MARKOWITZ PORTFOLIO OPTIMIZATION WITH MISSPECIFIED COVARIANCE MATRICES

A Thesis in
Industrial Engineering
by
Yuan Wang

© 2015 Yuan Wang

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2015
The thesis of Yuan Wang was reviewed and approved* by the following:

Necdet Serhat Aybat  
Assistant Professor of Industrial Engineering  
Thesis Advisor

Uday V. Shanbhag  
Associate Professor of Industrial Engineering

Lingzhou Xue  
Assistant Professor of Statistics

Harriet Black Nembhard  
Professor of Industrial Engineering  
Department Head

*Signatures are on file in the Graduate School.
Abstract

We consider portfolio optimization problems in which the true covariance matrix is misspecified and its value may be obtained by solving a suitably defined learning problem. We consider two types of learning problems to aid in such a resolution: (i) sparse covariance selection; and (ii) sparse precision matrix selection. A traditional sequential approach for addressing such a problem requires first solving the learning problem and then using the solution of this problem in solving the resulting computational problem. Unfortunately, exact solutions to the learning problem may only be obtained asymptotically; consequently, practical implementations of the sequential approach may provide approximate solutions, at best. Instead, we consider a simultaneous approach that solves both the learning problem and portfolio optimization problems simultaneously. In particular, we use the alternating direction method of multipliers (ADMM) to solve the learning problem while the projected gradient method is used to solve the computational problem. Asymptotic convergence statements and rate analysis is conducted for the simultaneous scheme. Preliminary numerics on a class of misspecified portfolio optimization problems suggests that the scheme provides accurate solutions with a comparable performance with the sequential approach.
Contents

List of Tables vi

Acknowledgments vii

Chapter 1
   Introduction 1
      1.1 Markowitz Portfolio Optimization Problem . . . . . . . . . . . . . 2
      1.2 Learning Problems . . . . . . . . . . . . . . . . . . . . . . . . 5
         1.2.1 Sparse Covariance Selection (SCS) Problem . . . . . . . 5
         1.2.2 Sparse Precision Selection (SPS) Problem . . . . . . . 6
         1.2.3 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
            1.2.3.1 Computational Problem . . . . . . . . . . . . . 7
            1.2.3.2 Learning Problems . . . . . . . . . . . . . . . . 8
      1.3 Alternating Direction Method of Multipliers . . . . . . . . . . . . 9
      1.4 Simultaneous Scheme . . . . . . . . . . . . . . . . . . . . . . . 11

Chapter 2
   Description of Algorithms 12
      2.1 Projected Gradient Method . . . . . . . . . . . . . . . . . . . . 12
      2.2 ADMM for SCS Learning Problem: First Approach . . . . . . . . . 13
      2.3 ADMM for SCS Learning Problem: Second Approach . . . . . . . . 17
      2.4 ADMM for SPS Learning Problem . . . . . . . . . . . . . . . . . 19
      2.5 Sequential Scheme . . . . . . . . . . . . . . . . . . . . . . . . . 22
         2.5.1 Markowitz Optimization with SCS Learning Problem: First
            Approach . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
         2.5.2 Markowitz Optimization with SCS Learning Problem: Sec-
            ond Approach . . . . . . . . . . . . . . . . . . . . . . . . . 23
Chapter 3
Theoretical Results for the Simultaneous Scheme
3.1 Lipschitz Continuity of Matrix Inverse
3.2 Convergence of Simultaneous Scheme
3.2.1 Analysis of Computational Problem
3.2.2 Case 1: SCS Learning Problem
3.2.3 Case 2: SPS Learning Problem

Chapter 4
Numerical examples
4.1 Data Generation and Problem Parameters
4.2 Covariance matrix of size 500 × 500
4.2.1 Markowitz Portfolio Optimization with SCS Learning
4.2.2 Markowitz Portfolio Optimization with SPS Learning
4.3 Covariance matrix of size 1000 × 1000
4.3.1 Markowitz Portfolio Optimization with SCS Learning
4.3.2 Markowitz Portfolio Optimization with SPS Learning

Bibliography
List of Tables

4.1 SCS, 1st approach, 500 \times 500 covariance  \hspace{0.5cm} 36
4.2 SCS, 2nd approach, 500 \times 500 covariance  \hspace{0.5cm} 36
4.3 SPS, 500 \times 500 covariance  \hspace{0.5cm} 37
4.4 SCS, 1st approach, 1000 \times 1000 covariance  \hspace{0.5cm} 38
4.5 SCS, 2nd approach, 1000 \times 1000 covariance  \hspace{0.5cm} 38
4.6 SPS, 1000 \times 1000 covariance  \hspace{0.5cm} 38
Acknowledgments

I wish to express my sincere thanks to my advisors Necdet Serhat Aybat and Uday V. Shanbhag, for providing me with all the necessary help for the research and for continuously encouraging me along the way. Their comments cheered me up, or made me bad, but always spurred me on. And their lectures are invaluable sources of the idea of the thesis.

Besides my advisors, I would like to thank Lingzhou Xue and Harriet Black Nembhard, for their close reading of this thesis.

I take this opportunity to express my gratitude to my parents, for their unceasing encouragement and support.

I also want to thank my girlfriend, Xijia Zhang, for her support, patience and generousness. When I felt down, she is always with me.

I am also very grateful to my friends, especially Yue Xie, Xue Wang and Sen Lin. They point out my fault in the code, and encourage me to complete the thesis.
Chapter 1

Introduction

Markowitz portfolio optimization problems lay the foundation of modern portfolio theory. Although the benefits of diversification in risk reduction have been appreciated since the inception of financial markets, Markowitz was the first to construct a mathematical model that explicitly modeled the risk-return trade-off [1]. In Markowitz portfolio selection model, the return of a portfolio is defined to be the expected value of the random portfolio return while the risk is defined to be the variance of the random portfolio returns. The model tries to minimize the risk under a given level of return, or equivalently to maximize the return under a given level of risk. Markowitz showed that when a lower bound of return is given, the optimal portfolio can be found by solving a convex quadratic problem [1].

Although Markowitz portfolio model may succeed in theory and has led to the Capital Asset Pricing Model (CAPM) for asset pricing [2], its practical impacts have been less pronounced [3]. Michaud had summarized the problem in [4], “Although Markowitz efficiency is a convenient and useful theoretical framework for portfolio optimality, in practice it is an error prone procedure that often results in error-maximized and investment-irrelevant portfolios.” This suggests that the optimal portfolios are quite sensitive to small perturbations in problem parameters, e.g., the mean and the variance of asset returns. Several methods have been proposed to reduce the sensitivity of the Markowitz-optimal portfolio to parameter uncertainty: Frost and Savarino proposed constraining the portfolio weights [5], Chopra et al. propose using a James-Stein estimator [6], Klein and Bawa, Frost and Savarino, and Black and Litterman propose Bayesian estimation of means
and covariance [7, 5, 8]. These approaches may mitigate the effects of parameter uncertainty, but they do not explicitly account for parameter uncertainty in the portfolio construction step [9].

To model the uncertainty of input, Ben-Tal and Nemrovski [10, 11] introduced robust optimization that models the uncertainty of input directly into the formulation of optimization. The basic idea is to construct an uncertainty set such that all possible input data lie within the uncertainty set and reformulate the robust optimization problem into an equivalent, but tractable form, called the robust counterpart of the original problem. Several types of uncertainty sets may be used, e.g., scenario-based, interval, and ellipsoidal uncertainty sets. We will discuss this approach in detail later in Section 1 of this chapter. A recent book [12] contains more optimization methods in finance industry.

In this chapter, we formulate the problem and consider the misspecification of problem parameter. To deal with the misspecification, we present two learning problems to estimate the unknown covariance matrix. The third section includes a brief introduction of Alternating Direction Method of Multipliers (ADMM), which will be used while solving the two learning problems. Traditionally, sequential approaches were employed that first estimated the covariance matrix by solving the learning problem and then solved the portfolio selection. But this is not favorable in practice because when learning problems are being solved, one acquire no information on optimal portfolio. In contrast, we may adopt a simultaneous scheme to solve learning and computational problems simultaneously, so that in each iteration one may get an approximation of the optimal portfolio.

1.1 Markowitz Portfolio Optimization Problem

Let $A_1, \ldots, A_n$ be $n$ assets with random returns. Assume that the joint distribution of returns is a multivariate normal distribution. We let $x_i$ denote the proportion of asset $i$ in the portfolio held throughout the given period. Hence, $x \in \mathbb{R}^n$ such that $\sum_{i=1}^n x_i = 1$, $x_i \geq 0$ for all $i = 1, \ldots, n$ corresponds to a feasible portfolio with no short selling. To decide the optimal portfolio, we face two competing objectives: minimize the risk and maximize the expected return. If the expected return of asset $A_i$ is $\mu_i$ and the covariance between returns of asset $A_i$ and $A_j$ is $\sigma_{ij}$, then
we denote the $n \times n$ covariance matrix associated with the assets $A_1, \ldots, A_n$ by $\Sigma := (\sigma_{ij})_{1 \leq i, j \leq n}$ and the expected return by $\mu := (\mu_i)_{i=1}^n$. We assume that the covariance matrix $\Sigma$ is positive definite, which means that there is no redundant asset in our collection $\{A_1, \ldots, A_n\}$.

The Markowitz portfolio optimization problem takes the following form to minimize the risk of a given predetermined level of return $R > 0$:

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} x^T \Sigma x, \\
\text{subject to} & \quad 1^T x = 1, \\
& \quad \mu^T x \geq R, \\
& \quad x \geq 0,
\end{align*}$$

where $1$ denotes the $n$-dimensional vector consisting of all 1’s and $0$ denotes the $n$-dimensional vector of all 0’s. Equivalently, one may also maximize the expected portfolio return for a given level of risk $R$:

$$\begin{align*}
\text{maximize} & \quad \mu^T x \\
\text{subject to} & \quad \frac{1}{2} x^T \Sigma x \leq V, \\
& \quad 1^T x = 1, \\
& \quad x \geq 0,
\end{align*}$$

where $V$ is a an upper bound on the variance of portfolio return that can be tolerated. Both forms can be equivalently written as follows:

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} x^T \Sigma x - \kappa \mu^T x \\
\text{subject to} & \quad 1^T x = 1 \\
& \quad x \geq 0,
\end{align*}$$

where $\kappa$ is a given positive parameter that represents the trade-off between return and risk. A small value $\kappa$ corresponds to “risk averse” portfolios while a large value $\kappa$ corresponds to “risk taking” ones. In this work we focus on the last form of the problem, because it has a special constraint structure, namely the probabilistic
simplex, i.e., \( X = \{ x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 1, \ x_i \geq 0, \ i = 1, \ldots, n \} \).

If the problem parameters \( \mu \) and \( \Sigma \) are known, then Markowitz optimization problem is just a convex quadratic optimization problem. However, knowing the true \( \mu \) and \( \Sigma \) exactly cannot often be taken for granted. In fact, their estimation under some prior distributions is not easy in practice, since the historical data sets are generally quite large. Another important property of \( \Sigma \) is the sparsity, meaning that there are many zero entries in \( \Sigma \). This is because among the \( n \) assets \( A_1, \ldots, A_n \), there would be many of them are pairwise independent. In [13] it is shown that \( (\Sigma^{-1})_{ij} = 0 \) when assets \( A_i \) and \( A_j \) are conditionally independent.

