ALGORITHMIC TRADING STRATEGIES USING DYNAMIC MODE DECOMPOSITION: APPLIED TO TURKISH STOCK MARKET

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET CAN SAVAŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
FINANCIAL MATHEMATICS

JULY 2017

Approval of the thesis:

ALGORITHMIC TRADING STRATEGIES USING DYNAMIC MODE DECOMPOSITION: APPLIED TO TURKISH STOCK MARKET

submitted by **MEHMET CAN SAVAŞ** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Financial Mathematics, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** ______________

Assoc. Prof. Dr. Yeliz Yolcu Okur
Head of Department, **Financial Mathematics** ______________

Assoc. Prof. Dr. Yeliz Yolcu Okur
Supervisor, **Financial Mathematics, METU** ______________

Prof. Dr. Bülent Karasözen
Co-supervisor, **Department of Mathematics & Institute of Applied Mathematics, METU** ______________

Examining Committee Members:

Prof. Dr. Gerhard Wilhelm Weber
Financial Mathematics, METU ______________

Assoc. Prof. Dr. Yeliz Yolcu Okur
Financial Mathematics, METU ______________

Prof. Dr. Bülent Karasözen
Department of Mathematics & Institute of Applied Mathematics, METU ______________

Asst. Prof. Dr. Hamdullah Yücel
Scientific Computing, METU ______________

Asst. Prof. Dr. Murat Uzunca
Department of Industrial Engineering, University of Turkish Aeronautical Association ______________

Date: ______________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: MEHMET CAN SAVAŞ

Signature :

ABSTRACT

ALGORITHMIC TRADING STRATEGIES USING DYNAMIC MODE DECOMPOSITION: APPLIED TO TURKISH STOCK MARKET

Savaş, Mehmet Can

M.S., Department of Financial Mathematics

Supervisor : Assoc. Prof. Dr. Yeliz Yolcu Okur

Co-Supervisor : Prof. Dr. Bülent Karasözen

July 2017, 48 pages

Algorithmic trading schemes are growing of importance in modern financial world. Each year, increasing proportion of the total trading volume is handled by algorithmic trading systems and they have become a fundamental element of modern day trading. We demonstrate the application of an algorithmic trading strategy using dynamic mode decomposition and genetic algorithm. The dynamic mode decomposition is a data analysis tool which is capable of characterizing the dynamical systems in an equation-free manner by decomposing the system into low-rank structures, dynamic modes, whose temporal evolution is known. The method enables financial market prediction using dynamic modes. In order to improve the prediction success of the method, we use a complementary technical analysis tool which is optimized with genetic algorithm. We are able to build algorithmic trading strategies using dynamic mode decomposition and test them in Turkish stock market. We conclude that dynamic mode decomposition is a capable method to analyze stock markets.

Keywords: Algorithmic trading, dynamic mode decomposition, portfolio selection, market timing, Koopman operator, dynamical system

ÖZ

DİNAMİK MOD AYRIŞIMI KULLANARAK ALGORİTMİK TİCARET STRATEJİLERİ: TÜRK HİSSE SENEDİ PİYASASINA UYGULAMASI

Savaş, Mehmet Can
Yüksek Lisans, Finansal Matematik Bölümü
Tez Yöneticisi : Doç. Dr. Yeliz Yolcu Okur
Ortak Tez Yöneticisi : Prof. Dr. Bülent Karasözen

Temmuz 2017, 48 sayfa

Algoritmik ticaretin önemi modern finans dünyasında gün geçtikçe artmaktadır. Her yıl, algoritmik ticaretin ticaret hacmindeki payı artmakta ve modern ticaretin önemli bir parçası haline gelmektedir. Bu çalışmada dinamik mod ayrışımı kullanan bir algoritmik ticaret stratejisi geliştireceğiz. Dinamik mod ayrışımı, dinamik sistemleri zaman dinamiği bilenen daha küçük kerteli yapılara ayırarak karakterize eden, denklemsiz bir veri analiz metodudur. Metot bu yapılar sayesinde finansal tahmin yapmamıza imkan vermektedir. Metodun tahmin yeteneğini geliştirmek için genetik algoritma ile optimize edilmiş tamamlayıcı bir finansal teknik analiz stratejisi kullanılmıştır. Bu sayede dinamik mod ayrışımı kullanan algoritmik ticaret stratejileri geliştirilmiştir. Sonuç olarak, dinamik mod ayrışımı hisse senedi piyasalarını analiz edebilecek uygun bir metottur.

Anahtar Kelimeler : algoritmik ticaret stratejisi, dinamik mod ayrışımı, portfolyo seçimi, ticaret zamanlaması, dinamik sistemler, Koopman operatörü

To My Family

ACKNOWLEDGMENTS

I would like to express my very great appreciation to my thesis supervisors Assoc. Prof. Dr. Yeliz Yolcu Okur and Prof. Dr. Bülent Karasözen for their patient guidance, enthusiastic encouragement and valuable advices during the development and preparation of this thesis.

TABLE OF CONTENTS

ABSTRACT vii

ÖZ ix

ACKNOWLEDGMENTS xiii

TABLE OF CONTENTS xv

LIST OF FIGURES xix

LIST OF TABLES xxi

LIST OF ABBREVIATIONS xxiii

CHAPTERS

1 INTRODUCTION 1

2 DYNAMIC MODE DECOMPOSITION 5

2.1 DMD Definition 6

2.2 Exact DMD Definition 10

2.3 Time-Delay Coordinates 11

2.4 Connection with Koopman Theory 14

3 IMPLEMENTATION OF THE DMD TO FINANCE AND BUILDING THE ALGORITHMIC TRADING STRATEGY 17

3.1 The DMD in Finance 18

3.2 Data 23

3.3 Preliminary Training Method 24

3.4 The Algorithmic Trading Strategy 28
3.5 Testing the Strategies 29
3.5.1 Transaction Costs 30
3.5.2 Performance of the Strategy I 31
3.5.3 Performance of the Strategy II 32
3.5.4 Performance of the Strategy III 33
4 CONCLUSION AND OUTLOOK 35
REFERENCES 37
APPENDICES
A 41
A.1 Krylov Sequence and Subspaces 41
A.2 QR Decomposition 41
A.3 Singular Value Decomposition (SVD) 41
A.4 Hessenberg Matrix 42
A.5 Arnoldi Iteration 42
A.6 Moore - Penrose Pseudoinverse 43
A.7 Savitzky-Golay Filter 43
B 45
B.1 Technical Analysis and Technical Indicators 45
B.2 Simple Moving Average and Simple MA Strategy 45
C 47
C.1 Genetic Algorithm 47

C.2 Optimization of the Simple Moving Average Strategy using Genetic Algorithm 48

LIST OF FIGURES

Figure 2.1 An overview of the historical data of BIST 100 Index. 12

Figure 2.2 Smoothed data using Savitzky-Golay filter. 12

Figure 2.3 The DMD solution of the smoothed data. 13

Figure 2.4 The DMD solution of the augmented data matrix. 14

Figure 3.1 An example plot of the DMD eigenvalues on the complex plane along with the unit circle. 19

Figure 3.2 The amplitudes of DMD modes. 21

Figure 3.3 The components of the growing DMD mode. 21

Figure 3.4 The flowchart of the DMD implementation in finance. 23

Figure 3.5 The success rate matrix. 25

Figure 3.6 The flowchart of the hotspot selection algorithm. 27

Figure 3.7 An overview of the BIST 100 Index over the test period. 30

Figure 3.8 Strategy I perfomance result. 31

Figure 3.9 Strategy II performance result. 32

Figure 3.10 Strategy III perfomance result. 34

Figure C.1 The result of the genetic algorithm. 48

LIST OF TABLES

Table 3.1 Algorithmic trading strategies. 29

Table 3.2 Commission rates table (2017). 31

Table 3.3 Performance table I 32

Table 3.4 Performance table II 33

Table 3.5 Performance table III 34

LIST OF ABBREVIATIONS

DMD	Dynamic Mode Decomposition
POD	Proper Orthogonal Decomposition
SVD	Singular Value Decomposition
SMA	Simple Moving Average

CHAPTER 1

INTRODUCTION

Developments of computers and computer sciences in last decades led a significant change in the finance industry. Nowadays, finance industry experiences a digital age. Markets have become fully electronic, therefore, investors can reach to a market from anywhere almost instantly. Information of all kind can be transferred from exchanges to data centers and investors with the speed of light. These advancements have reshaped the finance industry and have promoted growth and development.

The finance industry adopts technological advancements to improve the accessibility of exchanges, increase demand, and ease trading. An appropriate example is the ticker tape system. Basically, the system works by transmitting the latest transaction information from exchange to receiver's machine via telegraph wires. The receiver's machine which is called ticker tape machine prints the information on the tape. This system was used from the late nineteenth century to the mid-twentieth century. Following the advent of the computer networks and powerful computers, the finance industry adopted them in the same manner. In the 1960s, computers replaced ticker tape system made it obsolete. Major exchanges became electronic after the mid-1990s and early 2000s. This transition indeed improved the activity of the exchanges. According to the World Bank data, the total value of stocks traded increased substantially in United States from 3.6 trillion US dollars in 1994 to 29.7 trillion US dollars in 2000 then to 42 trillion US dollars in 2016 (The World Bank 2016). Moreover, the average daily volume has soared by 164% since 2005, according to the data from New York Stock Exchange. Trading costs plummeted and everyone can trade effortlessly.

The technological advancements triggered the growth of cheap computer power and made available for sophisticated trading as well as trading strategy development and testing platforms. This growth changed the way of trading. Physical pits in the exchanges replaced by computer terminals. Orders started to be placed electronically and executed electronically. A new trading mechanism called *algorithmic trading* emerged. *Algorithmic trading* has many definitions but basically, it refers to the trading of securities based strictly on the buy/sell decisions of computer algorithms [6]. The computer algorithms are designed and programmed by traders or experts through rigorous research. They employ mathematical, statistical and financial models incorporated with relevant data. Algorithmic trading schemes include the decision-making process and trading execution process. Often, definitions overlap each other. Quantitative trading is also used to refer to algorithmic trading or its decision-making process and sometimes algorithmic trading is used to refer solely to the execution algorithm. In this thesis, we will use algorithmic trading and quantitative trading interchangeably to refer the decision making process.

The trading strategy is a plan of action designed to achieve a profitable return by going long or short in markets on organized financial exchanges [27]. Algorithmic trading strategies try to achieve this profitable return by computer algorithms. The ultimate goal of algorithmic trading is having an edge over other traders in the means of forecasting ability and speed. The latest technological improvements in exchanges have pushed down their latencies and promoted high-speed trading. For example, Nasdaq reports required time to handle an average order is 98 microseconds. The special type of algorithmic trading strategy called *high-frequency trading* seeks statistical arbitrage opportunities, inefficiencies in the market and tries to exploit them with this speed advantage to have an edge. The forecasting ability of the algorithmic trading strategies depends on the models used and how well they optimized and put together. At first glance, the forecasting ability of computer algorithm seems to be disadvantaged in a field such as investing, where returns may be affected by almost any type of human activity. The human mind has the ability to digest and synthesize a wide variety of information. A computer algorithm, on the other hand, acts only according to instructions given them. However, the computer algorithm has two significant advantages.The first advantage is the speed advantage. It can quickly process a huge amount of data. The second one which seems irrelevant at first, it does not have emotions. The second one is especially important. Fenton et al. showed in [10] that emotions and their regulation play a central role in traders decision making, even though the work is predominantly theorized and dominated by rational analysis. The properly designed algorithmic trading strategy pursues trading profit with the persistent consistency and objectivity of computer logic. This shows the importance of algorithmic trading and provides a rational reason to use it.

Building an algorithmic trading strategy involves many technical steps but the first step is the decision strategy. This strategy sometimes called *alpha model* [24]. An algorithmic trading strategy usually consists several parts which handle different tasks of the trading. Alpha model is the part of the algorithmic trading strategy that forecasts the market and tries to make a profit. Usually, most of the research are focused on this part. There may be other parts such as risk model, transaction cost model, portfolio construction model and execution model. In this thesis, we discard these parts and focus on the alpha model only. Henceforth, when we use the term algorithmic trading strategy, we actually refer to the alpha model of the algorithmic trading strategy.

Basically, we can categorize algorithmic trading strategies into two main categories: theory-driven and data-driven [24]. Theory-driven strategies assume a theory to explain the certain behaviors of the markets and then test these theories whether they can be used to predict the future. These theories are usually backed with economical and mathematical concepts. There are many theory-driven strategies, e.g., trend following strategies [3, 12], mean reversion strategies [2, 9, 20] and mathematical, statistical models [1, 14, 32]. In the case of data-driven models, what they do basically is data mining at some level. The usual inputs are sourced from markets, e.g., price, volume data. Typically these strategies make no assumptions and seek to find patterns that may have some explanatory power about the future. They are based on the premise that the data has sufficient information to predict future and it can be extracted by using certain analytical techniques. Data-driven strategies are usually less widely practiced. The reasons for that are data-driven strategies are significantly more difficult to understand and the mathematics are far more complicated [24]. However, since data-driven strategies are more technically challenging when compared with theory-driven strategies, there are fewer competitors.

