypothesis tests (t-tests) are performed to verify these results and are presented in Appendix E. For better viewing, we present the actual values of the error and optimal regularizations for the Modified Graphical Lasso, the Graphical Lasso and the pseudoinverse in Appendix E also. Based on the results of the hypothesis tests in Appendix E Table E.1, for the sparse model at $p = 10$, the two methods perform essentially the same, except when $N > 70$ where the original Graphical Lasso performs statistically significantly better than the modified Graphical Lasso. The pseudoinverse performs worse than the Graphical Lasso and Modified Graphical Lasso as expected in this sparse scenario, although its performance gets better as the number of samples increase.

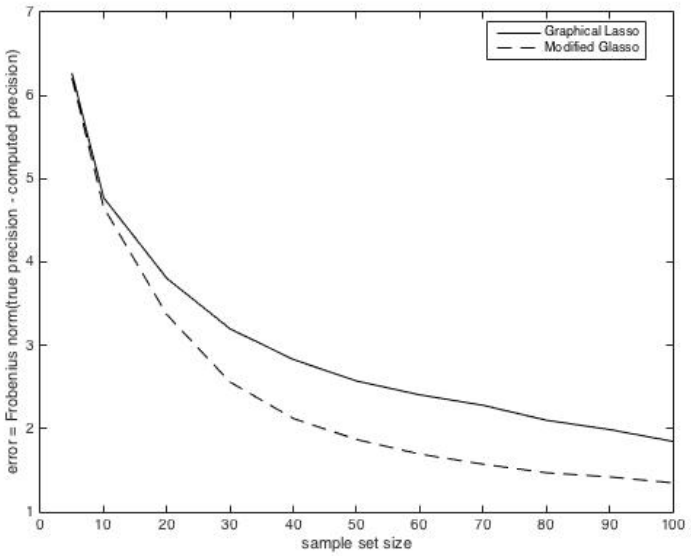

Figure 2.9 Model 1 error between Graphical Lasso and Modified Graphical Lasso (p =30).

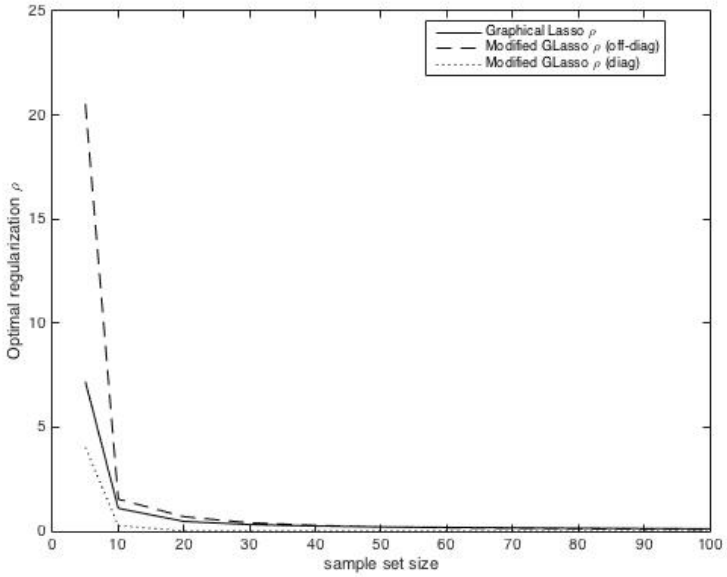

Figure 2.10 Model 1 optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=30).

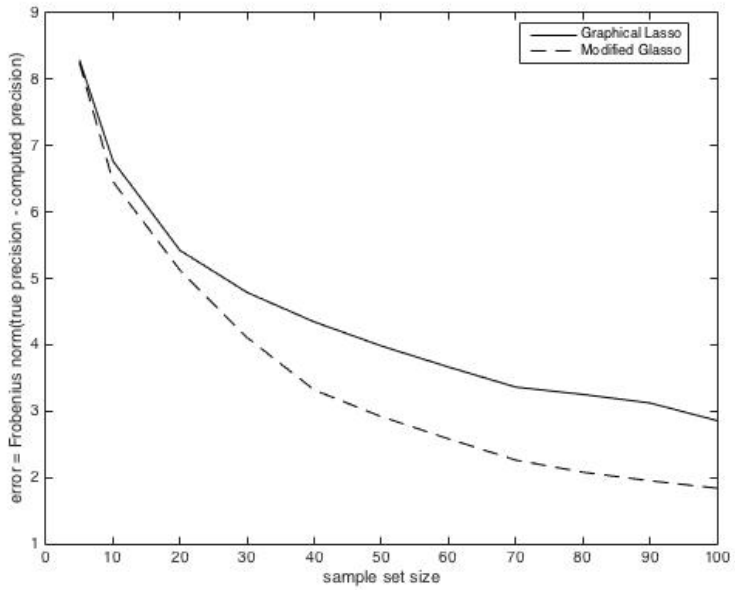

Figure 2.11 Model 1 error between Graphical Lasso and Modified Graphical Lasso (p =50).

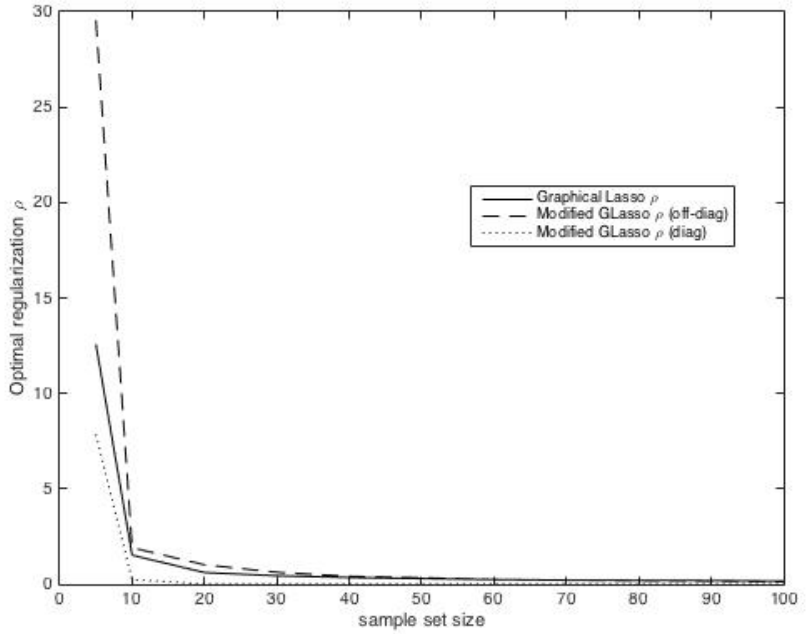

Figure 2.12 Model 1 optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=50).

Figure 2.9 shows the error across different samples for the sparse model when p=30, while Figure 2.11 shows the error across different samples for the sparse model when p=50. In these two figures, the Modified Graphical Lasso method appears to perform better than the Graphical Lasso method. Statistical hypothesis tests in Appendix E Table E.3 and Table E.5 however show that the two methods perform essentially the same across all samples, and both methods perform better than the pseudoinverse across all sample sizes. We now examine a larger variable size (p=70).

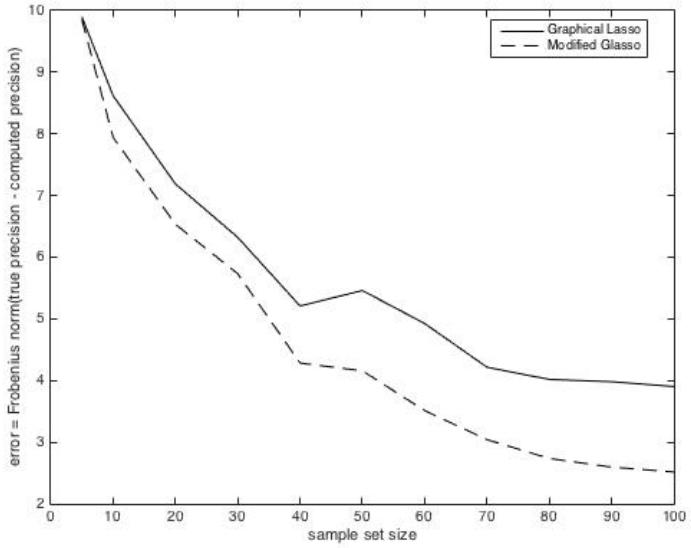

Figure 2.13 Model 1 error between Graphical Lasso and Modified Graphical Lasso (p =70).

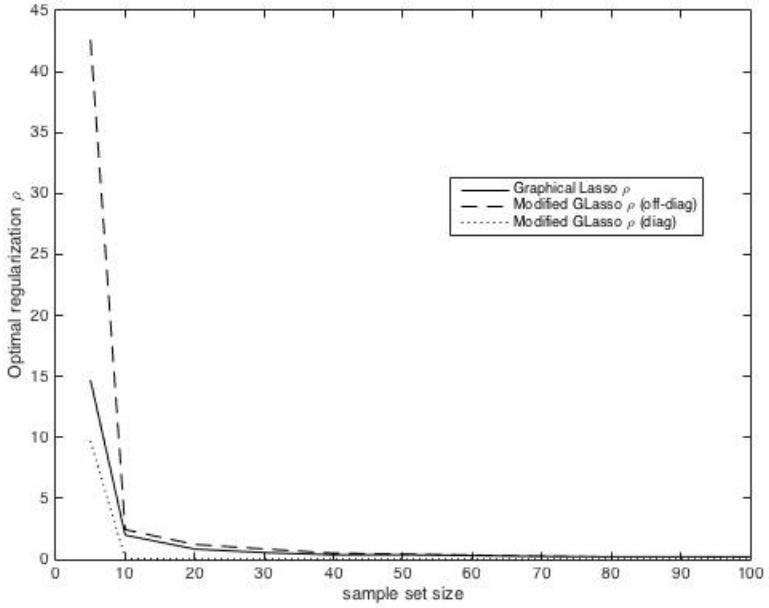

Figure 2.14 Model 1 optimal penalty for Graphical Lasso and Modified Graphical Lasso (p =70).

Figure 2.13 shows the error across different sample sizes for the sparse model when p=70. In this figure, the Modified Graphical Lasso method appears to perform better than the Graphical Lasso method. Statistical hypothesis tests in Appendix E Table E.7 however show that the two methods perform essentially the same across all samples, and both methods perform better than the pseudoinverse in general.

Figure 2.8, Figure 2.10, Figure 2.12 and Figure 2.14 show the optimal regularization as the sample size increases for the sparse model when p=10, p=30, p=50 and p=70 respectively. From these figures, we can see that the optimal regularization for the Graphical Lasso method is almost always an intermediate value between the optimal off-diagonal and diagonal regularizations for the Modified Graphical Lasso method across all sample sizes.

Based on the experiment in section 2.3.1 which showed the over-estimation of the diagonal elements as the regularization increases, we expected that for the Modified Graphical Lasso, a higher regularization will be needed for the diagonal elements, while a lower regularization will be needed for the off-diagonal elements, however, our results go against our hypothesis. Our results across all variable and sample sizes show that for the Modified Graphical Lasso method, the off-diagonal optimal regularization is higher than that of the diagonal elements.

2.5.2 Model 2 Results

A dense model: $(\mathbf{X})_{i,i} = 2$, $(\mathbf{X})_{i,i'} = 1$ otherwise

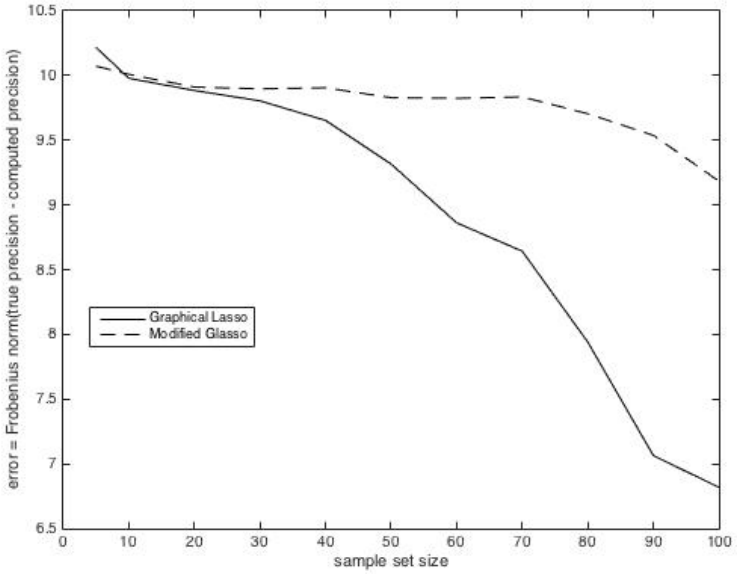

Figure 2.15 Model 2 Error between Graphical Lasso and Modified Graphical Lasso (p =10).

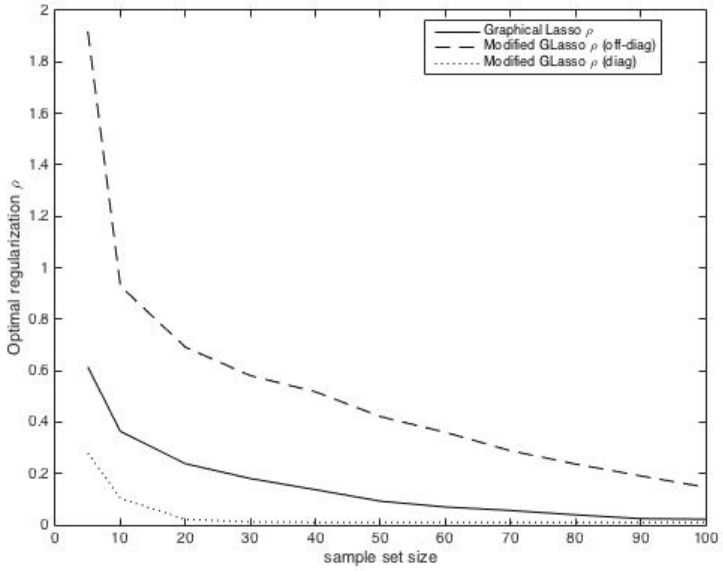

Figure 2.16 Model 2 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p =10).

Figure 2.15 shows the error across different samples for the dense model when p=10. From Figure 2.15, it appears that at most sample sizes, the Modified Graphical Lasso performs better than the original Graphical Lasso, having lower error. Statistical hypothesis tests in Appendix E Table E.9 however show that the original Graphical Lasso method performs statistically significantly better than the Modified Graphical Lasso when $N > 30$. When $N < 30$, the two methods perform essentially the same. The pseudoinverse performs better than both the Graphical Lasso and Modified Graphical Lasso when $N > 30$ as expected in the dense scenario, with its performance getting better as the number of samples increase.

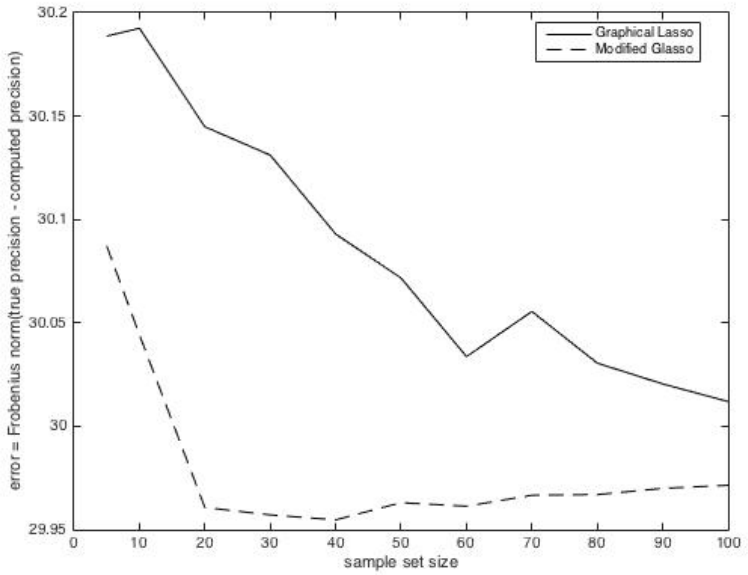

Figure 2.17 Model 2 Error between Graphical Lasso and Modified Graphical Lasso (p =30).

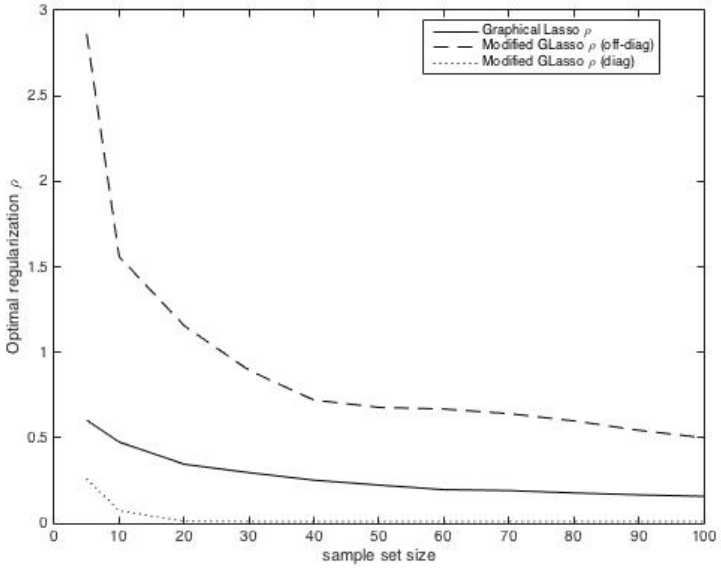

Figure 2.18 Model 2 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p =30).

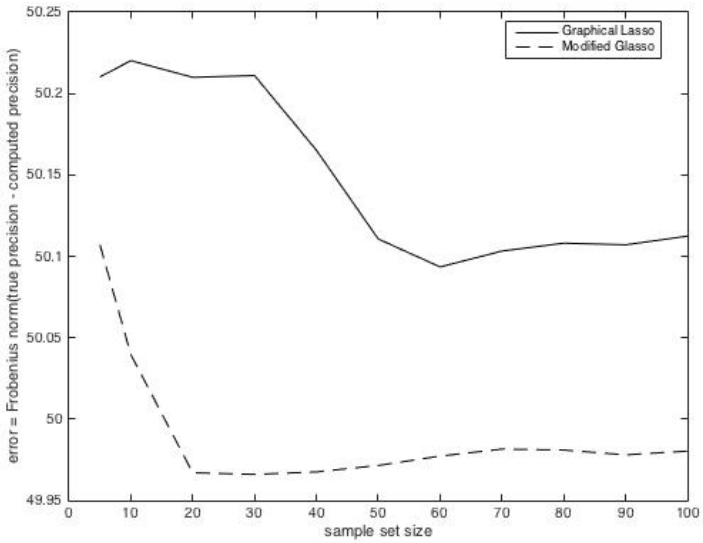

Figure 2.19 Model 2 error between Graphical Lasso and Modified Graphical Lasso (p =50).

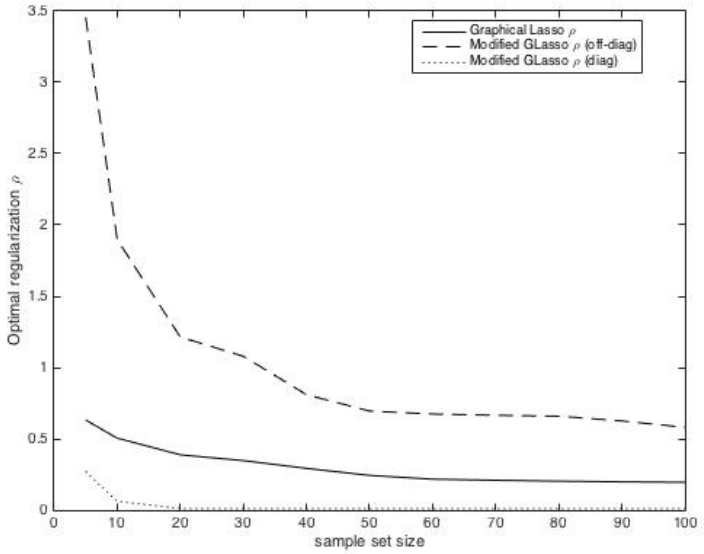

Figure 2.20 Model 2 optimal penalty for Graphical Lasso and Modified Graphical Lasso (p =50).

Figure 2.17 shows the error across different samples for the dense model when p=30, while Figure 2.19 shows the error across different samples for the sparse model when p=50. In these two figures, the Modified Graphical Lasso method appears to perform better than the Graphical Lasso method. Statistical hypothesis tests in Appendix E Table E.11 and Table E.13 however show that the two methods perform essentially the same across all samples. When p=30, both methods perform better than the pseudoinverse as seen in Appendix E Table E.11 when $N < 70$. When p=50, Appendix E Table E.13 shows that both methods perform better than the pseudoinverse when $N > 30$. We now look at performance when p=70.

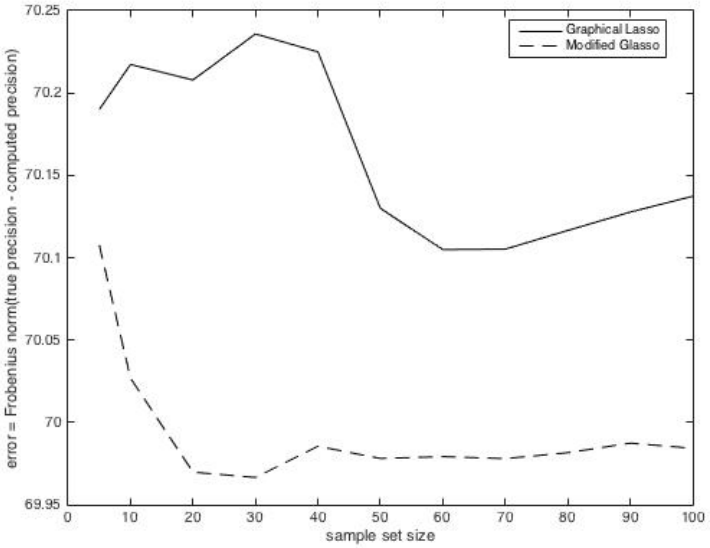

Figure 2.21 Model 2 error between Graphical Lasso and Modified Graphical Lasso (p =70).

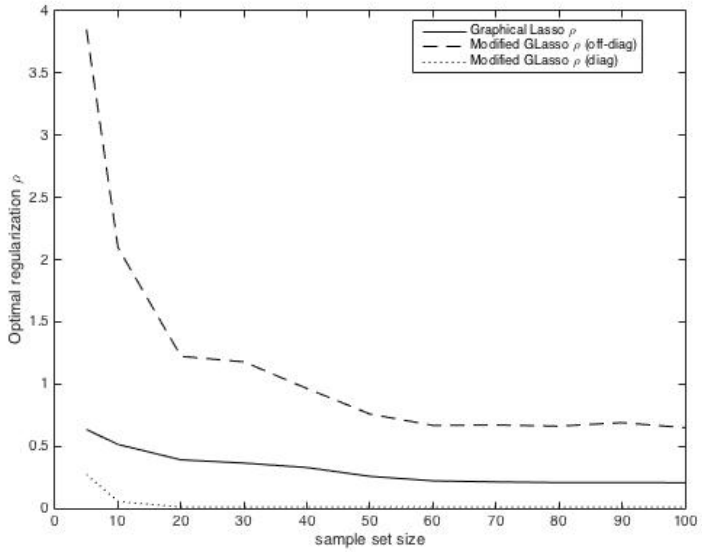

Figure 2.22 Model 2 optimal penalty for Graphical Lasso and Modified Graphical Lasso (p =70).

As we increase the variable size to p=70 for the dense model, Figure 2.21 shows the error across different sample sizes for the dense model when p=70. In this figure, the

Modified Graphical Lasso method appears to perform better than the Graphical Lasso method. Statistical hypothesis tests in Appendix E Table E.15 however show that the two methods perform essentially the same across all samples, and both methods perform better than the pseudoinverse when $N > 40$.

Figure 2.16, Figure 2.18, Figure 2.20 and Figure 2.2 show the optimal regularization as the sample size increases for the dense model when p=10, p=30, p=50 and p=70 respectively. From these figures, we can see that the optimal regularization for the Graphical Lasso method is always an intermediate value between the optimal off-diagonal and diagonal regularizations for the Modified Graphical Lasso method across all sample sizes.

Once again, based on the experiment in section 2.3.1 which showed the over-estimation of the diagonal elements as the regularization increases, we expected that for the Modified Graphical Lasso, a higher regularization will be needed for the diagonal elements, while a lower regularization will be needed for the off-diagonal elements, however, our results go against our hypothesis. Our results across all variable and sample sizes show that for the Modified Graphical Lasso method, the off-diagonal optimal regularization is higher than that of the diagonal elements.

2.6 Summary

In this chapter, we presented the Graphical Lasso algorithm which is the method that we use to estimate all sparse inverse covariances for application in bioinformatics and finance problems in this thesis. We showed the importance of regularization in section 2.3.1 and presented existing methods for approximating the optimal penalty

parameter in section 2.3.2. We pointed out a characteristic of Graphical Lasso to over-estimate the diagonal elements as the regularization amount increased, and presented a new method to remedy this problem by using two different penalties in section 2.4.1. We presented the results of this new method known as the 'Modified Graphical Lasso', which showed performance that was essentially the same as the original Graphical Lasso performance when $p > 10$ on synthetically generated data from sparse and dense inverse covariance models.

For the sparse model, the Graphical Lasso and Modified Graphical Lasso performed better than the pseudoinverse across all variable and sample sizes, while for the dense model, the pseudoinverse showed improved performance and performed as well or better than the Graphical Lasso and Modified Graphical Lasso methods at various instances, especially when the sample sizes were very high relative to the number of variables. These results were consistent with our hypothesis, where we expected the Graphical Lasso and Modified Graphical Lasso to perform better than the pseudoinverse when the data comes from a sparse model. In such scenarios, the Graphical Lasso is known to be a better approximation of the inverse covariance matrix than actually inverting the sample covariance matrix or using the pseudoinverse. Based on our results shown in section 2.3.1, we hypothesized that for the Modified Graphical Lasso, a higher regularization would be needed for the diagonal elements to remedy the diagonal over-estimation problem, while a lower regularization would be needed for the off-diagonal elements. However, our results went against our hypothesis. Our results were optimal when the off-diagonal elements had a higher regularization than the diagonal-elements.

Chapter 3

Graphical Lasso Application to Bioinformatics

Machine learning and data mining have found a multitude of successful applications in microarray analysis, with gene clustering and classification of tissue samples being widely cited examples [32]. Deoxyribonucleic acid (DNA) microarray technology provides useful tools for profiling global gene expression patterns in different cell/tissue samples. One major challenge is the large number of genes p relative to the number of samples N. The use of all genes can suppress or reduce the performance of a classification rule due to the noise of non-discriminatory genes [33]. Selection of an optimal subset from the original gene set becomes an important pre-step in sample classification [33].

In this chapter, we propose the use of the sparse inverse covariance estimator, Graphical Lasso, which was introduced in chapter 2 to estimate the inverse covariance matrix even when $N < p$. The estimated sparse inverse covariance is used for dimensionality reduction and to classify tissue samples given gene microarray data.

3.1 Introduction

DNA microarrays enable scientists to study an entire genome's expression under a variety of conditions. The advent of DNA microarrays has facilitated a fundamental shift from gene-centric science to genome-centric science [34].

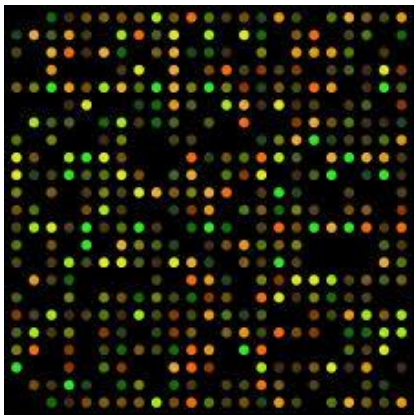

Figure 3.1 DNA microarray image.

DNA microarrays are typically constructed by mounting a unique fragment of complementary DNA (cDNA) for a particular gene to a specific location on the microarray [34]. This process is repeated for *N* genes. The microarray is then hybridized with two solutions, one containing experimental DNA tagged with green fluorescent dye, the other containing reference or control DNA tagged with red fluorescent dye [34].

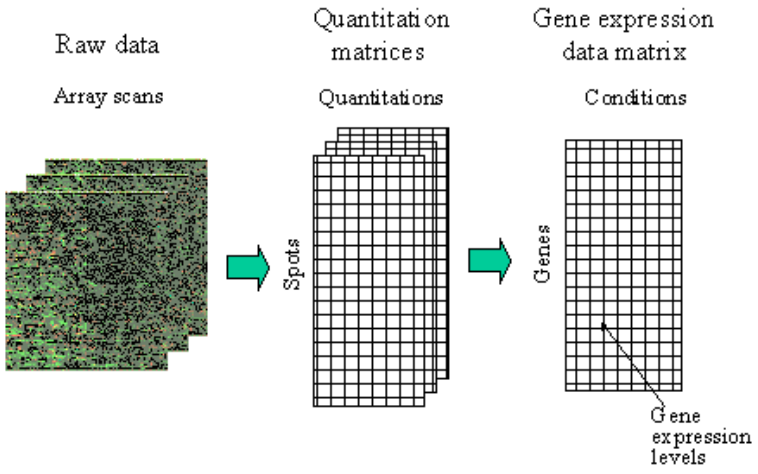

Figure 3.2 Acquiring the gene expression data from DNA microarray.

DNA microarrays are composed of thousands of individual DNA sequences printed in a high density array on a glass microscope slide using a robotic arrayer as shown in Fig. 3.2. The relative abundance of these spotted DNA sequences in two DNA or RNA samples may be assessed by monitoring the differential hybridization of the two samples to the sequences on the array [34]. For mRNA samples, the two samples are reverse-transcribed into cDNA, labeled using different fluorescent dyes mixed (red-fluorescent dye and green-fluorescent dye). After the hybridization of these samples with the arrayed DNA probes, the slides are imaged using a scanner that makes fluorescence measurements for each dye [34]. The log ratio between the two intensities of each dye is used as the gene expression data [35-37].

Traditionally, genes have been studied in isolation in an attempt to characterize their behavior [34]. While this technique has been successful to a limited extent, it suffers from several fundamental drawbacks. The most significant of these drawbacks is the

fact that in a real biological system, genes do not act alone; rather, they act in concert to affect a particular state in a cell [34]. As such, examining cells in isolation offers a perturbed and very limited view of their function [34]. DNA microarrays allow a scientist to observe the expression level of tens of thousands of genes at once. Rather than considering individual genes, a scientist now has the capability of observing the expression level of an entire genome.

The power of DNA microarrays is a double-edged sword: to handle the enormous amount of data being generated by microarray experiments, we need sophisticated data analysis techniques to match [34]. More specifically, we need to extract biologically meaningful insights from the morass of DNA microarray data, and apply this newly gained knowledge in a meaningful way. The types of information scientists want to extract from DNA microarray data can be regarded as patterns or regularities in the data [34]. One important application of gene expression data is classification of samples into categories. For example, a scientist may want to discover which samples belong to particular tissue or which samples belong to healthy/unhealthy patients. They may also want to know which genes are co-regulated, or attempt to infer what the gene expression regulator pathways are [34]. Alternatively, a doctor may want to know if the gene expression profile of an unhealthy patient can help predict an optimal treatment. In combination with classification methods, other machine learning techniques are designed to extract such patterns which can be useful for supporting clinical management decisions for individual patients, e.g. in oncology. Standard statistic methodologies in classification or prediction do not work well

when the number of variables p (genes) is much larger than the number of samples N which is the case in gene microarray expression data [34].

3.1.1 DNA Microarray Data Characteristics

While offering unprecedented research opportunities, DNA microarray data also present a new set of challenges. There are a variety of challenges facing scientists who attempt to work with DNA microarray data [34].

3.1.1.1 The Curse of Dimensionality

One of the main challenges in dealing with microarray data is the dimensionality of the data [34]. If we represent a 10,000-gene microarray experiment as a vector, we are forced to work in 10,000-dimensional space. For an algorithm to work effectively, it must be able to deal robustly with the dimensionality of this feature space. In some cases, the curse of dimensionality is offset by having a large number of samples to use in data analysis [34]. Unfortunately, it is often the case that microarray data sets are composed of tens or hundreds of samples, not thousands as one would hope. Machine learning algorithms must thus be able to deal not only with high dimensional data, but relatively small data sets from which to learn [34].

3.1.1.2 Noise and Data Normalization

Another fundamental challenge in dealing with DNA microarray data is data normalization [34]. Due to technical limitations, the constant of proportionality between the actual number of mRNA samples per cell and the relative amount

measured by a microarray experiment is unknown, and will vary across microarray experiments [34]. This variance introduces noise to experiments, and requires that we normalize microarray data by multiplying array results by an appropriate factor. Understanding the normalization process is key, as it will affect the ability to determine whether a gene's varying expression level is meaningful or simply a byproduct of noise [34].

3.2 Graphical Models and Gene Microarray Data

DNA microarrays remain a powerful tool for identifying changes in gene expression between different environmental conditions or developmental stages [38]. Coordinated regulation of gene expression is typically studied by identifying groups of genes with correlated changes in mRNA abundance across different experimental conditions [38]. Recently, gene co-expression networks have become a more and more active research area [39-42]. Recent computational methods attempt to reconstruct networks of gene regulation from global expression patterns [38]. A gene co-expression network is essentially a graph where nodes in the graph correspond to genes, and edges between genes represent their co-expression relationship [38]. Traditionally, several clustering techniques have been studied in literature, but such methods neglect gene neighbor relations (such as topology) in the networks [41]. In order to elucidate functional interaction, and as a basis for subsequent clustering and network inference, a popular strategy in bioinformatics is to compute the standard Pearson correlation between any two genes [43]. If the correlation coefficient exceeds

a certain *a priori* specified threshold, then an edge is drawn between the appropriate genes. The resulting graph is called a "*relevance network*" where missing edges denote marginal independence [43]. In statistical terminology, this type of network model is also known as a "*covariance graph model*". However, for understanding gene interaction, this approach is only of limited use.

For instance, a high correlation coefficient between two genes may be indicative of either (i) direct interaction, (ii) indirect interaction, or (iii) regulation by a common gene. In learning a genetic network from data we need to be able to distinguish among these three alternatives. Therefore, for constructing a "*gene association network*" where only direct interactions among genes are depicted by edges, another framework is needed: "*graphical Gaussian models*" (GGMs) [43]. The key idea behind GGMs is to use partial correlations as a measure of conditional independence between any two genes. This overcomes the edge identifiability problems of standard correlation networks [43]. Consequently, GGMs (also known as "*covariance selection*" or "*concentration graph*" models) have recently become a popular tool to study gene association networks. Note that GGMs and the covariance graph models are only superficially similar approaches. However, both conceptually as well as practically, they constitute completely different theories [43].

Graphical models [26, 44] are promising tools for the analysis of gene interaction because they allow the stochastic description of net-like association and dependency structures in complex highly structured data. At the same time, graphical models offer an advanced statistical framework for inference. In theory, this makes them perfectly suited for modeling biological processes in the cell such as biochemical interactions

and regulatory activities [43]. However, the practical application of graphical models in systems biology is strongly limited by the amount of available experimental data. This apparent paradox arises as today's high-throughput facilities allow to investigate experimentally a greatly increased number of features while the number of samples has not, and cannot, similarly be increased [43]. For instance, in a typical microarray data set, the number of genes p will exceed by far the number of sample points N. This poses a serious challenge to any statistical inference procedure, and also renders estimation of gene regulatory networks an extremely hard problem [43]. This is corroborated by a recent study on the popular Bayesian network method where [45] demonstrated that this approach tends to perform poorly on sparse microarray data. In this chapter, we focus on sparse graphical Gaussian models for modeling genome data. These are similar to the bioinformatics community widely applied "*relevance networks*" in that edges indicate some degree of correlation between two genes. However, in contrast to correlation networks, GGMs allow one to distinguish direct from indirect interactions, i.e. whether gene A acts on gene B directly or via mediation through a third gene C [43]. More precisely, GGMs are based on the concept of conditional independence. In this respect, GGMs behave similarly as Bayesian networks. However, unlike the latter, GGMs contain only undirected edges, hence they do not suffer from a restriction inherent in Bayesian networks, namely that they can only be applied to network graphs without feedback loops, i.e. directed cycles [43]. [46] show how Gaussian graphical models are useful in gene expression data. The notion of sparsity of graphical models for gene expression data reflects the view that patterns of variation in expression for a gene, G, will be well predicted by those

of a relatively small subset of other genes. Beyond parsimony, this embodies a view that transcriptional regulation of a single gene or pathway component is generally defined by a small set of regulatory elements [46].

Unfortunately, a number of difficulties arise when the standard graphical Gaussian modeling concept is applied for the analysis of high-dimensional data such as from a microarray experiment [43]. First, classical GGM theory [26] may only be applied when $N > p$, because otherwise the sample covariance and correlation matrices are not well conditioned, which in turn prevents the computation of partial correlations. Moreover, often there are additional linear dependencies between the variables, which leads to the problem of multicollinearity [43]. This, again, renders standard theory of graphical Gaussian modeling inapplicable to microarray data. Second, the statistical tests widely used in the literature for selecting an appropriate GGM (e.g. deviance tests) are valid only for large sample sizes, and hence are inappropriate for the very small sample sizes present in microarray data sets. In this case, instead of asymptotic tests, an exact model selection procedure is required [43]. Note that the small N large p problem affects both GGMs and relevance networks [43]. In particular, the standard correlation estimates are not valid for small sample size N, a fact that appears to have gone largely unnoticed in the bioinformatics community [43].

3.3 Microarray Data Analysis using Graphical Lasso

In this section, we propose the use of the sparse inverse covariance estimator, Graphical Lasso, for gene microarray data analysis. We create graphical Gaussian

models, which we use to perform the supervised classification task of identifying tissue samples.

3.3.1 The Data

Background information of patient recruitment strategy and subsequent transcriptional profiling was carried out at University College London Hospital (Principal Investigator: Dr. Mahdad Noursadeghi, Infection and Immunity, University College London). The data was generously provided by Professor Benjamin Chain and Dr. Nandi Simpson (Infection and Immunity, University College London).

Participants were recruited at the University College London Hospital for a study program. The cohort consisted of UK participants who were given Unique Identification Numbers (UIN) by the research nurse at the end of the study leading to the data being anonymized. The microarray data set consists of 255 cell samples isolated and/or cultured from human white blood cells (in all cases healthy human volunteers). Each sample is associated with gene expression levels of 4100 genes and from 1 of 11 different human tissue types under varying stimuli and exposure times. In the microarray data representation, each row represents a sample and each column represents a gene. Gene selection was performed by [47] such that the genes whose variances across all samples that were within the top 90% quartile were those chosen out of a total of 40,000 genes, reducing the gene size to 4100. The basis for selecting 4100 genes out of 40,000 is that, these genes showed the highest variation across all the samples, and with gene expression data, the most important dynamics

are the ones with the largest variance. Such differentially expressed genes are viewed as potential candidates that may provide clues to the behavior of a system under a given perturbation.

The microarray data was then processed using a previously developed pipeline by [47]. In the paper, the authors have described and validated a series of data extraction, transformation and normalization steps which are implemented through a package in R known as 'agilp'. The raw data was pre-processed according to the following steps:

1. Log2 transformation of the raw data.
2. Data normalization using LOESS normalization (Local Regression, also known as Locally Weighted Scatter Plot Smoothing) i.e. the assumption is that most genes in a specific range of signal do not change.

For ease of representation in the rest of this chapter, each gene will correspond to a number between 1 and 4100 and each tissue type will correspond to a class between 1 and 11. The distribution of the tissue samples is shown in Table 3.1.

Class	Corresponding Tissue	Tissue Description	No of Samples
1	D4_MDDC	Monocyte derived dendritic cells after 4 days culture	27
2	DC	Dendritic cells isolated directly from blood	24
3	MDDC	Monocyte derived dendritic cells after 7 days culture	32
4	MDM	Monocyte derived macrophages	53
5	BLOOD	Unfractionated sample of white blood cells	46
6	M3DC	Dendritic cells derived from Mutz3 cell line	15
7	AM	Alveolar macrophages	14
8	HC	HACAT cells ; a human keratinocyte tumour cell line	16
9	HL	Hela cells : a cervical cancer cell line	12
10	2T	3T3 cells. A human transformed fibroblast line	8
11	HOS	A bone osteosarcoma line	8

Table 3.1 The 11 different tissue types and the distribution of the 255 tissue samples

3.3.2 Microarray Classification Problem Definition

In a microarray classification problem, we are given a training dataset of N samples which we represent as a training dataset of N training sample pairs: $\{x_i , y_i \}$, $i = 1, \ldots, N$, where $x_i \in R^p$ is the i-th training sample, and y_i is the corresponding class label, which is either +1 or -1.

An important goal of the analysis of such data is to determine an explicit or implicit function that maps the points of the feature vector from the input space to an output space. This mapping has to be derived based on a finite number of data, assuming that a proper sampling of the space has been performed. If the predicted quantity is

categorical and if we know the value that corresponds to each element of the training set, then the question becomes how to identify the mapping that connects the feature vector and the corresponding categorical value.

3.3.3 The Proposed Method

Assuming a sparse model, we use Graphical Lasso to define graphical Gaussian models of varying sparsity in section 3.3.5. We perform dimensionality reduction by running Graphical Lasso on the entire microarray data using a chosen regularization, which defines the corresponding graphical Gaussian model in section 3.3.6. Specifically, we use the *k*-nearest neighbor technique introduced in Appendix A section A.4.2 and defined in section 3.3.6 to reduce the dimensionality of the graphical Gaussian model or baseline covariance model (defined by the sample covariance). We then perform feature selection (introduced in Appendix A section A.2.1 and A.4.5) by treating connected components (sub-networks) in each model (graphical Gaussian model or covariance model) as individual features with the feature value being the average expression level of all the genes in a particular sub-network across all samples. Using the new feature space, we perform supervised classification using regression.

3.3.4 Data Pre-processing: Centering the Data

Prior to the application of many multivariate methods, data are often pre-processed. This pre-processing involves transforming the data into a suitable form for the analysis. Among the different pre-processing methods, one of the most common

operations is well known as mean-centering. Mean centering involves the subtraction of the variable averages from the data. Since multivariate data is typically handled in matrix format with columns as variables, mean-centering is often referred to as *column centering*. In mean-centering, one calculates the average value of each variable and then subtracts it from the data. This implies that each column will be transformed in such a way that the resulting variable will have a zero mean. In addition to the initial pre-processing steps applied to the microarray data by [47] and described in section 3.3.1, we performed column centering. Recalling that the microarray data consists of 4100 genes and 255 samples, let $\mathbf{X}$ represent the microarray data where each row i is a sample and each column j is a gene. The data centering is defined as follows

$$\mathbf{X}^c = \sum_{i=1}^{255} \mathbf{X}_{i,j} \sum_{j=1}^{4100} \overline{\mathbf{X}}_j, \tag{3.1}$$

where $\overline{\mathbf{X}}_j$ is the mean gene expression level for a particular gene j and $\mathbf{X}^c$ is the newly centered data. Figure 3.3 and Figure 3.4 show the distribution of the mean expression levels of all 4100 genes across all samples before and after centering the data.

The final step in data pre-processing involved re-organizing the microarray data according to tissue samples i.e. samples belonging to the same tissue type were grouped together.

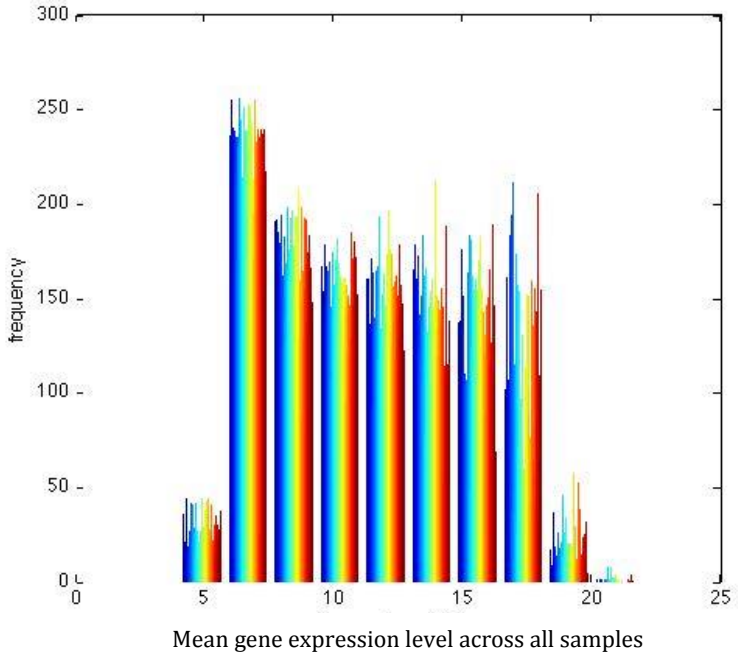

Figure 3.3 The distribution of the microarray data consisting of 4100 genes and 255 samples before centering.

