DISSERTATION

EMPIRICAL EVALUATION OF A DIMENSION-REDUCTION METHOD FOR TIME-SERIES PREDICTION

Submitted by

Mahsa Ghorbani

Department of Systems Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2020

Doctoral Committee:

Advisor: Edwin K. P. Chong

Ali Pezeshki
Peter Young
Thomas Bradley

ABSTRACT

EMPIRICAL EVALUATION OF A DIMENSION-REDUCTION METHOD FOR TIME-SERIES PREDICTION

Stock price prediction is one of the most challenging problems in finance . The multivariate conditional mean is a point estimator to minimize the mean square error of prediction giver past data. However, the calculation of the condition mean and covariance involves the numerical inverse of a typically ill-conditioned matrix, leading to numerical issues. To overcome this problem, we develop a method based on filtering the data using principle components. Principal component analysis (PCA) identifies a small number of principle components that explain most of the variation in a data set. This method is often used for dimensionality reduction and analysis of the data. Our method bears some similarities with subspace filtering methods. Projecting the noisy observation onto a principle subspace leads to significantly better numerical conditioning. Our method accounts for time-varying covariance information.

We first introduce our method for predicting future price values over a short period of time using just historical price values. The literature provides strong evidence that stock price values can be predicted from past price data. Different economic variables have also been used in the literature to estimate stock-price values with high accuracy. To accommodate using historical data for such economic variables, we build on our method to include multiple predictors. We use multichannel cross-correlation coefficient as a measure for selecting the most correlated set of variables for each stock. Then we apply our filtering operation based on the local covariance of the data. Our method is easily implemented and can be configured to include an arbitrary number of predictors, subject to computational constraints.

Time-series prediction can be posed as a matrix completion problem. Matrix completion is an important problem in many fields and has been receiving considerable attention in recent years.

Different approaches and algorithms have been proposed to solve this problem. We investigate the effectiveness of an iterative rank minimizing matrix completion algorithm for predicting financial time series. As a key performance to compare different schemes, we use computational complexity, which focuses on the computational burden of these schemes. We compare the prediction results from the iterative matrix completion method to our method in terms of asymptotic and empirical computational complexity. Both methods show similar performance for forecasting future stock price values in terms of different performance metrics, but our proposed method has lower computational complexity.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep appreciation to my advisor, Prof. Edwin K. P. Chong for accepting me as a Ph. D student, for training me in doing research, writing papers, and doing presentations, for encouraging me to broaden my horizons by attending conferences, summer schools, workshops, and doing internships, and for all the financial and spiritual support during my graduate studies at Colorado State University. It has been a great honor and pleasure to study and conduct research under the supervision of Prof. Chong. He is not only my mentors in research, but also in life!

I would also like to express my sincere thanks to my committee members, Prof. Ali Pezeshki, Prof. Peter Young and Prof. Thomas Bradley for their important suggestions and remarks in my research and for their time and efforts in serving as my committee.

I would especially like to thank Dr. Yajing Liu and Dr. Savini Samarasinghe for their friendship and assistance on my research work. Thanks also go to my great friends Dr. Sabiheh Faghih and Dr. Katayoun Neshatpour for their encouragement in both research and life.

I acknowledge the people who mean a lot to me, my parents for working hard to support me to go to college and giving me liberty to choose what I desired and my sister, for her selfless love and care.

I owe many thanks to a very special person, my husband, Dr. Ali Tasdighi for his continued and unfailing love, support, and understanding during my pursuit of Ph.D degree that made the completion of this thesis possible.

DEDICATION

I would like to dedicate this dissertation to my family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Stock Price Prediction Using Multiple Predictors[1][2]

1.1 Introduction

Predicting future stock price values is a very challenging task. There is a big body of literature on different methods and different predictors to incorporate into those methods to predict the future values as closely as possible. The literature provides strong evidence that past price/return data can be used to predict future stock prices. Some studied have found significant auto-correlation for returns over a short period of time. French and Roll find negative correlation for individual securities for daily returns [2]. Some other studies show there is a positive correlation for returns over the period of weeks or months [3]. Studies also demonstrate stock return correlation over the period of multiple months or years. Fama and French report that the auto-correlation is stronger for longer periods, three to five years, compared to daily or weekly periods [4]. Cutler et al. report positive auto-correlation over the horizon of several months and negative auto-correlation over the horizon of three to five years [5]. There are some other studies that also show correlation in stock returns over a multiple year interval [6, 7] which all confirm that price/return values are predictable from past price/return values.

Bogousslavsky shows that trading by investors with heterogeneous rebalancing horizons can give rise to autocorrelation in the returns at different frequencies [8]. Chowdhury et al. investigate the autocorrelation structure of seven Gulf Cooperation Council (GCC) stock markets. All the markets except for Dubai and Kuwait show significant first-order autocorrelation of returns. They also find that autocorrelation between weekdays is usually larger than that between the first and last trading days of the week [9]. Li et al. study the nonlinear autoregressive dynamics of stock index returns in seven major advanced economies (G7) and China using the quantile autoregres-

[1]This chapter is published in *PLOS One* Journal [1]

[2]The chapters are written in a way that they are self-contained

sion model. For the stock markets in the seven developed economies, the autoregressive parameters generally follow a decreasing pattern across the quantiles with significant portions outside the ordinary least squares estimate intervals [10]. Another study investigates the autocorrelation structure of stock and portfolio returns in the unique market setting of Saudi Arabia [11]. Their results show that there is significantly positive autocorrelation in individual stock and market returns. Another study applies the threshold quantile autoregressive model to study stock return autocorrelations in the Chinese stock market [12]. They report negative autocorrelations in the lower regime and positive autocorrelations in the higher regime.

Other fundamental or macroeconomic factors can also be used in predicting future stock price values. Macroeconomic factors such as interest rates, expected inflation, and dividend can be used in stock return predictions models [4, 13]. Also fundamental variables such as earnings yield, cash flow yield, size and book to market equity [14, 15] have been found to have estimation power in predicting future price/return values.

Silvennoinen and Teräsvirta report correlation between individual U.S. stocks and the aggregate U.S. market [16]. Dennis et al. study the dynamic relation between daily stock returns and daily volatility innovations, and they report negative correlations [17]. Another study investigates the effect of common factors on the relationship among stocks and on the distribution of the investment weights for stocks [18]. They report that market plays a dominant role in both structuring the relationship among stocks and in constructing a well-diversified portfolio. Dimic et al. examine the impact of global financial market uncertainty and domestic macroeconomic factors on stock–bond correlation in emerging markets [19]. In another study, the focus is analyzing the impact of oil price shocks on the interactions of oil-stock prices [20]. The results show that negative changes in oil prices have a significant impact on the stock market.

In this chapter, we describe a general method for predicting future stock price values based on historical price data, using time-varying covariance information. When the number of observations is large compared to the number of predictors, the maximum-likelihood covariance estimate [21] or even the empirical covariance is a good estimate of the covariance of the data, but that is not always

the case. When the number of observations is smaller than the matrix dimension, the problem is even worse because the matrix is not positive definite [22]. This problem, which happens quite often in finance, gives rise to a new class of estimators such as shrinkage estimators. For example Ledoit and Wolf, shrink the sample covariance towards a scaled identity matrix using a shrinkage coefficient that minimizes the mean squared error of the prediction [23]. Some other studies in this field include [24, 25, 26]. In our numerical evaluations in this work we have sufficient empirical data to reliably track the covariance matrix over time.

Momentum-based forecasting relies on prices following a trend, either upwards or downwards. Based on the assumption that trends like this exist and can be exploited, momentum is used as a heuristic rule for forecasting and is probably the most popular technical indicator used by traders; in particular, the method of *Direction Movement Index (DMI)*, due to Wilder [27]. This kind of heuristic is a special case of pattern-based forecasting, where, in the case of momentum, the pattern is simply the upward or downward trend. Our method is a systematic method to capture arbitrary patterns, not just upward or downward trends. Indeed, we compute prevalent patterns in the form of eigenvectors (or "eigen-patterns") of the local covariance matrix. As such, we are able to exploit more general patterns that are prevalent (but not necessary known beforehand) in price time series.

The mean squared error (MSE) measures the distance between predicted and real values and is a very common metric to evaluate the performance of predictive methods [28]. Multivariate conditional mean minimizes the mean squared error [29] and is a good estimator for future price values. However, numerical results using this method cannot always be trusted because of associated ill-conditioning issues. In this work we introduce a method with similar estimation efficiency that does not suffer from this issue.

Principal component analysis (PCA), which is a method for dimensionality reduction of the data, is used in different fields such as statistical variables analysis [30], pattern recognition, feature extraction, data compression, and visualization of high dimensional data [31]. It also has various application in exploring financial time series [32], dynamic trading strategies [33], financial risk

computations [34, 33], and statistical arbitrage [35]. In this work, we implement PCA in estimating future stock price values.

Yu et al. introduce a machine-learning method to construct a stock-selection model, which can perform nonlinear classification of stocks. They use PCA to extract the low-dimensional and efficient information [36]. In another study, three mature dimensionality reduction techniques, PCA, fuzzy robust principal component analysis, and kernel-based PCA, are applied to the whole data set to simplify and rearrange the original data structure [37]. Wang et al. present a stochastic function based on PCA developed for financial time-series prediction [38]. In another study, PCA is applied to three subgroups of stocks of the Down Jones Industrial (DJI) index to optimize portfolios [39]. Narayan et al. apply PCA to test for predictability of excess stock returns for 18 emerging markets using a range of macroeconomic and institutional factors [40].

Factor analysis is a technique to describe the variability of observed data through a few factors and is in some sense similar to PCA. There is a long debate in the literature on which method is superior [41, 42]. Factor analysis begins with the assumption that the data comes from a specific model where underlying factors satisfy certain assumptions [43]. If the initial model formulation is not done properly, then the method will not perform well. PCA on the other hand involves no assumption on the form of the covariance matrix. In this work, we focus on developing an algorithm that can ultimately be used in different fields without prior knowledge of the system, and therefore PCA is the method of choice. In the case study presented in the following section, although only price data is used, it would have been also possible to include multiple predictors to estimate futures values of stock prices.

Our method bears some similarity with subspace filtering methods. Such methods assume a low-rank model for the data [44]. The noisy data is decomposed onto a signal subspace and noise based on a modified singular value decomposition (SVD) of data matrices [45]. The orthogonal decomposition can be done by an SVD of the noisy observation matrix or equivalently by an eigenvalue decomposition of the noisy signal covariance matrix [44].

We compare the performance of our proposed methods in terms of MSE and directional change statistic. Stock-price direction prediction is an important issue in the financial world. Even small improvements in predictive performance can be very profitable [46]. Directional change statistic calculates whether our method can predict the correct *direction* of change in price values [47]. It is an important evaluation measure of the performance because predicting the direction of price movement is very important in some market strategies.

Another important parameter that we are interested in is standard deviation, one of the key fundamental risk measures in portfolio management [48]. The standard deviation is a statistical measure of volatility, often used by investors to measure the risk of a stock or portfolio.

As mentioned above, in this work we focus on forecasting stock prices from daily historical price data. In Section 3.2, we introduce our technical methodology, and in particular estimation techniques using covariance information. In Section 1.3, we describe our method for processing the data and estimating the time-varying covariance matrix from empirical data, including data normalization. We also demonstrate the performance of our method.

1.2 Theoretical Methodology

1.2.1 Estimation Techniques

In this section we introduce a new computationally appealing method for estimating future stock price values using covariance information. The empirical covariance can be used as an estimate of the covariance matrix if enough empirical data is available, or we can use techniques similar to the ones introduced in the previous section, though the time-varying nature of the covariance must be addressed.

Suppose that we are given the stock price values for M days. Our goal is to predict company stock prices for $M + 1$ to N trading days, using the observed values of the past consecutive M days. The reason for introducing N will be clear below.

Gauss-Bayes or Conditional Estimation of z given y

Suppose that x is a random vector of length N. Let $M \leq N$ and suppose that the first M data points of vector x represent the end-of-day prices of a company stock over the past M consecutive trading days. The multivariate random vector x and can be partitioned in the form

$$x = \begin{bmatrix} y & z \end{bmatrix}. \tag{1.1}$$

Let random vector y represent the first M data points and z the price of the next $N - M$ days in the future. We wish to estimate z from y.

The covariance matrix for the random vector x can be written as

$$\Sigma_{xx} = \begin{bmatrix} \Sigma_{yy} & \Sigma_{yz} \\ \Sigma_{zy} & \Sigma_{zz} \end{bmatrix}, \tag{1.2}$$

where Σ_{yy} is the covariance of y and Σ_{zz} is the covariance of z. Assuming that y and z are jointly normally distributed, knowing the prior distribution of $x = [y, z]$, the Bayesian posterior distribution of z given y is given by

$$\begin{aligned} \widehat{z}_{z|y} &= \Sigma_{zy}\Sigma_{yy}^{-1}y \\ \widehat{\Sigma}_{z|y} &= \Sigma_{zz} - \Sigma_{zy}\Sigma_{yy}^{-1}\Sigma_{yz}. \end{aligned} \tag{1.3}$$

The $\widehat{\Sigma}_{z|y}$ matrix, representing the conditional covariance of z given y, is also called the Schur complement of Σ_{yy} in Σ_{xx}. Note that the posterior covariance does not depend on the specific realization of y.

The Gauss-Bayes point estimator for the price prediction, the conditional mean $\widehat{z}_{z|y}$, minimizes the mean squared error of the estimate in the Gaussian case [29]. Moreover,in the Guassian case, for a specific observation y, the inverse of the conditional covariance is the Fisher Information

matrix associated with estimating z from y, and therefore $\widehat{\Sigma}_{z|y}$ is the lower bound on the error covariance matrix for any unbiased estimator of z [29].

The same set of equations arise in Kalman's filtering. Kalman's own view of this process is as a completely deterministic operation [49], and does not rely on assuming normality. Although the point estimator $\widehat{z}_{z|y}$ is optimal in term of mean squared error, in practice there are numerical complications involved in this method: The matrix Σ_{yy} is typically not well conditioned, so the numerical calculation of Σ_{yy}^{-1} cannot always be trusted. To overcome this problem, we propose a better conditioned estimator, which has a behavior close to Gauss-Bayes.