There are several ways to overcome the difficulty caused by uncertainty. One of them is by robust optimization which we briefly discussed above. In 2000, Halldórsson and Tütüncü [14] showed that if the uncertain mean return vector \( \mu \) and the uncertain covariance matrix \( \Sigma \) of the random asset returns \( r \in \mathbb{R}^n \) belong to box uncertainty sets, i.e., \( \mu \in \{ \mu|\mu^L \leq \mu \leq \mu^U \} \) and \( \Sigma \in \{ \Sigma \succeq 0|\Sigma^L \preceq \Sigma \preceq \Sigma^U \} \), then the robust problem reduces to saddle-point problem that involves semidefinite constraints. A multi-period robust model, where the uncertainty set having a finite number of elements, is proposed by Ben-Tal et al. [15]. In [9], Goldfarb and Iyengar presented a robust formulation for Markowitz portfolio problem. Assume the single period return \( r \) is a random vector given by

\[
    r = \mu + V^T f + \epsilon,
\]

where \( \mu \in \mathbb{R}^n \) is the mean vector of returns, \( f \sim N(0, F) \) is the vector of returns of the factors that drive the market, \( V \in \mathbb{R}^{m \times n} \) is the matrix of factor loadings of the \( n \) assets and \( \epsilon \sim N(0, D) \) is the vector of residual returns. Further assume that \( \epsilon \) and \( f \) are independent random vectors, \( F \succ 0 \) and \( D = \text{diag}(d) \succeq 0 \), i.e., \( d_i \geq 0, \ i = 1, \ldots, n \). Then the vector of asset returns \( r \sim N(\mu, V^T F V + D) \). Suppose the matrix \( D \) lies in the uncertainty set

\[
    S_d = \{ D : D = \text{diag}(d), \ d_i \in [l_i, u_i], \ i = 1, \ldots, n \},
\]

and the matrix \( V \) lies in the ellipsoidal uncertainty set

\[
    S_v = \{ V : V = V_0 + W, \ ||W_i||_G \leq \rho_i, \ i = 1, \ldots, n \},
\]
where $W_i$ is the $i$-th column of $W$ and $\|w\|_G = \sqrt{w^T G w}$ denotes the elliptic norm of $w$ with respect to a given symmetric, positive definite matrix $G$. Method of computing $G$ is given in the appendices of [9]. Further assume that the mean return vector lies in the uncertainty set

$$S_m = \{ \mu : \mu = \mu_0 + \xi, \ |\xi_i| \leq \gamma_i, \ i = 1, \ldots, n \}.$$ 

Let $x$ denote the portfolio, which is our decision variable. The return of $x$ is $r^T x \sim N(\mu^T x, x^T (V^T F V + D)x)$. The robust formulation given in [9] is as follows:

$$\begin{array}{ll}
\text{minimize} & \max_{V \in S_v, D \in S_d} \text{Var}[r^T x] \\
\text{subject to} & \min_{\mu \in S_m} E[r^T x] \geq \alpha,
\end{array}$$

$$1^T x = 1.$$ 

The robust Markowitz portfolio optimization minimizes the worst case variance of the portfolio subject to the constraint that the worst case expected return is at least $\alpha$.

The robust optimization approach utilizes an uncertainty set and focuses on the worst-case solution. We consider a setting when there is a true nominal value for the covariance matrix and is given by the solution to a suitably defined learning problem. The two learning problems we have chosen are sparse covariance matrix selection and sparse precision matrix selection learning problems, which are explained Section 1.2.

### 1.2 Learning Problems

#### 1.2.1 Sparse Covariance Selection (SCS) Problem

Given a sample of returns for the $n$ assets and with sample size equal to $p$. In practice, we usually have $p \ll n$, which means that the number of assets exceeds far more than the sample size. Let $S = (s_{ij})_{1 \leq i,j \leq n}$ be the sample covariance. Since $p < n$, we know that $S$ cannot be positive definite and so we cannot use $S$ as our true covariance estimator, because the problem parameter $\Sigma$ is assumed to
be positive definite. To resolve this issue, we consider the following optimization problem, which is called sparse covariance selection problem:

\[
(L_{SCS}) : \min_{\Sigma} \frac{1}{2} \| \Sigma - S \|_F^2 + \lambda |\Sigma|_1 \\
\text{subject to} \quad \Sigma \succeq \epsilon I,
\]

where \( \lambda \) is a given positive parameter, \( \epsilon \) is an arbitrarily small positive number, \( \| \cdot \|_F \) is the Frobenius norm, and \( |\cdot|_1 \) is the element-wise \( \ell_1 \)-norm of all off-diagonal elements. Notice that the constraint in this problem guarantees that the estimate \( \Sigma \) is positive definite and the \( \ell_1 \) penalized term in the objective guarantees that the optimal solution is a sparse matrix. Therefore, the optimal solution \( \Sigma^* \) will satisfy our full-rank assumption on the covariance matrix.

Note that SCS learning problem estimates the true covariance matrix. Covariance estimation is a fundamental statistical problem. To quote Xue et al. from [16], “the usual sample covariance matrix is optimal in the classical setting with large samples and fixed low dimensions [17], but it performs very poorly in the high dimensional setting [18].” Rothman et al. defined the general thresholding rule to be a function of the sample covariance matrix [19], e.g., hard thresholding, soft thresholding and adaptive lasso thresholding [20], etc. Note that the SCS problem is motivated by the soft-thresholding covariance estimator [16]:

\[
\hat{\Sigma} = \arg\min_{\Sigma \in S^n} \frac{1}{2} \| \Sigma - S \|_F^2 + \lambda |\Sigma|_1,
\]

where \( S^n \) is the space of all \( n \times n \) symmetric matrices. However this formulation cannot guarantee that \( \hat{\Sigma} \) is positive definite, hence the constraint \( \Sigma \succeq \epsilon I \) is added to the formulation in \((L_{SCS})\) to guarantee the positive-definiteness of the estimator.

### 1.2.2 Sparse Precision Selection (SPS) Problem

In the same setting as in Section 1.2.1, rather than use the SCS estimator, one may also use the maximum likelihood estimator. It is known that the sample covariance is the maximum likelihood estimator of covariance matrix. But when the sample size \( p \) is less than \( n \), the sample covariance matrix is rank deficient and cannot be used as our covariance matrix, motivating the consideration of the
sparse inverse covariance selection problem, which is also known as sparse precision selection (SPS). The sparsity property of the precision matrix can be utilized to simplify the covariance structure of a multivariate normal distribution [13]. The SCS problem in [21] is to estimate the corresponding precision matrix by solving a regularized maximum likelihood problem:

\[
\min_{P>0} \langle P, S \rangle - \log \det(P) + \alpha \text{card}(P),
\]

where \text{card}(P) is the number of nonzero elements of \( P \). But this problem is generally challenging to solve because of the non-convex cardinality function. In [21], d’Aspremont et al. consider an approximation of the above problem, which we introduce now:

\[
(\mathcal{L}_{SPS}) : \minimize_P \langle S, P \rangle - \log \det(P) + \alpha \|P\|_1
\]

subject to \( P > 0 \),

where \( \alpha \) is a given positive number and \( \| \cdot \|_1 \) is the \( \ell_1 \) norm. Here, the decision variable \( P \) corresponds to the inverse of the covariance matrix \( \Sigma \).

1.2.3 Summary

1.2.3.1 Computational Problem

The main (computational) problem we are trying to solve is the classical Markowitz portfolio optimization problem:

\[
(\mathcal{C}) : \minimize_x \frac{1}{2} x^T \Sigma x - \kappa \mu^T x
\]

subject to \( 1^T x = 1 \),

\[
x \geq 0,
\]

where \( \kappa > 0 \) represents the payoff between risk and return.

As we have said before, the data for the computational problem \( \Sigma \) and \( \mu \) are often not observable and we have to use an estimator. For the expected return \( \mu \) we could use sample mean as an estimator since it is unbiased. But for the covariance matrix \( \Sigma \), we consider two learning problems.
1.2.3.2 Learning Problems

Let $S \in \mathbb{S}^n$ denote the sample covariance matrix. Then we have the following two learning problems:

- **SCS learning problem**
  \[
  \mathcal{L}_{SCS}: \min_{\Sigma \succeq 0} \|\Sigma - S\|_F^2/2 + \lambda|\Sigma|_1,
  \]
  where $\epsilon$ and $\lambda$ are positive parameters, $\|\cdot\|_F$ is the Frobenius norm, and $|\cdot|_1$ is the off-diagonal norm, i.e., $|A|_1 = \sum_{i \neq j} |A_{ij}|$.

- **SPS learning problem**
  \[
  \mathcal{L}_{SPS}: \min_{P > 0} \langle S, P \rangle - \log \det(P) + \alpha\|P\|_1,
  \]
  where $\alpha$ is a trade-off that controls sparsity, $\langle \cdot, \cdot \rangle$ denotes the inner product of two matrices and $\|\cdot\|_1$ is the $\ell_1$ norm of a matrix, i.e., $\langle A, B \rangle = \sum_{i,j=1}^n A_{ij}B_{ij}$ and $\|A\|_1 = \sum_{i,j=1}^n |A_{ij}|$.

Once the covariance matrix is learned, the computational problem is solved using projected gradient method, which is widely used for convex optimization due to its simplicity of implementation.

We are interested in two kinds of schemes. One is the **sequential scheme**, that first solves the learning problem up to a predetermined accuracy and then solves the computational problem using the inexact solution to the first stage as a fixed problem parameter. The second one is the **simultaneous** scheme, that solves the computational and learning problems simultaneously. In fact, the simultaneous scheme is more important since in practice, when the size of the problem is very large, acquiring an accurate estimator of $\Sigma$ is often time consuming for large $n$, hence, one cannot expect the learning process to terminate in a reasonable time. Moreover, using the inexact solution to the learning problem will cause error propagation in the computational problem.
1.3 Alternating Direction Method of Multipliers

The Alternating Direction Method of Multipliers (ADMM) is a simple but powerful algorithm that is well suited to problems arising in applied statistics and machine learning. It can be viewed as an attempt to blend the benefits of dual decomposition and augmented Lagrangian methods for constrained optimization [22]. The history of ADMM dates back to early 1980s, and it has attracted attention recently due to its practical value to machine learning and data mining problems. In [23], Eckstein summarizes the reasons why ADMM is becoming popular: “ADMM performs reasonably well on recent applications and take the advantage of the structure of these problems, namely the separability of the objective function.” It is also pointed out that it is easy to implement the ADMM in a distributed-memory, parallel manner which is very important for “big data” problems [23]. A recent survey [22] by Boyd et al. provides many applications of ADMM and also has some numerical examples.

Consider the following convex optimization problem with decomposable objective and equality constraints:

\[
(P): \begin{array}{ll}
\text{minimize} & f(x) + g(y) \\
\text{subject to} & Ax + By = c,
\end{array}
\]

with variables \(x \in \mathbb{R}^n\) and \(y \in \mathbb{R}^m\), where \(A \in \mathbb{R}^{p \times n}\), \(B \in \mathbb{R}^{p \times m}\) and \(c \in \mathbb{R}^p\) are the given problem data. Here we assume \(f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}\) and \(g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}\) are both closed and convex.