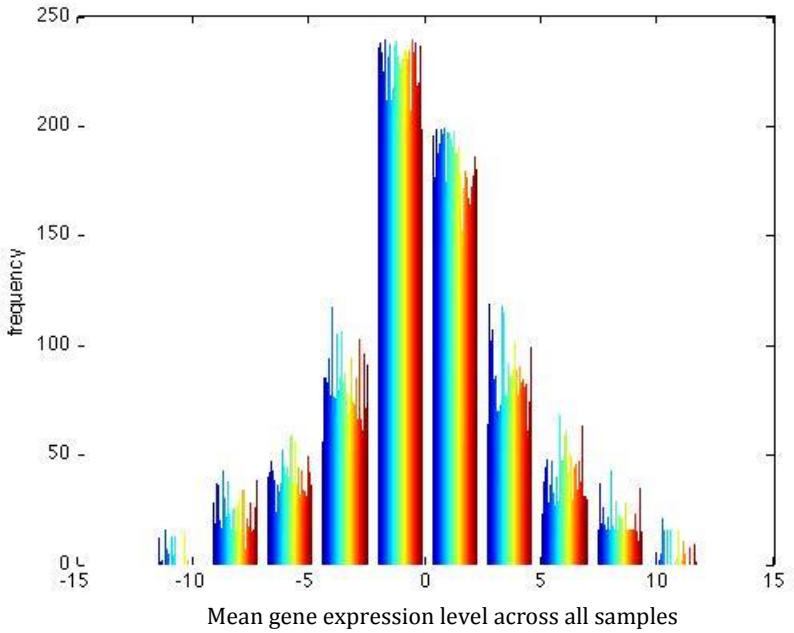

Figure 3.4 The distribution of the microarray data consisting of 4100 genes and 255 samples after centering.

3.3.5 Choosing the Regularization

In order to visualize the interaction of the genes, we use Graphical Lasso to generate graphical Gaussian models representing the gene microarray data. If we recall from chapter 2, one of the important prerequisites of Graphical Lasso is choosing the correct regularization amount. To choose the correct regularization to represent the microarray data set, we first look at how the sparsity pattern varies for randomly chosen regularizations of ρ=1 and ρ=3 and ρ=10 and choose to use two different regularizations of ρ=1 and ρ=3 for the rest of the experiments in this chapter. The sparsity of the randomly chosen regularizations can be seen in Figure 3.5, Figure 3.6 and Figure 3.7.

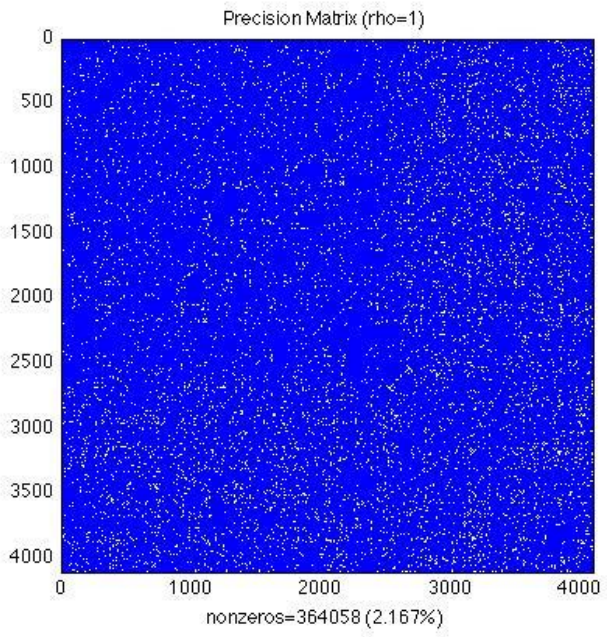

Figure 3.5 Sparsity of estimated precision (ρ=1).

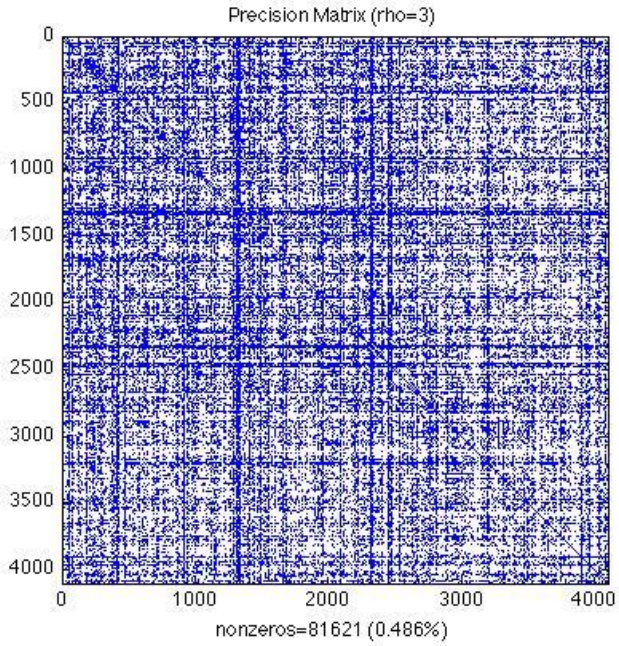

Figure 3.6 Sparsity of estimated precision (ρ=3).

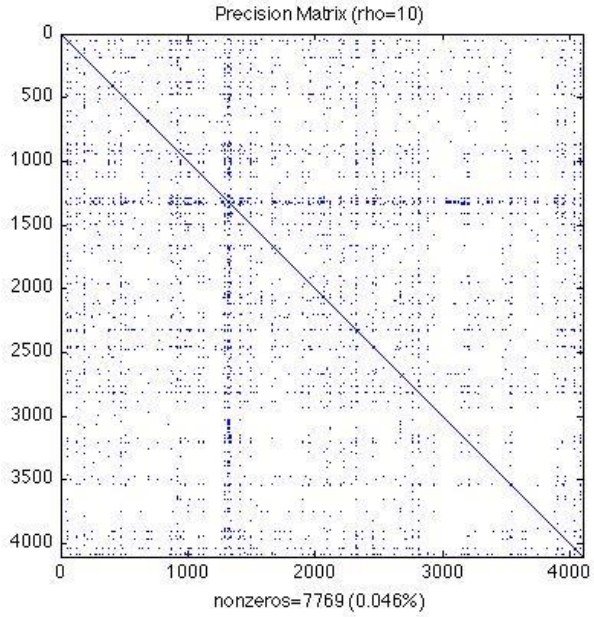

Figure 3.7 Sparsity of estimated precision (ρ=10).

In order to test the stability of the Graphical Lasso estimated precisions, we see how the structure of the precision is maintained as the regularization amount ρ is increased. We start with a baseline precision and estimate a new precision at a higher regularization and see how many zeros are estimated correctly in the new precision. We vary the number of iterations from 10 to 250 and report the average results in Table 3.2. The results remained constant regardless of the number of iterations performed.

Baseline Precision	New Precision	No of zeros in baseline	Identified Zeros from Baseline (%)	Unidentified Zeros from Baseline (%)
ρ=1	ρ=3	16437743	99.63	0.37
ρ=1	ρ=10	16437743	99.98	0.02
ρ=3	ρ=10	16720180	99.98	0.02

Table 3.2 Structure of the Precision as regularization ρ is increased

3.3.6 *k*-Nearest Neighbour Graphs to Visualize Gene Interaction

Adjacency matrices, $\boldsymbol{A}$, are created from the estimated inverse covariance matrix following Algorithm 3.1.

Algorithm 3.1 Building adjacency matrices

1: Given the estimated inverse covariance $\widehat{\boldsymbol{X}}$, Let $\boldsymbol{A}$ represent an adjacency matrix
2: **for** $i = 1{:}\ 4100$
3: **for** $j = 1{:}\ 4100$
4: **if** $\widehat{\boldsymbol{X}}_{i,j} \neq 0$
5: $\boldsymbol{A}_{i,j} = 1$ and $\boldsymbol{A}_{j,i} = 1$
6: **else**
7: $\boldsymbol{A}_{i,j} = 0$ and $\boldsymbol{A}_{j,i} = 0$
8: **end if**
8: **end for**
9: **end for**

We additionally define an adjacency matrix $\boldsymbol{A}$ for the baseline sample covariance by following Algorithm 3.1 and replacing $\widehat{\boldsymbol{X}}$ with the sample covariance $\boldsymbol{\Sigma}$.

Next, k-nearest neighbor (k-NN) graphs are built by following Algorithm 3.2. We take the top 20 largest entries (in terms of magnitude) in both the sample covariance and estimated precisions and their 3 nearest neighbors($k = 3$). The idea behind this is to capture the strongest interactions in the sample covariance and estimated precisions, so as to avoid working with the entire graphs which are very large.

Algorithm 3.2 Building k-nearest neighbour graph

1: Given adjacency matrix $\boldsymbol{A}$ and $k = 3$
2: Define new adjacency matrix $\boldsymbol{A}^*$ for graph building
3: Define a matrix $\boldsymbol{Y}$ corresponding the top 20 i, j positions corresponding to the row
4: and column of the largest magnitudes in $\widehat{\boldsymbol{X}}$ with 20 rows and 2 columns
5: **for** $i = 1{:}\ 20$
6: edge_pair = $\boldsymbol{Y}(i, :)$ and is a vector containing the i, j positions from $\boldsymbol{Y}$
7: Let $\boldsymbol{z}$ be the variable set corresponding to the top 3 shortest distances from
8: $\boldsymbol{A}$(edge_pair) to all other variables in $\boldsymbol{A}$, where distance is the Euclidean
9: distance defined in Appendix A equation (A.14).
10: $\boldsymbol{A}^*$(edge_pair) $= 1$ and $\boldsymbol{A}^*$(edge_pair′) $= 1$
11: **for** $j = 1{:}\,\text{length}(\boldsymbol{z})$
12: $\boldsymbol{A}^*_{\text{edge\_pair}(1),\boldsymbol{z}(j)} = 1$ and $\boldsymbol{A}^*_{\boldsymbol{z}(j),\text{edge\_pair}(1)} = 1$
13: **end for**
14: **end for**
15: Use $\boldsymbol{A}^*$ to build graph by drawing an edge between variables i and j if $\boldsymbol{A}^*_{i,j} = 1$.
16: The edge $\boldsymbol{A}^*_{i,j}$ is red if $\widehat{\boldsymbol{X}}_{i,j} > 0$ and black if $\widehat{\boldsymbol{X}}_{i,j} < 0$

':' represents all the rows or all the columns in a matrix and ′ represents the symmetric counterpart of an edge i.e. the symmetric counterpart of edge (i, j) is (j, i)

We build k-nearest neighbour graphs corresponding to the estimated precisions and sample covariance of the microarray data. These graphs can be seen in Figure 3.8,

Figure 3.9 and Figure 3.10. The graph in Figure 3.8 for the baseline sample covariance is built by following Algorithm 3.2 and replacing $\widehat{\boldsymbol{X}}$ with the sample covariance $\boldsymbol{\Sigma}$.

In the figures, node names correspond to the gene row entries in the original microarray data (a number between 1 and 4100). Red edges represent positive values in the precision/covariance matrix while black edges represent negative values.

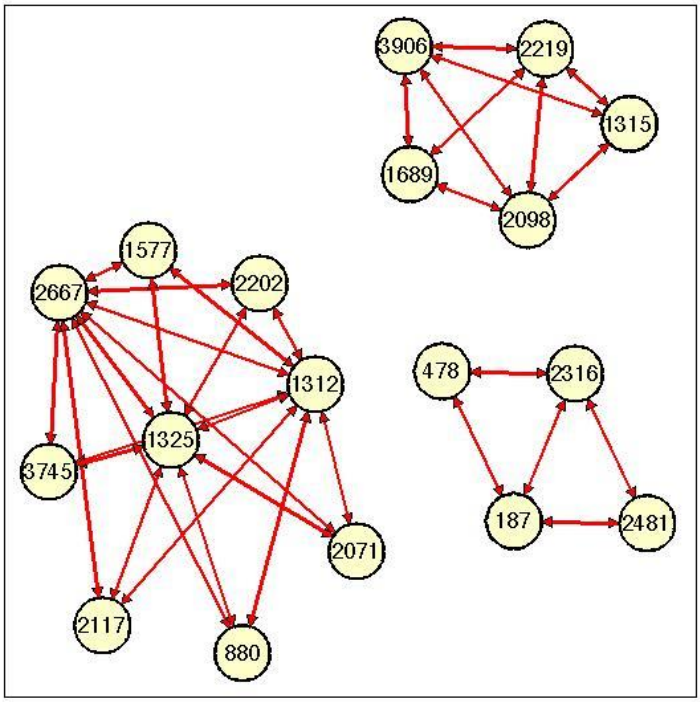

Figure 3.8 k-nearest neighbor graph ($k = 3$) *built from the microarray data sample covariance. Each node number represents a gene (genes are annotated from 1 to 4100). This graph shows 3 separate sub-networks.*

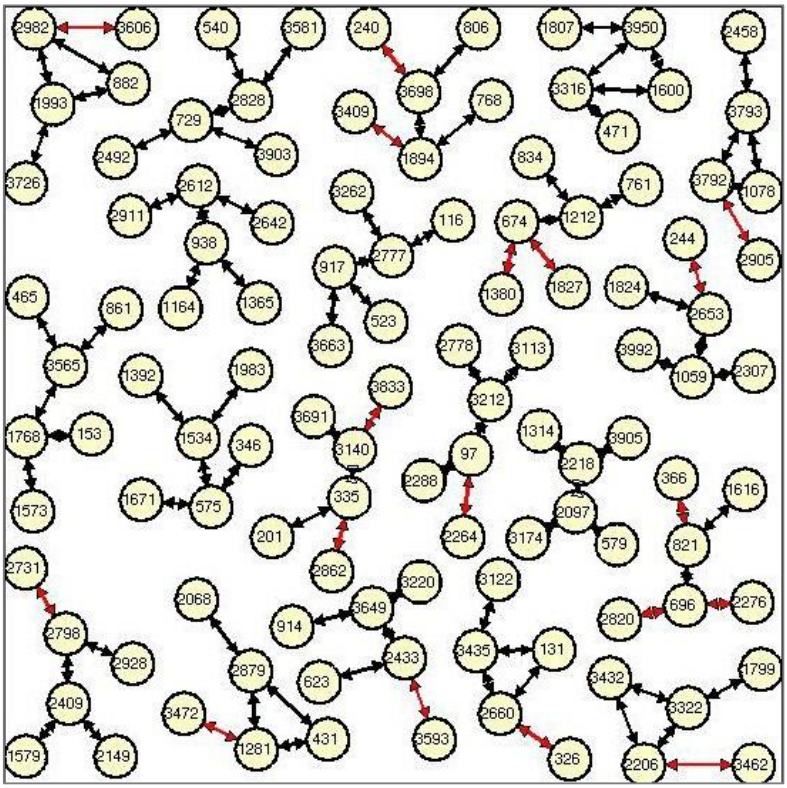

Figure 3.9 k-nearest neighbor graph $(k = 3)$ *built by running Graphical Lasso on the microarray data at a regularization of ρ=1. Each node number represents a gene (genes are annotated from 1 to 4100). This graph shows 20 separate sub-networks.*

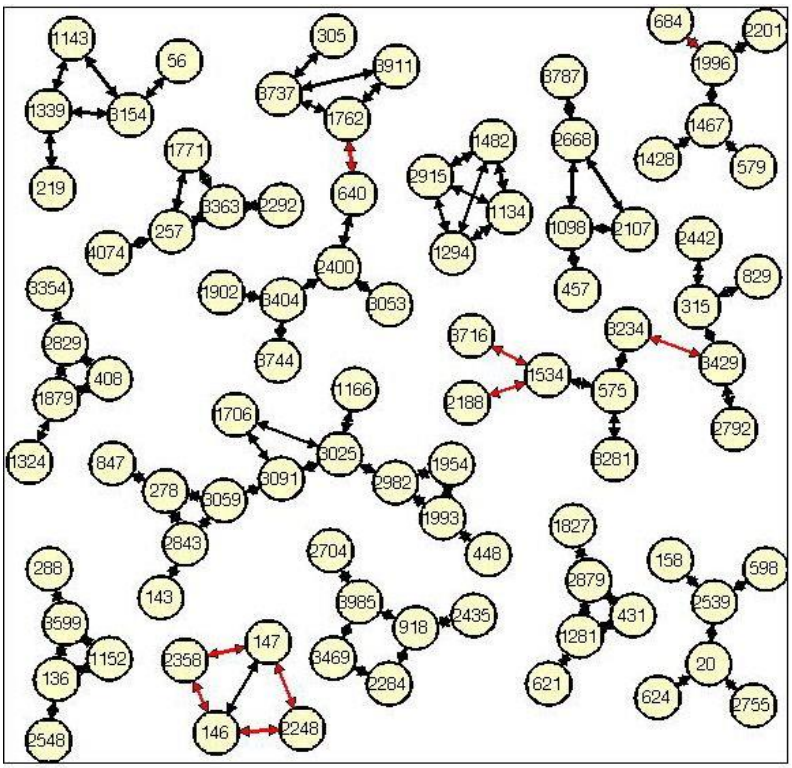

Figure 3.10 k-nearest neighbor graph $(k = 3)$ *built by running Graphical Lasso on the microarray data at a regularization of ρ=3. Each node number represents a gene (genes are annotated from 1 to 4100). This graph shows 14 separate sub-networks.*

3.3.7 Supervised Learning Methodology

To use the graphical Gaussian models and the covariance model presented in section 3.3.6 for the tissue classification task, we follow the methodology described in section 3.3.3. Specifically, we use the graphs in Figure 3.8, Figure 3.9 and Figure 3.10, which represent the microarray data with its dimension reduced. For each graph, we perform feature selection by treating connected components (sub-networks) as individual features, with the feature value being the average expression level of all the genes in a particular sub-network across all samples.

By doing this, we end up with a feature vector for the covariance which consists of 3 features representing the average expression levels of each of the 3 sub-networks in the *k*-nearest neighbour graph in Figure 3.8 across all 255 samples. For the estimated precision at ρ=1, we end up with a feature vector which consists of 20 features representing the average expression level of each of the 20 networks in the *k*- nearest neighbour graph in Figure 3.9 across all 255 samples. For the estimated precision at ρ=3, we end up with a feature vector which consists of 14 features representing the average expression levels of each of the 14 networks in the *k*- nearest neighbour graph in Figure 3.10 across all 255 samples.

We are essentially reducing the dimensionality of the problem from a 4100 dimensional problem with 255 samples to a 3 dimensional problem for the covariance, a 20 dimensional problem for the precision at ρ=1 and a 14 dimensional problem for the precision at ρ=3. These feature vectors will be used in regression for classification, where $x \in R^d$ is the feature vector and $y \in R$ is the tissue label.

3.3.7.1 Linear Least Squares Regression Classifier

Fit a function $g(\boldsymbol{x})$ through a set of samples S $= \{(x_1, y_1),\ (x_2, y_2), \ldots\ldots\ldots (x_l, y_l)\}$, where $\boldsymbol{x} \in R^p$ and $y \in R$

We look for:

$$y_i \cong g(\boldsymbol{x}_i) = \boldsymbol{x}_i^T \boldsymbol{w}$$

The weight vector that minimizes the mean of the squared errors on all training samples is:

$$\boldsymbol{w}^* = argmin_w \frac{1}{l}\sum_{i=1}^{l}(\boldsymbol{x}_i^T\boldsymbol{w} - y_i)^2$$

Using the notation $\mathbf{X} = (\boldsymbol{x}_1\ \boldsymbol{x}_2 \dots \dots \dots \boldsymbol{x}_l)^T$, a matrix containing training sample vectors as its rows, the cost function can be rewritten as:

$$\boldsymbol{w}^* = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$$

A detailed description of linear least squares regression can be found in Appendix A section A.5.

3.3.7.2 Ridge Regression Classifier

Fit a function $g(\boldsymbol{x})$ through a set of samples S $= \{(x_1, y_1),\ (x_2, y_2), \dots \dots . (x_l, y_l)\}$, where $\boldsymbol{x} \in R^p$ and $y \in R$

We look for:

$$y_i \cong g(\boldsymbol{x}_i) = \boldsymbol{x}_i^T\boldsymbol{w}$$

Using the notation $\mathbf{X} = (\boldsymbol{x}_1\ \boldsymbol{x}_2 \dots \dots \dots \boldsymbol{x}_l)^T$, a matrix containing training sample vectors as its rows, the cost function can be rewritten as:

$$\boldsymbol{w}^* = (\mathbf{X}^T\mathbf{X} + \gamma l\boldsymbol{I})^{-1}\mathbf{X}^T\boldsymbol{y},$$

where γ is the penalty parameter and $\boldsymbol{I}$ is a $p \times p$ identity matrix.

A detailed description of ridge regression can be found in Appendix A section A.5.2.1.

3.4 Two-class Classification to Identify Tissue Samples

We perform two-class (binary) classification, where we try and distinguish a particular tissue sample from each of the other 10 tissue samples. For each class k, training is done on class k versus all the remaining 10 classes individually using the labels {-1, 1}. Class k is given the positive label while the other classes being considered are given the negative label. To train, 2/3 of the 255 samples are used and called $\boldsymbol{X}_{train}$ and, $\boldsymbol{y}_{train}$. The 2/3 training samples are picked by randomly selecting 2/3 of class k samples and 2/3 of all other remaining samples, ensuring that there are enough training and testing samples from class k. The selection process is random and guarantees that 2/3 of class k is always selected. The remaining 1/3 samples are the test samples and are called $\boldsymbol{X}_{test}$ and $\boldsymbol{y}_{test}$. For regression, the labels {-1, 1} create a decision boundary at the origin, where positive regression results will correspond to the positive class while negative regression results will correspond to the negative class.

3.4.1 Methodology

Using linear least squares (LLS) and ridge regression, we perform supervised classification following Algorithm 3.3, and the number of correctly classified tissue types are averaged over 300 iterations and reported in Table 3.4, Table 3.5 and Table 3.6. For ridge regression, the optimal penalty parameter γ^* is set by using leave-one-out cross validation (introduced in Appendix A section A.4.3) on the training set and choosing γ with the minimum mean squared error on the validation set. The weight vector is calculated following section 3.3.7.2.

Algorithm 3.3 Supervised classification methodology

1: Given: $\boldsymbol{X}_{train}, \boldsymbol{y}_{train}, \boldsymbol{X}_{test}, \boldsymbol{y}_{test}$
2: Minimize sum of squares on training data: $\boldsymbol{X}_{train}, \boldsymbol{y}_{train}$
3: If linear least squares regression: $\boldsymbol{w}^* = (\boldsymbol{X}_{train}{}^T \boldsymbol{X}_{train})^{-1} \boldsymbol{X}_{train}{}^T \boldsymbol{y}_{train}$
4: Do prediction on test data, $\boldsymbol{X}_{test}, \boldsymbol{y}_{test}$, using $\boldsymbol{w}^*$
5: $\hat{\boldsymbol{y}} = \boldsymbol{X}_{test} \boldsymbol{w}^* = (\boldsymbol{X}_{test}{}^T \boldsymbol{X}_{test})^{-1} \boldsymbol{X}_{test}{}^T \boldsymbol{y}_{test}$
6: Report average classification success rate over 300 iterations

* Correct classification if $\hat{\boldsymbol{y}} > 0$ and incorrect classification if $\hat{\boldsymbol{y}} < 0$
* Success rate is defined as the number of correctly classified tissue samples

3.4.2 Linear Least Squares Regression Classification Results

Covariance (%)	Precision (ρ=1) (%)	Precision (ρ=3) (%)
95.91	96.77	97.64

Table 3.3 LLS two-class average classification results for the precision and covariance classifiers. The average represents the percentage of correctly classified tissue samples for all 11 tissue types.

We present the overall average binary classification results for all tissue combinations in Table 3.3 for the covariance and precision models. From Table 3.3, it is evident that the two estimated precisions appear to perform slightly better than the sample covariance. The difference however is minute, and so we can conclude that the precisions perform at least as well as the covariance classifier. For a more detailed look, Table 3.4, Table 3.5 and Table 3.6 show each binary classification result for the covariance and estimated precisions.

	1	2	3	4	5	6	7	8	9	10	11
1		92.98	56.17	94.85	100.00	99.74	100.00	100.00	100.00	100.00	100.00
2	92.90		94.86	94.91	100.00	99.59	100.00	100.00	100.00	100.00	100.00
3	55.65	94.28		95.02	100.00	98.92	97.77	100.00	99.98	100.00	100.00
4	95.10	94.56	95.28		99.10	100.00	86.88	99.96	99.97	99.97	100.00
5	100.00	100.00	100.00	99.06		100.00	100.00	100.00	100.00	100.00	100.00
6	99.86	99.87	99.00	100.00	100.00		100.00	100.00	98.52	96.88	98.33
7	100.00	100.00	97.92	88.06	100.00	99.97		100.00	100.00	100.00	100.00
8	100.00	100.00	100.00	99.99	100.00	100.00	100.00		96.33	100.00	100.00
9	100.00	100.00	100.00	100.00	100.00	98.89	100.00	96.44		61.62	55.33
10	100.00	100.00	100.00	100.00	100.00	96.67	100.00	100.00	63.62		56.61
11	100.00	100.00	100.00	100.00	100.00	98.67	100.00	100.00	54.52	55.00	

Table 3.4 LLS two-class classification results for the covariance classifier. Each row represents the positive tissue class and each column represents the negative tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus each of the other 10 tissue types.

	1	2	3	4	5	6	7	8	9	10	11
1		94.22	83.78	91.93	99.29	98.17	98.36	100.00	99.18	98.69	98.64
2	94.41		98.04	99.71	99.99	99.31	98.79	100.00	99.72	99.55	99.12
3	83.77	98.26		96.69	99.32	99.79	98.52	99.56	99.82	99.60	99.31
4	91.99	99.88	97.07		99.77	98.00	96.35	100.00	99.95	99.78	99.84
5	99.18	100.00	99.45	99.81		99.83	99.63	99.80	100.00	99.46	99.35
6	97.88	99.18	99.54	98.06	99.88		92.53	96.77	87.41	90.88	87.50
7	98.52	99.23	98.52	96.81	99.75	94.30		90.83	93.78	91.54	91.08
8	99.90	99.97	99.71	100.00	99.78	97.37	90.73		88.81	96.13	95.33
9	99.51	99.50	99.76	99.94	99.98	87.85	94.11	91.30		91.95	91.38
10	98.64	99.03	99.52	99.78	99.57	89.75	93.67	97.71	93.43		89.89
11	98.14	99.36	99.05	99.81	99.56	88.38	92.58	96.21	89.95	89.28	

Table 3.5 LLS two-class classification results for the precision (ρ=1) classifier. Each row represents the positive tissue class and each column represents the negative tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus each of the other 10 tissue types.

	1	2	3	4	5	6	7	8	9	10	11
1		88.37	69.53	94.12	100.00	98.90	99.95	100.00	100.00	100.00	100.00
2	88.00		89.84	97.64	100.00	99.87	99.54	100.00	100.00	100.00	99.79
3	70.00	89.88		95.25	100.00	98.40	99.85	99.54	99.93	99.95	99.88
4	93.93	97.83	94.76		99.35	99.52	99.61	99.84	99.94	100.00	99.92
5	100.00	100.00	100.00	99.27		100.00	100.00	100.00	98.23	99.63	98.76
6	98.69	99.56	98.06	99.20	100.00		99.13	100.00	97.85	98.96	96.04
7	99.98	99.67	99.77	99.65	100.00	99.37		99.97	99.78	93.92	94.21
8	100.00	100.00	99.48	99.86	100.00	100.00	100.00		99.70	98.71	99.63
9	100.00	100.00	100.00	99.86	98.44	97.81	99.56	99.78		86.24	91.33
10	100.00	100.00	99.93	99.92	99.65	99.25	91.29	99.67	87.10		91.00
11	100.00	99.97	99.86	99.95	98.80	97.17	94.42	99.79	88.76	90.94	

Table 3.6 LLS two-class classification results for the precision (ρ=3) classifier. Each row represents the positive tissue class and each column represents the negative tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus each of the other 10 tissue types.

Table 3.4, 3.5 and 3.6 show that the covariance and precisions perform essentially equally. These individual results are consistent with the average performance in Table 3.3.

3.4.3 Ridge Regression Classification Results

We perform ridge regression classification and report the overall average performance in Table 3.7. The results show that the precisions perform slightly better than the covariance.

Covariance (%)	Precision (ρ=1) (%)	Precision (ρ=3) (%)
93.68	98.52	98.36

Table 3.7 LLS two-class average classification results for the precision and covariance classifiers. The average represents the percentage of correctly classified tissue samples for all 11 tissue types.

	1	2	3	4	5	6	7	8	9	10	11
1		88.20	53.75	95.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00
2	88.22		94.47	94.44	100.00	99.95	100.00	100.00	100.00	100.00	100.00
3	53.82	94.68		95.39	100.00	99.88	98.63	100.00	100.00	100.00	100.00
4	95.17	95.00	95.09		99.45	100.00	88.32	99.91	100.00	99.94	99.94
5	100.00	100.00	100.00	99.43		99.95	100.00	100.00	100.00	100.00	100.00
6	100.00	99.90	99.88	100.00	99.95		100.00	100.00	100.00	100.00	100.00
7	100.00	100.00	98.75	88.68	100.00	100.00		100.00	100.00	100.00	100.00
8	100.00	100.00	100.00	99.91	100.00	100.00	100.00		55.56	62.50	62.50
9	100.00	100.00	100.00	99.88	100.00	100.00	100.00	55.59		57.14	57.14
10	100.00	100.00	100.00	99.94	100.00	100.00	100.00	62.50	57.14		49.89
11	100.00	100.00	100.00	99.84	100.00	100.00	100.00	62.50	57.14	49.89	

Table 3.8 Ridge regression two-class classification results for the covariance classifier. Each row represents the positive tissue class and each column represents the negative tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus each of the other 10 tissue types.

	1	2	3	4	5	6	7	8	9	10	11
1		96.16	73.37	94.90	100.00	99.90	100.00	100.00	100.00	100.00	100.00
2	96.67		93.75	95.13	100.00	100.00	100.00	100.00	100.00	100.00	100.00
3	74.50	94.25		94.13	100.00	98.21	98.83	100.00	100.00	100.00	100.00
4	94.77	95.00	93.98		97.98	95.65	95.13	96.58	96.50	96.17	96.24
5	100.00	100.00	100.00	98.04		100.00	100.00	100.00	100.00	100.00	100.00
6	99.93	100.00	98.15	95.28	100.00		100.00	100.00	100.00	100.00	100.00
7	99.98	100.00	98.92	95.06	100.00	100.00		100.00	100.00	100.00	100.00
8	100.00	100.00	100.00	96.22	100.00	100.00	100.00		100.00	100.00	100.00
9	100.00	100.00	100.00	96.26	100.00	100.00	100.00	100.00		100.00	100.00
10	100.00	100.00	100.00	95.76	100.00	100.00	100.00	100.00	100.00		100.00
11	100.00	100.00	100.00	96.60	100.00	100.00	100.00	100.00	100.00	100.00	

Table 3.9 Ridge regression two-class classification results for the precision (ρ=1) classifier. Each row represents the positive tissue class and each column represents the negative tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus each of the other 10 tissue types.

	1	2	3	4	5	6	7	8	9	10	11
1		91.02	69.75	94.85	100.00	95.48	100.00	100.00	100.00	100.00	100.00
2	91.53		90.60	94.99	100.00	100.00	100.00	100.00	100.00	100.00	100.00
3	69.18	90.61		94.43	100.00	96.29	99.44	100.00	100.00	100.00	100.00
4	94.78	94.50	94.36		99.01	96.06	97.96	98.71	99.64	99.51	99.89
5	100.00	100.00	100.00	99.03		100.00	100.00	98.40	98.09	98.00	98.28
6	95.38	100.00	95.60	96.88	100.00		100.00	100.00	100.00	100.00	100.00
7	100.00	100.00	99.40	97.64	100.00	100.00		100.00	100.00	100.00	100.00
8	100.00	100.00	100.00	98.33	98.53	100.00	100.00		100.00	100.00	100.00
9	100.00	100.00	100.00	99.30	98.16	100.00	100.00	100.00		99.95	100.00
10	100.00	100.00	100.00	99.56	98.13	100.00	100.00	100.00	99.71		100.00
11	100.00	100.00	100.00	99.86	98.28	100.00	100.00	100.00	100.00	100.00	

Table 3.10 Ridge regression two-class classification results for the precision (ρ=3) classifier. Each row represents the positive tissue class and each column represents the negative tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus each of the other 10 tissue types.

Individual binary classification performance shown in Table 3.8, Table 3.9 and Table 3.10 show that the precision classifiers have more cases where they achieve perfect classification for particular binary tissue combinations compared to the covariance. For the cases where the covariance matrix performs worse than the precision classifiers, its performance appears to be significantly worse.

3.5 One-versus-all Classification to Identify Tissue Samples

For the one-versus-all classification, we follow the same methodology presented in section 3.4 of using the sub-networks in each of the different graphical models to calculate average expression levels across all sample to be used as features.

3.5.1 Methodology

We follow similar methodology as the binary classification except that the training and test sets are divided differently. For each class *k*, training is done on class *k* versus all the remaining 10 classes, using the labels {-1, 1}. Class *k* is given the positive label while all other classes are given the negative label. To train, 2/3 of the 255 samples are used and called $\boldsymbol{X}_{train}$ and, $\boldsymbol{y}_{train}$. The 2/3 training samples are picked by randomly selecting 2/3 of class *k* samples and 2/3 of all other remaining samples, ensuring that there are enough training and testing samples from class *k*. The selection process is random and guarantees that 2/3 of class *k* is always selected. The remaining 1/3 samples are the test samples and are called $\boldsymbol{X}_{test}$ and $\boldsymbol{y}_{test}$. The number of correctly classified tissue types are averaged over 300 trials and the results are reported in Table 3.11, Table 3.12, Table 3.13, and Table 3.14.

3.5.2 Linear Least Squares Regression Classification Results

For the one-versus-all classification problem, the average results for linear least squares classification shows that the precision classifiers do not perform better than the covariance classifier, as shown in Table 3.11. We look at individual tissue classification results and test to see if the mean classification performance of each precision classifier is different from the mean performance of the covariance classifier using hypothesis tests (*t*-tests) which are presented in Appendix E table E.19 and Table 3.12. Results are statistically significant at the 1% level when '**' is present and significant at the 5% level when '*' is present. For all hypothesis tests in this thesis, we would use the same '**' and '*' to show statistical significance levels.

Covariance (%)	Precision (ρ=1) (%)	Precision (ρ=3) (%)
64.37	61.59	62.10

Table 3.11 LLS one-versus-all average classification results for the precision and covariance classifiers. The average represents the percentage of correctly classified tissue samples for all 11 tissue types.

Class	Tissue	Covariance		Precision (ρ=1)		Precision (ρ=3)	
		Correct(%)	std	Correct(%)	std	Correct(%)	std
1	D4_MDDC	56.68	4.74	56.07	2.49	59.25	2.71
2	DC	58.35	4.19	60.78	3.11	58.93	2.57
3	MDDC	66.83	2.82	63.28	2.89**	61.19	2.48**
4	MDM	89.69	1.44	82.16	2.44**	83.73	2.16**
5	BLOOD	84.94	2.40	89.81	2.46	88.13	2.26
6	M3DC	48.65	5.74	55.63	2.78	55.99	2.71
7	AM	55.24	6.35	57.42	2.84	57.49	2.44
8	HC	69.20	3.50	60.35	3.12**	64.58	2.81*
9	HL	62.56	4.50	54.57	2.99*	54.21	3.01*
10	2T	58.02	4.84	48.20	2.89*	50.07	2.82*
11	HOS	57.91	4.72	49.20	2.89*	49.50	3.14*

Table 3.12 LLS one-versus-all classification results for the precision and covariance classifiers. Each row represents the positive tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus all other 10 tissue types.

From Table 3.12, it is evident that the performance of each classifier is a lot worse than the binary classification results which implies that it appears to be more difficult for each classifier to distinguish a particular tissue from all the other tissue samples considered at the same time, rather than one tissue distinguished from another tissue. Also, the results for the tissue classification for the precisions are statistically significant for 6 out of the 11 tissue types, and for each of these scenarios, the covariance always performs better than the precision classifiers. We can conclude

that the covariance is a better classifier than the precision classifiers in the one-versus-all cases when linear least square regression is used.

3.5.3 Ridge Regression Classification Results

The ridge regression average classification results in Table 3.13 show improved performance from the linear least squares classification results. This is as expected, and consistent with the results from the binary classification problem in section 3.4. This time around, the precisions appear to perform slightly better on average than the covariance classifier, and we test to see if this is true by performing hypothesis tests (*t*-tests) and present results in in Appendix E table E.20 and Table 3.14.

Covariance (%)	Precision (ρ=1) (%)	Precision (ρ=3) (%)
65.25	67.39	68.34

Table 3.13 Ridge regression one-versus-all average classification results for the precision and covariance classifiers. The average represents the percentage of correctly classified tissue samples for all 11 tissue types.

Like the linear least squares classifier, individual tissue classification performance is a lot worse than the binary classification performance, once again implying that it appears to be more difficult for each classifier to distinguish a particular tissue from all the other tissue samples considered at the same time, rather than one tissue distinguished from another tissue.

Results in Table 3.14 and Appendix E table E.20 show that 3 out of the 11 tissue for the precision (ρ=1) and for 2 out of the 11 tissue types for the precision (ρ=3)

classification problems are statistically significant, and for each of these scenarios, the covariance always performs better than the precision classifiers. For the other 8 and 9 tissue classification tasks for each precision respectively, we can say that the covariance and precision classifiers perform essentially the same. From these results, we can conclude that the covariance is a better classifier than the precision classifiers in the one-versus-all cases when ridge regression is used.

Class	Tissue	Covariance		Precision (ρ=1)		Precision (ρ=3)	
		Correct(%)	std	Correct(%)	std	Correct(%)	std
1	D4_MDDC	66.45	2.81	56.69	3.57**	62.70	2.80**
2	DC	57.47	4.02	61.82	2.69	61.87	2.74
3	MDDC	66.72	2.75	64.38	2.86*	66.29	2.36
4	MDM	89.83	1.54	84.01	2.01**	85.22	2.13**
5	BLOOD	84.70	2.20	94.98	1.92	89.83	2.24
6	M3DC	54.11	5.95	57.76	2.78	54.43	2.64
7	AM	49.19	5.76	59.58	3.51	55.59	2.60
8	HC	68.55	3.29	70.75	3.08	71.42	3.13
9	HL	63.37	4.10	62.92	4.58	68.23	4.42
10	2T	59.22	5.16	61.30	4.96	70.11	5.10
11	HOS	58.13	5.07	67.12	5.24	66.04	6.40

Table 3.14 Ridge regression one-versus-all classification results for the precision and covariance classifiers. Each row represents the positive tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus all other 10 tissue types.

3.6 Using Random Genes to Identify Tissue Samples

In an attempt to see if the estimated covariance and precisions perform gene selection in such a way that they are biologically meaningful, we replace the genes in the *k*-

nearest neighbour graphs in section 3.3.4 with randomly selected genes, while maintaining the topology of the graphs. The purpose of this is to see if by performing the same classification experiments as in section 3.4 and 3.5, we are able to predict tissue types more or less accurately. If gene selection is done in a biologically meaningful way, we will expect that classification performance will decrease when the optimally chosen genes are replaced with randomly selected genes for both the covariance and precision classifiers. Section 3.4 and section 3.5 show that the ridge regression classifiers perform better than the linear least squares regression classifiers, therefore, for this experiment, we only use ridge regression for classification. Results from the one-versus-all classification as well as the binary classification are presented in the subsequent tables. We compare classification results between the classifiers with randomly replaced genes, and the original classifiers from section 3.4 and section 3.5.

3.6.1 Two-class Classification to Identify Tissue Samples

Following section 3.4, we perform binary classification, where we try and distinguish a particular tissue sample from each of the other 10 tissue samples.

3.6.1.1 Ridge Regression Classification Results

Table 3.15 presents the average results of the binary classification problem using ridge regression, when we have random gene replacement. Table 3.16 shows the old classification results using ridge regression without the random gene replacement.

We expect the results with the random gene effect to be worse, compared to the past results in Table 3.16.

Covariance (%)	Precision (ρ=1) (%)	Precision (ρ=3) (%)
86.89	98.23	97.57

Table 3.15 Ridge regression two-class average classification results for the precision and covariance classifiers. The average represents the percentage of correctly classified tissue samples for all 11 tissue types when random genes are used to replace the genes selected from the precision and covariance models.

Covariance (%)	Precision (ρ=1) (%)	Precision (ρ=3) (%)
93.68	98.52	98.36

Table 3.16 Ridge regression two-class average classification results for the precision and covariance classifiers. The average represents the percentage of correctly classified tissue samples for all 11 tissue types using the genes selected from the precision and covariance models.

By comparing Table 3.15 and Table 3.16, it appears that the classifiers with the random gene replacement perform worse only for the covariance classifier. Individual binary classification results are presented in Appendix C. This implies that the gene selection process for the covariance classifier appears to be biologically meaningful, however, we do not have enough evidence to make the same claim for the precision classifiers. More detailed individual binary classification results can be found in Appendix C.

3.6.2 One-versus-all Classification to Identify Tissue Samples

Following section 3.5, we perform one-versus-all classification, where we try and distinguish a particular tissue sample from all the other 10 tissue samples.

3.6.2.1 Ridge Regression Classification Results

Table 3.17 presents the average results of the one-versus-all classification problem using ridge regression, with the random gene replacement. Table 3.18 shows the old classification results using ridge regression. We expect the results with the random gene replacement to be worse when compared to the past results in Table 3.18.

Covariance (%)	Precision (ρ=1) (%)	Precision (ρ=3) (%)
58.38	67.34	67.82

Table 3.17 Ridge regression one-versus-all average classification results for the precision and covariance classifiers. The average represents the percentage of correctly classified tissue samples for all 11 tissue types when random genes are used to replace the genes selected from the precision and covariance models.

Covariance (%)	Precision (ρ=1) (%)	Precision (ρ=3) (%)
65.25	67.39	68.34

Table 3.18 Ridge regression one-versus-all average classification results for the precision and covariance classifiers. The average represents the percentage of correctly classified tissue samples for all 11 tissue types using the genes selected from the precision and covariance models.

By comparing Table 3.17 and Table 3.18, we see the same performance pattern as the binary classification problem and it is evident that once again that only the covariance

classifier with the random gene replacement appears to perform worse than the old covariance classifier without random gene replacements. These results support our earlier conclusion that the covariance appears to be the only model with biologically meaningful gene selection in the *k*-nearest neighbour graphs. We perform hypothesis tests (*t*-tests) to verify this and present the results in in Appendix E table E.21, Table E.22, Table E.23 and report individual tissue classification results in Table 3.19, Table 3.20 and Table 3.21.

From Table 3.19, the covariance performance with random gene replacement performs statistically significantly worse in 6 out of the 11 tissue classification problems. For the precision classification performance in Table 3.20 and Table 3.21, we only see statistically significantly better performance for the precision (ρ=3) classifier and for only 1 out 11 tissue classification problems. From these results, we can conclude that the covariance appears to be the only model performing gene selection in a biologically meaningful way.

Class	Tissue	Covariance without Random Effect		Covariance with Random Effect	
		% correct	std dev	% correct	std dev
1	D4_MDDC	66.45	2.81	53.88	2.59**
2	DC	57.47	4.02	53.93	3.17*
3	MDDC	66.72	2.75	60.36	2.45**
4	MDM	89.83	1.54	66.49	2.37**
5	BLOOD	84.70	2.20	72.88	2.40**
6	M3DC	54.11	5.95	53.89	2.67
7	AM	49.19	5.76	50.75	2.69
8	HC	68.55	3.29	49.90	2.55**
9	HL	63.37	4.10	63.20	2.82
10	2T	59.22	5.16	58.75	2.92
11	HOS	58.13	5.07	58.20	2.83

Table 3.19 Ridge regression one-versus-all classification results for the covariance classifier with and without random gene replacement. Each row represents the positive tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus all other 10 tissue types.

Class	Tissue	Precision (ρ=1) without Random Effect		Precision (ρ=1) with Random Effect	
		% correct	std dev	% correct	std dev
1	D4_MDDC	56.69	3.57	56.72	2.96
2	DC	61.82	2.69	62.09	3.09
3	MDDC	64.38	2.86	64.42	2.61
4	MDM	84.01	2.01	84.13	1.98
5	BLOOD	94.98	1.92	94.24	2.10
6	M3DC	57.76	2.78	58.85	3.09
7	AM	59.58	3.51	58.83	3.17
8	HC	70.75	3.08	69.66	3.57
9	HL	62.92	4.58	62.51	4.33
10	2T	61.30	4.96	62.39	4.31
11	HOS	67.12	5.24	66.93	3.93

Table 3.20 Ridge regression one-versus-all classification results for the precision (ρ=1) classifier with and without random gene replacement. Each row represents the positive tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus all other 10 tissue types.

Class	Tissue	Precision (ρ=3) without Random Effect		Precision (ρ=3) with Random Effect	
		% correct	std dev	% correct	std dev
1	D4_MDDC	62.70	2.80	62.75	2.78
2	DC	61.87	2.74	62.97	2.73
3	MDDC	66.29	2.36	67.00	2.46**
4	MDM	85.22	2.13	84.12	2.07**
5	BLOOD	89.83	2.24	88.26	2.68
6	M3DC	54.43	2.64	55.01	2.64
7	AM	55.59	2.60	58.49	3.01
8	HC	71.42	3.13	69.73	3.13**
9	HL	68.23	4.42	64.24	4.71*
10	2T	70.11	5.10	68.26	4.88*
11	HOS	66.04	6.40	65.23	4.90*

Table 3.21 Ridge regression one-versus-all classification results for the precision (ρ=3) classifier with and without random gene replacement. Each row represents the positive tissue class. The results reported are the percentage of correctly classified tissue samples of each row tissue type versus all other 10 tissue types.

3.7 Summary

In this chapter, we applied Graphical Lasso to a gene microarray dataset in order to classify tissue samples. Graphical Lasso was used to generate graphical Gaussian models, which we used to reduce the dimensionality of the problem by selecting features from the graphs which corresponded to the average expression levels of each sub-network in a particular graphical model. We performed binary as well as one-versus-all classification using both linear least squares regression and ridge regression.