Both strategy types have advantages and disadvantages. The advantage of using theory-driven strategy is when the theory is in hand that means we have a complete understanding of the phenomenon. We know how it works and we can forecast it in accordance with the theory. But the prominent disadvantage is formulating the theory. Also, most of the time the theory is an approximation or is simplified version of the reality which many parameters are discarded. This makes theory suitable for some cases and not suitable or invalid for others. The advantage of using the data-driven strategy is that they are able to discern the behaviors regardless of their true reason. Simply put, data-driven strategies can favor empirical approaches without concerning the theory or rationalization behind behaviors. Employing a data-driven strategy is considerably more challenging but at the same time, it is more flexible and rewarding. The important disadvantage of data-driven strategies is the data itself. These strategies are sensitive to the data. It should be sufficiently large to reveal essential information. Also, it should be bias and noise free. Noisy data may contain many false signals. The convenient data can be obtained by data preperation step before feeding it to the strategy which can be time consuming. Another way is buying it from data vendors which may be costly.

After choosing the appropriate strategy type, the next steps are formulating the strategy and backtesting it. The aim of this thesis is building algorithmic trading strategies by adopting the data-driven technique. It is a judicious assumption to think a financial market as a dynamical system. A financial market involves countless traders and their decisions which affect the market. Also, many securities bought and sold prices are changing continuously, financial markets have a strong relevance to time as well. It is difficult to reveal all these relationships and formulate them, instead, we want to use the data created by the financial market to have essential information to speculate. Because of this purpose the fundamental element of the strategy is the dynamic mode decomposition. *Dynamic mode decomposition (DMD)*[7, 18, 28, 30, 31, 34] is an equation-free data analysis tool capable of extracting coherent structures of the dynamical systems. The DMD method requires no assumptions or equations about the system, uses only collected data. Using the DMD method, we will reveal important information which will be sufficient for us to speculate in the financial market.

The significance of this thesis is that it builds an accurate connection between the DMD and the financial market. It shows DMD is a capable tool to use as a building block for algorithmic trading strategies. Also, it shows that accuracy of the DMD strategy can be improved by combining it with other technical analysis tools such as simple moving average.

In the literature, there are several applications of the DMD method in finance. Mann and Kutz [21] applied the DMD method in finance. They showed it is possible to make market predictions using temporal structures of dynamic modes. Also, they used a learning algorithm to find the best inputs for the DMD method for trading in a financial market. We use the same learning algorithm for that purpose. In addition to that, we introduce a search algorithm to find the best parameters for the DMD method if more than one hotspot exists. Hua et al. [13] used the DMD method to find persistent cyclic activities in the market. Cui and Long [8] showed that portfolio selection can be done using dynamic modes and tested their strategy in Chinese stock market. In this thesis, we introduce an asset allocation method to allocate the stocks which are found using their way.

The thesis consists of three chapters. In the next chapter, the dynamic mode decomposition will be introduced. The connection between dynamic mode decomposition and the Koopman theory will be given. In Chapter 3, the application method of the DMD in finance will be defined. Also, we define the algorithmic trading strategies and then we present their backtest results. Chapter 4 concludes the thesis and gives an outlook of the future work.

CHAPTER 2

DYNAMIC MODE DECOMPOSITION

Dynamical systems spontaneously appear in almost every model which is related to a real physical phenomenon. We should write down and solve its governing equations to have a solid understanding of the system. However, this approach poses several challenges. The first challenge emerges from the complexity of the dynamical systems. Dynamical systems are often complicated without expert knowledge about the underlying system and it is almost impossible to write down actual governing equations for most cases. For nearly every case, models are simplified leading to an approximation, sketchy formulation of the system in hand. Sometimes this makes models incapable of explaining the complete phenomenon. The second challenge is after prescribing the governing equations, solving them. Most of the cases, they do not have exact solutions. But numerical solutions can overcome this challenge and they are abundant. Because of the challenges mentioned, data analysis and data minining methods are grown vast in number. The goal of these methods is extracting useful information from data and transform it to the structure for further use.

As mentioned before, we assume that a financial market is a dynamical system. Under this assumption, we adopted dynamic mode decomposition method to analyze the system. Instead of writing down the theory of the financial market, we want to use the data which creates for speculating.

The dynamic mode decomposition is a data analysis tool to extract coherent structures from the data. It can be seen as an ideal combination of spaital dimensionality-reduction techniques, such as the proper orthogonal decomposition (POD) with Fourier transforms in time [19]. At the highest level, the DMD constructs the best-fit linear dynamical system in the least-square sense to the nonlinear dynamical system that generates the data [19]. By using this approximate linear dynamical system we can diagnose the system and even we can make predictions for the future.

The important aspect of the DMD method is the dynamic modes. The dynamic modes are spatially coherent meaning that they are physically more meaningful. Also, they are correlated with time. Each mode is associated with temporal structure that gives growth rate and frequency of oscillation.

The computation of the DMD method is simple. It only takes the collected snapshots of the data. Snapshots are data vectors which represent the data in one state. They contain measurements of the system from different spatial locations. They can be thought as slices in time, together they make up the data. The snapshots can be sampled with regularly spaced intervals with different intervals. The DMD method takes data snapshots $\boldsymbol{v_i}$ from a dynamical system for $i \in \{1, 2, 3, ..., m\}$. m represents the total number of snapshots. Then, it tries to find best fit $\boldsymbol{A}$ of $\boldsymbol{v_{i+1}} = \boldsymbol{A v_i}$ over the $\{i = 1, 2, 3, ..., m - 1\}$ snapshots. The $\boldsymbol{A}$ represents proxy, approximate linear dynamics of the underlying system. Eigendecomposition of the approximate linear dynamics gives dynamic modes and associated eigenvalues. Using them in approximate solution, the best-fit data can be reconstructed.

The DMD method is related with the Koopman operator. The Koopman operator is an infinite-dimensional linear operator that evolves a non-linear dynamical system in time on space of all scalar observable functions. The DMD method approximates the Koopman operator, constructs a linear dynamical system directly from data without using observable functions [19].

The DMD method originated from fluid dynamics community. A brief history about dynamic mode decomposition starts with 2005 when Mezic brought together the ideas of Koopman operator and relating with nonlinear dynamical systems [22]. In 2008, Schmid and Sesterhenn introduced the dynamic mode decomposition [31]. In 2009, Rowley et al. introduced the spectral analysis and decomposition of nonlinear flows [28]. In 2010, Schmid published the dynamic mode decomposition algorithm [30]. In 2014, Williams et al. introduced the kernel based approach for data-driven koopman analysis [36]. In 2014, Tu et al. showed that the dynamic modes computed using Schmid's algorithm computes projections of dynamic modes and improved the algorithm to compute the exact dynamic modes. Also, Tu et. al. DMD algorithm allows using data sampled with irregularly spaced time intervals [34].

The aim of this chapter is to give a solid understanding about the dynamic mode decomposition. The chapter organized to be parallel to the historical evolution of the DMD method. First, Schmid's definition of the DMD method will be given then exact DMD definition defined by Tu et al. will be given as well. The chapter will be concluded with the solution of an important problem of the DMD method.

2.1 DMD Definition

This is the dynamic mode decomposition algorithm defined by Schmid [30]. This definition later improved by Tu et al. [34]. The dynamic modes computed by this definition are not exact modes but their projection. Because of this reason, in this thesis, the dynamic modes are computed by Tu et al. definition.

$$m = \text{number of snapshots taken,}$$
$$n = \text{number of spatial points saved per time snapshot.}$$

Consider ordered data snapshots $\boldsymbol{v_i}$ where $i = 1, 2, 3, ..., m$, sampled by a constant sampling time Δt. These snapshots are concatenated into a data matrix $\boldsymbol{V_1^m}$ as follows,

$$\boldsymbol{V_1^m} = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_m \\ | & | & & | \end{bmatrix}.$$

Each snapshot is a column of the data matrix and the oldest snapshot is the first column, the newest snapshot is the last column of the data matrix. V_1^m is n $\times$ m matrix. The subscript of the V_1^m indicates from which snapshot the data matrix starts and the superscript indicates the last snapshot. So, V_1^m starts from the first snapshot and finishes with the last snapshot.

It is assumed that there is a linear mapping $\boldsymbol{A}$ connects the snapshot v_i to the subsequent snapshot v_{i+1} ,

$$v_{i+1} = \boldsymbol{A} v_i \, .$$

In fact, this $\boldsymbol{A}$ is the Koopman operator that will be discussed in the last section. The mapping is assumed to be constant and holds for whole sampling window. This assumption allows formulating the data snapshots as a Krylov sequence in the following form,

$$\boldsymbol{V_1^m} = \begin{bmatrix} | & | & | & & | \\ v_1 & \boldsymbol{A} v_1 & \boldsymbol{A}^2 v_1 & \dots & \boldsymbol{A}^{m-1} v_1 \\ | & | & | & & | \end{bmatrix}.$$

The goal here is to extract the characteristics of the dynamical process described by $\boldsymbol{A}$.

As the number of the snapshots increases the matrix $\boldsymbol{V_1^m}$ becomes large enough to capture the dominant features of the underlying system. But after a critical number of snapshots, the snapshots becomes linearly dependent. After that point, adding more snapshot does not improve the vector space spanned by the data matrix $\boldsymbol{V_1^m}$. After this critical point, we can express the last snapshot as a linear combination of previous and linearly independent snapshots.

$$v_m = a_1 v_1 + a_2 v_2 + ... + a_{m-1} v_{m-1} + r \tag{2.1}$$

or in matrix form

$$v_m = \boldsymbol{V_1^{m-1}} a + r,$$

with $a^T = \{a_1, a_2, a_3, ..., a_{m-1}\}$ and r is the residual vector. Continue the following,

$$\boldsymbol{A} \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \dots & v_{m-1} \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ v_2 & v_3 & \dots & v_m \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ v_2 & v_3 & \dots & \boldsymbol{V_1^{m-1}} a \\ | & | & & | \end{bmatrix} + r e_{m-1}^T$$

or in matrix form

$$\boldsymbol{A V_1^{m-1}} = \boldsymbol{V_2^m} = \boldsymbol{V_1^{m-1}} S + r e_{m-1}^T \tag{2.2}$$

with $e_{m-1}\in \mathbb{R}^{m-1}$ as the $(m-1)$th unit vector. S corresponds the following matrix,

$$S = \begin{bmatrix} 0 & & & & a_1 \\ 1 & 0 & & & a_2 \\ & \ddots & \ddots & & \vdots \\ & & 1 & 0 & a_{m-2} \\ & & & 1 & a_{m-1} \end{bmatrix}.$$

This S matrix is a companion matrix. It shifts the vectors by one column and approximates the last column of V_2^m in the form of (2.1). The only unknows in S are the coefficients $(a_1, a_2, a_3, ..., a_{m-1})$. By (2.2), the eigenvalues of $\boldsymbol{A}$ can be approximated by the eigenvalues of S through similarity transformation. Then the computation of the least-squares solution for a full rank matrix V_1^{m-1} is given by,

$$\begin{aligned} QRS &= QQ^H V_2^m, \\ RS &= Q^H V_2^m, \\ S &= R^{-1} Q^H V_2^m, \\ a &= R^{-1} Q^H v_m, \end{aligned}$$

with $QR = \boldsymbol{V_1^{m-1}}$ as the economy-size QR-decomposition of the data sequence $\boldsymbol{V_1^{m-1}}$ and Q^H denotes the conjugate-transpose of Q. Another method to get a similar decomposition is a reduction of $\boldsymbol{A}$ to Hessenberg form by the Arnoldi iteration. We can take QR-decomposition of our Krylov matrix which is $\boldsymbol{V_1^{m-1}}$ as $\boldsymbol{V_1^{m-1}} = QR$. Then construct $H = RSR^{-1}$ as a Hessenberg matrix. This yields a decomposition $AQ \approx QH$. Again through similarity transformation eigenvalues of H approximate some of the eigenvalues of $\boldsymbol{A}$. Another way to get the Hessenberg matrix is projections onto successive Krylov spaces but for that we should have $\boldsymbol{A}$ explicitly available . This makes classic Arnoldi iteration unattractive for us.

Despite the fact that above formulation is mathematically true, it can lead an ill-conditioned algorithm which is often not capable of extracting more than the first or the first two dominant dynamic modes [30]. For this reason, more robust implementation is taken into account using singular value decomposition (SVD) and orthogonal similarity transform (see Appendix A for SVD). The SVD of $\boldsymbol{V_1^{m-1}}$ is

$$\boldsymbol{V_1^{m-1}} = U\Sigma W^H,$$

substituting this into (2.2) results

$$\boldsymbol{A} U\Sigma W^H = V_2^m.$$

Rearranging the expression with similarity transform, $\tilde{A} = U^H \boldsymbol{A} U$ gives

$$U^H \boldsymbol{A} U = U^H V_2^m W \Sigma^{-1} \approx \tilde{A}.$$

Another advantage of this operation is dimensionality reduction achieved by the singular value decomposition of $\boldsymbol{V_1^{m-1}} = U\Sigma W^H$ where $U \in \mathbb{C}^{n\times r}$, $\Sigma \in \mathbb{C}^{r\times r}$ and $W \in \mathbb{C}^{(m-1)\times r}$. The parameter r can be choosen to keep the largest dominant modes only. Now, the eigenvalues can be approximated by finding the eigenvalues of the matrix $\tilde{A}$ which is computationally more advantageous. The matrix $\tilde{A}$ is a least-square sense optimal low-dimensional representation of the original dynamical mapping $\boldsymbol{A}$ on the subspace spanned by the proper orthogonal modes (POD) of $\boldsymbol{V_1^{m-1}}$ [18]. Consider the eigenvalue problem:

$$\tilde{A}v_k = \lambda_k v_k,$$

where $k = 1, 2, 3, ..., r$. r is the rank of the approximation chosen for the SVD. The eigenvalues λ_k capture the time dynamics of the originial map $\boldsymbol{A}$. In order to construct the DMD modes, these eigenvalues and eigenvectors mapped back to higher dimensional space of $\boldsymbol{A}$:

$$\phi_k = Uv_k,$$

where ϕ_k denotes the k'th DMD mode and $\boldsymbol{U}$ is the right singular vector of $\boldsymbol{V_1^{m-1}}$.