The augmented Lagrangian function for this problem is

\[
L_{\rho}(x, y, z) = f(x) + g(y) + z^T(Ax + By - c) + \frac{\rho}{2}\|Ax + By - c\|_2^2,
\]

where \(z \in \mathbb{R}^p\) denotes the dual variable. ADMM scheme for solving problem \((P)\) consists of the following iterations [22]:

\[
x^{k+1} := \arg\min_x L_{\rho}(x, y^k, z^k),
\]

\[
y^{k+1} := \arg\min_y L_{\rho}(x^{k+1}, y, z^k),
\]

\[
z^{k+1} := z^k + \rho(Ax^{k+1} + By^{k+1} - c),
\]

with \(\rho > 0\) being the fixed penalty parameter. Equation (1.1) denotes the \(x\)-update
step, (1.2) is the $y$-update step and (1.3) is the dual step. In contrast, the method of multipliers for problem ($P$) is given by the following:

$$
(x^{k+1}, y^{k+1}) = \arg\min_{x,y} L_{\rho}(x, y, z^k),
$$

and

$$
z^{k+1} = z^k + \rho(Ax^{k+1} + By^{k+1} - c).
$$

Here the variables $x$ and $y$ are updated jointly, while in ADMM $x$ and $y$ are updated in sequence. Suppose that $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ are closed, proper and convex functions and that the Lagrangian $L_0$ has a saddle point, then it can be shown that ADMM satisfies the following [22]:

- Residual convergence, i.e., $r^k = Ax^k + By^k - c \to 0$ as $k \to \infty$.
- Objective convergence, i.e., $f(x^k) + g(y^k) \to p^*$ as $k \to \infty$, where $p^*$ is the optimal value of problem ($P$).
- Dual variable convergence, i.e., $z^k \to z^*$ as $k \to \infty$.

Under additional assumptions, it can be also shown that $x^k$ and $y^k$ converges to optimal solutions. Hence, ADMM indeed converges for the two learning problems considered in this thesis.

To quote from [22], “although some simple examples show that ADMM can be very slow to converge to high accuracy, it is often the case that ADMM converges to modest accuracy within a few tens of iterations.” For the convergence rate, in [24] it is shown that if either $f$ or $g$ is strongly convex with a Lipschitz continuous gradient, along with some rank assumptions on $A$ and $B$, then ADMM converges linearly. In [25], the global linear convergence rate is established for minimizing the sum of any number of convex separable functions belonging to a very special class, which answers a key question regarding the convergence of ADMM when the number of blocks is more than two. Furthermore, in [26], the authors study the convergence of ADMM as a matrix recurrence for the particular case of quadratic or linear programs, and the theory developed shows that ADMM should exhibit linear convergence when the iterates are close enough to optimal solution.
1.4 Simultaneous Scheme

According to robust optimization, $\Sigma^*$ is not known but belongs to an uncertainty set. However, for problems with parametric misspecification, even though $\Sigma^*$ is not known, we assume that there exists a learning problem such that its iterates $\Sigma^k \to \Sigma^*$ in the limit. The natural way to do this is by solving the learning problem first and then solving the computational problem, the so-called sequential scheme. In the sequential scheme, the process is as follows:

$$\Sigma^1 \to \Sigma^2 \to \cdots \to \Sigma^m, \ x^1 \to x^2 \to \cdots \to x^n,$$

hoping that $\Sigma^m$ is a good estimator of $\Sigma^*$ so that $x^n$ would be a good approximation of $x^*$. While in the simultaneous scheme, the process is as follows:

$$\Sigma^1 \to x^1 \to \Sigma^2 \to x^2 \to \cdots \to \Sigma^n \to x^n.$$

This process will indeed converge to optimal solutions to both computational and learning problems.

In [27], Jiang and Shanbhag investigates solving a stochastic convex optimization problem $\mathbb{E}[f(x; \theta^*, \xi)]$ over a closed and convex set $X$, where the data $\theta^*$ may be obtained by minimizing $\mathbb{E}[g(\theta; \eta)]$ over a closed and convex set $\Theta$. Jiang and Shanbhag provide a simultaneous scheme for simultaneously solving both the computational and learning problems. They show that when $f$ is convex, the simultaneous scheme has almost sure convergence property and provide the degradation in the convergence rate due to learning process. In [28], Ahmadi and Shanbhag investigate the deterministic problem $\min_x f(x; \theta^*)$ where $\theta^*$ is unknown but may be learned by a parallel learning process $\min_\theta g(\theta)$ under different assumptions on $f$. They examine the convergence rates of the simultaneous schemes. In particular, they show that when both computational and learning problems are strongly convex, then the simultaneous scheme has a sub-linear convergence rate, in contrast to the linear rate when problem data are known.
Description of Algorithms

In this chapter, we introduce the algorithms used to resolve the computational and learning problems. We will denote the objective function of the computational problem \((C)\) by \(f(x; \Sigma) := \frac{1}{2}x^T \Sigma x - \kappa \mu^T x\) and the feasible set of \((C)\) by \(X := \{x \in \mathbb{R}^n | 1^T x = 1, x \geq 0\}\). The first section of this chapter reviews the basics of projected gradient method, along with an efficient algorithm used to project an element onto probabilistic simplex. In Sections 2.2, 2.3 and 2.4 we discuss how to implement ADMM to learning problems \((\mathcal{L}_{SCS})\) and \((\mathcal{L}_{SPS})\). The remaining of this chapter focuses on sequential and simultaneous schemes.

2.1 Projected Gradient Method

Assuming the covariance matrix \(\Sigma\) is known, we may solve the computational problem \((C)\) using the projected gradient method:

\[
x_{k+1} = \Pi_X(x_k - \gamma_k \nabla_x f(x_k; \Sigma)),
\]

where \(\Pi_X(y)\) denotes the Euclidean projection of \(y\) onto the feasible set \(X\), \(\gamma_k > 0\) is an appropriately chosen step length, and \(\nabla_x f(x; \Sigma) = \Sigma x - \kappa \mu\) is the gradient of the objective \(f\) with respect to variable \(x\). To deal with the projection operator \(\Pi_X(\cdot)\), we use an algorithm proposed in [29]. The algorithm is displayed as Algorithm 1. In the computational steps of both simultaneous and sequential schemes, we will use projected gradient method with constant step length, i.e., \(\gamma_k = \gamma\) for all \(k\). For
the sequential scheme, we choose $\gamma$ to be $\frac{1}{\|\Sigma\|}$, while for the simultaneous scheme, the choice of step length is discussed in Theorem 2 in Section 3.2.1.

Algorithm 1: Projection onto probabilistic simplex

**Data:** A vector $x \in \mathbb{R}^n$

**Result:** $\Pi_X(x)$, the projection of $x$ onto the probabilistic simplex

**begin**

Sort $x$ into $y$ in descending order: $y_1 \geq \ldots \geq y_n$ so that $y$ is the sorted $x$

Find $\rho = \max \left\{ j : y_j - \frac{1}{j} \left( \sum_{k=1}^{j} y_k - 1 \right) > 0 \right\}$

Define $\eta = \frac{1}{\rho} \left( \sum_{k=1}^{\rho} y_k - 1 \right)$

$[\Pi_X(x)]_k = \max(x_k - \eta, 0)$

**end**

2.2 ADMM for SCS Learning Problem: First Approach

To better understand the simultaneous scheme and sequential scheme for solving the problem proposed in Chapter 2, it is essential to understand how ADMM works for solving the learning problems. This section is devoted to application of ADMM to SCS learning problems, while in the next section we focus on ADMM for SPS learning problem.

In order to apply ADMM to SCS learning problem $(\mathcal{L}_{SCS})$, we first introduce a new variable $\Theta$ and an equality constraint as follows:

$$(\mathcal{L}_{SCS}^1): \minimize_{\Theta, \Sigma} ||\Sigma - S||_F^2/2 + \lambda||\Sigma||_1 + 1_Q(\Theta)$$

subject to $\Sigma = \Theta$,

where $1_Q$ is the indicator function of the set $Q = \{ \Sigma \in \mathbb{S}^n : \Sigma \succeq \epsilon I \}$. Define two functions $\Psi_1 : \mathbb{S}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ and $\Phi_1 : \mathbb{S}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ such that $\Psi_1(\Sigma) = ||\Sigma - S||_F^2/2 + \lambda||\Sigma||_1$ and $\Phi_1(\Sigma) = 1_Q(\Sigma)$ for all $\Sigma \in \mathbb{S}^n$. Clearly, $(\mathcal{L}_{SCS}^1)$ is equivalent to $(\mathcal{L}_{SCS})$.

To deal with the equality constraint, we write the augmented Lagrangian with
penalty parameter $\rho > 0$ as:

$$L_\rho(\Sigma, \Theta; \Lambda) = \Psi_1(\Sigma) + \Phi_1(\Theta) + \langle \Sigma - \Theta, \Lambda \rangle + \frac{\rho}{2}\|\Sigma - \Theta\|^2_F,$$

where $\Lambda$ is the Lagrangian multiplier. Given initial points $\Theta^0$ with $\Theta^0$ and $\Lambda^0$, ADMM updates variables $\Sigma$ and $\Theta$ alternatingly as follows:

- **$\Sigma$ step:** $\Sigma^{k+1} := \arg\min_\Sigma L_\rho(\Sigma, \Theta^k; \Lambda^k)$, \hspace{1cm} (2.2)
- **$\Theta$ step:** $\Theta^{k+1} := \arg\min_\Theta L_\rho(\Sigma^{k+1}, \Theta; \Lambda^k)$, \hspace{1cm} (2.3)
- **$\Lambda$ step:** $\Lambda^{k+1} := \Lambda^k + \rho(\Sigma^{k+1} - \Theta^{k+1})$. \hspace{1cm} (2.4)

Note that both $\Sigma$ step and $\Theta$ step require solving sub-problems. Fortunately, closed forms for these two steps are available in [16]. Note the slight difference between our approach and the one in [16]. In each iteration, our approach first updates $\Sigma$ while in [16] $\Theta$ is updated first. Use the notation in [16], denote $(Z)_+$ the projection of $Z \in \mathbb{S}^n$ onto the cone $Q = \{\Sigma \in \mathbb{S}^n : \Sigma \succeq \epsilon I\}$. If the matrix $Z$ has the eigen-decomposition $\sum_{i=1}^n \lambda_i v_i v_i^T$, then we may compute $(Z)_+$ as follows:

$$ (Z)_+ = \sum_{i=1}^n \max(\lambda_i, \epsilon) v_i v_i^T. \hspace{1cm} (2.5) $$

Furthermore, for matrix $Z \in \mathbb{S}^n$ and a positive parameter $\tau$ define:

$$ S(Z, \tau) = (s(z_{ij}, \tau))_{i,j=1}^n, \quad s(z_{ij}, \tau) = \text{sign}(z_{ij})\max(|z_{ij}| - \tau, 0)I_{\{i \neq j\}} + z_{ij}I_{\{i = j\}}, \hspace{1cm} (2.6) $$

where $I_A$ equals 1 if condition $A$ is true and 0 otherwise.

In [16], Xue et al. have given the explicit formulae for $\Sigma$ step and $\Theta$ step. For the sake of completeness, we reproduce the statements and provide complete proofs for the $\Sigma$ and $\Theta$ update in Lemma 1 and Lemma 2.

