ADVANCES IN QUANTITATIVE INVESTMENT WITH MACHINE LEARNING AND FINANCIAL NETWORK

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Weilong Guo

May 2020

ADVANCES IN QUANTITATIVE INVESTMENT WITH MACHINE LEARNING AND FINANCIAL NETWORK

Weilong Guo, Ph.D.

Cornell University 2020

Quantitative models are changing virtually every aspect of investment. In this thesis, we focus on the application of machine learning and financial network in investment. On the one hand, machine learning models can be used to detect complex patterns among financial data and make predictions about the market in the future. On the other hand, network science and topology facilitate the understanding of the structure that governs a complex system. Given the intricate and hierarchical nature of the financial market, it is vital to develop new network models for a better comprehension of its mechanism.

The rest of the thesis is organized as follows. In the first chapter, we construct a financial network among portfolios based on their common asset holdings and propose a cascade mechanism to explain how the linkage in the financial network can influence the portfolio returns. Then we apply the network structure in the design of a regularization function for a vector autoregression model, with the purpose to predict portfolio returns. In the second chapter, we devise a strategy to exploit arbitrage opportunities due to cascade behaviors among investors. The behaviors are detected with structural break tests while moving average methods are used to predict market directions. We then apply machine learning methods to intensify the strategy. Additionally, volatility prediction for stock return is another important topic in quantitative investment, as it is essential in various fields such as option pricing, portfolio allocation, and risk management of portfolios. In the third chapter, we propose a machine learning-based method for daily volatility prediction which outperforms existing methods in terms of various prediction errors.

BIOGRAPHICAL SKETCH

Weilong Guo started his PhD program in the Department of Operations Research and Information Engineering at Cornell University in August 2015. His concentration is in Applied Probability and Statistics, with two PhD minors in Finance and Artificial Intelligence. His research interest lies in the field of statistics, machine learning, financial network, and their application in quantitative investment, and he is co-author of several academic papers in the related fields. He also has several hand-on experiences in finance during the PhD study, including a semester-long visit to CFM Imperial Institute of Quantitative Finance in London, and a summer internship at Enlightenment Research in New York City.

This document is dedicated to all Cornell graduate students.

ACKNOWLEDGEMENTS

I would like to thank my wonderful PhD committee, Professor Andreea Minca (Chairperson), Professor Robert Jarrow (Finance) and Professor Bart Selman (Artificial Intelligence), for the great guidance of my thesis. First and foremost, I would like to express my deepest gratitude to Professor Minca for the guidance of my research and the generous share of knowledge and experience in the field of quantitative finance. My sincere thanks also go to Professor Jarrow for his instruction in asset pricing, the guidance of my project in portfolio optimization, and the suggestions on the construction of my career. Besides, I highly appreciate Professor Selman for his instruction in artificial intelligence and the great insight for the topics in the thesis.

Additionally, I would like to thank all of my fellow PhD students for the share of knowledge, and the department officers for their great job in coordination. Especially, I would like to thank Mr. Henry Lam and Mrs. Tara Woodard for the flexibility in room assignment. I wish them the best for the future.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vi
List of Tables . . . viii
List of Figures . . . ix

1 Financial Network of Portfolios **1**
1.1 The Topology of Overlapping Portfolio Networks . . . 1
1.1.1 Introduction . . . 1
1.1.2 Weight and Degree of Portfolios . . . 4
1.1.3 Data Description . . . 7
1.1.4 Regression Analysis . . . 13
1.1.5 Topology of the Network . . . 16
1.1.6 Discussion and Future Work . . . 19
1.2 VARX Model with Network Regularization . . . 21
1.2.1 Introduction . . . 21
1.2.2 Theory of Network Regularization Method . . . 24
1.2.3 Empirical Results . . . 28
1.2.4 Discussion . . . 33
1.3 Exhibits for Chapter One . . . 34

2 Statistical Arbitrage in Exponential Patterns **46**
2.1 Market Timing with Backward SADF Test . . . 46
2.1.1 Introduction . . . 46
2.1.2 Theoretical Development . . . 49
2.1.3 Methodology . . . 51
2.1.4 Experiments . . . 54
2.1.5 Conclusion . . . 58
2.2 Machine Learning Method to Improve Market Timing . . . 60
2.2.1 Introduction . . . 60
2.2.2 Meta Strategy . . . 60
2.2.3 Experiments . . . 62
2.3 Exhibits for Chapter Two . . . 64

3 Volatility Prediction with Random Forest **72**
3.1 Introduction . . . 72
3.2 Methodology . . . 74
3.2.1 Classical Volatility Estimators . . . 74
3.2.2 Averaging . . . 76
3.2.3 Random Forest . . . 77
3.3 Data Preparation . . . 81
3.4 Experiments . . . 81

3.4.1 Validation and Testing . . . 81
3.4.2 Crisis and Non-Crisis Period . . . 83
3.5 Conclusions . . . 84
3.6 Exhibits for Chapter Three . . . 86

A Models for Chapter 1 **89**
A.1 Group lasso and sparse group lasso . . . 89
A.2 Existing penalty methods for VARX . . . 90

B Practical Issues **92**
B.1 Issues in Financial Data . . . 92
B.1.1 Missing Data . . . 93
B.1.2 Outliers . . . 93
B.1.3 Data Revision . . . 94
B.2 Interpretation of Machine Learning Models . . . 94
B.2.1 Interpretation of Features . . . 95
B.2.2 Interpretation of Complex Models . . . 96
B.3 Trading Constraints . . . 97

Bibliography **98**

LIST OF TABLES

1.1 Types of financial institutions in *13F* by type codes . . . 34
1.2 Category statistics in *13F* by institution type codes (2003Q1 - 2012Q3) . . . 34
1.3 17 keywords used in filtering . . . 34
1.4 Category "5" statistics after filtering by a list of keywords (2003Q1 - 2012Q3) 35
1.5 Quarterly data sets statistics (2003Q1 - 2012Q3) . . . 35
1.6 Quarterly stock number statistics (2003Q1 - 2012Q3) . . . 35
1.7 Maddala-Wu Stationarity test for panel data . . . 36
1.8 Comparisons with Carhart 4 factors . . . 36
1.9 Robustness Study . . . 37
1.10 Comparisons with Carhart 4 factors plus average market depth . . . 38
1.11 Regression from Q3 2007 to Q3 2008 . . . 39
1.12 Size of 20 largest clusters. . . . 42
1.13 Experiment results for different penalty functions. . . . 45
1.14 Comparison with Sparse Own/Other method . . . 45

2.1 Trading result with out-of-sample data . . . 64
2.2 Comparison of trading result without and with meta strategy using out-of-sample data . . . 68

3.1 Prediction Errors with Equal Weights: Validation Set . . . 86
3.2 Prediction Errors with Market Cap Weights: Validation Set . . . 87
3.3 Prediction Errors with Equal Weights: Testing Set . . . 87
3.4 Prediction Errors with Market Cap Weights: Testing Set . . . 87
3.5 Prediction Errors with Equal Weights: 2008 Financial Crisis . . . 87
3.6 Prediction Errors with Market Cap Weights: 2008 Financial Crisis . . . 88

LIST OF FIGURES

1.1 Return and In-degree . . . 35
1.2 Average degree centrality over time . . . 37
1.3 Out degree (number of outgoing weak links) . . . 39
1.4 In degree (number of incoming weak links) . . . 40
1.5 Degree (number of adjacent weak links) . . . 40
1.6 Clustering coefficient distribution . . . 41
1.7 Average clustering coefficient over time . . . 41
1.8 Community Structure by Greedy Algorithm 1 . . . 42
1.9 Community Structure by Greedy Algorithm 2 . . . 43
1.10 Plot of portfolio returns in quarters from 2003Q1 to 2012Q3 . . . 43
1.11 Dendrogram for hiearchical clustering . . . 44
1.12 Autocorrelation coefficients of four quarters for portfolio returns . . . 45

2.1 The BSADF statistic for Barclays with price and trading volume . . . 65
2.2 3D plots of trading returns for each dataset . . . 66
2.3 3D plots of holding period volatility for each dataset . . . 67
2.4 3D plots of win rate for each dataset . . . 68
2.5 3D plots of trading returns for Bank data without and with meta model . . . 69
2.6 3D plots of holding period volatility for Bank data without and with meta model . . . 69
2.7 3D plots of win rate for Bank data without and with meta model . . . 69
2.8 3D plots of trading returns for Tech data without and with meta model . . . 70
2.9 3D plots of holding period volatility for Tech data without and with meta model . . . 70
2.10 3D plots of win rate for Tech data without and with meta model . . . 70
2.11 Feature importance plots for meta models . . . 71

3.1 Averaged Volatility of Stocks in the Finance Sector with Equal Weights . . . 86
3.2 Averaged Volatility of Stocks in the Finance Sector with Market Cap Weights . . . 86

CHAPTER 1

FINANCIAL NETWORK OF PORTFOLIOS

1.1 The Topology of Overlapping Portfolio Networks

1.1.1 Introduction

One channel of contagion among financial institutions is through common asset holdings, and such linkages can cause bigger losses than direct balance sheet exposures. This mechanism can be described as follows: following some external shock, a financial institution is forced to liquidate a part of its assets. If those assets are illiquid, other institutions within the financial network may be affected due to the price impact of liquidations, [71, 26]. In [38], the authors analyze fire-sale spillovers of U.S. commercial banks that liquidate due to leverage targeting, and show that these spillovers could be one order of magnitude above the initial shock.

In [71], the author proposes a model to measure the linkages among portfolios, using the market depths of the constituents of those portfolios. A network representation emerges, in which nodes are portfolios and edge weights represent the bilateral losses imposed due to the price impact of liquidations. Using data on mutual funds holdings, network-based measures such as vulnerability are shown to be correlated with returns. The returns during crises suffer from network effects. The propagation of distress when liquidations are induced by leverage targeting is also studied from a theoretical standpoint in [20]. Using a generalized branching process approximation for contagion, they find that the system is stable when leverage is under a threshold. This is consistent with theoretical studies on phase transitions for contagion in networks of balance sheet exposures, see [7, 6].

Given that networks of common asset holdings can propagate losses, it is critical to understand the topology of these networks. The fact that topology is a key determinant of cascading behavior is known for many types of networks, in particular social networks, e.g. [78, 40] and the references therein. Within the financial contagion context, several works have studied the topology of the interbank network, where the linkages are the balance sheet exposures. [15] analyze the network of exposures among Austrian banks. [72] look at the global banking system network using cross-border lending data. Most of these studies, e.g. [60, 11, 24], analyze the degree distribution and clustering in these networks and find that they exhibit some properties of scale free networks.

Despite the diversity of previous works on bank network analysis, research on portfolio networks, especially using hedge fund holdings, is very scarce. One of the reasons is that hedge funds are not obliged to report their positions, and their portfolio holdings have only recently become available. Our method, to the best of our knowledge, is the first one to explore the linkages among hedge fund portfolios induced by common asset holdings. Like any channel of contagion, it critically depends on the behavior of network participants in response to a received shock. In our case, the relevant behavior is represented by the propensity to liquidate. This is driven in the case of actively managed portfolios by the performance-outflow relation. The empirical literature on flow-performance relations for actively managed portfolios documents that there is a convex flow-performance relation for mutual-funds, and a concave flow-performance relation for hedge funds. While investors in hedge funds do not immediately withdraw following an initial shock, hedge fund investors withdraw three times as fast, see [12] and the references therein.

Our goal in this paper is to analyze the topology of the fund networks induced by common asset holdings. In particular we focus on the sub-graph of weak links, where there is a weak link between a node A to node B if liquidation by node A is higher than a

threshold relative to the size of B so that it causes B to also liquidate. We expect that this threshold is higher for hedge funds than for mutual funds, and thus we focus on hedge funds' portfolios.

An important data limitation is that only the long positions of hedge funds are available. Our network is thus a network of the long portfolios held by hedge funds. We will validate the network representation by showing that the returns of these long portfolios are affected through the network of common asset holdings. There are, of course, other factors that affect the returns of hedge funds, in particular their short portfolios and for this reason we do not use in the regressions the returns of the hedge funds that own the long portfolios, but only the returns of the long portfolios. If the same firm reports several portfolios, then we include the portfolios as separate nodes in the network. Indeed, these portfolios may have different sets of investors, whose outflow in response to the performance of the portfolio they hold induces the liquidations.

In [87], the author investigates the impact of liquidation on hedge fund returns by adding the Sadka liquidation factor to the benchmark model [47]. For equity funds, this benchmark model is the Carhart four factor model, which adds momentum factor (MOM) into the Fama-French three factor model [45]. In order to validate our weight attribution and network construction, we follow this path and add to the Carhart four factor model the degree of a node (in the sub-graph of weak links) as an additional factor. We control for the market liquidity and we show that this factor is significant. In this sense, we show here that while liquidity is a critical determinant of returns, so is the number of weak links induced by illiquidity. One difference between our paper and previous studies is that most of these studies perform the panel regression using classes of hedge funds (hedge funds are classified based on their investment style) and using the average returns over each class. Instead we include in the panel regression all hedge funds in the network. It will come out of the analysis that a large proportion of the funds concentrate in a few

large clusters. Within each cluster, funds use similar strategies and the fact that some of these strategies (or variations thereof) are used by a large proportion of the funds leads to contagion effects.

This section is organized as follows. In Section 1.1.2 we first construct a network among managed portfolios (mostly hedge funds) where each node represents a portfolio and each edge captures the connection between two portfolios due to common stock holdings. We next define directed weak links between portfolios. Section 1.1.3 describes the data. In Section 1.1.4, we test the stationarity of in-degree over time before using them in regression models to explain the excess fund returns. Carhart's four factor model is used as the benchmark model for the excess return of the funds and we examine the significance of the in-degree as an additional factor. In Section 1.1.5, we study the topology of managed portfolio network based on our weak links. We then identify the clusters in the network by modularity maximization method.

1.1.2 Weight and Degree of Portfolios

Weight between portfolios

A weighted link between portfolios is used to capture how connected they are. In this section, we use the weight attribution introduced by [70]. Consider a market with N portfolios and K stocks. The weight of the link between portfolio i and j is defined as

$$W_{ij} := \sum_{k=1}^{K} B_{ik} B_{jk} \frac{p_k}{\lambda_k} \tag{1.1}$$

where B_{ik} is the number of shares of stock k held by portfolio i, λ_k is the market depth (adimensional) and p_k is the price (in a numeraire, say dollars) of stock k. The weight is thus expressed in dollars. It is obvious that the higher the weight, the more connected the

two portfolios are, in the sense that the higher will be the impact due to liquidations.

If two portfolios do not have common holdings then, $W_{ij} = 0$. Also, if the market depth is infinite, then there is no liquidation impact even if the funds have the same strategies. The highest impact (and the highest weight) is achieved when two funds hold the same illiquid assets in their portfolios. It is important to notice that the weights are symmetric, i.e., $W_{ij} = W_{ji}$, so the impact is symmetric.

Weak links

We have so far considered the bilateral weights among portfolios. Contagion is determined by the relative weight with respect to the portfolio size. This is the idea leading us to define the weak links. Fixing a threshold θ, we say that there is a weak link between i and j if and only if

$$\frac{W_{ij}}{v_j} \geq \theta, \tag{1.2}$$

where

$$v_i := \sum_{k=1}^{K} B_{ik} p_k,$$

represents the size of portfolio i. The idea of weak links originates in social networks and is related to the adoption of a behavior or technology. The relation with our model is as follows. A portfolio has two choices of behavior, for some given $\epsilon > 0$:

1. To liquidate more than ϵ percent of its portfolio;
2. Not to liquidate at all or to liquidate less than ϵ percent its portfolio.