In general, the binary classification results showed improvement in classifying tissue samples correctly compared to the one-versus-all classification for both the covariance and precision classifiers. Biologically, samples from the same tissue tend

to have similar expression levels, so it makes sense that the binary classification would lead to a reduction in noise, and hence, better performance. For the binary classification, the precision classifiers performed at least as good as the covariance classifier when linear least squares regression was used while the precision classifiers performed slightly better than the covariance classifier when ridge regression was used. For the one-versus-all classification the covariance always performed better than the precision classifiers.

When random genes were used to replace the optimally selected genes in the *k*-nearest neighbour graphs and used to perform the same classification tasks, on average, for the covariance, these new classifiers performed significantly worse, while for the precisions, the performance was essentially the same. These results showed that the covariance appeared to be the only model performing gene selection in a biologically meaningful way. For the precision classifiers, we can conclude that given any number of random genes and gene expression levels, this information is enough to correctly classify tissue types of randomly picked samples.

In conclusion, we expected Graphical Lasso to produce *k*-nearest neighbour graphs that were biologically meaningful, however this was not the case. Classification performance showed that only the covariance appeared to produce *k*-nearest neighbour graphs that were biologically meaningful.

Part II: Graphical Lasso Application to Finance

Chapter 4

Graphical Lasso and Portfolio Optimization

This chapter provides the background knowledge on Markowitz mean-variance portfolio optimization for this thesis. It starts with a formal definition of the Markowitz mean-variance (MV) model. It illustrates the estimation risk of the MV model. Additionally, we discuss several current methods used to address the estimation risk of the covariance of asset returns. Finally, the use of Graphical Lasso for covariance estimation in the mean-variance framework is illustrated using artificially generated stock market data.

4.1 Markowitz Mean-Variance Model

The traditional objective of active portfolio management is to consistently deliver excess return against a benchmark index with a given amount of risk. The benchmark in question could be one of the traditional market indices, such as the Standard & Poor's (S&P) 500 Index and the Russell 2000 Index, or a cash return such as Treasury bill rate [48]. To be successful, one must rely on four key components in the investment process. First is the alpha model, which predicts the relative returns of stocks within a specified investment. The second component is a risk model that estimates the risks of individual stocks and the return correlations among different stocks. The third piece is a portfolio construction methodology to combine both

return forecasts and risk forecasts to form an optimal portfolio. Lastly, one must have the portfolio implementation process to execute all trades [48]. Return and risk are two inherent characteristics of any investment. The return of an uncertain investment is best described by a probability distribution. For stocks, the normal (Gaussian) distribution or lognormal distribution are normally used to model asset returns.

The classical Markowitz portfolio framework [2], defines *portfolio risk* as the variance of the portfolio return, and seeks a portfolio weight vector $\boldsymbol{w}$, with the highest expected return. Similarly, for a given level of expected return, $\mu_{P,}$ a rational investor would choose the portfolio with the lowest risk. In other words, given a target value μ_P for the mean return of a portfolio, Markowitz characterizes an efficient portfolio (MV efficient), if there is no portfolio having the same expected return with a lower risk.

4.2 Efficient Frontier

The efficient frontier is the curve that shows all efficient portfolios in a risk-return framework. An efficient frontier is defined as the portfolio that maximizes the expected return for a given amount of risk (variance of the portfolio return), or the portfolio that minimizes the risk subject to a given expected return. An investor will always invest in an efficient portfolio. If he desires a certain a certain amount of risk, the most logical thing to do would be to aim for the highest possible expected return.

If on the other hand, he wants a specific expected return, he would want to achieve this with the minimum possible amount of risk.

We make the following definitions:

N= number of assets.

C_0= capital to be invested.

C_{end}= capital at the end of the investment period.

R_P= total portfolio return.

μ_P= expected return of the portfolio.

σ_P^2= variance of the portfolio return.

r_i= rate of return on asset i.

μ_i= expected rate of return on asset i.

σ_{ij}= covariance between the returns of asset i and j.

σ_{ii}= variance on the return of asset i.

$\mathbf{\Sigma} = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1N} \\ \vdots & \ddots & \vdots \\ \sigma_{N1} & \cdots & \sigma_{NN} \end{bmatrix}$ = matrix of covariances of r.

w_i= amount invested in asset i.

4.2.1 Mathematical Notations

Suppose an investor desires to invest in a portfolio that contains N assets. Let $\mu \in R^N$ be the mean vector with μ_i as the mean return of asset $i, 1 \leq i \leq N$, and $w \in R^N$ be the decision vector with w_i as the weight of holding in the i^{th} asset.

The *portfolio expected return* μ_P is the weighted average of individual asset returns given by

$$\mu_P = \sum_{i=1}^{N} \mu_i w_i = \boldsymbol{\mu}^T \boldsymbol{w} \tag{4.2}$$

The variance and covariance of individual assets are characterized by a N-by-N positive semi-definite matrix $\boldsymbol{\Sigma}$, such that

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1N} \\ \vdots & \ddots & \vdots \\ \sigma_{N1} & \cdots & \sigma_{NN} \end{bmatrix}, \tag{4.3}$$

where σ_{ii} is the variance on the return of asset i, and σ_{ij} is the covariance between the returns of asset i and j.

Therefore, the *variance of portfolio return*, σ_P^2 , can be calculated by

$$\sigma_P^2 = \sum_{i=1}^{N} \sum_{j=1}^{N} w_i w_j \sigma_{ij} = \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w} \tag{4.4}$$

To calculate the efficient frontier we have to minimize the risk given some expected return. The objective function is the function that has to be minimized, which is the variance of the portfolio return given by

$$var(C_{end}) = var(C_0 + R_P) = var(R_P) = var(\boldsymbol{r}^T \boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w} \tag{4.5}$$

where C_{end} is the capital at the end of the investment period, C_0 is the capital that can be invested, R_P is the total portfolio return, $\boldsymbol{r}^T$ is the vector of the rate of return on all assets, $\boldsymbol{w}$ is the vector of portfolio weights and $\boldsymbol{\Sigma}$ is the covariance of asset returns.

There are two constraints that must hold for minimizing this objective function. First, the expected return must be fixed, because we are minimizing risk given this return. This fixed portfolio mean is defined by μ_P. The second constraint is that we can only invest the capital we have at this moment, so the amounts we invest in each single asset must add up to this amount C_0. This gives the following two constraints:

$$\boldsymbol{\mu}^T \boldsymbol{w} = \mu_P \qquad \text{and} \qquad \overline{\mathbf{1}}^T \boldsymbol{w} = C_0$$

where $\overline{\mathbf{1}}$ is an N-dimensional vector of ones.

We are looking for the investment with minimum variance, so we have to solve the following problem [49]:

$$\min_{\boldsymbol{w}} \quad \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w} \tag{4.6}$$

$$\text{s.t.} \quad \boldsymbol{A}^T \boldsymbol{w} = B$$

with

$$\boldsymbol{A} = (\boldsymbol{\mu} \quad \overline{\mathbf{1}}) \quad \text{and} \quad \boldsymbol{B} = \begin{pmatrix} \mu_P \\ C_0 \end{pmatrix}$$

We use Lagrange method to solve this system. We get the following conditions, where λ_0 is the Lagrange multiplier:

$$\begin{cases} 2\boldsymbol{\Sigma} \boldsymbol{w} + \boldsymbol{A} \lambda_0 = 0 \\ \boldsymbol{A}^T \boldsymbol{w} = \boldsymbol{B} \end{cases} \qquad \text{with} \qquad \lambda_0 = \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \tag{4.7}$$

Solving the first equation (4.7) for $\boldsymbol{w}$ gives, with a redefinition of the vector $\lambda = -1/2\lambda_0$

$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1} \boldsymbol{A} \lambda$$

So the second equation of (4.7) becomes

$$\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A}\lambda = \boldsymbol{B} \qquad \Rightarrow \qquad \lambda = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^{-1}\boldsymbol{B} = \boldsymbol{H}^{-1}\boldsymbol{B}$$

where $\boldsymbol{H} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})$ and $\boldsymbol{H}^T = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})^T = \boldsymbol{A}^T(\boldsymbol{\Sigma}^{-1})^T\boldsymbol{A} = \boldsymbol{H}$, so $\boldsymbol{H}$ is a symmetric (2×2)-matrix. Filling in these expressions in the variance formula, we get

$$var(R_P) = \ \boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w} = \ \boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\boldsymbol{A}\lambda = \ \boldsymbol{w}^T\boldsymbol{A}\lambda = (\boldsymbol{A}^T)^T\boldsymbol{H}^{-1}\boldsymbol{B} = \boldsymbol{B}^T\boldsymbol{H}^{-1}\boldsymbol{B}$$

We have seen that $\boldsymbol{H}$ is a symmetric (2×2)-matrix, so suppose that

$$\boldsymbol{H} = \begin{pmatrix} a & c \\ b & d \end{pmatrix} \quad \Rightarrow \quad \boldsymbol{H}^{-1} = \frac{1}{ac - b^2}\begin{pmatrix} c & -b \\ -b & a \end{pmatrix}$$

Define $d = \det(\boldsymbol{H}) = ac - b^2$. Because $\boldsymbol{H} = (\boldsymbol{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{A})$ it is easy to see that:

$a = \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$,

$b = \boldsymbol{\mu}^T\boldsymbol{\Sigma}^{-1}\overline{\mathbf{1}} = \overline{\mathbf{1}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$,

$c = \overline{\mathbf{1}}\Sigma^{-1}\overline{\mathbf{1}}$,

$d = ac - b^2$.

We will show that the parameters a, c and d are positive: Because we have assumed that the covariance matrix $\boldsymbol{\Sigma}$ is positive definite, the inverse matrix $\boldsymbol{\Sigma}^{-1}$ is also positive definite [49]. This means that $\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} > 0$ for all nonzero $(N \times 1)$-vectors $\boldsymbol{x}$, so it is clear that

$$a > 0, \qquad c > 0$$

But also $(b\boldsymbol{\mu} - a\overline{\mathbf{1}})^T\boldsymbol{\Sigma}^{-1}(b\boldsymbol{\mu} - a\overline{\mathbf{1}}) = bba - abb - abb + aac = a(ac - b^2) = ad > 0$, and because $a > 0$ we know that

$$d > 0$$

With the definition of $\boldsymbol{H}$ our expression for the variance becomes

$$var(R_P) = \frac{1}{d}(\mu_P \quad C_0)\begin{pmatrix} c & -b \\ -b & a \end{pmatrix}\begin{pmatrix} \mu_P \\ C_0 \end{pmatrix}$$

$$= \frac{1}{d}(c\mu_P^2 - 2bC_0\mu_P + aC_0^2)$$

This gives the expression for the efficient frontier in a risk-return framework [49]. Note that the upper half of the efficient frontier in Figure 4.1 is the efficient set, because portfolios at the lower half can be chosen on the upper half so more return is obtained with the same level of risk. The formula for the efficient frontier is given by [49].

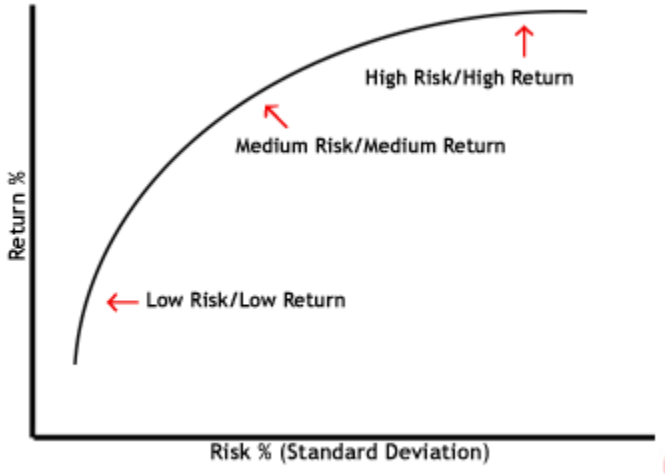

Figure 4.1 The efficient frontier

4.3 Linear Constraints

In real investment practice, in order to capture real world restrictions on investments, there are several constraints that need to be considered [48]. One may

want to ensure the portfolio turnover is at a certain specified level. Another constraint is the holding constraint for stocks, which has several variations. For example, one may require that any individual stock holding in a portfolio be no more than a certain percentage of the portfolio. In terms of active weights, one may require that any individual active weight be less than a certain percentage. These constraints are aimed at controlling the specific risk of individual holdings and limiting the damage that the poor performance of any single stock can inflict on the total portfolio [48]. There may need to be other constraints such as transaction costs and trading size limits on certain assets.

Markowitz optimization typically gives both positive and negative portfolio weights and, especially for large portfolios, it usually gives large negative weights for a certain number of assets [48]. A negative weight corresponds to a short selling position (selling an asset without owning it) and it is sometimes difficult to implement in practice or forbidden. For this reason it is common practice to impose constraints to the portfolio weights in the optimization procedure. When one adds constraints on the range of variation of the $w_i s$, the optimization problem cannot be solved analytically, and quadratic programming must be used. Quadratic programming algorithms are implemented in most numerical programs, such as Matlab or R [50]. In the following experiments, we will consider the portfolio optimization problem both with and without the no short selling constraint. The second constraint we consider is ensuring that all money available for the investment is allocated [48]. Both constraints are mathematically defined as

Short sales restriction:

$$w_i \geq 0 \quad \forall\, i$$

Using all available money in investment:

$$\sum_{i=1}^{N} w_i = 1$$

4.4 Global Minimum-Variance Portfolio Optimization

It has been discussed in chapter 1 that expected stock returns are hard to estimate, and lead to estimation errors that result in suboptimal portfolio performance. Several papers [57,59] suggest avoiding the estimation of expected returns and instead, assume that all stocks have equal expected returns. Under this assumption, all stock portfolios differ only with respect to their risk. Therefore, the only efficient stock portfolio is the one with the smallest risk i.e. the global minimum variance portfolio. The composition of the global minimum variance portfolio (GMVP) depends only on the covariance matrix of stock returns. Since the covariance matrix can be estimated much more precisely than the expected returns, the estimation risk of the investor is expected to be reduced. The global minimum variance portfolio is the stock portfolio with the lowest return variance for a given covariance matrix $\mathbf{\Sigma}$.

Suppose an investor desires to invest in a portfolio with the least amount of risk. He doesn't care about his expected return, he only wants to invest all his money with the lowest possible amount of risk. Because he will always invest in an efficient portfolio,

he will choose the portfolio on the efficient frontier with minimum standard deviation. At this point, also the variance is minimal. That is why this portfolio called the global minimum variance portfolio. The global minimum variance portfolio can be calculated by minimizing the variance subject to the necessary constraint that an investor can only invest the amount of capital he has, also known as the *budget constraint*. The minimization is [49]

$$\min_{w} \quad \boldsymbol{w}^T \boldsymbol{\Sigma} \boldsymbol{w} \qquad (4.8)$$

$$\text{s.t.} \quad \overline{\mathbf{1}}^T \boldsymbol{w} = C_0$$

Using Lagrange to solve this set, we get

$$\begin{cases} 2\boldsymbol{\Sigma}\boldsymbol{w} + \overline{\mathbf{1}}\lambda_0 = 0 \\ \overline{\mathbf{1}}^T \boldsymbol{w} = C_0 \end{cases} \qquad \text{with} \qquad \lambda_0 \text{ a constant} \qquad (4.9)$$

Solving the first equation (4.9) for $\boldsymbol{w}$ gives, with a new constant $\lambda = -1/2\lambda_0$

$$\boldsymbol{w} = \boldsymbol{\Sigma}^{-1}\overline{\mathbf{1}}\lambda$$

Using this expression for w in the second equation of (4.9) gives

$$\overline{\mathbf{1}}^T \boldsymbol{\Sigma}^{-1}\overline{\mathbf{1}}\lambda = C_0 \qquad \Rightarrow \qquad \lambda = \frac{C_0}{\overline{\mathbf{1}}^T \boldsymbol{\Sigma}^{-1}\overline{\mathbf{1}}} = \frac{C_0}{c}$$

where $c = \overline{\mathbf{1}}^T \boldsymbol{\Sigma}^{-1}\overline{\mathbf{1}}$ is defined as the element h_{22} in the matrix H in the previous section. Filling in this expression for λ in the above expression for w gives

$$\boldsymbol{w}_{GMVP} = \boldsymbol{\Sigma}^{-1}\overline{\mathbf{1}} \, \frac{C_0}{c} \qquad (4.10)$$

(4.10) is the portfolio allocation when an investor desires minimum risk. Clearly, the application of the GMVP does not require the estimation of the problematic mean return and therefore it is less sensitive to estimation risk than the traditional mean variance portfolio by Markowitz. The GMVP is appealing since its weights are merely determined by the inverse covariance matrix and not the means. Given that the estimation errors in the means are typically large [54], the GMVP is an attractive alternative to the Markowitz mean-variance portfolio. It is important to note that the choice of using the GMVP is not a limiting one, as this portfolio is characterized by an out-of-sample Sharpe ratio (the ratio between the portfolio return and it standard deviation, a key portfolio performance measure) which is as good as that of other efficient portfolios [55,56,57].

4.5 Limitations of the Markowitz Approach

The Markowitz model presents several reasons for its impracticality. The first is the sheer number of necessary input parameters. We need an estimate of the expected return and the risk of each asset, plus estimates of the correlation between each pair of securities. This is a total of $2N + N \times (N - 1)/2$ values. All of these values cannot be known exactly and they change over time. [48].

The parameters of the MV model are the asset mean returns and the covariance matrix of returns. The true values of these parameters are unknown in practice and investors traditionally estimate them using historical data. Generally, a sample of T historical returns on risky assets, i.e., a set of observations $J_T = \{R_1, \dots, R_T\}$ is

available for estimation. The traditional practice employs the Maximum Likelihood (ML) estimators under the assumption that $T > N$ in order to ensure the non-singularity of the sample covariance matrix [58]:

$$\hat{\mu}_P = \frac{1}{T}\sum_{t=1}^{T} R_T \,, \tag{4.11}$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{T}\sum_{t=1}^{T}(R_T - \hat{\mu})(R_T - \hat{\mu}\,)^T \,, \tag{4.12}$$

Due to the estimation error introduced in the estimation process, the estimated parameters can have large errors. Therefore, the resulting portfolio weights fluctuate substantially and out-of-sample performance of these portfolios can be quite poor [58].

4.6 Synthetic Data Experiment

As pointed out in chapter 2, the Graphical Lasso is a method of sparse inverse covariance estimation, which offers the same benefits that Lasso does for linear regression. This variable reduction capability is particular important in portfolio optimization, where investors prefer to invest in a smaller basket of stocks. The present work in this chapter deals with parameter uncertainty by applying the Graphical Lasso methodology for the estimation of the inverse covariance matrix that aims to improve portfolio performance.

In order to evaluate the proposed sparse precision approach, we apply Graphical Lasso to artificially created stock financial daily return time series. In the experiments, we estimate covariance matrices of stock returns and use the

covariance estimates for portfolio optimization. The realized risks, empirical risks, realized likelihoods and empirical likelihoods are compared for the different covariance estimates and would be defined in section 4.6.4.

4.6.1 Generating Synthetic Data

To simulate synthetic data, we follow the properties of the dataset used in [50] and similarly, chapter 5, that consists of the daily returns of N = 90 highly capitalized stocks traded at NYSE and included in the NYSE US 100 Index. For these stocks, the closing prices are available in the eleven year period from 1 January 1997 to 31 December 2007, which is equivalent to 2761 trading days.

We generate the artificial stock data from a true inverse covariance solution from the NYSE stock data. This inverse covariance $\mathbf{\Sigma}^{-1*}$ is generated by running Graphical Lasso on 2 years' worth of randomly selected stock returns at a regularization of $\rho = 9.9 \times 10^{-5}$, producing a Graphical Lasso inverse covariance matrix solution with a sparsity level of 75.10% (24.90% density). Details of the stock data can be found in chapter 5 and [50]. Following the pattern of the stock market data used in chapter 5 which has stock returns of $N = 90$ NYSE stocks over $p = 2761$ trading days, we simulate independent multivariate Gaussian samples from the distribution $\mathbf{\Sigma}^{-1*}$. This data generation technique is described in Algorithm 4.1.

Algorithm 4.1 Synthetic data generation from a known inverse covariance

1: Given: N=2761, p=90, $\widehat{\boldsymbol{\Sigma}}^{-1} = \boldsymbol{\Sigma}^{-1*}$
2: $\widehat{\boldsymbol{\Sigma}} = (\boldsymbol{\Sigma}^{-1*})^{-1}$
3: Generate random samples from a standard Gaussian using Matlab function
4: '@randn'
5: $\mathbf{X}_1$ = randn (N, p)
6: Multiply the samples by a square root of the covariance matrix using '@chol'
7: $\mathbf{X}_2$ = X_1*chol($\widehat{\boldsymbol{\Sigma}}$)
8: The final data $\mathbf{X}_2$ is an $N \times p$ matrix drawn from the covariance $\widehat{\boldsymbol{\Sigma}} = (\boldsymbol{\Sigma}^{-1*})^{-1}$

4.6.2 Methodology

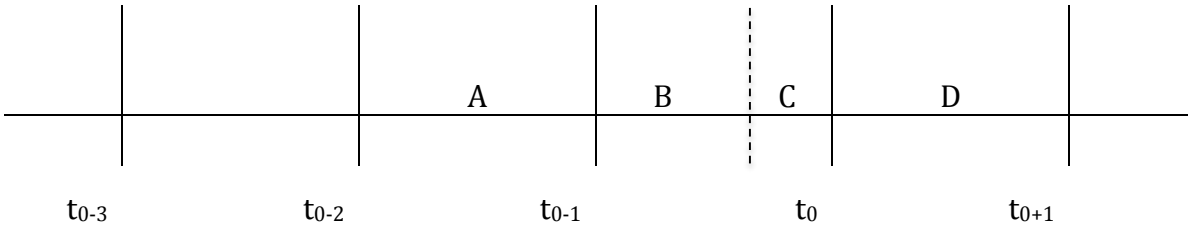

Figure 4.2 Portfolio Rebalancing Periods.

Let $t_{0-3}, t_{0-2}, .., t_{0+1}$ be portfolio rebalancing times and let A, B+C and D be the time periods between each consecutive rebalancing period illustrated in Figure 4.2. We assume that period A, B + C and D are of equal size, each consisting of T days. Period B consists of 80% of consecutive stock data between t_{0-1} and t_0, which is equivalent to 80% of T days, while period C consists of 20% of consecutive stock data between t_{0-1} and t_0, equivalent to 20% of T days. We call the T days preceding t_0 (period B+C) the *in-sample training period* and the T days after t_0 (period D) the *out-of-sample test period*. For example, if we assume that we are at time t_0, at this point in time, the

portfolio is selected by choosing the optimal weights that solve the global minimum variance optimization problem with and without short selling constraints, using the estimated covariance, $\boldsymbol{S}^{(EST)}$, from the in-sample training period (B + C) in figure (4.2). Specifically, we solve these two optimizations:

$$\min_{w} \quad \boldsymbol{w}^T \boldsymbol{S}^{(EST)} \boldsymbol{w} \tag{4.13}$$

$$\text{s.t: } (1) \sum_i w_i = 1$$

$$(2)\ w_i \geq 0 \quad \forall\, i$$

$$\min_{w} \quad \boldsymbol{w}^T \boldsymbol{S}^{(EST)} \boldsymbol{w} \tag{4.14}$$

$$\text{s.t: } (1) \sum_i w_i = 1$$

Problem (4.13) is the global minimum variance optimization problem with short selling constraints which gives rise to *long-only portfolios*. Problem (4.14) is the global minimum variance problem without any restrictions on short selling, which gives rise to *long-short portfolios*.

The input to the optimization problem is the estimated covariance matrix $\boldsymbol{S}^{(EST)}$ calculated using the in-sample training period and obtained using one of the portfolio strategies that will be described in section 4.6.3. We call $\boldsymbol{S}^{EMP}$ the empirical covariance matrix. In accordance with [50], we focus on the global minimum variance portfolio, since we are only interested in comparing the performance of different covariance estimators in portfolio selection and ignore estimation errors of asset

returns. The output of the global minimum optimization problem with the appropriate constraints is the optimal weight vector

$$\boldsymbol{w}^{(EST)} = arg \min_{\boldsymbol{w}} \boldsymbol{w}^T \boldsymbol{S}^{(EST)} \boldsymbol{w} \tag{4.15}$$

The time window T is varied on a wide range. In our empirical study, we use seven different time windows T of 1, 2, 3, 6, 9, 12, and 24 months. There are 20 trading days in a month, therefore we select the portfolio monthly (T = 20), bimonthly (T = 40), quarterly (T = 60), every six-months (T = 125), every nine-months (T = 187), yearly (T = 250), and biannually (T = 500). We use a rolling window style whereby the previous out-of-sample test period becomes the current in-sample training period and so forth. Since the total number of trading days is 2761, we consider 131, 65, 43, 13, 21, 10, and 8 portfolio optimizations for the time horizon T equal to 1, 2, 3, 6, 9, 12, and 24 months, respectively (for the 24 months case, in order to improve the statistics, following [50], we repeat the optimization process starting from 1 January 1998).

To evaluate portfolio performance, we estimate the covariance matrix in the in-sample training period using all the different portfolio strategies that will be discussed in section 4.6.3 to calculate the optimal portfolio weights and evaluate them in the future out of sample testing period. We repeat this optimization at each rebalancing period using a rolling window style and report the average annualized performance using the annualization factors found in Appendix G Table G.1.

4.6.3 Graphical Lasso Portfolio Strategies

Recall that the L_1 log-likelihood equation that the Graphical Lasso maximizes is defined by

$$\arg\max_{\boldsymbol{X}} \quad log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS}) - \rho\|\boldsymbol{X}\|_1, \tag{4.16}$$

where the optimal value of the penalty term ρ must somehow be approximated. We present validation methods for selecting the penalty parameter ρ to obtain estimates of the inverse covariance matrix that are optimal under certain portfolio performance criteria.

The Graphical Lasso strategies are:

i. GLasso$^{\text{MIN REALIZED RISK}}$
ii. GLasso$^{\text{MAX LIKELIHOOD-2}}$
iii. GLasso$^{\text{MAX LIKELIHOOD}}$
iv. GLasso$^{\text{ORACLE}}$

We refer to Figure 4.2 for the following portfolio strategy descriptions:

GLasso$^{\text{MIN REALIZED RISK}}$

We obtain different covariance estimates for period B data using Graphical Lasso with different regularizations, ρ. The optimal regularization, ρ^*, is selected by optimizing for certain portfolio criteria in period C (e.g. GLasso$^{\text{MIN REALIZED RISK}}$ selects the optimal regularization that maximizes the portfolio realized risk in period C). The sparsity

pattern for this optimal regularization is chosen and the covariance is re-estimated using period B data again to select the optimal covariance $\boldsymbol{S}^{(EST)}$. This method calculates Predicted risk and Empirical risk on period B data. Realized risk is calculated on period D. These portfolio measures are covered in section 4.6.4.

GLasso$^{\text{MAX LIKELIHOOD-2}}$

We obtain different covariance estimates for period B and C data using Graphical Lasso with different regularizations, ρ. The regularization with the highest likelihood in periods B + C is chosen as the optimal regularization, ρ^*. This method calculates predicted risk on periods B and C. Empirical risk is calculated on period B (for equality to other methods). Realized risk is calculated on period D.

GLasso$^{\text{MAX LIKELIHOOD}}$

We obtain different covariance estimates for period B data using Graphical Lasso with different regularizations, ρ. The regularization with the highest likelihood in period C is chosen as the optimal regularization, ρ^*. This method calculates predicted risk on periods B. Empirical risk is also calculated on period B (for equality to other methods). Realized risk is calculated on period D.

GLasso$^{\text{ORACLE}}$

This method estimates the optimal precision in period B based on sparsity level equality to the true precision. Using the estimated precision, we obtain an estimate for the optimal covariance matrix $\boldsymbol{S}^{(EST)}$. This method calculates Predicted risk and

Empirical risk on period B. Realized risk is calculated on period D. Note that period C is ignored

4.6.4 Performance Measures

To evaluate the performance of the different portfolio strategies, we compare the portfolio risks (predicted, realized and empirical), likelihoods (empirical and realized) sparsity levels and sparsity structure.

4.6.4.1 Predicted Risk

The predicted risk is calculated using $\boldsymbol{S}^{(EST)}$, which is estimated using the training period (time period T before the rebalancing point). The predicted risk can be seen as the learned quality of the estimated covariance on period B (period B+C for GLasso$^{\text{MAX LIKELIHOOD-2}}$) for the chosen regularization and is defined as

$$s_{Predicted} = \sqrt{\boldsymbol{w}^{(EST)}\boldsymbol{S}^{(EST)}\,\boldsymbol{w}^{(EST)}} \tag{4.17}$$

4.6.4.2 Realized Risk

The realized risk is calculated using the empirical covariance $\boldsymbol{S}^{EMP}$ which is estimated using the test period (time period T after the rebalancing point). A portfolio is said to be less risky than another when its realized risk is smaller. Based on the portfolio strategies defined in section 4.6.3, the realized risk is calculated on period D.

$$s_{Realized} = \sqrt{\boldsymbol{w}^{(EST)}\boldsymbol{S}^{EMP}\,\boldsymbol{w}^{(EST)}} \tag{4.18}$$

4.6.4.3 Empirical Risk

The empirical risk is calculated using the empirical covariance $\boldsymbol{S}^{EMP}$ which is estimated in the training period (time period T before the rebalancing point).

$$s_{Empirical} = \sqrt{\boldsymbol{w}^{(EST)} \boldsymbol{S}^{EMP} \boldsymbol{w}^{(EST)}} \tag{4.19}$$

The empirical risk for our portfolio strategies is calculated on period B. This is the actual performance on the empirical data on period B and can be seen as a regularized version of the predicted risk. We will expect that this is lower than the realized risk, which is into the future.

4.6.4.4 Realized Likelihood

Recall that the Gaussian log likelihood for a single observation x is

$$\ln(L) = -\frac{1}{2}\ln(|\sum|) - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \sum^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) - \frac{k}{2}\ln(2\pi), \tag{4.20}$$

where k is the number of variables and $\sum = \boldsymbol{S}^{(EST)*}$

The realized likelihood is the sum of the log likelihoods of all the samples in the test period D.

4.6.4.5 Empirical Likelihood

Following the definition of the Gaussian log likelihood of a single observation $\boldsymbol{x}$ given in (4.20), the empirical likelihood is the sum of the log likelihoods of all the samples in the training period B.

4.6.4.6 Sparsity Level and Zero-overlap

The sparsity level is the percentage of zero entries in the estimated inverse covariance. The zero-overlap on the other hand is the accuracy of the estimated inverse covariance in terms of matching the sparsity structure (correct zero pattern) of the true inverse covariance. In this experiment, we look at the average sparsity level and zero-overlap of the estimated inverse covariance for each portfolio strategy to see if there is a correlation between the sparsity and zero-overlap and portfolio performance. We also compare performance as the estimated sparsity deviates from the true sparsity level of the inverse covariance used to generate the stock data.

4.7 Experiment Results

In this section, we present the results obtained from repeated portfolio optimization by using the portfolio strategies derived from using the Graphical Lasso validation techniques presented in section 4.6.3. For ease of representation in the results, we refer to the long-short portfolios as having 'short selling allowed' (S.S) and the long-only portfolio as those with the 'no short selling constraint' (N.S.S).

4.7.1 Long-short Portfolio Results

The average long-short portfolio results over all possible rebalancing periods (excluding 1 month and 2 months) are presented in Tables 4.1, Table 4.6 and Table 4.10, while each individual rebalancing period result can be found in Appendix D. We exclude the 1 month and 2 month results due to the fact that the maximum likelihoods at those periods are not defined because the estimated covariances are not invertible.

For the realized risk results, to assess the statistical robustness of the difference observed between a given portfolio strategy and the GLasso$^{\text{ORACLE}}$ portfolio strategy, for individual rebalancing periods, we report hypothesis test (t-test) results evaluating whether the observed difference between realized risks $\left(s_{Realized}^{\text{GLasso}^{\text{ORACLE}}} - s_{Realized}^{\text{GLasso}^{\text{NON-ORACLE}}}\right)$ has mean value equal to zero for individual rebalancing periods. We test for statistical significance at the 1% and 5% levels. Details of the hypothesis test can be found in Appendix E Table E.17. The result is significant at the 5% level when the symbol '*' is present, and is significant at the 1% level when the symbol '**' is present.

4.7.1.1 Realized Risk

Average S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.51	0.08	6.57	0.10	60.17	64.98
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.44	0.05	**6.52**	0.09	54.09	58.49
GLasso$^{\text{MAX LIKELIHOOD}}$	5.60	0.04	6.70	0.10	78.36	85.12
GLasso$^{\text{ORACLE}}$	5.65	0.04	6.54	0.09	75.09	81.61

Table 4.1 Long-short portfolio average realized risks

In general, we expect the predicted risks to be lower than the realized risks, which holds for all portfolio strategies in Table 4.1. For the realized risk, the oracle method, GLasso$^{\text{ORACLE}}$, having been fed the correct sparsity level, is expected to perform very well, possibly performing best. From Table 4.1, all the methods except for GLasso$^{\text{MAX LIKELIHOOD}}$ seem to perform very closely to one another. The GLasso$^{\text{MAX LIKELIHOOD-2}}$ has

the lowest realized risk, which is good, and we can say that it performs at least as well as the GLasso^{ORACLE} method, which is what we want. We look at individual rebalancing period performance.

3 months S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{MIN\ REALIZED\ RISK}$	4.91	0.13	7.32	0.12**	59.42	62.34
GLasso$^{MAX\ LIKELIHOOD\text{-}2}$	4.58	0.05	7.24	0.10	44.46	46.44
GLasso$^{MAX\ LIKELIHOOD}$	5.40	0.03	7.47	0.13**	83.21	87.67
GLasso^{ORACLE}	5.47	0.04	**7.04**	0.11	75.09	79.07

Table 4.2 Long-short portfolio 3 months realized risks

6 months S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{MIN\ REALIZED\ RISK}$	5.49	0.07	6.71	0.11**	64.22	68.55
GLasso$^{MAX\ LIKELIHOOD\text{-}2}$	5.36	0.06	6.63	0.09	48.74	51.73
GLasso$^{MAX\ LIKELIHOOD}$	5.57	0.04	6.79	0.09**	79.58	85.38
GLasso^{ORACLE}	5.65	0.04	**6.60**	0.09	75.09	80.43

Table 4.3 Long-short portfolio 6 months realized risks

For the 3 months and 6 months rebalancing periods, results are consistent with the average results in terms of predicted risks. All predicted risks are smaller than the realized risks for all the different methods. Table 4.2 and Table 4.3 show that at the 3 month and 6 month rebalancing periods, the oracle method and the likelihood method, GLasso$^{MAX\ LIKELIHOOD\text{-}2}$ perform essentially the same in terms of realized risks. The other two methods, GLasso$^{MIN\ REALIZED\ RISK}$ and GLasso$^{MAX\ LIKELIHOOD}$ have realized risks

that are statistically significantly higher than the realized risks of the oracle method, so we can say these two methods perform worse than the oracle method.

1 year S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.68	0.08	6.27	0.11	59.70	65.16
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.79	0.06	**6.19**	0.10**	57.22	62.22
GLasso$^{\text{MAX LIKELIHOOD}}$	5.68	0.04	6.41	0.12**	77.17	84.74
GLasso$^{\text{ORACLE}}$	5.73	0.04	6.33	0.11	75.11	82.46

Table 4.4 Long-short portfolio 1 year realized risks

2 year S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.80	0.05	**6.12**	0.07	53.44	59.53
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.82	0.04	6.13	0.07**	66.67	74.75
GLasso$^{\text{MAX LIKELIHOOD}}$	5.75	0.03	6.19	0.07	72.85	81.94
GLasso$^{\text{ORACLE}}$	5.69	0.02	6.25	0.07	75.11	84.57

Table 4.5 Long-short portfolio 2 years realized risks

For the 1 year and 2 year periods, results in Table 4.4 and Table 4.5 show that the likelihood method, GLasso$^{\text{MAX LIKELIHOOD-2}}$, achieves statistically significantly lower realized risks than the oracle method. At 1 year, GLasso$^{\text{MAX LIKELIHOOD}}$ achieves a realized risk which is statistically significantly worse than the oracle method. For the 2 year period, the oracle method and all other methods except for GLasso$^{\text{MAX LIKELIHOOD-2}}$ perform equally. For all rebalancing periods, the likelihood method that uses 2 periods for training always performs better than the likelihood method that uses 1 period and it appears that the likelihood method, GLasso$^{\text{MAX LIKELIHOOD-2}}$, improves as T gets larger.

4.7.1.2 Empirical Risk

Average S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.51	0.08	5.18	0.14	6.57	0.10	60.17	64.98
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.44	0.05	4.86	0.07	**6.52**	0.09	54.09	58.49
GLasso$^{\text{MAX LIKELIHOOD}}$	5.60	0.04	5.88	0.08	6.70	0.10	78.36	85.12
GLasso$^{\text{ORACLE}}$	5.65	0.04	5.59	0.07	6.54	0.09	75.09	81.61

Table 4.6 Long-short portfolio average empirical risks

From the definition of empirical risk, we expect that the empirical risk will be lower than the realized risk. The realized risk is essentially calculated using the same optimally selected regularization to estimate the portfolio weights, but on future data. The average results in Table 4.6 shows that this is always true for all the Graphical Lasso methods. We check to see if these results are consistent over time by looking at individual rebalancing periods.

3 months S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	4.91	0.13	4.13	0.21	7.32	0.12	59.42	62.34
GLasso$^{\text{MAX LIKELIHOOD-2}}$	4.58	0.05	3.29	0.06	7.24	0.10	44.46	46.44
GLasso$^{\text{MAX LIKELIHOOD}}$	5.40	0.03	5.87	0.09	7.47	0.13	83.21	87.67
GLasso$^{\text{ORACLE}}$	5.47	0.04	4.97	0.07	**7.04**	0.11	75.09	79.07

Table 4.7 Long-short portfolio 3 months empirical risks

1 year S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.68	0.08	5.54	0.17	6.27	0.11	59.70	65.16
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.79	0.06	5.46	0.09	**6.19**	0.10	57.22	62.22
GLasso$^{\text{MAX LIKELIHOOD}}$	5.68	0.04	5.94	0.09	6.41	0.12	77.17	84.74
GLasso$^{\text{ORACLE}}$	5.73	0.04	5.85	0.09	6.33	0.11	75.11	82.46

Table 4.8 Long-short portfolio 1 year empirical risks

2 year S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.80	0.05	5.63	0.06	**6.12**	0.07	53.44	59.53
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.82	0.04	5.74	0.05	6.13	0.07	66.67	74.75
GLasso$^{\text{MAX LIKELIHOOD}}$	5.75	0.03	5.85	0.05	6.19	0.07	72.85	81.94
GLasso$^{\text{ORACLE}}$	5.69	0.02	5.92	0.06	6.25	0.07	75.11	84.57

Table 4.9 Long-short portfolio 2 years empirical risks

Results in Appendix D show that the empirical risk is always smaller than the realized risk at all rebalancing periods and can be seen in Table 4.7, Table 4.8 and Table 4.9. The difference between the empirical risks and realized risks appears to be larger for $\frac{T}{N} < 1$, with the 3 months results showing the largest difference.

4.7.1.3 Empirical and Realized Likelihood

Average S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.57	0.10	60.17	64.98	52.17	48187.15	49763.63
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.52**	0.09	54.09	58.49	57.27	48154.77	**49961.80**
GLasso$^{\text{MAX LIKELIHOOD}}$	6.70	0.10	78.36	85.12	39.28	**48449.65**	49146.59
GLasso$^{\text{ORACLE}}$	6.54	0.09	75.09	81.61	41.94	48447.75	49266.49

Table 4.10 Long-short portfolio average empirical and realized likelihoods

We expect that there will be some correlation between the realized likelihood and portfolio performance in terms of realized risk. We also expect that the Graphical Lasso method that maximizes Gaussian likelihood using 2 periods for training will have the highest likelihood, which is the case in Table 4.10. The average results in Table 4.10 also show that although the performance is not perfectly correlated with the realized likelihood, the oracle and likelihood methods, GLasso$^{\text{ORACLE}}$ and GLasso$^{\text{MAX LIKELIHOOD}}$, have the highest realized likelihoods. The empirical likelihood on the other hand shows perfect correlation with portfolio performance. We look at some individual rebalancing period results from Appendix D to see if there is consistency in performance.

3 months S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	7.32	0.12	59.42	62.34	47.03	12657.11	14599.16
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.24	0.10	44.46	46.44	59.69	12566.99	**15053.35**
GLasso$^{\text{MAX LIKELIHOOD}}$	7.47	0.13	83.21	87.67	26.97	**13333.21**	13784.95
GLasso$^{\text{ORACLE}}$	**7.04**	0.11	75.09	79.07	33.94	13321.53	14064.92

Table 4.11 Long-short portfolio 3 months empirical and realized likelihoods

1 year S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.27	0.11	59.70	65.16	54.73	53668.56	55178.46
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.19**	0.10	57.22	62.22	55.89	53737.92	**55290.66**
GLasso$^{\text{MAX LIKELIHOOD}}$	6.41	0.12	77.17	84.74	43.08	53854.40	54590.93
GLasso$^{\text{ORACLE}}$	6.33	0.11	75.11	82.46	44.55	**53864.81**	54681.56

Table 4.12 Long-short portfolio 1 year empirical and realized likelihoods

2 year S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	**6.12**	0.07	53.44	59.53	63.25	107779.65	**109590.02**
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.13	0.07	66.67	74.75	55.69	107929.11	109179.59
GLasso$^{\text{MAX LIKELIHOOD}}$	6.19	0.07	72.85	81.94	52.40	**107930.82**	108926.92
GLasso$^{\text{ORACLE}}$	6.25	0.07	75.11	84.57	51.19	107911.28	108815.31

Table 4.13 Long-short portfolio 2 years empirical and realized Likelihoods

Individual rebalancing period results are consistent with the average results, showing that although the portfolio realized risk is not perfectly correlated with the realized likelihood, the oracle and likelihood methods always have the highest realized likelihoods. The empirical likelihood on the other hand shows no correlation with portfolio performance, but in almost all instances, the likelihood method that uses 2 periods for training, GLasso$^{\text{MAX LIKELIHOOD-2}}$, always has the highest empirical likelihood as expected. More correlation analyses will be performed in section 4.7.1.4 additionally using hypothesis tests.

4.7.1.4 Portfolio Measures Correlation Analysis

In order to better understand the results from the synthetic experiment, we look at the results over all rebalancing periods for each Graphical Lasso portfolio strategy and see if there is a correlation between the sparsity, zero-overlap, empirical likelihood and realized likelihood with portfolio performance (realized risk). We perform an OLS multiple linear regression (introduced in Appendix A section A.5.3) using STATA where the realized risk is treated as the dependent variable and sparsity, zero-overlap, empirical likelihood and realized likelihood as the independent

variables for all portfolio strategies. We estimate the regression coefficients for each predictor variable (sparsity, zero-overlap etc.) and perform hypothesis tests (t-test) testing if these coefficients are equal to zero. Results are presented in Table 4.14.

Realized risk	Coefficient(β)	Std.Err.	t-stat	p-value	[95% Conf.	Interval]
Sparsity	.5950606	.078562	7.57	0.000	.4306284	.7594929
Zero-overlap	-.4870172	.0690815	-7.05	0.000	-.6316065	-.342428
Realized likelihood	-.0011408	.0002808	-4.06	0.001	-.0017284	-.0005531
Empirical likelihood	.0011537	.0002816	4.10	0.001	.0005643	.001743
constant	.0768646	1.415855	0.05	0.957	-2.886553	3.040282

Table 4.14 OLS multiple linear regression and t-tests showing portfolio realized risk predicted using sparsity, zero-overlap, empirical likelihood and realized likelihood for the different Graphical Lasso portfolio strategies

Table 4.14 presents the multiple regression results where each β is the associated regression coefficient for a particular predictor variable. From Table 4.14, the OLS multiple regression model is given by

$$\begin{aligned} Realized\ risk = {} & 0.0769 + 0.5951\ \text{Sparsity} - 0.4870\ \text{Zero\_overlap} \\ & - .0011\ \text{Realized likelihood} + .0012\ \text{Empirical likelihood} \end{aligned}$$

From Table 4.14, the p-value for sparsity is less than 0.05, and the coefficient is positive, therefore, at the 5% level of significance, the higher the sparsity, the higher the portfolio realized risk. The p-value for zero-overlap is less than 0.05, and the

coefficient is negative, therefore at the 5% level of significance, the higher the zero-overlap, the lower the portfolio realized risk. The p-value for realized likelihood is less than 0.05, and the coefficient is negative, therefore at the 5% level of significance, the higher the realized likelihood, the lower the portfolio realized risk. Lastly, the p-value for empirical likelihood is less than 0.05, and the coefficient is positive, therefore at the 5% level of significance, the higher the empirical likelihood, the higher the portfolio realized risk.

4.7.2 Long-only Portfolio Results

Following section 4.7.1, the average long-only portfolio results over all possible rebalancing periods (excluding 1 month and 2 months) are presented in Tables 4.15, Table 4.20 and Table 4.24, while each individual rebalancing period result can be found in Appendix D. For the realized risk results, to assess the statistical robustness of the difference observed between a given portfolio strategy and the $\text{GLasso}^{\text{ORACLE}}$ portfolio strategy, for individual rebalancing periods, we report hypothesis test (*t*-test) results evaluating whether the observed difference between realized risks $\left(s_{Realized}^{\text{GLasso}^{\text{ORACLE}}} - s_{Realized}^{\text{GLasso}^{\text{NON-ORACLE}}}\right)$ has mean value equal to zero for individual rebalancing periods. The results of these hypothesis tests are presented in E Table E.18.