**Lemma 1.** At the $(k+1)$th iteration, the $\Sigma$ step is given by

$$ \Sigma^{k+1} = \arg\min_\Sigma L_\rho(\Sigma, \Theta^k; \Lambda^k) = S \left( \frac{1}{\rho}(S - \Lambda^k) + \Theta^k, \frac{\lambda}{\rho} \right) \big/ \left( 1 + \frac{1}{\rho} \right). \hspace{1cm} (2.7) $$
Proof. We need to solve the problem:

$$
\arg\min_{\Sigma} \|\Sigma - S\|_F^2 / 2 + \lambda|\Sigma|_1 + \langle \Sigma - \Theta^k, \Lambda^k \rangle + \frac{\rho}{2}\|\Theta^k - \Sigma\|_F^2. \tag{2.8}
$$

By definition of Frobenius norm and inner product of matrices, problem (2.8) is equivalent to:

$$
\arg\min_{\Sigma_{ij}} \frac{1}{2} \left( \sum_{i,j=1}^n (\Sigma_{ij} - S_{ij})^2 \right) + \lambda \sum_{i,j=1}^n |\Sigma_{ij}| + \sum_{i,j=1}^n \Lambda^k_{ij} \Sigma_{ij} + \frac{\rho}{2} \left( \sum_{i,j=1}^n (\Sigma_{ij} - \Theta^k_{ij})^2 \right).
$$

Now it is obvious that this problem is separable. We consider two cases:

**Case 1:** $i = j$. We need to solve $n$ problems of the following form:

$$
\arg\min_{\Sigma_{ij}} \frac{1}{2} (\Sigma_{ij} - S_{ij})^2 + \Lambda_i^k \Sigma_{ij} + \frac{\rho}{2} (\Sigma_{ij} - \Theta^k_{ij})^2, 
$$

where $\Sigma_{ij}$ is the decision variable, or equivalently,

$$
\arg\min_{\Sigma_{ij}} \frac{1}{\rho} (\Sigma_{ij} - S_{ij})^2 + \left( \frac{2}{\rho} \Lambda_i^k \right) \Sigma_{ij} + (\Sigma_{ij} - \Theta^k_{ij})^2, \tag{2.9}
$$

$$
\equiv \arg\min_{\Sigma_{ij}} \left( 1 + \frac{1}{\rho} \right) \Sigma_{ij}^2 - 2 \left( \frac{1}{\rho} (S_{ij} - \Lambda_i^k) + \Theta^k_{ij} \right) \Sigma_{ij}. \tag{2.10}
$$

This problem is a quadratic problem in one dimension and its optimal solution is given by:

$$
\Sigma^*_{ij} = \left( \frac{1}{\rho} (S_{ij} - \Lambda_i^k) + \Theta^{k+1}_{ij} \right) / \left( 1 + \frac{1}{\rho} \right) \tag{2.11}
$$

**Case 2:** $i \neq j$. We need to solve $\frac{n^2-n}{2}$ problems of the following form:

$$
\arg\min_{\Sigma_{ij}} \frac{1}{2} (\Sigma_{ij} - S_{ij})^2 + \Lambda_i^k \Sigma_{ij} + \frac{\rho}{2} (\Sigma_{ij} - \Theta^{k+1}_{ij})^2 + \lambda|\Sigma_{ij}|,
$$

or equivalently,

$$
\arg\min_{\Sigma_{ij}} \frac{1}{\rho} (\Sigma_{ij} - S_{ij})^2 + \left( \frac{2}{\rho} \Lambda_i^k \right) \Sigma_{ij} + (\Sigma_{ij} - \Theta^{k+1}_{ij})^2 + \frac{2\lambda}{\rho} |\Sigma_{ij}|. \tag{2.12}
$$
\begin{align*}
\equiv \arg\min_{\Sigma_{ij}} \left(1 + \frac{1}{\rho}\right) \Sigma_{ij}^2 - 2 \left(\frac{1}{\rho} (S_{ij} - \Lambda_{ij}^k) + \Theta_{ij}^{k+1}\right) \Sigma_{ij} + \frac{2\lambda}{\rho} |\Sigma_{ij}|. \quad (2.13)
\end{align*}

Define \( a := 1 + \frac{1}{\rho} > 0, \) \( b_{ij} := 2 \left(\frac{1}{\rho} (S_{ij} - \Lambda_{ij}^k) + \Theta_{ij}^{k+1}\right) \) and \( c := \frac{2\lambda}{\rho}, \) then we need to solve
\begin{align*}
\arg\min_{\Sigma_{ij}} \ a \Sigma_{ij}^2 - b_{ij} \Sigma_{ij} + c |\Sigma_{ij}|. \quad (2.14)
\end{align*}

If \( b_{ij} \geq 0, \) then the optimal solution \( \Sigma_{ij}^* \geq 0 \) and problem (2.14) is equivalent to \( \arg\min_{\Sigma_{ij} \geq 0} \ a \Sigma_{ij}^2 + (c - b_{ij}) \Sigma_{ij}. \) So the optimal solution is:
\begin{align*}
\Sigma_{ij}^* = \max \left( \frac{b_{ij} - c}{2a}, 0 \right) = \max \left( \frac{|b_{ij}| - c}{2a}, 0 \right). \quad (2.15)
\end{align*}

If \( b_{ij} < 0, \) then the optimal solution \( \Sigma_{ij}^* \leq 0 \) and problem (2.14) is equivalent to \( \arg\min_{\Sigma_{ij} \leq 0} \ a \Sigma_{ij}^2 - (b_{ij} + c) \Sigma_{ij}. \) So the optimal solution is:
\begin{align*}
\Sigma_{ij}^* = \min \left( \frac{b_{ij} + c}{2a}, 0 \right) = -\max \left( -\frac{b_{ij} - c}{2a}, 0 \right) = -\max \left( \frac{|b_{ij}| - c}{2a}, 0 \right). \quad (2.16)
\end{align*}

The result follows from the definition of \( S. \)

**Lemma 2.** At the \((k+1)^{th}\) iteration, the \(\Theta\) step is given by
\begin{align*}
\Theta^{k+1} = \arg\min_{\Theta} L_{\rho}(\Sigma^{k+1}, \Theta; \Lambda^k) = \left(\Sigma^{k+1} + \frac{1}{\rho} \Lambda^k\right)^+. \quad (2.17)
\end{align*}

**Proof.** We need to solve the problem:
\begin{align*}
\arg\min_{\Theta \geq \ell} \|\Sigma^{k+1} - S\|_F^2 / 2 + \lambda |\Sigma^{k+1}|_1 + \langle \Sigma^{k+1} - \Theta, \Lambda^k \rangle + \frac{\rho}{2} \|\Sigma^{k+1} - \Theta\|^2_F. \quad (2.18)
\end{align*}

Since \( \Sigma^{k+1} \) and \( \Lambda^k \) are fixed in this problem, the problem (2.18) is equivalent to:
\begin{align*}
\arg\min_{\Theta \geq \ell} -\langle \Lambda^k, \Theta \rangle + \frac{\rho}{2} \|\Theta - \Sigma^{k+1}\|^2_F, \\
\equiv \arg\min_{\Theta \geq \ell} \|\Theta - \Sigma^{k+1}\|^2_F - 2 \left\langle \Theta, \frac{1}{\rho} \Lambda^k \right\rangle, \\
\equiv \arg\min_{\Theta \geq \ell} \|\Theta - \Sigma^{k+1}\|^2_F - 2 \left\langle \Theta, \frac{1}{\rho} \Lambda^k \right\rangle + 2 \left\langle \Sigma^{k+1}, \frac{1}{\rho} \Lambda^k \right\rangle + \left\langle \frac{1}{\rho} \Lambda^k, \frac{1}{\rho} \Lambda^k \right\rangle, \\
\end{align*}
The result follows from equation (2.5). \hfill \square

Lemma 1 and Lemma 2 are useful when we implement the algorithm, since they give us the explicit formulae for computing $\Sigma$ and $\Theta$ update steps. The ADMM for solving $\mathcal{L}_{SCS}$ is given in Algorithm 2.

\textbf{Algorithm 2: ADMM for SCS learning problem}

\textbf{Data:} Sample covariance matrix $S$, Initial points $\Theta^0$ and $\Lambda^0$

\textbf{Result:} Covariance estimator $\Sigma^*$

\begin{algorithmic}
\State \textbf{begin}
\State \textbf{for} $k = 0$ \textbf{to} MaxItr \textbf{do}
\State \hspace{1em} $\Sigma^{k+1} = S(\frac{1}{\rho}(S - \Lambda^k) + \Theta^k; \frac{\Lambda^k}{\rho})/(1 + \frac{1}{\rho})$
\State \hspace{1em} $\Theta^{k+1} = (\Sigma^{k+1} + \frac{1}{\rho}\Lambda^k)_+$
\State \hspace{1em} $\Lambda^{k+1} = \Lambda^k + \rho(\Sigma^{k+1} - \Theta^{k+1})$
\State \hspace{2em} \textbf{while} Primal and dual residuals are below a specified tolerance \textbf{do}
\State \hspace{3em} \textbf{stop} \textbf{for}-loop;
\State \hspace{2em} \textbf{end}
\State \hspace{1em} \textbf{end}
\State $\Sigma^* = \Sigma^k$
\State \textbf{end}
\end{algorithmic}

It has been shown in [16] that when ADMM is used for solving SCS learning problem, it indeed converges.

\subsection{ADMM for SCS Learning Problem: Second Approach}

Though the ADMM discussed above may successfully solve the SCS learning problem, it has an important theoretical drawback that it is hard to estimate its convergence rate. In this section, we discuss an alternative approach to solve the SCS
learning problem, for which the rate can be established using known results in the literature.

Using variable splitting, we can rewrite \((\mathcal{L}_{SCS})\) equivalently as follows:

\[
\begin{align*}
\min_{\Sigma, \Theta \in \mathbb{S}^n} & \quad \frac{1}{2} \| \Sigma - S \|_F^2 + 1_Q(\Sigma) + \lambda |\Theta|_1 \\
\text{subject to} & \quad \Sigma = \Theta,
\end{align*}
\]

where \(1_Q\) is the indicator function of \(Q\).

Define \(\Phi_2(\Sigma) := \frac{1}{2} \| \Sigma - S \|_F^2 + 1_Q(\Sigma)\) and \(\Phi_2(\Theta) := \lambda |\Theta|_1\), and let \(L_\rho(\Sigma, \Theta; \Lambda)\) be given by

\[
L_\rho(\Sigma, \Theta; \Lambda) = \Phi_2(\Sigma) + \Phi_2(\Theta) + \langle \Lambda, \Sigma - \Theta \rangle + \frac{\rho}{2} \| \Sigma - \Theta \|_F^2.
\]

Given initial points \(\Theta^0\) and \(\Lambda^0\), ADMM steps can be written as follows:

\[
\begin{align*}
\Sigma \text{ step} : & \quad \Sigma^{k+1} = \arg\min_{\Sigma} L_\rho(\Sigma, \Theta^k; \Lambda^k), \\
\Theta \text{ step} : & \quad \Theta^{k+1} = \arg\min_{\Theta} L_\rho(\Sigma^{k+1}, \Theta; \Lambda^k), \\
\Lambda \text{ step} : & \quad \Lambda^{k+1} = \Lambda^k + \rho(\Sigma^{k+1} - \Theta^{k+1}).
\end{align*}
\]

Before we give explicit formulae for \(\Sigma\) and \(\Theta\) steps, we comment why this approach is theoretically more favorable. Recall that in this new approach we set \(\Psi_2(\Sigma) := \frac{1}{2} \| \Sigma - S \|_F^2 + 1_Q(\Sigma)\) and \(\Phi_2(\Theta) := \lambda |\Theta|_1\). But if we use the original ADMM developed in \([16]\), we have \(\Psi_1(\Sigma) := \frac{1}{2} \| \Sigma - S \|_F^2 + \lambda |\Sigma|_1\) and \(\Phi_1(\Theta) = 1_Q(\Theta)\). Note that the original \(\Psi_1\) is not even continuously differentiable because of the off-diagonal norm term. On the other hand, our new \(\Psi_2\) is strongly convex and has a Lipschitz continuous gradient \(\nabla \Psi_2(\Sigma) = \Sigma - S\) on the set \(Q\). Deng and Yin \([24]\) show that ADMM has a linear convergence rate under this assumption.