If i adopts behavior 1 then j suffers a percentage loss of at least

$$\frac{\epsilon \cdot W_{ij}}{v_j}.$$

Within the social networks literature, see e.g. [40], a link is said to be weak if it transmits behavior 1. It is well known in this literature that contagion goes through the weak links (the so-called "early adopters") and it is sufficient to study the structure of weak links in order to understand the cascading behavior of the network. We can now interpret the threshold θ and relate it to the behavior of a fund. By definition, a weak link between fund i and fund j leads j to liquidate more than ϵ whenever i liquidates more than ϵ. By 1.2, there is a weak link between i and j if and only if

$$\epsilon \frac{W_{ij}}{v_j} \geq \epsilon\theta, \tag{1.3}$$

This means that a fund will liquidate at least ϵ of its portfolio if and only if it suffers a loss of at least $\theta\epsilon$ on the asset side. So $\frac{1}{\theta}$ gives the ratio between the amount liquidated by the fund and the losses on the asset side. This is not due to leverage targeting as in the case of banks (studied by [38, 52]), but due to investor outflows, see [25]. Say that a fund liquidates fully its portfolio, i.e. $\epsilon = 1$. This full liquidation is triggered by a loss $\epsilon\theta = \theta$ on the asset side. It is reasonable to assume that $\theta \leq 1$. In this case, the full liquidation of the fund, driven for example by investor outflows, will occur before the asset side loses its value entirely. We will therefore assume $\theta \in [0, 1]$. The lower the θ, the faster will be the liquidations in response to losses on the asset side.

It is beyond the scope of our model to endogenize θ. In general, it would require a sophisticated equilibrium model for the investors, who in turn should take into account the network effects. Here we will simply test that even for θ constant these network effects are present. To do so, we will consider the network of weak links, which are the links that transmit shocks among portfolios. We let the adjacency matrix of the weak links

$$A_{ij} := \mathbf{1}_{\frac{W_{ij}}{v_j} > \theta}.$$

The presence of a weak link between i and j indicates that i has a notable impact on portfolio j.

Degree centrality

The centrality of nodes identifies which nodes in the graph are most influential. After we calculate the adjacency matrix, we calculate its row sum as out-degree and column sum as in-degree, and they are defined as

$$\text{Out-degree}_k = \sum_{j=1}^{N} A_{kj}$$

and

$$\text{In-degree}_k = \sum_{i=1}^{N} A_{ik}.$$

These are indicators of how much a portfolio can influence others and how much it is influenced by others respectively.

1.1.3 Data Description

We first use *Form 13F* to obtain quarterly portfolio holdings data, and then use daily stock data from CRSP US Stock Database to calculate quarterly stock statistics. Finally, we combine the portfolio holdings data and stock statistics to calculate portfolio returns and weights between portfolios that are defined in our model.

Quarterly portfolio holdings data from *13F*

Quarterly portfolio holdings data are extracted from *13F* provided by Thomson Reuters, which contains reports filed by institutional investment managers that exercise investment discretion over $100 million or more. To only consider portfolios under active management, we aim to select portfolios that belong to hedge funds in *13F*. Our sample is constructed for the period from 2003Q1 to 2012Q3, i.e. 39 quarters in total.

Thomson Reuters, the provider of our *13F* database, labels financial institutions with type codes, from 1 to 5, based on their types[1]. Table 1.1 shows the classification used in the database. We keep the entries whose type code equal ”5” as it includes all hedge funds. Table 1.2 shows the numbers of entries, where each entry records the number of shares of a stock in a portfolio, and the numbers of portfolios in all categories in *13F* from 2003Q1 to 2012Q3. It is clear that the majorities of entries and portfolios (to be precise, more than 70% of the entries and 65% of the portfolios) are in category ”5”.

It is also interesting that category ”1”, ”2” and ”3” all have comparable average number of entries per portfolio, six times larger than the average number of entries per portfolio for category ”4”. This means, on average, more positions in the portfolios held by banks, insurance companies and investment companies that report in *13F*. The average number of entries per portfolio of category ”5” is also three to four times smaller than for the first three categories. There could be of course a size effect, but also potentially lower diversification for those firms labeled with type code ”5”. Our focus in the current section is on hedge funds, which belong to category ”5”, given that hedge funds are the fastest to liquidate in response to asset shocks, which the main idea behind our definition of the weak links.

We could easily include banks, insurance companies and investment companies in our study, but they are unlikely to liquidate for the same reasons as hedge funds (banks liquidate due to leverage targeting) and they are also unlikely to liquidate *as fast* as hedge funds, so it is unclear that we could define the weak links with the same threshold. Given this, it is understood that the firms we include in the study could have weak links to firms that we exclude from the study, which means that we essentially have a subnetwork of the weak links.

Portfolios that do not belong to hedge funds, e.g. the ones reported by pension funds

[1]See Thomson Reuters Legacy Institutional 13F Holdings Data Feeds Specification

and university endowments, and those misclassified in the database are also included in category "5". Hence, we use a list of keywords to exclude those portfolios (shown in Table 1.3): if any part of the name of a financial institution contains one of the keywords, its entries in *13F* are not included in our sample. Table 1.4 shows that, after filtering, our sample still contains a significant amount of data ($> 88\%$ of the category 5 which includes the actively managed portfolios), and we have a higher confidence that it is a comprehensive sample of portfolios under active management, and selection bias is kept at minimum.

Our chosen subset of *13F* entries do not suffer from survivorship bias, because, in *13F*, portfolios are included in each quarter no matter whether they survive after the next quarter or not. Our method of selecting these entries does not modify this property of the data set. A well-known problem of using *13F* to extract holdings information is that short positions are unknown. Therefore, we only consider the long portfolios and the returns of these long portfolios.

We then partition our sample into 39 quarterly data sets. Table 1.5 shows that the sizes of the quarterly data sets are consistent across the quarters from 2003Q1 to 2012Q3. The distributions of the number of entries, the number of portfolios, the number of stocks[2] and entry-to-portfolio ratio all have coefficients of variation less than 9%. This means that all quarterly data sets contain similar numbers of portfolios and stocks, as well as similar average sizes of portfolios (entry-to-portfolio ratios). It further implies that no quarterly data sets in our sample need to be excluded or modified, since there are no outliers with extremely large or small sizes.

It is worth noting that the average number of portfolios in each quarter is 1861, which is around one half (55.11%) of the total number of portfolios that appear in any of the 39 quarters. One explanation is that some hedge funds became inactive during the period from 2003Q1 to 2012Q3, while some new funds joined the market. Another possible

[2]The method of calculating the number of stocks in each quarter is specified in Section 1.1.3.

reason is that the reporting threshold of $100 million may be so high that, on average, only one half of the hedge fund portfolios in our sample have assets under management (AUM) above the threshold in every quarter (and the other half pass below this threshold in some of the quarters). The fact that hedge funds sometimes seek confidential treatment by the Securities and Exchange Commission (SEC), who regulates the reporting of *13F*, also contributes to the absence of portfolios in the quarterly data sets. However, it is not only impossible to construct an inclusive sample with a consistent set of portfolios, but also unfavorable due to potential survivorship bias. Hence, the quarterly average number of portfolios in our sample, which covers more than 50% of the total number of portfolios, is appropriate.

Quarterly stock statistics from CRSP US Stock Database

Using daily stock data from CRSP US Stock Database, we compute market depth, λ_k, of each stock k in each quarter. The market depth of stock k is defined as the Amihud measure [5]

$$\lambda_k = \frac{ADV_k}{\tilde{\lambda} \cdot \sigma_k}, \tag{1.4}$$

where ADV_k is the average daily trading volume of stock k, and σ_k is the standard deviation of daily returns of stock k, and $\tilde{\lambda}$ is an invariant across stocks [Kyle and Obizhaeva, 2011]. It is fairly straightforward to obtain average daily trading volumes and daily return standard deviations. We first group daily volume and daily return data according to stock and quarter. Then, ADV is simply the average of daily trading volumes of each stock in each quarter, and σ_k is the standard deviation of daily returns. Finally, the market depth is calculated as in Equation 1.4.

Stocks prices at the end of each quarter are obtained from the database. For each stock k in each quarter q_0, we are interested in both the current price (stock price in the cur-

rent quarter), $P_k(q_0)$, and the future price (stock price in the next quarter), $P_k(q_1)$. If both $P_k(q_0)$ and $P_k(q_1)$ are available in the database, we include stock k in quarter q_0 stock data set; otherwise, we do not include stock k in quarter q_0. The reason is that we compute the returns of the portfolios using the returns of the stocks they own, and not the returns that the funds declare. In other words, the returns we consider are in fact the returns of the long equity part of the hedge funds' portfolios, which is reasonable given that our purpose is to demonstrate network effects due to liquidations of stocks. It is beyond the scope of this section to consider the overall return of the funds, especially those which are not long equity funds.

For the period from 2003Q1 to 2012Q3, quarterly stock market depths, current prices and future prices are extracted from CRSP US Stock Database. Out of 267468 stock data entries in the 39 quarters, more than 98% (262478) are valid with complete market depth, current price and future price information. From Table 1.6, we can see that the average number of stocks per quarter is 6730, and the numbers of stocks in the quarters are fairly consistent, in the range from 6542 to 6993.

Quarterly portfolio network weights and portfolio returns

Now that both portfolio holdings data and stock statistics data are ready, we can construct quarterly portfolio networks and compute portfolio returns using current and future stock prices. The first step is to match stock identifications in the stock statistics data to those of the stocks in the portfolio holdings data. Since the CUSIP and ticker of a stock could change over time, we use PERMNO (maintained by the CRSP US Stock Database) as the only stock identification, which is kept constant for a stock over time. From the CRSP US Stock Database, we create a mapping from CUSIP to PERMNO and another mapping from ticker to PERMNO, and then use the two mappings to translate CUSIP/ticker in the portfolio holdings data to PERMNO. Note that

several CUSIP/tickers could map to the same PERMNO, because a stock can have different CUSIP/tickers at different times but only one PERMNO at any time.

In every quarter from 2003Q1 to 2012Q3, more than 96% of the entries in our portfolio sample are correctly labeled with PERMNO. This means that our method of mapping improves data consistency and accuracy (by using PERMNO) without sacrificing a significant amount of information (by keeping more than 96% of the entries). Also, more than 96% of the stocks in the stock statistics data can be found in the portfolio holdings data in each quarter. This implies that our portfolio holdings data sample is comprehensive, not only in terms of portfolios but also in terms of stocks held in portfolios.

The portfolio holdings data in each quarter can be represented in a matrix B, where the rows correspond portfolios and columns correspond to stocks (PERMNO). Each entry B_{ij} means portfolio i holds B_{ij} shares of stock j. For example, there are 1544 portfolios and 6523 stocks in 2003Q1, and thus B for that quarter has a dimension of 1544×6523. Current prices, future prices and market depth values of all stocks in each quarter can be represented in column vectors p^c, p^f and λ. For example, for 2003Q1, p^c, p^f and λ all have the same length of 6523. Since edge weight between two portfolios, *i* and *j*, is defined as

$$W_{ij} = \sum_{k=1}^{K} B_{ik} B_{jk} \frac{p_k^c}{\lambda_k} \tag{1.5}$$

where K is the number of stocks in the quarter. We calculate the weight matrix W as

$$W = \left(B \odot \begin{bmatrix} (p^c \oslash \lambda)^\intercal \\ (p^c \oslash \lambda)^\intercal \\ \cdots \\ (p^c \oslash \lambda)^\intercal \end{bmatrix} \right) \cdot B^\intercal \tag{1.6}$$

where $\odot$ denotes component-wise multiplication and $\oslash$ denotes component-wise division. For each quarter, the values of the portfolios, v^c, is

$$v^c = B \cdot p^c \tag{1.7}$$

and the values of portfolios in the next quarter if holdings are kept unchanged, v^f, is

$$v^f = B \cdot p^f. \tag{1.8}$$

Then the return of the portfolio, r, is simply

$$r = \left(v^f - v^c\right)/v^c. \tag{1.9}$$

1.1.4 Regression Analysis

In this section, we add the degree centrality as a factor to explain excess returns. The benchmark model we use is the Carhart four factor model [22], where the four factors are the Fama-French three factors [45] and a momentum factor.[3] Explicitly, the Carhart four factor model can be written as

$$R_p - R_f = \alpha + mMkt.RF + sSMB + hHML + wMOM + \epsilon \tag{1.10}$$

The dependent variable on the left hand side is the excessive return, where R_p and R_f represent the portfolio return and the risk-free interest rate respectively. In the regressors, α (Jensen's alpha) stands for the extra return due to active portfolio management. $Mkt.RF$ measures the excessive return of the market portfolio over risk-free interest rate. SMB is the return on the diversified portfolio of small stocks minus the return on the diversified portfolio of big stocks; HML is the difference between the returns on diversified portfolios of high and low book-to-market value stocks [45]. The momentum factor, MOM, is the difference in return between portfolios with better return and those with worse return in the previous month. This factor is added into the Fama-French's model based on the assumption that the stocks which perform well in the previous term tend to continue their outperformance over a period of time. It is worth noting that the Carhart

[3]The description and the time series for these factors are available at `http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`

four factor model was originally developed for mutual funds, and used for equity hedge funds in [47] as part of the commonly used Fung-Hsieh factors.

Our goal in the current section is to demonstrate the significance of our degree centrality measure after we control for these well known factors. The threshold value for degrees are set to be

$$\theta = 0.033 \tag{1.11}$$

after several trials to balance between over-sparsity and noisiness. We later find that the regression results are robust when we allow theta to change from 0.028 to 0.04. While the coefficient of in-degree remains significant and positive for different theta, its magnitude becomes larger when theta is larger. However, the adjusted R square begins to drop when we move the theta far away from 0.33. The regression results are displayed in Table 1.9.

In time series analysis and panel data analysis, the non-stationarity of variables may cause spurious regression [51]. Therefore, before we use the degree centrality for regression, we test whether they are stationary over time. The test we utilize here was introduced by [69]. Only the portfolios with complete data are included in the test, which is sufficient to test the stationarity of the degrees. The result in Table 1.7 shows that both out-degree and in-degree centrality are stationary over time, and thus can be used in regression.

We now proceed to perform the panel regression. Considering that not all portfolios appear in all quarters, we only include those portfolios which appear in more than 60% of the quarters in our study. By doing this, we automatically exclude those portfolios that were created in the sample after the crisis (which represent 40% percent of the quarters).

The result of the Carhart four factor regression model is given in Table 1.8, where every factor is highly significant. We then test the significance of the in-degree (number of incoming weak links) and out-degree (number of outgoing weak links) as additional

factors. The result shows that in-degree passes the significance test while the out-degree does not. This is reasonable since the impact of a portfolio on other portfolios should not influence its own return. The positive sign of in-degree coefficient indicates that the common asset holdings with other portfolios have a positive effect on the return, on average over the entire period, from 2003 to 2012.

It was proposed by [87] to add a measure of liquidity as an additional factor to the the Fung-Hsieh seven factor model for hedge fund returns. Since liquidity, proxyed by market depth, is key in the definition of the weights in our model, we will need to control for it in order to demonstrate that our degree centrality does not merely capture the liquidity in the market. We define liquidity using the average market depth across stocks and weighted by their ownership and the individual stock the market depth is given in (1.4). When we add the liquidity factor, the in-degree is still significant and robust, as shown Table 1.10.

While averaged over the entire period network connectivity has positive effect on returns, in periods of mass liquidations (such as the crisis) the opposite is expected [71]. It is now accepted that while diversification reduces risk measured at the individual level, it can lead to overlapping assets and as a consequence it can also make systemic crisis more likely, see e.g. [98]. To this end, we conduct research using data during a crisis period. First we consider the third quarter of 2008, associated to the fall of Lehman Brothers. Figure 1.1 displays the in-degree against excessive return, with the fitted line. The negative relation between excessive return and our in-degree is evident. To investigate the relation between excess return and our in-degree during crises for a longer period, we filter our data and include only those quarters from the Quant crisis of August 2007 to the financial crisis of Fall 2008. The result of the panel regression is given in Table 1.11.

We notice that the in-degree is still significant, and the dependence is negative. It indicates that the negative relation between portfolio excess returns and their number of

weak in-coming links from other portfolio holds over a longer period, conditional on it being a crisis period. While vulnerability measures have been proposed before and are expected to be negatively correlated with returns, these results are stronger. They replace vulnerability (simply the sum of the weights relative to the size of the portfolio) with the number of in coming weak links. Recall that to define a weak link, one starts from given holdings, defines weights and specifies a threshold. If the threshold is too high then there are no weak links. If it is too low then the number of weak links becomes uninformative. The value used in the section for the threshold in the definition of the weak links was chosen by performing the regression for various values and choosing the minimal one such that below this value the in-degree is no longer significant. The notion of weak links in financial networks has been used previously in [7] for networks of balance sheet exposures, and explored from a theoretic standpoint in relation to contagion in inhomogenous financial networks in [8].