Principal Components and Estimation in Lower Dimension

Principal component analysis (PCA) is a well-established mathematical procedure for dimensionality reduction of data and has wide applications across various fields. In this work, we consider its application in forecasting stock prices.

Consider the singular value decomposition (SVD) of Σ_{xx}:

$$\Sigma_{xx} = VSV', \tag{1.4}$$

where S is a diagonal matrix of the same dimension as x with non-negative diagonal elements in decreasing order, and V is a unitary matrix ($VV^{'} = I_N$). The diagonal elements of S are the eigenvalues of Σ_{xx}.

In general, the first few eigenvalues account for the bulk of the sum of all the eigenvalues. The "large" eigenvalues are called the principal eigenvalues. The corresponding eigenvectors are called the principal components.

Let $L < N$ be such that the first L eigenvalues in S account for the bulk part (say 85% or more) of the sum of the eigenvalues. Let V_L be the first L columns of unitary matrix V. Then the random vector x is approximately equal to the linear combination of the first L columns of V:

$$x \approx V_L \alpha, \tag{1.5}$$

where α is a random vector of length L. Because L is a small number compared to N, equation (1.5) suggests that a less “noisy" subspace with a lower dimension than N can represent most of the information. Projecting onto this principle subspace can resolve the ill-conditioned problem of Σ_{yy}. The idea is that instead of including all eigenvalues in representing Σ_{xx}, which vary greatly in magnitude, we use a subset which only includes the “large” ones, and therefore the range of eigenvalues is significantly reduced. The same concept is implemented in speed signal subspace filtering methods, which are based on the orthogonal decomposition of noisy speech observation space onto a signal subspace and a noise subspace [44]. Let $V_{M,L}$ be the first M rows and first L columns of V. We have

$$y = V_{M,L}\alpha + \text{Noise}. \tag{1.6}$$

Mathematically resolving noisy observation vector y onto the principle subspace can be written as a filtering operation in the form of

$$w = Gy, \tag{1.7}$$

where G is given by

$$G = (V_{M,L}^{'} V_{M,L})^{-1} V_{M,L}^{'}. \tag{1.8}$$

The vector w is actually the coordinates of the orthogonal projection of y onto the subspace equal to the range of $V_{M,L}$. We can also think of w as an estimate of α based on least squares. Substituting y by w in (1.3) leads to a better conditioned set of equations:

$$\widehat{z}_{z|w} = \Sigma_{zw}\Sigma_{ww}^{-1}w$$
$$\widehat{\Sigma}_{z|w} = \Sigma_{zz} - \Sigma_{zw}\Sigma_{ww}^{-1}\Sigma_{wz}, \tag{1.9}$$

because the condition number of Σ_{ww} is much lower than that of Σ_{yy}, as we will demonstrate later. In (3.13) we have

$$\Sigma_{zw} = E\left[zw^{'}\right] = \Sigma_{zy}G^{'}, \tag{1.10}$$

and

$$\Sigma_{ww} = E\left[ww^{'}\right] = G\Sigma_{yy}G^{'}. \tag{1.11}$$

If the posterior distribution of z estimated based on (3.13) has a similar behavior to the distribution estimated by (1.3), it can be considered a good substitute for the Gauss-Bayes method. Our numerical results demonstrate that this is indeed the case, which we will show in Section 1.3.

Moving Average

Technical traders and investors often use technical trading rules, and one of the most popular methods used by technical traders and researchers are the moving average (MA) rules [50, 51]. Satchell investigates the reason general MA trading rules are widely used by technical analysts [52]. He shows that autocorrelation amplification is one of the reasons such trading rules are popular. Using simulated results, we show that the MA rule may be popular because it can identify the price momentum and is a simple way of assessing and exploiting the price autocorrelation without necessarily knowing its precise structure. Moving average, which is the average of prices over a period of time, is probably the simplest estimator for z:

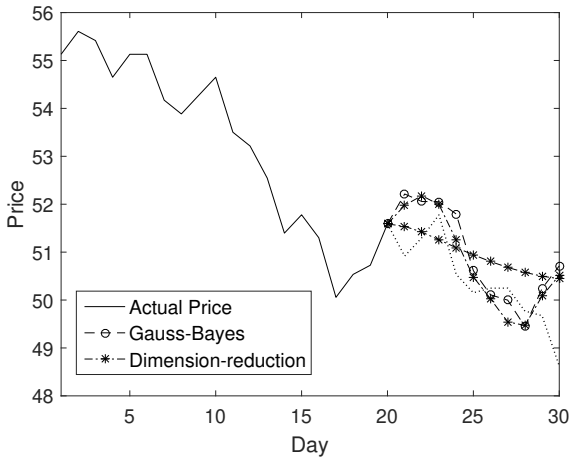

Figure 1.1: Predicting price for $M+1$ to N days, Actual Price: Solid line, GB: $-o-$, RD: $-*-$ (two lines, one for a small value of L, and one for a relatively large number).

$$\widehat{z}_{\text{MA}} = \frac{1}{K_{MA}} \sum_{i=N-K_{MA}+1}^{N} x_i \tag{1.12}$$

where the quantity K_{MA} is the number of data points included to calculate the average, and $\hat{z}_{MA}$ is the average of the most recent K_{MA} price values.

There are different possible values of K_{MA} for calculating the average, from short to medium to long term periods. Here we use periods of 10 and 50 days, which are typical short and midterm values used in the literature. We will use the moving average estimator for comparison purposes, as we will see in Section 1.3 below.

Fig 1.1 shows an example of our stock predictions. Assume that we are given the price values for the past 20 days ($M = 20$), and we want to use those values to predict the future prices over the next 10 business days, from day $M+1$ to day N ($N = 30$). In our reduced-dimension technique, we can get a relatively smooth plot of the predicted value for a relatively small L, to a plot almost the same as Gauss-Bayes, for larger values of L, as we can can see in Fig 1.1.

1.2.2 Performance Metrics

Mean Squared Error

To compare the performance of the methods described above, we evaluate the expected value of the squared error between the actual and estimated values. The mean squared error of an estimate $\hat{z}$ is given by:

$$MSE = E\left[\|z - \hat{z}\|^2\right] = E\left[\|z\|^2\right] + E\left[\|\hat{z}\|^2\right] - 2E\left[\|z'\hat{z}\|\right].$$

The MSE can be expressed in terms of the covariance matrices in (1.2), by substituting the appropriate form of $\hat{z}$. Alternatively, the mean squared error of an estimator $\hat{z}$ can be written in terms of the variance of the estimator plus its squared bias. The conditional MSE given x is written as

$$MSE_{\hat{z}|z} = E\left[\|z - \hat{z}\|^2 \mid z\right] = trace(\Sigma_{\hat{z}|z}) + \left\|E\left[\hat{z} \mid z\right] - z\right\|^2.$$

The first term is called the variance, and the second term is the squared bias. The expected value of MSE over all observations is the actual MSE, which can be calculated by taking expectations on both sides:

$$MSE = E\left[E\left[\|z - \hat{z}\|^2 \mid z\right]\right] = trace(E\left\|\Sigma_{\hat{z}|z}\right\|) + E\left[\left\|E\left[\hat{z} \mid z\right] - z\right\|^2\right]. \tag{1.13}$$

It turns out that Gauss-Bayes estimator is unbiased, which means that the second term is 0, while the proposed reduced-dimension methods is a biased estimator.

Directional Change Statistic

$$b_{ij} = \begin{cases} 1, & \text{if } (z_{ij} - z_0)(\hat{z}_{ij} - z_0) > 0 \\ 0, & \text{otherwise} \end{cases}. \tag{1.14}$$

Then D_j, the directional change statistic for day j, averaged over K samples, is equal to

$$D_j = \frac{1}{K} \sum_{i=1}^{K} b_{ij}, \tag{1.15}$$

which is a number between 0 and 1 (the higher the better).

1.3 Empirical Methodology and Results

In this section we describe how we estimate the covariance matrix based on a normalized data set, and we evaluate the performance of our method using empirical data.

1.3.1 General Setting

Suppose that we have K samples of vector data, each of length N, where $N < K$. Call these row vectors $x_1, x_2, ..., x_K$, where each $x_i \in \mathbb{R}^N (i = 1, ..., K)$ is a row vector of length N:

$$x_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iN} \end{bmatrix}. \tag{1.16}$$

We assume that the vectors $x_1, x_2, \ldots, x_K$ are drawn from the same underlying distribution. We can stack these vectors together as rows of a $K \times N$ matrix:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ x_{K1} & x_{K2} & \cdots & x_{KN} \end{bmatrix}.$$

Let $M \leq N$ and suppose that we are given a vector $y \in \mathbb{R}^M$ representing the first M data points of a vector we believe is drawn from the same distribution as $x_1, x_2, \ldots, x_K$. Again, these

M data points represent the end-of-day prices of a company stock over the past M consecutive trading days. Let z be the price of the next $N - M$ days in the future. We wish to estimate z from y.

Since the vector x_i is a multivariate random vector that can be partitioned in the form

$$x_i = \begin{bmatrix} y_i & z_i \end{bmatrix}, \tag{1.17}$$

where y_i has length M and z_i has length $N - M$, accordingly the data matrix X can be divided into two sub-matrices Y and Z as follow:

$$X = \begin{bmatrix} Y & Z \end{bmatrix}.$$

We can think of Y as a data matrix consisting of samples of historical data, and Z as a data matrix consisting of the corresponding future values of prices.

1.3.2 Normalizing and Centering the Data

In the case of stock-price data, the vectors $x_1, x_2, \ldots, x_K$ might come from prices spanning several months or more. If so, the basic assumption that they are drawn from the same distribution may not hold because the value of a US dollar has changed over time, as a result of inflation. To overcome this issue, a scaling approach should be used to meaningfully normalize the prices (we will deal with the time-varying nature of the covariance later). One such approach is presented here. Suppose that $t_i = [t_{i1}, t_{i2}, \ldots, t_{iN}]$ is a vector of "raw" (unprocessed) stock prices over N consecutive trading days. Suppose that $Q \leq N$ is also given. Then we apply the following normalization to obtain x_i:

$$x_i = \frac{t_i}{t_i(Q)}. \tag{1.18}$$

This normalization has the interpretation that the x_i vector contains stock prices as a fraction of the value on the Qth day, and is meaningful if we believe that the pattern of such fractions over the days $1, \ldots, N$ are drawn from the same distribution. Note that $x_i(Q) = 1$.

We believe normalizing the data with this method captures the pattern in the price data better than simply using return data. Although similar to return, the resulting time series still suffers from being non-stationary over time. We propose to resolve this issue by using a weighting averaging method as explained in the next section.

For the purpose of applying our method based on PCA, we assume that the vectors $x_1, x_2, \ldots, x_K$ are drawn from the same underlying distribution and that the mean, $\bar{x}$, is equal to zero. However because x_i represents price values, in general the mean is not zero. The mean $\bar{x}$ can be estimated by averaging the vector $x_i \in \mathbb{R}^N (i = 1, ..., K)$,

$$\bar{x} = \frac{1}{K} \sum_{i=1}^{K} x_i, \tag{1.19}$$

and then this average vector is deducted from each x_i to center the data.

Even though this normalization makes the data stationary in the mean, since stock prices are very volatile, there is no guarantee that the covariance of the data would be stationary as well. In order to address this issue, we assign exponential weights $(\gamma^0, \gamma^1, \cdots, \gamma^k)$ to observations, where $0 < \gamma < 1$, to emphasize the most recent periods of data. Using an exponential weighting approach to deal with volatility of financial data has been suggested in multiple studies such as [53]. For each observation x_i, the last K samples prior to that observation are transformed into a Hankel matrix and normalized. Then (decreasing) exponential weights are assigned to the K samples and numerical results are calculated. This process, creating the matrix of data, normalizing, and assigning weights, is repeated for each observation.

To select the value of K we use

$$K = \min\{k : \gamma^k < 10^{-3}\}. \tag{1.20}$$

1.4 Experiments

The daily historical price data for 150 different companies from different market-capitalization categories were downloaded from finance.yahoo.com. Market capitalization is a measure of the company's wealth and refers to the total value of all a company's shares of stock. We randomly select 50 stocks from each of the three market capitalization (cap for short) categories: Big market-cap (125 B$ to 922 B$), Mid market-cap (2 B$ to 10 B$) and Small market-cap (300 M$ to 1.2 B$). The stocks from the Big market-cap category are normally the most stable ones relative to the Small-cap stocks, which have the most volatility. Historical data for four market indexes, S&P500 (GSPC), Dow Jones Industrial Average (DJI), NASDAQ Composite (IXIC), and Russell 2000 (RUT), were also included in this study. The data was transformed into matrices with different sizes as explained in next section. In each case, the daily price value for next 10 days are predicted and the estimation methods are compare based on their out-of-sample performance.

1.4.1 Constructing Data Matrix

The daily stock price data is transformed into a matrix with K rows, samples of vector data, each of length N. We get that by stacking K rows (K samples), each one time shifted from the previous one, all in one big matrix, called the Hankel matrix.

More precisely, the Hankel matrix for this problem is constructed in the following format:

$$\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_K \end{bmatrix} = \begin{bmatrix} P(1) & P(2) & \cdots & P(N) \\ P(2) & P(3) & \cdots & P(N+1) \\ \cdots & \cdots & \cdots & \cdots \\ P(K) & P(K+1) & \cdots & P(K+N-1) \end{bmatrix},$$

where $P(i)$ represents the price for day i. This is our matrix of data, before normalization and centering.

We first normalize each row (observation) by Qth entry, as described earlier, and then subtract the average vector $\bar{x}$ from each row. The prediction is done using the processed data. After doing the prediction, we add back the average vector $\bar{x}_{N-M}$ (last $N-M$ components of $\bar{x}$) from days $M+1$ through N and also multiply the result by the value of Qth that was used for normalizing to get back to actual stock prices. We tested different values for Q in terms of MSE and estimation variance. For the purpose of this study, we chose $Q = M$ because it shows the best results in this setting. Recall that $x_i(M) = 1$. This column is removed from the data matrix because it does not provide any information. From now on matrix X represents normalized and centered price data.