4.7.2.1 Realized Risk

Average N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.10	0.09	7.10	0.12	58.82	63.56
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.10	0.08	**7.03**	0.11	54.09	58.49
GLasso$^{\text{MAX LIKELIHOOD}}$	6.08	0.06	7.21	0.11	78.36	85.12
GLasso$^{\text{ORACLE}}$	6.17	0.06	7.08	0.11	75.09	81.61

Table 4.15 Long-only portfolio average realized risks

Like the long-short portfolio, we expect the predicted risks to be lower than the realized risks, which holds for all portfolio strategies in Table 4.15. The oracle method, GLasso$^{\text{ORACLE}}$, is expected to perform very well, possibly performing best. From Table 4.15, the GLasso$^{\text{MAX LIKELIHOOD-2}}$ method has the lowest risk, so it is safe to conclude that this method performs at least as well as the oracle method. The other methods, GLasso$^{\text{MIN REALIZED RISK}}$ and GLasso$^{\text{MAX LIKELIHOOD}}$ appear to have realized risks that are higher than that of the oracle method. We check to see individual rebalancing period performance using hypothesis tests to check for statistical significance in performance differences.

3 months N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.46	0.11	7.76	0.13**	56.05	58.82
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.40	0.08	7.62	0.12	44.46	46.44
GLasso$^{\text{MAX LIKELIHOOD}}$	5.77	0.05	7.88	0.14**	83.21	87.67
GLasso$^{\text{ORACLE}}$	5.94	0.06	**7.53**	0.12	75.09	79.07

Table 4.16 Long-only portfolio 3 months realized risks

6 months N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.05	0.09	7.16	0.14**	62.73	66.88
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.04	0.09	**7.06**	0.12**	48.74	51.73
GLasso$^{\text{MAX LIKELIHOOD}}$	6.03	0.07	7.26	0.13**	79.58	85.38
GLasso$^{\text{ORACLE}}$	6.15	0.07	7.11	0.12	75.09	80.43

Table 4.17 Long-only portfolio 6 months realized risks

1 year N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.35	0.08	6.85	0.12	63.35	69.18
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.38	0.07	**6.79**	0.10**	57.22	62.22
GLasso$^{\text{MAX LIKELIHOOD}}$	6.20	0.06	6.98	0.11**	77.17	84.74
GLasso$^{\text{ORACLE}}$	6.26	0.05	6.92	0.10	75.11	82.46

Table 4.18 Long-only portfolio 1 year realized risks

2 year N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.41	0.07	6.75	0.08	54.16	60.43
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.40	0.05	**6.74**	0.08**	66.67	74.75
GLasso$^{\text{MAX LIKELIHOOD}}$	6.32	0.04	6.79	0.08	72.85	81.94
GLasso$^{\text{ORACLE}}$	6.25	0.03	6.83	0.09	75.11	84.57

Table 4.19 Long-only portfolio 2 years realized risks

At 6 months, 1 year and 2 years, only the GLasso$^{\text{MAX LIKELIHOOD-2}}$ performs statistically significantly better than the oracle method. At 3 months, 6 months and 1 year, the GLasso$^{\text{MAX LIKELIHOOD}}$ method performs statistically significantly worse than the oracle method. From these results, we can conclude that GLasso$^{\text{MAX LIKELIHOOD-2}}$ performs better than GLasso$^{\text{MAX LIKELIHOOD}}$ in general. The GLasso$^{\text{MAX LIKELIHOOD-2}}$ achieves an even better performance than the long-short portfolio problem without the short selling

constraint. At the 3 months rebalancing period, the oracle method and GLasso$^{\text{MAX LIKELIHOOD-2}}$ perform the same while at the 6 months rebalancing period, GLasso$^{\text{MAX LIKELIHOOD-2}}$ has a statistically significantly better realized risk than the oracle method. For both the 3 months and 6 months rebalancing periods, GLasso$^{\text{MAX LIKELIHOOD}}$ and GLasso$^{\text{MIN REALIZED RISK}}$ have statistically significantly worse realized risks than the oracle method. At 1 year, all methods except the likelihood methods perform essentially the same as the oracle method. At 2 years, all methods perform same as the oracle method except for the GLasso$^{\text{MAX LIKELIHOOD-2}}$ which achieves a statistically significantly better realized risk than the oracle method. In general, the GLasso$^{\text{MAX LIKELIHOOD-2}}$ method performs very well despite not using any additional information about realized risks and the true sparsity level, which shows that using likelihood to select the correct regularization is a very promising method.

4.7.2.2 Empirical Risk

Average N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.10	0.09	5.98	0.11	7.10	0.12	58.82	63.56
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.10	0.08	5.81	0.09	**7.03**	0.11	54.09	58.49
GLasso$^{\text{MAX LIKELIHOOD}}$	6.08	0.06	6.48	0.09	7.21	0.11	78.36	85.12
GLasso$^{\text{ORACLE}}$	6.17	0.06	6.26	0.09	7.08	0.11	75.09	81.61

Table 4.20 Long-only portfolio average empirical risks

Like the long-short portfolio results, the empirical risk is lower than the realized risk as seen in Table 4.20, which is what we expect. We check to see if these results are consistent over time by looking at individual rebalancing periods from Appendix D.

3 months N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.46	0.11	5.10	0.17	7.76	0.13	56.05	58.82
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.40	0.08	4.67	0.10	7.62	0.12	44.46	46.44
GLasso$^{\text{MAX LIKELIHOOD}}$	5.77	0.05	6.36	0.10	7.88	0.14	83.21	87.67
GLasso$^{\text{ORACLE}}$	5.94	0.06	5.70	0.10	**7.53**	0.12	75.09	79.07

Table 4.21 Long-only portfolio 3 months empirical risks

1 year N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.35	0.08	6.32	0.09	6.85	0.12	63.35	69.18
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.38	0.07	6.23	0.07	**6.79**	0.10	57.22	62.22
GLasso$^{\text{MAX LIKELIHOOD}}$	6.20	0.06	6.55	0.07	6.98	0.11	77.17	84.74
GLasso$^{\text{ORACLE}}$	6.26	0.05	6.48	0.07	6.92	0.10	75.11	82.46

Table 4.22 Long-only portfolio 1 year empirical risks

2 year N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.41	0.07	6.40	0.03	6.75	0.08	54.16	60.43
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.40	0.05	6.45	0.05	**6.74**	0.08	66.67	74.75
GLasso$^{\text{MAX LIKELIHOOD}}$	6.32	0.04	6.52	0.06	6.79	0.08	72.85	81.94
GLasso$^{\text{ORACLE}}$	6.25	0.03	6.57	0.06	6.83	0.09	75.11	84.57

Table 4.23 Long-only portfolio 2 year empirical risks

Results in Appendix D show that the empirical risk is always lower than the realized risk at all rebalancing periods and can be seen in Table 4.21, Table 4.22 and Table

4.23. The difference between the empirical risks and realized risks appears to be larger for $\frac{T}{N} < 1$, with the 3 months results showing the largest difference.

4.7.2.3 Empirical and Realized Likelihood

Average N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	7.10	0.12	58.82	63.56	53.37	48127.37	49802.66
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**7.03**	0.11	54.09	58.49	57.27	48154.77	**49961.80**
GLasso$^{\text{MAX LIKELIHOOD}}$	7.21	0.11	78.36	85.12	39.28	**48449.65**	49146.59
GLasso$^{\text{ORACLE}}$	7.08	0.11	75.09	81.61	41.94	48447.75	49266.49

Table 4.24 Long-only portfolio average empirical and realized likelihoods

Like the long-short portfolio results, the average results in Table 4.24 show that although the performance is not perfectly correlated with the realized likelihood, the oracle and likelihood methods, GLasso$^{\text{ORACLE}}$ and GLasso$^{\text{MAX LIKELIHOOD}}$, have the highest realized likelihoods. The empirical likelihood also shows perfect correlation with portfolio performance like the long-short portfolio case. We look individual rebalancing period results to see if there is consistency in performance.

3 months N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	7.76	0.13	56.05	58.82	50.09	12464.87	14720.15
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.62	0.12	44.46	46.44	59.69	12566.99	**15053.35**
GLasso$^{\text{MAX LIKELIHOOD}}$	7.88	0.14	83.21	87.67	26.97	**13333.21**	13784.95
GLasso$^{\text{ORACLE}}$	**7.53**	0.12	75.09	79.07	33.94	13321.53	14064.92

Table 4.25 Long-only portfolio 3 months empirical and realized likelihoods

1 year N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.85	0.12	63.35	69.18	52.05	53780.21	55081.61
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.79**	0.10	57.22	62.22	55.89	53737.92	**55290.66**
GLasso$^{\text{MAX LIKELIHOOD}}$	6.98	0.11	77.17	84.74	43.08	53854.40	54590.93
GLasso$^{\text{ORACLE}}$	6.92	0.10	75.11	82.46	44.55	**53864.81**	54681.56

Table 4.26 Long-only portfolio 1 year empirical and realized likelihoods

2 year N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.75	0.08	54.16	60.43	63.07	107762.24	**109537.14**
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.74**	0.08	66.67	74.75	55.69	107929.11	109179.59
GLasso$^{\text{MAX LIKELIHOOD}}$	6.79	0.08	72.85	81.94	52.40	**107930.82**	108926.92
GLasso$^{\text{ORACLE}}$	6.83	0.09	75.11	84.57	51.19	107911.28	108815.31

Table 4.27 Long-only portfolio 2 years empirical and realized likelihoods

Individual rebalancing period results verify our conclusion from the average results, and also are the same as the long-short portfolio results. Although the portfolio realized risk is not perfectly correlated with the realized likelihood, the oracle and likelihood methods always have the highest realized likelihoods. The empirical likelihood on the other hand shows no correlation with portfolio performance, but in almost all instances, the likelihood method that uses 2 periods for training, GLasso$^{\text{MAX LIKELIHOOD-2}}$, always has the highest empirical likelihood as expected. More correlation analyses will be performed in section 4.7.2.4 additionally using hypothesis tests.

4.7.2.4 Portfolio Measures Correlation Analysis

Like the long-short portfolio problem, we look at the results over all rebalancing periods for each Graphical Lasso portfolio strategy and see if there is a correlation between the sparsity, zero-overlap, empirical likelihood and realized likelihood with portfolio performance (realized risk). We perform an OLS multiple linear regression (detailed in Appendix A) using STATA, where the realized risk is treated as the dependent variable and sparsity, zero-overlap, empirical likelihood and realized likelihood as the independent variables for all portfolio strategies. We estimate the regression coefficients for each predictor variable (sparsity, zero-overlap etc.) and perform hypothesis tests (*t*-test) testing if these coefficients are equal to zero. Results are presented in Table 4.14.

Realized risk	Coefficient(β)	Std. Err.	t	P>|t|	[95% Conf.	Interval]
Sparsity	.5758099	.0736858	7.81	0.000	.4215837	.7300361
Zero-overlap	-.4722801	.0651268	-7.25	0.000	-.6085921	-.3359681
Realized likelihood	-.001087	.0002415	-4.50	0.000	-.0015926	-.0005815
Empirical likelihood	.0011011	.0002423	4.54	0.000	.0005938	.0016083
constant	.8298601	1.237367	0.67	0.511	-1.759978	3.419699

Table 4.28 OLS multiple linear regression and t-tests showing portfolio realized risk predicted using sparsity, zero-overlap, empirical likelihood and realized likelihood for the different Graphical Lasso portfolio strategies

Table 4.28 presents the multiple regression results where each β is the associated regression coefficient for a particular predictor variable. Using STATA, the OLS

multiple linear regression is fit and the model with the associated t-test results are shown in Table 4.28. The OLS multiple regression model is given by

$$\begin{aligned} Realized\ risk = {} & 0.8299 + 0.5758\ \text{Sparsity} - 0.4723\ \text{Zero\_overlap} \\ & - .0011\ \text{Realized likelihood} + .0011\ \text{Empirical likelihood} \end{aligned}$$

From Table 4.28, the p-value for sparsity is less than 0.05, and the coefficient is positive, therefore at the 5% level of significance, the higher the sparsity, the higher the portfolio realized risk. The p-value for zero-overlap is less than 0.05, and the coefficient is negative therefore at the 5% level of significance, the higher the zero-overlap, the lower the portfolio realized risk. The p-value for realized likelihood is less than 0.05, and the coefficient is negative, therefore at the 5% level of significance, the higher the realized likelihood, the lower the portfolio realized risk. The p-value for empirical likelihood is less than 0.05, and the coefficient is positive, therefore at the 5% level of significance, the higher the empirical likelihood, the higher the portfolio realized risk. These results are consistent with the long-short portfolio case.

4.8 Summary

In order to apply the sparse inverse covariance estimation methodology in Finance, we performed Markowitz global minimum variance portfolio optimization using Graphical Lasso on synthetically generated stock market data from a known inverse covariance. Using validation techniques, we optimized certain portfolio criteria, and come up with new portfolio strategies, which we compared to a benchmark portfolio

strategy, GLasso$^{\text{ORACLE}}$, that estimates the inverse covariance with identical sparsity level as the true inverse covariance. We used this oracle method as a baseline method for performance evaluation. We evaluated portfolio performance of all the different portfolio strategies across different rebalancing periods with and without short selling constraints.

Our results for the long-short portfolio problem and long-only portfolio problem were almost identical. They both resulted in predicted risks that were smaller than realized risks at all rebalancing periods. For the long-short portfolio problem and the long-only portfolio problem, the likelihood method, GLasso$^{\text{MAX LIKELIHOOD-2}}$ achieved statistically significantly best realized risks from 6 months and above. The two other competing methods, GLasso$^{\text{MIN REALIZED RISK}}$ and GLasso$^{\text{MAX LIKELIHOOD}}$ achieved statistically significantly worse realized risks than the oracle method at almost all the rebalancing periods. Performance of the GLasso$^{\text{MAX LIKELIHOOD-2}}$ was even better with the addition of the short selling constraint.

The realized risk and empirical risk are essentially calculated using the same optimally selected regularization to estimate the portfolio weights, but the realized risk is calculated on future data. For this reason, we expected the realized risk to be larger than the empirical risk, and this was true for both the long-short and long-only portfolios. The difference between the empirical risks and realized was larger for $\frac{T}{N} < 1$, with the 3 months results showing the largest difference.

For both the long-short portfolios and the long-only portfolios, GLasso$^{\text{MAX LIKELIHOOD-2}}$ almost always had the highest empirical likelihood as expected. We saw a positive correlation with sparsity and portfolio realized risk, a negative correlation with zero-

overlap and portfolio realized risk, and a negative correlation between realized likelihood and portfolio realized risk. These results were as expected.

In summary, our results showed that the likelihood method, GLasso$^{\text{MAX LIKELIHOOD-2}}$, performed very well despite not using any additional information about realized risks and the true sparsity level, which shows that using likelihood to select the correct regularization for Graphical Lasso is a very promising method and should be explored further.

Chapter 5

Covariance Estimation and Portfolio Optimization: A Comparison between Existing Methods and the New Sparse Inverse Covariance Method

5.1 Covariance and Mean-Variance Portfolio Optimization

There are different applications of covariance matrices in portfolio optimization that have been studied in literature. We present some of the popular covariance estimators used in the Markowitz mean-variance portfolio optimization problem in literature and compare the new method of sparse inverse covariance estimation presented in chapter 4. Using the newly developed sparse portfolio strategies from chapter 4, we perform an in-depth comparative analysis against existing methods when applied to a stock market portfolio optimization problem. In addition to the existing methods in literature, we also compare performance with the Naïve baseline method, which is the equally weighted portfolio. We look at how the different methods perform in terms of portfolio realized risk, portfolio expected return and portfolio Sharpe ratio.

5.2 Covariance Estimation: Existing Methods

5.2.1 Direct Optimization

The sample covariance matrix estimator is the simplest covariance estimator of N asset returns. For an estimation time horizon of length T, the number of available data is $N \times T$. A very common circumstance in portfolio selection is that the number of assets N is of the same order of magnitude or larger than the estimation time horizon T. The sample covariance matrix suffers from two main deficiencies. Firstly, when the number of observations is less than the number of assets N, the sample covariance matrix is not full rank, hence it is not invertible. Secondly, even if the sample covariance is full rank, its inverse only provides a biased estimator of the inverse population covariance matrix. As suggested in literature [59], in the portfolio optimization problems, we use the Moore-Penrose pseudoinverse, also called the generalized inverse [60], of the covariance matrix for $T < N$. Replacing the inverse of the covariance matrix with the pseudoinverse in the optimization problem allows one to get a unique combination of portfolio weights. It should be noted that when $T < N$, the optimization problem remains undetermined and the pseudoinverse solution is just a natural choice among the infinite undetermined solutions to the portfolio optimization problem [50].

In the same regime $T < N$, the problem does not arise for the other covariance estimators that will be presented subsequently, because they typically give positive definite covariance matrices for any value of T/N including $\frac{T}{N} < 1$ [50].

5.2.2 Spectral Estimators

The first class of methods includes three different estimators of the covariance matrix, which make use of the spectral properties of the correlation matrix. The fundamental idea behind the spectral estimators is that the eigenvalues of the sample covariance matrix carry different information depending on their value [50].

5.2.2.1 The Single Index Model

The first method we consider is Sharpe's single index (SI) model [59, 61-63]. In this model, for a set of stocks $i = 1, \dots, N$, stock returns $r_i(t)$ are described by the set of linear equations [9,64-66]

$$r_i(t) = \beta_i f(t) + \varepsilon_i(t), \tag{5.1}$$

where returns are given by the linear combination of a single random variable, the index $f(t)$, and an idiosyncratic stochastic term $\varepsilon_i(t)$. The parameters β_i can be estimated by linear regression of stock return time series on the index return. The covariance matrix associated with the model is $\boldsymbol{S}^{(SI)} = \sigma_{00}\boldsymbol{\beta}\boldsymbol{\beta}^T + \boldsymbol{D}$, where σ_{00} is the variance of the index, β is the vector of parameters β_i, and $\boldsymbol{D}$ is the diagonal matrix of variances of ε_i. This method will be referred to hereafter as SI.

5.2.2.2 Random Matrix Theory Models

The other two spectral methods make use of Random Matrix Theory (RMT) [60, 61, 67]. If the N variables of the system are independently and identically distributed

with finite variance σ^2, then in the limit $T, N \to \infty$, with a fixed ratio T/N, the eigenvalues of the sample covariance matrix are bounded from above by

$$\lambda_{max} = \sigma^2 \left(1 + \frac{N}{T} + 2\sqrt{\frac{N}{T}}\right), \tag{5.2}$$

where σ^2=1 for correlation matrices. In most practical cases, one finds that the largest eigenvalue $\lambda_1$of the sample correlation matrix of stocks is inconsistent with RMT, i.e. $\lambda_1 \gg \lambda_{max}$. In fact, the largest eigenvectors is typically identified with the market mode [50]. To cope with this, [61] propose to modify the null hypothesis of RMT so that system correlations can be described in terms of a one factor model instead of a pure random model. Under such less restrictive null hypothesis, the value of λ_{max} is still given by (5.2), but now $\sigma^2 = 1 - \lambda_1/N$.

We consider two different methods that apply RMT to the covariance estimation problem. These methods reduce the impact of eigenvalues smaller than λ_{max} onto the estimate of portfolio weights. The first method which will be referred to hereafter as 'RMT-0', was proposed by [68], while the second method was proposed by [69] and will be referred to as 'RMT-M'.

i. RMT-0

One diagonalizes the sample correlation matrix and replaces all eigenvalues smaller than λ_{max} with 0. One then transforms back the modified diagonal matrix in the standard basis obtaining the matrix $\boldsymbol{H}^{(RMT-0)}$. The filtered

correlation matrix $\boldsymbol{C}^{(RMT-0)}$ is obtained by simply forcing the diagonal elements of $\boldsymbol{H}^{(RMT-0)}$ to 1. Finally, the filtered covariance matrix $\boldsymbol{S}^{(RMT-0)}$ is the matrix of elements $\sigma_{ij}^{(RMT-0)} = c_{ij}^{(RMT-0)}\sqrt{\sigma_{ii}\sigma_{jj}}$, where $c_{ij}^{(RMT-0)}$ are the entries of $\boldsymbol{C}^{(RMT-0)}$ and $\sigma_{ii}\ and\ \sigma_{jj}$ are the sample variances of variables *i* and *j* respectively [50, 68].

ii. RMT-M

This method proposed by [69] diagonalizes the sample correlation matrix and replaces all the eigenvalues smaller than λ_{max} with their average value. Then one transforms back the modified diagonal matrix in the original basis obtaining the matrix $\boldsymbol{H}^{(RMT-M)}$ of elements $h_{ij}^{(RMT-M)}$. It is important to note that replacing the eigenvalues smaller than λ_{max} with their average value preserves the trace of the matrix. Finally, the filtered correlation matrix $\boldsymbol{C}^{(RMT-M)}$is the matrix of elements $c_{ij}^{(RMT-M)} = h_{ij}^{(RMT-M)}/\sqrt{h_{ii}^{(RMT-M)}h_{jj}^{(RMT-M)}}$.

The covariance matrix $\boldsymbol{S}^{(RMT-M)}$ to be used in the portfolio optimization is the matrix of elements $\sigma_{ij}^{(RMT-M)} = c_{ij}^{(RMT-M)}\sqrt{\sigma_{ii}\sigma_{jj}}$, where $\sigma_{ii}\ and\ \sigma_{jj}$ are again the sample variances of variables *i* and *j* respectively [50, 69].

5.2.3 Shrinkage Estimators

The shrinkage estimators comprise of linear shrinkage methods. Linear shrinkage is a well-established technique in high-dimensional inference problems, when the size of data is small compared to the number of unknown parameters in the model [50].

In such cases , the sample covariance matrix is the best estimator in terms of actual fit to the data, but it is suboptimal because the number of parameters to be fit is larger than the amount of data available[65,70]. The idea is to construct a more robust estimate $\boldsymbol{Q}$ of the covariance matrix by shrinking the sample covariance matrix $\boldsymbol{S}$ to a target matrix $\boldsymbol{T}$, which is typically positive definite and has a lower variance. The shrinking is obtained by computing

$$\boldsymbol{Q} = \alpha\boldsymbol{T} + (1 - \alpha)\boldsymbol{S}, \tag{5.3}$$

where α is a parameter named the shrinkage intensity.

Optimal Shrinkage Intensity

Following [50], as $\hat{\alpha}^*$ (the optimal shrinkage intensity), we use the unbiased estimate analytically calculated in [71]. It is well known that the optimal shrinkage intensity, $\hat{\alpha}^*$, may be determined analytically. Specifically [71] derived a simple theorem for choosing $\hat{\alpha}^*$ that guarantees minimal MSE without the need of having to specify any underlying distributions and without requiring computationally expensive procedures such as cross validation [71]. Each shrinkage method will be associated with an analytical formula for the optimal shrinkage intensity and will be presented subsequently.

We consider three different shrinkage estimators from literature. Each one is characterized by a specific target matrix.

5.2.3.1 Shrinkage to Single Index

The shrinkage to single index method estimates the covariance matrix of stock returns by an optimally weighted average of two existing estimators: the sample covariance matrix and the single-index covariance matrix.

This method uses the target matrix $\boldsymbol{T} = \boldsymbol{S}^{(SI)} = \sigma_{00}\boldsymbol{\beta}\boldsymbol{\beta}^T + \boldsymbol{D}$, i.e., the single index covariance matrix discussed in section (5.2.2.1). This target was first proposed in the context of portfolio optimization by [6].

$$t = s^{(SI)}$$

$$\hat{\alpha}^* = \frac{\sum_{i=1}^{N} Var(s_i) - Cov(t_i, s_i)}{\sum_{i=1}^{N} E[(t_i - s_i)]}$$

5.2.3.2 Shrinkage to Common Covariance

This method uses the target matrix $\boldsymbol{T}$, where the diagonal elements are all equal to the average of sample variances, while non-diagonal elements are equal to the average of sample covariances. With this method, the heterogeneity of stock variances and of stock covariances is minimized [65]. This method has been proposed for the analysis of bioinformatics data in [71] and used in the analysis of financial data in [65].

$$t_{i,j} = \begin{cases} v = avg(s_{ii}) & if\ i = j \\ c = avg(s_{ij}) & if\ i \neq j \end{cases}$$

$$\hat{\alpha}^* = \frac{\sum_{i \neq j} \widehat{Var}(s_{ij}) + \sum_i \widehat{Var}(s_{ii})}{\sum_{i \neq j} (s_{ij} - c)^2 + \sum_i (s_{ii} - v)^2},$$

where v is the average of sample variances and c is the average of sample covariances.

5.2.3.3 Shrinkage to Constant Correlation

This method has a more structured target matrix $\boldsymbol{T}$ and is used in [7]. The estimator is obtained by first shrinking the correlation matrix by the sample standard deviations. The constant correlation target $\boldsymbol{T}$ is a matrix with diagonal elements equal to one, and off-diagonal elements equal to the average sample correlation between the elements of the system.

$$t_{i,j} = \begin{cases} s_{ii} & if\ i = j \\ \bar{r}\sqrt{s_{ii} s_{jj}} & if\ i \neq j \end{cases}$$

$$\hat{\alpha}^* = \frac{\sum_{i \neq j} \widehat{Var}\left(s_{ij}\right) - \bar{r} f_{ij})}{\sum_{i \neq j} (s_{ij} - \bar{r}\sqrt{s_{ii} s_{jj}})^2},$$

where $\bar{r}$ is the average of sample correlations.

5.3 Experiment on Stock Market Data

In order to evaluate the proposed sparse precision approach, we apply Graphical Lasso to financial daily return time series. In the experiments, we estimate

covariance matrices of stock returns using both the Graphical Lasso portfolio strategies and the covariance estimators used in literature. The realized risk, portfolio return and Sharpe ratio are compared for the different covariance estimates. We also compare performance with the Naïve baseline method, which is the equally weighted portfolio.

5.3.1 Data

The dataset follows that used in [50] and consists of the daily returns of $N = 90$ highly capitalized stocks traded at NYSE and included in the NYSE US 100 Index. For these stocks, the closing prices are available in the eleven year period from 1 January 1997 to 31 December 2007. The ticker symbols of the investigated stocks are AA, ABT, AIG, ALL, APA, AXP, BA, BAC, BAX, BEN, BK, BMY, BNI, BRK-B, BUD, C, CAT, CCL, CL, COP, CVS, CVX, D, DD, DE, DIS, DNA, DOW, DVN, EMC, EMR, EXC, FCX, FDX, FNM, GD, GE, GLW, HAL, HD, HIG, HON, HPQ, IBM, ITW, JNJ, JPM, KMB, KO, LEH, LLY, LMT, LOW, MCD, MDT, MER, MMM, MO, MOT, MRK, MRO, MS, NWS-A, OXY, PCU, PEP, PFE, PG, RIG, S, SGP, SLB, SO, T, TGT, TRV, TWX, TXN, UNH, UNP, USB, UTX, VLO, VZ, WAG, WB, WFC, WMT, WYE, XOM. As a reference index in the SI model and in the shrinkage to single index model, we use the Standard & Poor's 500 index, which is a widely used broadly-based market index.

5.3.2 Methodology

We extend the work done by [50] by carrying out a comparative performance analysis of the newly proposed sparse inverse covariance portfolio strategies from chapter 4

with the covariance estimators presented in section 5.2. We follow the methodology from section 4.6.2, where at time t_0 the portfolio is selected by choosing the optimal weights that solve the global minimum variance optimization problem with and without short selling constraints.

Specifically, we solve the global minimum variance optimization problem without short selling constraints which gives rise to long-short portfolios, and with short selling constraints which gives rise to long-only portfolios. The input to the optimization problem is the estimated covariance matrix $\boldsymbol{S}^{(EST)}$ calculated using the T days preceding t_0 (in-sample training period) and obtained with one of the methods (i.e. Direct Optimization, SI, Graphical Lasso etc.). We call $\boldsymbol{S}^{(EMP)}$ the empirical covariance matrix (same as the one used in the Direct Optimization method). Like the synthetic data experiment, the portfolio performance is evaluated T days after t_0 (the out-of-sample test period). Following the synthetic data experiment, we use seven different time windows T of 1, 2, 3, 6, 9, 12, and 24 months.

In summary, the covariance estimators that have been presented and the newly proposed strategies from chapter 4 that will be evaluated during this experiment are as follows.

i. Direct Optimization: Markowitz direct optimization with the sample covariance matrix
ii. Single Index (SI) model: The Single Index model
iii. RMT-0: A random matrix theory estimator
iv. RMT-M: A random matrix theory estimator

v. Shrinkage_SI: Shrinkage to Single Index

vi. Shrinkage_Cov: Shrinkage to common covariance

vii. Shrinkage_Corr: Shrinkage to constant correlation

viii. Naïve: This is the equally weighted portfolio described in this section.

ix. $\text{GLasso}^{\text{MIN REALIZED RISK}}$: Graphical Lasso method that optimizes realized risk

x. $\text{GLasso}^{\text{MAX RETURN}}$: Graphical Lasso method that optimizes portfolio return

xi. $\text{GLasso}^{\text{MAX SHARPE}}$: Graphical Lasso method that optimizes Sharpe ratio

xii. $\text{GLasso}^{\text{MAX LIKELIHOOD-2}}$: Graphical Lasso method that optimizes Gaussian likelihood using 2 periods for training

xiii. $\text{GLasso}^{\text{MAX LIKELIHOOD}}$: Graphical Lasso method that optimizes Gaussian likelihood using 1 period for training

Note the addition of 2 new portfolio strategies, $\text{GLasso}^{\text{MAX RETURN}}$ and $\text{GLasso}^{\text{MAX SHARPE}}$ which follow the exact same methodology as $\text{GLasso}^{\text{MIN REALIZED RISK}}$ in chapter 4, only optimizing different portfolio criteria (portfolio return and Sharpe ratio).

5.3.3 Performance Measures

To evaluate the performance of the different covariance estimators, we compare portfolio realized risk (defined in chapter 4 section 4.6.4.2), portfolio expected return (defined in section 4.2.1) and the Sharpe ratio which will be described subsequently.

5.3.3.1 Sharpe ratio

The Sharpe ratio is a ratio of return versus risk, which our goal is to maximize [54,57,58]. The higher the Sharpe ratio is, the more return an investor is getting per

unit of risk. The lower the Sharpe ratio is, the more risk the investor is shouldering to earn additional returns. Thus, the Sharpe ratio ultimately "levels the playing field" among different portfolios by indicating which are shouldering excessive risk.

The Sharpe ratio is defined as

$$Sharpe\ ratio = \frac{\mu_P - r_f}{\sigma_P},$$

where μ_P is the expected portfolio return, r_f is the risk-free rate and σ_P is the portfolio standard deviation. For the experiments in this chapter, we assume that there is no risk free asset, therefore we only consider the return of the portfolio.

5.4 Experiment Results

In this section, we present the results obtained from repeated portfolio optimization using the covariance estimators described in section 5.3 and the newly proposed Graphical Lasso covariance estimators. The average annualized results over all possible rebalancing periods are presented in the subsequent tables, while the results for each rebalancing period can be found in Appendix F.

5.4.1 Long-Short Portfolio Results

For the long-short portfolio, we expect the Markowitz Direct Optimization method to not perform well in terms of realized risk especially when $T < N$, since this problem

is unconstrained, allowing the portfolio to take on large long and short positions. We expect that the newly proposed Graphical Lasso methods will perform better. Also, it is generally believed that the longer the out of sample period, the less stable the Markowitz Direct Optimization portfolios [50], so we expect the sparse methods to give rise to less risky portfolios for the longer rebalancing periods.

Additionally, we expect that amongst the Graphical Lasso methods that optimize certain portfolio criteria in the validation period (e.g. maximize portfolio return), when compared to other validation methods with the exception of the likelihood methods, each method that optimizes certain portfolio criteria will perform the best compared to the other validation methods that optimize other portfolio criteria.

5.4.1.1 Realized Risk

We look at the average performance over all rebalancing periods for all the methods in literature that we discussed and compare performance to the average performance of the newly proposed Graphical Lasso methods in Table 5.1.

Average S.S	Predicted Risk (%)	std dev	Realized Risk(%)	std dev
Direct Optimization	603.31	28.05	612.16	28.76
SI	5.45	0.28	12.27	0.96
RMT-0	6.12	0.42	11.87	0.95
RMT-M	6.18	0.42	11.77	0.94
Shrinkage_SI	33.57	13.22	32.15	10.22
Shrinkage_Cov	11.46	0.71	12.18	0.85
Shrinkage_Corr	7.61	0.53	12.31	1.00
Naïve	16.02	1.05	15.81	1.10
GLasso$^{\text{MAX RETURN}}$	5.77	0.57	12.31	0.87
GLasso$^{\text{MIN REALIZED RISK}}$	6.24	0.45	11.78	0.91
GLasso$^{\text{MAX SHARPE}}$	5.84	0.56	12.26	0.89
GLasso$^{\text{MAX LIKELIHOOD}}$	6.16	0.52	**11.63**	0.83
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.23	0.52	11.78	0.86

Table 5.1 Long-short portfolio average realized risks (all methods)

For the long-short portfolio problem, the summary table over all rebalancing periods, Table 5.1, shows that on average Markowitz Direct Optimization is not stable and results in portfolios with extremely large realized risks. The proposed sparse Graphical Lasso estimator that maximizes the Gaussian likelihood using 1 period for training and another period for validation (GLasso$^{\text{MAX LIKELIHOOD}}$) performs the best. It gives the least risky portfolios than all other methods. One of the random matrix methods, RMT-M, achieves the second best realized risk, with the other likelihood method (GLasso$^{\text{MAX LIKELIHOOD-2}}$) and the validation method that optimizes for realized risk (GLasso$^{\text{MIN REALIZED RISK}}$) achieving the third best realized risks. We now look at some individual rebalancing period results from Appendix F to see if the results are consistent over time.

3 months S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev
Direct Optimization	1410.62	76.06	1421.58	76.44
SI	5.12	0.24	11.47	0.63
RMT-0	5.27	0.25	11.34	0.63
RMT-M	5.33	0.26	11.27	0.62
Shrinkage_SI	5.35	0.29	11.51	0.57
Shrinkage_Cov	11.73	0.65	12.00	0.60
Shrinkage_Corr	6.97	0.38	11.77	0.65
Naïve	15.67	0.85	15.80	0.85
GLasso$^{\text{MAX RETURN}}$	5.23	0.27	11.88	0.68
GLasso$^{\text{MIN REALIZED RISK}}$	5.55	0.31	11.41	0.58
GLasso$^{\text{MAX SHARPE}}$	5.34	0.27	11.85	0.69
GLasso$^{\text{MAX LIKELIHOOD}}$	5.68	0.31	**11.24**	0.58
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.55	0.27	11.42	0.60

Table 5.2 Long-short portfolio 3 months realized risks

1 year S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev
Direct Optimization	6.97	0.63	13.63	1.25
SI	5.94	0.41	13.16	1.32
RMT-0	7.18	0.67	12.35	1.24
RMT-M	7.24	0.68	12.23	1.23
Shrinkage_SI	7.59	0.70	12.35	1.09
Shrinkage_Cov	10.54	0.91	12.07	1.07
Shrinkage_Corr	8.33	0.81	12.78	1.20
Naïve	16.20	1.43	16.09	1.43
GLasso$^{\text{MAX RETURN}}$	6.63	0.82	12.03	0.98
GLasso$^{\text{MIN REALIZED RISK}}$	6.67	0.62	12.03	1.17
GLasso$^{\text{MAX SHARPE}}$	6.81	0.88	11.96	0.99
GLasso$^{\text{MAX LIKELIHOOD}}$	7.06	0.87	**11.79**	1.11
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.31	0.92	11.92	1.11

Table 5.3 Long-short portfolio 1 year realized risks

2 years S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev
Direct Optimization	9.09	0.68	14.02	2.01
SI	6.71	0.27	14.31	1.58
RMT-0	8.71	0.70	13.63	1.78
RMT-M	8.77	0.69	13.42	1.75
Shrinkage_SI	9.26	0.69	13.54	1.87
Shrinkage_Cov	11.07	0.75	12.96	1.47
Shrinkage_Corr	9.86	0.75	13.99	2.13
Naïve	17.34	1.26	15.76	1.51
GLasso$^{\text{MAX RETURN}}$	8.16	1.18	13.76	1.55
GLasso$^{\text{MIN REALIZED RISK}}$	8.50	0.68	12.89	1.59
GLasso$^{\text{MAX SHARPE}}$	8.10	1.17	13.77	1.55
GLasso$^{\text{MAX LIKELIHOOD}}$	8.41	0.90	**12.40**	1.28
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.79	0.92	12.82	1.49

Table 5.4 Long-short portfolio 2 years realized risks

From Appendix F, Table 5.2, Table 5.3 and Table 5.4, it is evident that for all the methods, realized risks increase as T increases. We also see that the Direct Optimization method performs badly when $T < N$. At 3 months, which is the crossing point when the estimated covariance goes from singular to non-singular, the realized risk of the Direct Optimization method becomes extremely large. It gets even larger as T decreases, as evident in our 1 month and 2 months results in Appendix F. At the lower rebalancing periods, 1 and 2 months, the non-Graphical Lasso methods, with the exception of the Direct Optimization method, perform better than the Graphical Lasso methods. From 3 months and higher, the performance of the Graphical Lasso methods pick up, with the Graphical Lasso likelihood method, GLasso$^{\text{MAX LIKELIHOOD}}$, consistently achieving the best realized risk at every rebalancing period higher than 2 months. These results support our hypothesis where we expect that the sparse

Graphical Lasso methods will give rise to less risky portfolios at longer rebalancing periods, and also have lower realized risks than the Direct Optimization method.

In general, all Graphical Lasso strategies perform better than the Direct Optimization and Naïve methods at all rebalancing periods. Compared to the other covariance estimation methods from literature (excluding Direct Optimization and the Naïve method), all Graphical Lasso strategies perform better than 2 out of 6 of these methods at the 1 month rebalancing period. At the 2 month rebalancing period, the best performing Graphical Lasso strategy, $\text{GLasso}^{\text{MAX LIKELIHOOD}}$, performs better than 4 of the 6 methods from literature. From 3 months all the way to 2 years, $\text{GLasso}^{\text{MAX LIKELIHOOD}}$ performs better than all the 6 methods from literature. These results show that the Graphical Lasso portfolio strategies appear to be very competitive, especially the $\text{GLasso}^{\text{MAX LIKELIHOOD}}$, which is always the best performing Graphical Lasso strategy.

5.4.1.2 Expected Return of the Portfolio

In terms of portfolio return, we expect the sparse estimator which optimizes portfolio return to perform the best amongst the other Graphical Lasso strategies that optimize portfolio criteria.

Average S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev
Direct Optimization	603.31	28.05	612.16	28.76	341.89	172.27
SI	5.45	0.28	12.27	0.96	8.88	3.38
RMT-0	6.12	0.42	11.87	0.95	9.86	3.38
RMT-M	6.18	0.42	11.77	0.94	9.63	3.36
Shrinkage_SI	33.57	13.22	32.15	10.22	25.70	20.65
Shrinkage_Cov	11.46	0.71	12.18	0.85	9.58	3.28
Shrinkage_Corr	7.61	0.53	12.31	1.00	9.43	3.60
Naïve	16.02	1.05	15.81	1.10	8.88	4.03
GLasso$^{\text{MAX RETURN}}$	5.77	0.57	12.31	0.87	8.73	3.13
GLasso$^{\text{MIN REALIZED RISK}}$	6.24	0.45	11.78	0.91	8.96	3.26
GLasso$^{\text{MAX SHARPE}}$	5.84	0.56	12.26	0.89	8.92	3.16
GLasso$^{\text{MAX LIKELIHOOD}}$	6.16	0.52	**11.63**	0.83	8.68	3.05
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.23	0.52	11.78	0.86	8.81	3.17

Table 5.5 Long-short portfolio average portfolio return (all methods)

Amongst the Graphical Lasso methods that optimize portfolio criteria, we expect the one that maximizes portfolio return to have the highest return. However our results show a different performance. The other two Graphical Lasso methods that optimize realized risk and Sharpe ratio perform better. We now look at some individual rebalancing period results from Appendix F to see if the results are consistent over time.

3 months S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev
Direct Optimization	1410.62	76.06	1421.58	76.44	913.54	439.05
SI	5.12	0.24	11.47	0.63	7.16	3.39
RMT-0	5.27	0.25	11.34	0.63	6.24	3.41
RMT-M	5.33	0.26	11.27	0.62	6.23	3.39
Shrinkage_SI	5.35	0.29	11.51	0.57	8.12	3.53
Shrinkage_Cov	11.73	0.65	12.00	0.60	8.84	3.18
Shrinkage_Corr	6.97	0.38	11.77	0.65	6.28	3.41
Naïve	15.67	0.85	15.80	0.85	**10.15**	4.88
GLasso$^{\text{MAX RETURN}}$	5.23	0.27	11.88	0.68	6.56	3.48
GLasso$^{\text{MIN REALIZED RISK}}$	5.55	0.31	11.41	0.58	7.69	3.26
GLasso$^{\text{MAX SHARPE}}$	5.34	0.27	11.85	0.69	6.74	3.52
GLasso$^{\text{MAX LIKELIHOOD}}$	5.68	0.31	**11.24**	0.58	7.15	3.13
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.55	0.27	11.42	0.60	6.80	3.20

Table 5.6 Long-short portfolio 3 months portfolio return

1 year S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev
Direct Optimization	6.97	0.63	13.63	1.25	12.21	3.50
SI	5.94	0.41	13.16	1.32	11.05	3.39
RMT-0	7.18	0.67	12.35	1.24	**15.43**	3.37
RMT-M	7.24	0.68	12.23	1.23	14.89	3.38
Shrinkage_SI	7.59	0.70	12.35	1.09	12.15	3.24
Shrinkage_Cov	10.54	0.91	12.07	1.07	10.66	3.09
Shrinkage_Corr	8.33	0.81	12.78	1.20	12.63	3.45
Naïve	16.20	1.43	16.09	1.43	9.35	4.16
GLasso$^{\text{MAX RETURN}}$	6.63	0.82	12.03	0.98	11.17	2.50
GLasso$^{\text{MIN REALIZED RISK}}$	6.67	0.62	12.03	1.17	10.64	3.31
GLasso$^{\text{MAX SHARPE}}$	6.81	0.88	11.96	0.99	11.63	2.72
GLasso$^{\text{MAX LIKELIHOOD}}$	7.06	0.87	**11.79**	1.11	11.14	3.09
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.31	0.92	11.92	1.11	11.52	3.01

Table 5.7 Long-short portfolio 1 year portfolio return

2 years S.S	Predicted Risk (%)	std dev	Realized Risk (%)	std dev	Portfolio Return (%)	std dev
Direct Optimization	9.09	0.68	14.02	2.01	12.36	3.33
SI	6.71	0.27	14.31	1.58	13.38	2.42
RMT-0	8.71	0.70	13.63	1.78	**15.13**	3.32
RMT-M	8.77	0.69	13.42	1.75	14.54	3.33
Shrinkage_SI	9.26	0.69	13.54	1.87	12.80	3.29
Shrinkage_Cov	11.07	0.75	12.96	1.47	11.74	2.97
Shrinkage_Corr	9.86	0.75	13.99	2.13	12.69	3.55
Naïve	17.34	1.26	15.76	1.51	8.05	3.70
GLasso$^{\text{MAX RETURN}}$	8.16	1.18	13.76	1.55	11.83	3.09
GLasso$^{\text{MIN REALIZED RISK}}$	8.50	0.68	12.89	1.59	12.51	2.92
GLasso$^{\text{MAX SHARPE}}$	8.10	1.17	13.77	1.55	11.99	3.09
GLasso$^{\text{MAX LIKELIHOOD}}$	8.41	0.90	**12.40**	1.28	11.28	2.41
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.79	0.92	12.82	1.49	11.96	2.85

Table 5.8 Long-short portfolio 2 years portfolio return

From Table 5.6, the 3 months result shows the Direct Optimization method giving extremely high portfolio returns due to unstable covariance estimation. At the 1 month and 3 months rebalancing periods, the Naïve method gives the highest portfolio return. At 6 months and 9 months in Appendix F, the Direct Optimization method gives the highest portfolio return. For 1 year and 2 year periods, the random matrix method, RMT-0, gives the highest portfolio return. In general, for every rebalancing period, the non-Graphical Lasso methods perform better than the Graphical Lasso methods and the Naïve method, except at $T < N$, where the Naïve method gives the best portfolio returns.

5.4.1.3 Sharpe Ratio

In terms of Sharpe ratio, we expect the sparse Graphical Lasso estimator which optimizes Sharpe ratio to perform the best amongst the other Graphical Lasso strategies that optimize portfolio criteria.