1.1.5 Topology of the Network

Average degree over time

We set the threshold to $\theta = 0.03$ as in the previous section, which makes the number of incoming weak links (in-degree) significant in the panel regression. To visualize its relation with return, in Figure 1.2 we plot average return against average in-degree across time. The two spikes of the average in-degree corresponds to the 2008 financial crisis and the downgrade of U.S. debt by S&P. During the financial crisis from Q3 2007 to Q3 2008 they are clearly negatively correlated, which agrees with our previous regression.

Degree distribution

Degree distribution is a common statistics of complex networks. Figures 1.3-1.5 are the log-log scatter plot of three degrees along with the fitting line marked in red. From Figures 1.3-1.5, it is obvious that the in-degree and all degree of the portfolio network follow a power law distribution, while the evidence in the in-degree plot is weaker due to the small range of data. That is, the fraction of nodes in the network having k connections to other nodes is proportional to $k^{-\gamma}$, where γ is a the scaling exponent whose value is typically $2 < \gamma < 3$ [36]. In all three cases we find two regions which can be fitted by a regression line, which is in close resemblance to [15]. Thus, we fit one regression line to the points with smaller degree and another regression line for the points with larger degree. The break points are initiated first by observation of the scatter plots, and then iterated to maximize the log likelihood function [75] [76] . The power decay exponents to the tails of the degree distributions are $\gamma_{out}^{tail} = 0.17$, $\gamma_{in}^{tail} = 4.32$ and $\gamma^{tail} = 0.15$. For the left parts of the distribution, we find $\gamma_{out}^{small} = 2.61$, $\gamma_{in}^{small} = 2.31$ and $\gamma^{small} = 3.02$. The break points are found to be $10^{1.05}, 10^{1.02}, 10^{1.77}$ respectively, and the adjusted R square are all above 0.95. Power law distributions for the degrees are a signature of complex networks resulting from human activity, and in the context of financial networks of loans they have been evidenced in [15, 24, 14].

Clustering coefficient distribution

In graph theory, the clustering coefficient measures the likelihood that two adjacent nodes to a node have linkages among themselves. We provide the histograms of cluster coefficient for four quarters: Q1-2003, Q3-2007, Q3-2008 and Q3-2011, in which we note that there is a significant number of portfolios with high clustering coefficient. We also note that the right tail of the clustering coefficient distribution thinned between Q1-2003 to

Q3-2007, while the quarters Q3-2008 and Q3-2011 have a noticeable increase in the right tail compared to the beginning of the crisis in 2007. In Figure 1.7 we plot also give the average clustering coefficient, which also captures the two spikes in the recent crisis.

Clusters

Many networks can naturally be divided into clusters, where the connections within each cluster are notably denser than those between clusters. It is especially important in a financial contagion study. Cluster detection has always been a challenging task in network science. The appropriate algorithm for graph clustering depends on the degree distribution. From the density estimation of degree within each quarter as well as the entire data set, we find that the networks constructed by our method are scale-free, which means that the degree of nodes follows the power law approximately. In [77], the author suggests using modularity as the measure of the quality of graph clustering. Modularity maximization method can be used for community detection in scale-free networks [36]. Thus, we use a fast greedy algorithm for modularity maximization implemented in the $igraph$ package in R for community detection. For simplicity we ignore the direction of the graph, assuming that portfolio i and portfolio j are connected if either $W_{ij} = 1$ or $W_{ji} = 1$. We accomplish this by letting $\tilde{A} = A + A^T$, and define the new adjacency matrix $\bar{A}$ as

$$\bar{A_{ij}} = \begin{cases} 1 & : \tilde{A}_{ij} > 0 \\ 0 & : \tilde{A}_{ij} = 0 \end{cases}$$

In Figure 1.8 and Figure 1.9, we display the four largest clusters in Q3-2007 and Q3-2008. This shows that 27% of portfolios belong to the largest four clusters in Q3-2007. The concentration is higher in Q3-2008, when the top four clusters include 36 percent of the portfolios. Portfolios that belong to different clusters have strategies that are essentially orthogonal: there are very few weak links between those funds and the funds in any of

the other clusters, and vice versa. This means that even if they have a part of their illiquid portfolios that overlaps, this represents a very small fraction of their entire portfolio so that it does not induce weak links. On the other hand, within each cluster, the overlap in illiquid assets is a significant fraction of the portfolios.

There are many other clusters, but much smaller, as well as isolated portfolios forming themselves a cluster. The isolated portfolios have strategies essentially orthogonal to the strategies used by the funds in the biggest clusters and are not affected by liquidations. From the analysis, we note that portfolios were more tightly clustered during the financial crisis period, which means that for many funds the overlap in their illiquid portfolio became a significant fraction of their portfolio. From Q3-2007 to Q3-2008, this effect is likely due to decreasing market depth of the stocks, while the condition for the existence of such large clusters is largely due to similarity in strategies before the crisis.

1.1.6 Discussion and Future Work

We find that the degree centrality is correlated with returns. For an individual portfolio, higher degree is associated with higher return in the long run, but may deteriorate the performance of the portfolio during financial crises. At the aggregate level, stronger connectivity among portfolios is linked with higher systemic risk. We also show that the distribution of degrees follow a power law. These results are similar with those found in [15] for networks of balance sheet exposures. Several extensions are left for further research.

1. Short positions

   Since hedge funds are not currently required to disclose short positions, we did not include them. However, the existence of short positions would not necessarily

mitigate the network exposures due to common asset holdings, but on the contrary even neutral funds could face large liquidations as evidenced by the quant crisis of 2007.

2. Threshold selection

   In this section, the threshold was fixed. However, in reality it is influenced by market depth, market capitalization, and the behavior of portfolio managers and investor redemptions. Therefore for a more accurate model, the threshold may be designed as a function of its influential factors.

3. Centrality

   The degree centrality is but one of the centrality measures available, and it may oversimplify the network by ignoring the indirect connections between portfolios. Katz centrality and eigenvalue centrality may be used to solve the problem. What is more, apart from setting a fixed threshold value, we may also define a distance between portfolios and calculate closeness and betweenness centrality as a measure of their influence in the network.

4. Clustering

   Our clustering method does not completely solve the problem in the sense that we ignore the direction of the graph. Generalized weight cut method may be used for cluster detection in directed graphs, but the optimal weighting has not been proposed, and to our best knowledge, the method has not been used for scale-free networks so far. Thus, more research should be done for an appropriate algorithm for cluster detection in directed scale-free graphs.

Though simple, our network model for overlapping portfolios of hedge funds and its empirical study points out the heterogenous topology of this network, widely known to be associated with large scale instabilities even in presence of small shocks.

1.2 VARX Model with Network Regularization

1.2.1 Introduction

Sufficient empirical evidence has shown that stock returns have serial dependence, which can be modeled by the vector autoregressive (VAR) model. [57] studies the predictability of asset returns, and [58] finds momentum in asset returns and suggests that strategies which buy stocks with good historical performance and sell stocks with bad historical performance generate significant positive returns over three to twelve months holding periods. In addition, [35] shows that abnormal excess returns of portfolios can be generated with strategies that exploit the serial dependence of stock returns based on VAR model.

In finance, factor models are commonly used to explain excess returns of portfolios. For portfolios of stocks, the benchmark model is the Carhart four-factor model, which extends the Fama-French three factor model [45] by taking stock price momentum into consideration. The vector autoregression with exogenous variables (VARX) model allows us to take these factors into consideration.

Unfortunately, the dimensionality problem puts a limit on the applicability of the VARX model as the number of time series becomes large. The unrestricted VARX model, which assumes a linear relationship between a series' current value and its past value as well as those of other series, is generally overparametrized and computationally intractable. For instance, [13] points out that when it comes to monetary policy analysis, unrestricted VAR model can only employ six to eight time series to conserve the degree of freedom. However, larger dimension VARX models are preferable to smaller ones, as they may include series that can potentially increase the prediction power of the model.

Many attempts have been made to reduce the parameter space size of VAR model. A large class of methods applies Bayesian approach for parameter estimation. [64] uses Gaussian priors to formulate Bayesian VAR in the context of macroeconomic modeling, which was further developed into the so-called "Minnesota prior" in the Bayesian VAR literature. [10] apply a natural conjugate extension of the Minnesota prior to reduce the parameter space. The proposed method shrinks all VAR coefficients to zero except for those of the first lags of dependent variables, which are later shrunk either to one or zero based on their persistence. [61] shows that medium and large Bayesian VAR can forecast better than factor models.

In contrast, another group of strategies incorporates lasso-related regularizations into the VAR model. [54] proposes lasso-VAR, which reformulates the VAR model as a linear regression model. Similar approaches have been explored by [90], [92], [79]. In particular, [79] gives a comprehensive implementation of VARX models with regularizations. As [92] points out, despite the fact that lasso-related regularization methods ignore the temporal dependence of time series, which may result in a larger estimation error, the prediction errors represented by relative mean squared forecast error (RMSFE) are significantly reduced in each of their experiments. More recently, [32] design a two-stage strategy for model selection of sparse VAR model. In the first stage, non-zero coefficients are selected based on partial spectral coherence (PSC). In the second stage, the model is further reduced based on t-statistics. However, this model is computationally expensive as the number of lags as well as the number of non-zero coefficients are selected using Bayesian information criterium (BIC), thus not applicable under high dimensional settings. Also the coefficients of the non-zero parameters are estimated separately after the two-stage filtering.

Graph theory has been used to analyze complex systems in many fields. However, its application in the field of quantitative finance is relatively scarce. A notable exception

is [19] who are using return data series to construct realized networks among stocks.[21] propose the stochastic flow diagram, which can be used to visualize the financial market as a complex dynamic system and explain how the system will react after a shock in specific sectors. [34] introduces a tree-graph structure to represent the hierarchical structure of the stock market. Through tree clustering, the method avoids the numerical instability problem which appears in the Markowitz critical line algorithm for portfolio optimization. In parallel, there is a growing literature on systemic risk in networks of common asset holdings, see e.g., [17, 16, 88]. Additionally, [23] proposes a graph-guided method for regression using fused lasso introduced by [96], which penalizes the coefficient difference between nodes with an edge. However, despite its potential application in quantitative finance, the method is not formalized under time series settings, and does not differentiate diagonal elements and off diagonal elements in autoregressive coefficients.

The rest of the section is organized as follows. In Section 1.2.2 we first discuss the current VARX model with structured penalties. Then a fund network is constructed where each node represents a portfolio and each edge captures the connection between two portfolios due to common stock holdings. We later propose a two-stage method to reduce the dimension of the parameter space. In the first stage, we set a portion of coefficients of the AR parameters to zero based on the fund network induced by common asset holdings. In the next stage, the model selection and model estimation are performed at the same time based on group lasso methods. Section 1.1.3 describes the data. In Section 1.2.3, we construct smaller datasets of different degrees of homogeneity and test the out-of-sample performance of the method. Carhart's four factor model is used as exogenous variables for the excess return of the portfolios. Finally, in Section 1.2.4, we discuss the potential applications of this section and future work.

1.2.2 Theory of Network Regularization Method

VARX Model with Structured Penalties

The complete VARX model with full flexibility for portfolios can be represented as $VARX_{(k,m)}(p,s)$, allowing for p autoregressive and s exogenous lag terms for k portfolios and m exogenous variables respectively. It assumes a linear relationship between fund returns and their own lag terms with some exogenous explanatory variables:

$$y_t = \nu + \sum_{l=1}^{p} \Phi^{(l)} y_{t-l} + \sum_{j=1}^{s} \beta^{(j)} x_{t-j} + u_t, t = 1, 2, \cdots T$$

where y_t is a k dimensional vector representing the returns of k portfolios at time t; ν is the constant k dimensional intercept vector of the model; $\Phi^{(l)}$ denotes a $k \times k$ endogenous coefficient matrix; y_{t-l} is the l-th lag term of portfolio returns; $\beta^{(j)}$ is a $k \times m$ exogenous coefficient matrix; x_{t-j} is a m dimensional vector of exogenous variables, which is usually taken to be a factor in asset pricing models; u_t is a k dimensional independent, normally distributed white noise vector, and we assume that it follows a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \Sigma_u)$.

Under a low dimensional setting, where the number of coefficients to be estimated is relatively small, we can estimate Φ, β and ν by multivariate least squares:

$$\underset{\Phi,\beta,\nu}{\arg\min} \sum_{t=1}^{T} \|y_t - \nu - \sum_{l=1}^{p} \Phi^{(l)} y_{t-l} - \sum_{j=1}^{s} \beta^{(j)} x_{t-j}\|_F^2 \tag{1.12}$$

where $\|\|_F$ denotes the Frobenius norm, i.e. $\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$.

Note that the number of parameters in the model is $k(1 + kp + ms)$. Thus, it is obvious that the parameter space size grows quadratically with the number of portfolios, a phenomenon which is commonly referred to as a curse of dimensionality. In spite of the overparameterization issue, larger VARX model are often more preferable, as smaller models may exclude explanatory factors which are actually important in model fitting

and prediction. To this end, convex penalty functions are included in Equation 1.12. Consequently, the objective function to be minimized becomes:

$$\arg\min_{\Phi,\beta,\nu} \sum_{t=1}^{T} \|y_t - \nu - \sum_{l=1}^{p} \Phi^{(l)} y_{t-l} - \sum_{j=1}^{s} \beta^{(j)} x_{t-j}\|_F^2 + \lambda(\mathcal{P}_y(\Phi) + \mathcal{P}_x(\beta)) \tag{1.13}$$

where $\lambda \geq 0$ is the tuning parameter determined by cross validation; $\mathcal{P}_y(\Phi)$ is a convex penalty function for the coefficients of lag returns; $\mathcal{P}_x(\beta)$ is a convex penalty function for the coefficients of lag factors used to explain portfolio returns.

Various penalty functions have been proposed by [54], [92], [79], which adapt numerous regularization methods for parameter estimation. In the context of portfolio returns, however, more attention should be paid to the Own/Other lags penalty functions introduced by [92], which penalize diagonal and off-diagonal elements of autoregressive matrices in separate groups. This is because portfolio returns are more heavily influenced by their own lags then by the lags of other portfolios. The details of the Own/Other-related method are discussed in the appendix of the thesis.

Network Regularization Function

The Own/Other-related methods implemented by [79] adapt group lasso and sparse group lasso as penalty function $\mathcal{P}_y(\Phi)$ into the VARX model. In this section, we extend the basic Own/Other penalty by embedding an adjacency matrix of portfolios into it. The aggregated weight matrix at time T is defined as

$$\tilde{W}^T := \sum_{t=1}^{T} W^t exp\{-\alpha(T-t)/T\}.$$

W^t is the weight matrix at time t defined in section 1.1.2. The aggregated weight matrix is designed such that more recent weight matrices are assigned with larger weights,

while the older matrices are given less weights. The reason here is that we need to make prediction for the portfolio returns immediately after time T, thus the weight matrices that are closer to time T are presumably more important. The attenuation parameter α is decided by the user. Note that a larger α represents a greater penalty on the older weight matrices, and α needs to be larger when T is small in order to ensure a strong enough penalty on older matrices. Of course, the aggregation method here is by no means unique. The next step is to set M percent of off-diagonal elements of $\Phi^{(l)},\ l = 1, 2, \cdots p$ to zero while keeping the rest of them in the model together with the diagonal elements. To do this, we will utilize the aggregated weight matrix above, setting $\Phi_{ij}^{(l)}$ to zero if $\tilde{W}_{ij}^T$ is smaller than the M percent quantile of all off-diagonal elements in $\tilde{W}$. The two hyperparameters α and M can be decided by cross validation. That is, we can define an adjacency-like k by k matrix A, such that

$$A_{ij}^T = \begin{cases} 0 & \textit{if } i \neq j \textit{ and } \tilde{W}_{ij}^T < \tilde{W}_{\textit{off},M}^T \\ 1 & \textit{otherwise} \end{cases}$$

where $\tilde{W}_{\textit{off},M}^T$ is the M quantile of off-diagonal elements of the aggregated weight matrix. Then the new autoregression parameter matrices $\hat{\Phi}^l$ can be defined as:

$$\hat{\Phi}^{(l)} := \Phi^{(l)} \circ A^T \tag{1.14}$$

where "$\circ$" represents the Hadamard product of two matrices. The penalty function $\mathcal{P}_y(\hat{\Phi})$ is then designed as an extension of the Own/Other penalty:

$$\mathcal{P}_y(\hat{\Phi}) = \sqrt{k} \sum_{l=1}^{p} \|\hat{\Phi}_{on}^{(l)}\|_F + \sqrt{(1 - M/100)k(k-1)} \sum_{l=1}^{p} \|\hat{\Phi}_{off}^{(l)}\|_F \tag{1.15}$$

The $\hat{\Phi}_{on}^{(l)}$ and $\hat{\Phi}_{off}^{(l)}$ refer to the diagonal and off-diagonal elements in $\hat{\Phi}^{(l)}$respectively. In the sequel, we will refer to our penalty function as "Network Own/Other". For exoge-

nous coefficients, we will use the lag group penalty method, where coefficients of each factor in one time term are input into one group, i.e.,

$$\mathcal{P}_x(\beta) = \sqrt{k} \sum_{j=1}^{s} \sum_{i=1}^{m} \|\beta_{\cdot,i}^{(j)}\|_F. \tag{1.16}$$

Thus the new optimization problem becomes

$$\arg\min_{\hat{\Phi},\beta,\nu} \sum_{t=1}^{T} \|y_t - \nu - \sum_{l=1}^{p} \hat{\Phi}^{(l)} y_{t-l} - \sum_{j=1}^{s} \beta^{(j)} x_{t-j}\|_F^2 + \lambda(\mathcal{P}_y(\hat{\Phi}) + \mathcal{P}_x(\beta)). \tag{1.17}$$

Implementation

It is obvious that the new coefficient autoregressive matrices $\hat{\Phi}^{(l)}$ are no longer actual matrices after the new zero restriction was introduced on most of the off diagonal elements. However, the formulation as a group lasso problem does not change, as the parameter groups are vectorized before being input into the optimization algorithm. Moreover, as we shrink the size of off diagonal groups of each lag, whose size is of order k^2, by M percent, the time needed for the algorithm to converge will be reduced depending on the choice of M. As such, the two penalty terms are of the same order of magnitude for the parameters we consider. As the regularization methods are not scale invariant, the data must be normalized before being used for training.