The aim was to find the low-rank eigendecomposition of the proxy, locally approximate, linear dynamical system $\boldsymbol{A}$ that fits optimally to the measured trajectory v_i for $i = 1, 2, \ldots, m$ in a least-square sense that,

$$\|v_{i+1} - \boldsymbol{A}v_i\|_2 \ ,$$

which is minimized for all i. The optimality condition holds only for the snapshots window. Hovewer, it allows prediction for the future for a limited time.

It is important to note that the DMD modes constructed with this definition are later called *projected DMD modes* by Tu et al. [34] and proven that they are simply projection of the *exact DMD modes* onto the range of $\boldsymbol{V_1^{m-1}}$.

Algorithm of the DMD method introduced in Algorithm 1.

Algorithm 1 DMD (Schmid)

1: Arrange the snapshots $\begin{bmatrix} | & | & & | \\ v_1 & v_2 & \ldots & v_m \\ | & | & & | \end{bmatrix}$ into two matrices,

$$V_1^{m-1} = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \ldots & v_{m-1} \\ | & | & & | \end{bmatrix}, V_2^m = \begin{bmatrix} | & | & & | \\ v_2 & v_2 & \ldots & v_m \\ | & | & & | \end{bmatrix}.$$

2: Compute SVD of V_1^{m-1} ,

$$V_1^{m-1} = U\Sigma W^H.$$

3: Define the matrix $\tilde{A}$,

$$\tilde{A} = U^H V_2^m W \Sigma^{-1}.$$

4: Solve the eigenvalue problem to compute eigenvalues and eigenvectors of $\tilde{A}$,

$$\tilde{A}v_k = \lambda_k v_k.$$

5: Compute the DMD modes,

$$\phi = Uv.$$

2.2 Exact DMD Definition

In this section we present the necessary definition to compute the DMD modes. We indicated that the dynamic modes computed by using Schmid's definition are a projection of the exact dynamic modes. Also, this definition allows using snapshots which are sampled at irregularly spaced intervals.

Definition 2.1 (Exact DMD (Tu et al. [34]))**.** For dataset of two $n \times m$ data matrices,

$$X = [x_1, ..., x_m], Y = [y_1, ..., y_m],$$

define the operator

$$\boldsymbol{A} = YX^{\dagger}, \tag{2.3}$$

where $X^{\dagger}$ is the pseudoinverse of X (see Appendix A). The *dynamic mode decomposition* of the pair (X, Y) is given by the eigendecomposition of $\boldsymbol{A}$. The dynamic modes are the eigenvectors of $\boldsymbol{A}$.

The operator **A** is a least squares solution of the problem $\boldsymbol{A}X = Y$, which is potentially over- or under-constrained problem. If there is an exact solution to this problem (2.3) minimizing $\|A\|_F$, where $\|A\|_F = \sqrt{Tr(AA^H)}$ denotes Frobenius norm. If there is no exact solution then (2.3) minimizes $\|AX - Y\|_F$.

Computing eigendecomposition of $\boldsymbol{A}$ directly could be demanding and inefficient if *n* is large. I nstead of directly computing $\boldsymbol{A}$, rank-reduced and POD- projected representation $\tilde{A}$ will be computed.

Algorithm of Exact DMD formulation:

Algorithm 2 Exact DMD

1: Arrange the data pairs $\{(v_1, v_2), (v_3, v_4)..., (v_{m-1}, v_m)\}$ into matrices,

$$V_1 = \{v_1, v_2, v_3, ..., v_{m-1}\}, V_2 = \{v_2, v_3, v_4, ..., v_m\}.$$

2: Compute the reduced SVD of V_1,

$$V_1 = U\Sigma W^H. \tag{2.4}$$

3: Define the matrix $\tilde{A}$,

$$\tilde{A} = U^H V_2 W \Sigma^{-1}.$$

4: Solve the eigenvalue problem to compute eigenvalues and eigenvectors of $\tilde{A}$,

$$\tilde{A}v = \lambda v.$$

Each non-zero eigenvalue λ is a DMD eigenvalue.

5: Compute the DMD modes,

$$\phi = V_2 W \Sigma^{-1} v.$$

The SVD of V_1 at (2.4) can be utilized to perform a low-rank truncation of the data if low-rank structure is present. It is no necessarity to take all of the singular values into computation but only the dominant ones would be sufficient to explore the underlying dynamics. It is important to note that sometimes this truncation step poses an important problem regarding how many singular values should be kept. It has shown that problems related with control, low-energy singular values are important for balanced models. For the application of this thesis we do not truncate, all singular values are taken into consideration.

The computation of the matrix A could be directly done from $A = V_2 W \Sigma^{-1} U^H$, however, it is more efficient to compute $\tilde{A}$ as in the algorithm. Next theorem gives proof of the DMD modes which computed considering projection of exact DMD modes.

Theorem 2.1 (Koopman mode decomposition and DMD theorem [19])**.** *Let $\tilde{A}v = \lambda v$, with $\lambda \neq 0$, and let $\mathbb{P}_{V_1}$ denote the orthogonal projection onto the image of V_1. Then ϕ given by (1.18) is an eigenvector of $\mathbb{P}_{V_1}$ with eigenvalue λ. Furthermore, if $\hat{\phi}$ is given by (1.24), then $\hat{\phi} = \mathbb{P}_{V_1}\phi$.*

2.3 Time-Delay Coordinates

This section gives a solution to a central issue with the DMD method. It is observed by Tu et al. [34] that the DMD method fails to capture a standing wave in the data. It was startling because the DMD had been successful to extract spatial modes that oscillate at a single fixed frequency. If only measurements of a single sine or cosine wave are collected, the DMD fails to return the conjugate pair of complex eigenvalues and instead returns a single real eigenvalue, which does not capture periodic oscillations [19]. Another problem which is related to this one when data has fewer measurement points than time points. The DMD method was developed and used by the fluid mechanics community to study of large fluid flow fields, where usually $n > m$. n represents measurement points, the number of rows of the data and m represents time point, the number of columns of the data. When $n < m$ then the SVD step of the DMD process produces at most n singular values and this restricts the number of DMD modes and eigenvalues to n. The number of DMD modes and eigenvalues may be insufficient to capture the dynamics over m snapshots in time. The solution of this rank mismatch problem is stacking multiple time-shifted copies of the data in order to construct augmented data matrix.

We can demonstrate this problem and its solution with an example case from finance. If we try to construct the DMD solution of a financial index because of the large difference between the number of rows and the columns of the data matrix, we will have a rank mismatch issue. We are going to use BIST100 Index which is the index of the largest one hundred company that trades on Borsa Istanbul. Figure 2.1 shows the data.

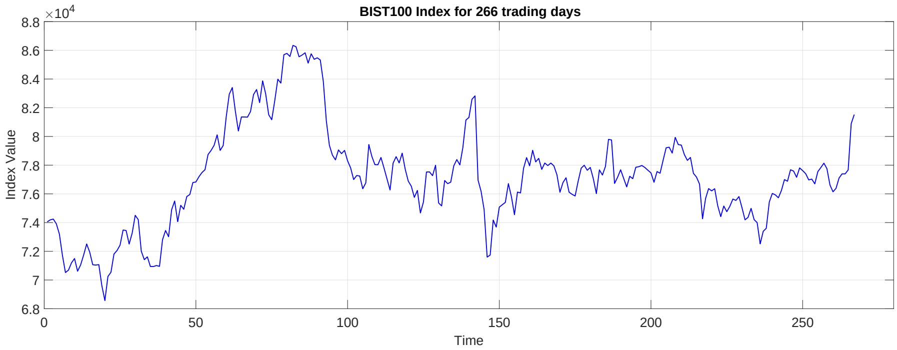

Figure 2.1: An overview of the historical data of BIST 100 Index.

For the simplicity, we smoothed the data by using Savitzky-Golay filter (see Appendix A). It is a filter that is used for smoothing the data to increase the signal-to-noise ratio without distorting the signal. Another filter such as moving average filter can also be used for this purpose [23]. The parameters for the filter are picked to have several peaks on the smooth data for the purpose of demonstration. The next figure shows the smoothed data.

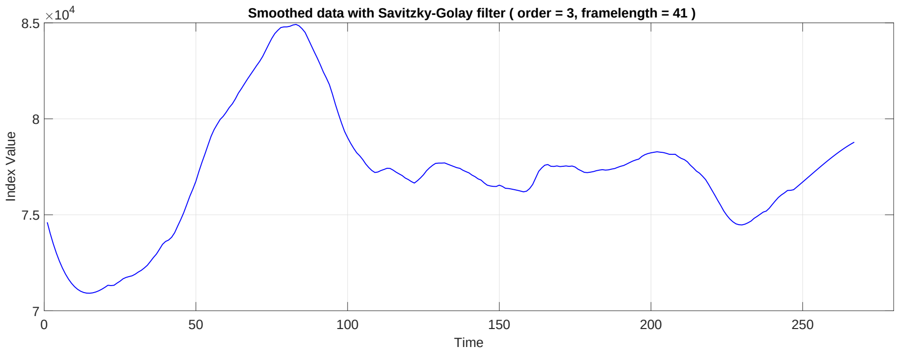

Figure 2.2: Smoothed data using Savitzky-Golay filter.

When we use the smoothed data to construct the DMD solution, the DMD method fails to capture the dynamics and the solution is completely inaccurate. In fact, this problem is quite similar to the standing wave problem [34, 19]. Again, the DMD method returns only one real eigenvalue which fails to capture the dynamics. The reason is we have a great rank mismatch issue. Figure 2.3 shows the DMD solution of the smoothed data.

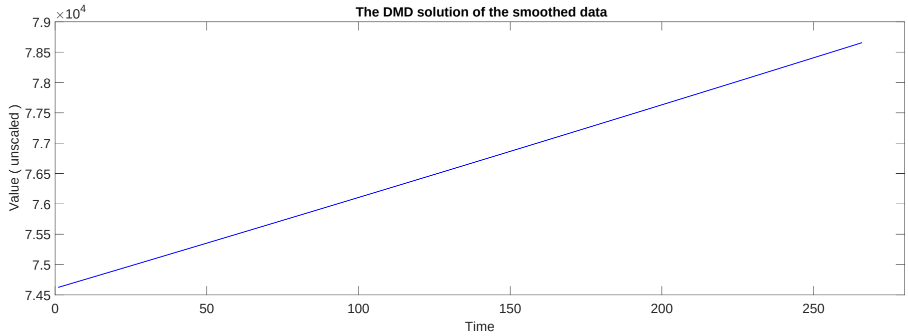

Figure 2.3: The DMD solution of the smoothed data.

In order to overcome the rank mismatch issue, we can use delay coordinates to augment the data matrix. Then we can perform the DMD method on the augmented data matrix.

Delay coordinates refer to an augmented vector obtained by stacking the states at the current time with copies of the states at future times. This stacking can be done either with past measurements or with future measurements even permuting the order will not impact the DMD method. We can construct the augmented data matrix as follows:

$$X_{\text{AUG}} = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_{m-h} \\ | & | & & | \\ | & | & & | \\ x_2 & x_3 & \cdots & x_{m-h+1} \\ | & | & & | \\ & & \vdots & \\ | & | & & | \\ x_h & x_{h+1} & \cdots & x_{m-1} \\ | & | & & | \end{bmatrix},$$

and similarly for X'_{AUG}

$$X'_{\text{AUG}} = \begin{bmatrix} | & | & & | \\ x_2 & x_3 & \cdots & x_{m-h+1} \\ | & | & & | \\ | & | & & | \\ x_3 & x_4 & \cdots & x_{m-h+2} \\ | & | & & | \\ & & \vdots & \\ | & | & & | \\ x_{h+1} & x_{h+2} & \cdots & x_m \\ | & | & & | \end{bmatrix}.$$

It is important that we must preserve the relation $X'_{\text{AUG}} = AX_{\text{AUG}}$. This shift-stacking matrix idea is inspired by the Hankel matrix constructed in the Eigenvalue Realization Algorithm (ERA) [15].

By using augmented data matrix instead of original data matrix, it is possible to increase the rank of the matrix until the system reaches the full rank numerically. The DMD method can be applied to augmented data matrices, the computed DMD mode matrix is now $hn \times m$ matrix, the computed DMD modes are also stacks of h repeats.

Now, we are going to perform the DMD method on augmented data matrix for the example case. In general, there is no principled way to determine to the number of stacking for the augmented matrix. Following the strategy in [5], we choose h such that $hn > m$ and we use the first n elements of the each mode.

Figure 2.4 shows the solution of the DMD method using the augmented data matrix.

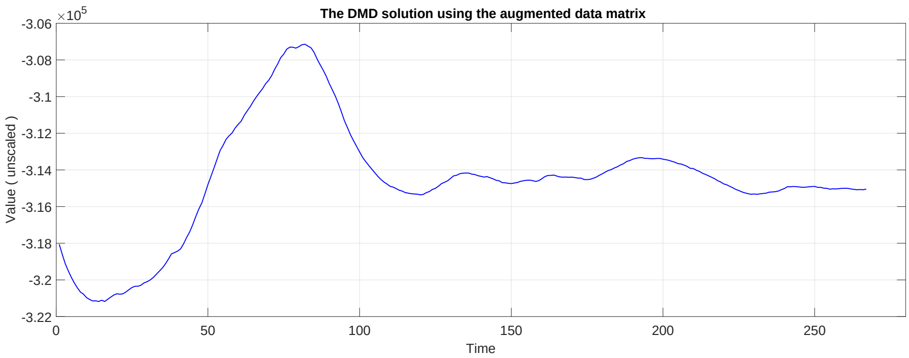

Figure 2.4: The DMD solution of the augmented data matrix.