Now we give explicit formulae for the new approach in the following lemmas:

**Lemma 3.** In \(\Sigma\) step, \(\Sigma^{k+1}\) is given by

\[
\Sigma^{k+1} = \left( \frac{S - \Lambda^k + \rho \Theta^k}{1 + \rho} \right)_+.
\]
Lemma 4. In Θ step, $\Theta^{k+1}$ is given by

$$\Theta^{k+1} = S \left( \Sigma^{k+1} + \frac{1}{\rho} \Lambda^k, \frac{\lambda}{\rho} \right),$$

where $S$ is given by equation (2.6).

Proofs of Lemmas 3 and 4 is similar to Lemmas 1 and 2 thus omitted. The ADMM using the new approach is given as Algorithm 3.

Algorithm 3: ADMM for SCS learning problem: 2nd approach

\textbf{Data:} Sample covariance matrix $S$, Initial points $\Theta^0$ and $\Lambda^0$
\textbf{Result:} Covariance estimator $\Sigma^*$

\begin{verbatim}
begin
for $k = 0$ to MaxItr do
    $\Sigma^{k+1} = \left( \frac{S - \Lambda^k + \rho \Theta^k}{1 + \rho} \right)_+$
    $\Theta^{k+1} = S \left( \Sigma^{k+1} + \frac{1}{\rho} \Lambda^k, \frac{\lambda}{\rho} \right)$
    $\Lambda^{k+1} = \Lambda^k + \rho (\Sigma^{k+1} - \Theta^{k+1})$
while Primal and dual residuals are below a specified tolerance do
    stop for-loop;
end
end
$\Sigma^* = \Sigma^k$
\end{verbatim}

2.4 ADMM for SPS Learning Problem

In this section, we discuss the application of ADMM to SPS learning problem ($\mathcal{L}_{SPS}$). For a more thorough discussion of this problem, see [30].

To solve ($\mathcal{L}_{SPS}$), we first consider a more general problem of the form:

$$\minimize_{P \in \mathbb{S}^n} \Psi(P) + \Phi(P), \quad (2.22)$$

where $\mathbb{S}^n$ denotes the space of all $n \times n$ symmetric matrices and $\Psi : \mathbb{S}^n \to \mathbb{R} \cup \{+\infty\}$ and $\Phi : \mathbb{S}^n \to \mathbb{R} \cup \{+\infty\}$ are proper closed convex functions. To use ADMM, we
first introduce a new variable $Z$ and an equality constraint as follows:

$$\begin{align*}
\text{minimize} & \quad \Psi(P) + \Phi(Z) \\
\text{subject to} & \quad P = Z, \\
& \quad Z \in \mathbb{S}^n.
\end{align*}$$

(2.23)

Clearly (2.22) and (2.23) are equivalent. To deal with the equality constraint, given a penalty parameter $\rho$, write down the augmented Lagrangian for problem (2.23) as:

$$L_\rho(P, Z, W) = \Psi(P) + \Phi(Z) + \langle W, P - Z \rangle + \frac{\rho}{2} \| P - Z \|_F^2.$$ 

Given initial points $Z^0$ and $W^0$, the ADMM steps are given as:

$$\begin{align*}
P \text{ update:} & \quad P^{k+1} := \arg\min_{P \in \mathbb{S}^n} L_\rho(P, Z^k, W^k), \\
Z \text{ update:} & \quad Z^{k+1} := \arg\min_{Z \in \mathbb{S}^n} L_\rho(P^{k+1}, Z, W^k), \\
W \text{ update:} & \quad W^{k+1} := W^k + \rho(P^{k+1} - Z^{k+1}).
\end{align*}$$

(2.24) (2.25) (2.26)

To derive closed formulae for the above algorithm, we define two prox functions $\prox_{\lambda\Psi} : \mathbb{S}^n \to \mathbb{S}^n$ and $\prox_{\lambda\Phi} : \mathbb{S}^n \to \mathbb{S}^n$ for a given $\lambda > 0$ as follows:

$$\begin{align*}
\prox_{\lambda\Psi}(B) & := \arg\min_{A \in \mathbb{S}^n} \lambda\Psi(A) + \frac{1}{2} \| A - B \|_F^2, \\
\prox_{\lambda\Phi}(B) & := \arg\min_{A \in \mathbb{S}^n} \lambda\Phi(A) + \frac{1}{2} \| A - B \|_F^2.
\end{align*}$$

(2.27) (2.28)

It follows that ADMM iterations can be rewritten as:

$$\begin{align*}
P^{k+1} & = \prox_{\psi/\rho}(Z^k - \frac{1}{\rho}W^k), \\
Z^{k+1} & = \prox_{\phi/\rho}(P^{k+1} + \frac{1}{\rho}W^k), \\
W^{k+1} & = W^k + \rho(P^{k+1} - Z^{k+1}).
\end{align*}$$

(2.29) (2.30) (2.31)
Indeed, to derive (2.29), note that from (2.24) we have

\[ P^{k+1} = \arg\min_{P \in \mathbb{S}^n} \Psi(P) + \Phi(Z^k) + \langle W^k, P - Z^k \rangle + \frac{\rho}{2} \|P - Z^k\|_F^2 \]

\[ = \arg\min_{P \in \mathbb{S}^n} \frac{1}{\rho} \Psi(P) + \langle P, \frac{1}{\rho} W^k \rangle + \frac{1}{2} \langle P - Z^k, P - Z^k \rangle \]

\[ = \arg\min_{P \in \mathbb{S}^n} \frac{1}{\rho} \Psi(P) + \frac{1}{2} \langle P - (Z^k - \frac{1}{\rho} W^k), P - (Z^k - \frac{1}{\rho} W^k) \rangle \]

\[ = \arg\min_{P \in \mathbb{S}^n} \frac{1}{\rho} \Psi(P) + \frac{1}{2} \|P - (Z^k - \frac{1}{\rho} W^k)\|_F^2 \]

\[ = \text{prox}_{\Psi/\rho}(Z^k - \frac{1}{\rho} W^k), \]

where the last step follows from (2.27). Equation (2.30) can be derived similarly.

Let \( a := \frac{1}{\|S\|_2 + n} \) and \( b := \frac{n}{a} \). Then it has been shown in [30] that \( \mathcal{L}_{SPS} \) is equivalent to problem (2.22) with:

\[ \Psi(P) = \langle S, P \rangle - \log \det(P) + 1_\mathcal{Q}(P), \mathcal{Q} := \{P \in \mathbb{S}^n : aI \preceq P \preceq bI\}, \]

\[ \Phi(P) = \alpha \|P\|_1 + 1'_\mathcal{Q}(P), \mathcal{Q}' := \{P \in \mathbb{S}^n : \text{diag}(P) \succeq 0\}. \]

Since \( \mathcal{L}_{SPS} \) is equivalent to (2.22) with above \( \Psi \) and \( \Phi \), it suffices to derive explicit formula for functions \( \text{prox}_{\Psi/\rho} \) and \( \text{prox}_{\Phi/\rho} \). These formulae are given by the following two lemmas [30]:

**Lemma 5.** For given \( A \in \mathbb{S}^n \) and \( \rho > 0 \), we have

\[ (\text{prox}_{\Psi/\rho}(A))_{ij} = \text{sign}(A_{ij}) \max \left( |A_{ij}| - \frac{\alpha}{\rho}, 0 \right) I_{\{i \neq j\}} + \max \left( A_{ij} - \frac{\alpha}{\rho}, 0 \right) I_{\{i = j\}} \]

(2.34)

**Lemma 6.** For given \( A \in \mathbb{S}^n \) and \( \rho > 0 \), if \( A - \frac{1}{\rho} S \) has eigen-decomposition \( A - \frac{1}{\rho} S = U \text{diag}(\lambda) U^T \), then \( \text{prox}_{\Phi/\rho}(A) = U \text{diag}(\lambda^*) U^T \), where

\[ \lambda_i^* = \max \left( \min \left( \frac{\bar{\lambda}_i + \sqrt{\bar{\lambda}_i^2 + 4\rho}}{2\rho}, b \right), a \right), i = 1, \ldots, n. \]

(2.35)

Now the ADMM for solving SPS learning problem \( \mathcal{L}_{SPS} \) is displayed as Algorithm 4.
Algorithm 4: ADMM for SPS learning problem

Data: Sample covariance matrix $S$, Initial points $Z_0$ and $W_0$

Result: Covariance estimator $\Sigma^*$

begin
for $k = 0$ to $MaxItr$ do
    $P^{k+1} = \text{prox}_{\psi/\rho}(Z^k - \frac{1}{\rho}W^k)$
    $Z^{k+1} = \text{prox}_{\phi/\rho}(P^{k+1} + \frac{1}{\rho}W^k)$
    $W^{k+1} = W^k + \rho(P^{k+1} - Z^{k+1})$
    while Primal and dual residuals are below a specified tolerance do
        stop for-loop
    end
end
$\Sigma^* = (P^k)^{-1}$:

end

Note that in this algorithm, we are calculating the inverse of the covariance matrix $\Sigma^*$.

2.5 Sequential Scheme

In the sequential scheme, we first solve the learning problem to acquire the estimator of covariance matrix and then solve the computational problem.