1.2.3 Empirical Results

Results for Time Series of Different Homogeneities

In this section we compare the prediction performance on data sets of different homogeneities. Before we start the analysis, we remove the portfolios with incomplete data, which leaves us with 626 portfolios. There are several important reasons for this processing in spite of the possible survival bias it may suffer. First, the major performance criteria of our study is the out-of-sample prediction error, which requires that the portfolio return must be available for the last period of our study. Thus the portfolios which disappear during our study period must be excluded. More importantly, we are only interested in analyzing and then making investment in portfolios that have robust performances over time. This criterion removes the extinct portfolios and those established shortly before the end point of our study period. By contrast, the portfolios with complete data have survived from the two major financial crises despite the tremendous tides of redemption. Now the only problem is the exclusion of portfolios which may have constant performance but were established shortly after the starting point of our study. However, the 626 portfolios are themselves sufficiently diverse for constructing a combination of portfolios with good performance. Last but not the least, the VARXL model which is the basis of our method was recently introduced, and the complexity and novelty of the model restrain us from designing parameter estimation methods for missing data.

As the monthly data of stocks and Carhart four factors are available, we augment our return data to monthly frequency by estimating the portfolios monthly returns assuming that the stock positions remain constant during the quarters. This leaves us with a data set of similar dimension as in [79]. Another important advantage of data augmentation is that the time series become stationary after the procedure. By contrast, 53 out of 626 quarterly return series are not stationary based on ADF tests where the significance level

is set to be 0.05.

Figure 1.10 plots the return of the portfolios over the study period of 39 quarters. It is evident from the plot that the majority of portfolio returns follow a similar pattern. By contrast, there is also a significant amount of curves which are divergent from the principal pattern. This observation raises our interest in examining the performance of our model for data sets of different homogeneities.

Homogeneity here can be understood in the following sense. In the first place we cluster the return data of 626 portfolios by orthogonal wavelet decomposition. Clustering methods are commonly used when we need to deal with high dimensional time series [37], as it can remove the redundant informations within the time series and keep the computation tractable. It is well-known that high level noise is often embedded within financial data, which may affect our clustering. Thus to define a proper distance between two time series, we first decompose each series via discrete wavelet transformation (DWT).

The DWT method uses a scale function family and a wavelet function family to represent a discrete signal. Let $\phi_{j,k}(n)$ be the scale functions and $\psi_{j,k}(n)$ be the wavelet functions generated by $\phi(n)$ and $\psi(n)$ with

$$\begin{cases} \phi_{j,k}(n) & = 2^{\frac{j}{2}}\phi(2^j t - k) \\ \psi_{j,k}(n) & = 2^{\frac{j}{2}}\psi(2^j t - k) \end{cases}$$

The original signal can be reconstructed as:

$$f(n) = \frac{1}{\sqrt{N}}\sum_k W_\phi[j_0, k]\phi_{j_0,k}[n] + \frac{1}{\sqrt{N}}\sum_{j=j_0}^{\infty}\sum_k W_\psi[j,k]\psi_{j,k}[n]$$

$W_\phi[j_0, k]$ are called approximation coefficients at resolution level j_0 and $W_\psi[j,k]$ are called detailed coefficients. It is clear that the first term can be viewed as an approximation of the original signal and $W_\phi[j_0, k]$ can be used as features of a time series. We can thus define the distance between two time series as the Euclidean distance of their approximation

coefficient vectors. After a distance is defined, we can use hierarchical clustering method to construct the structure of the market.

Considering the size of our data set, we choose Haar wavelet transformation introduced in [50], a method which has a simple form and can achieve a complexity of $\mathcal{O}(mn)$. An automatic method to select resolution level, i.e., the dimensionality of features was introduced by [102], and implemented in the R package TSclust by [73]. Then we calculate the pairwise Euclidean distance between the selected features among the 626 portfolios and generate a distance matrix.

After that, we use the hierarchical clustering method with the distance defined above. According to the dendrogram we plot in Figure 1.11, $h = 0.2$ is a good cut for clustering. A lower value for h would give a larger number of clusters but their size would be smaller. We have chosen $h = 0.2$ by taking into consideration this tradeoff: we obtain 50 clusters and their size ranges from 252 to 1. The top 20 largest clusters are listed in Table 1.12.

The clustered data series can be used to construct fictitious portfolio series with higher or lower homogeneity. To construct a data set with higher homogeneity, we randomly select 20 portfolios within the largest cluster (i.e., Cluster 1 in Figure 1.11). By contrast, to build up another data set where the return series fluctuate differently, we sample 20 clusters out of the 50 total and then we sample one portfolio from each of them. Considering that short-term prediction of portfolio returns can satisfy most purposes in equity research, we only take the last three months of data as test set, and the data in the month prior to the test set as validation set, while the rest of the data is used for training.

As to the number of lag terms to be considered, we calculate the autocorrelation coefficients of quarterly returns for each portfolio with the in-sample data and display the result in Figure 1.12. It can be found that portfolio return is positively related to the return

in the previous quarter as the correlation is around 0.15. By contrast, the coefficient turns negative for the second lag term. This can be explained by the momentum and reversal effect of portfolio return respectively. The magnitude of coefficient then decreases to zero for farther lag terms. Thus we include returns of the previous six months in the model for prediction. The weight matrix of the last quarter is discarded when we calculate the aggregated weight matrix $\tilde{W}$ in order to avoid using information from the future.

The results of our experiments are listed in Table 1.13 where $\alpha = 5$ and $M = 90$ are selected by cross validation. These values correspond to the situation where the weight on the first W matrix is roughly 1% and each portfolio is connected with two others on average. The main criteria we take are the mean square forecast error (MSFE), and the running time of the model. To demonstrate the superiority of our method, we display our results denoted by Network Own/Other method in Table 1.13 in contrast to those from the existing relevant regularization methods, namely the Lag penalty and Own/Other penalty. The details of the existing relevant regularization methods are given in Appendix A. We also compare their relative prediction performances with the conditional mean method, which uses the in-sample mean as the predicted value for each individual series. When it comes to out-of-sample prediction result, it is clear that the Lag penalty has the worst performance in all situations, which is expected because of the coarseness of the penalty method. By contrast, the out-of-sample MSFE of our model is significantly reduced for both data types, indicating that the prediction power of the model is enhanced. Our method works specially well when the data is homogeneous, where the MSFE drops by 90% compared to the naive sample mean estimation.

Another meritorious point of our method is that it significantly reduces the time needed to run the model for most of the cases, as the group size of off-diagonal parameters is reduced by 90 percent. From the result of the experiment listed, our model is two times faster than Own/Other method and 20% faster than the Lag method when the data

is heterogeneous, where all timings were carried out with an Intel Core 3.40 GHz processor. However, the Lag method can run faster when the data is homogeneous despite its poor performance in out-of-sample prediction. The reason for the similar running time of the two methods is that while the model with Network Own/Other penalty has less off-diagonal elements to be estimated, the Lag method penalizes the diagonal and off-diagonal elements together in one group. Additionally, the Own/Other and Network Own/Other methods are slower for homogeneous data.

Our method is highly robust when we alter the two new hyper parameters M and α. It is important to notice that the combination of α and M is our first trial, which avoids the backtest overfitting problem. Results with less prediction error and shorter training time are encountered during our experiments for different combinations of α and M but are not listed here because we demonstrate the performance under the first choice.

Comparison with Sparse Own/Other method

Sparse Own/Other method introduced by [91] and [79] also allows for within-group sparsity. However, the method is much more computationally expensive than naive Own/Other method, as another cross validation needs to be implemented to find the optimal mixing parameter α.

Table 1.14 lists the prediction errors as well as the running times for the two methods. The machine we use is the same as mentioned in the previous experiment. It is clear that the Sparse Own/Other method cannot significantly reduce the out-of-sample MSFE in both data sets compared to naive Own/Other method. More importantly, the computational time for Sparse Own/Other method takes up to 100 times longer then ours, making it almost impractical for larger data sets.

1.2.4 Discussion

We introduce a novel penalty method for VARX model in the context of portfolio returns, which aggregates the information from the financial networks among portfolios. It bridges time series analysis and graph theory. Our method can significantly reduce the time needed for computation in most of our experiments. Also, our model achieves better prediction performances under all scenarios, indicating that the topology of portfolios due to overlapping asset holdings can be applied as a structure for the penalty function. It works especially well when the fund returns are homogeneous.

Several extensions are left for further research.

1. Parameter selection

   As we discussed in the previous section, the attenuation rate α and the sparsity parameter M are introduced as new parameters, and the criteria of choosing optimal values for the two parameters is not explored in this section. However, according to our experiments, the prediction result from the model is robust for different α and M.

2. Missing data

   As we discussed before, survival bias may appear if we only keep those funds that survived during our study period. Thus, parameter estimation methods of our model with missing data remain to be studied.

3. High frequency data

   The stock positions of portfolios we use have quarterly frequency. However, hedge funds are known to trade at a higher frequency, and it is very hard to acquire higher frequency data on the holdings, as the hedge funds are not required to disclose their positions more frequently. Also, higher frequency data poses a tougher challenge on our model, as the computation becomes much more expensive.

1.3 Exhibits for Chapter One

type code	type	examples
1	banks	Bank of America Corporation First Virginia Bank
2	insurance companies	Allstate Insurance Company Ameritas Life Insurance Corp
3	investment companies	Royal Trust Company Northwestern Mutual Invt
4	independent investment advisors	Creative Planning Convergent Wealth Advr
5	all others	Citadel Investment Grp, L.L.C. Soros Fund Management, L.L.C. Arizona State Retirement Sys Yale University

Table 1.1: Types of financial institutions in *13F* by type codes

type code	# of entries	(%)	# of portfolios	(%)	average # of entries per portfolio
1	2817202	10.47	181	3.13	15564.65
2	519528	1.93	30	0.52	17317.6
3	319822	1.19	19	0.33	16832.73
4	4243216	15.77	1740	30.07	2438.62
5	**19008717**	**70.64**	**3817**	**65.96**	**4980.01**
total	26908485	100.00	5787	100.00	1.00

Table 1.2: Category statistics in *13F* by institution type codes (2003Q1 - 2012Q3)

bank	bk	college	corp	fndn	foundation
global	holding	ins	pens	pub	ret
state	school	teachers	trust	univ	

Table 1.3: 17 keywords used in filtering

type code	# of entries	% of "5"	% of total	# of portfolios	% of "5"	% of total
5 (excluded)	3834512	20.17	14.25	440	11.53	7.60
5 (included)	**15174205**	**79.83**	**56.39**	**3377**	**88.47**	**58.35**
5	19008717	100.00	70.64	3817	100.00	65.96

Table 1.4: Category "5" statistics after filtering by a list of keywords (2003Q1 - 2012Q3)

	# of entries	# of portfolios	# of stocks	e-to-p ratio
average	389082	1861	6353	209.3
standard deviation	33513	156	312	11.1
high	455268	2092	6753	227.5
median	380837	1889	6423	209.7
low	309595	1508	5581	186.2
coefficient of variation	**8.61%**	**8.36%**	**4.91%**	**5.29%**

Table 1.5: Quarterly data sets statistics (2003Q1 - 2012Q3)

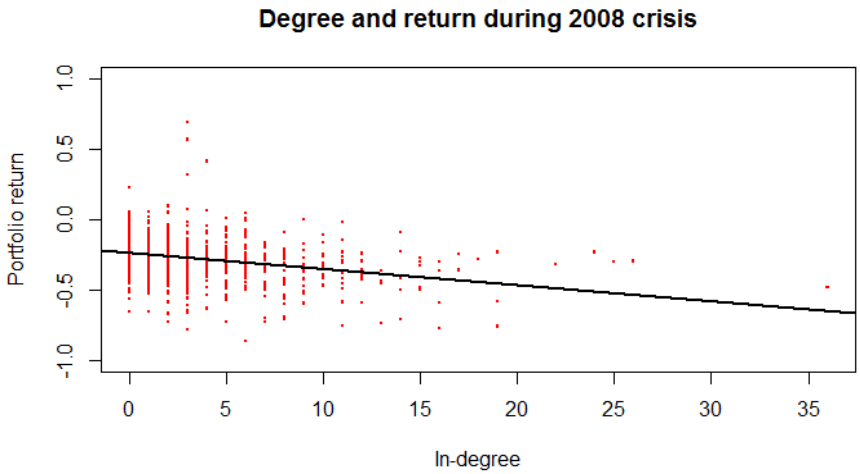


Figure 1.1: Return and In-degree

	# of stocks
average	6730
standard deviation	120
high	6993
median	6716
low	6542

Table 1.6: Quarterly stock number statistics (2003Q1 - 2012Q3)

Table 1.7: Maddala-Wu Stationarity test for panel data

	Chi-Square	Degree of Freedom	P-Value
Out-degree	6218.41	1252	2.20E-16
In-degree	8140.90	1252	2.20E-16

	Dependent variable:		
	Excess return		
	(1)	(2)	(3)
Mkt.RF	0.283^{***}	0.284^{***}	0.316^{***}
	(0.008)	(0.008)	(0.008)
SMB	-0.677^{***}	-0.677^{***}	-0.703^{***}
	(0.015)	(0.015)	(0.015)
HML	-0.856^{***}	-0.856^{***}	-0.826^{***}
	(0.018)	(0.018)	(0.018)
MOM	-0.390^{***}	-0.389^{***}	-0.381^{***}
	(0.006)	(0.006)	(0.006)
out_degree		0.00002	
		(0.00002)	
in_degree			0.008^{***}
			(0.0002)
Observations	176,361	176,361	176,361
R^2	0.051	0.051	0.058
Adjusted R^2	0.051	0.051	0.058
F Statistic	$2{,}348.527^{***}$	$1{,}879.211^{***}$	$2{,}154.455^{***}$
Note:			*p<0.1; **p<0.05; ***p<0.01

Table 1.8: Comparisons with Carhart 4 factors

Table 1.9: Robustness Study

	Dependent variable:		
	return2		
	$\theta = 0.28$	$\theta = 0.33$	$\theta = 0.4$
Mkt.RF	0.320***	0.317***	0.309***
	(0.008)	(0.008)	(0.008)
SMB	−0.706***	−0.703***	−0.699***
	(0.015)	(0.015)	(0.015)
HML	−0.836***	−0.826***	−0.824***
	(0.018)	(0.018)	(0.018)
MOM	−0.382***	−0.381***	−0.381***
	(0.006)	(0.006)	(0.006)
in_degree	0.0067***	0.0080***	0.0086***
	(0.0002)	(0.0002)	(0.0003)
Observations	176,361	176,361	176,361
R^2	0.057	0.058	0.057
Adjusted R^2	0.056	0.058	0.056
F Statistic (df = 5; 174602)	2,115.256***	2,156.681***	2,100.525***
Note:			*p<0.1; **p<0.05; ***p<0.01