Average S.S	Predicted Risk (%)	std dev	Realized Risk(%)	std dev	Sharpe Ratio
Direct Optimization	603.31	28.05	612.16	28.76	0.70
SI	5.45	0.28	12.27	0.96	0.68
RMT-0	6.12	0.42	11.87	0.95	0.78
RMT-M	6.18	0.42	11.77	0.94	0.77
Shrinkage_SI	33.57	13.22	32.15	10.22	0.66
Shrinkage_Cov	11.46	0.71	12.18	0.85	0.76
Shrinkage_Corr	7.61	0.53	12.31	1.00	0.72
Naïve	16.02	1.05	15.81	1.10	0.54
GLasso$^{\text{MAX RETURN}}$	5.77	0.57	12.31	0.87	0.96
GLasso$^{\text{MIN REALIZED RISK}}$	6.24	0.45	11.78	0.91	**1.01**
GLasso$^{\text{MAX SHARPE}}$	5.84	0.56	12.26	0.89	0.99
GLasso$^{\text{MAX LIKELIHOOD}}$	6.16	0.52	**11.63**	0.83	1.00
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.23	0.52	11.78	0.86	1.00

Table 5.9 Long-short portfolio average Sharpe ratio (all methods)

Table 5.9 shows that the Graphical Lasso methods appear to give portfolios with higher Sharpe ratios than the non-Graphical Lasso methods. We expect the Graphical Lasso method that optimizes Sharpe ratio to perform best when compared to the other Graphical Lasso methods that optimize portfolio criteria, but results show that the Graphical Lasso strategy that optimizes realized risk performs the best. We now look at individual average Sharpe ratio performance over all rebalancing periods for all the methods in Appendix F.

3 months S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sharpe Ratio
Direct Optimization	1410.62	76.06	1421.58	76.44	0.62
SI	5.12	0.24	11.47	0.63	0.60
RMT-0	5.27	0.25	11.34	0.63	0.53
RMT-M	5.33	0.26	11.27	0.62	0.53
Shrinkage_SI	5.35	0.29	11.51	0.57	0.68
Shrinkage_Cov	11.73	0.65	12.00	0.60	0.71
Shrinkage_Corr	6.97	0.38	11.77	0.65	0.51
Naive	15.67	0.85	15.80	0.85	0.62
GLasso$^{\text{MAX RETURN}}$	5.23	0.27	11.88	0.68	0.95
GLasso$^{\text{MIN REALIZED RISK}}$	5.55	0.31	11.41	0.58	**1.04**
GLasso$^{\text{MAX SHARPE}}$	5.34	0.27	11.85	0.69	1.00
GLasso$^{\text{MAX LIKELIHOOD}}$	5.68	0.31	**11.24**	0.58	1.00
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.55	0.27	11.42	0.60	0.97

Table 5.10 Long-short portfolio 3 months Sharpe ratio

1 year S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sharpe Ratio
Direct Optimization	6.97	0.63	13.63	1.25	0.87
SI	5.94	0.41	13.16	1.32	0.81
RMT-0	7.18	0.67	12.35	1.24	1.20
RMT-M	7.24	0.68	12.23	1.23	1.17
Shrinkage_SI	7.59	0.70	12.35	1.09	0.95
Shrinkage_Cov	10.54	0.91	12.07	1.07	0.86
Shrinkage_Corr	8.33	0.81	12.78	1.20	0.95
Naive	16.20	1.43	16.09	1.43	0.56
GLasso$^{\text{MAX RETURN}}$	6.63	0.82	12.03	0.98	1.01
GLasso$^{\text{MIN REALIZED RISK}}$	6.67	0.62	12.03	1.17	1.04
GLasso$^{\text{MAX SHARPE}}$	6.81	0.88	11.96	0.99	1.08
GLasso$^{\text{MAX LIKELIHOOD}}$	7.06	0.87	**11.79**	1.11	1.08
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.31	0.92	11.92	1.11	**1.11**

Table 5.11 Long-short portfolio 1 year Sharpe ratio

2 years S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sharpe Ratio
Direct Optimization	9.09	0.68	14.02	2.01	0.83
SI	6.71	0.27	14.31	1.58	0.90
RMT-0	8.71	0.70	13.63	1.78	1.05
RMT-M	8.77	0.69	13.42	1.75	1.03
Shrinkage_SI	9.26	0.69	13.54	1.87	0.89
Shrinkage_Cov	11.07	0.75	12.96	1.47	0.87
Shrinkage_Corr	9.86	0.75	13.99	2.13	0.84
Naive	17.34	1.26	15.76	1.51	0.50
GLasso$^{\text{MAX RETURN}}$	8.16	1.18	13.76	1.55	0.93
GLasso$^{\text{MIN REALIZED RISK}}$	8.50	0.68	12.89	1.59	**1.10**
GLasso$^{\text{MAX SHARPE}}$	8.10	1.17	13.77	1.55	0.94
GLasso$^{\text{MAX LIKELIHOOD}}$	8.41	0.90	**12.40**	1.28	1.03
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.79	0.92	12.82	1.49	1.04

Table 5.12 Long-short portfolio 2 years Sharpe ratio

Table 5.10, Table 5.11 and Table 5.12 all show similar results to the average results. Across all rebalancing periods, the Graphical Lasso strategies consistently result in Sharpe ratios > 0.9. It is well known that financial institutions typically want Sharpe ratios > 1 because it is believed that if this is so, an investor is making money most of the time. The Graphical Lasso strategies have many incidents of achieving Sharpe ratios > 1, which are significantly higher than the Sharpe ratios of the non-Graphical Lasso strategies, as evident in Appendix F, Table 5.10, Table 5.11 and Table 5.12. The only times the non-Graphical Lasso strategies achieve Sharpe ratios > 1 are at the 1 year and 2 year periods, when the RMT methods perform well, giving high Sharpe ratios.

5.4.2 Long-Only Portfolio Result

Compared to the long-short portfolio results, our only different expectation is that with the addition of the no short-selling constraint, the performance of the Direct Optimization method should significantly improve because this constraint prevents this method from taking on extreme long and short positions.

5.4.2.1 Realized Risk

We look at the average performance over all rebalancing periods for all the methods in literature that we discussed and compare performance to the average performance of the newly proposed Graphical Lasso methods in Table 5.13

Average N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev
Direct Optimization	8.11	0.59	**11.75**	1.01
SI	7.30	0.46	12.28	0.96
RMT-0	8.11	0.56	12.24	0.99
RMT-M	8.04	0.56	12.20	0.98
Shrinkage_SI	8.31	0.57	12.18	0.98
Shrinkage_Cov	12.32	0.77	12.38	0.89
Shrinkage_Corr	9.59	0.66	12.26	1.05
Naive	16.02	1.05	15.81	1.10
GLasso$^{\text{MAX RETURN}}$	6.89	0.75	12.46	0.82
GLasso$^{\text{MIN REALIZED RISK}}$	7.33	0.53	12.12	0.90
GLasso$^{\text{MAX SHARPE}}$	6.98	0.74	12.59	0.92
GLasso$^{\text{MAX LIKELIHOOD}}$	7.66	0.68	11.94	0.85
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.88	0.69	12.11	0.91

Table 5.13 Long-only portfolio average realized risks (all methods)

Results in Table 5.13 show that the Graphical Lasso methods on average achieve similar realized risks to the non-Graphical Lasso methods excluding the Naïve and Direct Optimization methods. The performance of the Direct Optimization method improves significantly, resulting in the best average realized risk. The second best realized risk is that of the proposed sparse Graphical Lasso estimator that maximizes the Gaussian likelihood using 1 period for training and another period for validation ($GLasso^{MAX\ LIKELIHOOD}$). This result supports our hypothesis which states that we expect the Direct Optimization performance to significantly improve due to the addition of the no short selling constraint. We now look at some individual rebalancing period results from Appendix F to see if the results are consistent over time.

3 months N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev
Direct Optimization	7.47	0.41	12.11	0.65
SI	6.89	0.38	11.60	0.61
RMT-0	7.35	0.41	11.64	0.61
RMT-M	7.31	0.41	11.61	0.61
Shrinkage_SI	7.60	0.41	11.60	0.61
Shrinkage_Cov	12.43	0.68	12.16	0.63
Shrinkage_Corr	9.11	0.50	11.65	0.67
Naïve	15.67	0.85	15.80	0.85
$GLasso^{MAX\ RETURN}$	6.51	0.49	12.25	0.67
$GLasso^{MIN\ REALIZED\ RISK}$	7.30	0.44	11.69	0.63
$GLasso^{MAX\ SHARPE}$	6.71	0.48	12.07	0.68
$GLasso^{MAX\ LIKELIHOOD}$	7.19	0.44	**11.46**	0.61
$GLasso^{MAX\ LIKELIHOOD\text{-}2}$	7.41	0.46	11.66	0.62

Table 5.14 Long-only portfolio 3 months realized risks

1 year N.S.S	Predicted Risk (%)	std dev	Realized Risk(%)	std dev
Direct Optimization	9.46	0.88	12.74	1.18
SI	7.90	0.64	12.94	1.19
RMT-0	9.18	0.84	12.82	1.24
RMT-M	9.08	0.83	12.76	1.23
Shrinkage_SI	9.35	0.85	12.63	1.15
Shrinkage_Cov	11.69	1.01	12.23	1.08
Shrinkage_Corr	10.05	0.98	12.83	1.23
Naïve	16.20	1.43	16.09	1.43
GLasso$^{\text{MAX RETURN}}$	8.02	1.16	12.72	1.05
GLasso$^{\text{MIN REALIZED RISK}}$	7.96	0.93	12.31	1.11
GLasso$^{\text{MAX SHARPE}}$	8.02	1.16	12.80	1.05
GLasso$^{\text{MAX LIKELIHOOD}}$	8.30	1.05	**12.17**	1.09
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.63	1.10	12.35	1.12

Table 5.15 Long-only portfolio 1 year realized risks

2 years N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev
Direct Optimization	11.03	0.79	**8.11**	2.25
SI	9.00	0.51	13.92	1.90
RMT-0	10.67	0.78	14.18	2.14
RMT-M	10.54	0.76	14.07	2.09
Shrinkage_SI	10.90	0.77	14.02	2.18
Shrinkage_Cov	12.30	0.83	13.23	1.63
Shrinkage_Corr	11.43	0.83	14.27	2.37
Naïve	17.34	1.26	15.76	1.51
GLasso$^{\text{MAX RETURN}}$	7.72	1.17	13.07	1.16
GLasso$^{\text{MIN REALIZED RISK}}$	8.64	0.46	13.19	1.64
GLasso$^{\text{MAX SHARPE}}$	8.11	1.18	14.22	1.80
GLasso$^{\text{MAX LIKELIHOOD}}$	9.63	1.03	12.68	1.35
GLasso$^{\text{MAX LIKELIHOOD-2}}$	10.09	1.03	13.35	1.68

Table 5.16 Long-only portfolio 2 years realized risks

Once again, at longer rebalancing periods, the realized risks for all methods except the direct optimization method, increase with T. From Appendix F, it is evident that for $T < N$, the Direct Optimization method has one of the highest realized risks, although significantly better than its performance for the long-short portfolio

problem. All other methods perform essentially the same as they did in the long-short portfolio case. The performance of the Graphical Lasso methods improves with increasing *T*. Amongst all the Graphical Lasso methods, the likelihood method, GLasso$^{\text{MAX LIKELIHOOD}}$, consistently achieves the best realized risk for all rebalancing periods except at 1 month. Compared to all methods (including non-Graphical Lasso methods), this method (GLasso$^{\text{MAX LIKELIHOOD}}$) always has the top 3 best realized risks. At the 1 year and 9 month rebalancing periods, it has the best realized risk compared to all other methods, but at the 2 year period, the Direct Optimization achieves the best performance.

In general, across all rebalancing periods, the Naïve method achieves the worst performance. For the 1 month, 2 months and 3 months rebalancing periods, the Direct Optimization method performs worse than all the Graphical Lasso strategies. From 6 months to 1 year, the Direct Optimization method performs worse than most Graphical Lasso strategies, but at the 2 year period achieves the best overall performance which is expected due to the large sample size and the addition of the no short selling constraint. The methods from literature perform well, though not as good as the Graphical Lasso portfolio strategies. The Graphical Lasso strategy (GLasso$^{\text{MAX LIKELIHOOD}}$) consistently is in the top 2 best performers from 3 months to 2 years. For the 1 month period, this Graphical Lasso strategy does not perform well and most of the methods from literature perform better. At 2 months, GLasso$^{\text{MAX LIKELIHOOD}}$ has the 3$^{\text{rd}}$ best realized risk, after 2 methods from literature.

5.4.2.2 Expected Return of the Portfolio

Like the long-short portfolio results, the non-Graphical Lasso methods achieve higher portfolio returns than the Graphical Lasso methods as seen in Table 5.17.

Average N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev
Direct Optimization	8.11	0.59	**11.75**	1.01	**10.70**	3.67
SI	7.30	0.46	12.28	0.96	10.15	3.53
RMT-0	8.11	0.56	12.24	0.99	10.29	3.57
RMT-M	8.04	0.56	12.20	0.98	10.18	3.54
Shrinkage_SI	8.31	0.57	12.18	0.98	10.50	3.50
Shrinkage_Cov	12.32	0.77	12.38	0.89	9.59	3.39
Shrinkage_Corr	9.59	0.66	12.26	1.05	10.00	3.58
Naive	16.02	1.05	15.81	1.10	8.88	4.03
GLasso$^{\text{MAX RETURN}}$	6.89	0.75	12.46	0.82	9.20	3.20
GLasso$^{\text{MIN REALIZED RISK}}$	7.33	0.53	12.12	0.90	9.18	3.37
GLasso$^{\text{MAX SHARPE}}$	6.98	0.74	12.59	0.92	9.46	3.38
GLasso$^{\text{MAX LIKELIHOOD}}$	7.66	0.68	11.94	0.85	9.09	3.22
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.88	0.69	12.11	0.91	9.45	3.33

Table 5.17 Long-only portfolio average portfolio return (all methods)

The results in Table 5.17 show that the non-Graphical Lasso methods have higher portfolio returns than the Graphical Lasso methods. Amongst the Graphical Lasso methods that optimize portfolio criteria, we expect the one that maximizes return to have the highest return, but this is not the case. Like the long-short portfolio results, the Graphical Lasso method that optimizes Sharpe ratio performs best. We now look at some individual rebalancing period results from Appendix F to see if the results are consistent over time.

3 months N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev
Direct Optimization	7.47	0.41	12.11	0.65	9.88	3.57
SI	6.89	0.38	11.60	0.61	9.15	3.54
RMT-0	7.35	0.41	11.64	0.61	8.18	3.63
RMT-M	7.31	0.41	11.61	0.61	8.12	3.61
Shrinkage_SI	7.60	0.41	11.60	0.61	9.15	3.40
Shrinkage_Cov	12.43	0.68	12.16	0.63	9.82	3.51
Shrinkage_Corr	9.11	0.50	11.65	0.67	7.45	3.18
Naive	15.67	0.85	15.80	0.85	**10.15**	4.88
GLasso$^{\text{MAX RETURN}}$	6.51	0.49	12.25	0.67	8.23	3.41
GLasso$^{\text{MIN REALIZED RISK}}$	7.30	0.44	11.69	0.63	7.47	3.47
GLasso$^{\text{MAX SHARPE}}$	6.71	0.48	12.07	0.68	7.64	3.43
GLasso$^{\text{MAX LIKELIHOOD}}$	7.19	0.44	**11.46**	0.61	7.62	3.39
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.41	0.46	11.66	0.62	8.11	3.31

Table 5.18 Long-only portfolio 3 months portfolio return

1 year N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev
Direct Optimization	9.46	0.88	12.74	1.18	13.33	3.89
SI	7.90	0.64	12.94	1.19	11.88	3.80
RMT-0	9.18	0.84	12.82	1.24	**13.54**	4.01
RMT-M	9.08	0.83	12.76	1.23	13.31	3.97
Shrinkage_SI	9.35	0.85	12.63	1.15	12.99	3.83
Shrinkage_Cov	11.69	1.01	12.23	1.08	10.70	3.34
Shrinkage_Corr	10.05	0.98	12.83	1.23	13.50	4.08
Naive	16.20	1.43	16.09	1.43	9.35	4.16
GLasso$^{\text{MAX RETURN}}$	8.02	1.16	12.72	1.05	12.60	3.51
GLasso$^{\text{MIN REALIZED RISK}}$	7.96	0.93	12.31	1.11	11.19	3.69
GLasso$^{\text{MAX SHARPE}}$	8.02	1.16	12.80	1.05	12.76	3.54
GLasso$^{\text{MAX LIKELIHOOD}}$	8.30	1.05	**12.17**	1.09	11.05	3.55
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.63	1.10	12.35	1.12	11.69	3.64

Table 5.19 Long-only portfolio 1 year portfolio return

2 years N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev
Direct Optimization	11.03	0.79	**8.11**	2.25	13.12	4.23
SI	9.00	0.51	13.92	1.90	13.12	3.80
RMT-0	10.67	0.78	14.18	2.14	**13.61**	4.29
RMT-M	10.54	0.76	14.07	2.09	13.56	4.22
Shrinkage_SI	10.90	0.77	14.02	2.18	13.35	4.11
Shrinkage_Cov	12.30	0.83	13.23	1.63	12.22	3.46
Shrinkage_Corr	11.43	0.83	14.27	2.37	13.25	4.35
Naïve	17.34	1.26	15.76	1.51	8.05	3.70
GLasso$^{\text{MAX RETURN}}$	7.72	1.17	13.07	1.16	10.04	2.56
GLasso$^{\text{MIN REALIZED RISK}}$	8.64	0.46	13.19	1.64	12.43	3.50
GLasso$^{\text{MAX SHARPE}}$	8.11	1.18	14.22	1.80	12.10	3.92
GLasso$^{\text{MAX LIKELIHOOD}}$	9.63	1.03	12.68	1.35	11.47	2.92
GLasso$^{\text{MAX LIKELIHOOD-2}}$	10.09	1.03	13.35	1.68	12.47	3.61

Table 5.20 Long-only portfolio 2 years portfolio return

Appendix F, Table 5.18, Table 5.19 and Table 5.20 show that at the individual rebalancing periods, we achieve the same performance as the average results, with the non-Graphical Lasso methods having higher portfolio returns than the Graphical Lasso methods.

5.4.2.3 Sharpe Ratio

Like the long-short portfolio results, the Graphical Lasso methods achieve higher Sharpe ratios than the non-Graphical Lasso methods as seen in Table 5.21.

Average N.S.S	Pred Risk(%)	std dev	Realized Risk(%)	std dev	Sharpe Ratio
Direct Optimization	8.11	0.59	**11.75**	1.01	0.81
SI	7.30	0.46	12.28	0.96	0.79
RMT-0	8.11	0.56	12.24	0.99	0.80
RMT-M	8.04	0.56	12.20	0.98	0.79
Shrinkage_SI	8.31	0.57	12.18	0.98	0.82
Shrinkage_Cov	12.32	0.77	12.38	0.89	0.75
Shrinkage_Corr	9.59	0.66	12.26	1.05	0.77
Naive	16.02	1.05	15.81	1.10	0.54
GLasso$^{\text{MAX RETURN}}$	6.89	0.75	12.46	0.82	0.99
GLasso$^{\text{MIN REALIZED RISK}}$	7.33	0.53	12.12	0.90	1.01
GLasso$^{\text{MAX SHARPE}}$	6.98	0.74	12.59	0.92	0.99
GLasso$^{\text{MAX LIKELIHOOD}}$	7.66	0.68	11.94	0.85	1.02
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.88	0.69	12.11	0.91	**1.03**

Table 5.21 Long-only portfolio average Sharpe ratio

We now look at individual average Sharpe ratio performance over all rebalancing periods for all the methods in Appendix F.

3 months N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sharpe Ratio
Direct Optimization	7.47	0.41	12.11	0.65	0.78
SI	6.89	0.38	11.60	0.61	0.76
RMT-0	7.35	0.41	11.64	0.61	0.67
RMT-M	7.31	0.41	11.61	0.61	0.67
Shrinkage_SI	7.60	0.41	11.60	0.61	0.76
Shrinkage_Cov	12.43	0.68	12.16	0.63	0.78
Shrinkage_Corr	9.11	0.50	11.65	0.67	0.61
Naive	15.67	0.85	15.80	0.85	0.62
GLasso$^{\text{MAX RETURN}}$	6.51	0.49	12.25	0.67	**1.06**
GLasso$^{\text{MIN REALIZED RISK}}$	7.30	0.44	11.69	0.63	1.01
GLasso$^{\text{MAX SHARPE}}$	6.71	0.48	12.07	0.68	1.02
GLasso$^{\text{MAX LIKELIHOOD}}$	7.19	0.44	**11.46**	0.61	1.01
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.41	0.46	11.66	0.62	1.02

Table 5.22 Long-only portfolio 3 months Sharpe ratio

1 year N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sharpe Ratio
Direct Optimization	9.46	0.88	12.74	1.18	1.01
SI	7.90	0.64	12.94	1.19	0.89
RMT-0	9.18	0.84	12.82	1.24	1.02
RMT-M	9.08	0.83	12.76	1.23	1.01
Shrinkage_SI	9.35	0.85	12.63	1.15	1.00
Shrinkage_Cov	11.69	1.01	12.23	1.08	0.85
Shrinkage_Corr	10.05	0.98	12.83	1.23	1.02
Naive	16.20	1.43	16.09	1.43	0.56
GLasso$^{\text{MAX RETURN}}$	8.02	1.16	12.72	1.05	1.08
GLasso$^{\text{MIN REALIZED RISK}}$	7.96	0.93	12.31	1.11	1.03
GLasso$^{\text{MAX SHARPE}}$	8.02	1.16	12.80	1.05	**1.09**
GLasso$^{\text{MAX LIKELIHOOD}}$	8.30	1.05	**12.17**	1.09	1.04
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.63	1.10	12.35	1.12	1.07

Table 5.23 Long-only portfolio 1 year Sharpe ratio

2 years N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Sharpe Ratio
Direct Optimization	11.03	0.79	**8.11**	2.25	0.85
SI	9.00	0.51	13.92	1.90	0.89
RMT-0	10.67	0.78	14.18	2.14	0.89
RMT-M	10.54	0.76	14.07	2.09	0.90
Shrinkage_SI	10.90	0.77	14.02	2.18	0.88
Shrinkage_Cov	12.30	0.83	13.23	1.63	0.88
Shrinkage_Corr	11.43	0.83	14.27	2.37	0.85
Naive	17.34	1.26	15.76	1.51	0.50
GLasso$^{\text{MAX RETURN}}$	7.72	1.17	13.07	1.16	0.86
GLasso$^{\text{MIN REALIZED RISK}}$	8.64	0.46	13.19	1.64	**1.04**
GLasso$^{\text{MAX SHARPE}}$	8.11	1.18	14.22	1.80	0.89
GLasso$^{\text{MAX LIKELIHOOD}}$	9.63	1.03	12.68	1.35	1.02
GLasso$^{\text{MAX LIKELIHOOD-2}}$	10.09	1.03	13.35	1.68	1.03

Table 5.24 Long-only portfolio 2 years Sharpe ratio

Appendix F, Table 5.22 Table 5.23 and Table 5.24 show that at the individual rebalancing periods, we achieve the same performance as the average results, with the Graphical Lasso methods consistently achieving higher Sharpe ratios than the

non-Graphical Lasso methods. Amongst the Graphical Lasso methods that optimize portfolio criteria, we expect the one that maximizes Sharpe ratio to perform the best when compared to the other methods that optimize portfolio criteria, but this is only the case at some rebalancing periods e.g. 1 year.

5.5 Summary

In this chapter, we compared the performance of existing covariance estimators for portfolio optimization with the newly proposed Graphical Lasso methods on stock market data. We evaluated portfolio performance across different rebalancing periods with and without short selling constraints.

For both the long-short and long-only portfolios, our results showed that in general for all the methods, realized risks got worse, increasing as T increased, with the exception of the Direct Optimization method, which performed very well at 2 years in the long-only case . The Direct Optimization method performed badly when $T < N$, as expected, because the estimated covariance is singular at this point, causing it to take on extreme long and short positions, with the realized risks becoming worse the smaller $\frac{T}{N}$ was.

For the long-short portfolio case, across all rebalancing periods, all Graphical Lasso strategies always performed better than the Direct Optimization and Naïve methods in terms of realized risk. In general, GLasso$^{\text{MAX LIKELIHOOD}}$ always achieved the best realized risk amongst all the other Graphical Lasso strategies except at the 1 month period. When $T < N$, the Graphical Lasso strategy, GLasso$^{\text{MAX LIKELIHOOD}}$, performed

better than most of the methods from literature. When $T > N$, the performance of the Graphical Lasso methods picked up, with the Graphical Lasso likelihood method, GLasso$^{\text{MAX LIKELIHOOD}}$, consistently achieving the best realized risk compared to all the other methods (Naïve method, Direct Optimization method, methods from literature and the other Graphical Lasso methods).

For the long-only case, GLasso$^{\text{MAX LIKELIHOOD}}$ always achieved the best realized risks amongst all the other Graphical Lasso methods except at the 1 month period, and the top 3 best realized risks compared to all the methods (non-Graphical Lasso methods included) for all rebalancing periods, except at 1 month. The Graphical Lasso strategies always performed better than the Direct Optimization and Naïve methods except at the 2 year period, when the Direct Optimization method performed very well, achieving the lowest realized risk. When $< N$, the Graphical Lasso strategies always performed better than the Direct Optimization and Naïve methods. Also, at this point ($T < N$), the methods from literature had similar realized risk performances as the Graphical Lasso strategies, with the winning method varying depending on the rebalancing period length. When $T > N$, the Graphical Lasso strategies always performed better than the Direct Optimization and Naïve methods except at the 2 year period, where the Direct Optimization method had the best realized risk. At this point, ($T > N$), the methods from literature performed well, but the Graphical Lasso strategy (GLasso$^{\text{MAX LIKELIHOOD}}$) consistently performed better and was in the top 2 best performers from 6 months to 2 years.

In terms of portfolio return and Sharpe ratio, for both the long-short and long-only portfolio cases, the Graphical Lasso methods always had the best Sharpe ratios, while

the non-Graphical Lasso methods always had the best portfolio returns. The addition of the no short selling constraint improved the Direct Optimization method significantly in terms of realized risks, with this method achieving the best realized risk at the 2 year rebalancing period. Even with this improvement of the Direct Optimization realized risks, the Graphical Lasso strategies still performed better, except at the 2 year rebalancing period. The performance of all the other methods did not seem to be affected by the addition of the no short selling constraint.

These results show that using likelihood appears to be a very promising method for choosing the optimal regularization for Graphical Lasso application in Markowitz portfolio optimization. It has also been shown in experiment results that the proposed sparse Graphical Lasso strategies give rise to portfolios that are less risky, with higher Sharpe ratios than existing methods in literature.

Chapter 6

Conclusion and Future work

6.1 Conclusion

In this research, we introduced the use of modern machine learning methods for Markowitz minimum variance portfolio optimization. We began by identifying the covariance estimation problem that exists in real world financial and bioinformatics applications where we mostly deal with high dimensional data. We addressed these issues by introducing the use of sparse inverse covariance estimation with the use of existing methodology known as '*Graphical Lasso*'. One important prerequisite of Graphical Lasso is to choose the optimal regularization, which we addressed by using several validation techniques.

Initial synthetic data experiments showed the tendency of Graphical Lasso to over-estimate the diagonal elements of the estimated inverse covariance, while shrinking the off-diagonal elements with increasing regularization. To remedy this issue, we introduced a new way of setting the regularization by having two different regularizations for the diagonal and off diagonal elements of the estimated inverse covariance. This was different from the original Graphical Lasso methodology in literature which typically uses a single regularization for the estimated inverse covariance. We called our new methodology the '*Modified Graphical Lasso*' and performed experiments on synthetic data generated from known sparse and dense

inverse covariance models. The experiments used 5-fold cross-validation to estimate the inverse covariance on the data, and we quantified the difference between the estimated and actual inverse covariance using the Frobenius norm. Results showed that the Modified Graphical Lasso performed at least as well as the original Graphical Lasso in the sparse and dense scenarios, across various variable ($p > 10$) and sample sizes. Despite this performance, we chose to use the original Graphical Lasso for the bioinformatics and finance experiments in this thesis due to the high computational time requirements of the Modified Graphical Lasso.

Next, we applied Graphical Lasso to the bioinformatics problem of identifying tissue samples given gene microarray data. When Graphical Lasso was used to generate graphical Gaussian models to perform the supervised learning task of identifying tissue samples, the estimated precisions performed at least as well as the empirical covariance in some instances, and in other instances, the covariance performed better. In general there was no consistency in classification performance. When we randomly replaced the genes in the graphical models for the estimated precisions and empirical covariance, we noticed a significant decrease in performance for only the empirical covariance. From these results, we concluded that the empirical covariance appeared to be the only one performing gene selection in a biologically meaningful way. In that respect, Graphical Lasso did not perform as expected.

Moving on to financial applications which is central to this thesis, we first performed experiments on synthetic stock market data where the true inverse covariance was known. We created portfolio strategies by using Graphical Lasso to optimize portfolio criteria (realized risk) and Gaussian likelihood and performed repeated Markowitz

global minimum variance portfolio optimization with and without short-selling constraints. Our results showed that the Graphical Lasso likelihood method performed very well for both the long-short and long-only scenarios. The likelihood method that uses 2 periods to train, consistently performed at least as good as the oracle method and in some instances, better than the oracle method that uses *a priori* knowledge of the true sparsity level of the inverse covariance used to generate the data.

Lastly, we compared the newly developed Graphical Lasso portfolio strategies to existing covariance estimators used in literature for portfolio optimization. Results showed that in the long-short portfolio case, the Graphical Lasso likelihood method that uses 1 period to train and the other to validate, consistently had the best realized risks compared to all other Graphical Lasso portfolio strategies. In general, the Direct Optimization method which uses the empirical covariance matrix as well as the Moore-Penrose pseudoinverse (when $T < N$) performed badly from 3 months and lower, which is when $\frac{T}{N} < 1$. This performance was expected due to the non-singularity of the empirical covariance, causing the Direct Optimization portfolios to take extreme long and short positions. Also, the addition of the no short-selling constraint in the long-only portfolio case, improved the Direct Optimization method significantly, though all other methods did not show any significant improvement.

Experimental results showed that the Graphical Lasso methods gave rise to less risky portfolios when $T < N$ in both the long-short and long-only portfolio cases than the Direct Optimization and Naïve methods, though not always better than the other methods from literature. The Graphical Lasso methods always performed better than

the methods from literature in general when $T > N$. When $T < N$, for the long-short portfolio case, Graphical Lasso strategies performed better than the methods from literature in general. For the long-only case, the performance of the methods from literature were more competitive with the Graphical Lasso strategies, achieving comparable realized risks, and the winning performance varied depending on the rebalancing period length. On average, for both the long-short and long-only scenarios, the Graphical Lasso methods always had significantly higher Sharpe ratios than the non-Graphical Lasso methods, while the non-Graphical Lasso methods always had higher portfolio returns than the Graphical Lasso methods. These results were consistent across all rebalancing periods.

Experimental results have shown that the proposed sparse Graphical Lasso strategies have lower realized risks in general than all other methods and higher Sharpe ratios because there is less variance in the portfolio returns of these Graphical Lasso strategies. The implication for an investor is that these Graphical Lasso strategies (the winning likelihood method in particular) appear to give more stable returns (higher Sharpe ratios and lower realized risks). The proposed Graphical Lasso strategies may achieve less portfolio returns compared to the other strategies (Direct Optimization, Naïve and methods from literature), as evident in the experiment results in chapter 6, but their returns have been shown to be more stable. This implies that the Graphical Lasso strategies, particularly the one that optimizes Gaussian likelihood, have less volatility than the other methods, which is a desirable quality for an investor whose ultimate goal is to achieve as much return as possible for the lowest level of risk. It has been mentioned earlier that financial institutions

aim for Sharpe ratios > 1, and results in chapter 6 showed that the proposed sparse Graphical Lasso strategies consistently achieved this compared to the other non-Graphical Lasso methods.

It is well known that a lot of hedge funds are long-only funds, despite the fact that the idea of a hedge fund is to hedge risk by taking on both long and short positions. These long-only hedge funds only buy with the hope that the market will continue to go up. There is more risk involved since short positions are not taken, but the reward is known to be more. In general, for such a hedge fund wanting to invest in long-only portfolios, the Graphical Lasso strategies have been shown to be an attractive option based on the positive long-only results from chapter 6 (portfolios with lower risks and higher Sharpe ratios than other methods).

In conclusion, Graphical Lasso as a sparse inverse covariance estimator performed well when used for Markowitz global minimum variance portfolio optimization, especially with regards to realized risk and Sharpe ratio. Using Gaussian likelihood has been shown to be a good way to choose the optimal regularization, as this method always performed the best compared to all other Graphical Lasso methods, and also other methods in literature.

6.2 Future work

Extensions of the proposed Graphical Lasso portfolio methods include the use of *a priori* sparsity structure information in addition to sparsity level information in order to gain better insight as to how the accuracy of the estimation of the true inverse

covariance used to generate the data affects portfolio performance. In chapter 5, we created an oracle method that uses *a priori* sparsity level knowledge to estimate the optimal regularization for the Graphical Lasso estimated inverse covariance. Future work should extend this method to include the sparsity structure also (the exact zero positions in the inverse covariance).

Concerning the existing shrinkage methods from literature, an analytical formula was used to calculate the optimal shrinkage intensity. For a fairer comparison, in the future, similar validation techniques such as that used for Graphical Lasso optimizing portfolio criteria or Gaussian likelihood, can be used to set the optimal shrinkage intensity.

Lastly, the global minimum variance portfolio with and without short-selling constraints have been the main focus of this research. Future work should consider the addition of other constraints such as budget constraints and investment restrictions in order to see how the newly proposed methods perform in these situations. Also, it will be interesting to consider the Markowitz mean-variance portfolio optimization problem which involves the estimation of the mean return to further explore the practical usefulness of the suggested sparse approach for estimating the covariance.

Appendix A

Mathematical Background

This appendix provides mathematical background knowledge necessary for this thesis.

A.1 The Multivariate Gaussian distribution

The Gaussian, also known as the normal distribution, is a widely used model for the distribution of continuous variables. In the case of a single variable x, the Gaussian distribution can be written in the form [17]

$$N(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}}\exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}, \tag{A.1}$$

where μ is the mean and σ^2 is the variance. For a p-dimensional vector $\boldsymbol{x}$, the multivariate Gaussian distribution takes the form

$$N(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}, \tag{A.2}$$

where $\boldsymbol{\mu}$ is a p-dimensional mean vector, $\boldsymbol{\Sigma}$ is a $p \times p$ covariance matrix, and $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$.

Appendix A

A.1.1 Maximum Likelihood for the Gaussian

Given a data set $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^T$ in which the observations $\{\boldsymbol{x}_N\}$ are assumed to be drawn independently from a multivariate Gaussian distribution, we can estimate the parameters of the distribution by maximum likelihood (ML). Maximum likelihood makes the assumption that the most reasonable values of the parameters are those for which the probability of the observed samples, $\boldsymbol{X}$, are largest. This derivation follows very closely to that found in [72]. Since the data set $\boldsymbol{X}$ are independently and identically distributed (i.i.d), the probability of observing the data set is the product of the probabilities of each sample point, and given by:

$$p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} N(x_n|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{A.3}$$

We take the logarithm of the likelihood function yielding the log-likelihood function, as this will simplify the analysis. Since the argument that maximizes a function $f(x)$ is equal to the argument that maximizes $\ln f(x)$, the maximum likelihood solution is unaffected by taking the logarithm. The log-likelihood of the Gaussian is given by

$$\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \ln \prod_{n=1}^{N} N(\boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$= \ln \prod_{n=1}^{N} \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right\}$$

$$= -\frac{Np}{2} ln(2\pi) - \frac{N}{2} \ln|\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}) \quad \text{(A.4)}$$

By a simple rearrangement, we can see that the likelihood function depends on the data set only through two quantities.

Using the derivative of the log likelihood with respect to $\boldsymbol{\mu}$ and setting this derivative to zero, we obtain the solution for the maximum likelihood estimate of the mean given by

$$\sum_{n=1}^{N} \boldsymbol{x}_n, \qquad \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^T \quad \text{(A.5)}$$

These are known as the *sufficient statistics* for the Gaussian distribution.

We consider the following matrix derivative rule given a vector $\boldsymbol{a}$ and a scalar x:

$$\frac{\partial}{\partial \boldsymbol{x}} (x^T \boldsymbol{a}) = \frac{\partial}{\partial \boldsymbol{x}} (\boldsymbol{a}^T x) = \boldsymbol{a} \quad \text{(A.6)}$$

Following (A.6), the derivative of the log-likelihood with respect to $\boldsymbol{\mu}$ is given by

$$\frac{\partial}{\partial \boldsymbol{\mu}} \ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x_n} - \boldsymbol{\mu}) \quad \text{(A.7)}$$

And setting this derivative to zero, we obtain the solution for the maximum likelihood estimate of the mean given by

$$\widehat{\boldsymbol{\mu}}_{ML} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \tag{A.8}$$

(A.8) is the mean of the observed set of data points. The maximization of (A.4) with respect to $\boldsymbol{\Sigma}$ is given by

$$\widehat{\boldsymbol{\Sigma}}_{ML} = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x_n} - \boldsymbol{\mu}_{ML})(\boldsymbol{x_n} - \boldsymbol{\mu}_{ML})^T \tag{A.9}$$

(A.9) is simply the sample covariance matrix and involves $\boldsymbol{\mu}_{ML}$ because this is the result of a joint maximization with respect to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. If we evaluate the expectations of the maximum likelihood solutions under the true distribution, we obtain the following results [72]

$$E[\widehat{\boldsymbol{\mu}}_{ML}] = \boldsymbol{\mu} \tag{A.10}$$

$$E\left[\widehat{\boldsymbol{\Sigma}}_{ML}\right] = \frac{N-1}{N}\boldsymbol{\Sigma} \tag{A.11}$$

We can see that the expectation of the maximum likelihood estimate for the mean is equal to the true mean. However, the maximum likelihood estimate for the covariance

has an expectation that is less than the true value, hence it is biased. This bias can be corrected by defining a different estimator $\tilde{\Sigma}$ given by [72]

$$\tilde{\Sigma} = \frac{1}{N-1}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu}_{ML})(\boldsymbol{x}_n - \boldsymbol{\mu}_{ML})^T \quad \text{(A.12)}$$

From (A.9) and (A.11), the expectation of $\widetilde{\boldsymbol{\Sigma}}$ is equal to $\boldsymbol{\Sigma}$.

A.2 Introduction to Machine Learning

We are in an era of big data. For example, there are about 1 trillion web pages[2]; one hour of video is uploaded to YouTube every second, amounting to 10 years worth of content every day[3]. This deluge of data calls for automated methods of data analysis, which is what machine learning provides. Machine learning is defined as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty [73]. Machine learning is programming computers to optimize a performance criterion using example data or past experience. We have a model defined up to some parameters, and learning is the execution of a computer program to optimize the parameters of the model using training data or past experience [74]. The model may be *predictive* to make predictions in the future, or *descriptive* to gain knowledge from data, or both [74].

[2] http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html

[3] Source: http://www.youtube.com/t/press_statistics.

Appendix A

Machine learning uses the theory of statistics in building mathematical models, because the core task is making inference from a sample. The role of computer science is twofold: First, in training, we need efficient algorithms to solve the optimization problem, as well as to store and process the massive amount of data we generally have [74]. Second, once a model is learned, its representation and algorithmic solution for inference needs to be efficient as well [74]. In certain applications, the efficiency of the learning or inference algorithm, namely, its space and time complexity, may be as important as its predictive accuracy [74].

A.2.1 Types of Machine Learning

Machine learning is usually divided into two main types. In the *predictive* or *supervised learning* approach, the goal is to learn a mapping from inputs x to outputs y, given a labelled set of input-output pairs $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$. Here $\mathcal{D}$ is called the training set, and N is the number of training samples [73].

In the simplest setting, each training input $\boldsymbol{x}_i$ is a D-dimensional vector of numbers representing, say, the heights and weight of a person [73]. These are called *features*, *attributes* or *covariates*. In general, however, $\boldsymbol{x}_i$ could be a complex structured object, such as an image, a sentence, an email message, a time series, a graph etc. Similarly, the form of the output or response variable can in principle be anything, but most methods assume that y_i is a *categorical* or *nominal* variable for some finite set, $y_i \in \{1, \ldots, C\}$ (such as male or female), or that y_i is a real-valued scalar (such as income

level). When y_i is categorical, the problem is known as *classification*, and when y_i is real-valued, the problem is known as *regression*.

The second main type of machine learning is the *descriptive* or *unsupervised learning* approach [73]. Here we are only given inputs, $\mathcal{D} = \{(\boldsymbol{x}_i)\}_{i=1}^{N}$, and the goal is to find "interesting patterns" in the data. This is a much less well-defined problem, since we are not told what kinds of patters to look for, and there is no obvious error metric to use (unlike supervised learning, where we can compare our precision of y for a given $\boldsymbol{x}$ to the observed value).

A.3 Supervised Learning

We introduce supervised learning which is the form of machine learning most widely used in practice [73] and used throughout this thesis. Firstly, we introduce the linear regression model which is linked to the graphical Gaussian models used in chapter 3 that will be useful when we introduce the tissue classification problem using gene microarray data. In particular, we use validation techniques, which will be introduced subsequently, when selecting the correct regularization amounts for the inverse covariance matrix estimation.

A.3.1 Classification

The goal in classification is to learn a mapping from inputs $\boldsymbol{x}$ to outputs y, where $y \in \{1, \ldots, C\}$, with C being the number of classes. If $C = 2$, this is called *binary*

classification (in which case we assume $y \in \{0,1\}$ or alternatively $y \in \{+1,-1\}$. If $C > 2$, this is called *multiclass classification*. If the class labels are not mutually exclusive (e.g., somebody may be classified as tall and strong), we call it *multi-label classification*, but this is best viewed as predicting multiple related binary class labels (a so-called multiple output model). In this thesis, we only focus on binary classification.

One way to formalize the problem is as a *function approximation*. We assume that $y = f(\boldsymbol{x})$ for some unknown function f, and the goal of learning is to estimate the function f given a labelled training set, and then to make predictions using $\hat{y} = \hat{f}(\boldsymbol{x})$. We use the hat symbol to denote an estimate). Our main goal is to make predictions on novel inputs, meaning ones we have not seen before (this is called *generalization*), since predicting the response on the training set is easy.

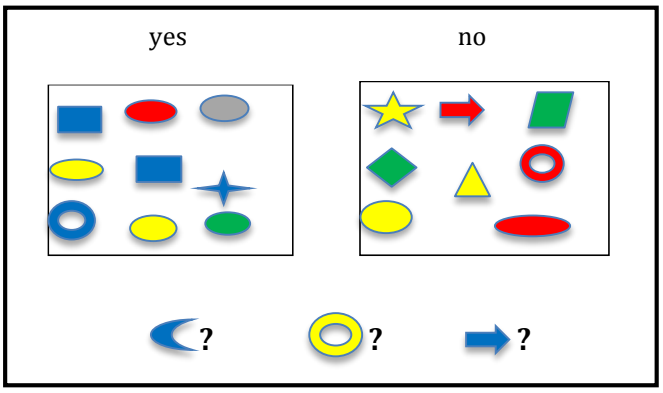

Figure A.1 Some labelled training examples of coloured shapes, along with 3 unlabelled test cases. Figure taken from [73].

Appendix A

D features (attributes)

	Color	Shape	Size (cm)	Label
N cases	Blue	Square	10	1
	Red	Ellipse	2.4	1
	Red	Ellipse	20.7	0

Figure A.2 Representing the training data as an $N \times D$ design matrix. Row i represents the feature vector $\boldsymbol{x}_i$. The last column is the label $y_i \in \{0,1\}$. Figure taken from [73].

Consider the problem illustrated in Figure A.1. We have two classes of object which correspond to labels 0 and 1. The inputs are coloured shapes. These have been described by a set of D features or attributes, which are stored in an $N \times D$ design matrix $\boldsymbol{X}$, shown in Figure A.2. The input features $\boldsymbol{x}$ can be discrete, continuous or a combination of the two. In addition to the inputs, we have a vector of training labels $\boldsymbol{y}$.

In Figure A.1, the test cases are a blue crescent, a yellow circle and a blue arrow. None of these have been seen before. Thus we are required to generalize beyond the training set. A reasonable guess is that the blue crescent should be $y = 1$, since all blue shapes are labeled 1 in the training set. The yellow circle is harder to classify, since some yellow things are labelled $y = 1$ and some are labelled $y = 0$, and some circles are labelled $y = 1$ and some $y = 0$. Consequently, it is not clear what the right

label should be in the case of the yellow circle. Similarly, the correct label for the blue arrow is unclear.

A.3.2 Regression

Regression is just like classification except the response variable is continuous. Figure A.3 shows a simple example: we have a single real-valued input $x_i \in \mathbb{R}$. We consider a straight line to the data. Various extensions of this basic problem can arise, such as having high-dimensional inputs, outliers, etc. Some examples of real world regression problems are:

- Predict tomorrow's stock market price given current market conditions and other possible side information.
- Predict the age of a viewer watching a given video on YouTube.
- Predict the temperature at any location inside a building using weather data, time, door, sensors, etc.

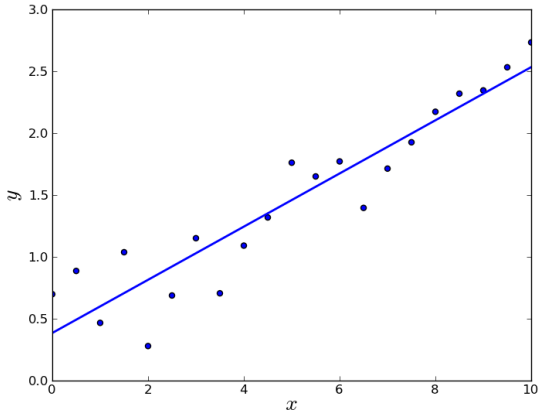

Figure A.3 Linear regression on some data.

Appendix A

A.4 Some Basic Concepts in Machine Learning

We provide an introduction to some key concepts in machine learning.