Figure 2.4 shows that the DMD solution obtained by the augmented data matrix captures the dynamics of the solution well. The figure is quite similar to Figure 2.2. This shows that augmenting data matrix greatly increases the performance of the DMD method if we have rank related issues.

2.4 Connection with Koopman Theory

The DMD method can be thought as a special case of Koopman Theory, defined by Bernard Koopman in 1931 [17]. In this section, first we give the definition Koopman operator. Then we reveal connection between the DMD method and the Koopman operator.

Definition 2.2. Let x $\in$ *M* is a state on a smooth n-dimensional manifold *M*. Given a continuous-time dynamical system,

$$\frac{\mathrm{d}}{\mathrm{dx}} = f(x), \tag{2.5}$$

The Koopman operator $\mathcal{K}$ is an infinite-dimensional linear operator that acts on all observable functions g: $M \to \mathbb{C}$ such that

$$\mathcal{K}g(x) = g\left(f\left(x\right)\right).$$

The Koopman operator is an infinite-dimensional linear operator that acts on Hilbert space of all scalar observable functions g [22, 28]. Thus, this transformation trades nonlinear finite-dimensional dynamics with linear but infinite-dimensional dynamics. This trade-off certainly poses some problems but, linear differential equations can be solved using spectral representation. Also, infinite-dimensional representation can be approximated by sum of modes that finite but sufficiently large.

The Koopman operator can also be defined for discrete-time dynamical systems. The Koopman operator $\mathcal{K}$ defined for a discrete dynamical system is

$$\xi_{k+1} = f(\xi_k)$$

evolving on a finite dimensional manifold M. The Koopman operator acts on scalar functions $g : M \to \mathbb{C}$ according to

$$\mathcal{K}g(\xi) \triangleq g(f(\xi)). \tag{2.6}$$

It is important to note that the linearity property of the Koopman operator is the result of the definition 2.6. It should not be understood as linearization.

In order to use the spectral decomposition of the Koopman operator, we consider the eigenvalue problem

$$\mathcal{K}\phi_j(\xi) = \lambda_j\phi_j(\xi), \quad j = 1, 2, \ldots, \tag{2.7}$$

of the Koopman operator $\mathcal{K}$. The functions $\phi_j(\xi)$ are Koopman eigenfunctions. The Koopman eigenfunctions define a set of coordinates on which it is possible to advance these observables with a linear dynamical system. The λ_j are the eigenvalues of the Koopman operator. From spectral theory, we can represent the evolution of the dynamics, in this case on the observables, using an eigenfunction expansion solution of the Koopman operator. So, we can express vector-valued observable functions in terms of the Koopman eigenfunctions ϕ_j as

$$g(\xi) = \sum_{j=1}^{\infty} \phi_j(\xi) v_j, \tag{2.8}$$

where $(v_j)_{j=1}^{\infty}$ represents the set of vector coefficients called Koopman modes associated with the j'th Koopman eigenfunction. The important assumption here is that we assume that each component of g lies in the span of the eigenfunctions. Using 2.7 and 2.8, we can write

$$\mathcal{K}g(\xi_k) = \sum_{j=1}^{\infty} \lambda_j^k \phi_j(\xi_0) v_j. \tag{2.9}$$

From the last expression, the future solutions can be computed directly by simple multiplication with the Koopman eigenvalue. The Koopman eigenvalues $\{\lambda_j\}_{j=1}^{\infty}$ dictates the growth rate and frequency of each mode [7]. The DMD is used to approximate to Koopman eigenvalues λ_j and modes v_j. Using equation 2.9, we can find the values of observable function at specific time and even we can evolve it in time using the Koopman eigenvalues.

The connection between the DMD method and the Koopman method is elegant. The DMD method approximates the Koopman eigenvalues and the modes directly from the data under suitable conditions. Simply put, the DMD method computes the eigenvalues and the eigenvectors of a finite-dimensional linear model that approximates the infinite-dimensional Koopman operator. There is a crucial point here. The Koopman spectral analysis 2.9 requires observable functions g to provide a better mapping from one state to another. Finding these observable functions are challenging and often requires expert knowledge about the system in hand. Also, there is no principled way of finding these functions. The DMD does not require them, the Koopman modes and eigenvalues are directly approximated from the measured data. So, the data plays critical role in the DMD methods performance. The measurement should be appropriate to have accurate prediction from the DMD method.

In order to explore the connection with the DMD and the Koopman operator, the Koopman framework is constructed. Consider the set of p observables,

$$g_j : \mathcal{M} \to \mathbb{C}, \quad j = 1, 2, \ldots, p.$$

Let $g = [g_1, g_2, ..., g_p]^T$ denote the column vector of observables. Now, construct data matrices Y and $Y^{'}$ with a set of initial conditions $\{x_1, x_2, ..., x_{m-1}\}$ to (2.5). The column of matrix Y are given by $y_k = g(x_k)$. The column of $Y^{'}$ are given by evolving the dynamical system (2.5) forward by Δt and getting the output vector via observables as $y_k^{'} = g\left(f\left(x_k\right)\right)$. Applying the DMD on the data of observables produces $A_Y = Y^{'}Y^{\dagger}$, which is the needed Koopman approximation. It is important to note that Y and $Y^{'}$ computes the DMD on the space of observables instead of on the state-space. The following theorem concludes the connection between DMD method and the Koopman theory [28, 34, 35].

Theorem 2.2 (Koopman mode decomposition and DMD [19])**.** *Let ϕ_k be an eigenfunction of $\mathcal{K}$ with eigenvalue λ_k, and suppose $\phi_k \in span\,\{g_j\}$, so that*

$$\phi_k(x) = w_1 g_1(x) + w_2 g_2(x) + ... + w_p g_p(x) = w \cdot g$$

for some $w = [w_1, w_2, ..., w_p]^T \in \mathbb{C}^p$. If $w \in R(Y)$ where R is the range, then w is a left eigenvector of A_Y, where $A_Y = Y^{'}Y^{\dagger}$ with eigenvalue λ_k so that $\tilde{w}^* A_Y = \lambda_k \tilde{w}^*$.

This shows that the Koopman eigenvalues are the DMD eigenvalues under assumptions that the set of observables is large so that $\phi_k(x) \in span\,\{g_j : j = 1, 2, \ldots, p\}$ and the data is sufficiently rich so that $w \in R(Y)$. This also shows the importance of the choice of observables which allows one to connect the DMD to the Koopman Theory. If this can be done, data snapshots from finite nonlinear dynamical systems can be taken and parametrized as linear infinite-dimensional system that allows to a spectral decomposition.

CHAPTER 3

IMPLEMENTATION OF THE DMD TO FINANCE AND BUILDING THE ALGORITHMIC TRADING STRATEGY

In this chapter, the aim is to build a profitable algorithmic trading strategy that based on dynamic mode decomposition. In order to reach this aim, we first define the method of using dynamic mode decomposition in finance. Then we build our algorithmic strategies using the method and tested them on Borsa Istanbul.

The algorithmic strategies in this chapter are applicable to other stock exchanges as well. But before deploying the strategies for testing, as explained in forthcoming sections, we need to find a couple of parameters to adjust the strategies for the market. In fact, this is a step of building the DMD based algorithmic trading strategy method. These parameters vary for different stock markets. By following the given method, algorithmic trading strategies for other exchanges can be build.

The Borsa Istanbul is a major stock exchange in Turkey. According to the official reports of Borsa Istanbul, more than 290 companies are listed as July of 2017. The total market capitalization of Borsa Istanbul is 187 million US dollars. The BIST 100 Index is the index consists the largest one hundred stocks from Borsa Istanbul, in the means of market capitalization and trading volume. It can be used to measure the performance of the Borsa Istanbul.

The algorithms try to make profits by trading listed stocks in BIST 100 Index.

In the next section, we build a solid framework of using the DMD in finance. In preliminary training method section, we explain the method we follow to build the algorithmic trading strategies. In data section, we give detailed information about data that used. In algorithmic trading strategies section, we explain the algorithms and their trading rules explicitly. In performance evaulation section, we evaulate the performances of our algorithms and compare them with major benchmarks.

In this application, the same learning algorithm to adapt the DMD method in the stock market used as [21] with different hot spot definition to choose the best timing strategy and a similar method used as [8] to construct a portfolio from the modes. As noted before [8], DMD method fails to capture the underlying system properly if there is an exogenous effect on market and makes false predictions. In order to increase the success of the DMD prediction, one of the widely used and accepted technical analysis tool called *simple moving average* is employed to reinforce the trading decision of the model. It will be observed that this tool is increased the prediction success of the model.

3.1 The DMD in Finance

Mann & Kutz [21] used the DMD method to have market timing strategies for different market sectors . They showed that the DMD method can be useful analysis tool for financial markets. Hua et al. [13] used the DMD method to extract cyclic behaviors from financial markets. Cui et al. [8] used the DMD method to build a financial trading strategy and tested in Chinese stock market. We use a similar approach to theirs but our implementation of the DMD method in finance differs at some critical points.

The main difference of this thesis from the literature is that it provides a way of picking a DMD parameter according to it's success when more than one hotspot exists and also it provides the connection between stock performances and their appearance in the dynamic modes using return data measure. Using this connection we can have a legitimate portfolio asset allocation method.

The contribution of this thesis that it gives a way of implementing the DMD method in finance. We explain the way of using the dynamic modes and eigenvalues to make predictions for the financial market. We want to have a rational, functional market timing and portfolio selection strategies.

The DMD method can be thought as decomposing a data matrix into space and time. These components are represented by the DMD modes and the eigenvalues respectively. The key observation here that the DMD method builds a proxy linear dynamical system that approximates the non-linear dynamical system in hand. By using the spectral theory, this proxy linear dynamical system can be decomposed into space and time components. From this rationale, the DMD solution of discrete dynamical system can be written as follows [4]:

$$X \approx \Phi \Lambda^t b \,, \tag{3.1}$$

where Φ represents the DMD modes matrix that each column is a DMD mode, Λ^t represents the eigenvalues matrix in a form of Vandermonde matrix, b represents the coefficients of the initial condition v_1 (this is the first column of the data matrix) in the eigenvector basis, so that $b = v_1\Phi^\dagger$, where $\Phi^\dagger$ is the pseudoinverse of Φ. In this solution, Φ contains the DMD modes that are spatial (space) components. The Λ is often called *time dynamics* and contains the powers of the eigenvalues in the form of Vandermonde matrix as follows:

$$Vm = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^t \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \lambda_2^t \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \lambda_m & \lambda_m^2 & \dots & \lambda_m^t \end{bmatrix} , \tag{3.2}$$

where t is the time index and m represents the number of eigenvalues.

By using equation 3.1, we can construct an approximation of our data matrix at every time point. So, using the DMD method we can get the DMD modes and the eigenvalues. Gathering them in matrices resulting time dynamics and the DMD mode matrix and by using them we can construct an approximate solution. Theoretically, with the DMD modes and eigenvalues, we captured the dynamics of our original dynamical system.

An important thing to know is that the DMD modes are not orthogonal like POD modes. But, unlike POD modes, each of the DMD modes has associated time structure which is the DMD eigenvalue. Therefore, we know how each DMD mode evolves in time. This is the information we are going to use for our market timing strategy.

The DMD eigenvalues represent the time evolution of the approximate linear dynamical system. They are computed using the eigendecomposition of best-fit linear operator $\boldsymbol{A}$ from Chapter 2. Most of the time, the eigenvalues are complex numbers. The real parts of the DMD eigenvalues give the growth rate and the complex parts give the frequency of the oscillation. For our purposes, we do not use the complex part of the eigenvalues, we only focus on the real parts. We represent the DMD eigenvalues on the complex plane along with the unit circle, see Figure 3.1.

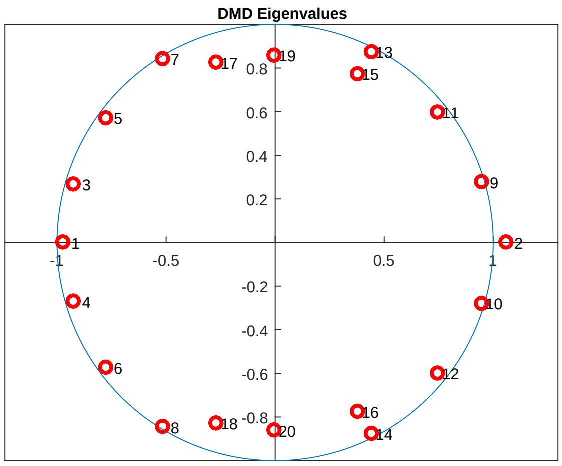

Figure 3.1: An example plot of the DMD eigenvalues on the complex plane along with the unit circle.

The real part of the DMD eigenvalue represents the growth rate of the associated DMD mode. Since we do not interested with their complex part, we can separate them into two groups: The first group is the eigenvalues which have growth rate larger than one and the second group is the eigenvalues which have growth rate smaller than one. We are interested with the first group, we call them *growing eigenvalues*.

Our market timing strategy is simply, get into the market before a stock market rally. The stock market rally means sustained increases in the prices of the stocks. We should be able to detect it before it starts or right after it starts in order to exploit it. The DMD eigenvalues help us to achieve this goal.