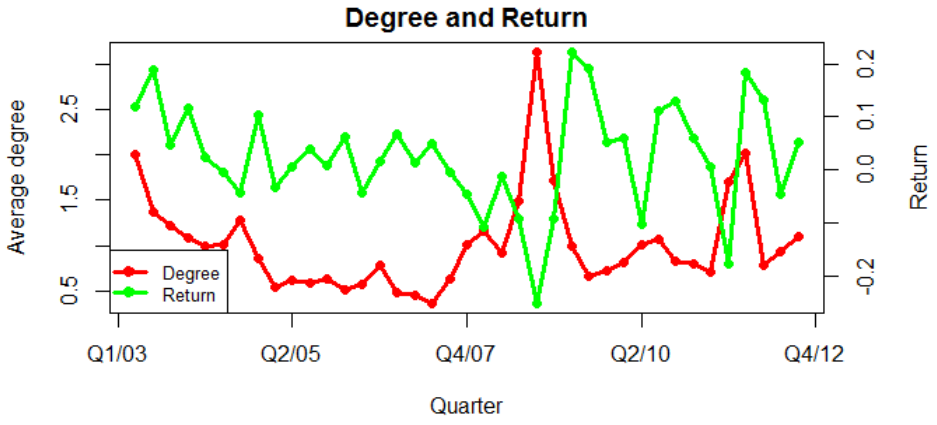


Figure 1.2: Average degree centrality over time

	Dependent variable:	
	Return	
	(1)	(2)
Mkt.RF	0.316^{***}	0.379^{***}
	(0.008)	(0.008)
SMB	-0.703^{***}	-0.745^{***}
	(0.015)	(0.015)
HML	-0.826^{***}	-0.825^{***}
	(0.018)	(0.018)
MOM	-0.381^{***}	-0.359^{***}
	(0.006)	(0.006)
Average market depth		-0.019^{***}
		(0.001)
in_degree	0.008^{***}	0.007^{***}
	(0.0002)	(0.0002)
Observations	176,361	176,361
R^2	0.058	0.062
Adjusted R^2	0.058	0.062
F Statistic	$2{,}154.455^{***}$	$1{,}933.308^{***}$
Note:	*p<0.1; **p<0.05; ***p<0.01	

Table 1.10: Comparisons with Carhart 4 factors plus average market depth

	Dependent variable:
	Excess return
Mkt.RF	0.038^{*}
	(0.023)
SMB	-1.701^{***}
	(0.032)
HML	-0.138^{**}
	(0.064)
MOM	0.306^{***}
	(0.017)
in_degree	-0.006^{***}
	(0.001)
Observations	22,064
R^2	0.197
Adjusted R^2	0.181
F Statistic	997.149^{***}
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 1.11: Regression from Q3 2007 to Q3 2008

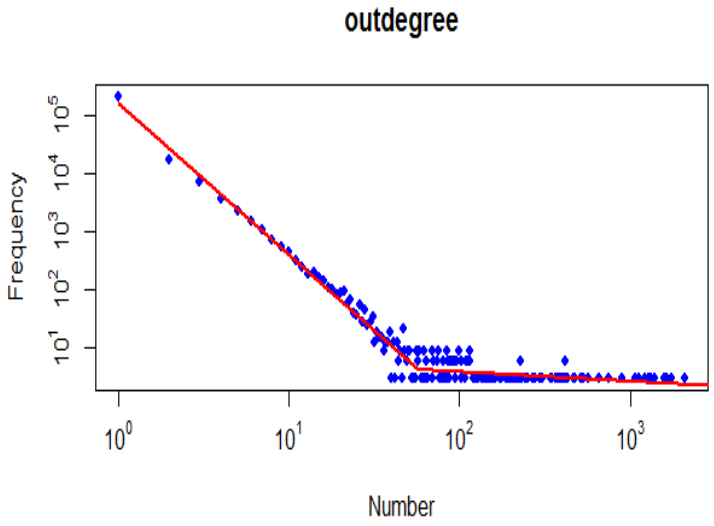


Figure 1.3: Out degree (number of outgoing weak links)

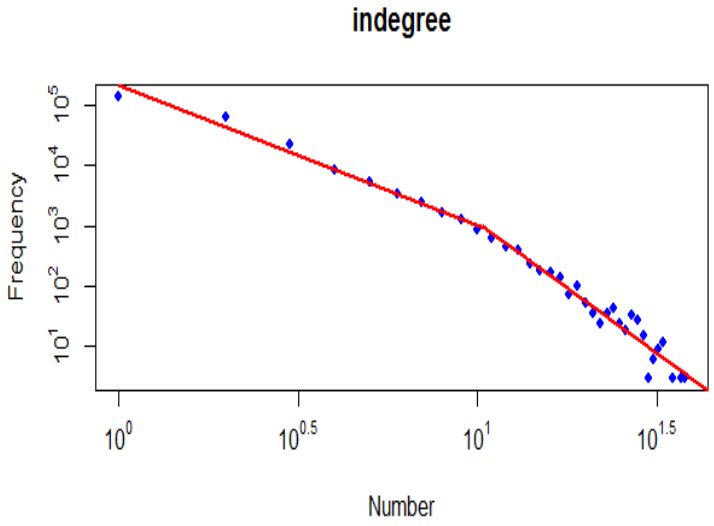


Figure 1.4: In degree (number of incoming weak links)

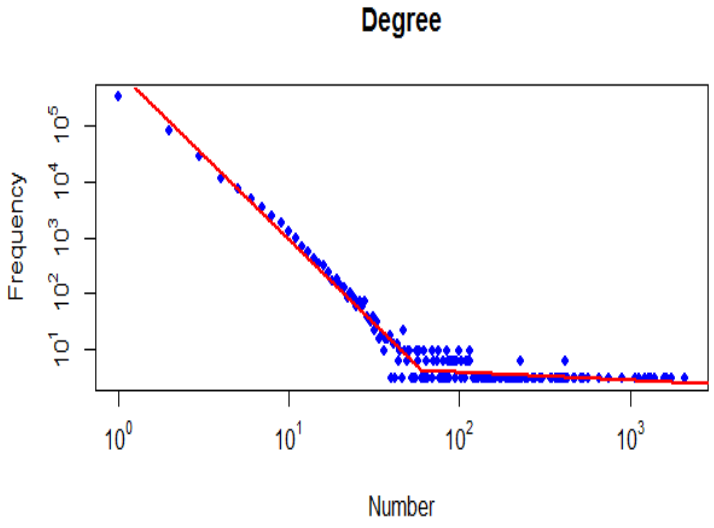


Figure 1.5: Degree (number of adjacent weak links)

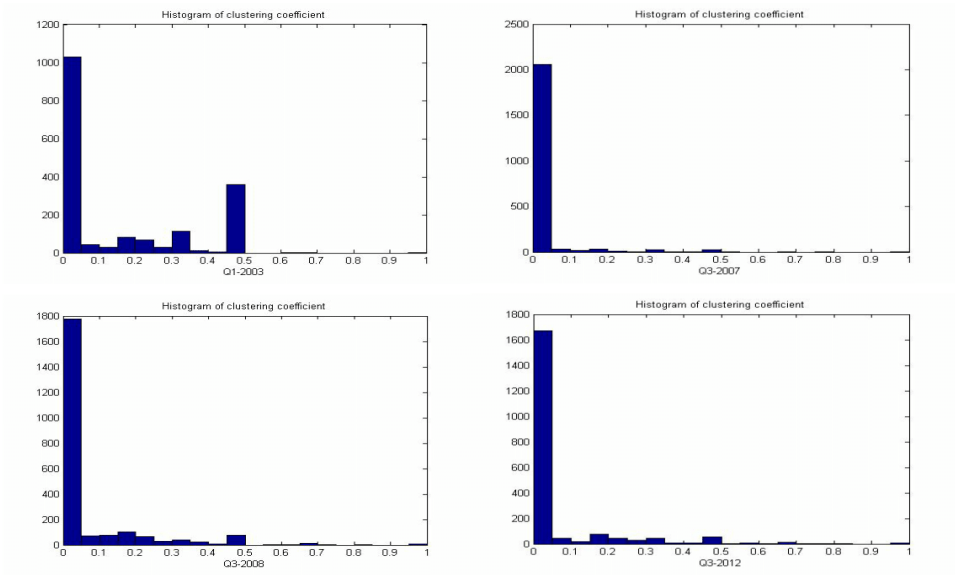


Figure 1.6: Clustering coefficient distribution

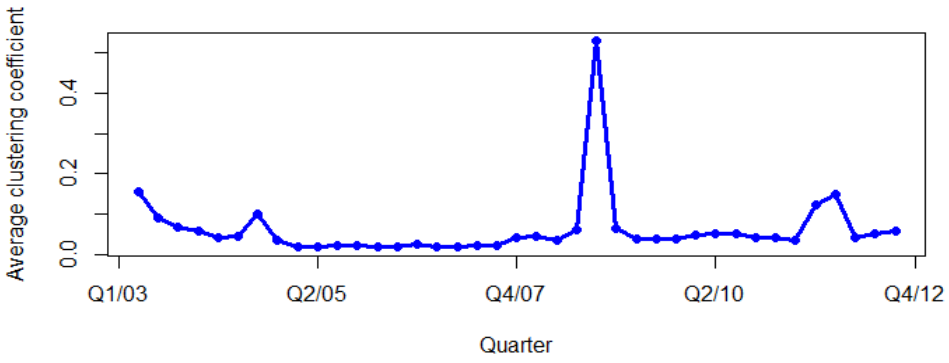


Figure 1.7: Average clustering coefficient over time

Figure 1.8: Community Structure by Greedy Algorithm 1

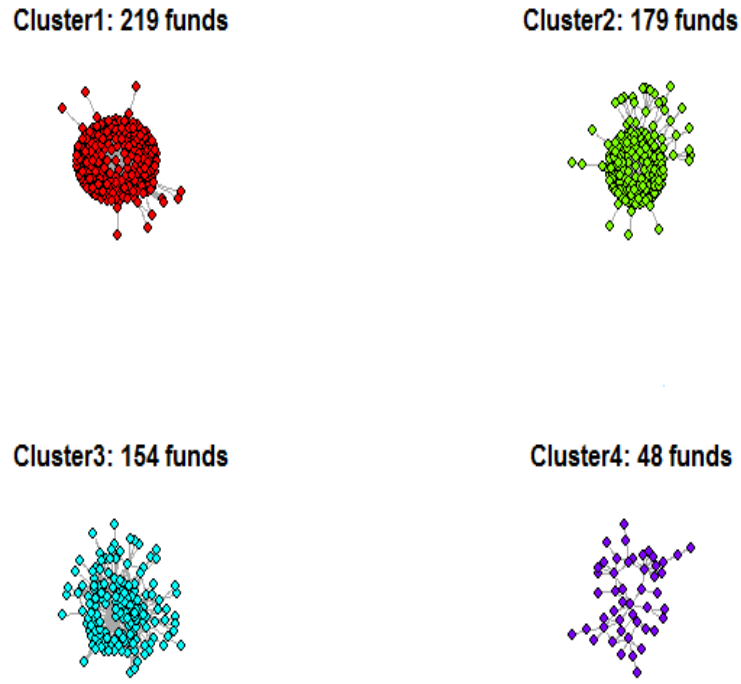
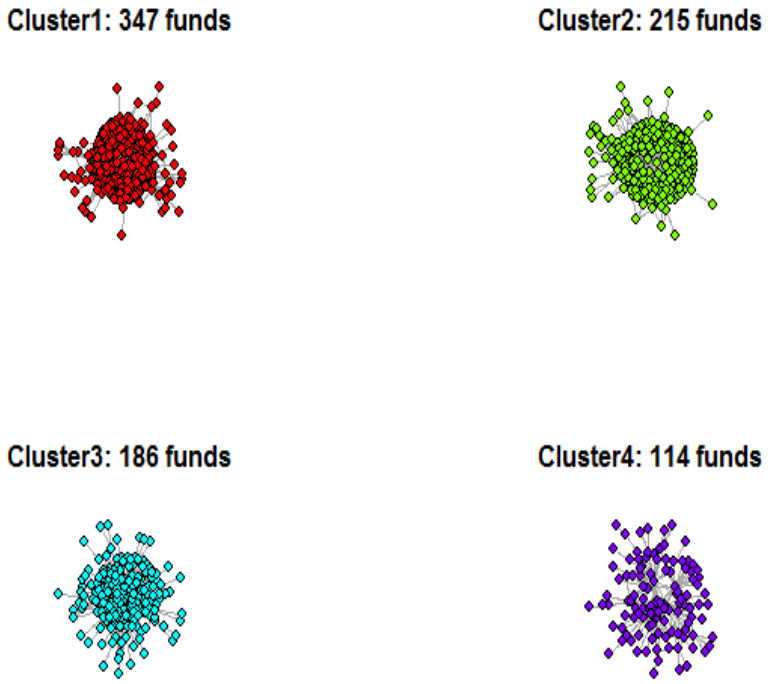


Table 1.12: Size of 20 largest clusters.

Cluster index	Number of portfolios	Cluster index	Number of portfolios
1	252	18	10
10	58	35	10
7	39	17	9
22	34	20	8
26	33	27	8
5	32	33	8
3	21	9	7
8	17	16	6
12	12	11	5
23	12	25	5

Figure 1.9: Community Structure by Greedy Algorithm 2



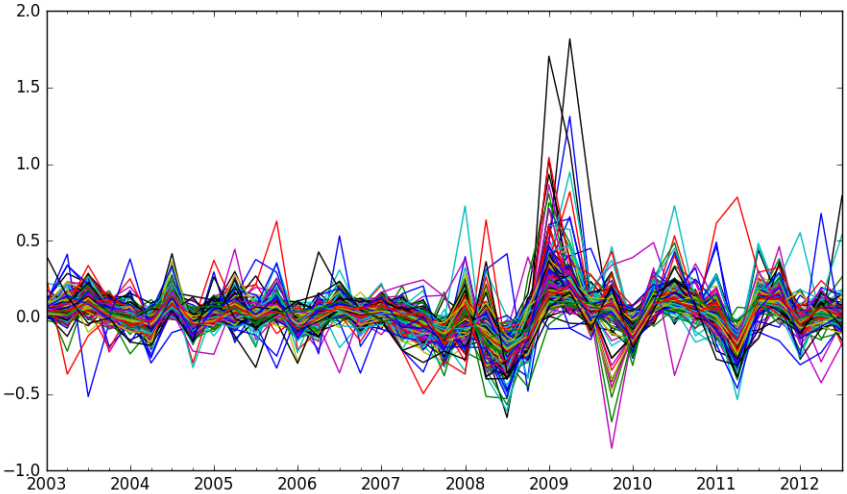


Figure 1.10: Plot of portfolio returns in quarters from 2003Q1 to 2012Q3

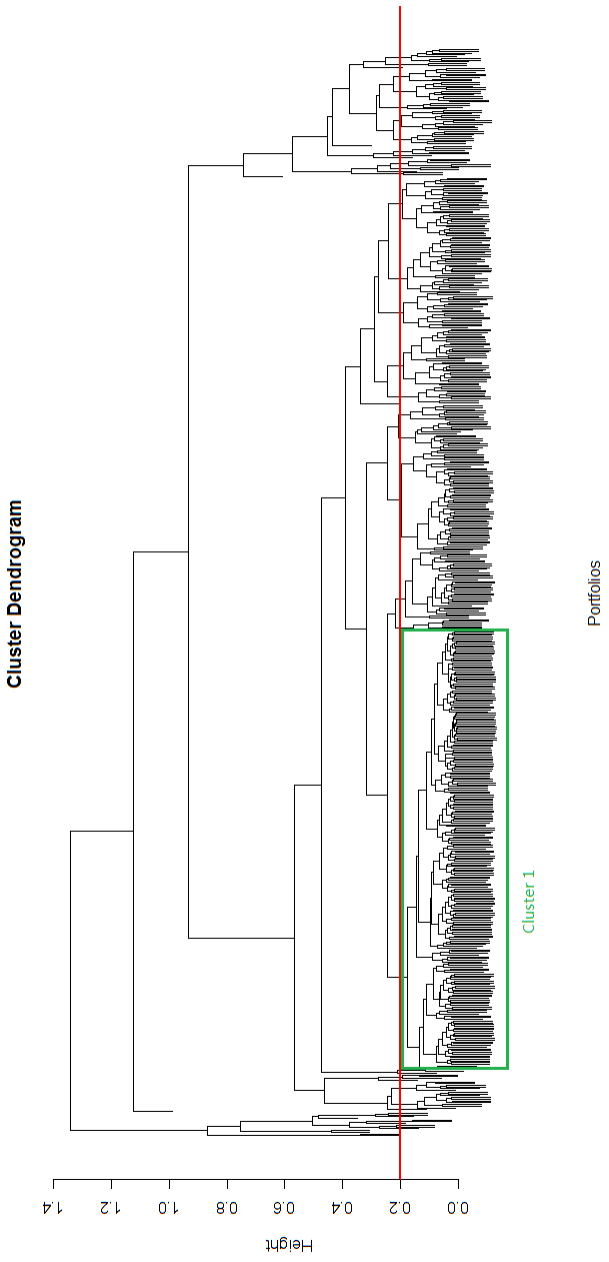


Figure 1.11: Dendrogram for hiearchical clustering

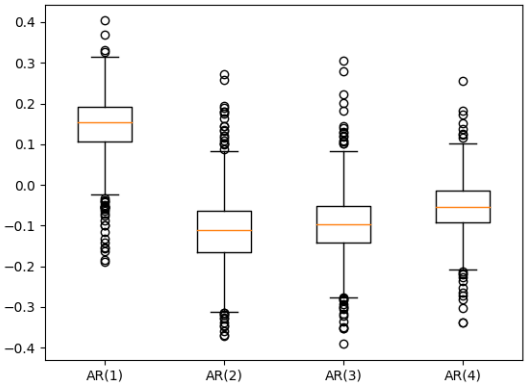


Figure 1.12: Autocorrelation coefficients of four quarters for portfolio returns

Table 1.13: Experiment results for different penalty functions.