To account for the nonstationarity of the covariance, we use an exponential averaging method as mentioned before. For this purpose, $\gamma = 0.98$ was selected and the weights smaller than 10^{-3} were considered zero. Then the sample covariance matrix is calculated as

$$\Sigma_{xx} = \left(\frac{1-\gamma}{1-\gamma^{k+1}}\right) X' \text{diag}(\gamma^0, \gamma^1, \cdots, \gamma^k) X, \tag{1.21}$$

where $\text{diag}(\gamma^0, \gamma^1, \cdots, \gamma^k)$ is a diagonal matrix with $(\gamma^0, \gamma^1, \cdots, \gamma^k)$ as the diagonal elements.

We obtained end-of-day stock prices for General Electric and converted this time series into Hankel matrices with different lengths as described above. 2000 samples were used to evaluate the out-of-sample performance of the methods. The values corresponding with the performance metrics presented in this section converge after a few hundred samples. We construct data matrices with 9 different sizes, M from 50 to 530 with a 60 day interval, to investigate the effect of length of observation vector on performance.

Fig 1.2 shows the histogram of normalized data as a representation of the distribution of normalized data; the curve resembles a bell shape.

1.4.2 MSE Performance

Three different estimation methods are implemented for each of the data matrices constructed above. The goal is to predict future price values for the next 10 days (days $M+1$ to N). when it comes to reduced-dimension method, for each M we try different values of L, the number of

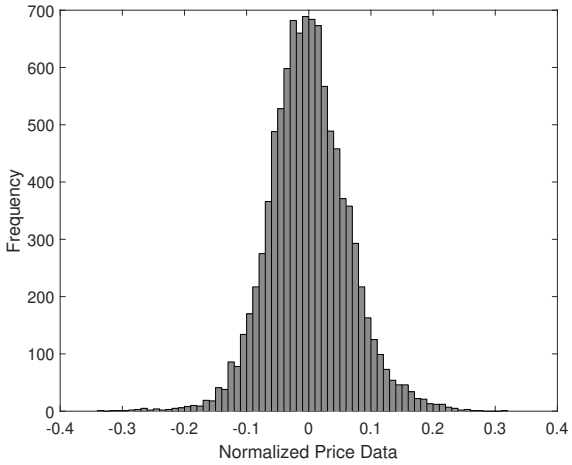

Figure 1.2: Histogram graph for normalized data

principle components. The general goal, as mentioned above, is an estimation technique that has a similar behavior as an ideal Gauss-Bayes estimator but does not have the associated calculation difficulties resulting from ill-conditioning.

We use General Electric price data to calculate the values illustrated in this section. We calculate the squared error (SE) for 2000 samples to evaluate the performance of the methods. We implement our reduced-dimension technique for different Ms, and for different numbers of principal eigenvalues, L.

Fig 1.3 shows the empirical Cumulative distribution function (CDF) of the SE for 2 different values of M, together with two-standard-deviation confidence interval. Note that to make our comparisons fair and meaningful, we normalized the results from the moving average predictors so that their values are equally normalized with the values from our RD method. When it comes to out-of-sample performance, the numerical complications compromise the estimation accuracy of Gauss-Bayes, causing the SE values for this method to become even worse than the SE plot for the moving average estimators. As we can see, in both plots, our reduced-dimension method is superior to the other two methods. For $M = 110$ some lines are relatively close together. As M gets larger, the plot for the reduced-dimension method improves and the plot for Gauss-Bayes gets worse.

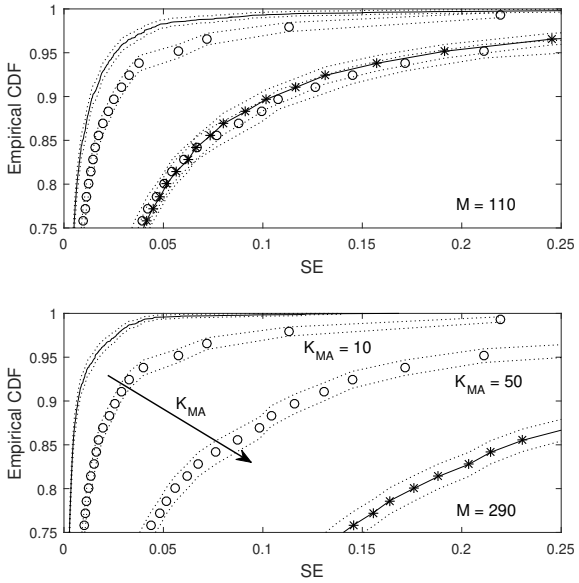

Figure 1.3: Empirical CDF of SE corresponding to $M = 110$ and 290. MA_{20} and MA_{50}: $-o-$, GB: $-*-$, RD: Solid lines. Dashed lines illustrate a two standard deviation confidence interval.
Plots toward the top and left represent better performance.

Another point worth mentioning is that although adding more data improves the performance of our proposed method, that is not the case for the moving average estimator. As the arrow on the plot on the bottom indicates, by adding more data, moving from $z_{MA_{10}}$ to $z_{MA_{50}}$, the performance of the moving average estimator deteriorates. This behavior is expected since the moving average relies on the momentum, in contrast to the reduced-dimension method, which extracts the essence of the information by projecting onto a smaller subspace.

Fig 1.4 shows the values of MSE over all days of estimation versus the value of L, for 9 different M, lengths of observation vector, from 50 to 530. As we can see, the MSE value is insensitive to the value of L for sufficiently large L. For small values of L, the MSE values fall quickly, but then eventually increase. So if we have a particular constraint on the condition number, we do not lose much in terms of MSE by choosing a reduced-dimension subspace, which leads to a better conditioned problem. After a certain point, adding more data is actually adding noise and the MSE values get worse.

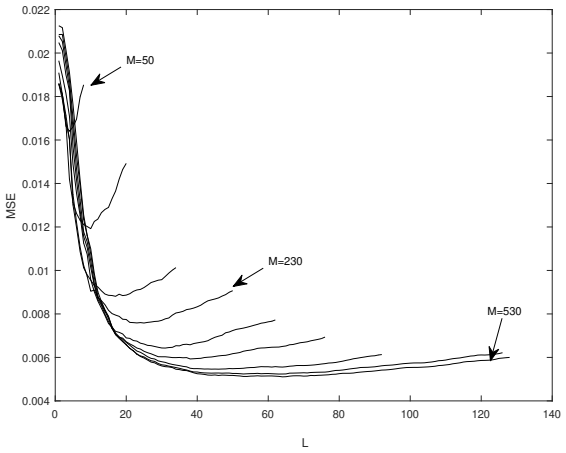

Figure 1.4: MSE versus L in the normalized domain for different Ms

The metric we are looking for is the sum of MSE values over all days of estimation.

For each length of M, the values for MSE are captured based on different constraints of the condition number of Σ_{ww}. The MSE values in the reduced-dimension method are significantly smaller relative to the other two methods.

Fig 1.5 shows the relative percentage of improvement (RPI) in the reduced-dimension method compared to the other two methods, calculated as

$$RPI_{\text{GB/MA}} = \frac{-100(MSE_{\text{RD}} - MSE_{\text{GB/MA}})}{MSE_{\text{GB/MA}}}. \tag{1.22}$$

Note that since the denominator in the equation is $MSE_{\text{GB/MA}}$, the improvement percentage does not exceed 100% but the actual MSE values are further apart in absolute terms than illustrated here. For example for $M = 350$, the MSE value for reduced-dimension is between 0.0052 to 0.018, while the MSE in Gauss-Bayes is around 6.33×10^6. The three (overlapping and therefore appears as only a single plot) lines on top (-*-) of Fig 1.5 compare the reduced-dimension to Gauss-Bayes (RPI_{GB}). The three lines on top (..o..) correspond to the comparison of the reduced-dimension and moving average ($RPI_{MA_{50}}$) and the three lines on the bottom (..o..) correspond to ($RPI_{MA_{10}}$). In each case the three lines are subject to different upper limits on the condition number (10^2, 10^3, and 10^4). It is worth mentioning that the condition number of Σ_{yy} starts from 10^3 for $M = 50$ and

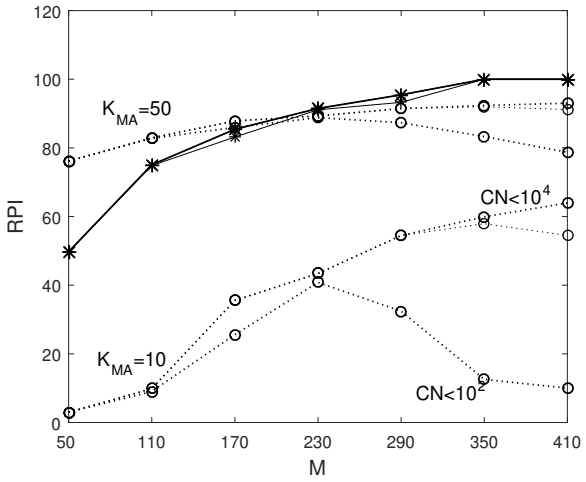

Figure 1.5: RPI values, subject to different upper limit on condition number of Σ_{ww}, in each case 10^2 associated with the line on the bottom, to 10^4 associated with the line on top,
RPI_{MA}: ..o.., RPI_{GB}: $-*-$.
Higher plots represent worse relative performance (relative to RD).

goes up to 10^{19} for $M = 530$. The upper limit on the condition number of Σ_{ww} changes from 10^2, associated with the lines on the bottom in each case, to 10^4, the lines on top, for all values of M.

In general, by increasing M, more information is available in each observation, resulting in better performance of the prediction in terms of smallest MSE values. This can be observed easily in the RPI plots in Fig 1.5 in comparison to the moving average cases since the MSE values in the those cases are almost constant for different values of M. The percent of improvement of MSE values corresponding to the reduced-dimension method increases as M increases. This is as expected since more information is available in each observation, resulting in better performance. However after a certain point the RPI flattens out suggesting adding more data at this point is increasing the noise and does not improve the performance.

As we can see, in some cases there is a slight decrease in the improvement rate of the reduced-dimension method compared to the moving average method. A possible explanation for this observation is that when we fix some constraint on condition number, we are actually limiting the value of L, and by increasing M, after a certain point, we mostly increase the noise, and the MSE value gets worse, which is consistent with Fig 1.4. Table 1.1 shows the average RPI values

Table 1.1: Average RPI values for stocks in different market-cap categories and average RPI values for market indexes ($M = 350$)

MSE	RPI_{GB}	$RPI_{MA_{10}}$	$RPI_{MA_{50}}$
Small-Cap	100%	51%	88%
Mid-Cap	100%	54%	88%
Big-Cap	100%	56%	89%
Market indexes	100%	55%	88%

Table 1.2: Statistical analysis for MSE values for stocks in different market cap categories ($M = 350$)

T-test	against	MSE_{GB}	$MSE_{MA_{10}}$	$MSE_{MA_{50}}$
Small-Cap	p-value	0.0024	0.0075	0.00068
	h	1	1	1
Mid-Cap	p-value	0.0283	0.0066	0.000038
	h	1	1	1
Big-Cap	p-value	0.0021	0.00048	0.00001
	h	1	1	1

for all stocks in different market-cap categories and average RPI values for market indexes. The reduced-dimension method consistently shows better performance than the other two methods.

Matlab's two-sample t-test function was used to determine the MSE values from our proposed method for 50 stocks in each market-cap category is significantly smaller than the average of the MSE values generated for the same sample using other methods at 5% significance level ($\alpha = 0.05$). When $p < \alpha$ and $h = 1$, the null hypothesis that the two samples have the same mean is rejected, concluding that the difference between the averages of the two sets of samples is statistically significant at α significance level. As shown in Table 1.2, the results indicate that the average of the MSE values for predictions from our method is significantly smaller than the average of MSE values from other competing methods at 0.05 significance level.

Recall that L represents the number of eigenvalues required from the diagonal matrix S to represent the bulk part of the information carried in x. Fig 1.6 investigates the dimension of the target subspace by plotting the value of L corresponding to best MSE for different Ms, subject to different limits on condition number (the same case as in Fig 1.5).

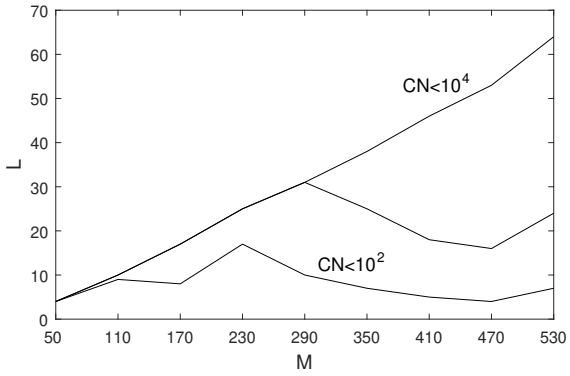

Figure 1.6: Best L corresponding to best MSE values subject to different limits on condition number, 10^2 associated with the line on the bottom, 10^3 associated with the line in the middle, and 10^4 associated with the line on top.

As the upper limit on condition number increases, the value of MSE improves as M increases, and we need a bigger subspace, bigger L, to extract the information. However, as the bottom three plots in Fig 1.6 show, the value for best L flattens out after a certain point.

1.4.3 Directional Change Statistic Performance

The other evaluation metric that we are interested in is the directional statistic which measures the matching of the actual and predicted values in terms of directional change. Figure 1.7 shows the average directional statistic over 10 days of estimation using the same $K = 2000$ samples. As the plot indicates, the reduced-dimension method is superior in terms of directional change statistic. It is interesting to note that the directional statistic improves as M increases, and then eventually flattens out, consistent with previous plots.

Table 1.3 shows the average value for directional statistic for stocks in different market cap categories and indexes for $M = 350$ for Σ_{ww} condition number limited to 10^4. The reduced-dimension method is superior to the other two methods in terms of directional change estimation. It is important to note that the values represented in Table 1.3 are associated with a specific M for all companies. In practice, it is recommended to tailor the value of M for each company to get the best results.