A.4.1 Parametric vs Non-parametric Models

In this thesis, we focus on probabilistic models of the form $p(y|\boldsymbol{x})$ (supervised learning). There are a number of ways to define such models, but the most important distinction is this: does the model have a fixed number of parameters, or does the number of parameters grow with the amount of training data? The former is called a parametric mode, and the latter is called a non-parametric model [73]. Parametric models such as linear regression have the advantage of often being faster to use, but the disadvantage of making stronger assumptions about the nature of the data distributions [73]. Non-parametric models are more flexible, but often computationally intractable for large datasets [73].

A.4.2 A Simple Non-parametric Classifier: *k*-Nearest Neighbours

A simple example of a non-parametric classifier is the k-Nearest Neighbour (k-NN) classifier. This simply "looks at" the k points in the training set that are nearest to the test input $\boldsymbol{x}$, and accounts how many members of each class are in this set, and returns that empirical fraction as the estimate. More formally [73],

$$p(y = c|\boldsymbol{x}, \mathcal{D}, k) = \frac{1}{k}\sum_{i \in N_k(\boldsymbol{x}, \mathcal{D})} \mathbb{I}(y_i = c) \tag{A.13}$$

where $N_k(\boldsymbol{x}, \mathcal{D})$ are the (indices of the) k nearest points to $\boldsymbol{x}$ in $\mathcal{D}$ and $\mathbb{I}(e)$ is the *indicator function* defines as follows:

$$\mathbb{I}(e) = \begin{cases} 1 & if\ e\ is\ true \\ 0 & if\ e\ is\ false \end{cases}$$

This most common distance metric to use is Euclidean distance (which limits the applicability of the technique to real-valued data), although other metrics can be used [73]. The Euclidean distance between two samples $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$ is given by:

$$\|\boldsymbol{x}_j - \boldsymbol{x}_k\| = \left(\sum_{l=1}^{p} (\boldsymbol{x}_{lj} - \boldsymbol{x}_{lk})^2\right)^{1/2}, \tag{A.14}$$

where p is the dimension of the data.

x_1	x_2	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}	h_{13}	h_{14}	h_{15}	h_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Table A.1 With two inputs, there are four possible cases and sixteen possible Boolean functions. Taken from [74].

A.4.3 Model Selection and Generalization

We begin with the case of learning a Boolean function from examples [74]. In a Boolean function, all inputs and output are binary. There are 2^d possible ways to write d binary values and therefore, with d inputs, the training set has at most 2^d Examples. As shown in Table A.1, each of these can be labeled 0 or 1, and therefore there are 2^d possible Boolean functions of d inputs.

Each distinct training example removes half the hypotheses, namely, those whose guesses are wrong. For example let us say we have $x_1 = 0, x_2 = 1$ and the ouput is 0; this removes $h_5, h_6, h_7, h_8, h_{13}, h_{14}, h_{15}, h_{16}$. This is one way to interpret learning: we start with all possible hypotheses and as we seed more training examples, we remove those hypothesis that are not consistent with the training data. In the case of a Boolean function, to end up with a single hypothesis, we need to see all 2^d training examples. If the training set we are given contains only a small subset of all possible instances, as it generally does, i.e. if we know what the output should be for only a small percentage of the cases, the solution is not unique. After seeing N example cases, there remains 2^{d-N} possible functions. This is an example of an *ill-posed problem* where the data by itself is not sufficient to find a unique solution.

The same problem also exists in other learning applications, in classification, and in regression. As we see more training examples, we know more about the underlying function, and we carve out more hypotheses that are inconsistent from the hypothesis class, but we still are left with many consistent hypotheses. So because learning is ill-

posed, and data itself is not sufficient to find the solution, we should make some extra assumptions to have a unique solution with the data we have. The set of assumptions we make to have learning possible is called the *inductive bias* of the learning algorithm. One way we introduce inductive bias is when we assume a hypothesis class *H.* In learning the class of family car, there are infinitely many ways of separating the positive examples form the negative examples. Assuming the shape of a rectangle is one inductive bias. In linear regression, assuming a linear function is an inductive bias, among all lines, choosing the one that minimizes squared error is another inductive bias.

Thus learning is not possible without inductive bias, and now the question is how to choose the right bias. This is called *model selection*, which is choosing between possible *H*. In answering this question, we should remember that the aim of machine learning is rarely to replicate the training data. But the prediction for new cases. That is we would like to be able to generate the right output for an input instance outside the training set, one for which the correct output is not given in the training set. How well a model trained on the training set predicts the right output for new instances is called *generalization*.

For best generalization, we should match the complexity of the hypothesis class *H* with the complexity of the function underlying the data. If *H* is less complex than the function, we have *underfitting*, for example, when trying to fit a line to data sampled from a third-order polynomial. In such a case, as we increase the complexity, the training error decreases, but if we have *H* that is too complex, the data is not enough

to constrain it and we may end up with a bad hypothesis, $h \in H$, for example, when fitting two rectangles to data sampled from one rectangle. Or if there is noise, an overcomplex hypothesis may learn not only the underlying function but also the noise in the data, and may make a bad fit, for example, when fitting a sixth-order polynomial to noisy data sampled from a third-order polynomial. This is called *overfitting*. In such a case, having more training data helps but only up to a certain point. Given a training set and H, we can find $h \in H$ that has the minimum training error but if H is not chosen well, no matter which $h \in H$ we pick, we will not have good generalization. In all learning algorithms that are trained from example data, there is a trade-off between three factors:

- The complexity of the hypothesis we fit to data, namely the capacity of the hypothesis class,
- the amount of training data, and
- the generalization error on new examples.

As the amount of training data increases, the generalization error decreases. As the complexity of the model class H increases, the generalization error decreases first and then starts to increase. The generalization error of an overcomplex H can be kept in check by increasing the amount of training data but only up to a point.

Appendix A

We can measure the generalization ability of a hypothesis, namely, the quality of its inductive bias, if we have access to data outside the training set. We simulate this by dividing the training set we have into two parts. We use one part for training (i.e. to fit the hypothesis), and the remaining part is called the *validation set* and is used to test the generalization ability. That is, given a set of possible hypothesis classes H_i, for each we fit the best $h_i \in H_i$ on the training set. Then, assuming large enough training and validation sets, the hypothesis that is the most accurate on the validation set is the best one (the one that has the best inductive bias). This process is called *cross-validation*.

The idea behind cross-validation (CV) is simple: we split the training data into *K* folds; then, for each fold (run) $k \in \{1, \ldots, K\}$, we train on all the folds but the k'th, as shown in Figure A.4 [73]. We then compute the error averaged over all the folds, and use this as a proxy for the test error. (Note that each point gets predicted only once, although it will be used for training $K - 1$ times) [73]. It is common to use $K = 5$; this is called 5-fold CV. If we set $K = N$, then we get a method called *leave-one-out cross validation*, or LOOCV, since in fold *i*, we train on all data cases except for *i*, and then test on *i*.

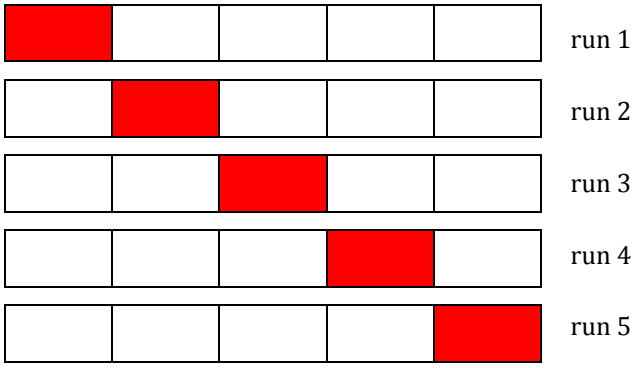

Figure A.4 Schematic of 5-fold cross validation. Figure from [73]

Note that if we need to report the error to give an idea about the expected error of our best model, we should not use the validation error. We have used the validation set to choose the best model, and it has effectively become a part of the training set. We need a third set, a *test set*, containing examples not used in training or validation.

A.4.4 Performance Evaluation

It is important to have methods to evaluate the result of learning. In supervised learning, the learned function is usually evaluated on a separate set of inputs and function values for them called the testing set [75]. A hypothesized function is said to generalize when it guesses well on the testing set. Both mean-squared-error and the total number of errors are common measures [75].

Mean-squared-error (MSE)

$$\widehat{MSE}(x) = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i))^2 \quad \text{(A.15)}$$

where y_i is the actual label of the test set sample,$f(x_i)$ is the predicted label of the test set sample and *N* is the number of samples in the test set.

Misclassification Rate

$$err(f, \mathrm{D}) = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) \neq y_i) \quad \text{(A.16)}$$

where $f(\boldsymbol{x})$ is our classifier, y_i is the actual label of the test set sample,$f(x_i)$ is the predicted label of the test set sample and *N* is the number of samples in the test set.

A.4.5 Dimensionality Reduction

Variable or *feature selection* [76,77] consists of selecting variables for a given prediction task. It has become the focus of much research [76,78,79], particularly in bioinformatics [80]. The analysis of biological data for example, in particular microarray data, generally involves many irrelevant and redundant variables [76,81] and often comparably few training examples. Microarray data also often contains noise. Moreover, the expression levels of many probes may be highly correlated. Such

a characteristic is explained by the co-regulation of many genes: it is assumed that similar patterns in gene expression profiles usually suggest relationships between genes [82]. Therefore, standard methods of supervised learning cannot be applied directly to obtain the parameter estimates. Including all of the genes in the predictive model increases its variance and leads to poor predictive performance [83]. There are several reasons why it is of interest to reduce the dimensionality [74]:

- In most learning algorithms, complexity depends on the number of input dimensions, p, as well as on the size of the data sample, N, and for reduced memory and computation, we are interested in reducing the dimensionality of the problem.
- Simpler models are more robust on small datasets. Simpler models have less variance, that is, they vary less depending on the particulars of a sample, including noise, outliers, and so forth.
- When data can be explained with fewer features, we get a better idea about the process that underlies the data and this allows knowledge extraction.

There are two main methods for reducing dimensionality: *feature selection* and *feature extraction* [74]. In feature selection, we are interested in finding k of p dimensions that give us the most information and we discard the other $p - k$ dimensions. In feature extraction, we are interested in finding a new set of k dimensions that are combinations of the original p dimensions.

A.4.6 Parametric Models for Classification and Regression

The main way to combat the curse of dimensionality is to make some assumptions about the nature of the data distributions, ($p(y|x)$ for a supervised problem) [73]. These assumptions, known as inductive bias, are often embodied in the form of a parametric model, which is a statistical model with a fixed number of parameters [73].

A.5 Linear Models and Least Squares Regression

The linear model has been a mainstay of statistics for the past couple of decades and remains one of our most important tools [18]. Given a vector of inputs $\boldsymbol{X}^T = (X_1, X_2, \ldots, X_p)$, we want to predict a real-valued output Y. The linear model either assumes that the regression function $E(Y|X)$ is linear, or that the linear model is a linear approximation. The linear regression model has the form [18]

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j \tag{A.17}$$

The term $\hat{\beta}_0$ is the intercept, also known as the *bias* in machine learning. Typically, we have a set of training data $(x_1, y_1) \ldots (x_N, y_N)$ from which to estimate the parameters $\hat{\beta}$. Each $x_i = \left(x_{i1}, x_{i2}, \ldots, x_{ip}\right)^T$ is a vector of feature measurements for the ith case. Often it is convenient to include the constant variable 1 in $\boldsymbol{X}$, include $\hat{\beta}_0$ in the vector of coefficients $\hat{\beta}$, and then write the linear model in vector form as an inner product

$$\hat{Y} = \boldsymbol{X}^T\hat{\beta}, \tag{A.18}$$

where $\boldsymbol{X}^T$ denotes vector or matrix transpose ($\boldsymbol{X}$ being a column vector). Here we are modelling a single output, so $\hat{Y}$ is a scalar; in general $\hat{Y}$ can be a K-vector, in which case β would be a $p \times K$ matrix of coefficients. In the $(p+1)$-dimensional input-output space, $(\boldsymbol{X}, \hat{Y})$ represents a hyperplane. If the constant is included in $\boldsymbol{X}$, then the hyperplane includes the origin and is a subspace; if not, it is an affine set cutting the Y-axis at the point $(0, \hat{\beta}_0)$. From now on, we assume that the intercept is included in $\hat{\beta}$.

There are several methods of fitting the linear model to a set of training data, but by far the most popular method is that of *least squares* [18]. In this approach, we pick the coefficients β to minimize the residual sum of squares [18]

$$RSS(\beta) = \sum_{i=1}^{N}(y_i - x_i^T)^2 \tag{A.19}$$

$RSS(\beta)$ is a quadratic function of the parameters, and hence its minimum always exists, but may not be unique [18]. The solution is easiest to characterize in matrix notation. We can write

$$RSS(\beta) = \big((\boldsymbol{y} - \boldsymbol{X}\beta)^T(\boldsymbol{y} - \boldsymbol{X}\beta)\big), \tag{A.20}$$

Where $\boldsymbol{X}$ is an $N \times p$ matrix with each row an input vector, and $\boldsymbol{y}$ is an N-vector of the outputs in the training set. Differentiating with respect to β we get the *normal equations*

$$\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta) = 0 \tag{A.21}$$

If $\boldsymbol{X}^T\boldsymbol{X}$ is non-singular, then the unique solution is given by [18]

$$\hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}, \tag{A.22}$$

And the fitted value at the *i*th input x_i is $\hat{y}_i = \hat{y}(x_i) = x_i^T\hat{\beta}$. At an arbitrary input x_0 the prediction is $\hat{y}(x_0) = x_0^T\hat{\beta}$. The entire fitted surface is characterized by p parameters $\hat{\beta}$.

A.5.1 Subset Selection

There are two reasons why we are often not satisfied with the least squares estimates [18].

- The first is *prediction accuracy*: the least squares estimates often have low bias but large variance. Prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so, we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.
- The second is *interpretation*. With a large number of predictors, we often would like to determine a smaller subset that exhibit the strongest effects. In order to get the "big picture," we are willing to sacrifice some of the small details.

We describe a number of approaches to variable subset selection via the shrinkage method for controlling variance.

A.5.2 Shrinkage Methods

Shrinkage methods produce a model that is interpretable and has possibly lower prediction error than the full model, and because they are continuous, don't suffer as much from high variability [18].

A.5.2.1 Ridge Regression

Ridge regression shrinks the regression coefficients by imposing a penalty on their size [18]. The ridge coefficients minimize a penalized residual sum of squares subject to a bound on the L_2-norm of the coefficients:

$$\hat{\beta}_{ridge} = arg\min_{\beta}\left\{\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right\} \quad \text{(A.23)}$$

Here $\lambda \geq 0$ is a complexity parameter that controls the amount of shrinkage: the larger the value of λ, the greater the amount of shrinkage. The coefficients are shrunk toward zero (and each other). An equivalent way to write the ridge problem is

$$\hat{\beta}_{ridge} = arg\min_{\beta}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 \quad \text{(A.24)}$$

$$subject\ to\ \sum_{j=1}^{p}\beta_j^2 \leq t,$$

which makes explicit the size constraint on the parameters. There is a one-to-one correspondence between the parameters λ in (A.23) and t in (A.24) [18]. Writing (A.23) in matrix form, we have [18]

$$RSS(\lambda) = (\boldsymbol{y} - \boldsymbol{X}\beta)^T(\boldsymbol{y} - \boldsymbol{X}\beta) + \lambda\beta^T\beta, \quad \text{(A,25)}$$

The ridge regression solutions are easily seen to be

$$\hat{\beta}_{ridge} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}, \quad \text{(A.26)}$$

where $\boldsymbol{I}$ is the $p \times p$ identity matrix. Notice that with the choice of quadratic penalty $\beta^T\beta$, the ridge regression solution is again a linear function of $\boldsymbol{y}$. The solution adds a positive constant to the diagonal of $\boldsymbol{X}^T\boldsymbol{X}$ before inversion. This makes the problem non-singular, even if $\boldsymbol{X}^T\boldsymbol{X}$ is not of full rank, and this was the main motivation for ridge regression [84] and traditional descriptions of this method start with (A.26). However, starting from (A.23) and (A.24) gives insight into how it works [18] and provides a general framework in which the other regularization method (Lasso) introduced in the subsequent section can be integrated.

A.5.2.2 The Lasso

The *Lasso* (least absolute shrinkage and selection operator) is a shrinkage method like ridge, with subtle but important differences [18]. Ridge regression does not provide any interpretable model because it does not set any coefficients to zero. To circumvent these issues, [29] introduced the Lasso, which minimizes the residual sum of squares subject to a bound on the L_1-norm of the coefficients [18],

$$\hat{\beta}_{Lasso} = arg\min_{\beta} \sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 \quad \text{subject to } \sum_{j=1}^{p}|\beta_j| \leq t \tag{A.27}$$

We can write the Lasso problem in the equivalent *Lagrangian form*

$$\hat{\beta}_{Lasso} = arg\min_{\beta}\left\{\frac{1}{2}\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\right\} \tag{A.28}$$

Notice the similarity to the ridge regression problem (A.23) and (A.24): the L_2 ridge penalty $\sum_{j=1}^{p} \beta_j^2$ is replaced by the L_1 Lasso penalty $\sum_{j=1}^{p} |\beta_j|$. This latter constraint makes the solutions nonlinear in the y_i, and as a consequence, a closed form solution does not exist as it does for ridge regression [18]. Computing a solution for the Lasso is a quadratic programming problem. Due to the L_1 penalty term, the Lasso solution will have some entries equal to zero for λ small enough, thus Lasso is a method of feature selection [83]. Feature selection is desirable because of a potential increase in accuracy, and by reducing the number of variables, the problem becomes easier to understand [18].

A.5.3 Multiple Linear Regression

The linear model (A.17) with $p > 1$ inputs is called the multiple linear regression model. The least squares estimates (A.22) for this model are best understood in terms of the estimates for the univariate ($p = 1$) liner model, as we indicate in this section. Suppose first that we have a univariate model with no intercept, that is [18],

$$\boldsymbol{Y} = \boldsymbol{X}\beta + \varepsilon \tag{A.29}$$

The least squares estimate and residuals are

$$\hat{\beta} = \frac{\sum_1^N x_i y_i}{\sum_1^N x_i^2} \tag{A.30}$$

$$r_i = y_i - x_i\hat{\beta}$$

Appendix A

In convenient vector notation, we let $= (y_1, \ldots, y_N)^T$, $\boldsymbol{x} = (x_1, \ldots, x_N)^T$ and define

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{N} x_i y_i,$$
$$= \boldsymbol{x}^T \boldsymbol{y} \qquad \text{(A.31)}$$

The inner product between $\boldsymbol{x}$ and $\boldsymbol{y}$[4]. Then we can write

$$\hat{\beta} = \frac{\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\langle \boldsymbol{x}, \boldsymbol{x} \rangle} \qquad \text{(A.32)}$$
$$\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{x}\hat{\beta}$$

This simple univariate regression provides the building block for multiple linear regression. Suppose next that the inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_p$ (the columns of data matrix $\boldsymbol{X}$) are orthogonal; that is $\langle \boldsymbol{x}_j, \boldsymbol{y}_k \rangle = 0$ for all $j \neq k$. Then it is easy to check that the multiple least squares estimates $\hat{\beta}_j$ are equal to $\langle \boldsymbol{x}_j, \boldsymbol{y} \rangle / \langle \boldsymbol{x}_j, \boldsymbol{x}_j \rangle$ - the univariate estimates. In other words, when the inputs are orthogonal, they have no effect on each other's parameter estimates in the model.

Orthogonal inputs occur most often with balanced, designed experiments (where orthogonality is enforced), but almost never with observational data. Hence we will have to orthogonalize them in order to carry this idea further [18]. Suppose next that

[4] The inner-product notation is suggestive of generalizations of linear regression to different metric spaces, as well as to probability spaces [18].

we have an intercept and a single input $\boldsymbol{x}$. Then the least squares coefficient of $\boldsymbol{x}$ has the form

$$\hat{\beta}_1 = \frac{\langle \boldsymbol{x} - \bar{x}\mathbf{1}, \boldsymbol{y} \rangle}{\langle \boldsymbol{X} - \bar{x}\mathbf{1},\ \boldsymbol{X} - \bar{x}\mathbf{1} \rangle}, \tag{A.33}$$

where $\bar{x} = \sum_i x_i / N$, and $\mathbf{1} = \boldsymbol{x}_0$, the vector of N ones. We can view the estimate (A.33) as the result of two applications of the simple regression (A.32). The steps are:

1. Regress $\boldsymbol{x}$ on $\mathbf{1}$ to produce the residual $\boldsymbol{z} = \boldsymbol{x} - \bar{x}\mathbf{1}$;
2. Regress $\boldsymbol{y}$ on the residual $\mathbf{z}$ to give the coefficient $\hat{\beta}_1$

In this procedure, "regress **b** on **a**" means a simple univariate regression of **b** on a with no intercept, producing coefficient $\hat{\gamma} = \langle \boldsymbol{a}, \boldsymbol{b} \rangle / \langle \boldsymbol{a}, \boldsymbol{a} \rangle$ and residual vector $\boldsymbol{b} - \hat{\gamma}\boldsymbol{a}$. We say that **b** is adjusted for **a**, or is "orthogonalized" with respect to **a**.

Step 1 orthogonalizes x with respect to $\boldsymbol{x}_0 = \mathbf{1}$. Step 2 is just a simple univariate regression, using the orthogonal predictors $\mathbf{1}$ and $\boldsymbol{z}$. This recipe generalizes to the case of p inputs, as shown in Algorithm A.1. Note that the inputs $\boldsymbol{z}_0, \ldots, \boldsymbol{z}_{j-1}$ in step 2 are orthogonal, hence the simple regression coefficients computed there are in fact also the multiple regression coefficients.

Appendix A

Algorithm A.1 *Regression by Successive Orthogonalization*

1: Initialize $\boldsymbol{z}_0 = \boldsymbol{x}_0 = \mathbf{1}$

2: For $j = 1,2,\ldots,p$

Regress $\boldsymbol{x}_j$ on $\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_{j-1}$ to produce coefficients $\gamma_{\ell j} = \langle \boldsymbol{z}_\ell, \boldsymbol{x}_j \rangle / \langle \boldsymbol{z}_\ell, \boldsymbol{z}_\ell \rangle$,

$\ell = 0, \ldots, j-1$ and residual vector $\boldsymbol{z}_j = \boldsymbol{x}_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} \boldsymbol{z}_k$

3: Regress $\boldsymbol{y}$ on the residual $\boldsymbol{z}_p$ to give the estimate $\hat{\beta}_p$

The result of this algorithm is

$$\hat{\beta}_p = \frac{\langle \boldsymbol{z}_p, \boldsymbol{y} \rangle}{\langle \boldsymbol{z}_p, \boldsymbol{z}_p \rangle} \tag{A.33}$$

Re arranging the residual in step 2, we can see that each of the $\boldsymbol{x}_j$ is a linear combination of the $\boldsymbol{z}_k, k \leq j$. Since the $\boldsymbol{z}_j$ are all orthogonal, they form a basis for the column space of $\boldsymbol{X}$, and hence the least squares projection onto this subspace is $\hat{\boldsymbol{y}}$. Since $\boldsymbol{z}_p$ alone involves $\boldsymbol{x}_p$ (with coefficient 1), we see that the coefficient (A.33) is the multiple regression coefficient of $\boldsymbol{y}$ on $\boldsymbol{x}_p$. This key result exposes the effect of correlated inputs in multiple regression.

The multiple regression coefficient $\hat{\beta}_j$ represents the additional contribution of $\boldsymbol{x}_j$ on $\boldsymbol{y}$, after $\boldsymbol{x}_j$ has been adjusted for $\boldsymbol{x}_0, \boldsymbol{x}_1, , \ldots, \boldsymbol{x}_{j-1}, \boldsymbol{x}_{j+1}, \ldots, \boldsymbol{x}_p$. Algorithm A.1 is known as the *Gram-Schmidt* procedure for multiple regression, and is also a useful strategy for computing the estimates.

A.5.4 Matrix Inverse

The $n \times n$ identitiy matrix $\boldsymbol{I}_n$ is defined by column partitioning

$$\boldsymbol{I}_n = [e_1, \ldots, e_n],$$

where e_k is the kth "canonical" vector:

$$e_{1=}\left(\underbrace{0,\ldots,0}_{k-1}, \underbrace{1,0,\ldots,0}_{n-k}\right)^T$$

The canonical vectors arise frequently in matrix analysis and if their dimension is ever ambiguous, we use superscripts, i.e., $e_k^{(n)} \in \mathbb{R}^n$.

If $\boldsymbol{A}$ and $\boldsymbol{X}$ are $\mathbb{R}^{n\times n}$ and satisfy $\boldsymbol{AX} = \boldsymbol{I}$, then $\boldsymbol{X}$ is the inverse $\boldsymbol{A}$ and is denoted by $\boldsymbol{A}^{-1}$.

If $\boldsymbol{A}^{-1}$exists, then $\boldsymbol{A}$ is said to be *nonsingular*. Otherwise, we say $\boldsymbol{A}$ is *singular*.

A.5.4.1 Moore-Penrose pseudoinverse

The Moore-Penrose pseudoinverse is a generalization of the inverse of a matrix and is defined for any matrix and is unique [85].

If $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, *then there exists a unique* $\boldsymbol{A}^+ \in \mathbb{R}^{m\times n}$ *that satisfies the four Penrose conditions*:

1. $\boldsymbol{AA}^+\boldsymbol{A} = \boldsymbol{A}$
2. $\boldsymbol{A}^+\boldsymbol{AA}^+ = \boldsymbol{A}^+$
3. $(\boldsymbol{A}^+\boldsymbol{A})^T = \boldsymbol{A}^+\boldsymbol{A}$
4. $(\boldsymbol{AA}^+)^T = \boldsymbol{AA}^+$

Furthermore, $\boldsymbol{A}^+$ always exists and is unique.

Appendix A

If $\mathbf{A}$ is nonsingular, then $\boldsymbol{A}^{+} = \boldsymbol{A}^{-1}$ trivially satisfies the four equations and are the generalized inverse of $\boldsymbol{A}$ [86]. Since the pseudoinverse is known to be unique, it follows that the pseudoinverse of a nonsingular matrix is the same as the ordinary inverse. A neat algebraic proof of the uniqueness of the pseudoinverse is given by [85], and a constructive proof is given by [87].

Appendix B

We consider a problem of fitting a large-scale covariance matrix to multivariate Gaussian data in such a way that the inverse is sparse, thus providing model selection. Beginning with a dense empirical covariance matrix, the maximum likelihood problem is solved with an L_1-norm penalty term added to encourage sparsity in the inverse.

A.1 *Notation*

For a $p \times p$ matrix $\boldsymbol{X}, \boldsymbol{X} \succeq 0$ means $\boldsymbol{X}$ is symmetric and positive semi-definite; $\|\boldsymbol{X}\|$ denotes the largest singular value norm, $\|\boldsymbol{X}\|_1$ the sum of the absolute values of its elements, and $\|\boldsymbol{X}\|_\infty$ their largest magnitude.

$$\|\boldsymbol{X}\|_1 = |x_{i,j}| + \cdots + |x_{i,n}| \tag{A.1}$$

$$\|\boldsymbol{X}\|_\infty = max(|x_{i,j}|, \cdots, |x_{i,n}|) \tag{A.2}$$

A.2 *Problem Setup*

Given $x_1 \cdots, x_n\ N(\boldsymbol{\mu}, \sum)$, where $N(\boldsymbol{\mu}, \sum)$ represents a p-dimensional multivariate Gaussian with mean vector $\mu \in \mathbb{R}^p$ and covariance matrix $\sum \in \mathbb{R}^{p \times p}$. Let $\boldsymbol{S} \geq 0$ be the given empirical covariance matrix. Let the variable $\boldsymbol{X}$ be our estimate of the inverse covariance matrix.

Appendix B

We consider the maximum likelihood problem,

$$f(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\sum|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\sum^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right], \tag{A.3}$$

where $\boldsymbol{x} \in \mathbb{R}^n$.

Gaussian Likelihood Function:

$$L(\boldsymbol{\mu}, \sum) = (constant)\prod_{i=1}^{n}|\sum|^{-\frac{1}{2}}exp\left[-\frac{1}{2}(x_i-\mu)^T\sum^{-1}(x_i-\mu)\right] \tag{A.4}$$

$$L(\mu, \sum)\ \alpha\ |\sum|^{-\frac{n}{2}}exp\left[-\frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^T\sum^{-1}(x_I-\mu)\right] \tag{A.5}$$

Replace μ with $\bar{x} = \sum_{i=1}^{n}x_i$. (A.6)

Now find the $\sum$ that maximizes $L(\bar{x}, \sum)$. We start by taking the log of the function since it is easier to work with, as the log is a strictly increasing function.

$$log[L(\bar{x}, \sum)] = -\frac{n}{2}log|\sum| - \frac{1}{2}\sum_{i=1}^{n}(x_i-\bar{x})^T\sum^{-1}(x_i-\bar{x}) \tag{A.7}$$

$Trace(\boldsymbol{AB}) = Trace(\boldsymbol{BA})$ when matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are shaped so that both products exist.

Appendix B

$$log[L(\bar{x}, \sum)] = -\frac{n}{2} log|\sum| - \frac{1}{2}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{x})^T \sum^{-1}(\boldsymbol{x}_i - \bar{x})$$

$$\alpha - \frac{n}{2} log|\sum| - \frac{1}{2}\sum_{i=1}^{n} Trace(\sum^{-1}(\boldsymbol{x}_i - \bar{x})(\boldsymbol{x}_i - \bar{x})^T)$$

$$= -\frac{n}{2} log|\sum| - \frac{1}{2} Trace(\sum^{-1}\sum_{i=1}^{n}[(\boldsymbol{x}_i - \bar{x})(\boldsymbol{x}_i - \bar{x})^T]) \quad \text{(A.8)}$$

Do a change of variable:

$$\sum = \boldsymbol{X}^{-1} \quad \text{(A.9a)}$$

$$\boldsymbol{X} = \sum^{-1} \quad \text{(A.9b)}$$

$\boldsymbol{S}$ is the sample covariance.

$$\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \bar{x})(\boldsymbol{x}_i - \bar{x})^T \quad \text{(A.10)}$$

$$log[L(\bar{x}, \sum)] = -\frac{n}{2} log|\boldsymbol{X}^{-1}| - \frac{n}{2} Trace(\boldsymbol{XS}) \quad \text{(A.11)}$$

$$\text{where } |\boldsymbol{X}^{-1}| = \frac{1}{|\boldsymbol{X}|} \quad \text{(A.12)}$$

Rewrite the log likelihood function:

$$log[L(\bar{x}, \sum)] = -\frac{n}{2} log\left(\frac{1}{|\boldsymbol{X}|}\right) - \frac{n}{2} Trace(\boldsymbol{XS}) \quad \text{(A.13)}$$

Appendix B

Rewrite the log likelihood function:

$$log[L(\bar{x}, \sum)] = \frac{n}{2} log(|\boldsymbol{X}|) - \frac{n}{2} Trace(\boldsymbol{XS}) \quad \text{(A.14)}$$

$$log[L(\bar{x}, \sum)] = \frac{n}{2} [log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS})] \quad \text{(A.15)}$$

We want to choose x that maximizes the log likelihood function. This is equivalent to (A.16)*:*

$$\underset{\boldsymbol{X} \succeq 0}{max} \left\{ \frac{n}{2} (log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS})) \right\} \quad \text{(A.16)}$$

Rewrite the optimization problem without the constant term:

$$\underset{\boldsymbol{X} \succeq 0}{max} \{(log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS}))\} \quad \text{(A.17)}$$

$$(\sum{}^{*})^{-1} = \boldsymbol{X}^{*} = \underset{\boldsymbol{X} \succeq 0}{argmax} \{(log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS}))\} \quad \text{(A.18)}$$

A.3 *Problem Setup (Continued)*

We consider the penalized maximum likelihood problem,

$$\underset{\boldsymbol{X} \succ 0}{maximize} \{log(|\boldsymbol{X}|) - Trace(\boldsymbol{XS}) - \rho \|\boldsymbol{X}\|_1\}, \quad \text{(A.19)}$$

where $Trace(\boldsymbol{XS})$ denotes the scalar product between two symmetric matrices $\boldsymbol{S}$ and $\boldsymbol{X}$, and the term $\|\boldsymbol{X}\|_1 \coloneqq \sum_{i,j}|\boldsymbol{X}_{i,j}|$ penalizes nonzero elements of $\boldsymbol{X}$. (A.19) is a convex optimization problem because the objective function is concave on a set of positive definite matrices [3]. Here, the scalar parameter $\rho > 0$ controls the size of the penalty, hence the sparsity of the solution. The penalty term involving the sum of absolute values of the entries of $\boldsymbol{X}$ is a proxy for the number of its non-zero elements. The classical maximum likelihood estimate of $\sum$ is recovered for $\rho = 0$, and is simply $\boldsymbol{S}$, the empirical covariance matrix. Due to noise in the data, however, $\boldsymbol{S}$ may not have a sparse inverse, even if there are many conditional independence properties in the underlying distribution. In the approach of the paper, we strike a tradeoff between maximality of the likelihood and the number of non-zero elements in the inverse covariance matrix, and this method is potentially useful for discovering conditional independence properties. Furthermore, for $p >> n$, the matrix $\boldsymbol{S}$ is likely to be singular. It is desirable for our estimate of $\sum$ to be invertible. It will be shown that the proposed estimator performs some regularization, so that our estimate is invertible for every $\rho > 0$.

lemma 1: Let $\|\cdot\|$ be a norm on R^n. The associated dual norm, denoted $\|\cdot\|_*$, is defined as,

$$\|z\|_* = sup\{z^T x : \|x\| \leq 1\} \tag{A.20}$$

Appendix B

The dual of the l_1-norm is the l_1-norm given by,

$$sup\{< z, x >: \|x\|_\infty \leq 1\} = \sum_{i=1}^{n} |z_i| = \|z\|_1 \tag{A.21}$$

$$sup\{z^T x: \|x\|_\infty \leq 1\} = \sum_{i=1}^{n} |z_i| = \|z\|_1 \tag{A.22}$$

The dual of the l_1-norm is the l_1-norm given by

$$sup\{< z, x >: \|x\|_1 \leq 1\} = \|z\|_\infty \tag{A.23}$$

$$sup\{z^T x: \|x\|_1 \leq 1\} = \|z\|_\infty \tag{A.24}$$

By introducing a dual variable $\boldsymbol{U}$*, and using lemma 1, we write (A.19) as,*

$$\underset{\boldsymbol{X} \succ 0}{max} \quad \underset{\|\boldsymbol{U}\|_\infty \leq \rho}{min} \{log|\boldsymbol{X}| - Trace(\boldsymbol{X}, \boldsymbol{S}) - Trace(\boldsymbol{X}, \boldsymbol{U})\} \tag{A.25}$$

This simplifies to,

$$\underset{\boldsymbol{X} \succ 0}{max} \quad \underset{\|\boldsymbol{U}\|_\infty \leq \rho}{min} \{log|\boldsymbol{X}| - Trace(\boldsymbol{X}, \boldsymbol{S} + \boldsymbol{U})\}, \tag{A.26}$$

where $\|\boldsymbol{U}\|_\infty$ denotes the maximal absolute value of the entries of $\boldsymbol{U}$. This corresponds to seeking an estimate with maximal worst-case likelihood, over all component-wise bounded additive perturbations $\boldsymbol{S} + \boldsymbol{U}$ of the empirical covariance matrix $\boldsymbol{S}$.

Obtain the dual problem by exchanging the max and min:. The inner optimization problem can be solved easily because it resembles the maximum likelihood problem.

$$\min_{\|\boldsymbol{U}\|_\infty \leq \rho} \max_{\boldsymbol{X} \succ 0} \{log|\boldsymbol{X}| - Trace(\boldsymbol{X}, \boldsymbol{S} + \boldsymbol{U})\} \tag{A.27}$$

Solve the inner optimization problem of (A.27),

$$\max_{\boldsymbol{X} \succ 0} \{log|\boldsymbol{X}| - Trace(\boldsymbol{X}, \boldsymbol{S} + \boldsymbol{U})\} \tag{A.28}$$

In the maximum likelihood problem,

$$\underset{\boldsymbol{X}}{argmax}\{log|\boldsymbol{X}| - Trace(\boldsymbol{X}, \boldsymbol{S})\} = \boldsymbol{S}^{-1} \tag{A.29}$$

From the MLE solution, we can see that the solution to (A.28) is given by,

Appendix B

$$\max_{\boldsymbol{X} \succ 0} \{log|\boldsymbol{X}| - Trace(\boldsymbol{X}, \boldsymbol{S} + \boldsymbol{U})\} = (\boldsymbol{S} + \boldsymbol{U})^{-1} \quad \text{(A.30)}$$

$$\text{So, } \boldsymbol{X}^* = (\boldsymbol{S} + \boldsymbol{U})^{-1} \quad \text{(A.31)}$$

Using solution (A.31), rewrite (A.27) as,

$$\min_{\|\boldsymbol{U}\|_\infty \leq \rho} \{log|(\boldsymbol{S} + \boldsymbol{U})^{-1}| - Trace((\mathbf{S} + \boldsymbol{U})^{-1}, \boldsymbol{S} + \boldsymbol{U}\} \quad \text{(A.32)}$$

After simplifying, this becomes,

$$\min_{\|\boldsymbol{U}\|_\infty \leq \rho} \{log|(\boldsymbol{S} + \boldsymbol{U})^{-1}| - p\} \quad s.t: \ \boldsymbol{S} + \boldsymbol{U} \succ 0 \quad \text{(A.33)}$$

$log|\boldsymbol{X}^{-1}| = -log|\boldsymbol{X}|$, so we can rewrite the optimization problem as,

$$\min_{\|\boldsymbol{U}\|_\infty \leq \rho} \{-log|(\boldsymbol{S} + \boldsymbol{U})| - p\} \quad s.t: \ \boldsymbol{S} + \boldsymbol{U} \succ 0 \quad \text{(A.34)}$$

The diagonal elements of an optimal $\boldsymbol{U}$ are simply $\widehat{\boldsymbol{U}}_{ii} = \rho$. The corresponding covariance matrix estimate is $\widehat{\sum} \coloneqq \mathbf{S} + \widehat{\boldsymbol{U}}$. Note that ρ is the number of variables.

Appendix B

A.2. *BLOCK COORDINATE DESCENT METHOD*

An efficient algorithm for solving the above dual problem (A.27) is based on the block coordinate descent.

A.2.1 *Algorithm*

We describe a method for solving (A.34) by optimizing over one column and row of $\boldsymbol{S} + \boldsymbol{U}$ at a time. Let $\boldsymbol{W} \coloneqq \boldsymbol{S} + \boldsymbol{U}$ be our estimate of the true covariance. The algorithm begins by initializing $\boldsymbol{W}^0 = \boldsymbol{S} + \rho\boldsymbol{I}$. The diagonal elements of $\boldsymbol{W}^0$ are set to their optimal values, and are left unchanged in what follows. We can permute rows and columns of $\boldsymbol{W}$, so that we optimizing over the last column and row.

Partition $\boldsymbol{W}$ and $\boldsymbol{S}$ as,

$$\boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}_{11} & \boldsymbol{w}_{12} \\ \boldsymbol{w}_{12}^T & w_{22} \end{pmatrix} \tag{A.35}$$

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_{11} & \boldsymbol{s}_{12} \\ \boldsymbol{s}_{12}^T & s_{22} \end{pmatrix}, \tag{A.36}$$

where $\boldsymbol{w}_{12}, \boldsymbol{s}_{12} \epsilon\ \mathbb{R}^{p-1}$.

Appendix B

Initialize $\boldsymbol{W}$ to its optimal value: $\boldsymbol{W}^0 = \boldsymbol{S} + \rho\boldsymbol{I}$. Permute rows and columns of $\boldsymbol{W}$ so that we are optimizing over the last column and row (due to symmetry).

The update rule is found by solving the dual problem (A.34), with $\boldsymbol{U}$ fixed except for its last column and row.

Rewrite the optimization problem (A.34),

$$\min_{\boldsymbol{U}} \quad \{-log|(\boldsymbol{S}+\boldsymbol{U})| - p\} \quad s.t.\ \boldsymbol{S}+\boldsymbol{U} \succ 0, \|\boldsymbol{U}\|_\infty \leq \rho \tag{A.37a}$$

$$\min_{\boldsymbol{w}_{12}} \quad \left\{-log\begin{vmatrix}\boldsymbol{W}_{11} & \boldsymbol{w}_{12}\\ \boldsymbol{w}_{12}^T & w_{22}\end{vmatrix} - p\right\} \quad s.t.: \quad \|\boldsymbol{w}_{12} - \boldsymbol{s}_{12}\|_\infty \leq \rho, \boldsymbol{W} \succ 0 \tag{A.37b}$$

$$\min_{\boldsymbol{w}_{12}} \quad \left\{-log\begin{vmatrix}\boldsymbol{W}_{11} & \boldsymbol{w}_{12}\\ \boldsymbol{w}_{12}^T & w_{22}\end{vmatrix}\right\} \quad s.t.: \quad \|\boldsymbol{w}_{12} - \boldsymbol{s}_{12}\|_\infty \leq \rho, \boldsymbol{W} \succ 0 \tag{A.37c}$$

$$\max_{\boldsymbol{w}_{12}} \quad \left\{\begin{vmatrix}\boldsymbol{W}_{11} & \boldsymbol{w}_{12}\\ \boldsymbol{w}_{12}^T & w_{22}\end{vmatrix}\right\} \quad s.t.: \quad \|\boldsymbol{w}_{12} - \boldsymbol{s}_{12}\|_\infty \leq \rho, \boldsymbol{W} \succ 0 \tag{A.37d}$$

lemma 2: Let $\boldsymbol{M}$ be an $n \times n$ Hermitian matrix partitioned as,

$$M = \begin{pmatrix}\boldsymbol{M}_{11} & \boldsymbol{M}_{12}\\ \boldsymbol{M}_{21} & M_{22}\end{pmatrix}, \tag{A.38}$$

in which $\boldsymbol{M}_{11}$ is square and nonsingular $k \times k$ block with $1 \leq k < n$. The Schur complement of $\boldsymbol{M}_{11}$ in $\boldsymbol{M}$ is defined and denoted by,

$$\boldsymbol{M}/_{\boldsymbol{M}_{11}} = M_{22} - \boldsymbol{M}_{21}\boldsymbol{M}_{11}^{-1}\boldsymbol{M}_{12}. \tag{A.39}$$

lemma 3: Let $\boldsymbol{M}$ be a square matrix partitioned as in *lemma 1*. If $\boldsymbol{M}_{11}$ is nonsingular, then,

$$|\boldsymbol{M}| = |\boldsymbol{M}_{11}| \left|\boldsymbol{M}/_{\boldsymbol{M}_{11}}\right| \tag{A.40}$$

Rewrite the optimization problem using lemma 2 and lemma 3,

$$\underset{\boldsymbol{w}_{12}}{max} \quad \{|\boldsymbol{W}_{11}|\,|\boldsymbol{w}_{22} - \boldsymbol{w}_{12}^T\boldsymbol{W}_{11}^{-1}\boldsymbol{w}_{12}|\} \qquad s.t:\ \|\boldsymbol{w}_{12} - \boldsymbol{s}_{12}\|_\infty \leq\ \rho, \boldsymbol{W} \succ 0 \tag{A.41}$$

w_{22} is largest when $w_{22} = s_{22} + \rho$ (initialized at the beginning of the algorithm). Therefore, (A.41) is solved by minimizing the expression $\boldsymbol{w}_{12}^T\boldsymbol{W}_{11}^{-1}\boldsymbol{w}_{12}$.

Let $\boldsymbol{y} = \boldsymbol{w}_{12}$

$$\widehat{\boldsymbol{w}}_{12} \coloneqq \underset{\boldsymbol{y}}{argmin}\{\boldsymbol{y}^T\boldsymbol{W}_{11}^{-1}\boldsymbol{y}\} \qquad s.t:\ \|\boldsymbol{y} - \boldsymbol{s}_{12}\|_\infty \leq\ \rho, \boldsymbol{W} \succ 0. \tag{A.42}$$

(A.42) is a box-constrained quadratic program (QP). We cycle through columns in order, solving a QP at each step. After each sweep through all columns, we check if the primal-dual gap is less than ε, a given tolerance.

Steps of the Algorithm:

1) Cycle through the columns in order

2) Compute $\boldsymbol{W}_{11}^{-1}$

3) Solve the quadratic optimization problem (A.42)

4) Repeat until the duality gap is less than ε, a given tolerance

The primal variable is related to $\boldsymbol{W}$ by $\boldsymbol{X} = \boldsymbol{W}^{-1}$.

The duality gap condition is then $Trace(\boldsymbol{SX}) + \rho\|\boldsymbol{X}\|_1 \leq p + \varepsilon$.