	Penalty Method	MSFE	MSFE/Sample Mean	Run time(s)
Experiment 1	Lag	2.526	0.1966	683.85
(Heterogeneous design)	Own/Other	1.982	0.1542	935.98
	Network Own/Other	1.892	0.1472	587.73
	(sample mean)	12.846	1.0000	-
Experiment 2	Lag	2.231	0.1938	794.56
(Heterogeneous design)	Own/Other	1.701	0.1476	1224.26
	Network Own/Other	1.589	0.1378	603.41
	(sample mean)	11.525	1.0000	-
Experiment 3	Lag	2.370	0.5229	461.56
(Homogeneous design)	Own/Other	1.684	0.3715	4896.24
	Network Own/Other	0.437	0.0964	1625.32
	(sample mean)	4.532	1.0000	-

Table 1.14: Comparison with Sparse Own/Other method

	Penalty Method	MSFE	MSFE/Sample Mean	Run time(s)
Experiment 1	Sparse Own/Other	1.924	0.1498	43652.36
(Heterogeneous design)	Network Own/Other	1.892	0.1472	587.73
	(sample mean)	12.846	1.0000	-
Experiment 2	Sparse Own/Other	1.370	0.3023	163325.74
(Homogeneous design)	Network Own/Other	0.437	0.0964	1625.32
	(sample mean)	4.532	1.0000	-

CHAPTER 2

STATISTICAL ARBITRAGE IN EXPONENTIAL PATTERNS

2.1 Market Timing with Backward SADF Test

2.1.1 Introduction

The cascading liquidation behaviors among stock investors described in the previous chapter may result in financial crisis marked by fast decrease in stock price. As a matter of fact, investors are greedy during bull markets and scared during bear markets. Those sentiments of investors can spread in a tremendous speed, which is also known as the herding effect and often leads to exponential changes in stock price. The detection of exponential growth or collapse among time series is a heavily researched subject in quantitative investment. One of the most widely used tests for the purpose is the Augmented Dickey-Fuller (ADF) test developed in [28], which extends [29]'s original Dickey-Fuller test. One problem with the model is that it cannot distinguish between a stationary process and a series with periodically collapsing bubbles. To this end, [85] further extends the ADF test by running it recursively on subsets of the time series with backward expanding endpoints and taking the supremum of the ADF statistics. This test, known as the supremum ADF (SADF) test, is shown to have strong discriminatory power in detecting periodically collapsing bubbles based on simulation and empirical tests. For instance,[85] uses the test for the NASDAQ Composite in the 1990s and successfully pinpoints the origin and conclusion dates of the exuberance periods.

[80] further extends the SADF by proposing an alternative approach to detect bubbles known as the generalized SADF test (GSADF). The GSADF test, in short, is a rolling window SADF test where the start points for the ADF tests are also flexible, which enables it

to detect bubbles that take place in any period of a time series. When observed on simulated data, it has been proved to detect explosive bubbles more effectively than the SADF test when there are multiple bubbles in the time interval. For real-time bubble detection, the authors of [80] further proposed a backward SADF test, referred to as the BSADF test. The BSADF moves recursively through the time series the opposite way of the SADF. That is, it has a fixed endpoint and changing starting points. [80] also compares the results of SADF tests, GSADF tests, together with CUSUM monitoring procedures in [81] and [82] on S&P 500 data over a time period from January 1791 to December 2010. All tests are able to detect the presence of multiple bubbles, but more bubbles are detected by GSADF compared to other methods.

The GSADF test has been applied to detect bubbles for economic time series in numerous papers. [68] uses the GSADF test to analyze Hong Kong's real estate market and detects several periods of exuberance and crisis, including the well known real estate bubble in 1997. In [101], the superiority of the GSADF to the SADF is further verified in an application to find bubbles for the Shanghai A-share stock market. The GSADF is able to detect two bubbles, one centering around the subprime mortgage crisis from October 2006 to January 2009, and the other from May 2014 to July 2015. The SADF test, by contrast, can only detect one of them.

Market timing, on the other hand, is also a widely studied yet controversial topic in quantitative finance. The long-time opposition to market timing strategies is the Buy-and-Hold approach. The idea of market timing has been criticized by many scholars. The Efficient Market Hypothesis (EMH) states that the prices of all stocks and bonds reflect all the information currently available when the market is perfectly efficient. From this perspective, any market timing effort is based on the prediction of news in the future, which can be extremely difficult. In addition to that, studies have found that most of the investors do poorly in timing the market. For instance, [74] reports that Dalbar, a finan-

cial research market firm, found that over the past 20 years returns of private equity funds lagged behind the S&P 500 index return by 4.66% annually.

However, market timing is an indispensable part of management for most of the actively managed funds. For example, Warren Buffett, in spite of being a representative investor for the Buy-and-Hold strategy, dumped PetroChina (PTR) in 2007 when he decided that the market circumstances warranted selling, which can be seen as a behavior for market timing. The Buy-and-Hold strategy is also sometimes infeasible as most actively managed funds have stop-loss lines. Additionally, scholars in recent years have proposed various market timing strategies that work on real data. For instance, [48] finds that the discrete regression model technique works well for market timing on a stock-bond portfolio. [89] shows that a few simple market timing strategies based on the spread between the price-to-earnings ratio and a short-term interest rate outperform the Buy-and-Hold strategy. In [94], the authors proposed three new market timing strategies and showed that they beat the Buy-and-Hold strategy on the S&P 500 index. For stock markets outside the U.S, [66] shows that the moving average strategy substantially beats the Buy-and-Hold strategy for stocks with high volatility in the London Stock Exchange.

In this section, we propose a new market timing strategy aimed to exploit arbitrary opportunities in the bubble periods of stocks, where the price series is formed based on incomplete information and marked by low entropy. The rest of the section is organized as follows. The second section introduces the details of our strategy, including the time to enter the market and direction of the stock positions. In section three, we test our method on three different datasets and calculate the averaged holding period volatility, win rate and returns of the trades with different rules to leave the market. Finally, we discuss the advantages and disadvantages of our method based on the result in the last section.

2.1.2 Theoretical Development

The exponential pattern of stock price during a bubble period has a solid theoretical background. For instance, the Johansen-Ledoit-Sornette (JLS) model has been developed in [1], [3], [2], [93], [97] to describe the dynamics of financial bubbles and crashes. The authors stated that bubbles are generated by behaviors of investor that create feedback in the valuation of assets, and often end with a finite-time singularity in the future followed by a new crash regime. In this section, we introduce the JLS models based on [30].

The JLS model starts from the rational expectation settings where the observed price p_0 of a stock can be written as

$$p_0 = p^* + p$$

where p^* and p represent the fundamental and bubble component of the asset price respectively. The JLS model specifies that the fundamental and bubble component are independent from each other and describes the later with a stochastic equation with drift and jump:

$$\frac{dp}{p} = \mu(t)dt + \sigma dW - \kappa dj \tag{2.1}$$

where p is the stock market bubble price, $\mu(t)$ is the drift and dW is the increment of a standard Wiener process. The term dj represents a discontinuous jump such that $j = 0$ before the crash and $j = 1$ after the crash occurs. The loss amplitude associated with the occurrence of a crash is determined by the parameter κ. Each successive crash corresponds to a jump of j by one unit. The dynamics of the jumps is governed by a crash hazard rate $h(t)$. Since $h(t)dt$ is the probability that the crash occurs between t and $t + dt$ conditional on the fact that it has not yet happened, we have $E_t[dj] = 1 \times h(t)dt + 0 \times (1 - h(t)dt)$. Therefore, the expectation of dj is given by

$$E_t[dj] = h(t)dt$$

Under the assumption of the JLS model, noise traders exhibit collective herding behaviors that may destabilize the market. The model assumes that the aggregate effect of noise traders can be accounted for by the following dynamics of the crash hazard rate:

$$h(t) = B^{'}(t_c - t)^{m-1} + C^{'}(t_c - t)^{m-1} cos(\omega ln(t_c - t) - \phi^{'}) \tag{2.2}$$

The cosine part of the second term in equation 2.2 takes into account the existence of possible hierarchical cascades of accelerating panic punctuating the growth of the bubble, resulting from a preexisting hierarchy in noise trader sizes and/or the interplay between market price impact inertia and nonlinear fundamental value investing. Equation 2.2 also contains a hyperbolic power law growth ending at a finite-time singularity, which embodies the positive feedbacks resulting from technical and behavioral mechanisms.

The non-arbitrage condition expresses that the unconditional expectation $E_t[dp]$ of the price increment must be 0, which leads to

$$\mu(t) \equiv E[\frac{dp/dt}{p}]_{no\ crash} = \kappa h(t) \tag{2.3}$$

by taking the expectation of 2.1. Note that $\mu(t)dt$ is the return dp/p over the infinitesimal time interval dt in the absence of crash. Using this and substituting 2.2 and integrating yields the log-periodic power law (LPPL) equation:

$$lnE[p(t)] = A + B(t_c - t)^m + C(t_c - t)^m cos(\omega ln(t_c - t) - \phi)$$

where $B = \kappa B^{'}/m$, $C = \kappa C^{'}/\sqrt{m^2 + \omega^2}$, and m ranges from 0 to 1. Note that this expression describes the average price dynamics only up to the end of the bubble. The JLS model does not specify what happens beyond t_c. This critical time t_c is the termination of the bubble regime and the transition time to another regime. The parameter t_c represents the non-random time of the termination of the bubble. However, its precise value is not known with absolute precision, and its estimation can be written as

$$t_c^{est} = t_c^{true} + \epsilon$$

where ϵ is an error term distributed according to some distribution, while t_c^{true} is deterministic.

2.1.3 Methodology

Supremum ADF Test

The ADF test can be used to test exponential patterns within a time series. The regression specification of the model can be represented as

$$\Delta y_t = \alpha + ct + \beta y_{t-1} + \sum_{l=1}^{L} \gamma_l \Delta y_{t-l} + \epsilon_t$$

where y_t is the time series, L is the number of lags to be considered, ϵ_t is a white noise series, α, β, γ_l and c are coefficients to be estimated. The hypothesis testing framework for exponential patterns can be written as

$$H_0 : \beta \leq 0, H_1 : \beta > 0$$

We conclude that a time series is explosive when the null hypothesis is rejected.

Despite the popularity of the ADF test in detecting financial exuberance and crisis in the early years, it can not distinguish between a stationary time series and a multiple-bubble series, and is thus exposed to criticism by [43]. To this end, [85] proposes the Supremum ADF (SADF) test, which calculates the ADF statistics of subseries of the original series with a moving endpoint. That is,

$$SADF(r_0) = \sup_{r_2 \in [r_0, 1]} ADF_0^{r_2}$$

One problem with the SADF test is that it may ignore bubbles that take place at the end of the time series. Thus, [80] introduces the Generalized SADF (GSADF) test to solve

the deficiency by allowing for flexible start and endpoints of subseries. That is,

$$GSADF(r_0) = \sup_{\substack{r_1 \in [0, r_2 - r_0] \\ r_2 \in [r_0, 1]}} ADF_{r_1}^{r_2}$$

Although the GSADF method can search bubbles thoroughly within a time series, it is not recommended for real time bubble detection, as an earlier bubble can keep the GSADF statistics significant long after the bubble disappears. Thus, the authors of the paper suggested the use of Backward SADF (BSADF) for real-time detection. That is,

$$BSADF(r_0) = \sup_{r_1 \in [0, r_2 - r_0]} ADF_{r_1}^{r_2} \tag{2.4}$$

The BSADF test performs a supremum ADF test on a backward expanding sample sequence where the endpoint is fixed at r_2 and the start point varies from 0 to $r_2 - r_0$. It is used to decide when to enter the market and hold a position in our strategy.

MACD

MACD is the acronym for moving average convergence divergence, which is a widely used trend-following momentum indicator for stocks. It is usually calculated as the difference between exponentially averaged stock prices for 12 days and 26 days. It is obvious that a positive MACD value indicates an upward momentum in stock price and vice versa. The 9 days exponential average of the MACD series is often used as a buy or sell signal. In our model, however, it is not used to decide when to start a position, but the side of the position.

The Strategy

Trade Signal A trade signal aims to decide when to move into and out of the financial market based on predictive models. In our experiments, the signal to long or short a

stock is completely constructed by the BSADF statistic. There are several parameters that must be decided in order to calculate the BSADF statistic, namely the number of lags in the differenced price series to be included, and the maximum and minimum length of the sliding windows used to estimate the ADF statistics for each test. The maximum length refers to the number of trading days to construct one BSADF statistic, corresponding to the length of the entire time series in 2.4. The minimum length is the minimum number of trading days to conduct one ADF test for a BSADF statistic, corresponding to $1 - r_2$ in equation 2.4.

Based on the recommendation from [53], the maximum length of the sliding window is set to be 42 trading days. Also, the minimum length of the sliding window should be as short as possible to capture more recent bubbles. However, if the time series is too short then the coefficient estimations tend to have large variances due to the scarcity of data, especially when the number of lags is large. To find a balance between the number of lags and the length of the time series, it is recommended in [4] that $L \approx (length(x) - 1)^{\frac{1}{3}}$ in order to acquire reliable parameter estimations, where x is the length of the time series. Thus we set the minimum length of the sliding windows to be 10 trading days and the number of lags as 2. A good implementation of BSADF in Python can be found in [33].

After the BSADF statistic is calculated, a threshold is needed to trigger a trade signal. In our experiments, we used 1.76 and 1.96 as thresholds which correspond to 0.96 and 0.975 upper quantiles of a standard normal distribution for simplicity. The more sophisticated critical values can be obtained with simulation methods suggested in [80]. We then longed or shorted a stock whenever the BSADF is above the threshold. The intuition here is that when bubbles are detected in the previous months, there is a high probability that the bubbles can continue in the near future.

A triple barrier strategy is then applied to decide when to leave the market. That is, we set take-profit, stop-loss, and maximum holding period lines and cleared our position

whenever one of the three lines is touched. For simplicity, the barriers are hardwired throughout each trading path. More sophisticated methods may adjust the barriers based on the stock price volatility. To verify the robustness of our model, we tested it under a large number of combinations of the three barrier levels, each chosen from a grid of values. Specifically, the take-profit line ranges from 0.2 to 1.2 with step size equals 0.1, the stop-loss line from 0.1 to 1 with step size equals 0.1, and the maximum holding period from 20 days to 60 days with a step size of 10 days.