2.5.1 Markowitz Optimization with SCS Learning Problem: First Approach

Using the same notation introduced above, the sequential scheme for Markowitz portfolio optimization with SCS learning problem is displayed in Algorithm 5.
Algorithm 5: Sequential scheme: Markowitz optimization with SCS learning problem: 1st approach

input : $S,x_0,\Sigma_0,\Lambda_0$,tolerance, MaxItr  
output: $\Sigma^*, x^*$  
/* ADMM begins */  
while dual residual or primal residual are above the tolerance do  
  $\Sigma^{k+1} = S\left(\frac{1}{\rho}(S - \Lambda^k) + \Theta^k, \frac{1}{\rho}\right)/(1 + \frac{1}{\rho})$;  
  $\Theta^{k+1} = (\Sigma^{k+1} + \frac{1}{\rho}\Lambda^k)_+$;  
  $\Lambda^{k+1} = \Lambda^k + \rho(\Sigma^{k+1} - \Theta^{k+1})$;  
end  
$\Sigma^* = \Sigma^k$;  /* Now we have the estimator for the covariance */  
/* Computational steps begin */  
$\gamma = \frac{1}{\|\Sigma^*\|}$;  
for $i = 0 : \text{MaxItr}$ do  
  $x^{i+1} = \Pi_X(x^i - \gamma \nabla f(x^i; \Sigma^*))$;  
end  
$x^* = x^i$;  /* We solve the computational problem */

2.5.2 Markowitz Optimization with SCS Learning Problem: Second Approach

Algorithm 6: Sequential scheme: Markowitz optimization with SCS learning problem: 2nd Approach

input : $S,x_0,\Theta^0,\Lambda^0$,tolerance, MaxItr  
output: $\Sigma^*, x^*$  
/* ADMM begins */  
while dual residual or primal residual are above the tolerance do  
  $\Sigma^{k+1} = S\left(S - \Lambda^k + \rho\Theta^k\right)_+$;  
  $\Theta^{k+1} = S\left(\Sigma^{k+1} + \frac{1}{\rho}\Lambda^k, \frac{1}{\rho}\right)$;  
  $\Lambda^{k+1} = \Lambda^k + \rho(\Sigma^{k+1} - \Theta^{k+1})$;  
end  
$\Sigma^* = \Sigma$;  /* Now we have the estimator for the covariance */  
/* Computational steps begin */  
$\gamma = \frac{1}{\|\Sigma^*\|}$;  
for $i = 0 : \text{MaxItr}$ do  
  $x^{i+1} = \Pi_X(x^i - \gamma \nabla f(x^i; \Sigma^*))$;  
end  
$x^* = x$;  /* We solve the computational problem */
2.5.3 Markowitz Optimization with SPS Learning Problem

Use the notations above, the sequential scheme for Markowitz portfolio optimization with SPS learning problem is displayed as Algorithm 7:

**Algorithm 7**: Sequential scheme: Markowitz optimization with SPS learning problem

**input**: $S, x_0, W_0, Z_0, \text{tolerance, MaxItr}$

**output**: $\Sigma^*, x^*$

/* ADMM begins */

while dual residual or primal residual are above the tolerance do

$$P_{k+1} = \text{prox}_{\psi/\rho}(Z_k - \frac{1}{\rho} W_k);$$
$$Z_{k+1} = \text{prox}_{\phi/\rho}(P_{k+1} + \frac{1}{\rho} W_k);$$
$$W_{k+1} = W_k + \rho(P_{k+1} - Z_{k+1});$$

end

$\Sigma^* = (P^k)^{-1};$ /* We get the covariance estimator */

/* Computational steps begin */

$\gamma = \frac{1}{\|\Sigma^*\|};$

for $i = 0$ to MaxItr do

$$x^{i+1} = \Pi_X(x^i - \gamma \nabla f(x^i; \Sigma^*));$$

end

$x^* = x;$ /* We solve the computational problem */

2.6 Simultaneous Scheme

2.6.1 Markowitz Optimization with SCS Learning Problem: First Approach

Use the same notations as before, the simultaneous scheme for Markowitz optimization with SCS learning problem is displayed in Algorithm 8:
Algorithm 8: Simultaneous scheme: Markowitz optimization with SCS learning problem: 1st approach

**input**: $S, x_0, \Sigma_0, \Lambda_0, \text{tolerance}, \text{MaxItr}$

**output**: $\Sigma^*, x^*$

Choose step length $\gamma$ according to Theorem 1;

while dual residual or primal residual are above the tolerance do

\[
\Sigma^{k+1} = S\left(\frac{1}{\rho} (S - \Lambda^k) + \Theta, \frac{1}{\rho}\right)/(1 + \frac{1}{\rho}); \\
\Theta^{k+1} = (\Sigma^{k+1} + \frac{1}{\rho} \Lambda^k)_+; \\
\Lambda^{k+1} = \Lambda^k + \rho(\Sigma^{k+1} - \Theta^{k+1}); \\
/* After learn the covariance, we take one or several computational steps */ \\
x^{k+1} = \Pi_X(x^k - \gamma \nabla f(x^k; \Sigma^{k+1})) \\
\]

end

$\Sigma^* = \Sigma^k; \quad /*$ The true covariance estimator $*/$

$x^* = x^k; \quad /*$ optimal solution for Markowitz optimization $*/$

2.6.2 Markowitz Optimization with SCS Learning Problem: Second Approach

Algorithm 9: Simultaneous scheme: Markowitz optimization with SCS learning problem: 2nd approach

**input**: $S, x_0, \Sigma_0, \Lambda_0, \text{tolerance}, \text{MaxItr}$

**output**: $\Sigma^*, x^*$

Choose step length $\gamma$ according to Theorem 1;

while dual residual or primal residual are above the tolerance do

\[
\Sigma^{k+1} = \left(S - \Lambda^k + \rho \Theta^k\right)_+; \\
\Theta^{k+1} = S \left(\Sigma^{k+1} + \frac{1}{\rho} \Lambda^k, \frac{1}{\rho}\right); \\
\Lambda^{k+1} = \Lambda^k + \rho(\Sigma^{k+1} - \Theta^{k+1}); \\
/* After one learning step, we take one or several computational steps */ \\
x^{k+1} = \Pi_X(x^k - \gamma \nabla f(x^k; \Sigma^{k+1})) \\
\]

end

$\Sigma^* = \Sigma^k; \quad /*$ The true covariance estimator $*/$

$x^* = x^k; \quad /*$ optimal solution for Markowitz optimization $*/$
2.6.3 Markowitz Optimization with SPS Learning Problem

Use the same notations as before, the simultaneous scheme for Markowitz optimization with SPS learning problem is displayed in Algorithm 10:

**Algorithm 10:** Simultaneous scheme: Markowitz optimization with SPS learning problem

- **input:** $S, x_0, W_0, Z_0$, tolerance, MaxItr
- **output:** $\Sigma^*, x^*$

Choose step length $\gamma$ according to Theorem 1;

**while** dual residual or primal residual are above the tolerance **do**

\[
P^{k+1} = \text{prox}_{\psi/\rho}(Z^k - \frac{1}{\rho}W^k);
\]

\[
Z^{k+1} = \text{prox}_{\phi/\rho}(P^{k+1} + \frac{1}{\rho}W^k);
\]

\[
W^{k+1} = W^k + \rho(P^{k+1} - Z^{k+1});
\]

/* Before take computational steps, we first need to invert the estimator $P$ */

\[
\Sigma^{k+1} = (P^{k+1})^{-1};
\]

\[
x^{k+1} = \Pi_X(x^k - \gamma \nabla f(x^k; \Sigma^{k+1}))
\]

**end**

$\Sigma^* = \Sigma^k;$ /* Estimator of true covariance matrix */

$x^* = x^k;$ /* Optimal solution for Markowitz optimization */
Theoretical Results for the Simultaneous Scheme

In this chapter, we analyze the convergence properties of schemes discussed in Chapter 2. In Section 3.1, we prove the Lipschitz continuity of matrix inverse function on the set of matrices whose minimal eigenvalue is greater than or equal to a positive real number. In Section 3.2, we analyze the convergence behavior of simultaneous scheme. In particular, we show that under the assumption that $\Sigma^*$ is bounded below and above, with an appropriately chosen constant step length, the simultaneous scheme produces a convergent sequence $\{x^k\}$ which converges at a sub-linear rate.

3.1 Lipschitz Continuity of Matrix Inverse

Let $\mathbb{R}^{n \times n}$ denote the space of all $n \times n$ real matrices and let $V^{n \times n}$ denote the space of all nonsingular $n \times n$ real matrices. Define $F : V^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ by $F(A) = A^{-1}$ for all $A \in V^{n \times n}$ so that $F$ is the matrix inverse function. In the remainder of this thesis, $\|A\|$ denotes the $\ell_2$ norm of a square matrix $A$. The following standard result from numeric linear algebra is often useful:

**Lemma 7.** *For a symmetric $n \times n$ real matrix $A$, we have that*

$$\|A\| = \max_{i=1,\ldots,n} |\lambda_i|,$$
where \( \{\lambda_1, \ldots, \lambda_n\} \) denotes the eigenvalues of \( A \).

In this section, we establish the Lipschitz continuity of \( F \) on some special sets. To establish this, we need the following lemma from [31].

**Lemma 8.** The function \( F \) is differentiable, and its derivative at \( A \in V^{n \times n} \) is given by \( [DF(A)] H = -A^{-1} H A^{-1} \) for any \( H \in \mathbb{R}^{n \times n} \).

We also utilize the following generalization of mean value theorem to Banach spaces [32].

**Lemma 9.** (Mean Value Theorem) Suppose \( E \) and \( E' \) are Banach spaces, \( G \) is a subset of \( E \), and \( f : G \to E' \). Suppose \( x \) and \( y \) are elements of \( G \) and that the closed line segment joining \( x \) and \( y \) is contained in \( G \). Assume that \( f \) is continuous at each point of the closed line segment joining \( x \) to \( y \), i.e., at each point \( (1-t)x + ty \) for \( 0 \leq t \leq 1 \). Further assume that \( f \) is differentiable at each point on the open line segment joining \( x \) and \( y \), i.e., at each point \( (1-t)x + ty \) for \( 0 < t < 1 \). Then the following hold:

1. There exists a \( t^* \in (0, 1) \) such that
   \[
   \|f(y) - f(x)\| \leq \|df_z(y - x)\| \leq \|df_z\| \|y - x\|
   \]
   for \( z = (1-t^*)x + t^* y \). Here \( df_z \) is the differential of \( f \) at the point \( z \).
2. If \( F = \mathbb{R} \), then there exists a \( t^* \) in \( (0, 1) \) such that
   \[
   f(y) - f(x) = df_z(y - x)
   \]
   for \( z = (1-t^*)x + t^* y \).

In the following, we will use \( \mathbb{S}^n \) to denote the space of all \( n \times n \) real symmetric matrices. Furthermore, for every positive real number \( c > 0 \), we define \( \mathcal{K}_c = \{B \in \mathbb{S}^n | B \succeq cI\} \), i.e., \( \mathcal{K}_c \) is the set of all symmetric matrices whose least eigenvalue is greater than or equal to \( c \). It is clear that \( \mathcal{K}_c \subset \mathbb{S}^n \cap V^{n \times n} \) is a convex set. Now we are in the position to state and prove the main theorem of this section.

**Lemma 10.** (Lipschitz continuity of \( F \)) The matrix inverse function \( F \) defined by \( F(A) = A^{-1}, \forall A \in V^{n \times n} \) is Lipschitz continuous on \( \mathcal{K}_c \) with Lipschitz constant \( \frac{1}{c} \) for every \( c > 0 \).
Proof. Let $c > 0$. Let $A, B \in \mathcal{K}_c$. Then the closed line segment joining $A$ and $B$ are contained in $\mathcal{K}_c$ since $\mathcal{K}_c$ is convex. By Lemma 8, $F$ is differentiable on the closed line segment joining $A$ and $B$ and so is continuous on this closed line segment. Also note that $\mathbb{R}^{n\times n}$, being a finite dimensional vector space, is a Banach space. Now by lemma 9,

$$\|F(A) - F(B)\| \leq \|[DF(C)](A - B)\|$$

(3.1)

for some matrix $C$ lying on the open line segment joining $A$ and $B$, where $[DF(C)]$ is the differential of $F$ at $C$.