We stated that the DMD eigenvalues capture the time dynamics of the dynamical system and we are interested in the growing eigenvalues. In equation 3.1, we constructed our solution with the help of time dynamics and the DMD mode matrix. If we fix the time index t the column number of our original data matrix minus one(the reason is that from first column of the data matrix we get the initial condition.), we exactly get the approximation of our original data matrix. By changing the t we can get a solution for a particular time point. The argument is: for financial speculation we do not have to construct this solution. We just need to know when the stock market will increase and which stocks will increase. It turns out that, if we pay particular attention to the growing eigenvalue we can see that from 3.2, the values in the row of the eigenvalue increase over time. At the same time, the other row values decrease because they are smaller than one and we are taking their powers. This means that the growing eigenvalue will be dominant in the calculation of 3.1. When we think it over 3.1, the growing DMD mode which is associated with the growing eigenvalue will be dominant for the calculation and will increase as time passes. So, we do not really have to consider all the DMD modes and eigenvalues but the growing ones are enough to speculate in the stock market.

Thus, for the market timing strategy we need a growing DMD eigenvalue. When we spot the growing eigenvalue we can say that the growing eigenvalue and its associated mode will be dominant in the calculation of the solution. Therefore, since they increase over time, we expect the overall stock market increase.

In fact, we also set a base framework for our portfolio selection strategy idea. The argument for portfolio selection strategy: We can use the growing DMD mode to determine which stocks are expected to increase as a result of the DMD analysis. Since we are considering only the growing components to approximate the full solution, the growing DMD mode will be scaled only with the growing eigenvalue. This means that we can use the individual components of the growing DMD mode to determine the portfolio stocks. Figure 3.2 and Figure 3.3 will help us to comprehend the idea.

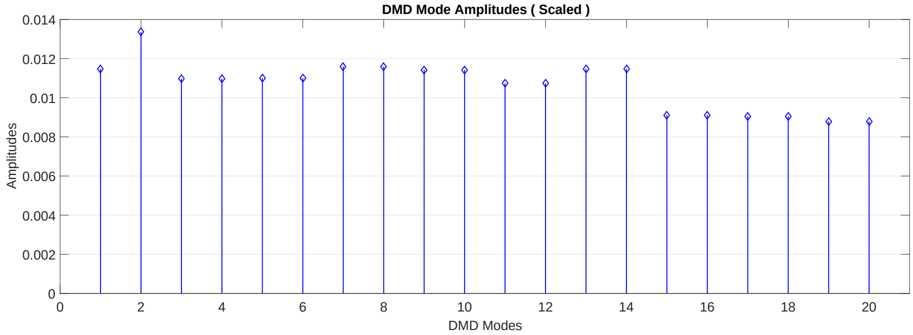

Figure 3.2: The amplitudes of DMD modes.

In Figure 3.2 we see the DMD modes which are associated with the DMD eigenvalues in Figure 3.1. The eigenvalue number two is a growing eigenvalue and the associated DMD mode number two is a growing mode. From Figure 3.2, the second DMD mode has the largest amplitude. The amplitudes are the norms of the DMD modes and they are scaled by multiplying with the real part of the associated eigenvalue then divided by the number of components that each mode has. The amplitude of the mode two will increase as time passes and other mode amplitudes will decrease, thus they can be discarded for the calculation of equation 3.1. Figure 3.3 shows the components of the DMD mode number two.

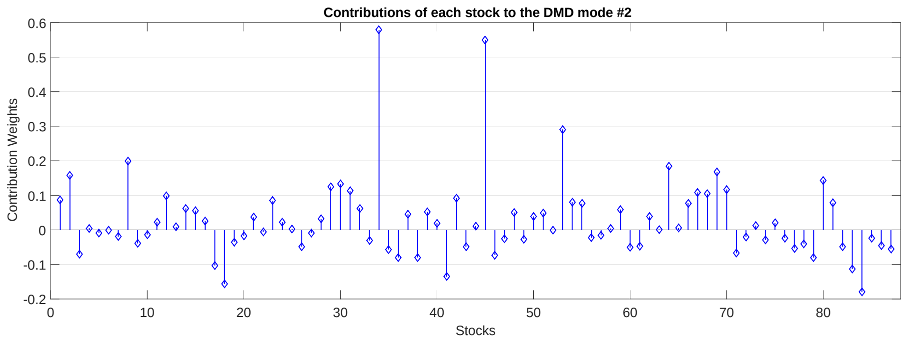

Figure 3.3: The components of the growing DMD mode.

Recall that the DMD modes are spatially coherent means that they are physically meaningful. Each component here represents a particular stock. (This is completely relevant to the way that how the data is gathered as a matrix). Each of them has different values on the dynamic modes, we call these values *contribution weights*. Now we know that the growing DMD mode scaled only by the growing eigenvalue then each stock contribution weights scaled as well. Since the growth rate of the growing eigenvalue is larger than one, the positive contribution weights remain positive and the negative contribution weights remain negative after scaling.

Thus, under two conditions we can say that the stocks which have the positive contribution on the growing DMD mode will increase in the future time horizon. The first condition is: the information of the DMD modes that we make the inference about the stocks future state should be meaningful. The second one is: the DMD approximation should be good. These conditions especially the first one, are closely related to the data used for the DMD analysis.

If we have growing DMD eigenvalue obtained from the given data then we conclude the market is feasible for trading. Then we select the stocks which have positive contribution weights on the growing DMD mode. Still, this method is not complete and we further improve it in this section.

If we get back to the conditions, we emphasized they are closely related to the data. Generally, if we use a good data measure, we should be able to get meaningful information from the DMD modes as we explained. Since they are spatially coherent, we should be able to know how strong is the contribution of the each spatial location to the current state of the dynamical system. The important thing here is that since the DMD method uses only the data, the measurement plays a critical role to have functional and accurate information. Recall that, in the Koopman framework, the observable functions are used to enrich this process but in the DMD framework we do not have that. Now, if we focus on the finance case; we have many data measurement options to choose such as price, volume, return, etc. Since we are speculating about stocks value in the future, price and return should be reasonable choice. Daily price data is used in different applications [8, 21]. But there is a problem related to our portfolio selection strategy if we use the price data. We observed that when price data is used, the expensive stocks have large positive contribution weights on the growing DMD mode. So, this leads to inaccurate portfolio selection strategy. Instead, we use return data. When return data is used, the mentioned problem is solved because all stock contribution weights are normalized. In this case, if a stock was performing well in the historical window that is the data used for the DMD analysis, that stock has higher contribution weight on the dynamic mode.

So far we built the outline of our implementation method of the DMD in finance but still there are several problems need to be considered. The DMD method uses some amount of data, in financial market case, tells us if we should enter the market or not. The problem here is how many days of daily data we should use to have an accurate signal. Another problem is when we enter the market, how many days we should keep our portfolio? The solution of the both problems is using an exhaustive search algorithm to find the best parameters for the DMD method. The idea is similar to the machine learning training and it will be discussed in Section 3.3.

We know which stocks to select into our portfolio from the DMD analysis but how we allocate our capital between the stocks? There could be several soluton to this problem. We can equally invest our capital in every stock. Another way could be investing according to the stocks contribution weights on the growing DMD mode. The rationale behind this idea is: since we use return data we assume that stock contributions on

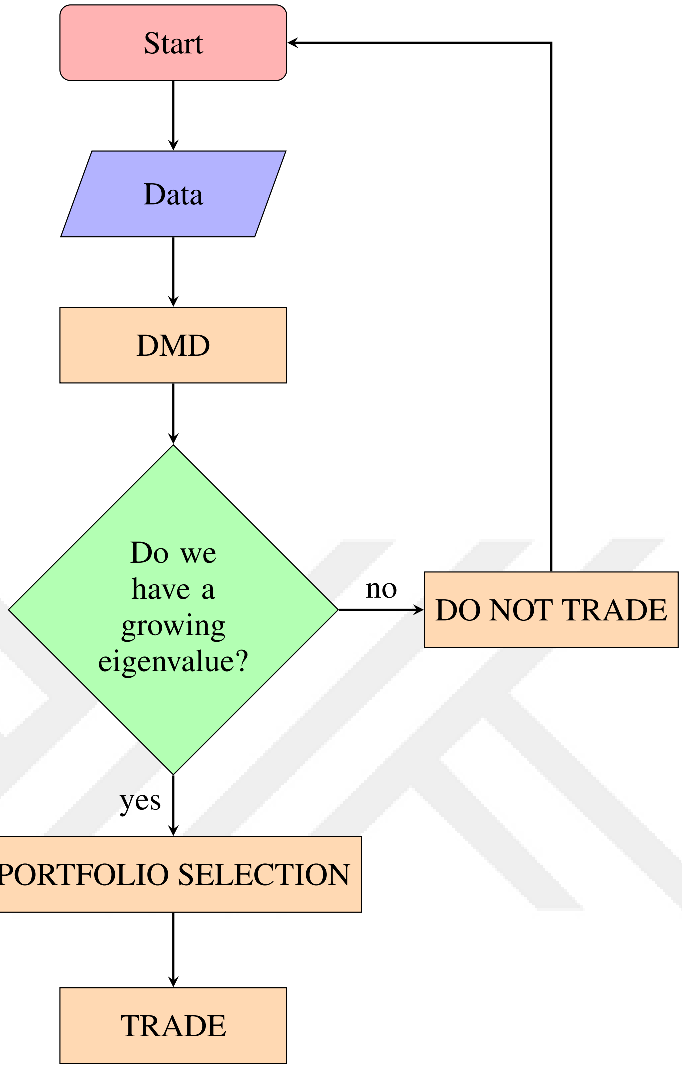

Figure 3.4: The flowchart of the DMD implementation in finance.

the DMD modes reflect the performances of the stocks. Simply put, the contribution weights show if market is increasing how much a particular stock is contributing to that trend. Because of this we should invest more in the stocks which have larger contribution weights. The equal allocation and the proposed way will be tested.

We finish this section by giving the flowchart of the implementation of the DMD in finance. The chart in Figure 3.4 explains the decision making steps of the DMD analysis for trading.

3.2 Data

In the previous chapters, we emphasized the importance of the data in the DMD method. In this section, we present the structure of the data that we use for the algorithmic trading strategy.

We use the daily stock returns of the stocks which are listed in the BIST 100 Index. The return data is obtained by observed price data using price2ret function of the MATLAB software. We used Google Finance as the data source [11].

The data contains of the daily returns of the available BIST 100 companies from January 2010 to Jan 2016 and for the DMD analysis, we put the data into the matrix as follows:

$$X = \begin{bmatrix} \dots & \vdots & \dots \\ \dots & 'AEFES' & \dots \\ \dots & 'AFYON' & \dots \\ \dots & 'AKBNK' & \dots \\ \dots & 'AKENR' & \dots \\ \dots & \vdots & \dots \end{bmatrix} \downarrow \quad \text{companies} \tag{3.3}$$

$$\rightarrow \quad \text{daily returns.}$$

By this way, each spatial location in the DMD mode is associated with a particular stock. Also, the oldest data is in the first column and the newest data is in the last column of the data matrix.

We split the data into two parts. The first part contains the data from January 2010 to January 2015 and this part will be used for the training of the algorithm. We refer to this data as training data. The second part contains the data from January 2015 to January 2016 which will be used for out-of-sample testing of the algorithmic trading strategy.

3.3 Preliminary Training Method

In the last section, we explained the way of implementation of the dynamic mode decomposition method in finance. In this section, we address that problem and define a method to deploy the algorithmic trading strategy that uses the DMD.

As we explained in Chapter 2, the DMD method takes some data and computes the DMD modes and eigenvalues. In Section 3.1, we defined a way of using these DMD modes and eigenvalues to speculate in a stock market. Our data consists daily returns of stocks. One of the problems was how many days of data should we feed into the DMD in order to get accurate DMD market timing signals? The other problem is what is our exit strategy? We use an exhaustive search algorithm to solve both of these problems.

Our exit strategy is to hold the portfolio for fixed time. Let us define two parameters (m, p), m represents the number of historical days and p represents the number prediction days. m indicates the historical data number that we feed into the DMD method. For example, if $m = 50$, this means that fifty days of the daily return data will be used for the DMD analysis. p is the number of days we should hold our portfolio. For example, if $p = 5$, this means we should hold our portfolio for five days. The goal is to find the best (m, p) pairing in the means of market timing prediction ability. The market timing prediction ability means that the success rate of prediction for a particular (m, p).

We use the approach proposed in [21] to find all success rate of predictions. We define an exhaustive search algorithm which evaluates the success rates of every (m, p) pairing over a data. The algorithm takes a (m, p) pairing and computes the DMD analysis over a data then takes the average of the success rates to find the total success rate of the pairing. It evaluates the success rate of every possible (m, p) pairing over the same data. The results are gathered into a matrix. We call this matrix *the success rate matrix*. Figure 3.5 shows the resulting success rate matrix for the training data.

In order to find the best pairing we evaluate the success rate matrix on the first part of our data which contains daily stock returns from Jan 2010 to Jan 2015.