Bet Side After the time to enter the market is specified, we need to decide the direction of the bet, that is whether we should take a long or short position. The direction should be determined by the predicted price trend in the following months. There is a large number of papers concentrated on the prediction of stock price directions, e.g. [67]. Due to the limitation in computing power, we used a simple strategy for direction with the MACD(12,26,9) statistic, longing the stock when it is greater than zero and taking a short position otherwise.

2.1.4 Experiments

Data Description

To demonstrate the performance of the strategy, we ran it on three different datasets. All three datasets span a time period from January 1st, 2003 to December 31st, 2018. We split the data into two parts on December 31st, 2014, where the first part is used for training and validation, and the second part is for testing where we fixed the hyperparameters.

The first dataset contains the stock of 9 large investment banks, which are widely known as Bulge Bracket. The stocks we included are Morgan Stanley, Goldman Sachs, JP-Morgan Chase, Deutsche Bank, UBS, Credit Suisse, Citibank, Barclays, and Wells Fargo.

The second dataset contains 8 technology companies in the U.S. with the largest revenue according to Fortune 500. The dataset includes Apple, Amazon, Alphabet/Google, Microsoft, IBM, Intel, HP, and Cisco. We excluded Dell Technologies whose revenue came after IBM, due to its long-term trading suspension from October 29th, 2013 to September 7th, 2016.

The third dataset includes stocks with lottery features based on the data in the training period, which shares the features of high volatility, high skewness in return, and low stock price according to [62]. We are interested in the lottery stocks as they are highly sensitive to the socio-economic environment and may contain more discernible bubble periods. To construct the dataset, we first selected top 25% of stocks with highest averaged squared returns, highest coefficient of variation in returns, highest skewness in returns and lowest averaged stock price respectively, and then found the intersection of the four clusters using the training data. This gave us 20 stocks in total, 6 of which stopped trading before the end of the training period. Thus, we used the data of 14 stocks left to test our strategy.

Visualization of BSADF

We used the Barclays' stock to visualize the BSADF statistic together with the price and log trading volume series respectively. We learned from Exhibit 2.1 that the BSADF statistic typically ranges from -1 to 0, but occasionally explodes over 2 indicating exponential patterns. Most of the exponential patterns can be explained as overreactions to some market signals or events. The overreactions are then reinforced by the fear or greed of investors, as well as the triggering of a series of stop-loss lines of portfolios. Usually, the stock price begins to revert to its mean shortly after the exponential pattern, correcting the initial behavioral bias. A good example of an explosion in BSADF followed by an exponential pattern can be found at the beginning of 2009 after the great financial crisis. However, the bubble directions before and after the BSADF spike are different. Addition-

ally, we also found that a spike in the BSADF statistic is sometimes preceded by a spike in trading volume. This observation is quite intuitive as a bubble starts with a sudden increase in trading volume, but it takes time for the BSADF statistic to recognize it.

Trading Result

In this section, we demonstrate the trading results using the data in the testing sets, where the strategy is described in section 2.1.3. First and foremost, we display 3D plots of three trading results, namely holding period volatility, win rate, and returns per trade, as functions of take-profit, stop-loss, and maximum holding period lines. The volatility is defined as the standard deviation of daily returns during the holding period divided by the standard deviation of returns throughout the entire testing period for the stock. Additionally, the win rate is the proportion of trades whose returns are strictly positive. Each of the points in the plots corresponds to the averaged value for all trades across different stocks within a dataset with a fixed signal threshold, and the value is represented by its color.

Exhibit 2.2, 2.3 and 2.4 contain the 3D plots of returns per trade, volatility, and win rate for each of the datasets. The panel on the top left is for the bank data, the panel on the top right is for the data of technology companies and the panel on the bottom is for the lottery stock data. It is obvious that the vast majority of the trades have positive returns, and negative returns only occur in a very limited spectrum. Additionally, the optimal maximum holding periods are different for each dataset. The highest returns take place in the investment bank dataset where the maximum holding period is 60 days. By contrast, the best maximum holding period for technology company stocks are 20 to 30 days. For lottery stocks, however, the highest returns occur when the holding period is around 40 days.

Additionally, we found that the holding period volatilities are usually higher than the averaged volatilities throughout the testing period, especially for strategies with short maximum holding periods, low take-profit and stop-loss lines. Meanwhile, the win rates are significantly greater than 0.5 for almost all cases, especially those whose holding periods are shorter. This shows that our method can predict the price trend in the following month or so with good accuracy, but the prediction power fades away with the passage of time.

In addition to experiments under different trading rules, we also list the trading results under the optimal trading rules with the highest averaged return per trade, and compare them with the Buy-and-Hold (BAH) and Daily Rebalance (DR) strategy in Exhibit 2.1. We did not calculate the Sharpe Ratio for the strategy as it penalizes sharp increases in equity returns, which are quite common in the bubble periods according to our observation. In fact, the Sharpe Ratio is often criticized by practitioners for penalizing sharp increases and decreases in returns equally, and the removal of the highest returns could increase Sharpe Ratio.

It is clear that for the investment bank and lottery stock data, our market timing model achieved significantly higher annualized returns compared to both BAH and DR. For instance, when the signal threshold is set to be 1.96 for lottery stock data, the averaged return from one trade is greater than the annualized return from the BAH and DR strategy, when the averaged holding period is merely 27.24 days. For stocks of large technology companies, however, our model is comparable with the BAH and DR strategy in terms of annualized return. Considering the risk of market timing strategy, management and transaction costs, it is still recommended to apply a Buy-and-Hold strategy for gigantic technology companies. Additionally, we observe that a reduction in threshold from 1.96 to 1.76 is detrimental to the quality of the signal, as both win rate and return per trade tend to decrease. Thus, it is recommended to maintain a higher level of signal thresh-

old, even with the cost of missing some arbitrage opportunities. It is important to notice that the optimal trading rules are not known during the trades. However, there is great potential that our strategy could achieve similar results in the real market, as all returns per trade, win rate and volatilities are stable in a wide range of trading rules. Also, the probability of losing money with the strategy is low according to the experiments.

2.1.5 Conclusion

In this section, we propose a new market timing strategy based on the BSADF statistic and an exponential moving average indicator. The BSADF statistic is used to detect bubbles in stock price that form in the previous trading days, and we presume that the bubble will continue in the following trading days. Meanwhile, the exponential moving average indicator is used to decide the direction of the bet. The strategy is tested with three different datasets containing stocks from investment banks, big technology companies, and lottery stocks from the beginning of 2015 to the end of 2018. We identified that our strategy is able to generate positive returns in most of the scenarios, but may lose money in some rare cases.

There are several merits of our method compared to previously published papers. First and foremost, our experiments are free from look-ahead bias and hindsight as we strictly separated the data into in-sample and out-of-sample parts. While we constructed our strategy based on 11 years of in-sample data, we ran our experiments once on the testing sets with fixed hyperparameters and recorded the results. Furthermore, apart from reporting the highest returns, we also reveal the trading results under hundreds of combinations of signal thresholds, take-profit, stop-loss, and maximum holding period barriers, which in essence describes the distribution of the trading results and helps investors understand the risk of the strategy. Additionally, to avoid the overfitting problem,

we did not test exhaustively on the training set to find the best hyperparameters for the model. Instead, the maximum and minimum window length to detect bubbles, as well as the parameters for the MACD we used in the experiments are recommended by previous papers.

It is important to stress that our study has limitations. First, we did not explore the way to find the optimal trading rules for the strategy. In fact, we believe that it is very hard to predict the optimal time to clear the position and leave the market in the future, as the distribution of stock returns changes drastically with the market environment. Second, there are periods of time without significant bubbles where our strategy cannot provoke any trades. In this case, we suggest the Buy-and-Hold strategy on stocks or bonds. Last but not least, the good performance of the strategy in the past may not continue in the future, which is a problem faced by almost all trading strategies.

2.2 Machine Learning Method to Improve Market Timing

2.2.1 Introduction

A meta strategy is a systematic approach to apply existing strategies for better problem solutions. In quantitative investment, meta strategies are designed to improve the performance of basic strategies. For instance, [56] proposes a two-layer bias decision tree to generate buy signals for stocks, where the first layer is used to predict the direction of stock price change and the second layer to predict whether the price change can surpass a threshold in the near future. Two relative strength index (RSI) statistics are used as base signals at each trading day, and their relationship with stock returns in the following 100 trading days are learned using the training data. The model is then applied in the testing set to decide the purchases of stocks. Additionally, [86] uses machine learning models to combine different basic trading rules to decide whether to buy, sell a stock or stand by in a daily re-balancing strategy. The authors first collected all decisions proposed by the basic trading rules at each training day, and labeled them with the optimal trading rule based on the stock price of the next day. For the next step, linear and non-linear models are used to detect relationships between the recommended decisions given by the basic rules and the optimal rule. Then the trained model is used to combine basic rules in the testing period.

2.2.2 Meta Strategy

According to our empirical results, 40% to 50% of trades initiated by the primary strategy have negative returns based on different barriers to leave the market. To reduce the num-

ber of failures from the primary strategy, we designed a meta strategy based on machine learning. That is, we ran the primary strategy on the training data, recorded the return for each trade and then used machine learning models to detect patterns between the trade parameters and returns. The trained meta models is then used to decide which trades are more likely to make higher profits.

Feature Engineering

We collected 6 features to describe the state of each trade when it is initiated. The first and the second features are the BSADF and MACD statistics which trigger the trade and decide the direction of the bet. The third feature is the percentage change in daily trade volume of the stock, which is intuitively related to how long the exponential pattern can continue. On the one hand, a continuous exponential pattern in stock price must be supported by increases in trade volume. On the other hand, an extreme increase in trade volume is usually coupled with a radical change in stock price, causing the BSASD statistic to reach the threshold which could initiate a trade. However, the volume spike may be due to some extreme events or anomalies whose impacts can fade away quickly. Additionally, we added the take-profit and stop-loss levels as well as the maximum holding period as features for our machine learning algorithm.

Random Forest Classifier

After the features have been constructed, we applied random forest to detect their relationship with trade returns. The random forest model is widely used in the finance industry as it has robust performance for noisy data. Considering the high noise level, we converted the trade returns into categorical labels and implemented the model as a binary classifier. Specifically, we labeled a trade as 1 if the trade return belongs to the

top 20% among all trades, and 0 if the trade return belongs to the bottom 20%. We then trained a random forest classifier based on the trading results in the training set and used it for the trade in the testing set. Whenever the BSADF statistic is above the threshold, we collected the 6 features for the trade and input them to the random forest model. The output of the model is a scaler ranging from 0 to 1, which gives the probability that the trade will make a "good" return. We then executed the trade when the probability is above a threshold (0.5 in our experiments), and abandoned the signal otherwise. In this way, we could filter out trades that presumably lose money and keep those that have good chance to earn money.

2.2.3 Experiments

Trading Result

We use the same datasets for stocks of large bank and technology companies. Lottery stocks are not included for the meta strategy as most of the stocks in the in-sample data disappear in the out-of-sample data. The result can be found in 2.2. After the meta model is applied to the primary strategy, the averaged returns per trade increase substantially. For cases where the threshold equals 1.96, the highest averaged return per trade increases by almost 1% for bank data and is around 4 times higher for stocks of technology companies. The improvement has been due to the fact that the meta model filters out the trades in the primary strategy which are likely to have low returns based on the prediction of the meta model. Additionally, we also observe notable improvements in annualized return per trade and win rate for both datasets and thresholds. For instance, the win rate rises from 62.65% to 81.60% for technology company data when the threshold is set to be 1.96.

Feature Importance Analysis

Apart from the trading result, we also analyzed the feature importance of the random forest models. First for each decision tree in a random forest, we calculated for each feature the averaged decrease in impurity weighted by the number of samples in the splits. Then we found the mean and confidence interval of the averaged decreases in impurity for each feature across all decision trees. The corresponding plots for both datasets can be found in Exhibit 2.11 when the threshold is set to be 1.96. The height of each green bar represents the mean decrease in impurity (MDI) and each black line gives the confidence interval. It is clear to all that the importance ranks of features are the same for both datasets. The most important feature is MACD followed by SADF and percentage change in trading volume. The three features above directly construct the trading signals. On the other hand, the three rules to leave the market only play minor rules in deciding whether a trade is highly profitable or not. This further verifies the robustness of our trading model under different trading rules.

2.3 Exhibits for Chapter Two

Table 2.1: Trading result with out-of-sample data

Dataset	Banks	Banks	Tech	Tech	Lottery	Lottery
Signal Threshold	1.96	1.76	1.96	1.76	1.96	1.76
Number of trades	73	102	78	99	123	151
Daily volatility	1.05	1.33	1.55	1.43	1.72	1.23
Win rate	68.12%	61.97%	62.65%	61.20%	63.34%	56.75%
Return per trade	4.97%	2.75%	2.19%	1.88%	3.05%	1.29%
Averaged holding period(days)	40.73	38.88	19.83	19.64	27.24	27.24
Annualized return per trade	36.57%	18.08%	27.83%	24.12%	28.21%	11.93%
Annualized return for BAH	-2.89%	-	23.93%	-	2.07%	-
Annualized return for DR	-1.66%	-	16.54%	-	2.31%	-