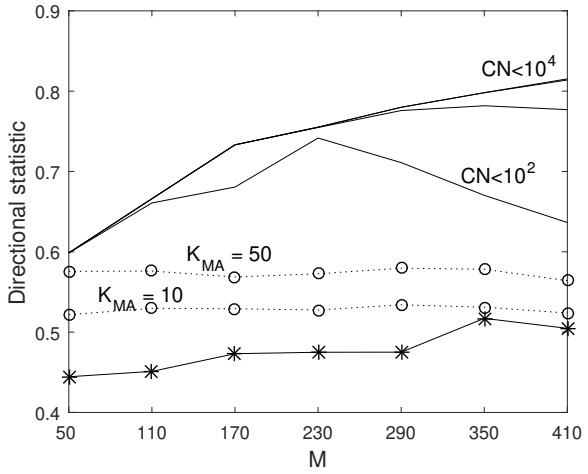

Figure 1.7: Best Directional Statistics subject to different upper limit on condition number of Σ_{ww}, 10^2 associated with the line on the bottom, to 10^4 associated with the line on top, GB: $-*-$, RD: Solid lines, MA: *..o..* (MA_{10} on the bottom and MA_{50} on top).
Higher plots represent better performance.

Table 1.3: Average directional statistics for stocks in different market cap categories ($M = 350$)

Directional Statistic	MA_{10}	MA_{50}	GB	RD
Small-Cap	0.51	0.53	0.51	0.78
Mid-Cap	0.53	0.55	0.51	0.79
Big-Cap	0.54	0.57	0.51	0.80
Market Indexes	0.55	0.60	0.50	0.79

Table 1.4: Statistical analysis for Directional Statistics values for stocks in different market-cap categories ($M = 350$)

T-test	against	D_{GB}	$D_{MA_{10}}$	$D_{MA_{50}}$
Small-Cap	p-value	$< 10^{-10}$	$< 10^{-10}$	$< 10^{-10}$
	h	1	1	1
Mid-Cap	p-value	$< 10^{-10}$	$< 10^{-10}$	$< 10^{-10}$
	h	1	1	1
Big-Cap	p-value	$< 10^{-10}$	$< 10^{-10}$	$< 10^{-10}$
	h	1	1	1

Matlab's two-sample t-test function was used to determine if the average of the directional statistics from our method for 50 stocks is significantly larger than the average of directional statistics from other methods. Table 1.4 lists the p-value and h-statistic for each test. The results also indicate that the average of directional statistics from our method is significantly larger than the average of the directional statistics from other competing methods at 5% significance level.

1.4.4 Volatility

Another important parameter that we estimate is the volatility of the prediction, measured in terms of its standard deviation. The square root of the diagonal elements of the estimated covariance, $\widehat{\Sigma}_{zz}$, are the estimated standard deviations for individual days of estimation. The estimate of the covariance in each method is

$$\begin{aligned} \widehat{\Sigma}_{GB} &= \widehat{\Sigma}_{z|y} = \Sigma_{zz} - \Sigma_{zy}\Sigma_{yy}^{-1}\Sigma_{yz}, \\ \widehat{\Sigma}_{RD} &= \widehat{\Sigma}_{z|w} = \Sigma_{zz} - \Sigma_{zw}\Sigma_{ww}^{-1}\Sigma_{wz}, \end{aligned} \tag{1.23}$$

However, note that because of the poor conditioning of Σ_{yy}, using the formula above for Σ_{GB} has numerical issues. Hence, we omit their values here. In general the standard deviation values increase moving from day 1 to day 10 of prediction, since less uncertainty is involved in the estimation of stock prices of days closer to the current day. In Fig 1.8, the standard deviation for individual days of estimation, days 1 to 10, are plotted versus M, the length of observation vec-

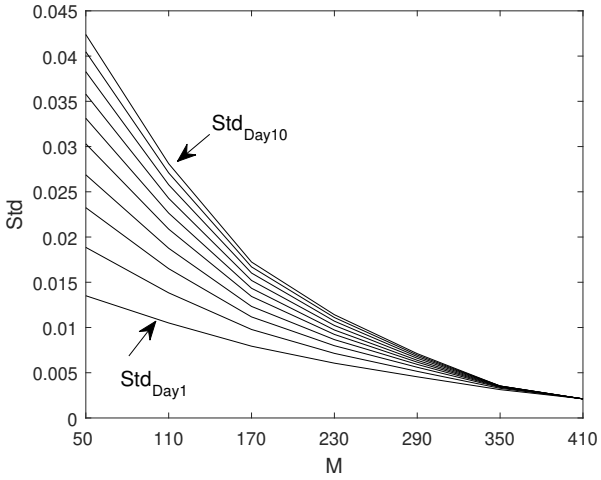

Figure 1.8: Standard deviation of individual days of estimation, RD: Solid line.

tor, for the reduced-dimension method. In the reduced-dimension method, the standard deviation values decrease as M increases because more information is provided in each observation. For sufficiently large Ms, the standard deviation values for different days are very close.

1.5 Conclusion

In this chapter we introduced a new method for predicting future stock price values based on covariance information. We develop this method based on a filtering operation using principle components to overcome the numerical complications of conditional mean. We also introduced a procedure for normalizing the data. The matrix of data was constructed in different sizes to investigate the effect of length of observation vector on prediction performance. Our method has showed consistently better out-of-sample performance than Gauss-Bayes (multivariate conditional mean), a numerically challenged estimator, and moving average, an easy to use estimator, for 150 different companies in terms of mean squared error and directional change statistic.

The proposed method can be modified to include multiple predictors. The significance of the proposed approach will be even more apparent when using multiple predictors because where observation vectors are longer it becomes almost impossible to rely on conditional mean due to the severe ill-conditioning of the covariance matrix.

1.6 Data Availability Statement

The data that support the findings of this study are openly available in Yahoo Finance at https://finance.yahoo.com/.

GE historical prices, General Electric Company, common stock, was retrieved $2016 - 10 - 06$ from https://finance.yahoo.com/quote/GE/history/.

Chapter 2

Stock Price Prediction Using Multiple Predictors[3][4]

2.1 Introduction

One of the most difficult tasks in finance and economics is predicting future stock prices. This has been the focus of an extensive literature for decades. There is substantial evidence that stock returns are predictable, either by past price data [4, 5, 7] or other economic variables.

Ang and Bekaert [54] focus on forecasting excess returns using dividend yields, and find that stock returns are predictable in a short horizon. Campbell and Thompson [55] argue that given some weak restrictions, some variables that reflect the market valuation ratios, interest rate, patterns in corporate finance, consumption to wealth ratio, etc., have better performance than historical average mean when it comes to forecasting future price values. A study by Rozeff [56] shows that dividend yields are directly related to stock returns and compared to historical returns they provide better performance in terms of bias, mean squared error, and mean absolute error. Hodrick [57] shows that changes in dividend yields forecast significant persistent changes in expected stock returns. Fama and French [58] show that the expected excess returns (returns net of the one-month Treasury bill rate) on corporate bonds and stocks move together. Campbell [59] argues that interest rates predict stock returns. Also he believes the risk premium on long-term treasury bonds is highly correlated with those on stocks. Lettau and Ludvigson [60] show that fluctuations in the consumption-wealth ratio is a good estimator of stock returns. They argue that this predictor is even better than dividend yields and some other popular predictors over a short or intermediate horizon. Lamont [61] argues that the dividend payout ratio forecasts excess return of stocks as well as those of corporate bonds. In this study we focus on predicting future stock price values from past price data in conjunction with different fundamental and macroeconomic variables. In

[3]This chapter is under review for publication

[4]The chapters are written in a way that they are self-contained

this work we only use data that is easily accessible and can be downloaded for free. Richer data sets exist and might well lead to better performance, but they are not accessible for free.

In order to choose the best *predictors*, by which we mean the variables used as a basis for prediction, we compute their multichannel correlation coefficient, which is a method for selecting reliable "channels" and was originally proposed for speaker localization problems [62]. Here we use this technique for choosing the most correlated set of predictors to predict the future price values. After choosing the predictors, we feed the data to our proposed Reduced-Dimension method based on *principle component analysis* and *subspace filtering* as explained in Section 2.2.4.

Although the potential predictors in this study were mostly chosen from the literature, our proposed framework does not work based on classic economic interpretation. Our algorithm takes a big data set, chooses the best predictors automatically, and eventually extracts the relevant information in the data to predict the future values of prices.

The performance of the proposed framework is first evaluated based on mean squared error, which is one of the most established metrics to measure the accuracy of prediction. It measures the magnitude of the squared error without taking into account the direction. In order to take into account the direction of the prediction as well, we use another metric called directional change statistic. This metric is very important in terms of measuring the quality of the forecast (correctness of gradient predictions) [47] .

In this work, we propose a method to predict price values based on multiple predictors. First we introduce our method in Section 2.2. Then in Section 2.3, we describe our data processing framework and we present the numerical results. We conclude the chapter in Section 2.4.

2.2 Theoretical Methodology

2.2.1 General Setting

We start by describing the general setting for the prediction techniques introduced in this work. Much of what follows in this subsection and the next 2.2.2 come from the previous chapter where a simpler version of this setting is described. To avoid excessive overlap, we have abbreviated the

description here to a bare minimum, instead referring the reader to the previous chapter for further details.

Suppose that we have K samples of vector data, each of length N, where $N < K$. Call these row vectors $x_1, x_2, ..., x_K$, where each $x_i \in \mathbb{R}^N (i = 1, ..., K)$ is a row vector of length N:

$$x_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iN} \end{bmatrix}. \tag{2.1}$$

The multivariate vector x_i can be partitioned in the form

$$x_i = \begin{bmatrix} y_i & z_i \end{bmatrix}, = \begin{bmatrix} y_i^1 & y_i^2 & y_i^3 & \cdots & y_i^J & z_i \end{bmatrix}, \tag{2.2}$$

where each y_i^j represents a vector with length M_j and vector z_i has length $N - \sum_{j=1}^{J} M_j$ where $\sum_{j=1}^{J} M_j \leq N$ (recall the definition of N earlier). Index j in the variable y_i^j represents a predictor, so the vector y_i^j itself represents the ith data vector from predictor j. The goal is to use the information from historical data in the y_i^js to predict corresponding future values of stock prices, z_i. By stacking K samples, we have a $K \times N$ *data* matrix:

$$X = \begin{bmatrix} y_1^1 & y_1^2 & \cdots & y_1^J & z_1 \\ y_2^1 & y_2^2 & \cdots & y_2^J & z_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_K^1 & y_K^2 & \cdots & y_K^J & z_K \end{bmatrix}.$$

2.2.2 Normalizing and Centering the Data

In the case of historical data, the vectors $x_1, x_2, \ldots, x_K$ come from data spanning over time so a scaling approach is needed to normalize the data over time. It is important to note that the normalizing approach depends on the type of data. For example rates, like interest rate, are already scaled, and therefore there is no need to normalize such data, while price values have a wider range over time and need to be scaled.

One normalizing approach is presented here. Suppose that $t_i^j = [t_i^j(1), t_i^j(2), \ldots, t_i^j(M_j)]$ is a vector of "raw" (unprocessed) historical data associated with the jth predictor. Suppose that $Q_j \leq M^j$ is also given. Then to normalize the data we have:

$$y_i^j = \frac{t_i^j}{t_i^j(Q_j)}, \tag{2.3}$$

where $t_i^j(Q_j)$ is the Q_jth value in vector t_i^j.

In this way of normalization, the y_i^j value is a fraction of the Qth value, and is meaningful if we believe that the pattern of such fractions over the periods $1, \ldots, M_j$ are drawn from the same distribution for each predictor. Note that $y_i^j(Q_j) = 1$. This normalization method is applied to every predictor that requires scaling. Then these blocks of data are placed together to construct the normalized matrix of data, X.

To apply our method we need to "center" the data so that it has has zero mean. The mean $\bar{x}$ is estimated by averaging vectors, and then the average vector is subtracted from each observation x_i to get a zero mean data set.

Even though this normalization makes the data stationary in the mean, since financial data are very volatile (heteroscedastic), there is no guarantee that the covariance of the data would be stationary as well. To address this issue, we assign exponential weights $(\gamma^1, \gamma^2, \cdots, \gamma^n)$ to observations, where $0 < \gamma < 1$, to emphasize the most recent periods of data. Using an exponential weighting approach to deal with volatility of financial data has been suggested in multiple studies such as [53]. Then the sample covariance matrix is calculated as

$$\Sigma_{xx} = \left(\frac{1-\gamma}{1-\gamma^{k+1}}\right) X^T diag(\gamma^0, \gamma^1, \cdots, \gamma^k) X, \tag{2.4}$$

where $diag(\gamma^0, \gamma^1, \cdots, \gamma^k)$ is a diagonal matrix with $(\gamma^0, \gamma^1, \cdots, \gamma^k)$ as the diagonal elements.

2.2.3 Selecting Predictors

We compute the multichannel correlation coefficient (MCCC) as a measure for selecting the predictors. The MCCC represents correlation among more than two channels (predictors) and is a generalization of the more familiar and commonly used concept of correlation coefficient. The MCCC metric calculates a number associated with a set of predictor values and future stock prices, Z. So for each collection of data sets of predictors and Z, we can calculate the associated MCCC to see how informative they are.

To describe how we calculate the MCCC values, assume we want to form a collection called C including future price values Z and some selected predictors. Initially we consider one predictor at a time and future price values Z and calculate the associated MCCC metric based on the following equation:

$$\zeta_c^2 = 1 - \frac{det[\Sigma_{CC}]}{\prod_{i=1}^{K} \sigma_i^2}, \tag{2.5}$$

where Σ_{CC} is the covariance matrix of data in C and the σ_i^2s are the diagonal values in Σ_{CC}. The predictor associated with the highest value of the MCCC metric is identified and permanently added to the collection. Collection C now includes two time series: the selected predictor and Z. In the next step, we add the not-yet-selected predictors to the collection one at a time, and each time calculate the MCCC metric. Again, the predictor with the highest metric value will be permanently added to C. We continue this search procedure until we have sufficient number of predictors identified. In this study we identify the top 10 predictors. The historical data of these predictors constitute the matrix of data, which is then used to estimate the covariance matrix as described in (3.5).

2.2.4 Prediction Techniques

Multivariate conditional mean is known to minimize the mean squared error of the prediction assuming normal data. However in practice, in some cases numerical complications overpower its predictive accuracy especially when it comes to big data sets. In order to overcome this problem

we adopt our method from the previous chapter, which exploits the properties of the mutlivariate conditional mean but has dramatically better numerical conditioning. This method involves principle component analysis and subspace filtering, described next.