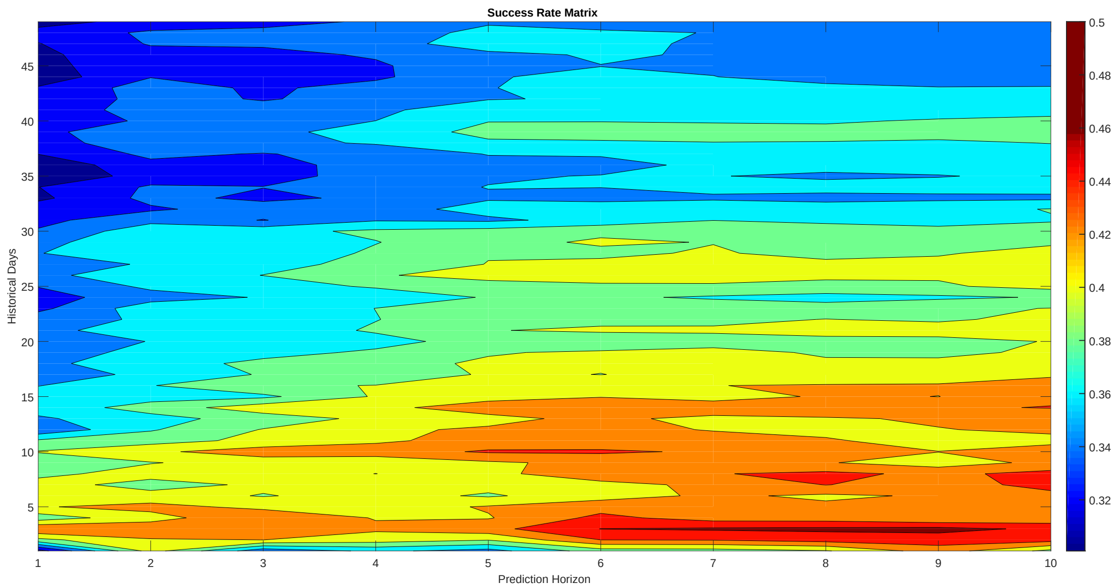

Figure 3.5: The success rate matrix.

The rows of this matrix represent the historical day number m and the columns represent the prediction day number p.

We chose to present this matrix as a filled contour plot because it easy to see which areas have high success rate. The red and orange areas are promising but it is not clear which (m, p) pairing we should pick. Again, there could be several approaches regarding picking the (m, p) pairing. One approach could be picking the pairing which has the highest success rate. It is possible that the highest success rate pairing could randomly occurred. Instead, we can pick with a principled way from an area which has high success rate. Areas have more than one pairings and this makes it more unlikely to occur as a random event.

The areas which have higher success rate in the matrix, we call them *trading hotspot*. The areas colored red and orange in Figure 3.5 are the trading hotspots. We will pick the best pairing from these hotspots. We use a modified form of the search algorithm proposed in [21] to find the best (m, p) pairing.

The main difference of our search algorithm from the one proposed in [21] is that it gives one (m, p) pairing regardless of the number of hotspots. Also, penalty value ensures that the pairing is the most successful pairing from the overall successful region and it is more unlikely to be occured randomly.

In order to have the best (m, p) pairing we define the search algorithm by fixing a success rate threshold. This threshold value can be selected as 0.5 which means half of the trades were accurate in the historical window. Only the pairings which are larger than this threshold are taken into the consideration. Then algorithm computes the penalty values of these pairings. The penalty value for a pairing from the success rate matrix is defined as follows:

$$P.V = \left(\sum_{k=i-1}^{i+1} \sum_{t=j-1}^{j+1} \left(sr_{ij} - sr_{kt} \right)^2 \right)^{1/2},$$

where sr_{ij} represents the pairing's success rate. Then the overall score of the pairing is calculated by substracting the success rate of the pairing from one and adding the calculated penalty value for the pairing. The pairing which has the minimum overall score is the pairing we use as the DMD parameter for the algorithmic trading strategy.

We use the penalty value in the calculation to promote the pairings which have close success rates with their surrounding pairings. If these pairings have close success rates it is more unlikely they occurred randomly altogether. In order to understand it better, we give the following simple flowchart of this search algorithm:

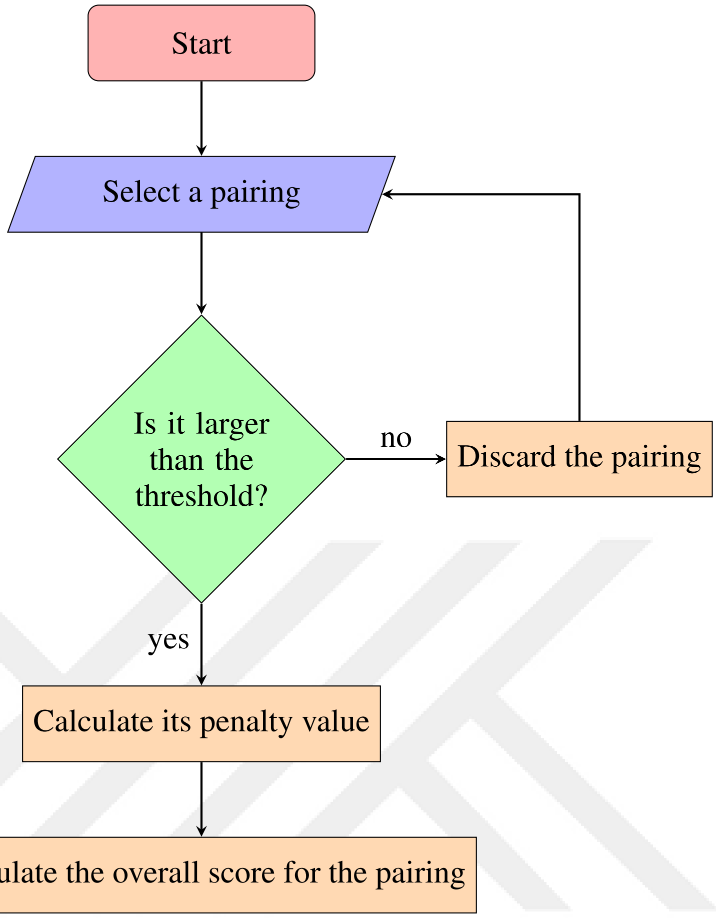

Figure 3.6: The flowchart of the hotspot selection algorithm.

We used the search algorithm on the first part of the data to determine the best DMD parameter for the algorithmic trading algorithm and it is found as (4,7). We will use this parameter for trading with the trading algorithm.

In summary in order to use the DMD in the algorithmic trading algorithm, we first implemented the ideas of the DMD and its usage in finance in section 3.1. From there, to find the required parameter we defined two algorithms in this Section. The first algorithm computes the DMD method for every possible (m, p) pairing and evaluates their success rates on a data then gathers all the information in the success rate matrix. The second algorithm which is explained in Figure 3.6 as a simple flowchart works on this matrix to find the best parameter to use for trading. For our case, this paramater is turned out to be (4,7). Now, we have everything to define our algorithmic trading strategy.

3.4 The Algorithmic Trading Strategy

In this section, we define our algorithmic trading strategy accordance with the information given in Sections 3.1 and 3.3. We would like to test different aspects of the implementation of the DMD in finance. In order to test them, we propose three strategies.

All trading strategies use the DMD method to give market timing signals and portfolio selection. As explained in Section 3.1, the strategies trade when a growing eigenvalue is spotted. Otherwise, they hold the capital and continue to compute the DMD analysis until a growing eigenvalue is spotted. If a growing eigenvalue is spotted, the strategies invest in the stocks that have positive contribution weights on the associated growing DMD mode. All strategies use (4,7) parameters for the DMD setting.

Strategy I: This strategy is the vanilla version of our final intent. It invests equally into the stocks which are indicated by the DMD analysis.

Strategy II: This strategy invests into the found stocks according to their contribution weights on the dominant DMD mode as explained in Section 3.1.

Using these two strategies, our aim is to test our portfolio allocation method which invests according to the contribution weights.

The dynamic mode decomposition uses data to construct the approximate linear dynamical system which is used to understand the dynamical system in hand. The success of this approximation is related to a couple of things as explained before. Another thing we need to consider is that the best-fit approximation holds only for locally. Simply put, it holds for only the historical window and we assume that the dynamics of the market do not change substantially afterward. If it changes, the DMD method fails to capture the dynamics. The financial markets are open, complex dynamical systems that are related to many things including politics. Since the DMD method does not use any input other than data, if exogenous effects exist it fails to capture the dynamics well and gives false trading signals. Improvements can be made to feed these exogenous effects into the DMD method but a simple solution can be using another tool as a complementary for the DMD method.

In order to minimize the number of false signals, we would like to use a widely accepted technical analysis tool as a complementary tool for the DMD method. *The simple moving average (SMA)* [23] is a versatile tool, it is used to identify current price trends and the potential for a change in the established trend. Detailed information about the simple moving average can be found in Appendix B. Our aim by using this tool is to prevent the DMD strategy giving out false trading signals. For example, consider there is a strong downtrend in the market prices and the DMD method gives buy signal. This signal can be caused by exogenous effect. In this situation, we do not trade and simply pass this signal. We trade if both tools give buy signal. So, we build our intended final strategy as follows:

Strategy III: This strategy also invests into the found stocks according to their contribution weights on the dominant DMD mode as explained in Section 3.1. In addition to that, we use the simple moving average strategy as a complementary tool for this strategy. We trade if both the DMD and the simple moving average give buy signals. Otherwise, the capital is held until the next signal.

Table 3.1 summarizes the algorithmic trading strategies.

Table 3.1: Algorithmic trading strategies.

	Market Timing	Stock Selection	Stock Weights
Strategy I	DMD	DMD	Equally
Strategy II	DMD	DMD	Proposed Asset Allo. Method
Strategy III	DMD + Optimized SMA	DMD	Proposed Asset Allo. Method

In order to use the SMA, we need two parameters. These parameters are called *lead* and *lag*. These parameters are important for having accurate signals from the SMA. In order to find these parameters we follow the strategy used to find the DMD parameters, see Section 3.3. We find the best-fit parameters for the training window and we use them for the testing as well. For this purpose, we use *genetic algorithm* to optimize the SMA on the training window. The genetic algorithm is a method to solve optimization problems using natural selection [29]. Details about the genetic algorithm and the optimization process about the SMA on the training data are given in Appendix C. The best-fit parameters are found to be one for the lead and twenty-one for the lag.

3.5 Testing the Strategies

In this section, we test the algorithmic trading strategies which are defined in the previous section. Testing is done on the second part of the data as out-of-sample testing. The test data contains daily stock data from Jan 2015 to Jan 2016.

We need to define several rules which all strategies will strictly follow in the test. The rules are defined as follows:

• The starting capital is 100,000 Turkish liras. No capital is added or removed during the test.

• Transaction costs are incurred for both buying and selling stocks.

• All capital is used when trading.

• The portfolios are held until the end of the prediction window and trading is not allowed before current portfolio is sold.

• All trading gains are reinvested.

All strategies are tested according to these rules. Also, we need benchmarks to compare our performance results with benchmark performances. We use BIST100 Index, interest rate and the optimized SMA strategy return rate over the test window as benchmarks for our trading strategies.

The interest rate is observed as 8.12% at the start of the test. Also, the final total return of the optimized simple moving average over the test period is 15.4%. Figure 3.7 shows the state of the BIST 100 Index over the test period.

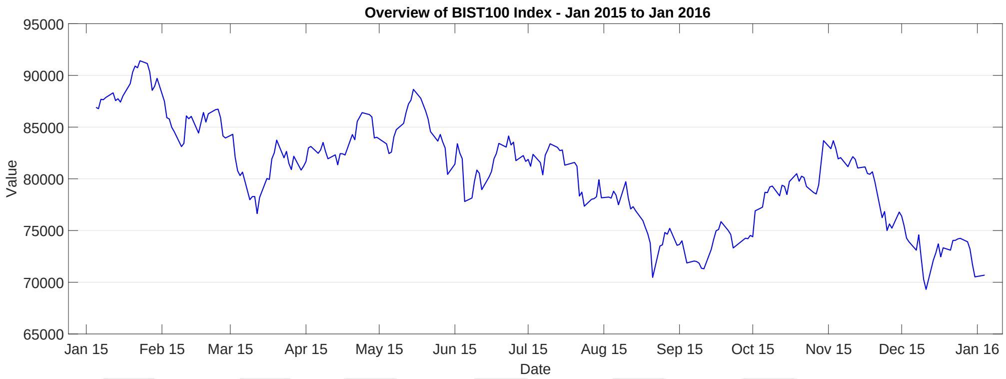

Figure 3.7: An overview of the BIST 100 Index over the test period.

In Figure 3.7 we observe that during the test period the BIST 100 Index was declined 21%. The year 2015 has seen as a rough year among investors according to the Exchange Trend Report-XIV published by Central Securities Depository of Turkey [25]. Serious fluctuations in the currencies and the markets have experienced due to heightened global growth concerns, Fed's rate hike expectations and rising geopolitical risks. All these factors led to a deterioration in global risk appetite and resulted in considerable outflows from Borsa Istanbul. During this period, Turkish lira also suffered and it was among the most fragile emerging market currencies and depreciated greatly against the US dollar. As a result total market capitilzation decreased 32% in US dollar terms and 12% in TL terms.

Next we look into the commission rates for trading in Turkey to define our transaction costs for the test.

3.5.1 Transaction Costs

We should take transaction costs into consideration. In Turkey, there are companies and banks which provide brokerage service to the public. They have different commission rate levels to different trading volumes. For example, one brokerage company applies 0.0018 commission rate to 0 - 50,000 TL trading volume and applies 0.0017 to 50,000-100,000 TL trading volume meaning that if we used this company as a broker and traded total 100,000 TL in volume, we would pay 175 TL commision for buying or selling orders. It is calculated by multiplying 50,000 with 0.0018, multiplying the

Table 3.2: Commission rates table (2017).

Companies	Com.R (0 - 50,000 TL)	Com.R (50,000 - 100,000 TL)
ATA Yatırım	0.0018	0.0017
Integral Yatırım	0.0005	0.0005
Sanko Yatırım	0.0007	0.0007
Garanti Yatırım	0.00185	0.00185
Deniz Yatırım	0.002	0.002
Ziraat Yatırım	0.002	0.002
İŞ Yatırım	0.0017	0.0017

other 50,000 with 0.0017 and adding them together. In Table 3.2, commission rates of the leading firms are given to have an idea about the current transaction costs.