By lemma 8, we have that the following holds:

$$[DF(C)](A - B) = -C^{-1}(A - B)C^{-1}.$$  

(3.2)

Combining (3.2) with (3.1), we have that

$$\|F(A) - F(B)\| \leq \|-C^{-1}(A - B)C^{-1}\|,$$

$$\leq \|C^{-1}\|^2 \|A - B\|,$$

(3.3)

where (3.3) follows from the sub-multiplicity of $\| \cdot \|$. Since $C \in \mathcal{K}_c$, by Lemma 7, we know that $\|C\| \geq c$, implying that $\|C^{-1}\| \leq \frac{1}{c}$ and so

$$\|C^{-1}\|^2 \leq \frac{1}{c^2}.$$  

(3.4)

From (3.3) and (3.4), we immediately have the following hold:

$$\|F(A) - F(B)\| \leq \frac{1}{c^2} \|A - B\|,$$

which completes the proof of Lemma 10.

\[\square\]

3.2 Convergence of Simultaneous Scheme

In this section, we prove the convergence results of the simultaneous scheme. Let $f(x; \Sigma) = \frac{1}{2} x^T \Sigma x - \kappa \mu^T x$ be the objective function of the computational prob-
lem (Markowitz problem). Then the gradient of $f$ with respect to variable $x$ is $\nabla_x f(x; \Sigma) = \Sigma x - \kappa \mu$. Let $X := \{ x \in \mathbb{R}^n | 1^T x = 1, x \geq 0 \}$ so that $X$ is the feasible set of the computational problem. In the simultaneous scheme, we use the projected gradient method to solve the computational problem and use ADMM to solve the learning problems. Let $x^k$ denote the result of the $k^{th}$ computational step while $\Sigma^k$ denotes the result of $k^{th}$ learning step. Furthermore, we let $\Sigma^*$ to denote the true covariance matrix while $x^*$ denotes the optimal solution for computational problem, i.e., $x^* = \arg\min_{x \in X} f(x; \Sigma^*)$.

### 3.2.1 Analysis of Computational Problem

In this subsection, we do not consider any specific learning problem and just assume that there is a sequence of matrices $\{\Sigma^k\}$ generated by some learning process such that $\|\Sigma^k - \Sigma^*\| \to 0$ as $k \to \infty$. Furthermore, we assume that this convergence is linear so that

$$\|\Sigma^{k+1} - \Sigma^*\| \leq \eta \cdot \|\Sigma^k - \Sigma^*\|$$

for some $\eta \in (0, 1)$ and any $k \geq 0$. We will use the following lemma from [28] in this subsection.

**Lemma 11.** Let the following hold:

$$u_{k+1} \leq q_k u_k + \alpha_k, 0 \leq q_k < 1, \alpha_k \geq 0, \sum_{k=1}^{\infty} (1 - q_k) = \infty, \lim_{k \to \infty} \frac{\alpha_k}{1 - q_k} = 0. \text{ Then } \lim_{k \to \infty} u_k \leq 0. \text{ In particular, if } u_k \geq 0, \text{ then } u_k \to 0 \text{ as } k \to \infty.$$

The general convergence result of simultaneous scheme is given by the following theorem.

**Theorem 1.** Suppose there exists a sequence of matrices $\Sigma^k$ generated by some learning process. Assume that $\|\Sigma^k - \Sigma^*\| \to 0$ as $k \to \infty$ and that the true covariance estimator $\Sigma^*$ is bounded by $cI \preceq \Sigma^* \preceq dI$ for some $0 < c < d$. If the computational step $\gamma$ satisfies $0 < \gamma < \frac{2}{\eta}$, then $x^k \to x^*$ as $k \to \infty$. Furthermore, if $\|\Sigma^k - \Sigma^*\| \to 0$ linearly, then the simultaneous scheme converges at a sub-linear rate.

**Proof.** In the simultaneous scheme, by the projected gradient method:

$$x^{k+1} := \Pi_X (x^k - \gamma \nabla_x f(x^k; \Sigma^{k+1})). \quad (3.5)$$
Let $x^*$ denote the optimal solution of the learning problem. Then the following holds:

$$x^* = \Pi_X(x^* - \gamma \nabla_x f(x^*; \Sigma^*)).$$

(3.6)

By (3.5) and (3.6), we have

$$\|x^{k+1} - x^*\| = \|\Pi_X(x^k - \gamma \nabla_x f(x^k; \Sigma^{k+1})) - \Pi_X(x^* - \gamma \nabla_x f(x^*; \Sigma^*))\|,$$

$$\leq \|(x^k - x^*) - \gamma(\nabla_x f(x^k; \Sigma^{k+1}) - \nabla_x f(x^*; \Sigma^*))\|,$$

$$= \|(x^k - x^*) - \gamma(\Sigma^{k+1} x^k - \Sigma^* x^k)\|,$$

$$= \|(I - \gamma \Sigma^*)(x^k - x^*) - \gamma(\Sigma^{k+1} - \Sigma^*)x^k\|,$$

$$\leq \|I - \gamma \Sigma^*\| \|x^k - x^*\| + \|\Sigma^{k+1} - \Sigma^*\| \|x^k\|,$$

where the two inequalities follow by the non-expansivity of the Euclidean projection and the triangle inequality. Since $x^k \in X$, we have $\|x^k\| \leq 1$. It follows that

$$\|x^{k+1} - x^*\| \leq \|I - \gamma \Sigma^*\| \|x^k - x^*\| + \|\Sigma^{k+1} - \Sigma^*\| .$$

(3.7)

Define $u_k := \|x^k - x^*\|$, $q_k \equiv q := \|I - \gamma \Sigma^*\|$ and $\alpha_k := \gamma \|\Sigma^{k+1} - \Sigma^*\|$. Since $\|\Sigma^k - \Sigma^*\| \to 0$ by assumption, we know that $\alpha_k \to 0$ as $k \to \infty$. Since $q_k \equiv q$ is constant, by Lemma 10, it remains to check that $q = \|I - \gamma \Sigma^*\| < 1$. By Lemma 7, we know that $\|I - \gamma \Sigma^*\|$ is the largest absolute value of eigenvalues of $I - \gamma \Sigma^*$. Let $\lambda$ be any eigenvalue of $I - \gamma \Sigma^*$. We now show that $|\lambda| < 1$. Since $cI \preceq \Sigma^* \preceq dI$ by assumption, we have that $\gamma cI \preceq \gamma \Sigma^* \preceq \gamma dI$ and therefore

$$(1 - \gamma c)I \preceq I - \gamma \Sigma^* \preceq (1 - \gamma d)I.$$  

Since $0 < \gamma < \frac{2}{d}$, it follows that $1 - \gamma d > -1$ and $1 - \gamma c < 1$, implying that $|\lambda| < 1$.

Since all the assumptions in Lemma 11 are satisfied, $u_k = \|x^k - x^*\| \to 0$ as $k \to \infty$, which means that simultaneous scheme generates a sequence $\{x^k\}$ with $x^k \to x^*$ as $k \to \infty$.

Now suppose $\alpha_k \to 0$ linearly as $k \to \infty$. To see $u_k \to 0$ at a sub-linear rate, recall that

$$u_{k+1} \leq q u_k + \gamma \alpha_k.$$  

(3.8)
Since $\alpha_k \to 0$ linearly as $k \to \infty$, we know that $\alpha_k \leq \eta \alpha_0$ for some $0 \leq \eta < 1$. By expanding (3.8), we obtain
\[ u_{k+1} \leq q^{k+1} u_0 + (k + 1) \gamma L^k \alpha_0, \] (3.9)
where $L := \max(\eta, q) < 1$. It follows that
\[ \|x^{k+1} - x^*\| \leq q^{k+1} \|x^0 - x^*\| + (k + 1) \gamma L^k \|\Sigma^1 - \Sigma^*\|. \] (3.10)

\[ \square \]

**Remark 1.** Note that in (3.10), there is a degradation in the convergence rate from the standard linear rate to a sub-linear rate because of the learning process. Also note that when we have an access to the true covariance estimator $\Sigma^*$, the linear rate is recovered.

### 3.2.2 Case 1: SCS Learning Problem

To use the above convergence theorem of simultaneous scheme, we first need to provide lower and upper bounds of $\Sigma^*$. First, we derive lower and upper bounds for $\Sigma^*$. The lower bound is available directly from the constraint $\Sigma \succeq \epsilon I$, and therefore $\Sigma^* \succeq \epsilon I$. For an upper bound, we need the following lemma:

**Lemma 12.** For a symmetric $n \times n$ matrix $A \in \mathbb{S}^n$, we have that the following holds:
\[ \|A\| \leq \|A\|_F \leq \sqrt{n} \|A\|, \]
where $\| \cdot \|$ denotes matrix 2-norm.

Now we may now provide an upper bound on $\Sigma^*$:

**Lemma 13.** If $\Sigma^* = \arg\min_{\Sigma \succeq \epsilon I} \left( \|\Sigma - S\|_F^2 / 2 + \lambda |\Sigma|_1 \right)$ is the optimal solution of SCS learning problem, then $\Sigma^* \preceq \left( \sqrt{n}\epsilon + (\sqrt{n} + 1) \|S\| \right) I$.

**Proof.** It suffices to show that $\|\Sigma^*\| \leq \sqrt{n}\epsilon + (\sqrt{n} + 1) \|S\|$. By the triangle inequality, we have that
\[ \|\Sigma^*\| \leq \|\Sigma^* - S\| + \|S\|. \] (3.11)
By Lemma 12, we know that
\[
\|\Sigma^* - S\| \leq \|\Sigma^* - S\|_F. \tag{3.12}
\]
Since \(\epsilon I\) is a feasible solution of SCS problem, \(\Sigma^*\) is the optimal solution of SCS problem and \(|:\cdot:\|_1\) is the off-diagonal norm, we have that
\[
\|\Sigma^* - S\|^2_F/2 \leq \|\Sigma^* - S\|^2_F/2 + \lambda |\Sigma^*|_1 \leq \|\epsilon I - S\|^2_F/2 + \lambda |\epsilon I|_1 = \|\epsilon I - S\|^2_F/2. \tag{3.13}
\]
It then follows that
\[
\|\Sigma^* - S\|_F \leq \|\epsilon I - S\|_F. \tag{3.14}
\]
Again by Lemma 12,
\[
\|\epsilon I - S\|_F \leq \sqrt{n}\|\epsilon I - S\| \leq \sqrt{n}(\|\epsilon I\| + \|S\|) = \sqrt{n}(\epsilon + \|S\|). \tag{3.15}
\]
Comparing (3.11), (3.12) and (3.15), the result follows.

From this lemma, we can see that simultaneous scheme indeed provides a convergent sequence if we use the SCS problem as our learning problem. Recall that there are two approaches available to solve the SCS problem. If we use the first one, we observe that \(\Psi_1\) is not continuously differentiable and therefore it is hard to estimate the convergence rate of ADMM. But in the second one, we see that \(\Psi_2\) is strongly convex and has a Lipschitz continuous gradient. In [24] it is shown that this is sufficient for ADMM to converge linearly. Thus we have the following result.

**Theorem 2.** If the computational step \(\gamma\) is chosen to be a constant with \(\gamma < \frac{2}{\sqrt{n} + (\sqrt{n} + 1)\|S\|}\), then the simultaneous scheme for Markowitz portfolio optimization with SCS learning problem produces a convergent sequence \(\{x^k\}\). Furthermore, if Algorithm 3 is used to solve the learning problem, then simultaneous scheme produces a sequence that converges sub-linearly.