Principle component analysis (PCA) is a mathematical procedure that transforms a high-dimensional data set into a lower-dimensional set while keeping the variation in the data. PCA is used in various fields in finance. Ince and Trafalis use kernel PCA to identify the input predictors for their heuristic model to predict future stock prices [32]. Hotelling use PCA in determining hedge fund strategies [30]. Other applications include [34] in market risk analysis and [35] in Arbitrage Pricing Theory.

Subspace filtering involves the orthogonal decomposition into signal and noise subspaces based on the singular value decomposition of the noisy signal observation [44, 45]. In our previous chapter, we proposed a filtering operation similar to subspace filtering, which uses the principle components to extract the information out of a data set and reduces the dimension of the problem using a single predictor. The multivariate version of this method for big data sets involving multiple predictors is developed in this work and results are compared to the single predictor case as follows.

Suppose that we are given the historical data of J predictors, consisting of vectors of length M_j ($j = 1, \ldots, J$). Using these values, we want to predict the company stock prices z. Recall that vector z has the length equal to $\Theta = N - \sum_{j=1}^{J} M_j$, which represents the stock values over the next Θ consecutive trading days.

Gauss-Bayes Method

Recall that the multivariate vector x_i, with length N, can be partitioned in the form

$$x_i = \begin{bmatrix} y_i & z_i \end{bmatrix}, = \begin{bmatrix} y_i^1 & y_i^2 & y_i^3 & \cdots & y_i^J & z_i \end{bmatrix}, \tag{2.6}$$

where each y_i^j represents a vector with length M_j and vector z_i has length $N - \sum_{j=1}^{J} M_j$ where $\sum_{j=1}^{J} M_j \leq N$. Each y_i^j represents historical values of predictor j over the past consecutive trading days, months, or quarters (depending on the specific predictor).

The covariance matrix for the random vector x can be written as

$$\Sigma_{xx} = \begin{bmatrix} \Sigma_{yy} & \Sigma_{yz} \\ \Sigma_{zy} & \Sigma_{zz} \end{bmatrix}, \tag{2.7}$$

where Σ_{yy} is the covariance of y and Σ_{zz} is the covariance of z. Knowing the prior distribution of $x = [y, z]$, assuming Gaussian, the Bayesian posterior distribution of z given y is given by

$$\begin{aligned} \widehat{z}_{z|y} &= \Sigma_{zy}\Sigma_{yy}^{-1}y \\ \widehat{\Sigma}_{z|y} &= \Sigma_{zz} - \Sigma_{zy}\Sigma_{yy}^{-1}\Sigma_{yz}. \end{aligned} \tag{2.8}$$

The conditional covariance of z given y, $\widehat{\Sigma}_{z|y}$, is also called the Schur complement of Σ_{yy} in Σ_{xx}. The conditional mean [29], $\widehat{z}_{z|y}$ is known to minimize the mean squared error of the estimate compared to other unbiased estimators, but in practice the numerical complications, especially the high condition number in this setting, deteriorates the predictive power of this method. Our proposed method overcomes this problem.

Reduced-Dimension Method

Consider the singular value decomposition (SVD) of Σ_{xx}:

$$\Sigma_{xx} = VSV', \tag{2.9}$$

where S is a diagonal matrix of the same dimension as x with non-negative diagonal elements in decreasing order, and V is a unitary matrix.

The "large" eigenvalues (diagonal elements of S) are called the principal eigenvalues and in general the first few eigenvalues account for the bulk of the sum of all the eigenvalues. The corresponding eigenvectors are the called the principal components.

Let $L < N$ be such that the first L eigenvalues in S account for the bulk part of the sum of the eigenvalues. Let V_L be the first L columns of unitary matrix V, and $V_{M,L}$ be the fist M rows of

V_L. Our filtering operation which is mathematically resolving the noisy observation vector y onto the subspace spanned by the principle eigenvectors (henceforth called the *target subspace*) can be written as

$$w = Gy, \tag{2.10}$$

where G is

$$G = (V_{M,L}^{'} V_{M,L})^{-1} V_{M,L}^{'}. \tag{2.11}$$

Substituting y by w in (3.8) leads to:

$$\begin{aligned} \widehat{z}_{z|w} &= \Sigma_{zw} \Sigma_{ww}^{-1} w \\ \widehat{\Sigma}_{z|w} &= \Sigma_{zz} - \Sigma_{zw} \Sigma_{ww}^{-1} \Sigma_{wz}. \end{aligned} \tag{2.12}$$

As we will demonstrate later, this set of equations has a similar behavior to (3.8). At the same time, since Σ_{ww} has a much lower condition number than Σ_{yy}, the prediction results are more numerically trustworthy.

On top of these two methods, we also demonstrate the results from the *Moving Average* method, which is a common method often used by technical traders and investors [51]. The Moving Average prediction of z is:

$$\widehat{z}_{\text{MA}} = \frac{1}{K_{MA}} \sum_{i=N-K_{MA}+1}^{N} x_i \tag{2.13}$$

where $\hat{z}_{MA}$ is the average of prices over a short period of time or in other words the most recent K_{MA} price values. We will demonstrate the results in Section 2.3.

2.2.5 Performance Metrics

Two metrics are introduced here to measure the performance of the methods: mean squared error and directional change statistic.

Mean Squared Error

The mean squared error (MSE) measures the average squared difference between the actual and predicted values. The mean squared error of prediction $\hat{z}$ is calculated using

$$\text{MSE} = \frac{1}{K}\sum_{i=1}^{K} \|z_i - \hat{z}_i\|^2, \tag{2.14}$$

where k is the number of samples used in estimating the MSE, $\hat{z}_i$ is the prediction of z_i and $\|.\|$ is the Euclidean norm.

Directional Change Statistic

This evaluation metric measures the directional change between the actual and the predicted values. For $\theta = 1, 2, ..., \Theta$ we have:

$$b_{i\theta} = \begin{cases} 1, & \text{if } (z_{i\theta} - z_{\text{today}})(\hat{z}_{i\theta} - z_{\text{today}}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.15}$$

where $\hat{z}_i = [\hat{z}_{i(1)}, \hat{z}_{i(2)}, \cdots, \hat{z}_{i\theta}]$ is the price prediction for the next $\Theta = N - \sum_{j=1}^{J} M_j$ days, z_{today} is today's stock price, and $z_{i\theta}$ is the true price value at day θ in the future. Then D_θ, the direction statistic for day θ, is equal to

$$D_\theta = \frac{1}{K}\sum_{i=1}^{K} b_{i\theta}, \tag{2.16}$$

which is a number between 0 and 1.

2.3 Empirical Methodology and Results

The data we use in this study was downloaded from finance.yahoo.com and Kenneth R. French Data Library [63] and is available to all researchers for free. The shortest scope of data available to us was daily such as stock price values. We also have monthly data such as Industrial Production Index and Personal Consumption Expenditures, and finally quarterly data such as Total Assets of various companies (see below) and Earnings Before Tax.

The prediction results for stock price prediction of five companies from different industries are presented in this study: Exxon Mobil Corporation (XOM), Wal-Mart Stores, Inc. (WMT), Intel Corporation (INTC), and Caterpillar Inc. (CAT). Price data for SPDR S&P500 ETF (SPY), which is a fund following the S&P500 index, was also included. Each data set, including the price data, is transformed into a Hankel matrix and then centered and normalized as described earlier (see below). We first use this matrix of data to compute the MCCC and rank the predictors. Then we construct the data matrix for doing the prediction by putting together the blocks of data of selected predictors.

2.3.1 Constructing Data Matrix

End-of-day stock prices for Intel Corporation and values of other daily, weekly, and quarterly predictors were downloaded and converted into Hankel matrices as described below. The historical data for each predictor is transformed into a Hankel matrix with K samples (rows), each of length M_j, as shown here:

$$\begin{bmatrix} t_1^j \\ t_2^j \\ \vdots \\ t_K^j \end{bmatrix} = \begin{bmatrix} t_1^j(1) & t_1^j(2) & \cdots & t_1^j(M_j) \\ t_2^j(2) & t_2^j(3) & \cdots & t_2^j(M_j+1) \\ \cdots & \cdots & \cdots & \cdots \\ t_k^j(K) & t_k^j(K+1) & \cdots & t_k^j(K+M_j-1) \end{bmatrix},$$

where $t_i^j(1)$ represents the first data point in sample i for predictor j. This is one block of the data matrix, before normalization and centering.

For predictors that require normalizing, we first normalize each row (observation) by the Qth entry as described earlier. Then we put all the blocks in one matrix and subtract the average vector from each row. We set $Q_j = M_j$ for all blocks. Note that $y_i^j(M_j) = 0$ for all rows $i = 1, 2, \ldots, K$. Therefore, we remove the entire M_jth column. Henceforth, the data matrix X represents normalized and centered price data. To address the volatility of the data, the exponential weighted average is implemented with $\gamma = 0.985$.

As discussed before, MCCC was used to rank the predictors. The top 10 predictors were added to the matrix of data one by one, to investigate the effect of adding multiple predictors on the performance metrics. We investigate the results associated with different values of M_j for the historical price block which we will henceforth call M_{price} to investigate the effect of the length of the price observation vector on the results. For all other blocks, $M_j = 20$ was chosen for the purpose of this experiment.

2.3.2 MSE Performance

We implement our proposed prediction method for each data set constructed above. The Gauss-Bayes and Moving Average results are also reported for comparison. We implement our Reduced-Dimension technique for various values of L, the number of principal eigenvalues. The plots illustrating the performance of the method are based on Intel Corporation price data. The results reported in this section are based on $K = 1000$ samples.

Fig. 2.1 shows the MSE values averaged over the next 10 days of prediction for the Reduced-Dimension (RD) and Moving Average (MA) methods. The values associated with Reduced-Dimension method are presented for both the single predictor (SP) case and the multiple predictors (MP) case. Moving Average values are calculated over the most recent 10 days of observation. The horizontal axis shows the length of M_{price}. The plot on top, which is almost flat, belongs to the Moving Average method. The next plot corresponds to our Reduced-Dimension method with only historical prices as the predictor, and the next plots correspond to cases where three, five, seven, and nine predictors are included in the matrix of data (including historical price data).

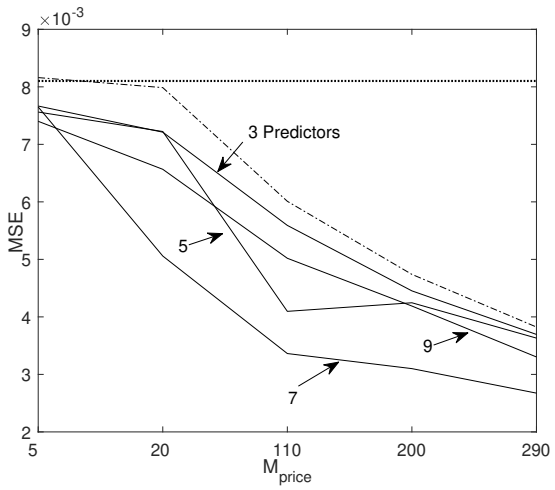

Figure 2.1: MSE values for different methods of prediction for different numbers of predictors. MA: ..., RD-SP: $-.$, RD-MP: Solid lines.

As we can see in Fig. 2.1, in general by increasing M_{price} the performance improves. Adding other predictors also gradually improves the performance of the Reduced-Dimension method as well up to a certain point.

The plots in Fig. 2.1 suggest that in this case the best number of predictors is approximately seven. By adding more predictors, the performance starts deteriorating. At this point adding more predictors is actually adding noise and the MSE values get worse. MSE values corresponding to Gauss-Bayes are not included in Fig. 2.1 because those values are much bigger than to other methods.

To better investigate the effect of adding predictors, Fig. 2.2 shows the *relative percentage of improvement* (RPI) in MSE, based on (2.17) below for the case with only historical price values as predictor (MSE_{SP}), compared to the case with multiple predictors (MSE_{MP}):

$$\text{RPI}_{\text{MP/SP}} = \frac{-100(\text{MSE}_{\text{MP}} - \text{MSE}_{\text{SP}})}{\text{MSE}_{\text{MP}}}. \tag{2.17}$$

So, the value of $\text{RPI}_{\text{MP/SP}}$ is the relative increase in the MSE when going from MP to SP (relative to MP and expressed as a percentage).

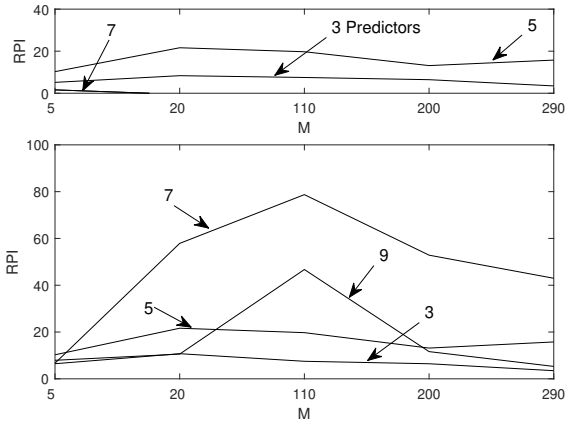

Figure 2.2: RPI for Reduced-Dimension method by adding predictors subject to two upper limits on condition number of the covariance matrix. Top: Condition number $< 10^4$ and Bottom: Condition number $< 10^{11}$.

Recall that since Σ_{ww} has a much lower condition number than Σ_{yy}, the prediction results of Reduced-Dimension method are more numerically trustworthy than Gauss-Bayes. To demonstrate the effect of the condition number of Σ_{ww} on the performance of the method, the two individual figures (sets of axes) depicted in Fig. 2.2 correspond to two different upper limits on condition number of the covariance matrix Σ_{ww}. The top figure, which corresponds to an upper limit of 10^4, shows that using five predictors is better than using three but adding more predictors does not improve the performance. In this setting, the target subspace (with dimension L) is small and the small subspace dimension limits the amount of information that can be exploited by adding more predictors. After some point, adding more predictors does not add more exploitable information but does add more noise. By increasing the upper limit on condition number, the target subspace becomes bigger. As a result, incorporating more predictors provides more information and the performance improves. This is well illustrated in the bottom figure in Fig. 2.2, which shows that using seven predictors gives the best performance relative to all the graphs shown there.