A.2.2 *Connection to Lasso*

The dual of (A.42) is penalized least squares problem, often referred to as Lasso and is given by,

$$\min_{\boldsymbol{x}} \quad \{\boldsymbol{x}^T W_{11}\boldsymbol{x} - \boldsymbol{s}_{12}^T\boldsymbol{x} + \rho\|\boldsymbol{x}\|_1\} \tag{A.43}$$

Appendix B

Proof:

Rewrite (A.43) using dual norm definition in *lemma 1* and using dual variable *V,*

$$\min_{\boldsymbol{x}} \quad \{\boldsymbol{x}^T \boldsymbol{W}_{11} \boldsymbol{x} - \boldsymbol{s}_{12}^T \boldsymbol{x} + \max\{\boldsymbol{v}^T \boldsymbol{x} \ : \|\boldsymbol{V}\|_\infty \leq \rho\}\} \tag{A.44}$$

The first two terms in the minimization do not depend on the new dual variable $\boldsymbol{V}$, so we can write (A.44) as,

$$\min_{\boldsymbol{x}} \quad max \qquad \{\|\boldsymbol{V}\|_\infty \leq \rho\} \boldsymbol{x}^T \boldsymbol{W}_{11} \boldsymbol{x} - (\boldsymbol{s}_{12} - \boldsymbol{v})^T \boldsymbol{x} \tag{A.45a}$$

$$\min_{\boldsymbol{x}} \quad \max_{\|\boldsymbol{V}\|_\infty \leq \rho} \quad \{\boldsymbol{x}^T \boldsymbol{W}_{11} \boldsymbol{x} - (\boldsymbol{s}_{12} - \boldsymbol{v})^T \boldsymbol{x}\} \tag{A.45b}$$

Take the dual by exchanging the *max* and *min*,

$$\max_{\|\boldsymbol{V}\|_\infty \leq \rho} \quad \min_{\boldsymbol{x}} \qquad \{\boldsymbol{x}^T \boldsymbol{W}_{11} \boldsymbol{x} - (\boldsymbol{s}_{12} - \boldsymbol{v})^T \boldsymbol{x}\} \tag{A.46}$$

Solve the inner minimization problem analytically,

$$\nabla f(\boldsymbol{x}) = 0 \tag{A.47a}$$

$$\nabla_{\boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{W}_{11} \boldsymbol{x} - (\boldsymbol{s}_{12} - \boldsymbol{v})^T \boldsymbol{x} \tag{A.47b}$$

$$2\boldsymbol{W}_{11} \boldsymbol{x} - (\boldsymbol{s}_{12} - \boldsymbol{v}) = 0 \tag{A.47c}$$

$$2\boldsymbol{W}_{11} \boldsymbol{x} = \boldsymbol{s}_{12} - \boldsymbol{v} \tag{A.47d}$$

$$\boldsymbol{x} = \tfrac{1}{2} \boldsymbol{W}_{11}^{-1} (\boldsymbol{s}_{12} - \boldsymbol{v}) \tag{A.47e}$$

Use the solution to the inner minimization problem in (A.46),

$$\underset{\|\boldsymbol{V}\|_\infty \leq \rho}{max} \left\{ (\tfrac{1}{2} \boldsymbol{W}_{11} (\boldsymbol{s}_{12} - \boldsymbol{v}))^T \boldsymbol{W}_{11} (\tfrac{1}{2} \boldsymbol{s}_{12} - \boldsymbol{v})) - (\boldsymbol{s}_{12} - \boldsymbol{v})^T (\tfrac{1}{2} \boldsymbol{W}_{11} (\boldsymbol{s}_{12} - \boldsymbol{v})) \right\} \tag{A.47}$$

By simplifying the optimization problem becomes,

$$\underset{\|\boldsymbol{V}\|_\infty \leq \rho}{max} \left\{ (\frac{1}{4} (\boldsymbol{s}_{12} - \boldsymbol{v})^T \boldsymbol{W}_{11}^{-1} \boldsymbol{W}_{11} \boldsymbol{W}_{11}^{-1} (\boldsymbol{s}_{12} - \boldsymbol{v}) - (\frac{1}{2} (\boldsymbol{s}_{12} - \boldsymbol{v})^T \boldsymbol{W}_{11}^{-1} (\boldsymbol{s}_{12} - \boldsymbol{v})) \right\}$$

$$\underset{\|\boldsymbol{V}\|_\infty \leq \rho}{max} \quad \left\{ -(\tfrac{1}{4} (\boldsymbol{s}_{12} - \boldsymbol{v})^T \boldsymbol{W}_{11}^{-1} (\boldsymbol{s}_{12} - \boldsymbol{v})) \right\} \tag{A.48}$$

Doing a variable substitution: $\boldsymbol{y} = \boldsymbol{s}_{12} - \boldsymbol{v}$,

$$\max_{\|\boldsymbol{y}-\boldsymbol{s}_{12}\|_\infty \le \rho} \left\{-\left(\frac{1}{4}\boldsymbol{y}^T\boldsymbol{W}_{11}^{-1}\boldsymbol{y}\right)\right\} \tag{A.49a}$$

$$\min_{\|\boldsymbol{y}-\boldsymbol{s}_{12}\|_\infty \le \rho} \left\{\left(\frac{1}{4}\boldsymbol{y}^T\boldsymbol{W}_{11}^{-1}\boldsymbol{y}\right)\right\} \tag{A.49b}$$

$$\min_{\|\boldsymbol{y}-\boldsymbol{s}_{12}\|_\infty \le \rho} \{\boldsymbol{y}^T\boldsymbol{W}_{11}^{-1}\boldsymbol{y}\} \tag{A.49c}$$

This proves that (A.42) and (A.43) are equal.

Strong duality obtains so that (A.42) and (A.43) are equivalent. If we let $\boldsymbol{Q}$ denote the square root of $\boldsymbol{W}_{11}$, and $\boldsymbol{b} = \frac{1}{2}\boldsymbol{Q}^{-1}\boldsymbol{s}_{12}$, then we write (A.43) as,

$$\min_{\boldsymbol{x}} \{\|\boldsymbol{Q}\boldsymbol{x}-\boldsymbol{b}\|_2^2 + \rho\|\boldsymbol{x}\|_1\} \tag{A.50}$$

Proof:

We want to show that (A.43) is equivalent to (A.50). Comparing (A.43) and (A.50), we need to show what values of $\boldsymbol{Q}$ and $\boldsymbol{b}$ make the two equations equal.

This is equivalent to showing that,

$$\|\boldsymbol{Q}\boldsymbol{x}-\boldsymbol{b}\|_2^2 = \boldsymbol{x}^T\boldsymbol{W}_{11}\boldsymbol{x} - \boldsymbol{s}_{12}^T x \tag{A.51}$$

Appendix B

We start by expanding the right hand side of the equation,

$$\|\boldsymbol{Qx} - \boldsymbol{b}\|_2^2 = (\boldsymbol{Qx} - \boldsymbol{b})_T(\boldsymbol{Qx} - \boldsymbol{b}) \tag{A.52a}$$

$$= \boldsymbol{x}^T\boldsymbol{Q}^T\boldsymbol{Qx} - (2\boldsymbol{Qb})^T\boldsymbol{x} + \|\boldsymbol{b}\|^2 \tag{A.52b}$$

Since we are minimizing over x, the term $\|\boldsymbol{b}\|^2$ can be dropped.

$\boldsymbol{W}_{11}$ is symmetric and positive definite, so it will have a square root $\boldsymbol{Q}$ that is also symmetric and positive definite.

$$\boldsymbol{Q} = \boldsymbol{Q}^T \tag{A.53}$$

$\boldsymbol{Q}^{-1}$ exists because $\boldsymbol{Q}$ is positive definite.

By symmetry of $\boldsymbol{W}_{11}$, we can write an eigenvalue decomposition.

$$\boldsymbol{W}_{11} = \boldsymbol{ULU}^T \tag{A.54a}$$

$$\boldsymbol{Q} = \boldsymbol{UL}^{\frac{1}{2}}\boldsymbol{U} \tag{A.54b}$$

$$\therefore \boldsymbol{Q} = \boldsymbol{W}_{11}^{\frac{1}{2}} \tag{A.54c}$$

Appendix B

By using the solution of $\boldsymbol{Q}$, we can find $\boldsymbol{b}$,

$$2\boldsymbol{Q}\boldsymbol{b} = \boldsymbol{s}_{12} \tag{A.55a}$$

$$\boldsymbol{b} = \frac{1}{2}\boldsymbol{Q}^{-1}\boldsymbol{s}_{12} \tag{A.55b}$$

$$\therefore \boldsymbol{b} = \frac{1}{2}\boldsymbol{W}_{11}^{-\frac{1}{2}}\boldsymbol{s}_{12} \tag{A.55c}$$

(A.50) is referred to as the penalized least-squares problem, often referred to as Lasso.

Appendix C

Binary Classification Results for the Covariance matrix vs covariance matrix with randomly selected genes

Covariance

% CORRECT

	1	2	3	4	5	6	7	8	9	10	11
1		88.20	53.75	95.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00
2	88.22		94.47	94.44	100.00	99.95	100.00	100.00	100.00	100.00	100.00
3	53.82	94.68		95.39	100.00	99.88	98.63	100.00	100.00	100.00	100.00
4	95.17	95.00	95.09		99.45	100.00	88.32	99.91	100.00	99.94	99.94
5	100.00	100.00	100.00	99.43		99.95	100.00	100.00	100.00	100.00	100.00
6	100.00	99.90	99.88	100.00	99.95		100.00	100.00	100.00	100.00	100.00
7	100.00	100.00	98.75	88.68	100.00	100.00		100.00	100.00	100.00	100.00
8	100.00	100.00	100.00	99.91	100.00	100.00	100.00		55.56	62.50	62.50
9	100.00	100.00	100.00	99.88	100.00	100.00	100.00	55.59		57.14	57.14
10	100.00	100.00	100.00	99.94	100.00	100.00	100.00	62.50	57.14		49.89
11	100.00	100.00	100.00	99.84	100.00	100.00	100.00	62.50	57.14	49.89	

Random Effect

% CORRECT

	1	2	3	4	5	6	7	8	9	10	11
1		55.75	72.12	92.02	99.81	71.33	62.90	45.02	82.05	92.39	93.56
2	55.55		64.84	87.78	100.00	79.33	59.87	66.82	96.94	98.24	98.88
3	72.47	65.51		91.78	99.27	72.50	59.02	81.92	93.62	94.74	95.24
4	91.74	87.65	91.66		87.13	94.12	93.83	93.99	93.79	94.16	93.71
5	99.83	99.99	99.22	87.52		100.00	100.00	100.00	100.00	100.00	100.00
6	70.64	80.51	73.58	93.90	100.00		85.27	69.23	97.89	99.50	100.00
7	61.83	59.33	58.75	93.94	100.00	86.13		80.80	96.41	100.00	100.00
8	44.31	66.92	82.54	94.20	100.00	69.13	80.07		100.00	100.00	100.00
9	82.23	96.92	92.84	93.50	100.00	97.70	96.26	100.00		58.71	80.00
10	91.22	98.18	94.57	93.51	100.00	99.67	100.00	100.00	58.05		53.83
11	92.89	99.06	95.48	93.57	100.00	100.00	100.00	100.00	79.95	54.94	

Appendix C

Binary *Classification Results for the* Precision (ρ=1) *vs* Precision (ρ=1) *with randomly selected genes*

% CORRECT

	1	2	3	4	5	6	7	8	9	10	11
1		96.16	73.37	94.90	100.00	99.90	100.00	100.00	100.00	100.00	100.00
2	96.67		93.75	95.13	100.00	100.00	100.00	100.00	100.00	100.00	100.00
3	74.50	94.25		94.13	100.00	98.21	98.83	100.00	100.00	100.00	100.00
4	94.77	95.00	93.98		97.98	95.65	95.13	96.58	96.50	96.17	96.24
5	100.00	100.00	100.00	98.04		100.00	100.00	100.00	100.00	100.00	100.00
6	99.93	100.00	98.15	95.28	100.00		100.00	100.00	100.00	100.00	100.00
7	99.98	100.00	98.92	95.06	100.00	100.00		100.00	100.00	100.00	100.00
8	100.00	100.00	100.00	96.22	100.00	100.00	100.00		100.00	100.00	100.00
9	100.00	100.00	100.00	96.26	100.00	100.00	100.00	100.00		100.00	100.00
10	100.00	100.00	100.00	95.76	100.00	100.00	100.00	100.00	100.00		100.00
11	100.00	100.00	100.00	96.60	100.00	100.00	100.00	100.00	100.00	100.00	

Random Effect

% CORRECT

	1	2	3	4	5	6	7	8	9	10	11
1		96.08	72.15	94.07	98.58	99.07	97.81	98.21	97.87	97.42	97.11
2	95.96		94.40	94.87	100.00	100.00	100.00	100.00	100.00	100.00	100.00
3	72.63	94.51		94.03	100.00	98.58	98.35	100.00	100.00	100.00	100.00
4	93.72	94.88	94.06		98.22	95.77	95.26	96.42	96.24	96.06	96.44
5	98.51	100.00	100.00	98.12		100.00	100.00	100.00	100.00	100.00	100.00
6	98.95	100.00	98.08	95.74	100.00		100.00	100.00	100.00	100.00	100.00
7	97.26	100.00	98.85	94.81	100.00	100.00		100.00	100.00	100.00	100.00
8	97.52	100.00	100.00	96.33	100.00	100.00	100.00		100.00	100.00	100.00
9	98.46	100.00	100.00	96.41	100.00	100.00	100.00	100.00		100.00	100.00
10	97.42	100.00	100.00	96.25	100.00	100.00	100.00	100.00	100.00		100.00
11	97.22	100.00	100.00	96.27	100.00	100.00	100.00	100.00	100.00	100.00	

Appendix C

Binary Classification Results for the Precision (ρ=3) vs Precision (ρ=3) with randomly selected genes

% CORRECT

	1	2	3	4	5	6	7	8	9	10	11
1		91.02	69.75	94.85	100.00	95.48	100.00	100.00	100.00	100.00	100.00
2	91.53		90.60	94.99	100.00	100.00	100.00	100.00	100.00	100.00	100.00
3	69.18	90.61		94.43	100.00	96.29	99.44	100.00	100.00	100.00	100.00
4	94.78	94.50	94.36		99.01	96.06	97.96	98.71	99.64	99.51	99.89
5	100.00	100.00	100.00	99.03		100.00	100.00	98.40	98.09	98.00	98.28
6	95.38	100.00	95.60	96.88	100.00		100.00	100.00	100.00	100.00	100.00
7	100.00	100.00	99.40	97.64	100.00	100.00		100.00	100.00	100.00	100.00
8	100.00	100.00	100.00	98.33	98.53	100.00	100.00		100.00	100.00	100.00
9	100.00	100.00	100.00	99.30	98.16	100.00	100.00	100.00		99.95	100.00
10	100.00	100.00	100.00	99.56	98.13	100.00	100.00	100.00	99.71		100.00
11	100.00	100.00	100.00	99.86	98.28	100.00	100.00	100.00	100.00	100.00	

Random Effect

% CORRECT

	1	2	3	4	5	6	7	8	9	10	11
1		86.76	72.07	95.28	98.86	96.00	100.00	100.00	100.00	100.00	100.00
2	86.90		92.21	95.58	98.67	94.90	100.00	100.00	100.00	100.00	100.00
3	71.75	91.88		92.00	97.10	94.35	96.00	97.48	97.02	97.12	97.45
4	95.15	96.01	91.68		97.40	95.17	95.41	97.48	97.56	97.52	97.30
5	98.75	98.64	96.92	97.02		98.40	98.57	98.23	98.56	98.19	98.09
6	96.00	95.54	94.85	95.36	98.23		100.00	100.00	100.00	100.00	100.00
7	100.00	100.00	95.79	95.41	98.30	100.00		100.00	100.00	100.00	100.00
8	100.00	100.00	96.69	97.23	98.40	100.00	100.00		100.00	100.00	100.00
9	100.00	100.00	97.07	97.73	98.25	100.00	100.00	100.00		100.00	100.00
10	100.00	100.00	96.90	97.60	98.39	100.00	100.00	100.00	100.00		100.00
11	100.00	100.00	97.29	97.92	98.13	100.00	100.00	100.00	100.00	100.00	

Appendix D

Long-short Portfolio Results

Empirical and Realized Risk

2 year S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.80	0.05	5.63	0.06	**6.12**	0.07	53.44	59.53
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.82	0.04	5.74	0.05	6.13	0.07**	66.67	74.75
GLasso$^{\text{MAX LIKELIHOOD}}$	5.75	0.03	5.85	0.05	6.19	0.07	72.85	81.94
GLasso$^{\text{ORACLE}}$	5.69	0.02	5.92	0.06	6.25	0.07	75.11	84.57

Table D.1 Long-short portfolio 2 years empirical and realized risks

1 year S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.68	0.08	5.54	0.17	6.27	0.11	59.70	65.16
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.79	0.06	5.46	0.09	**6.19**	0.10**	57.22	62.22
GLasso$^{\text{MAX LIKELIHOOD}}$	5.68	0.04	5.94	0.09	6.41	0.12**	77.17	84.74
GLasso$^{\text{ORACLE}}$	5.73	0.04	5.85	0.09	6.33	0.11	75.11	82.46

Table D.2 Long-short portfolio 1 year empirical and realized risks

9 months S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.67	0.05	5.41	0.12	6.44	0.07	64.09	69.33
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.64	0.05	5.15	0.05	**6.42**	0.07**	53.35	57.33
GLasso$^{\text{MAX LIKELIHOOD}}$	5.60	0.04	5.92	0.08	6.65	0.08**	79.01	85.89
GLasso$^{\text{ORACLE}}$	5.71	0.04	5.70	0.06	6.50	0.07	75.06	81.52

Table D.3 Long-short portfolio 9 months empirical and realized risks

Appendix D

6 months S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.49	0.07	5.19	0.15	6.71	0.11**	64.22	68.55
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.36	0.06	4.65	0.08	**6.63**	0.09	48.74	51.73
GLasso$^{\text{MAX LIKELIHOOD}}$	5.57	0.04	5.83	0.09	6.79	0.09**	79.58	85.38
GLasso$^{\text{ORACLE}}$	5.65	0.04	5.51	0.09	6.60	0.09	75.09	80.43

Table D.4 Long-short portfolio 6 months empirical and realized risks

3 months S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	4.91	0.13	4.13	0.21	7.32	0.12**	59.42	62.34
GLasso$^{\text{MAX LIKELIHOOD-2}}$	4.58	0.05	3.29	0.06	7.24	0.10	44.46	46.44
GLasso$^{\text{MAX LIKELIHOOD}}$	5.40	0.03	5.87	0.09	7.47	0.13**	83.21	87.67
GLasso$^{\text{ORACLE}}$	5.47	0.04	4.97	0.07	**7.04**	0.11	75.09	79.07

Table D.5 Long-short portfolio 3 months empirical and realized risks

2 months S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	~~4.46~~	~~0.13~~	~~3.31~~	~~0.19~~	~~7.60~~	~~0.12~~	~~60.53~~	~~63.07~~
GLasso$^{\text{MAX LIKELIHOOD-2}}$	3.89	0.03	2.23	0.05	7.57	0.12	45.81	47.55
GLasso$^{\text{MAX LIKELIHOOD}}$	5.34	0.03	5.66	0.06	7.87	0.11**	84.40	88.22
GLasso$^{\text{ORACLE}}$	5.28	0.04	4.38	0.06	**7.39**	0.11	75.10	78.46

Table D.6 Long-short portfolio 2 months empirical and realized risks

Appendix D

1 month S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	~~3.20~~	~~0.11~~	~~1.52~~	~~0.10~~	~~8.54~~	~~0.14~~	~~62.13~~	~~64.09~~
GLasso$^{\text{MAX LIKELIHOOD-2}}$	~~2.69~~	~~0.02~~	~~0.78~~	~~0.02~~	~~8.54~~	~~0.14~~	~~55.73~~	~~57.45~~
GLasso$^{\text{MAX LIKELIHOOD}}$	~~2.69~~	~~0.02~~	~~0.78~~	~~0.02~~	~~8.54~~	~~0.14~~	~~55.73~~	~~57.45~~
GLasso$^{\text{ORACLE}}$	4.63	0.03	2.98	0.04	**8.50**	0.14	75.11	77.58

Table D.7 Long-short portfolio 1 month empirical and realized risks

Appendix D

Empirical and Realized Likelihood

2 year S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	**6.12**	0.07	53.44	59.53	63.25	107779.65	109590.02
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.13	0.07	66.67	74.75	55.69	107929.11	109179.59
GLasso$^{\text{MAX LIKELIHOOD}}$	6.19	0.07	72.85	81.94	52.40	**107930.82**	108926.92
GLasso$^{\text{ORACLE}}$	6.25	0.07	75.11	84.57	51.19	107911.28	108815.31

Table D.8 Long-short portfolio 2 years empirical and realized likelihoods

1 year S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.27	0.11	59.70	65.16	54.73	53668.56	55178.46
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.19**	0.10	57.22	62.22	55.89	53737.92	55290.66
GLasso$^{\text{MAX LIKELIHOOD}}$	6.41	0.12	77.17	84.74	43.08	53854.40	54590.93
GLasso$^{\text{ORACLE}}$	6.33	0.11	75.11	82.46	44.55	**53864.81**	54681.56

Table D.9 Long-short portfolio 1 year empirical and realized likelihoods

9 months S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.44	0.07	64.09	69.33	49.41	40205.95	41510.30
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.42**	0.07	53.35	57.33	56.70	40096.36	41855.82
GLasso$^{\text{MAX LIKELIHOOD}}$	6.65	0.08	79.01	85.89	38.97	40303.79	40995.59
GLasso$^{\text{ORACLE}}$	6.50	0.07	75.06	81.52	41.82	**40315.34**	41160.91

Table D.10 Long-short portfolio 9 months empirical and realized likelihoods

Appendix D

6 months S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.71	0.11	64.22	68.55	46.42	26624.48	27940.22
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.63**	0.09	48.74	51.73	58.39	26443.48	28429.58
GLasso$^{\text{MAX LIKELIHOOD}}$	6.79	0.09	79.58	85.38	34.96	**26826.03**	27434.59
GLasso$^{\text{ORACLE}}$	6.60	0.09	75.09	80.43	38.22	26825.81	27609.77

Table D.11 Long-short portfolio 6 months empirical and realized likelihoods

3 months S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	7.32	0.12	59.42	62.34	47.03	12657.11	14599.16
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.24	0.10	44.46	46.44	59.69	12566.99	15053.35
GLasso$^{\text{MAX LIKELIHOOD}}$	7.47	0.13	83.21	87.67	26.97	**13333.21**	13784.95
GLasso$^{\text{ORACLE}}$	**7.04**	0.11	75.09	79.07	33.94	13321.53	14064.92

Table D.12 Long-short portfolio 3 months empirical and realized likelihoods

2 months S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	~~7.60~~	~~0.12~~	~~60.53~~	~~63.07~~	~~44.65~~	~~Inf~~	~~Inf~~
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.57	0.12	45.81	47.55	57.51	7707.04	10594.81
GLasso$^{\text{MAX LIKELIHOOD}}$	7.87	0.11	84.40	88.22	23.72	8845.52	9251.25
GLasso$^{\text{ORACLE}}$	**7.39**	0.11	75.10	78.46	31.97	8815.23	9554.85

Table D.13 Long-short portfolio 2 months empirical and realized likelihoods

Appendix D

1 month S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	~~8.54~~	~~0.14~~	~~62.13~~	~~64.09~~	~~41.13~~	~~Inf~~	~~Inf~~
GLasso$^{\text{MAX LIKELIHOOD-2}}$	~~8.54~~	~~0.14~~	~~55.73~~	~~57.45~~	~~47.08~~	~~Inf~~	~~Inf~~
GLasso$^{\text{MAX LIKELIHOOD}}$	~~8.54~~	~~0.14~~	~~55.73~~	~~57.45~~	~~47.08~~	~~Inf~~	~~Inf~~
GLasso$^{\text{ORACLE}}$	**8.50**	0.14	75.11	77.58	29.20	4303.05	5023.61

Table D.14 Long-short portfolio 1 month empirical and realized likelihoods

Appendix D

Long-only Portfolio Results

Empirical and Realized Risk

2 year N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.41	0.07	6.40	0.03	6.75	0.08	54.16	60.43
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.40	0.05	6.45	0.05	**6.74**	0.08**	66.67	74.75
GLasso$^{\text{MAX LIKELIHOOD}}$	6.32	0.04	6.52	0.06	6.79	0.08	72.85	81.94
GLasso$^{\text{ORACLE}}$	6.25	0.03	6.57	0.06	6.83	0.09	75.11	84.57

Table D.15 Long-only portfolio 2 years empirical and realized risks

1 year N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.35	0.08	6.32	0.09	6.85	0.12	63.35	69.18
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.38	0.07	6.23	0.07	**6.79**	0.10**	57.22	62.22
GLasso$^{\text{MAX LIKELIHOOD}}$	6.20	0.06	6.55	0.07	6.98	0.11**	77.17	84.74
GLasso$^{\text{ORACLE}}$	6.26	0.05	6.48	0.07	6.92	0.10	75.11	82.46

Table D.16 Long-only portfolio 1 year empirical and realized risks

9 months N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.21	0.10	6.11	0.13	6.98	0.12	57.83	62.50
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.28	0.09	6.03	0.10	**6.92**	0.12**	53.35	57.33
GLasso$^{\text{MAX LIKELIHOOD}}$	6.10	0.07	6.53	0.10	7.14	0.12**	79.01	85.89
GLasso$^{\text{ORACLE}}$	6.23	0.07	6.37	0.09	7.01	0.12	75.06	81.52

Table D.17 Long-only portfolio 9 months empirical and realized risks

Appendix D

6 months N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	6.05	0.09	5.97	0.14	7.16	0.14**	62.73	66.88
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.04	0.09	5.66	0.12	**7.06**	0.12**	48.74	51.73
GLasso$^{\text{MAX LIKELIHOOD}}$	6.03	0.07	6.43	0.10	7.26	0.13**	79.58	85.38
GLasso$^{\text{ORACLE}}$	6.15	0.07	6.19	0.11	7.11	0.12	75.09	80.43

Table D.18 Long-only portfolio 6 months empirical and realized risks

3 months N.S.S	Predicted Risk(%)	std dev	Empirical risk (%)	std dev	Realized risk(%)	std dev	Sparsity (%)	Zero Overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	5.46	0.11	5.10	0.17	7.76	0.13**	56.05	58.82
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.40	0.08	4.67	0.10	7.62	0.12	44.46	46.44
GLasso$^{\text{MAX LIKELIHOOD}}$	5.77	0.05	6.36	0.10	7.88	0.14**	83.21	87.67
GLasso$^{\text{ORACLE}}$	5.94	0.06	5.70	0.10	**7.53**	0.12	75.09	79.07

Table D.19 Long-only portfolio 3 months empirical and realized risks

2 months N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	~~4.99~~	~~0.11~~	~~4.25~~	~~0.15~~	~~8.13~~	~~0.12~~	~~57.04~~	~~59.36~~
GLasso$^{\text{MAX LIKELIHOOD-2}}$	4.72	0.06	3.72	0.09	8.14	0.13	45.81	47.55
GLasso$^{\text{MAX LIKELIHOOD}}$	5.67	0.05	6.11	0.07	8.24	0.13**	84.40	88.22
GLasso$^{\text{ORACLE}}$	5.72	0.05	5.13	0.08	**7.90**	0.12	75.10	78.46

Table D.20 Long-only portfolio 2 months empirical and realized risks

Appendix D

1 month N.S.S	Predicted Risk(%)	std dev	Empirical Risk (%)	std dev	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)
GLasso$^{\text{MIN REALIZED RISK}}$	~~3.97~~	~~0.11~~	~~2.59~~	~~0.12~~	~~9.12~~	~~0.14~~	~~65.15~~	~~67.21~~
GLasso$^{\text{MAX LIKELIHOOD-2}}$	~~3.25~~	~~0.05~~	~~1.66~~	~~0.06~~	~~9.23~~	~~0.15~~	~~55.73~~	~~57.45~~
GLasso$^{\text{MAX LIKELIHOOD}}$	~~3.25~~	~~0.05~~	~~1.66~~	~~0.06~~	~~9.23~~	~~0.15~~	~~55.73~~	~~57.45~~
GLasso$^{\text{ORACLE}}$	4.99	0.05	3.65	0.07	**8.88**	0.14	75.11	77.58

Table D.21 Long-only portfolio 1 month empirical and realized risks

Appendix D

Empirical and Realized Likelihood

2 year N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.75	0.08	54.16	60.43	63.07	107762.24	109537.14
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.74**	0.08	66.67	74.75	55.69	107929.11	109179.59
GLasso$^{\text{MAX LIKELIHOOD}}$	6.79	0.08	72.85	81.94	52.40	**107930.82**	108926.92
GLasso$^{\text{ORACLE}}$	6.83	0.09	75.11	84.57	51.19	107911.28	108815.31

Table D.22 Long-only portfolio 2 years empirical and realized likelihoods

1 year N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.85	0.12	63.35	69.18	52.05	53780.21	55081.61
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.79**	0.10	57.22	62.22	55.89	53737.92	55290.66
GLasso$^{\text{MAX LIKELIHOOD}}$	6.98	0.11	77.17	84.74	43.08	53854.40	54590.93
GLasso$^{\text{ORACLE}}$	6.92	0.10	75.11	82.46	44.55	**53864.81**	54681.56

Table D.23 Long-only portfolio 1 year empirical and realized likelihoods

9 months N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	6.98	0.12	57.83	62.50	54.20	40036.72	41693.77
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**6.92**	0.12	53.35	57.33	56.70	40096.36	41855.82
GLasso$^{\text{MAX LIKELIHOOD}}$	7.14	0.12	79.01	85.89	38.97	40303.79	40995.59
GLasso$^{\text{ORACLE}}$	7.01	0.12	75.06	81.52	41.82	**40315.34**	41160.91

Table D.24 Long-only portfolio 9 months empirical and realized likelihoods

Appendix D

6 months N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	7.16	0.14	62.73	66.88	47.42	26592.77	27980.61
GLasso$^{\text{MAX LIKELIHOOD-2}}$	**7.06**	0.12	48.74	51.73	58.39	26443.48	28429.58
GLasso$^{\text{MAX LIKELIHOOD}}$	7.26	0.13	79.58	85.38	34.96	**26826.03**	27434.59
GLasso$^{\text{ORACLE}}$	7.11	0.12	75.09	80.43	38.22	26825.81	27609.77

Table D.25 Long-only portfolio 6 months empirical and realized likelihoods

3 months N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	7.76	0.13	56.05	58.82	50.09	12464.87	14720.15
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.62	0.12	44.46	46.44	59.69	12566.99	15053.35
GLasso$^{\text{MAX LIKELIHOOD}}$	7.88	0.14	83.21	87.67	26.97	**13333.21**	13784.95
GLasso$^{\text{ORACLE}}$	**7.53**	0.12	75.09	79.07	33.94	13321.53	14064.92

Table D.26 Long-only portfolio 3 months empirical and realized likelihoods

2 months N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	~~8.13~~	~~0.12~~	~~57.04~~	~~59.36~~	~~47.60~~	~~Inf~~	~~Inf~~
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.14	0.13	45.81	47.55	57.51	7707.04	10594.81
GLasso$^{\text{MAX LIKELIHOOD}}$	8.24	0.13	84.40	88.22	23.72	8845.52	9251.25
GLasso$^{\text{ORACLE}}$	**7.90**	0.12	75.10	78.46	31.97	8815.23	9554.85

Table D.27 Long-only portfolio 2 months empirical and realized likelihoods

Appendix D

1 month N.S.S	Realized Risk(%)	std dev	Sparsity (%)	Zero-overlap (%)	Non-zero overlap (%)	Realized Likelihood	Empirical Likelihood
GLasso$^{\text{MIN REALIZED RISK}}$	~~9.12~~	~~0.14~~	~~65.15~~	~~67.21~~	~~38.30~~	~~Inf~~	~~Inf~~
GLasso$^{\text{MAX LIKELIHOOD-2}}$	~~9.23~~	~~0.15~~	~~55.73~~	~~57.45~~	~~47.08~~	~~Inf~~	~~Inf~~
GLasso$^{\text{MAX LIKELIHOOD}}$	~~9.23~~	~~0.15~~	~~55.73~~	~~57.45~~	~~47.08~~	~~Inf~~	~~Inf~~
GLasso$^{\text{ORACLE}}$	**8.88**	0.14	75.11	77.58	29.20	4303.05	5023.61

Table D.28 Long-only portfolio 1 month empirical and realized likelihoods

Appendix E

Statistical Hypothesis test (*t*-test) to test for the difference between two means

The *t*-test is the statistical hypothesis test used in this thesis to determine if the mean value for two different groups are statistically significantly different from each other. We let $\bar{x}_1$ represent the mean value of group 1, $\bar{x}_2$ represent the mean value of group 2, N_1 be the sample size of group 1, N_2 be the sample size group 2, s_1 be the standard deviation of group 1 values and s_2 be the standard deviation of group 2 values.

Null hypothesis: $H_0: \bar{x}_1 - \bar{x}_2 = 0$

Alternative hypothesis: $H_A: \bar{x}_1 - \bar{x}_2 \neq 0$

$$t\text{-statistic for independent means (pooled)} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\left(\frac{(N_1-1)s_1^2+(N_2-1)s_2^2}{N_1+N_2-2}\right)\left(\frac{1}{N_1}+\frac{1}{N_2}\right)}}$$

If the two standard deviations for group 1 and group 2 are not similar (one is more than twice of the other), then the unpooled *t*-statistic is used.

$$t\text{-statistic for independent means (unpooled)} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\left(\frac{s_1^2}{N_1}+\frac{s_2^2}{N_2}\right)}}$$

The *t*-statistic is used to determine the p-value. Experiments are performed at significance levels of $\alpha = 1\%$ and $\alpha = 5\%$. Results statistically significant at the 1%

level are represented with the symbol '**' while results statistically significant at the 5% level are represented with the symbol '*'.

	Decision
$p \leq \alpha$	▪ Result is statistically significant; reject H_0
$p > \alpha$	▪ Result is not statistically significant; fail to reject H_0

where p is the p-value.

Appendix E

Chapter 2 *t*-test results

Model 1 Results

A sparse model: $(\mathbf{X})_{ii} = 1$, $(\mathbf{X})_{i,i-1} = (\mathbf{X})_{i-1,i} = 0.5$, and 0 otherwise.

Sample Size	GLasso Error	std	Modified GLasso Error	std	Pseudoinverse Error	std	t-stat	p-value
5	3.24	0.22	3.16	0.24	3.66	0.34	-0.58963	0.71415
10	2.21	0.44	2.45	0.46	395.29	2367.99	1.16391	0.12983
20	1.61	0.27	1.64	0.50	9.95	5.46	0.24744	0.40295
30	1.36	0.23	1.24	0.27	4.44	1.63	-1.87668	0.96720
40	1.22	0.19	1.11	0.23	2.95	0.94	-2.25225	0.98644
50	1.04	0.16	1.05	0.24	2.46	0.66	0.24279	0.40434
60	0.98	0.13	1.02	0.24	2.08	0.50	0.97522	0.16572
70	0.91	0.13	0.97	0.20*	1.78	0.44	2.03894	0.02168
80	0.87	0.14	0.92	0.19*	1.52	0.35	1.79007	0.03768
90	0.82	0.13	0.88	0.20**	1.43	0.33	2.43523	0.00793
100	0.77	0.12	0.89	0.20**	1.37	0.31	5.04824	0.00000

Table E.1 Error between Graphical Lasso and Modified Graphical Lasso (p =10)

Sample size	GLasso (Optimal ρ)	Modified GLasso (Off-Diagonal Optimal ρ)	Modified GLasso (Diagonal Optimal ρ)
5	2.83	7.51	1.51
10	0.58	1.41	0.25
20	0.24	0.45	0.02
30	0.17	0.19	0.01
40	0.13	0.12	0.01
50	0.10	0.08	0.01
60	0.08	0.06	0.01
70	0.07	0.04	0.01
80	0.07	0.03	0.01
90	0.06	0.03	0.01
100	0.05	0.02	0.01

Table E.2 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=10)

Appendix E

Sample Size	GLasso Error	std	Modified GLasso Error	std	Pseudoinverse Error	std	t-statistic	p-value
5	6.27	0.16	6.21	0.19	6.66	0.01	-0.51470	0.68966
10	4.77	0.44	4.66	0.44	6.49	0.04	-0.58364	0.71665
20	3.80	0.30	3.37	0.37	10.71	3.13	-4.10819	0.99990
30	3.20	0.24	2.56	0.24	1364.39	3245.91	-10.16021	1.00000
40	2.83	0.29	2.12	0.24	61.91	28.42	-11.78619	1.00000
50	2.57	0.23	1.87	0.18	23.75	5.49	-16.90823	1.00000
60	2.41	0.22	1.69	0.17	14.30	2.42	-20.15201	1.00000
70	2.28	0.22	1.57	0.14	10.44	1.61	-22.25229	1.00000
80	2.10	0.20	1.47	0.12	8.64	1.10	-23.74518	1.00000
90	1.99	0.20	1.42	0.12	7.18	0.86	-22.93527	1.00000
100	1.85	0.18	1.35	0.11	6.18	0.67	-23.39206	1.00000

Table E.3 Error between Graphical Lasso and Modified Graphical Lasso (p =30)

Sample size	GLasso (Optimal ρ)	Modified GLasso (Off-Diagonal Optimal ρ)	Modified GLasso (Diagonal Optimal ρ)
5	7.19	20.53	4.02
10	1.10	1.54	0.27
20	0.47	0.70	0.01
30	0.32	0.41	0.01
40	0.25	0.28	0.01
50	0.21	0.21	0.01
60	0.19	0.17	0.01
70	0.17	0.14	0.01
80	0.15	0.12	0.01
90	0.13	0.10	0.01
100	0.12	0.09	0.01

Table E.4 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=30)

Appendix E

Sample Size	GLasso Error	std	Modified GLasso Error	std	Pseudoinverse Error	std	t-statistic	p-value
5	8.29	0.15	8.25	0.17	8.63	0.00	-0.43453	0.66231
10	6.77	0.54	6.46	0.20	8.59	0.01	-1.71174	0.94794
20	5.42	0.40	5.12	0.38	8.20	0.04	-2.41614	0.98970
30	4.79	0.35	4.11	0.30	10.84	1.43	-8.11342	1.00000
40	4.34	0.44	3.32	0.38	51.88	14.05	-11.09239	1.00000
50	3.98	0.30	2.91	0.17	2854.55	1916.77	-21.98789	1.00000
60	3.66	0.31	2.58	0.30	144.90	67.82	-19.60404	1.00000
70	3.36	0.10	2.26	0.16	53.71	8.60	-48.69923	1.00000
80	3.25	0.15	2.08	0.16	30.70	3.75	-48.80764	1.00000
90	3.12	0.22	1.95	0.13	23.42	2.67	-43.87422	1.00000
100	2.86	0.25	1.84	0.14	17.90	1.58	-35.22075	1.00000

Table E.5 Error between Graphical Lasso and Modified Graphical Lasso (p =50)

Sample size	GLasso (Optimal ρ)	Modified GLasso (Off-Diagonal Optimal ρ)	Modified GLasso (Diagonal Optimal ρ)
5	12.57	29.55	7.85
10	1.52	1.91	0.24
20	0.61	1.02	0.01
30	0.44	0.62	0.01
40	0.35	0.41	0.01
50	0.30	0.33	0.01
60	0.24	0.25	0.01
70	0.21	0.21	0.01
80	0.20	0.18	0.01
90	0.19	0.16	0.01
100	0.16	0.13	0.01

Table E.6 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=50)

Appendix E

Sample Size	GLasso Error	std	Modified GLasso Error	std	Pseudoinverse Error	std	t-statistic	p-value
5	9.90	0.15	9.85	0.20	10.22	0.00	-0.37339	0.64072
10	8.62	0.38	7.94	0.14	10.21	0.00	-5.22296	0.99997
20	7.19	0.43	6.53	0.15	10.02	0.00	-6.48907	1.00000
30	6.32	0.17	5.73	0.38	9.62	0.04	-7.70994	1.00000
40	5.21	0.14	4.28	0.19	11.54	0.25	-24.79665	1.00000
50	5.46	0.04	4.16	0.40	24.85	3.07	-22.77702	1.00000
60	4.92	0.29	3.51	0.15	133.67	51.45	-33.03166	1.00000
70	4.22	0.23	3.04	0.36	4863.33	2779.00	-23.15923	1.00000
80	4.02	0.28	2.74	0.27	264.87	50.26	-29.61906	1.00000
90	3.98	0.19	2.60	0.14	80.59	11.33	-55.48635	1.00000
100	3.90	0.11	2.52	0.13	55.01	8.91	-80.39420	1.00000

Table E.7 Error between Graphical Lasso and Modified Graphical Lasso (p =70)

Sample size	GLasso (Optimal ρ)	Modified GLasso (Off-Diagonal Optimal ρ)	Modified GLasso (Diagonal Optimal ρ)
5	14.70	42.61	9.70
10	1.98	2.44	0.08
20	0.84	1.22	0.01
30	0.56	0.86	0.01
40	0.37	0.51	0.01
50	0.38	0.44	0.01
60	0.32	0.34	0.01
70	0.24	0.28	0.01
80	0.21	0.21	0.01
90	0.21	0.21	0.01
100	0.21	0.20	0.01

Table E.8 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=70)

Appendix E

Model 2 Results

A dense model: $(\mathbf{X})_{ii} = 2$, $(\mathbf{X})_{ii'} = 1$ otherwise

Sample Size	GLasso Error	std	Modified GLasso Error	std	Pseudoinverse Error	std	t-statistic	p-value
5	10.22	0.17	10.07	0.14	11.32	0.47	-1.52302	0.91687
10	9.98	0.31	10.01	0.38	575.76	2880.04	0.19345	0.42439
20	9.88	0.23	9.91	0.11	22.33	15.36	0.52623	0.30089
30	9.80	0.22	9.90	0.09	9.48	6.07	2.30429	0.01240
40	9.65	0.31	9.91	0.08	6.72	3.64**	5.13619	0.00000
50	9.32	0.59	9.83	0.41	5.46	2.75**	5.01933	0.00000
60	8.86	1.09	9.82	0.47	4.53	2.18**	6.26457	0.00000
70	8.64	1.19	9.83	0.15	3.74	1.85**	8.30091	0.00000
80	7.94	1.51	9.70	0.65	3.32	1.47**	9.57562	0.00000
90	7.06	1.55	9.54	0.95	2.97	1.15**	12.94158	0.00000
100	6.81	1.54	9.18	1.37	2.89	1.08**	11.49822	0.00000

Table E.9 Error between Graphical Lasso and Modified Graphical Lasso (p =10)

Sample size	GLasso (Optimal ρ)	Modified GLasso (Off-Diagonal Optimal ρ)	Modified GLasso (Diagonal Optimal ρ)
5	0.61	1.92	0.28
10	0.36	0.93	0.10
20	0.24	0.69	0.02
30	0.18	0.58	0.01
40	0.14	0.52	0.01
50	0.09	0.42	0.01
60	0.07	0.36	0.01
70	0.06	0.29	0.01
80	0.04	0.24	0.01
90	0.02	0.19	0.01
100	0.02	0.15	0.01

Table E.10 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=10)

Appendix E

Sample Size	GLasso Error	std	Modified GLasso Error	std	Pseudoinverse Error	std	t-statistic	p-value
5	30.19	0.13	30.09	0.08	31.44	0.00	-1.46842	0.90991
10	30.19	0.14	30.04	0.16	31.37	0.01	-2.19657	0.97930
20	30.14	0.11	29.96	0.06	33.80	2.03	-6.81487	1.00000
30	30.13	0.09	29.96	0.05	2552.45	6364.53	-9.30352	1.00000
40	30.09	0.08	29.95	0.04	160.52	113.90	-9.57063	1.00000
50	30.07	0.08	29.96	0.04	78.30	41.67	-8.39318	1.00000
60	30.03	0.07	29.96	0.03	40.16	16.11	-7.58635	1.00000
70	30.06	0.06	29.97	0.03	28.01	11.42	-11.43926	1.00000
80	30.03	0.07	29.97	0.03	25.88	10.64	-7.92087	1.00000
90	30.02	0.08	29.97	0.03	20.61	8.05	-5.61654	1.00000
100	30.01	0.08	29.97	0.03	17.55	6.34	-4.93610	1.00000

Table E.11 Error between Graphical Lasso and Modified Graphical Lasso (p =30)

Sample size	GLasso (Optimal ρ)	Modified GLasso (Off-Diagonal Optimal ρ)	Modified GLasso (Diagonal Optimal ρ)
5	0.60	2.86	0.26
10	0.47	1.56	0.07
20	0.35	1.16	0.01
30	0.30	0.89	0.01
40	0.25	0.72	0.01
50	0.22	0.68	0.01
60	0.20	0.67	0.01
70	0.19	0.64	0.01
80	0.18	0.60	0.01
90	0.17	0.54	0.01
100	0.16	0.50	0.01

Table E.12 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=30)

Appendix E

Sample Size	GLasso Error	std	Modified GLasso Error	std	Pseudoinverse Error	std	t-statistic	p-value
5	50.21	0.10	50.11	0.07	51.47	0.00	-1.85269	0.94947
10	50.22	0.11	50.04	0.07	51.44	0.00	-4.36492	0.99981
20	50.21	0.07	49.97	0.05	51.36	0.03	-12.42399	1.00000
30	50.21	0.05	49.97	0.04	52.15	0.46	-21.59777	1.00000
40	50.17	0.08	49.97	0.03	75.92	14.70	-14.45059	1.00000
50	50.11	0.08	49.97	0.03	13557.13	47805.89	-12.02228	1.00000
60	50.09	0.04	49.98	0.02	362.30	158.79	-18.13701	1.00000
70	50.10	0.04	49.98	0.02	175.15	83.26	-23.02482	1.00000
80	50.11	0.03	49.98	0.02	102.75	33.24	-30.95300	1.00000
90	50.11	0.03	49.98	0.02	77.11	30.73	-30.40423	1.00000
100	50.11	0.03	49.98	0.02	61.31	21.26	-35.28482	1.00000

Table E.13 Error between Graphical Lasso and Modified Graphical Lasso (p =50)

Sample size	GLasso (Optimal ρ)	Modified GLasso (Off-Diagonal Optimal ρ)	Modified GLasso (Diagonal Optimal ρ)
5	0.63	3.46	0.27
10	0.50	1.90	0.06
20	0.39	1.21	0.01
30	0.35	1.08	0.01
40	0.29	0.81	0.01
50	0.24	0.69	0.01
60	0.22	0.67	0.01
70	0.21	0.66	0.01
80	0.20	0.66	0.01
90	0.20	0.62	0.01
100	0.19	0.58	0.01

Table E.14 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=50)

Appendix E

Sample Size	GLasso Error	std	Modified GLasso Error	std	Pseudoinverse Error	std	t-statistic	p-value
5	70.19	0.09	70.11	0.05	71.48	0.00	-1.82305	0.94712
10	70.22	0.12	70.03	0.06	71.47	0.00	-4.41110	0.99983
20	70.21	0.06	69.97	0.04	71.41	0.00	-15.60250	1.00000
30	70.24	0.04	69.97	0.03	71.37	0.02	-28.82724	1.00000
40	70.22	0.06	69.99	0.02	71.96	0.41	-22.67870	1.00000
50	70.13	0.07	69.98	0.02	77.73	1.97	-14.54820	1.00000
60	70.10	0.05	69.98	0.02	147.31	48.48	-16.46651	1.00000
70	70.11	0.04	69.98	0.02	11918.77	12649.67	-25.49720	1.00000
80	70.12	0.02	69.98	0.02	688.80	420.89	-44.59990	1.00000
90	70.13	0.02	69.99	0.02	302.43	80.86	-58.47538	1.00000
100	70.14	0.01	69.98	0.02	212.32	64.56	-73.38807	1.00000

Table E.15 Error between Graphical Lasso and Modified Graphical Lasso (p =70)

Sample size	GLasso (Optimal ρ)	Modified GLasso (Off-Diagonal Optimal ρ)	Modified GLasso (Diagonal Optimal ρ)
5	0.63	3.85	0.27
10	0.51	2.10	0.05
20	0.39	1.22	0.01
30	0.36	1.18	0.01
40	0.33	0.96	0.01
50	0.26	0.76	0.01
60	0.22	0.67	0.01
70	0.21	0.67	0.01
80	0.21	0.66	0.01
90	0.21	0.69	0.01
100	0.21	0.65	0.01

Table E.16 Optimal penalty for Graphical Lasso and Modified Graphical Lasso (p=70)

Appendix E

Chapter 4 *t*-test results for the synthetic data experiment

Hypothesis t-tests are performed to determine if the mean realized risk for the proposed Graphical Lasso strategies perform significantly differently from the baseline oracle method. We compare the mean realized results of the $GLasso^{ORACLE}$ method to all the other methods {$GLasso^{MIN\ REALIZED\ RISK}$, $GLasso^{MAX\ LIKELIHOOD-2}$, $GLasso^{MAX\ LIKELIHOOD}$} and present the results. $GLasso^{ORACLE}$ is treated as group 1 while the competing Graphical Lasso strategy is treated as group 2.