**Proof.** Take \(c = \epsilon\) and \(d = \sqrt{n}\epsilon + (\sqrt{n} + 1)\|S\|\) in Theorem 1, then simultaneous scheme produce a convergent sequence \(\{x^k\}\). Moreover, if adopt Algorithm 3 to solve the SCS learning problem, then \(\|\Sigma^k - \Sigma^*\| \to 0\) as \(k \to \infty\) at a linear rate. Therefore, \(x^k \to x^*\) at a sub-linear rate.
3.2.3 Case 2: SPS Learning Problem

For SPS learning problem, we already have lower and upper bounds. The only question is that when \( k \rightarrow \infty \), \( k \leq k \leq k \geq 0 \) at a linear rate. When solving SPS learning problem, what we actually obtain is a sequence \( (k) \). Although in [30] it is shown that \( (k) \rightarrow 0 \) linearly, there is no guarantee that \( (k) \rightarrow 0 \) linearly. Fortunately, by the Lipschitz continuous property of matrix inverse function, we indeed have \( (k) \rightarrow 0 \) linearly. Let \( P^* := (\Sigma^*)^{-1} \) and \( P = (\Sigma^k)^{-1} \). Recall that in SPS learning problem, \( P^*, P \geq aI \) for some \( a > 0 \).

**Lemma 14.** If ADMM is adopted to solve SPS learning problem, then \( (k) \rightarrow 0 \) at a linear rate as \( k \rightarrow \infty \).

**Proof.** By Lemma 10, we have

\[
\|k - k\| = \|(P^k)^{-1} - (P^*)^{-1}\| \leq \frac{1}{a^2} \|P^k - P^*\|.
\]

Since \( \|P^k - P^*\| \rightarrow 0 \) at a linear rate, \( (k) \rightarrow 0 \) at a linear rate. \( \square \)

Now we have the following theorem.

**Theorem 3.** If the computational step \( \gamma \) is chosen to be a constant with \( \gamma < \frac{2}{\|S\|_2^2 + \alpha} \), then the simultaneous scheme for Markowitz portfolio optimization with SPS learning problem converges in a sub-linear rate.

**Proof.** Recall that in SPS learning problem, we have \( aI \leq (\Sigma^*)^{-1} \leq bI \), where \( a = \frac{1}{\|S\|_2^2 + \alpha} \) and \( b = \frac{n}{\alpha} \). Therefore, \( \frac{1}{b} I \leq \Sigma^* \leq \frac{1}{a} I \). Take \( c = \frac{1}{b} \) and \( d = \frac{1}{a} \) in Theorem 1, then \( \frac{2}{d} = \frac{2}{\|S\|_2^2 + \alpha} \), and it can be seen that simultaneous scheme provides a sequence \( x^k \rightarrow x^* \). Moreover, by Lemma 14, since \( (k) \rightarrow 0 \) at a linear rate, theorem 1 shows that \( x^k \rightarrow x^* \) at a sub-linear rate. \( \square \)
Chapter 4

Numerical examples

In this chapter, we use some preliminary numerical examples to illustrate the results discussed in Chapter 3. We implement the algorithms in Chapter 2 using MATLAB 2014b on a computer with a 2.5GHz Intel Core i7 CPU.

4.1 Data Generation and Problem Parameters

First we construct the true covariance matrix as $\Sigma = (\sigma_{ij})_{1 \leq i, j \leq n}$ based on the following rule: $\sigma_{ij} = \max(1 - |i-j|/10, 0)$ for $n = 500$ and 1000 [16]. Then we generate a sample of random vectors from a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix $\Sigma$. The sample size is set as $\frac{n}{2}$, and we use this sample to calculate the sample mean $\mu$ and sample covariance matrix $S$.

In the Markowitz portfolio selection problem, we set the problem parameter $\kappa = 1$, and in both the SCS and SPS learning problems, we set the penalty parameter $\rho = 1$. Finally, in the sequential scheme, the step length in the projected gradient method is set to be $\gamma = \frac{1}{\|\Sigma^*\|}$, where $\Sigma^*$ is the optimal solution of learning problems, while in the simultaneous scheme, $\gamma$ is set to be $\frac{2}{c+d}$, where $c$ and $d$ are the lower and upper bound of $\Sigma^*$ discussed in Theorem 2.

4.2 Covariance matrix of size $500 \times 500$

In this section, we provide numerical results for the Markowitz portfolio selection problem with a $500 \times 500$ covariance matrix.
4.2.1 Markowitz Portfolio Optimization with SCS Learning

For portfolio selection with SCS learning, in the sequential scheme, we set the number of learning steps to be 10, 20, 30, 40 and 50. The number of computational steps is set to be 6000. In the simultaneous scheme, we take 10, 20, 30, 40 and 50 to be the number of learning steps and in each iteration, the number of computational steps is set to be 600, 300, 200, 150 and 120, respectively, to ensure that the overall effort is identical.

The numerical result for the first approach is given in Table 4.1.

<table>
<thead>
<tr>
<th>Learning steps</th>
<th>Computational steps</th>
<th>$|x^k - x^*|$</th>
<th>CPU time</th>
<th>Iterations</th>
<th>Computational steps/iteration</th>
<th>$|x^k - x^*|$</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6000</td>
<td>2.40e-04</td>
<td>14.17s</td>
<td>10</td>
<td>600</td>
<td>2.53e-04</td>
<td>14.21s</td>
</tr>
<tr>
<td>20</td>
<td>6000</td>
<td>2.56e-06</td>
<td>16.98s</td>
<td>20</td>
<td>300</td>
<td>3.32e-06</td>
<td>16.61s</td>
</tr>
<tr>
<td>30</td>
<td>6000</td>
<td>7.36e-08</td>
<td>19.06s</td>
<td>30</td>
<td>200</td>
<td>1.23e-07</td>
<td>18.96s</td>
</tr>
<tr>
<td>40</td>
<td>6000</td>
<td>2.75e-09</td>
<td>21.44s</td>
<td>40</td>
<td>150</td>
<td>5.56e-09</td>
<td>21.51s</td>
</tr>
<tr>
<td>50</td>
<td>6000</td>
<td>1.01e-10</td>
<td>24.20s</td>
<td>50</td>
<td>120</td>
<td>6.60e-10</td>
<td>23.87s</td>
</tr>
</tbody>
</table>

Table 4.1. SCS, 1st approach, $500 \times 500$ covariance

From Table 4.1, we see that the results of simultaneous scheme and sequential scheme are comparable in terms of accuracy.

The numerical results for the second approach is given in Table 4.2.

<table>
<thead>
<tr>
<th>Learning steps</th>
<th>Computational steps</th>
<th>$|x^k - x^*|$</th>
<th>CPU time</th>
<th>Iterations</th>
<th>Computational steps/iteration</th>
<th>$|x^k - x^*|$</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>6000</td>
<td>4.38e-04</td>
<td>14.20s</td>
<td>10</td>
<td>600</td>
<td>4.64e-04</td>
<td>14.17s</td>
</tr>
<tr>
<td>20</td>
<td>6000</td>
<td>7.08e-06</td>
<td>16.60s</td>
<td>20</td>
<td>300</td>
<td>8.39e-06</td>
<td>16.65s</td>
</tr>
<tr>
<td>30</td>
<td>6000</td>
<td>3.39e-07</td>
<td>19.01s</td>
<td>30</td>
<td>200</td>
<td>4.71e-07</td>
<td>19.04s</td>
</tr>
<tr>
<td>40</td>
<td>6000</td>
<td>1.78e-08</td>
<td>21.52s</td>
<td>40</td>
<td>150</td>
<td>3.03e-08</td>
<td>21.49s</td>
</tr>
<tr>
<td>50</td>
<td>6000</td>
<td>9.43e-10</td>
<td>23.89s</td>
<td>50</td>
<td>120</td>
<td>1.91e-09</td>
<td>23.93s</td>
</tr>
</tbody>
</table>

Table 4.2. SCS, 2nd approach, $500 \times 500$ covariance

Compare Table 4.1 and Table 4.2, the second approach to solve SCS does not perform as well as the 1st approach, but again both schemes give comparable results.
4.2.2 Markowitz Portfolio Optimization with SPS Learning

In our numerical examples it turns out that solving SPS learning problem is harder than solving SCS learning problem and it takes more iterations to solve the SCS learning problem. The comparison between sequential scheme and simultaneous scheme is given in Table 4.3.

<table>
<thead>
<tr>
<th>Learning steps</th>
<th>Computational steps</th>
<th>$|x^k - x^*|$</th>
<th>CPU time</th>
<th>Iterations</th>
<th>Computational steps/iteration</th>
<th>$|x^k - x^*|$</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>6000</td>
<td>$1.07e-02$</td>
<td>18.89s</td>
<td>50</td>
<td>$1.07e-02$</td>
<td>19.06s</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>6000</td>
<td>$3.34e-03$</td>
<td>25.33s</td>
<td>100</td>
<td>$3.35e-03$</td>
<td>25.67s</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>6000</td>
<td>$1.13e-03$</td>
<td>32.03s</td>
<td>150</td>
<td>$1.14e-03$</td>
<td>32.86s</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>6000</td>
<td>$3.74e-04$</td>
<td>38.61s</td>
<td>200</td>
<td>$3.80e-04$</td>
<td>39.62s</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>6000</td>
<td>$1.02e-04$</td>
<td>45.36s</td>
<td>250</td>
<td>$1.05e-04$</td>
<td>46.72s</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3. SPS, 500 $\times$ 500 covariance

From table 4.3, it is still true that simultaneous scheme works as well as sequential scheme.

4.3 Covariance matrix of size 1000 $\times$ 1000

In this section, we demonstrate the numerical results for Markowitz portfolio selection with 1000 $\times$ 1000 covariance matrix.

4.3.1 Markowitz Portfolio Optimization with SCS Learning

It is more time consuming to solve the problem since the size of problem has been doubled. But it turns out that in all scenarios, the result of simultaneous scheme works as well as sequential scheme. For the first approach, see Table 4.4. For the second approach, see Table 4.5.

4.3.2 Markowitz Portfolio Optimization with SPS Learning

For Markowitz portfolio with SPS learning, the results are provided in Table 4.6.
From all these numerical results, we see that simultaneous scheme indeed provides convergent sequence $x^k \to x^*$, as proved in Chapter 3. Compare numerical results for $500 \times 500$ and $1000 \times 1000$ covariance matrices, we find that when the size of the covariance matrix becomes larger, the effort for learning and computation increases. Furthermore, it is more time consuming to solve the portfolio selection with SPS learning problem than with SCS learning problem. One of the reasons is that after each learning iteration, the resulting precision matrix need to be inverted to obtain corresponding covariance matrix. The numerical results presented in this chapter is only a preliminary result, which only shows that simultaneous scheme is an alternative approach to sequential scheme. Future numerical experiments need to be conducted to show that the theoretical rate of convergence, which is derived in equation (3.10), is valid.
Bibliography


[26] M. Fortin and R. Glowinski, Augmented Lagrangian methods: applications to

and variational problems in imperfect information regimes,” arXiv preprint

optimization and monotone variational inequality problems,” arXiv preprint

onto the 1 1-ball for learning in high dimensions,” in Proceedings of the 25th

selection for fitting gaussian random field models to large data sets,” arXiv

[31] J. Hubbard and B. B. Hubbard, Vector Calculus, Linear Algebra, and Differ-