Once we increase the upper limit on the condition number, the MSE values improve up to 85%, which is for the case with seven predictors. Beyond this point, no additional information can be

Table 2.1: Improvement in MSE values in Reduced-Dimension method compared to other methods for different companies for condition number limited to 10^4, $M\text{price} = 210$.

MSE	$\text{RIP}_{\text{MP/GB}}$	$\text{RIP}_{\text{MP/MA}}$	$\text{RIP}_{\text{SP/GB}}$	$\text{RIP}_{\text{SP/MA}}$
INTC	>100%	61%	>100%	42%
XOM	>100%	60%	>100%	30%
CAT	>100.%	64%	>100%	33%
WMT	>100.%	67%	>100%	35%
SPY	>100%	63%	>100%	37%

Table 2.2: Best number of predictors based on RIP in Reduced-Dimension method for different companies for condition number limited to 10^4, $M\text{price} = 210$.

Stock	Best Number of Predictors
INTC	7
XOM	5
CAT	6
WMT	6
SPY	7

exploited from adding more data. Doing so effectively only adds noise, resulting in a decrease in the predictive power.

Table 2.1 shows the percentage of improvement in MSE values for five different companies for our Reduced-Dimension method compared to Moving Average ($\text{RPI}_{\text{MP/MA}}$, $\text{RPI}_{\text{SP/MA}}$) and Gauss-Bayes ($\text{RPI}_{\text{MP/GB}}$, $\text{RPI}_{\text{SP/GB}}$) methods. As we can see, the Reduced-Dimension method with multiple predictors consistently shows better performance. We investigated the effect of including one to ten predictors in each case. Table 2.2 shows the best number of predictors for predicting the prices of six companies. The values do not not differ by much, confined to be between 5 and 7.

2.3.3 Target Subspace Dimension

Recall that L represents the number of eigenvalues required from the diagonal matrix S to represent the bulk of the information carried in X. It is also the dimension of the target subspace. Fig. 2.3 plots the value of L corresponding to the best MSE for different numbers of predictors, subject to two different upper limits on the condition number, 10^4 in the top diagram and 10^{11} in the bottom diagram (the same as in Fig. 2.2). In other words, Fig. 2.3 investigates the dimension of the target corresponding to the best MSE for different numbers of predictors.

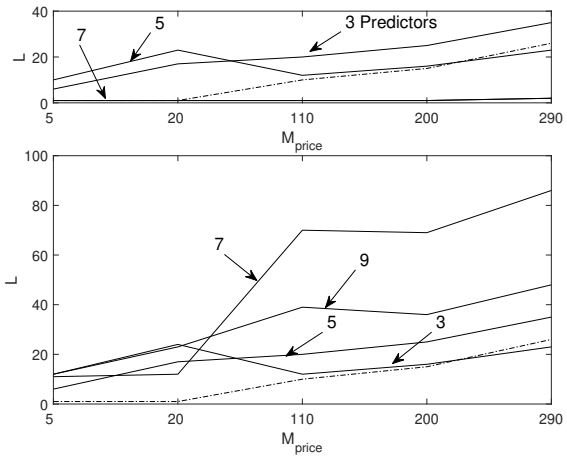

Figure 2.3: L corresponding to best MSE values subject to an upper limit on condition number of the covariance matrix Top: Condition number $< 10^4$ and Bottom: Condition number $< 10^{11}$. RD-SP: $-.$, RD-MP: Solid lines.

As the upper limit on the condition number increases, the MSE improves as we add predictors, and we need a bigger subspace, bigger L, to exploit the information. It is important to notice that the value of L corresponding to the best performance is typically a small fraction of the size of the original subspace, which is $\sum_{j=1}^{J} M_j$.

2.3.4 Directional Change Statistic Performance

Fig. 2.4 shows the directional change statistic values averaged over the next 10 days of prediction. Unsurprisingly, the directional change statistic in the Reduced-Dimension method improves as M_{price} increases, and as we add predictors to a certain point. The best number of predictors, similar to the MSE case, is seven in this plot. The Reduced-Dimension method is also superior to the other two methods, Moving Average and Gauss-Bayes.

Table 2.3 shows the directional change statistic values for different companies. As we can see, the results are consistent for all stocks: the Reduced-Dimension method shows the best performance. It is important to note that the values in Table 2.3 uses the number of predictors associated with best MSE values for the specific company, as represented in Table 2.2, and the same M_{price} for all different cases. In other words, the number of predictors was tuned to MSE and not to the

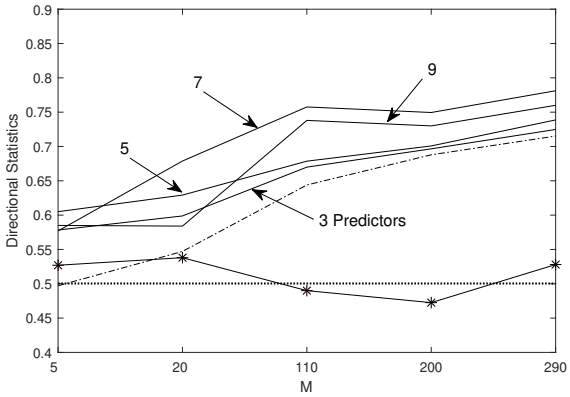

Figure 2.4: Directional Statistic. MA: ..., GB: $-*-$, RD-SP: $-.-$, RD-MP: Solid lines.

Table 2.3: Directional change statistic for different companies for condition number limited to 10^4, $M_{\text{price}} = 210$.

Stock	MP	SP	GB	MA
INTC	0.83	0.72	0.53	0.50
XOM	0.84	0.69	0.50	0.54
CAT	0.87	0.68	0.51	0.55
WMT	0.82	0.73	0.50	0.51
SPY	0.81	0.71	0.51	0.53

directional change statistic. Of course, in practice the number of predictors should be tuned to the specific performance metric of interest. For example, for SPY, six predictors will result in a directional change statistic value that is 0.82 for our Reduced-Dimension method, which is better than the value in the Table 2.3.

2.4 Conclusion

In this work we investigated the effect of using multiple predictors in predicting future price values. The proposed method was illustrated based on predicting daily price data for different companies. The historical data for multiple predictors were downloaded and normalized. The multichannel correlation coefficient was used to select the top predictors for each company. To address the volatility of financial data, an exponential weighting approach was used to track the covariance matrix over time. To do the prediction, we used a filtering operation to extract the information based on principal components. We evaluated our method based on mean squared error

and directional change statistic, and the best number of predictors was identified for each company. The proposed method shows superior performance compared to the conditional mean predictor and the Moving Average method based on both metrics. We also showed the quantitative improvement, for different companies, in the prediction performance when using multiple predictors compared to using only a single predictor, price.

2.5 Data Availability Statement

The data that support the findings of this study are openly available in Yahoo Finance at https://finance.yahoo.com/ and Kenneth R. French Data Library [63].

Chapter 3

Matrix Completion and Stock Price Prediction[5][6]

3.1 Introduction

Matrix completion has application across various fields, such as climate change, image processing, biology, etc. Negahban et al. [64] analyze the nuclear norm relaxation for a general class of noisy observation models, and obtain non-asymptotic error bounds on the Frobenius norm that hold under high-dimensional scaling, and are applicable to both exactly and approximately low-rank matrices. They argue that their proposed method is suitable for different sets of data including financial time series. Athey et al. [65] propose a new approach that estimates the original (incomplete) matrix with lower complexity according to a matrix norm, where they consider the family of Schatten norms based on the singular values of the matrix. They demonstrate the performance of their method using daily return data for different stocks over a 10 year period.

Matrix completion involves filling in the missing entries in a matrix of data. Most of the time, the matrix we wish to recover has a low rank. One approach to solve a matrix completion problem is to minimize the rank of the matrix.

Assume X is the matrix we are trying to recover with observed values indexed by the set Ω, where $\Omega = \{i, j : X_{i,j} \text{ is observed}\}$, The rank minimization problem is formulated as

$$\begin{aligned} &\text{minimize} \quad \text{rank}(Z) \\ &\text{subject to} \quad Z_{ij} = X_{ij}, \quad ij \in \Omega \end{aligned} \tag{3.1}$$

where $Z = [Z_{i,j}]$ is an unknown matrix. The rank minimization problem is a NP- hard [66]. To solve the problem, some relaxation approaches have been proposed. One of the most popular approaches is minimizing the nuclear norm

[5]This chapter is under review for publication

[6]The chapters are written in a way that they are self-contained

$$\begin{aligned} &\text{minimize} \quad \|Z\|_* \\ &\text{subject to} \quad Z_{ij} = X_{ij}, \quad ij \in \Omega, \end{aligned} \tag{3.2}$$

where $\|Z\|_*$ is the nuclear norm of the matrix Z and is equal to the sum of its singular values. It turns out that other than being convex, the nuclear norm is the tightest convex approximation of rank [66].

In real applications, the observed entries may be noisy, and the equality constraint in (3.2) will be too strict, resulting in over-fitting. Therefore, the following relaxed form of (3.2) is often considered for matrix completion with noise:

$$\text{minimize} \quad \frac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_p^2 + \lambda\|Z\|_*, \tag{3.3}$$

where $P_\Omega(X)$ is the operation of projecting matrix X to the space of all matrices with nonzero elements restricted in Ω. Note that $P_\Omega(X) = P_\Omega(Z)$ if and only if $X_{i,j} = Z_{i,j}$, $i, j \in \Omega$. The parameter λ controls the rank of X, and the selection of λ should depend on the noise level [67].

There are different optimization algorithms to solve problem (3.3). Cai et al. [68] developed a singular value thresholding operator followed by an iterative singular value thresholding algorithm to solve the above problem. Different algorithms have been developed based on operator developed in [68] for different problems. Two of the most popular techniques are the Proximal Gradient method and the Augmented Lagrangian method. The Proximal Gradient method is very useful to solve norm-regularized maximum-likelihood problems, while the Augmented Lagrangian method provides a powerful framework to solve convex problems with equality constraints [67].

Since matrix completion recovers missing and/or noisy observations, it also provides a potential prediction method, which has received recent interest. Agarwal et al. [69] propose a principled, robust algorithm with provable guarantees that produces predictions in the presence of missing/noisy data. Their algorithm first "de-noises" the data via singular value thresholding and then learns a relationship between the rows of the estimated low rank matrix through linear regression.

Gillard and Usevich [70] consider the low-rank matrix completion problem, with a specific application to prediction in time series analysis. They consider a matrix completion problem for Hankel matrices and a convex relaxation based on the nuclear norm. Based on new theoretical results and a number of numerical and real examples, they investigate the cases in which the proposed approach can work.

A simple algorithm to solve (3.3) is obtained first by assuming $\hat{X} = UDV'$. Then $\hat{Z}$ is updated by $US_\lambda(D)V'$, where S_λ is a soft-thresholding operator [71], which replaces each diagonal elements of D with $(D_{ii} - \lambda)$. Mazumder et al. [72] then propose an iterative algorithm called Soft-Impute based on this concept with decreasing values of λ. We use a slightly modified version of this algorithm in our proposed setting.

Forecasting financial time series using Matrix Completion methods is a topic that has not received a lot of attention in the literature to the best of our knowledge. In this work we show the performance of the above mentioned algorithm to predict future stock price values. We compare the results of the matrix completion algorithm to a reduced dimension method constructed based on principal components. We compare the numerical results in terms of computational complexity, root-mean-squared error (RMSE) and change statistics of prediction.

3.2 Theoretical Methodology

3.2.1 Proposed Framework

Suppose that we have K samples of vector data, each of length N. Call these row vectors $x_1, x_2, ..., x_K$, where each $x_i \in \mathbb{R}^N (i = 1, ..., K)$ is a row vector of length N and is a sample of end of day price values for N consecutive days:

$$x_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iN} \end{bmatrix}. \tag{3.4}$$

We can stack these vectors together as rows of a $K \times N$ matrix. Suppose that we are now given a new row (row $K + 1$) for which we only have the first M data points, where $M < N$. We wish to

use the information we have so far to estimate the future values for day $M+1$ to N in row $K+1$. The matrix below is a representation of the data matrix together with the new sample in row $K+1$.

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1(M+1)} & \cdots & x_{1N} \\ x_{21} & \cdots & x_{2(M+1)} & \cdots & x_{2N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{K1} & \cdots & x_{K(M+1)} & \cdots & x_{KN} \\ x_{(K+1)1} & \cdots & x_{(K+1)(M+1)} & \cdots & x_{(K+1)N} \end{bmatrix}.$$

The first M data points in each row of X represent the values of end-of-day price data. In terms of the matrix completion problem, the part we plan to predict, $x_{(K+1)(M+1)}$ to $x_{(K+1)N}$, are the missing values, and the rest of the above matrix are the observed data points, Ω.

To implement a matrix completion algorithm, first we need to normalize the data. We divide each row by its Qth entry, and then subtract the row-average vector from each row to get zero mean data.

In order to address the heteroscedasticity of financial data we also assign exponential weights to observations. The matrix of data in the objective functions in equations (3.1), (3.2), and (3.3) is replaced with $\mathrm{diag}(\gamma^{\frac{n}{2}}, \cdots, \gamma^{\frac{1}{2}}, \gamma^{\frac{0}{2}})X$, where $\mathrm{diag}(\gamma^{\frac{n}{2}}, \cdots, \gamma^{\frac{1}{2}}, \gamma^{\frac{0}{2}})$ is a diagonal matrix with $(\gamma^{\frac{n}{2}}, \cdots, \gamma^{\frac{1}{2}}, \gamma^{\frac{0}{2}})$ as diagonal elements with $0 < \gamma < 1$.

$$\Sigma_{xx} = \left(\frac{1-\gamma}{1-\gamma^{k+1}}\right) X^T diag(\gamma^0, \gamma^1, \cdots, \gamma^k) X, \tag{3.5}$$

where $diag(\gamma^0, \gamma^1, \cdots, \gamma^k)$ is a diagonal matrix with $(\gamma^0, \gamma^1, \cdots, \gamma^k)$ as the diagonal elements.

3.2.2 Estimation Techniques

Based on the proposed framework we compare two methods of estimation for future price values.