Long-Short Portfolio Results (S.S)

Rebalancing period	Method	*t*-statistic	p-value
2 months	GLasso ($C.V^{MIN\ REAL\ RISK}$)	8.10719	0.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD-2}$)	-7.07561	1.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD}$)	19.02842	0.00000
3 months	GLasso ($C.V^{MIN\ REAL\ RISK}$)	13.73468	0.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD-2}$)	-10.65159	1.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD}$)	20.36420	0.00000
6 months	GLasso ($C.V^{MIN\ REAL\ RISK}$)	8.47403	0.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD-2}$)	-2.04683	0.97914
	GLasso ($C.V^{MAX\ LIKELIHOOD}$)	15.70695	0.00000
9 months	GLasso ($C.V^{MIN\ REAL\ RISK}$)	-7.97720	1.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD-2}$)	10.77498	0.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD}$)	20.13784	0.00000
1 year	GLasso ($C.V^{MIN\ REAL\ RISK}$)	-5.73926	1.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD-2}$)	15.06257	0.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD}$)	7.88193	0.00000
2 years	GLasso ($C.V^{MIN\ REAL\ RISK}$)	-26.86037	1.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD-2}$)	25.15037	0.00000
	GLasso ($C.V^{MAX\ LIKELIHOOD}$)	-12.34491	1.00000

Table E.17 Synthetic data long-short portfolio hypothesis test results

Appendix E

Long-Only Portfolio Results

Rebalancing period	Method	*t*-statistic	p-value
2 months	GLasso ($C.V^{\text{MIN REAL RISK}}$)	8.53684	0.00000
	GLasso ($C.V^{\text{MAX LIKELIHOOD-2}}$)	-8.72794	1.00000
	GLasso ($C.V^{\text{MAX LIKELIHOOD}}$)	12.46691	0.00000
3 months	GLasso ($C.V^{\text{MIN REAL RISK}}$)	9.58933	0.00000
	GLasso ($C.V^{\text{MAX LIKELIHOOD-2}}$)	-4.00367	0.99995
	GLasso ($C.V^{\text{MAX LIKELIHOOD}}$)	14.50693	0.00000
6 months	GLasso ($C.V^{\text{MIN REAL RISK}}$)	2.93595	0.00182
	GLasso ($C.V^{\text{MAX LIKELIHOOD-2}}$)	3.65735	0.00016
	GLasso ($C.V^{\text{MAX LIKELIHOOD}}$)	9.45577	0.00000
9 months	GLasso ($C.V^{\text{MIN REAL RISK}}$)	-2.49365	0.99346
	GLasso ($C.V^{\text{MAX LIKELIHOOD-2}}$)	7.56108	0.00000
	GLasso ($C.V^{\text{MAX LIKELIHOOD}}$)	9.78853	0.00000
1 year	GLasso ($C.V^{\text{MIN REAL RISK}}$)	-7.20203	1.00000
	GLasso ($C.V^{\text{MAX LIKELIHOOD-2}}$)	14.13637	0.00000
	GLasso ($C.V^{\text{MAX LIKELIHOOD}}$)	6.70122	0.00000
2 years	GLasso ($C.V^{\text{MIN REAL RISK}}$)	-15.57531	1.00000
	GLasso ($C.V^{\text{MAX LIKELIHOOD-2}}$)	17.08919	0.00000
	GLasso ($C.V^{\text{MAX LIKELIHOOD}}$)	-7.95821	1.00000

Table E.18 Synthetic data long-only portfolio hypothesis test results

Appendix E

Chapter 3 *t*-test results

Hypothesis *t*-tests are performed to determine if the mean classification performance of the two estimated precisions are significantly different from the mean performance of the covariance classifier.

Class	Tissue	Precision (ρ=1)		Precision (ρ=3)	
		t-statistic	p-value	*t*-statistic	p-value
1	D4_MDDC	0.34179	0.36848	-1.4121	0.91146
2	DC	-1.3172	0.89553	-0.3337	0.62824
3	MDDC	2.91589	0.00427	4.98107	3.6E-05
4	MDM	11.2758	2.5E-13	9.74043	1.1E-11
5	BLOOD	-5.4881	1	-3.7477	0.99959
6	M3DC	-2.4472	0.97994	-2.5857	0.98384
7	AM	-0.7008	0.74834	-0.7396	0.75966
8	HC	4.22057	0.00146	2.30161	0.02517
9	HL	2.95773	0.01268	3.08466	0.01077
10	2T	3.01725	0.01963	2.45819	0.03491
11	HOS	2.72585	0.02633	2.56949	0.03101

Table E.19 One-versus-all LLS classification performance t-tests results of the significance in the difference between the mean performance of the covariance classifier and the mean performances of the precision classifiers

Class	Tissue	Precision (ρ=1)		Precision (ρ=3)	
		t-statistic	p-value	*t*-statistic	p-value
1	D4_MDDC	6.44474	4.1E-06	2.83599	0.00596
2	DC	-2.5437	0.9883	-2.5581	0.98862
3	MDDC	1.95605	0.03229	0.39355	0.34904
4	MDM	9.75153	1.1E-11	7.44124	6.2E-09
5	BLOOD	-13.635	1	-6.3282	1
6	M3DC	-1.2427	0.87543	-0.1099	0.54241
7	AM	-3.4443	0.99562	-2.2645	0.97333
8	HC	-1.0916	0.8466	-1.4132	0.90235
9	HL	0.14641	0.4442	-1.6123	0.92098
10	2T	-0.5034	0.67941	-2.5999	0.96997
11	HOS	-2.1356	0.95021	-1.678	0.91567

Table E.20 One-versus-all ridge regression classification performance t-tests results of the significance in the difference between the mean performance of the covariance classifier and the mean performances of the precision classifiers

Appendix E

Hypothesis *t*-tests are performed to determine if the mean classification performance of the covariance and precision classifiers with random gene replacement is significantly different.

Class	Tissue	Covariance with Random Effect	
		t-statistic	p-value
1	D4_MDDC	9.86774	1.7E-08
2	DC	1.95578	0.03537
3	MDDC	5.72722	6.6E-06
4	MDM	35.0352	0
5	BLOOD	14.0608	1.6E-14
6	M3DC	0.07543	0.47086
7	AM	-0.5487	0.70091
8	HC	10.0186	4.2E-06
9	HL	0.06833	0.47387
10	2T	0.1373	0.44871
11	HOS	-0.0209	0.50783

Table E.21 Ridge regression classification performance t-tests results when random genes are used to replace optimally selected genes in the covariance k-NN graph

Class	Tissue	Precision (ρ=1) with Random Effect	
		t-statistic	p-value
1	D4_MDDC	-0.0194	0.50762
2	DC	-0.1864	0.5726
3	MDDC	-0.0343	0.5135
4	MDM	-0.1804	0.57106
5	BLOOD	1.00724	0.16122
6	M3DC	-0.5864	0.71311
7	AM	0.35459	0.36603
8	HC	0.51693	0.3096
9	HL	0.1301	0.45037
10	2T	-0.2873	0.60593
11	HOS	0.05024	0.48117

Table E.22 Ridge regression classification performance t-tests results when random genes are used to replace optimally selected genes in the Precision (ρ=1) k-NN graph

Class	Tissue	Precision (ρ=3) with Random Effect	
		t-statistic	p-value
1	D4_MDDC	0.34179	0.36848
2	DC	-1.3172	0.89553
3	MDDC	2.91589	0.00427
4	MDM	11.2758	2.5E-13
5	BLOOD	-5.4881	1
6	M3DC	-2.4472	0.97994
7	AM	-0.7008	0.74834
8	HC	4.22057	0.00146
9	HL	2.95773	0.01268
10	2T	3.01725	0.01963
11	HOS	2.72585	0.02633

Table E.23 Ridge regression classification performance t-tests results when random genes are used to replace optimally selected genes in the Precision (ρ=3) k-NN graph

Appendix F

Long-short Portfolio Results

Realized Risk, Portfolio Return and Sharpe Ratio

2 years S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	9.09	0.68	14.02	2.01	12.36	3.33	0.83
SI	6.71	0.27	14.31	1.58	13.38	2.42	0.90
RMT-0	8.71	0.70	13.63	1.78	**15.13**	3.32	1.05
RMT-M	8.77	0.69	13.42	1.75	14.54	3.33	1.03
Shrinkage_SI	9.26	0.69	13.54	1.87	12.80	3.29	0.89
Shrinkage_Cov	11.07	0.75	12.96	1.47	11.74	2.97	0.87
Shrinkage_Corr	9.86	0.75	13.99	2.13	12.69	3.55	0.84
Naïve	17.34	1.26	15.76	1.51	8.05	3.70	0.50
GLasso$^{\text{MAX RETURN}}$	8.16	1.18	13.76	1.55	11.83	3.09	0.93
GLasso$^{\text{MIN REALIZED RISK}}$	8.50	0.68	12.89	1.59	12.51	2.92	**1.10**
GLasso$^{\text{MAX SHARPE}}$	8.10	1.17	13.77	1.55	11.99	3.09	0.94
GLasso$^{\text{MAX LIKELIHOOD}}$	8.41	0.90	**12.40**	1.28	11.28	2.41	1.03
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.79	0.92	12.82	1.49	11.96	2.85	1.04

Table F.1 Long-short portfolio 2 years realized risks, portfolio return and Sharpe ratio

1 year S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	6.97	0.63	13.63	1.25	12.21	3.50	0.87
SI	5.94	0.41	13.16	1.32	11.05	3.39	0.81
RMT-0	7.18	0.67	12.35	1.24	**15.43**	3.37	**1.20**
RMT-M	7.24	0.68	12.23	1.23	14.89	3.38	1.17
Shrinkage_SI	7.59	0.70	12.35	1.09	12.15	3.24	0.95
Shrinkage_Cov	10.54	0.91	12.07	1.07	10.66	3.09	0.86
Shrinkage_Corr	8.33	0.81	12.78	1.20	12.63	3.45	0.95
Naïve	16.20	1.43	16.09	1.43	9.35	4.16	0.56
GLasso$^{\text{MAX RETURN}}$	6.63	0.82	12.03	0.98	11.17	2.50	1.01
GLasso$^{\text{MIN REALIZED RISK}}$	6.67	0.62	12.03	1.17	10.64	3.31	1.04
GLasso$^{\text{MAX SHARPE}}$	6.81	0.88	11.96	0.99	11.63	2.72	1.08
GLasso$^{\text{MAX LIKELIHOOD}}$	7.06	0.87	**11.79**	1.11	11.14	3.09	1.08
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.31	0.92	11.92	1.11	11.52	3.01	1.11

Table F.2 Long-short portfolio 1 year realized risks, portfolio return and Sharpe ratio

Appendix F

9 months S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	6.26	0.52	14.72	1.52	**12.45**	4.35	0.80
SI	5.89	0.41	12.57	1.24	8.86	3.26	0.67
RMT-0	6.90	0.56	11.78	1.11	10.82	3.19	0.88
RMT-M	6.96	0.57	11.68	1.11	10.46	3.10	0.85
Shrinkage_SI	7.28	0.60	11.95	1.04	10.60	3.39	0.85
Shrinkage_Cov	10.80	0.84	11.98	1.04	9.53	3.29	0.76
Shrinkage_Corr	8.17	0.71	12.22	1.10	11.14	3.63	0.87
Naïve	16.28	1.33	15.92	1.41	9.28	3.90	0.56
GLasso$^{\text{MAX RETURN}}$	6.37	0.82	12.05	0.96	10.52	2.54	1.03
GLasso$^{\text{MIN REALIZED RISK}}$	6.90	0.61	11.60	1.08	9.81	2.86	1.04
GLasso$^{\text{MAX SHARPE}}$	6.55	0.76	11.83	1.00	11.08	2.66	**1.14**
GLasso$^{\text{MAX LIKELIHOOD}}$	7.02	0.74	**11.49**	1.06	9.54	2.73	1.03
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.10	0.75	11.54	1.06	9.82	2.93	1.04

Table F.3 Long-short portfolio 9 months realized risks, portfolio return and Sharpe ratio

6 months S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	4.23	0.30	18.15	1.53	**14.93**	5.25	0.77
SI	5.52	0.33	12.05	0.92	9.45	3.50	0.75
RMT-0	6.10	0.42	11.91	0.96	8.25	3.15	0.66
RMT-M	6.17	0.43	11.80	0.95	8.19	3.13	0.66
Shrinkage_SI	6.41	0.43	11.72	0.82	10.34	3.22	0.85
Shrinkage_Cov	10.77	0.76	11.73	0.80	10.73	2.99	0.88
Shrinkage_Corr	7.51	0.53	12.05	0.88	10.30	3.46	0.82
Naïve	15.71	1.16	15.94	1.17	9.47	3.25	0.57
GLasso$^{\text{MAX RETURN}}$	5.62	0.51	12.66	0.97	8.37	2.69	0.94
GLasso$^{\text{MIN REALIZED RISK}}$	6.11	0.48	11.41	0.84	8.47	2.78	0.98
GLasso$^{\text{MAX SHARPE}}$	5.65	0.52	12.53	0.99	7.92	2.55	0.88
GLasso$^{\text{MAX LIKELIHOOD}}$	6.29	0.53	**11.45**	0.83	8.70	2.64	1.00
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.43	0.53	11.52	0.85	8.75	2.74	**1.01**

Table F.4 Long-short portfolio 6 months realized risks, portfolio return and Sharpe ratio

Appendix F

3 months S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Opt	1410.62	76.06	1421.58	76.44	913.54	439.05	0.62
SI	5.12	0.24	11.47	0.63	7.16	3.39	0.60
RMT-0	5.27	0.25	11.34	0.63	6.24	3.41	0.53
RMT-M	5.33	0.26	11.27	0.62	6.23	3.39	0.53
Shrinkage_SI	5.35	0.29	11.51	0.57	8.12	3.53	0.68
Shrinkage_Cov	11.73	0.65	12.00	0.60	8.84	3.18	0.71
Shrinkage_Corr	6.97	0.38	11.77	0.65	6.28	3.41	0.51
Naïve	15.67	0.85	15.80	0.85	**10.15**	4.88	0.62
GLasso$^{\text{MAX RETURN}}$	5.23	0.27	11.88	0.68	6.56	3.48	0.95
GLasso$^{\text{MIN REALIZED RISK}}$	5.55	0.31	11.41	0.58	7.69	3.26	**1.04**
GLasso$^{\text{MAX SHARPE}}$	5.34	0.27	11.85	0.69	6.74	3.52	1.00
GLasso$^{\text{MAX LIKELIHOOD}}$	5.68	0.31	**11.24**	0.58	7.15	3.13	1.00
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.55	0.27	11.42	0.60	6.80	3.20	0.97

Table F.5 Long-short portfolio 3 months realized risks, portfolio return and Sharpe ratio

2 month S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	1395.44	66.88	1408.71	67.39	763.53	375.40	0.52
SI	4.79	0.19	11.37	0.60	7.23	3.99	0.60
RMT-0	4.80	0.19	11.13	0.52	7.45	3.81	0.64
RMT-M	4.87	0.20	**11.07**	0.52	7.40	3.79	0.64
Shrinkage_SI	67.97	44.44	54.61	30.46	58.46	36.92	0.25
Shrinkage_Cov	12.18	0.57	12.10	0.55	**8.98**	3.85	0.71
Shrinkage_Corr	6.59	0.30	11.81	0.59	7.15	3.84	0.57
Naïve	15.50	0.74	15.65	0.75	8.48	4.17	0.52
GLasso$^{\text{MAX RETURN}}$	4.75	0.23	11.87	0.57	6.91	3.96	0.93
GLasso$^{\text{MIN REALIZED RISK}}$	4.98	0.24	11.55	0.54	6.80	3.86	0.93
GLasso$^{\text{MAX SHARPE}}$	4.75	0.23	11.88	0.60	7.35	4.03	**0.95**
GLasso$^{\text{MAX LIKELIHOOD}}$	5.11	0.22	11.35	0.53	6.95	3.77	0.91
GLasso$^{\text{MAX LIKELIHOOD-2}}$	4.83	0.16	11.54	0.55	6.83	3.89	0.89

Table F.6 Long-short portfolio 2 months realized risks, portfolio return and Sharpe ratio

Appendix F

1 month S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	1390.55	51.25	1394.29	51.22	664.19	375.00	0.47
SI	4.15	0.12	11.00	0.42	5.05	3.73	0.44
RMT-0	3.84	0.11	10.94	0.39	5.74	3.42	0.51
RMT-M	3.90	0.12	**10.91**	0.39	5.71	3.41	0.51
Shrinkage_SI	131.13	45.38	109.37	35.70	67.45	90.98	0.18
Shrinkage_Cov	13.10	0.47	12.44	0.42	6.62	3.58	0.52
Shrinkage_Corr	5.87	0.20	11.56	0.45	5.80	3.89	0.48
Naïve	15.45	0.57	15.49	0.57	**7.38**	4.17	0.47
GLasso$^{\text{MAX RETURN}}$	3.62	0.13	11.90	0.39	5.73	3.64	**0.96**
GLasso$^{\text{MIN REALIZED RISK}}$	4.98	0.24	11.55	0.54	6.80	3.86	0.93
GLasso$^{\text{MAX SHARPE}}$	3.68	0.12	11.97	0.39	5.71	3.56	0.94
GLasso$^{\text{MAX LIKELIHOOD}}$	3.56	0.07	11.69	0.39	5.96	3.56	0.94
GLasso$^{\text{MAX LIKELIHOOD-2}}$	3.56	0.07	11.69	0.39	5.96	3.56	0.94

Table F.7 Long-short portfolio 1 month realized risks, portfolio return and Sharpe ratio

Appendix F

Long-only Portfolio Results

Realized Risk, Portfolio Return and Sharpe Ratio

2 years N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	11.03	0.79	**8.11**	2.25	13.12	4.23	0.85
SI	9.00	0.51	13.92	1.90	13.12	3.80	0.89
RMT-0	10.67	0.78	14.18	2.14	**13.61**	4.29	0.89
RMT-M	10.54	0.76	14.07	2.09	13.56	4.22	0.90
Shrinkage_SI	10.90	0.77	14.02	2.18	13.35	4.11	0.88
Shrinkage_Cov	12.30	0.83	13.23	1.63	12.22	3.46	0.88
Shrinkage_Corr	11.43	0.83	14.27	2.37	13.25	4.35	0.85
Naïve	17.34	1.26	15.76	1.51	8.05	3.70	0.50
GLasso$^{\text{MAX RETURN}}$	7.72	1.17	13.07	1.16	10.04	2.56	0.86
GLasso$^{\text{MIN REALIZED RISK}}$	8.64	0.46	13.19	1.64	12.43	3.50	**1.04**
GLasso$^{\text{MAX SHARPE}}$	8.11	1.18	14.22	1.80	12.10	3.92	0.89
GLasso$^{\text{MAX LIKELIHOOD}}$	9.63	1.03	12.68	1.35	11.47	2.92	1.02
GLasso$^{\text{MAX LIKELIHOOD-2}}$	10.09	1.03	13.35	1.68	12.47	3.61	1.03

Table F.8 Long-only portfolio 2 years realized risks, portfolio return and Sharpe ratio

1 year N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	9.46	0.88	12.74	1.18	13.33	3.89	1.01
SI	7.90	0.64	12.94	1.19	11.88	3.80	0.89
RMT-0	9.18	0.84	12.82	1.24	**13.54**	4.01	1.02
RMT-M	9.08	0.83	12.76	1.23	13.31	3.97	1.01
Shrinkage_SI	9.35	0.85	12.63	1.15	12.99	3.83	1.00
Shrinkage_Cov	11.69	1.01	12.23	1.08	10.70	3.34	0.85
Shrinkage_Corr	10.05	0.98	12.83	1.23	13.50	4.08	1.02
Naïve	16.20	1.43	16.09	1.43	9.35	4.16	0.56
GLasso$^{\text{MAX RETURN}}$	8.02	1.16	12.72	1.05	12.60	3.51	1.08
GLasso$^{\text{MIN REALIZED RISK}}$	7.96	0.93	12.31	1.11	11.19	3.69	1.03
GLasso$^{\text{MAX SHARPE}}$	8.02	1.16	12.80	1.05	12.76	3.54	**1.09**
GLasso$^{\text{MAX LIKELIHOOD}}$	8.30	1.05	**12.17**	1.09	11.05	3.55	1.04
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.63	1.10	12.35	1.12	11.69	3.64	1.07

Table F.9 Long-only portfolio 1 year realized risks, portfolio return and Sharpe ratio

Appendix F

9 months N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	9.27	0.82	12.08	1.04	10.84	3.53	0.86
SI	7.80	0.63	12.32	1.13	10.57	3.27	0.82
RMT-0	8.99	0.77	12.12	1.08	10.72	3.14	0.85
RMT-M	8.90	0.76	12.07	1.07	10.54	3.12	0.84
Shrinkage_SI	9.14	0.78	11.99	1.03	10.73	3.38	0.86
Shrinkage_Cov	11.88	0.94	12.06	1.04	9.05	3.26	0.72
Shrinkage_Corr	10.01	0.90	12.05	1.07	**11.34**	3.47	0.90
Naïve	16.28	1.33	15.92	1.41	9.28	3.90	0.56
GLasso$^{\text{MAX RETURN}}$	7.47	1.06	12.41	0.96	11.31	3.27	1.08
GLasso$^{\text{MIN REALIZED RISK}}$	7.92	0.71	11.92	1.06	10.24	3.17	1.01
GLasso$^{\text{MAX SHARPE}}$	7.49	0.95	12.29	0.99	11.45	3.10	**1.10**
GLasso$^{\text{MAX LIKELIHOOD}}$	8.35	0.94	**11.86**	1.04	10.09	3.06	1.02
GLasso$^{\text{MAX LIKELIHOOD-2}}$	8.50	0.96	11.91	1.05	10.21	3.19	1.03

Table F.10 Long-only portfolio 9 months realized risks, portfolio return and Sharpe ratio

6 months N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	8.57	0.63	11.85	0.87	10.64	2.94	0.86
SI	7.40	0.52	11.98	0.86	**11.10**	3.00	0.89
RMT-0	8.27	0.62	11.83	0.86	9.11	2.92	0.74
RMT-M	8.20	0.61	11.81	0.86	9.04	2.88	0.73
Shrinkage_SI	8.48	0.61	**11.69**	0.87	10.37	2.73	0.85
Shrinkage_Cov	11.79	0.84	11.84	0.85	9.91	2.81	0.80
Shrinkage_Corr	9.48	0.71	11.86	0.93	9.47	2.69	0.76
Naïve	15.71	1.16	15.94	1.17	9.47	3.25	0.57
GLasso$^{\text{MAX RETURN}}$	6.57	0.80	12.76	0.94	8.27	2.69	0.87
GLasso$^{\text{MIN REALIZED RISK}}$	7.38	0.61	11.75	0.88	9.15	2.82	**1.03**
GLasso$^{\text{MAX SHARPE}}$	6.62	0.80	12.63	0.94	8.40	2.71	0.89
GLasso$^{\text{MAX LIKELIHOOD}}$	7.63	0.72	11.71	0.87	9.06	2.66	1.01
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.97	0.75	11.72	0.87	9.18	2.68	1.02

Table F.11 Long-only portfolio 6 months realized risks, portfolio return and Sharpe ratio

Appendix F

3 months N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	7.47	0.41	12.11	0.65	9.88	3.57	0.78
SI	6.89	0.38	11.60	0.61	9.15	3.54	0.76
RMT-0	7.35	0.41	11.64	0.61	8.18	3.63	0.67
RMT-M	7.31	0.41	11.61	0.61	8.12	3.61	0.67
Shrinkage_SI	7.60	0.41	11.60	0.61	9.15	3.40	0.76
Shrinkage_Cov	12.43	0.68	12.16	0.63	9.82	3.51	0.78
Shrinkage_Corr	9.11	0.50	11.65	0.67	7.45	3.18	0.61
Naïve	15.67	0.85	15.80	0.85	**10.15**	4.88	0.62
GLasso$^{\text{MAX RETURN}}$	6.51	0.49	12.25	0.67	8.23	3.41	**1.06**
GLasso$^{\text{MIN REALIZED RISK}}$	7.30	0.44	11.69	0.63	7.47	3.47	1.01
GLasso$^{\text{MAX SHARPE}}$	6.71	0.48	12.07	0.68	7.64	3.43	1.02
GLasso$^{\text{MAX LIKELIHOOD}}$	7.19	0.44	**11.46**	0.61	7.62	3.39	1.01
GLasso$^{\text{MAX LIKELIHOOD-2}}$	7.41	0.46	11.66	0.62	8.11	3.31	1.02

Table F.12 Long-only portfolio 3 months realized risks, portfolio return and Sharpe ratio

2 month N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	6.61	0.33	12.22	0.60	9.43	3.78	0.73
SI	6.51	0.30	11.63	0.59	9.28	3.89	0.75
RMT-0	6.82	0.32	11.54	0.57	9.65	3.68	0.79
RMT-M	6.80	0.32	**11.51**	0.57	9.58	3.67	0.79
Shrinkage_SI	7.02	0.32	11.57	0.56	**10.31**	3.65	0.85
Shrinkage_Cov	12.73	0.60	12.40	0.58	9.74	3.73	0.75
Shrinkage_Corr	8.84	0.41	11.59	0.60	8.57	3.72	0.70
Naïve	15.50	0.74	15.65	0.75	8.48	4.17	0.52
GLasso$^{\text{MAX RETURN}}$	6.39	0.38	11.92	0.56	7.76	3.75	1.05
GLasso$^{\text{MIN REALIZED RISK}}$	6.50	0.35	11.87	0.56	7.78	3.50	1.00
GLasso$^{\text{MAX SHARPE}}$	6.30	0.38	11.98	0.56	7.40	3.75	1.00
GLasso$^{\text{MAX LIKELIHOOD}}$	6.89	0.34	11.56	0.56	8.36	3.65	1.10
GLasso$^{\text{MAX LIKELIHOOD-2}}$	6.92	0.35	11.65	0.55	8.52	3.56	**1.11**

Table F.13 Long-only portfolio 2 months realized risks, portfolio return and Sharpe ratio

Appendix F

1 month N.S.S	Predicted Risk(%)	std dev	Realized Risk(%)	std dev	Portfolio Return (%)	std dev	Sharpe Ratio
Direct Optimization	4.38	0.24	13.11	0.52	**7.64**	3.74	0.56
SI	5.60	0.20	11.60	0.44	5.98	3.45	0.50
RMT-0	5.48	0.21	11.57	0.42	7.20	3.33	0.61
RMT-M	5.49	0.21	**11.54**	0.42	7.13	3.32	0.60
Shrinkage_SI	5.72	0.21	11.76	0.43	6.57	3.38	0.54
Shrinkage_Cov	13.39	0.48	12.74	0.44	5.67	3.60	0.44
Shrinkage_Corr	8.20	0.29	11.56	0.47	6.44	3.55	0.53
Naïve	15.45	0.57	15.49	0.57	7.38	4.17	0.47
GLasso$^{\text{MAX RETURN}}$	5.56	0.22	12.12	0.44	6.20	3.24	**0.92**
GLasso$^{\text{MIN REALIZED RISK}}$	5.58	0.22	12.13	0.43	6.02	3.43	0.92
GLasso$^{\text{MAX SHARPE}}$	5.60	0.22	12.11	0.44	6.48	3.23	0.94
GLasso$^{\text{MAX LIKELIHOOD}}$	5.65	0.21	12.12	0.45	5.98	3.31	0.94
GLasso$^{\text{MAX LIKELIHOOD-2}}$	5.65	0.21	12.12	0.45	5.98	3.31	0.94

Table F.14 Long-only portfolio 1 month realized risks, portfolio return and Sharpe ratio

Appendix G

Portfolio Performance Measure Annualization Factor

Performance Measure	Annualization Factor
Realized risk	$\times \sqrt{250}$
Predicted risk	$\times \sqrt{250}$
Empirical risk	$\times \sqrt{250}$
Sharpe ratio	$\times \sqrt{250}$
Portfolio return	$\times \ frequency$

Table G.1 Portfolio performance measure annualization factor

where $frequency$ is the number of rebalancing times per year (e.g. for the 2 months rebalancing period, $frequency = \frac{12}{2} = 6$).

*Note that there are 250 trading days per year.

Bibliography

1. O. Ledoit and M. Wolf, “Nonlinear shrinkage of the covariance matrix for portfolio selection: Markowitz meets Goldilocks,’’ SSRN 2383361 SPIE press, 2014.

2. H. Markowitz, ‘‘Portfolio selection,’’ Journal of Finance, vol. 7, issue 1, pp. 77-91, 1952.

3. R. C. Merton, ‘‘On estimating the expected return of the market: An explanatory investigation,’’ Journal of Financial Economics, vol. 8, pp. 323-361, 1980.

4. V. DeMiguel, L. Garlappi, F. J. Nogales and R. Uppal, ‘‘A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms,” Management Science, vol. 55, no.5, pp 798-812, 2009.

5. J. Green, J. R. M. Hand and X. F. Zhang, ‘‘The supraview of return predictive signals,” Review of Accounting Studies, vol. 18, no. 3, pp 692-730, 2013.

6. Ledoit and M. Wolf, ‘‘Improved estimation of the covariance matrix of stock returns with an application to portfolio selection,” Journal of Empirical Finance, vol. 10, no. 5, pp. 603-621, 2003.

7. O. Ledoit and M. Wolf, ‘‘Honey, I shrunk the sample covariance matrix,’’ Journal of Portfolio Management, vol. 30, no. 4, pp 110-119, 2004a.

8. Ledoit and M. Wolf, ‘‘Honey, I shrunk the sample covariance matrix,’’ Journal of Portfolio Management, vol. 30, no. 4, pp 110-119, 2004b.

9. Ledoit and M. Wolf, “A well-conditioned estimator for large-dimensional covariance matrices,” Journal of Multivariate Analysis, vol. 88, pp. 365-411, 2004.

10. M. W. Brandt, P. Santa-Clara and R. Valkanov, "Parametric portfolio policies: Exploiting characteristics in the cross-section of equity returns,'' Review of Financial Studies, vol. 22, pp. 3411-3447, 2009.

11. V. DeMiguel, L. Garlappi, F. J. Nogales and R. Uppal, ''A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms," Management Science, vol. 55, no.5, pp 798-812, 2009.

12. V. DeMiguel, A. Martin-Utrera and F. J. Nogales, "Size matters: Optimal calibration of shrinkage estimators for portfolio selection,'' Journal of Banking & Finance, vol. 37, pp. 3018-3034, 2013.

13. G. Frahm and C. Memmel, "Dominating estimators for minimum-variance portfolios,'' vol. 159, pp. 289-302, 2010.

14. J. Tu and G. Zhou, "A combination of sophisticated and naive diversification strategies,'' Journal of Financial Economics, vol. 99, pp. 204-215, 2011.

15. A Kourtis, G. Dotsis, and R. N. Markellos, "Parameter uncertainty in portfolio selection: Shrinking the inverse covariance matrix,'' Journal of Banking and Finance, vol. 36, pp. 2522-2531, 2012.

16. J. Friedman, T. Hastie and R. Tibshirani, "Sparse inverse covariance estimation with the graphical Lasso," Biostatistics, vol. 9, pp. 432–441, 2007.

17. C. Bishop, "Pattern recognition and machine learning," Information Science and Statistics, Springer, 1st edition, 2007.

18. T. Hastie, R. Tibshirani and J. Friedman, "The Elements of Statistical Learning," Springer. New York, 2001.

19. O. Banerjee, L. E. Ghaoui and A. D'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," Journal of Machine Learning Research, pp. 485 – 516, 2008.

20. P. A. Dempster, "Covariance selection," Biometrics, pp. 157-175, 1972.

21. N. Meinhausen and P. Buhlmann, "High-dimensional graphs and variable selection with the Lasso," The Annals of Statistics, vol. 34, no.3, pp. 1436-1462, 2006.

22. M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," Biometrika, vol. 94, no. 1, pp. 19 - 21, 2007.

23. K. Sachs, O. Perez, D. Pe'er, D. A Lauffenburger, and G.P Nolan, "Causal protein-signaling networks derived from multiparameter single-cell data," Science 308, no. 5721, pp. 523-529, 2005.

24. L. Wasserman, "All of Statistics,'' Springer, 2004.

25. K.V Mardia, J.T Kent, and J.M Bibby. "Multivariate analysis," Academic press, 1979.

26. J. Whitaker, "Graphical Models in Applied Multivariate Statistics," Chichester: John Wiley and Sons, 1990.

27. N. Meinhaussen and P. Bulhlmann, "High-dimensional graphs and variable selection with the Lasso," Ann. Statist, vol. 34, pp. 1436 – 1462, 2006.

28. L. Breiman, "Heuristics of instability in model selection," Berkeley: Statistics Department, University of California, 1994.

29. R. Tibshirani, "Regression shrinkage and selection via the Lasso," Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 58, no.1, pp. 267-288, 1996.

30. O. Banerjee, L. E. Ghaoui and A. D'Aspremont, "Convex optimization techniques for fitting sparse Gaussian graphical models," Proceedings of the 23rd international conference on machine learning, pp. 89-96, NY, USA, 2006.

31. Y. Nesterov and A. Nemisrovski, "Interior-point polynomial algorithms in convex programming," vol. 13, Philadelphia: Society for industrial and applied mathematics, 1994.

32. B. I. Rubinstein, J. McAuliffe, S. Cawley, M. Palaniswami, K. Ramamohanarao and P. T. Speed, "Machine learning in low-level microarray analysis," ACM SIGKDD Explorations Newsletter, vol. *5*, no. 2, pp. 130-139, 2003.

33. C. A. Tsai, C .H. Chen, T. C. Lee, I. C. Ho, U. C. Yang and J. J. Chen, "Gene selection for sample classifications in microarray experiments," DNA and cell biology vol. 23, no. 10, pp. 607-614, 2004.

34. J. Newton, "Analysis of microarray gene expression data using machine learning techniques," University of Alberta, Edmonton, Canada, 2002.

35. D. A. Lashkari, J. L. Deris, J. H. McCusker, A. F. Namath, C. Gentile, S. Y. Hwang, P. O. Brown and R. W. Davis, "Yeast microarrays for genome wide parallel genetic and gene expression analysis," Proceedings of the National Academy of Sciences, vol. 94, no. 24, pp. 13057- 13062, 1997.

36. M. B. Eisen, P. T. Spellman and P. Brown," Cluster analysis and display of genomewide expression patterns," Proceedings of the National Academy of Science, vol. 95, pp. 14863-14868, 1997.

37. J. L. DeRisi, V. R. Iyer and P. O. Brown, "Exploring the metabolic and genetic control of gene expression on a genomic scale," Science278, no. 5338, pp. 680-686, 1997.

38. P. Hu, S. B. Bull and H. Jiang, "Gene network modular-based classification of microarray samples," BMC bioinformatics 13, Suppl 10: S17, 2012.

39 L. Elo, H. Jarvenpaa, M. Oresic, R. Lahesmaa and T. Aittokallio, "Systematic construction of gene coexpression networks with applications to human T helper cell differentiation process," Bioinformatics, vol. 23, pp. 2096-2103, 2007.

40. A. Presson. R. Sobel, J. Papp, C. Suarez, T. Whistler, M. Rajeevan, et al," Integrated weighted gene co-expression network analysis with an application to chronic fatigue syndrome," BMC Syst Biol 2:95, 2008.

41. S. Horvath and J. Dong, "Geometric interpretation of gene coexpression network analysis," PLoS Comput Biol 4:e1000117, 2008

42. I. W. Taylor, R. Linding, D. Warde-Farley, Y. Liu, C. Pesquita, D. Faria et al, " Dynamic modularity in protein interaction networks predicts breast cancer outcome, " Nat Biotechnol, vol. 27, pp. 199-204, 2009.

43. J. Schafer and K. Strimmer, "Learning large-scale graphical Gaussian models from genomic data," Science of Complex Networks from Biology to the Internet and WWW 776, pp. 263-276, 2005.

44. S. L. Lauritzen, "Graphical models," Oxford University Press, 1996.

45. D. Husmeier, "Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks." Bioinformatics 19, no. 17, pp. 2271-2282, 2003.

46. A. Dobra, C. Hans, B. Jones, J. R. Nevins and M. West, "Sparse graphical models for exploring gene expression data," Journal of Multivariate Analysis, vol. 90, no. 1, pp. 196-212, 2004.

47. B. Chain, H. Bowen, J. Hammond, W. Posch, J. Rasaiyaah, J. Tsang and M. Noursadeghi, "Error, reproducibility and sensitivity: a pipeline for data processing of Agilent oligonucleotide expression arrays, " BMC Bioinformatics, Vol.11, no. 344, ISSN 1471-2105, 2010.

48. E. Qian, "Quantitative Equity Portfolio Management: Modern Techniques and Applications,'' vol. 6, Chapman & Hall/CRC Press, 2007.

49. E. Marnix. "Portfolio Optimization: Beyond Markowitz." Master's thesis, Leiden University, 2004.

50. E. Pantaleo, M. Tumminello, F. Lillo and R. Mantegna, "When do improved covariance matrix estimators enhance portfolio optimization? an empirical comparative study of nine estimators," Quantitative Finance, vol. 11, pp. 1067-1080, 2011.

51. W. James and C. Stein, "Estimation with quadratic loss," Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, California, pp. 361-379, University of California Press, 1961.

52. A. Stuart and S. Liang, "Modeling the normal and neoplastic cell with 'realistic Boolean genetic networks': Their application for understanding carcinogenesis and assessing therapeutic strategies," Pacific Symposium on Biocomputing, Singapore, vol. 3, pp. 66-67, World Scientific Publishing, 1998.

53. R. E. Bellman, “Adaptive Control Processes: A Guided Tour,’’ Princeton University Press, 1961.

54. V. Chopra and W. Ziemba, “The effects of errors in means, variances and covariances on optimal portfolio choice,” Journal of Portfolio Management, vol. 19, pp. 6-12, 1993.

55. J. E. Ingersoll, ‘‘Theory of financial decision making, Rowman & Littlefield Pub Inc, 1987.

56. P. Jorion, “International Portfolio Diversification with Estimation Risk,” Journal of Business, vol. 58, pp. 259-278, 1985.

57. R. Jagannathan and T. Ma, Journal of Finance, American Finance Association, vol. 58, pp. 1641-1684, 2003.

58. M. J. Best and R. R. Grauer, “On the Sensitivity of mean-variance-efficient portfolios to changes in asset means: some analytical and computational results,” Review of Financial Studies, vol. 4, pp. 315-342, 1991.

59. O. Ledoit and M. Wolf, “Improved estimation of the covariance matrix of stock returns with an application to portfolio selection,” Journal of Empirical Finance, vol. 10, pp. 603-621 2003.

60. K. V. Mardia, J. T. Kent, and J. M. Bibby, “Multivariate analysis, ” Academic Press, San Diego, CA, 1979.

61. L. Laloux, P. Cizeau, J. P Bouchaud, & M. Potters, “Noise dressing of financial correlation matrices,” Physical review letters, vol. 83, no.7, pp. 1467, 1999.

62. O. Ledoit and M. Wolf, Journal of Portfolio Management vol. 30, pp. 110–119, 2004.

63. W. F. Sharpe, "A simplified model for portfolio analysis," Management science, vol. 9.2 pp. 277-293, 1963.

64. R. C. Green and B. Hollifield, Journal of Finance, American Finance Association, vol. 47, pp. 1785– 1809, 1992.

65. S. Pafka and I. Kondor, "Noisy covariance matrices and portfolio optimization II," Physica, A319, pp. 487-494, 2003.

66. I. Kondor, S. Pafka and G.Nagy, "Noisy sensitivity of portfolio selection under various risk measures," Journal of Banking and Finance, vol. 31, pp. 1545-1573, 2007.

67. M. Tumminello, C. Coronnello, F. Lillo, S. Miccich`e, and R.N. Mantegna, "Spanning trees and bootstrap reliability estimation in correlation based networks," Int. J. Bifurcation Chaos, vol. 17, no.7, pp. 2319-2329, 2007.

68. B, Rosenow, V, Plerou, P, Gopikrishnan, L. A. N, Amaral, T, Guhr, and H.E, Stanley, "Random matrix approach to cross correlations in financial data," Physical Review E, vol. 65, no. 6 , 2002.

69. M. Potters, J.-P. Bouchaud, and L. Laloux, "Financial applications of random matrix theory: old laces and new pieces," Acta Phys. Pol. B 36, no. 9, pp. 2767-2784, 2005.

70. C. Stein, "Inadmissibility of the usual estimator for the mean of a multivariate normal distribution," University of California Press, Berkeley, 3rd edition, pp. 197-206, 1956.

71. J. Schafer and K. Strimmer, "A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics," Statistical Applications in Genetics and Molecular Biology, vol. 4, no. 1, 2005.

72. C. Bishop, “Pattern recognition and machine learning,” Information Science and Statistics, Springer, 1st edition, 2007.

73. K.P Murphy, "Machine learning: a probabilistic perspective (adaptive computation and machine learning series)," MIT Press, pp.15, 2012.

74. E. Alpaydin, “Introduction to machine learning,” MIT press, 2014.

75. N.J Nilsson, "Introduction to machine learning. An early draft of a proposed textbook," (1996).

76. I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

77. I. Guyon, S. Gunn, M. Nikravesh and L. Zadeh, “Feature extraction: foundations and applications,” Springer, 2006.

78. A. Blum and P. Langley, “Selection of relevant features and examples in machine learning: Artificial Intelligence,” vol. 97, no. 1-2, pp. 245-271, 1997.

79. R. Kohavi and G. H. John, “Wrappers for feature subset selection. Artificial Intelligence,” vol. 97, no. 1-2, pp. 273-324, 1997.

80. Y. Saeys, I. Inza and P. Larranaga, “P. A review of feature selection techniques in bioinformatics. Bioinformatics,” vol.23, no. 19, pp. 2507 -2517, 2007.

81. G. H. John, R. Kohavi and K. Pfleger, “Irrelevant features and the subset selection problem,” machine learning: proceeding of the 11th International Conference, pp. 121-129, 1994.

82. H. Yu, N. M. Luscombei, J. Qian and M. Gerstein, "Genomic analysis of gene expression relationships in transcriptional regulatory networks," Trends in Genetics, vol. 19, no. 8, pp. 422-427, 2003.

83. K. Kontos, "Gaussian graphical model selection for gene regulatory network reverse engineering and function prediction," PhD thesis, Free University of Brussels, Belgium, 2003.

84. A, E. Hoerl and R. Kennard, "Ridge regression: biased estimation of nonorthogonal problems," Technometrics, vol. 12, pp. 55-67,1970.

85. R. E. Kalman, "Algebraic aspects of the generalized inverse of a rectangular matrix," Academic Press, New York, 1976.

86. R. MacAusland, "The Moore-Penrose Inverse and Least Squares," Academic Press, 2014.

87. G. W. Stewart and J. G. Sun, "Matrix Perturbation Theory," Academic Press, London, 1990.