These are the leading companies that provide brokerage services. The highest commission rate is 0.002. Therefore, one should expect to pay at most 200 TL commission for 100,000 TL trading volume which is our defined capital. Generally, commission rates decrease when total trading volume increases but we assume that commission rate is 0.002 constant for all trading volumes.

3.5.2 Performance of the Strategy I

The strategy I uses the DMD analysis for the market timing strategy and portfolio selection. The portfolio allocation rule of the strategy I is investing equally in the selected stocks. Figure 3.8 shows the result of the test of the strategy I over the test period.

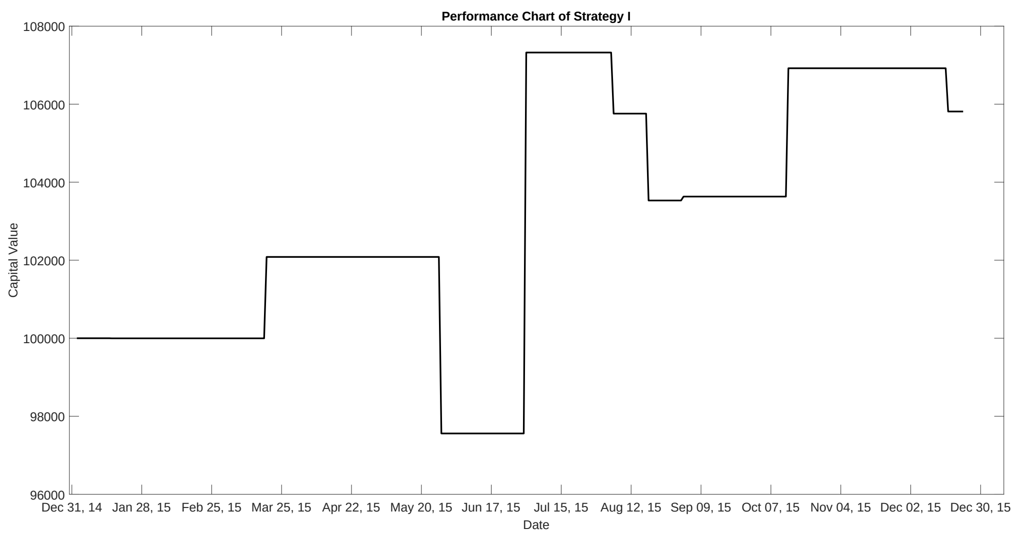

Figure 3.8: Strategy I perfomance result.

The ups and downs in the figure indicates the dates which strategy sold the portfolios. The reason we have this particular shape is that the strategies do not trade continuously. Recall from Section 3.1, the strategies only trade when a growing eigenvalue is spotted.

The strategy I made 5.81% return in the test period. In the next table we compare strategy I return with the benchmarks.

Table 3.3: Performance table I

Investment Options	Annualized Returns
Strategy I	5.81 %
Simple Mov. Avg.	15.4 %
BIST 100 Index	-21 %
Interest Rate	8.12 %

3.5.3 Performance of the Strategy II

As emphasized before, the strategy II tests the key element of our algorithmic trading strategy. This strategy allocates the portfolio of the selected stocks according to their contribution weights on the dominant DMD mode. Figure 3.9 shows the result of the test of the strategy II over the test period.

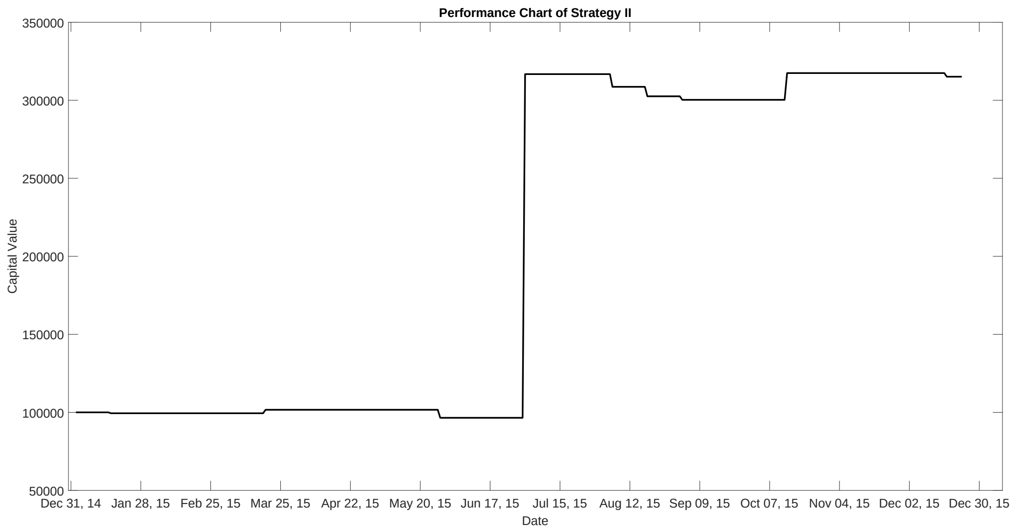

Figure 3.9: Strategy II performance result.

It seems remarkable that by allocating capital according to the weights of the stocks, total portfolio return from each trade is magnified. Daily return data is taken for the measurement gives the advantage of linking the overall performances of the stocks in the historical window with the weights of the stocks in the growing DMD mode.

If we examine Figure 3.9 we see that it has the same up and down pattern with 3.8 but it gained more profit from trades. They share the same settings but the only difference is their portfolio allocation rule. The strategy II made 215.11% return over the test period. This clearly shows that the proposed portfolio allocation rule is valid as expected. Table 3.4 shows the performance of the strategy II with the benchmarks and the strategy I.

Table 3.4: Performance table II

Investment Options	Annualized Returns
Strategy II	215.11 %
Strategy I	5.81 %
Simple Mov. Avg.	15.4 %
BIST 100 Index	-21 %
Interest Rate	8.12 %

3.5.4 Performance of the Strategy III

We can observe from the figures of the previous strategies, there are several false trading signals. The strategy III has complementary technical analysis tool which is optimized by using genetic algorithm over the training data to prevent false signals. This strategy has additional rule for trading other than spotting a growing eigenvalue. In order to trade the DMD analysis and the optimized SMA must give trade signal. Figure 3.10 shows the result of the test of the strategy III over the test period.

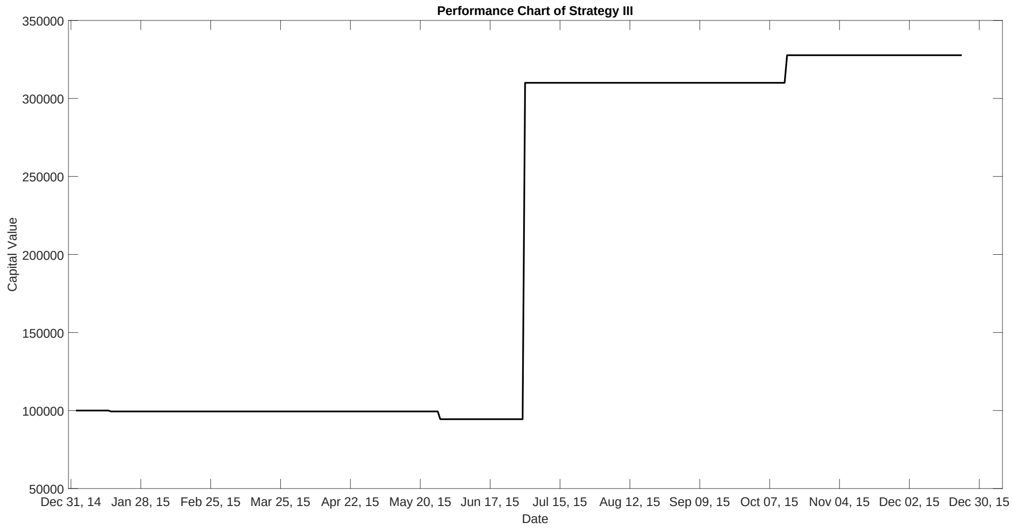

Figure 3.10: Strategy III perfomance result.

Figure 3.10 shows a different pattern than other strategy figures. This is because of the additional trading rule. The strategy I and II made total five false trades during the test period. But with the help of the optimized SMA, the strategy III made only one false trade during the test period meaning that complementary technical tool prevented 80% of the false trades. Also, preventing false trades and decreasing the total number of trades decreased the total transaction costs, resulted in a higher return than the strategy II.

The performances of all strategies are gathered in Table 3.5.

Table 3.5: Performance table III

Investment Options	Annualized Returns
Strategy III	227.72 %
Strategy II	215.11 %
Strategy I	5.81 %
Simple Mov. Avg.	15.4 %
BIST 100 Index	-21 %
Interest Rate	8.12 %

CHAPTER 4

CONCLUSION AND OUTLOOK

Our goal is to build a robust and profitable algorithmic trading strategy using dynamic mode decomposition. The main discussion is whether the dynamic mode decomposition is a capable tool to use as an alpha model for algorithmic trading strategies. Here, we summarize briefly the work and we try to answer that question.

In Chapter 2 we gave a detailed information about the DMD. It is a data analysis tool applied to different problems from different areas of science. The power of the DMD comes from its connection with the Koopman theory. We showed this connection and the differences between the Koopman analysis and the DMD. From there we understood that the success of extracting important information from the dynamical system in hand depends on a couple of important things. Therefore, the DMD has some limitations. The very first limitation comes from the definition of the DMD. The DMD approximates the dynamical system in hand by constructing a proxy linear dynamical system. The success of approximation depends on the data. Even though this approximation is good, this does not guarantee the accuracy of the predictions of the DMD analysis. Because past data is used to construct the approximation and it is best-fit approximation on that data. Also, as reported in the literature, the DMD method had a problem to capture standing waves in some applications. We addressed to this problem and gave a related example. Under these limitations, we tried to implement the DMD to finance.

The DMD analysis results finding dynamic modes and eigenvalues. As explained in Chapter 2 , these are the eigenvectors and the eigenvalues of the linear mapping that evolves snapshots in time. The useful aspects of the dynamic modes are that they are spatially coherent meaning that they are physically meaningful and each of the dynamic modes is associated with a temporal structure. In finance, we use both of these aspects to have a market timing strategy and portfolio selection strategy from the DMD analysis.

In the implementation, we used daily stock returns as the data for the dynamic mode decomposition analysis. This is rather subtle point. In this way, the dynamic modes became accurately meaningful. In order to utilize the DMD analysis completely, we proposed a new portfolio asset allocation method using the values of the spatial locations on the dynamic modes. Our aim was to invest more in stocks which we have a higher expectancy to rise.

In order to test the success of the DMD and the proposed portfolio asset allocation rule, we constructed several strategies. All strategies were capable of testing the market timing success of the DMD and two strategies were testing the success of the proposed portfolio asset allocation rule. As a result of the test, we see that the portfolio asset allocation method is valid and has serious implications for the return of the strategy. But we also observed that the dynamic mode decomposition analysis was not greatly successful to give accurate market timing signals. In order to improve it, we used a complementary technical analysis tool. The simple moving average tool was optimized with genetic algorithm to give the best-fit result for the market. As a result, the DMD analysis which is improved by optimized simple moving average was able to prevent 80% of the false trading signals.

As a conclusion, the DMD method is easy to implement and use for data analysis. We showed that it can be used as an alpha model for an algorithmic trading strategy. Also, it's success can be improved further with technical analysis tools.

The future work may be on improving market timing accuracy of the DMD analysis using different and several financial technical analysis tool. They may be optized together using genetic algorithm to have a collective market timing signal. In addition to that, exogenous effects of financial markets can be implemented into the DMD analysis directly using DMD with control mechanism.

REFERENCES

[1] R. Almgren and J. Lorenz, Bayesian adaptive trading with a daily cycle, The Journal of Trading, 1(4), pp. 38–46, 2006.

[2] M. Avellaneda and J.-H. Lee, Statistical arbitrage in the us equities market, Quantitative Finance, 10(7), pp. 761–782, 2010.

[3] R. J. Bianchi, M. E. Drew, and J. Polichronis, A test of momentum trading strategies in foreign exchange markets: Evidence from the g7, Global Business and Economics Review, 7(2-3), pp. 155–179, 2005.

[4] W. E. Boyce and R. C. DiPrima, *Elementary differential equations and boundary value problems*, Wiley, 2013.

[5] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz, Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition, Journal of Neuroscience Methods, 258, pp. 1–15, 2016.

[6] E. Chan, *Quantitative trading: how to build your own algorithmic trading business*, volume 430, John Wiley & Sons, 2009.

[7] K. K. Chen, J. H. Tu, and C. W. Rowley, Variants of dynamic mode decomposition: boundary condition, Koopman, and Fourier analyses, Journal of Nonlinear Science, 22(6), pp. 887–915, 2012.

[8] L.-x. Cui and W. Long, Trading strategy based on dynamic mode decomposition: Tested in Chinese stock market, Physica A: Statistical Mechanics and its Applications, 461, pp. 498–508, 2016.

[9] A. d'Aspremont, Identifying small mean-reverting portfolios, Quantitative Finance, 11(3), pp. 351–364, 2011.

[10] M. Fenton-O'Creevy, E. Soane, N. Nicholson, and P. Willman, Thinking, feeling and deciding: The influence of emotions on the decision making and performance of traders, Journal of Organizational Behavior, 32(8), pp. 1044–1061, 2011.

[11] G. Finance, https://www.google.com/finance.

[12] R. D. Harris and F. Yilmaz, A momentum trading strategy based on the low frequency component of the exchange rate, Journal of Banking & Finance, 33(9), pp. 1575–1585, 2009.