Iterative thresholding

As mentioned above, points $x_{(K+1)(M+1)}$ to $x_{(K+1)N}$ are missing in matrix X. For iteratively solving the problem, we assume 0 as the initial value for the missing points, and then we implement a modified version of Soft-Impute algorithm proposed by [72]. Remember S_{λ_t} notation is the soft-thresholding operator. Here is the algorithm we use in this study:

assume: $Z^0 = 0$

then for $\lambda_1 \succ \lambda_2 \succ ... \succ \lambda_T$

we have: $Z^t = S_{\lambda_t}(Z^{t-1})$

continue until the stopping criterion is met.

In other words, a series of λ values are used in decreasing order, and the outcome of each step is used as the starting point for the next step. A smaller λ results in a lowering of the importance of the rank term in the objective function, so in each iteration we are actually increasing the rank of the resulting matrix.

Reduced-Dimension

The Reduced-Dimension method was introduced in the previous chapter as an alternative to multivariate conditional mean to predict future stock price values. Principal component analysis is a dimensionality reduction method which is used in various fields including finance [32, 30, 35]. Reduced dimension method is a filtering operation based on principal components that extracts the information out of a data set and reduces the dimension of the problem.

Assume the vector x_i is a multivariate random vector that can be partitioned in the form

$$x_i = \begin{bmatrix} y_i & z_i \end{bmatrix}, \tag{3.6}$$

where y_i has length M and z_i has length $N - M$. The covariance matrix for the random vector x can be written as

$$\Sigma_{xx} = \begin{bmatrix} \Sigma_{yy} & \Sigma_{yz} \\ \Sigma_{zy} & \Sigma_{zz} \end{bmatrix}, \tag{3.7}$$

where Σ_{yy} is the covariance of y and Σ_{zz} is the covariance of z. Knowing the prior distribution of $x = [y, z]$, the Bayesian posterior distribution of z given y is

$$\begin{aligned} \widehat{z}_{z|y} &= \Sigma_{zy}\Sigma_{yy}^{-1}y \\ \widehat{\Sigma}_{z|y} &= \Sigma_{zz} - \Sigma_{zy}\Sigma_{yy}^{-1}\Sigma_{yz}. \end{aligned} \tag{3.8}$$

Consider the eigen-decomposition (or, alternatively, the singular-value decomposition (SVD)) of Σ_{xx}:

$$\Sigma_{xx} = VSV', \tag{3.9}$$

where S is a diagonal matrix of the same dimension as x with non-negative diagonal elements in decreasing order, and V is a unitary matrix. Let $L < N$ be such that the first L eigenvalues in S account for the bulk part of the sum of the eigenvalues. Let V_L be the first L columns of unitary matrix V, and $V_{M,L}$ be the first M rows of V_L. Our filtering operation which is mathematically resolving the noisy observation vector y onto the principle subspace can be written as

$$w = Gy, \tag{3.10}$$

where G is

$$G = (V_{M,L}^{'}V_{M,L})^{-1}V_{M,L}^{'}. \tag{3.11}$$

Substituting y by w in (3.8) leads to:

$$\begin{aligned} \widehat{z}_{z|w} &= \Sigma_{zw}\Sigma_{ww}^{-1}w \\ \widehat{\Sigma}_{z|w} &= \Sigma_{zz} - \Sigma_{zw}\Sigma_{ww}^{-1}\Sigma_{wz}. \end{aligned} \tag{3.12}$$

This set of equations have a similar behavior to (3.8) and since Σ_{ww} has a much lower condition number compared to Σ_{yy}, the prediction results are more trustworthy.

Table 3.1: Computational complexity for different matrix operations

Operation	Complexity	Algorithm
$A \times B$	$\mathcal{O}(n^{nmp})$	Schoolbook matrix multiplication ([73])
C^{-1}	$\mathcal{O}(n^3)$	Gauss-Jordan elimination ([74])
SVD (A)	$\mathcal{O}(n^{mn^2})$	Bidiagonalization and QR algorithm

We compare the numerical results from these two methods first based on computational complexity of the estimation. Then we present plots for RMSE and change statistics.

3.2.3 Performance Metrics

Computational complexity

Computational complexity answers the simple question: how much computational resources are required to do a series of computations? Since computing the exact answer to this question is not always feasible, or justifiable, the *asymptotic* complexity is often discussed instead.

Asymptotic computational complexity We are often interested in functions rate of growth rather than their exact behavior. The rate of growth is often indicated using $O(n)$ notation [73]. Assume A, B, and C are matrices with sizes $n \times m$, $m \times p$, and $n \times n$. Table 3.1 summarizes the computational complexity for various matrix operations, which we will use to calculate the overall complexity of the previous schemes.

We compute the asymptotic computational complexity of the previous two methods. In the Reduced-Dimension we have:

$$\widehat{z}_{RD} = \Sigma_{zw}\Sigma_{ww}^{-1}w \tag{3.13}$$

where $w = Gy$ and, $G = (V_{M,L}^{'}V_{M,L})^{-1}V_{M,L}^{'}$, so we have:

$$\widehat{z}_{RD} = \Sigma_{zw}\Sigma_{ww}^{-1}(V_{M,L}^{'}V_{M,L})^{-1}V_{M,L}^{'}y. \tag{3.14}$$

Using Table 3.1 as reference, the asymptotic complexity of the Reduced-Dimension (ACC_{RD}) method is summarized as:

$$ACC_{RD} = \mathcal{O}\left[KN^2 + N^3 + L^2(3M) + L(M^2 + M(N-M) + M + 2(N-M) + 2L^3\right] \tag{3.15}$$

which, assuming $K \gg N$, ultimately simplifies to $\mathcal{O}(KN^2)$.

For Soft-Impute, we have:

$$\widehat{z}_{SI} = US^*V^T. \tag{3.16}$$

Assuming this operation is done N_{iter} times, based on table 3.1 for the Soft-Impute we have:

$$ACC_{SI} = \mathcal{O}\left[KN^2 + N + KL^2 + L^2N\right] \times N_{\text{iter}}, \tag{3.17}$$

which similar to Reduced-Dimension is equivalent to $\mathcal{O}(KN^2) \times N_{\text{iter}}$. We estimate N_{iter} to be a linear function of N, meaning we can write the computational complexity of this method as $\mathcal{O}(KN^3)$. We can conclude that Soft-Impute has higher computational complexity than the Reduced-Dimension.

Empirical computational complexity Another way to compare the computational complexity of the two methods is by looking at *empirical* computational complexity, i.e., the amount of time required for each algorithm to complete. First we need a stopping criterion for the Soft-Impute. The stopping criterion used in this study is given by (3.18). Based on this criterion, the algorithm continues to iterate until the normalized improvement in a new iteration is less than a pre-specified threshold, relative to the improvement achieved in the first iteration. Once the threshold, ε, is reached, we assume the algorithm has converged.

$$\frac{\left\|\widehat{z}_{SI_t} - \widehat{z}_{SI_{t-1}}\right\|}{\left\|\widehat{z}_{SI_2} - \widehat{z}_{SI_1}\right\|} < \varepsilon \tag{3.18}$$

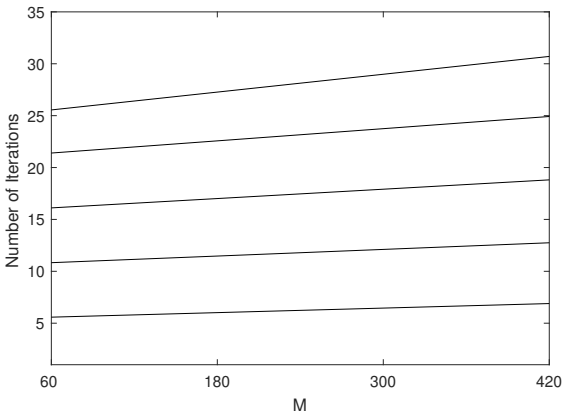

Figure 3.1: Number of iterations for Soft-Impute for different thresholds, 10^{-1} (at the bottom), 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5} (on top).

where $\widehat{z}_{SI_t}$ is the prediction from Soft-Impute at iteration t.

3.3 Numerical Results

To illustrate and evaluate the matrix completion approach and to compare it with the Reduced-Dimension, the historical price data for General Electrics was downloaded from Yahoo Finance[7]. The raw data was transformed into four Hankel matrices with different lengths. We use the values 60, 180, 300 and 420 for M. The data in each matrix was then normalized, centered, and exponentially weighted ($\gamma = 0.985$). The results in this sections are based on averaging over 1000 observations.

Empirical computational complexity

Figure.3.1 shows the number of iterations in the SoftImpute algorithm for five different thresholds: 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}.

As the plot shows, the smaller the threshold value, the higher the number of iterations. Also the larger the value of M, the higher the number of iterations required before stopping. As we will show later, the performance of the method improves with bigger M values.

[7]https://finance.yahoo.com/quote/GE/history/

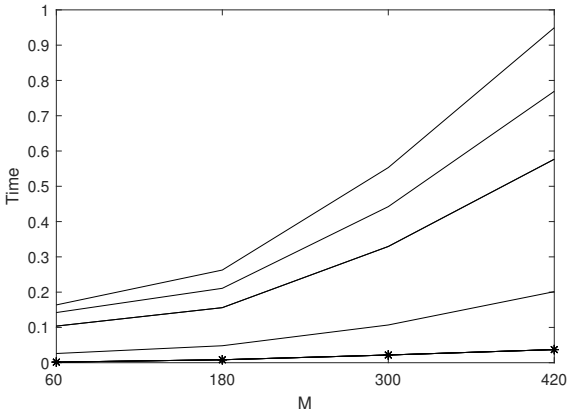

Figure 3.2: Solid line: Execution time for Soft-Impute for different thresholds,10^{-1} (at the bottom), 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5} (on top). *: Execution time for Reduced-Dimension.

As we mentioned earlier, we measure the empirical computational complexity based on the amount of time it takes to generate a prediction for each observation. Having determined how many iterations it takes based on each threshold and value of M, we now investigate the amount of time required to get to that point. Figure 3.2 shows the amount of time required for each algorithm to run for a single prediction.

As the plot shows, for Soft-Impute, as the thresholds gets smaller we need more iterations which results in more amount of time. That is also the case when M gets bigger which is consistent with Fig 3.1. The execution time associated with Reduced-Dimension is presented in Figure.3.2 and as we can see is significantly smaller than the other method since fewer number of iterations requires less amount of time. The difference between the two methods efficiency becomes more obvious important when we consider performance metrics associated with different thresholds in the Soft-Impute.

RMSE

To compare the performance of the methods described above, next we consider RMSE, which is one of the most important metrics in prediction performance. Figure.3.3 shows the RMSE of the two different methods. As we can see, the Soft-Impute has relatively low RMSE values. Recall as the threshold gets smaller, more iterations are required. At the same time, more iterations result

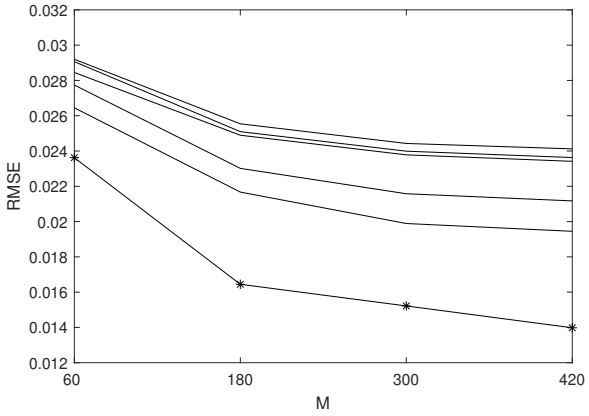

Figure 3.3: Solid line: RMSE for Soft-Impute for different thresholds, 10^{-1} (at the bottom), 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5} (on top).*: RMSE for Reduced-Dimension.

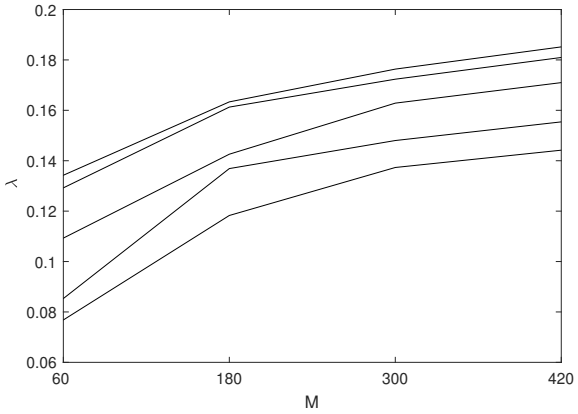

Figure 3.4: λ for Soft-Impute for different thresholds, 10^{-1} (on top), 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5} (at the bottom).

in smaller RMSE values. Although the RMSE values associated with Soft-Impute are fairly small, the RMSE values from Reduced-Dimension is much smaller, which means Reduced-Dimension is superior in terms of RMSE.

Dimension of the Subspace

Remember Soft-Impute algorithm uses a series of λ values in decreasing order and in each step by choosing a smaller λ, the rank of the resulting matrix increases. The λ values associated with Figure.3.5 are demonstrated in Figure.3.4. As M gets larger, more iterations are involved and the

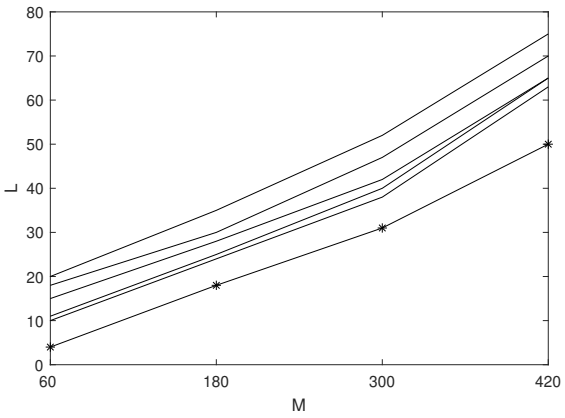

Figure 3.5: Solid line: Best L for Soft-Impute for different thresholds, 10^{-1} (at the bottom), 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5} (on top).*: Best L for Reduced-Dimension.

value of λ increases.

Remember L represent the first L eigenvalues or diagonal elements of S, so we can look at the L values associated with each λ. As λ increases, the rank of the resulting matrix increases, more eigenvalues are included in the algorithm and the associated L increases as well. As M gets larger, there is more information to extract, and a larger subspace is needed.