[13] J.-C. Hua, S. Roy, J. L. McCauley, and G. H. Gunaratne, Using dynamic mode decomposition to extract cyclic behavior in the stock market, Physica A: Statistical Mechanics and its Applications, 448, pp. 172–180, 2016.

[14] R. A. Jarrow, D. Lando, and S. M. Turnbull, A markov model for the term structure of credit risk spreads, The Review of Financial Studies, 10(2), pp. 481–523, 1997.

[15] J.-N. Juang and R. S. Pappa, An eigensystem realization algorithm for modal parameter identification and model reduction, Journal of Guidance, 8(5), pp. 620–627, 1985.

[16] P. J. Kaufman, *Trading systems and methods*, John Wiley & Sons, 2013.

[17] B. O. Koopman, Hamiltonian systems and transformation in Hilbert space, Proceedings of the National Academy of Sciences, 17(5), pp. 315–318, 1931.

[18] J. Kutz, *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*, OUP Oxford, 2013.

[19] J. Kutz, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, Society for Industrial and Applied Mathematics, 2016.

[20] A. W. Lo and A. C. MacKinlay, When are contrarian profits due to stock market overreaction?, The Review of Financial Studies, 3(2), pp. 175–205, 1990.

[21] J. Mann and J. N. Kutz, Dynamic mode decomposition for financial trading strategies, Quantitative Finance, 16(11), pp. 1643–1655, 2016.

[22] I. Mezić, Spectral properties of dynamical systems, model reduction and decompositions, Nonlinear Dynamics, 41(1), pp. 309–325, 2005.

[23] J. Murphy, *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*, New York Institute of Finance Series, New York Institute of Finance, 1999.

[24] R. Narang, *Inside the Black Box: A Simple Guide to Quantitative and High Frequency Trading*, Wiley Finance, Wiley, 2013.

[25] C. S. D. of Turkey, Bist trend report- xiv september-2015, 2015.

[26] S. J. Orfanidis, *Introduction to signal processing*, Prentice-Hall, Inc., 1995.

[27] R. Pardo, *The evaluation and optimization of trading strategies*, volume 314, John Wiley & Sons, 2011.

[28] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, Spectral analysis of nonlinear flows, Journal of Fluid Mechanics, 641, pp. 115–127, 2009.

[29] M. A. Ruggiero, *Cybernetic Trading Strategies: developing a profitable trading system with state-of-the-art technologies*, volume 68, John Wiley & Sons, 1997.

[30] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, Journal of Fluid Mechanics, 656, pp. 5–28, 2010.

[31] J. Sesterhenn and P. Schmid, Dynamical mode decomposition of numerical and experimental data, Bulletin of the American Physical Society, 2008.

[32] P. Shen, Market timing strategies that worked, The Journal of Portfolio Management, 29(2), pp. 57–68, 2003.

[33] L. Trefethen and D. Bau, *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, 1997.

[34] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, On dynamic mode decomposition: Theory and applications, Journal of Computational Dynamics, 1(2), pp. 391–421, 2014.

[35] M. O. Williams, I. G. Kevreki is, and C. W. Rowley, A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition, Journal of Nonlinear Science, 25(6), pp. 1307–1346, 2015.

[36] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, A kernel-based approach to data-driven Koopman spectral analysis, Journal of Computational Dynamics, 2(2), 2015.

APPENDIX A

A.1 Krylov Sequence and Subspaces

Given a matrix A and vector b, the associated *Krylov sequence* is the set of vectors b, Ab, A^2b, A^3b, ..., which can be computed in the form $b, Ab, A(Ab), A(A(Ab)), \ldots$. The corresponding *Krylov subspaces* are the spaces spanned by successively larger groups of these vectors [33]. It is named after Russian applied mathematician and naval engineer Alexei Krylov. The method used in modern iterative methods for finding eigenvalues or solving large linear equations.

A.2 QR Decomposition

Consider $A \in \mathbb{C}^{m \times n}$, $(m \geq n)$ has full rank n. The decomposition,

$$A = QR$$

is called 'reduced QR decomposition' where Q is $m \times n$ orthogonal matrix and R is $n \times n$ upper triangular matrix [33].

A full QR decomposition of $A \in \mathbb{C}^{m \times n}$ appends additional $m-n$ orthonormal columns to Q so that it becomes an $m \times m$ unitary matrix. In the process, rows of zeros are appended to R so that it becomes an $m \times n$ matrix, still upper triangular. All matrices have QR decompositions.

A.3 Singular Value Decomposition (SVD)

Consider $A \in \mathbb{C}^{m \times n}$, A has full rank n. The decomposition,

$$A = U \Sigma W^H$$

is called 'reduced singular value decomposition (SVD) [33] where U is $m \times n$ matrix with orthonormal columns, W is $n \times n$ matrix with orthonormal columns, Σ is $n \times n$ diagonal matrix with positive real entries and W^H is conjugate transpose of W, having orthonormal columns.

The definition can be extended to form 'full SVD' by adjoining additional $m - n$ orthonormal columns to U and $m - n$ rows of zeros below Σ.

Note that the diagonal matrix Σ has the same shape as A even when A is not square, but U and V are always square unitary matrices. Every matrix has singular value decomposition.

A.4 Hessenberg Matrix

Hessenberg matrix is a special square matrix. Upper Hessenberg matrix is a matrix with zeros below the first subdiagonal and lower Hessenberg matrix has zeros above the first subdiagonal [33]. Since many algorithms requires significantly less computational effort when applied to triangular matrices, reducing a matrix to triangular matrix is appropriate. If the reduction is restricted, it is best to reduce the matrix to Hessenberg matrix.

A.5 Arnoldi Iteration

Arnoldi iteration is a method to reduce a nonhermitian matrix to Hessenberg form by orthogonal similarity transformations. Reduction of $A \in \mathbb{C}^{m\times m}$, to Hessenberg form by an orhogonal similarity transformation can be written

$$A = QHQ^H,$$

where Q^H is conjugate transpose of Q unitary matrix and H is upper Hessenberg matrix. Arnoldi is an iterative method and m is usually huge, instead of full reduction consider $n < m$, first n columns of Q and let $\tilde{H}_n$ be the $(n+1)\times n$ upper left section of H, which is also a Hessenberg matrix. Then we have

$$AQ_n = Q_{n+1}\tilde{H}_n. \tag{A.1}$$

The nth column of this equation can be written as follows:

$$Aq_n = h_{1n}q_1 + \cdots + h_{n,n}q_n + h_{n+1,n}q_{n+1}. \tag{A.2}$$

From (A.2), it can be observed that the vectors $\{q_j\}$ form bases of the successive Krylov subspaces generated by A and b, defined as follows:

$$\mathcal{K}_n = \left\langle b, Ab, \ldots, A^{n-1}b \right\rangle = \langle q_1, q_1, \ldots, q_n \rangle \subseteq \mathbb{C}^m.$$

Since the vectors q_j are orthonormal, these are orthonormal bases. Thus Arnoldi process can be described as the systematic construction of orthonormal bases for successive Krylov subspaces [33]. Define the K_n to be the $m \times n$ Krylov matrix, then K_n has reduced QR decomposition

$$K_n = Q_n R_n,$$

where Q_n is the same matrix in (A.1). This is the Arnoldi iteration based upon the QR decomposition of the Krylov matrix.

A.6 Moore - Penrose Pseudoinverse

If matrix $A \in \mathbb{C}^{m \times n}$ has full rank then Moore - Penrose pseudoinverse or simply pseudoinverse of A denoted by $A^{\dagger}$ can be computed as follows:

$$A^{\dagger} = (A^H A)^{-1} A^H,$$

where A^H denotes the conjugate transpose of the matrix A [33].

A.7 Savitzky-Golay Filter

The Savitzky-Golay smoothing filters, also known as polynomial smoothing, or least-squares smoothing filters can preserve better the high-frequency content of the desired signal, at the expense of not removing as much noise as the averager filters [26]. This is achieved by a process called convulation, fitting successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares.

MATLAB has function called sgolayfilt which can be applied to a vector or matrix. For detailed information about this filter and its theoretical explanation refer to *Introduction to Signal Processing* by Sophocles J. Orfanidis [26].

APPENDIX B

B.1 Technical Analysis and Technical Indicators

In finance, technical analysis is a methodology for forecasting the prices of securities by studying the past market data, mostly price and volume [23].

A technical indicator is a mathematical calculation based on historical data such as price and volume, aims to forecast financial market direction.

B.2 Simple Moving Average and Simple MA Strategy

The *simple moving average* is an arithmetic moving average calculated by adding the closing price of the security for a number of time periods and then diving it by the total number of periods. It is one of the most versatile and widely used of all technical indicators. Because of the way that it is constructed and the fact that it can be so easily quantified and tested, it is the basis for many mechanical trend-following systems, see, e.g., [23].

The simple average is made up by two different price averages of the same security by using different time periods. They are often called *lead* and *lag*. Lead average's time period is smaller than the lag average's time period. It is important to select proper number of terms to determine price trends over specific time periods [16].

The *simple moving average strategy* is a trading strategy uses simple moving average. The common implementation is if the lead curve crosses the lag curve, the strategy gives buy signal. If the lag curve crosses the lead curve, the strategy gives the sell signal.

APPENDIX C

C.1 Genetic Algorithm

The genetic algorithm is a method uses natural selection ‘the survival of the fittest’, to solve optimization problems [29]. There are three main components of a genetic algorithm:

1. Describing the problem in terms of genetic code like chromosomes and genes.

2. A way to stimulate the evolution by creating offspring of the chromosomes.

3. A method to test the fitness of the each offspring.

In summary genetic algorithms works as the following: The initial step of solving an optimization problem using genetic algorithm is encode the problem into a string of numbers called a ’chromosome’. Each numerical value in this chromosome is called gene. The chromosome must contain all the information in order to solve the problem. The next step is initializing the population of chromosomes. Each gene for the chromosomes is randomly selected using valid values. The next step is evaluating each of the chromosomes using the fitness function. A fitness function evaluates chromosomes for their ability or fitness for solving a given problem. The next step is reproducing new chromosomes by mating two chromosomes. This is the heart of the genetic algorithm. This process involves two important steps: selecting a pair of chromosomes to use as parents and second, the process of combining these into children. There are several methods to achieve different children crossovers. Also, a low probability process called mutation sometimes happens in this step. Mutation is random changing of a gene on a chromosome. The next step is evaluating the new chromosomes. After this step, the chromosomes which have less fit than the new chromosome are deleted and new chromosomes are inserted into the population.

This process continues recursively until the limit for generations or time is reached. When the limit is reached, the algorithm picks the best-fit chromosome.

C.2 Optimization of the Simple Moving Average Strategy using Genetic Algorithm

In order to optimize the simple moving average strategy using genetic algorithm on the the data from Jan 2014 to Jan 2015 (this data is a part of the training data), we need to define the optimization problem in terms of genes and chromosomes. Also, we need to define a fitness function.

In order to trade with the simple moving average strategy we need lead and lag parameters. These parameters are the genes in a chromosome. The fitness function takes a chromosome, the parameters of lead and lag, then trades with this values according to the common implementation which is explained in Appendix B.2. At the final the fitness function gives out the sharpe ratio of the strategy and it is used for fitness value of the chromosome.

The goal is to maximize the sharpe ratio. If sharpe ratio is maximized, the chromosome which maximized it, is the best-fit parameters for the SMA strategy. Figure C.1 shows the result of this genetic algorithm.

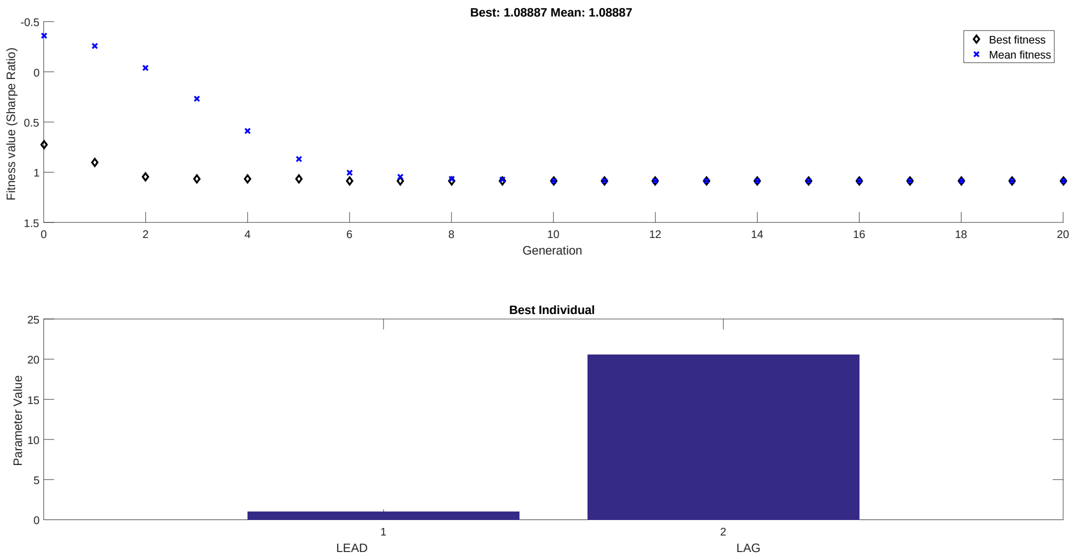

Figure C.1: The result of the genetic algorithm.

The best parameters in the optimization window is found to be (1,21) which gives 1.08 sharpe ratio. We use this parameter for the simple moving average which helps to the DMD method for the market timing decision.