Change statistics

In applications involving stock-price data, other than predicting the actual future values, predicting the direction of the change in the price values is also very important [47]. We use the change statistic as a metric to measure the ability of the Soft-Impute and Reduced-Dimension algorithms in predicting correct directions of change in the price values.

This evaluation metric measures the direction of change between the actual and the estimated values. For $j = 1, 2, ..., N - M$ we define $b_{i\theta}$ as:

$$b_{i\theta} = \begin{cases} 1, & \text{if } (z_{ij} - z_{today})(\hat{z}_{ij} - z_{today}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.19}$$

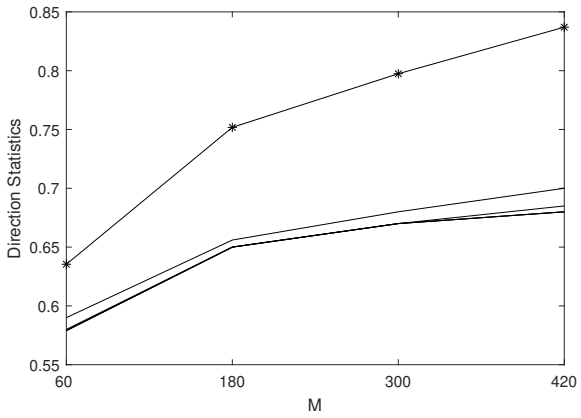

Figure 3.6: Change statistics for Soft-Impute for different thresholds, from 10^{-1} at the bottom to 10^{-5} on top.

where z_{today} is stock price for today, and z_{ij} is the true price. Then D_j, the change statistic for day j, is equal to

$$D_j = \frac{1}{K} \sum_{i=1}^{K} b_{ij}, \tag{3.20}$$

which is a number between 0 and 1.

Figure.3.6 demonstrates the results for this metric. As we can see, as M gets larger, the performance of both methods improve. For Soft-Impute, different thresholds do not make much of a difference in terms of this metric, so if the direction of the market is the main focus of the prediction using Soft-Impute, then a bigger threshold is recommended since it requires fewer iterations and ultimately less time to execute. As the Figure.3.6 shows, although Soft-Impute shows good performance especially for larger M values, the Reduced-Dimension shows significantly better performance which ultimately makes this method a better candidate in estimating future price values.

3.4 Conclusion

In this work we investigated an iterative matrix completion method called Soft-Impute to estimate future price values. This method shows good performance in terms of different metrics.

We compared the Soft-Impute to Reduced-Dimension, a method based on principal components. The Reduced-Dimension shows superior behavior in terms of empirical computational complexity. The performance of both methods is close when it comes to RMSE and change statistics, with a slight advantage in favor of Reduced-Dimension.

Although matrix completion has received considerable attention in the past few years, investigating the power of this group of methods as estimators for prediction, weather in finance or in other application domains, remains to be fully explored.

Bibliography

[1] Mahsa Ghorbani and Edwin KP Chong. Stock price prediction using principal components. *Plos one*, 15(3):e0230124, 2020.

[2] Kenneth R French and Richard Roll. Stock return variances: The arrival of information and the reaction of traders. *Journal of Financial Economics*, 17(1):5–26, 1986.

[3] Andrew W Lo and A Craig MacKinlay. Stock market prices do not follow random walks: Evidence from a simple specification test. *The Review of Financial Studies*, 1(1):41–66, 1988.

[4] Eugene F Fama and Kenneth R French. Permanent and temporary components of stock prices. *Journal of Political Economy*, 96(2):246–273, 1988.

[5] David M Cutler, James M Poterba, and Lawrence H Summers. Speculative dynamics. *The Review of Economic Studies*, 58(3):529–546, 1991.

[6] Navin Chopra, Josef Lakonishok, Jay R Ritter, et al. Performance measurement methodology and the question of whether stocks overreact/1991: 130. *BEBR faculty working paper; no. 91-0130*, 1991.

[7] Werner FM Bondt and Richard Thaler. Does the stock market overreact? *The Journal of Finance*, 40(3):793–805, 1985.

[8] Vincent Bogousslavsky. Infrequent rebalancing, return autocorrelation, and seasonality. *The Journal of Finance*, 71(6):2967–3006, 2016.

[9] Shah Saeed Hassan Chowdhury, M Arifur Rahman, and M Shibley Sadique. Behaviour of stock return autocorrelation in the gcc stock markets. *Global Business Review*, 16(5):737–746, 2015.

[10] Lili Li, Shan Leng, Jun Yang, and Mei Yu. Stock market autoregressive dynamics: A multinational comparative study with quantile regression. *Mathematical Problems in Engineering*, 2016, 2016.

[11] Shah Saeed Hassan Chowdhury, M Arifur Rahman, and M Shibley Sadique. Stock return autocorrelation, day of the week and volatility. *Review of Accounting and Finance*, 2017.

[12] Wen-Jun Xue and Li-Wen Zhang. Stock return autocorrelations and predictability in the chinese stock market—evidence from threshold quantile autoregressive models. *Economic Modelling*, 60:391–401, 2017.

[13] Eugene F Fama and Kenneth R French. Dividend yields and expected stock returns. *Journal of Financial Economics*, 22(1):3–25, 1988.

[14] Jeffrey Jaffe, Donald B Keim, and Randolph Westerfield. Earnings yields, market values, and stock returns. *The Journal of Finance*, 44(1):135–148, 1989.

[15] Eugene F Fama and Kenneth R French. The cross-section of expected stock returns. *The Journal of Finance*, 47(2):427–465, 1992.

[16] Annastiina Silvennoinen and Timo Teräsvirta. Multivariate autoregressive conditional heteroskedasticity with smooth transitions in conditional correlations. Technical report, SSE/EFI Working Paper Series in Economics and Finance, 2005.

[17] Patrick Dennis, Stewart Mayhew, and Chris Stivers. Stock returns, implied volatility innovations, and the asymmetric volatility phenomenon. *Journal of Financial and Quantitative Analysis*, 41(2):381–406, 2006.

[18] Cheoljun Eom and Jong Won Park. Effects of common factors on stock correlation networks and portfolio diversification. *International Review of Financial Analysis*, 49:1–11, 2017.

[19] Nebojsa Dimic, Jarno Kiviaho, Vanja Piljak, and Janne Äijö. Impact of financial market uncertainty and macroeconomic factors on stock–bond correlation in emerging markets. *Research in International Business and Finance*, 36:41–51, 2016.

[20] Liyan Han, Qiuna Lv, and Libo Yin. The effect of oil returns on the stock markets network. *Physica A: Statistical Mechanics and its Applications*, 533:122044, 2019.

[21] Maciej Augustyniak. Maximum likelihood estimation of the markov-switching garch model. *Computational Statistics & Data Analysis*, 76:61–75, 2014.

[22] Charles Stein. Estimation of a covariance matrix, Rietz lecture. In *39th Annual Meeting IMS, Atlanta, GA, 1975*, 1975.

[23] Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

[24] Saburo Ohno and Tomohiro Ando. Stock return predictability: A factor-augmented predictive regression system with shrinkage method. *Econometric Reviews*, 37(1):29–60, 2018.

[25] Liusha Yang, Romain Couillet, and Matthew R McKay. Minimum variance portfolio optimization with robust shrinkage covariance estimation. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 1326–1330. IEEE, 2014.

[26] Olivier Ledoit and Michael Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621, 2003.

[27] J Welles Wilder. *New concepts in technical trading systems*. Trend Research, 1978.

[28] Jonathan Rougier. Ensemble averaging and mean squared error. *Journal of Climate*, 29(24):8865–8870, 2016.

[29] Louis L Scharf and Cédric Demeure. *Statistical signal processing: detection, estimation, and time series analysis*, volume 63. Addison-Wesley Reading, MA, 1991.

[30] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.

[31] Ian Jolliffe. Principal component analysis. In *International Encyclopedia of Statistical Science*, pages 1094–1096. Springer, 2011.

[32] Huseyin Ince and Theodore B Trafalis. Kernel principal component analysis and support vector machines for stock price prediction. *IIE Transactions*, 39(6):629–637, 2007.

[33] William Fung and David A Hsieh. Empirical characteristics of dynamic trading strategies: The case of hedge funds. *The Review of Financial Studies*, 10(2):275–302, 1997.

[34] Carol Alexander. *Market risk analysis, value at risk models*, volume 4. John Wiley & Sons, 2009.

[35] Ravi Shukla and Charles Trzcinka. Sequential tests of the arbitrage pricing theory: a comparison of principal components and maximum likelihood factors. *The Journal of Finance*, 45(5):1541–1564, 1990.

[36] Huanhuan Yu, Rongda Chen, and Guoping Zhang. A svm stock selection model within pca. *Procedia computer science*, 31:406–412, 2014.

[37] Xiao Zhong and David Enke. Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications*, 67:126–139, 2017.

[38] Jie Wang and Jun Wang. Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks. *Neurocomputing*, 156:68–78, 2015.

[39] Giorgia Pasini. Principal component analysis for stock portfolio management. *International Journal of Pure and Applied Mathematics*, 115(1):153–167, 2017.

[40] Paresh Kumar Narayan, Seema Narayan, and Kannan Sivananthan Thuraisamy. Can institutions and macroeconomic factors predict stock returns in emerging markets? *Emerging Markets Review*, 19:77–95, 2014.

[41] Wayne F Velicer and Douglas N Jackson. Component analysis versus common factor analysis: Some issues in selecting an appropriate procedure. *Multivariate behavioral research*, 25(1):1–28, 1990.

[42] David J Bartholomew, Fiona Steele, and Irini Moustaki. *Analysis of multivariate social science data*. Chapman and Hall/CRC, 2008.

[43] Robert R Meglen. Examining large databases: a chemometric approach using principal component analysis. *Marine Chemistry*, 39(1-3):217–237, 1992.

[44] Kris Hermus, Patrick Wambacq, et al. A review of signal subspace speech enhancement and its application to noise robust speech recognition. *EURASIP Journal on Advances in Signal Processing*, 2007(1):045821, 2006.

[45] Donald W Tufts, Ramdas Kumaresan, and Ivars Kirsteins. Data adaptive signal estimation by singular value decomposition of a data matrix. *Proceedings of the IEEE*, 70(6):684–685, 1982.

[46] Michel Ballings, Dirk Van den Poel, Nathalie Hespeels, and Ruben Gryp. Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, 42(20):7046–7056, 2015.

[47] Jingtao Yao and Chew Lim Tan. A case study on using neural networks to perform technical forecasting of forex. *Neurocomputing*, 34(1-4):79–98, 2000.

[48] Mustafa U Torun, Ali N Akansu, and Marco Avellaneda. Portfolio risk in multiple frequencies. *IEEE Signal Processing Magazine*, 28(5):61–71, 2011.

[49] Christopher I Byrnes, Anders Lindquist, and Yishao Zhou. On the nonlinear dynamics of fast filtering algorithms. *SIAM Journal on Control and Optimization*, 32(3):744–789, 1994.

[50] William Brock, Josef Lakonishok, and Blake LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of finance*, 47(5):1731–1764, 1992.

[51] Mark P Taylor and Helen Allen. The use of technical analysis in the foreign exchange market. *Journal of international Money and Finance*, 11(3):304–314, 1992.

[52] KJ Hong and Stephen Satchell. Time series momentum trading strategy and autocorrelation amplification. *Quantitative Finance*, 15(9):1471–1487, 2015.

[53] Szilárd Pafka, Marc Potters, and Imre Kondor. Exponential weighting and random-matrix-theory-based filtering of financial covariance matrices for portfolio optimization. *arXiv preprint cond-mat/0402573*, 2004.

[54] Andrew Ang and Geert Bekaert. Stock return predictability: Is it there? *The Review of Financial Studies*, 20(3):651–707, 2006.

[55] John Y Campbell and Samuel B Thompson. Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies*, 21(4):1509–1531, 2007.

[56] Michael S Rozeff. Dividend yields are equity risk premiums. 1984.

[57] Robert J Hodrick. Dividend yields and expected stock returns: Alternative procedures for inference and measurement. *The Review of Financial Studies*, 5(3):357–386, 1992.

[58] Eugene F Fama and Kenneth R French. Business conditions and expected returns on stocks and bonds. *Journal of financial economics*, 25(1):23–49, 1989.

[59] John Y Campbell. Stock returns and the term structure. *Journal of financial economics*, 18(2):373–399, 1987.

[60] Martin Lettau and Sydney Ludvigson. Consumption, aggregate wealth, and expected stock returns. *the Journal of Finance*, 56(3):815–849, 2001.

[61] Owen Lamont. Earnings and expected returns. *The journal of Finance*, 53(5):1563–1587, 1998.

[62] Jacob Benesty, Jingdong Chen, and Yiteng Huang. *Microphone array signal processing*, volume 1. Springer Science & Business Media, 2008.

[63] Kenneth R French. Kenneth r. french-data library. *Tuck-MBA program web server. http://mba. tuck. dartmouth. edu/pages/faculty/ken. french/data_library. html (accessed October 20, 2010)*, 2012.

[64] Sahand Negahban and Martin J Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, pages 1069–1097, 2011.

[65] Susan Athey, Mohsen Bayati, Nikolay Doudchenko, Guido Imbens, and Khashayar Khosravi. Matrix completion methods for causal panel data models. Technical report, National Bureau of Economic Research, 2018.

[66] Maryam Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.

[67] Xiaowei Zhou, Can Yang, Hongyu Zhao, and Weichuan Yu. Low-rank modeling and its applications in image analysis. *ACM Computing Surveys (CSUR)*, 47(2):36, 2015.

[68] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):2010, 1956.

[69] Anish Agarwal, Muhammad Jehangir Amjad, Devavrat Shah, and Dennis Shen. Time series analysis via matrix estimation. *arXiv preprint arXiv:1802.09064*, 2018.

[70] Jonathan Gillard and Konstantin Usevich. Structured low-rank matrix completion for forecasting in time series analysis. *International Journal of Forecasting*, 34(4):582–597, 2018.

[71] David L Donoho, Iain M Johnstone, Gérard Kerkyacharian, and Dominique Picard. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(2):301–337, 1995.

[72] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322, 2010.

[73] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[74] Richard William Farebrother. *Linear least squares computations*. Marcel Dekker, Inc., 1988.