Abbreviation: BAH = Buy-and-Hold portfolio; DR = Daily rebalance portfolio

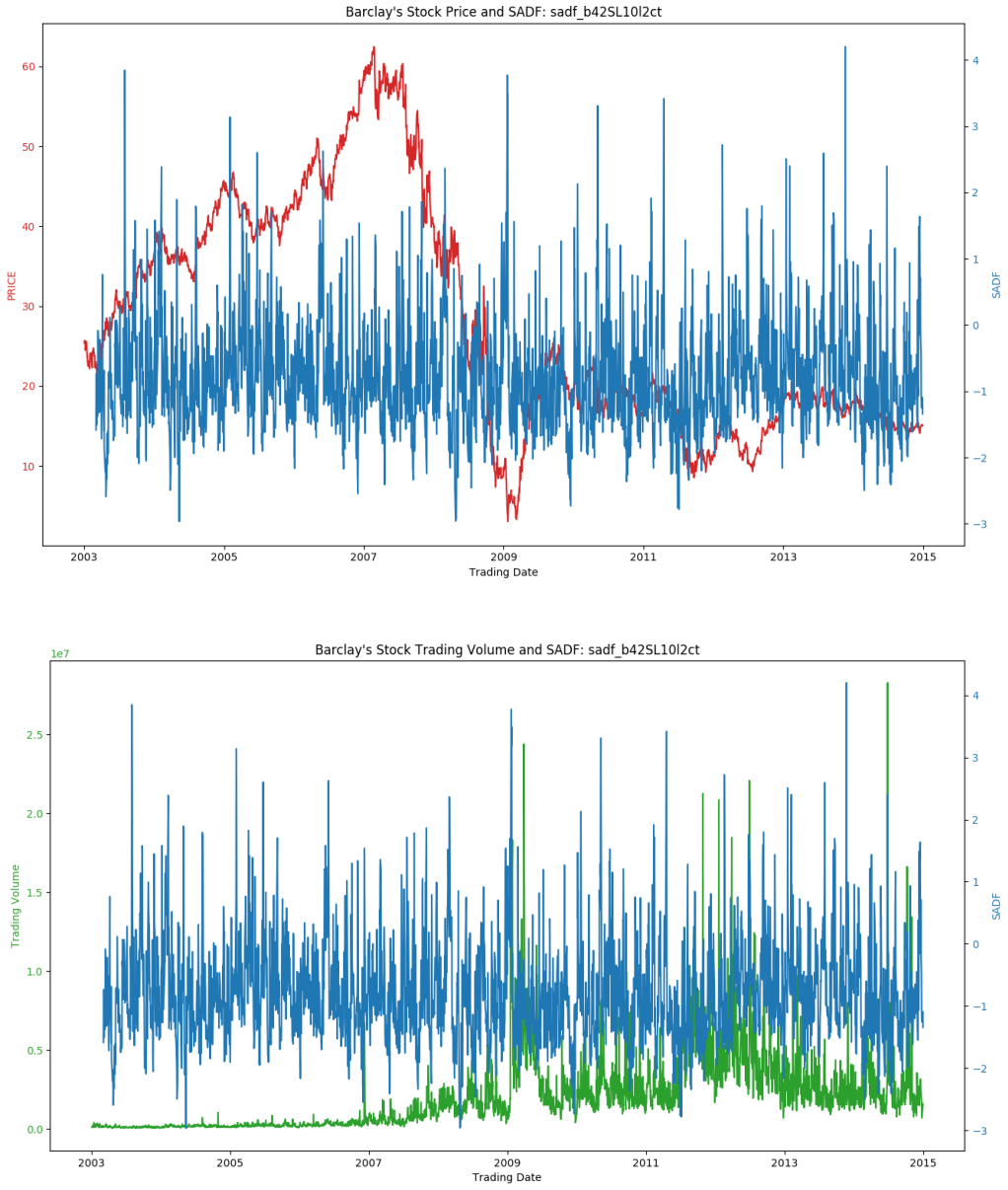


Figure 2.1: The BSADF statistic for Barclays with price and trading volume

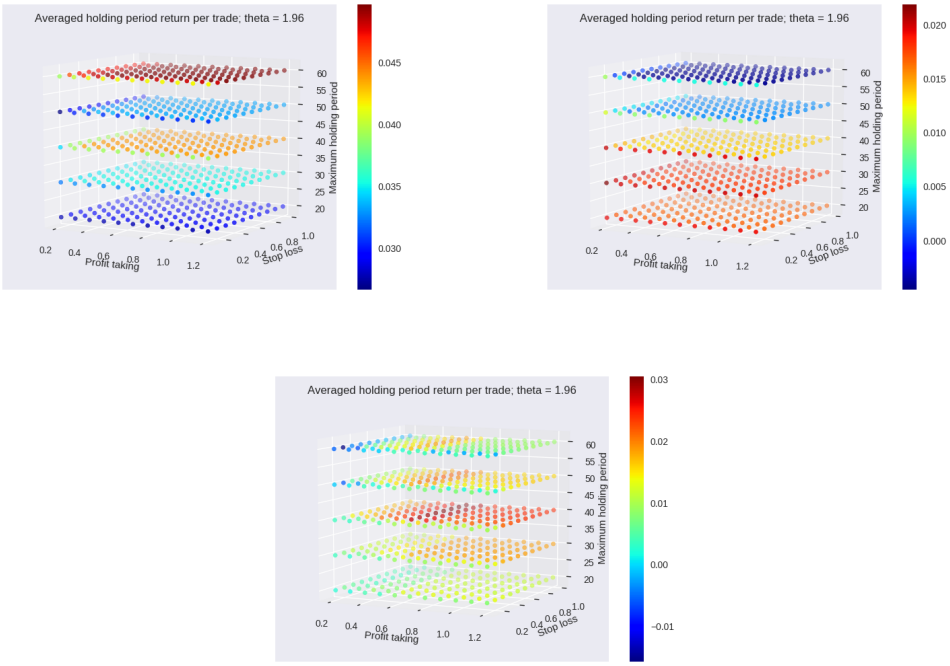


Figure 2.2: 3D plots of trading returns for each dataset

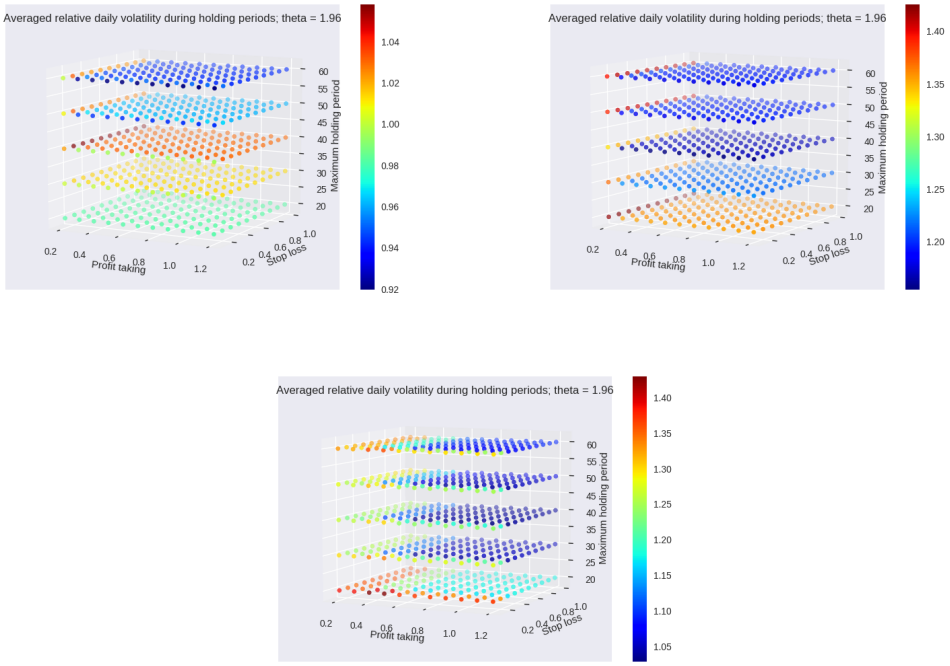


Figure 2.3: 3D plots of holding period volatility for each dataset

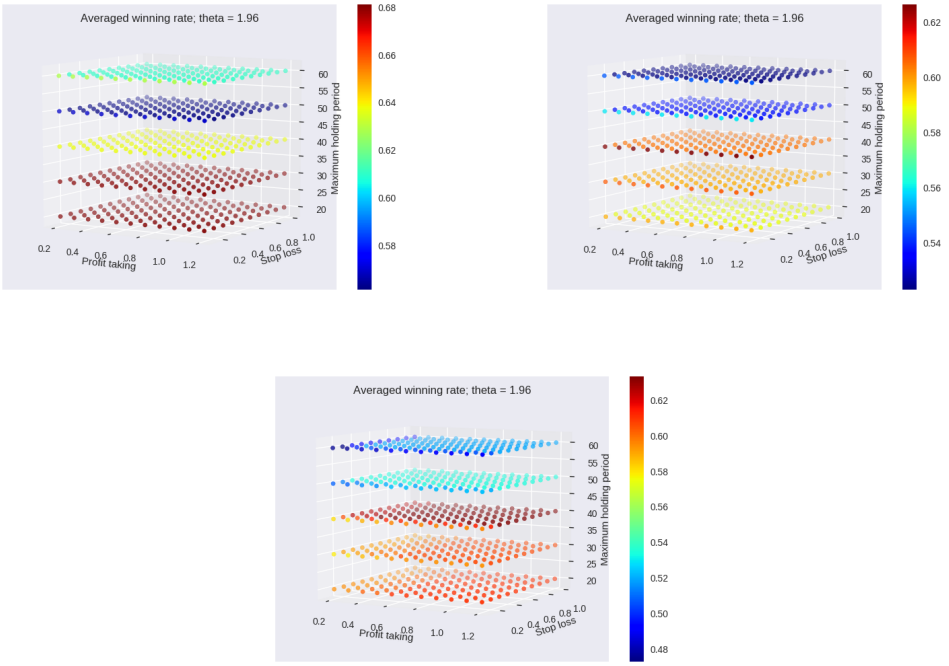


Figure 2.4: 3D plots of win rate for each dataset

Table 2.2: Comparison of trading result without and with meta strategy using out-of-sample data

Dataset	Banks	Banks	Tech	Tech	Banks	Banks	Tech	Tech
With Meta Strategy	No	No	No	No	**Yes**	**Yes**	**Yes**	**Yes**
Signal Threshold	1.96	1.76	1.96	1.76	1.96	1.76	1.96	1.76
Classification Accuracy	-	-	-	-	69%	67%	66%	67%
Number of trades	73	102	78	99	56	63	29	78
Daily volatility	1.05	1.33	1.55	1.43	0.91	1.56	1.08	2.47
Win rate	68.12%	61.97%	62.65%	61.20%	70.80%	68.81%	81.60%	76.21%
Return per trade	4.97%	2.75%	2.19%	1.88%	**5.94**%	**3.82**%	**8.26**%	**5.92**%
Averaged holding period(days)	40.73	38.88	19.83	19.64	34.24	34.35	33.82%	19.50%
Annualized return per trade	36.57%	18.08%	27.83%	24.12%	**43.71**%	**28.02**%	**61.54**%	**76.50**%
Annualized return for BAH	-2.89%	-	23.93%	-	-	-	-	-
Annualized return for DR	-1.66%	-	16.54%	-	-	-	-	-

Abbreviation: BAH = Buy-and-Hold portfolio; DR = Daily rebalance portfolio

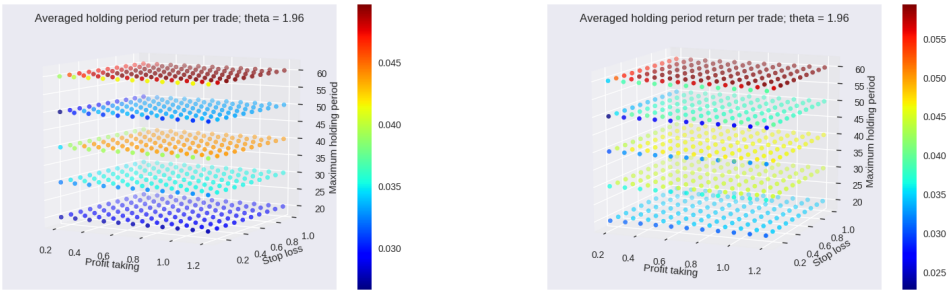


Figure 2.5: 3D plots of trading returns for Bank data without and with meta model

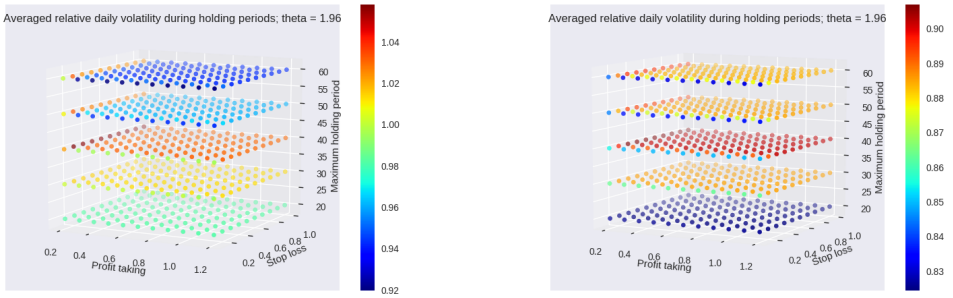


Figure 2.6: 3D plots of holding period volatility for Bank data without and with meta model

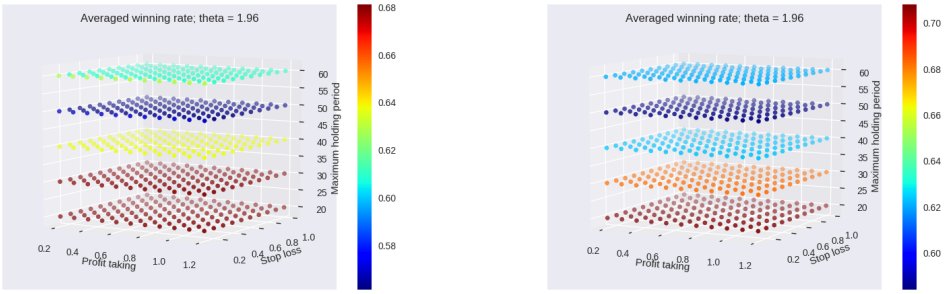


Figure 2.7: 3D plots of win rate for Bank data without and with meta model

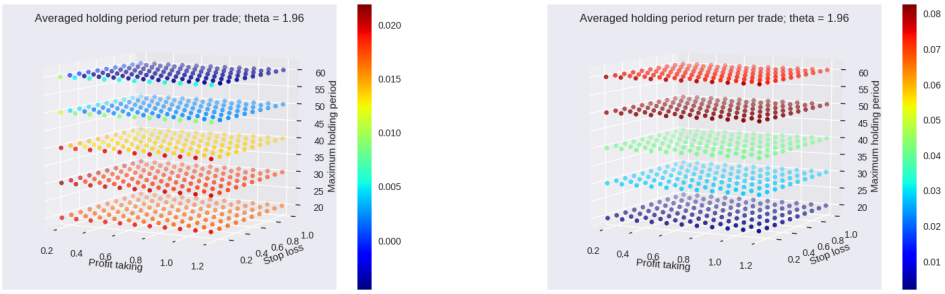


Figure 2.8: 3D plots of trading returns for Tech data without and with meta model

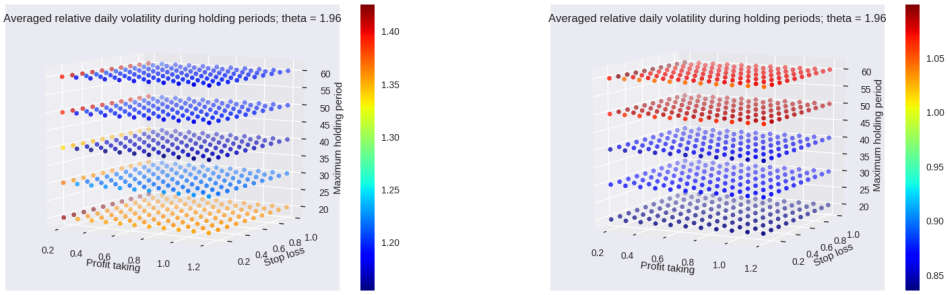


Figure 2.9: 3D plots of holding period volatility for Tech data without and with meta model

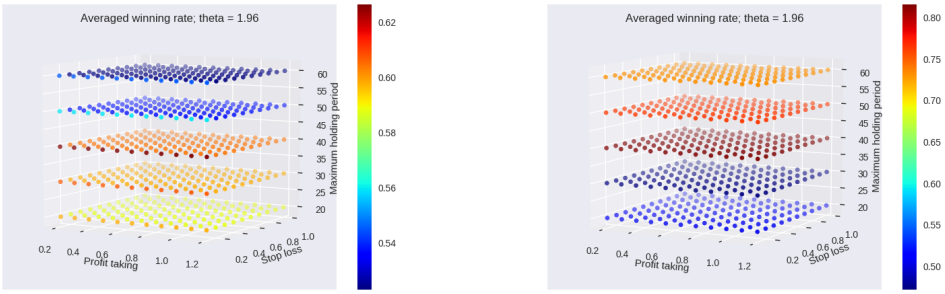


Figure 2.10: 3D plots of win rate for Tech data without and with meta model

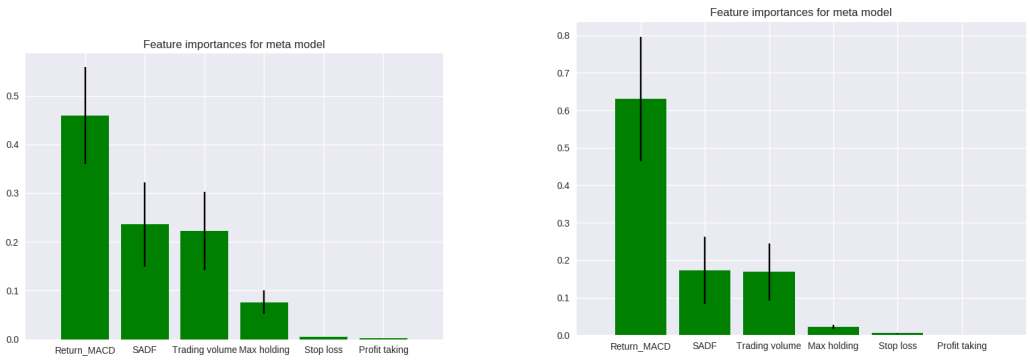


Figure 2.11: Feature importance plots for meta models
Left panel: Bank data; Right panel: Tech data

CHAPTER 3

VOLATILITY PREDICTION WITH RANDOM FOREST

3.1 Introduction

Volatility prediction for stock return is a widely studied problem in quantitative finance, as it is essential in various fields such as option pricing, portfolio allocation, and risk management of portfolios. One important feature of volatility prediction problem is that the real volatility cannot be observed directly, thus it is crucial to find a proxy for it. One popular proxy for volatility is the squared return of adjusted close price of stocks. It is an unbiased estimator of the true variance when the stock return is modeled as a diffusion process without a drift. However, squared return is considered to be a noisy measurement of daily volatility, and it has been found in numerous studies, such as [27], [59], [46], that squared return is poorly explained by standard volatility models such as the GARCH type models. Other related volatility estimators are introduced in [83], [49], [63], [99], etc. Those estimates usually only require the highest, lowest, open and close price of a stock during the trading days. Another more complicated volatility proxy is the cumulative squared intraday return proposed by [9], which is usually referred to as the realized volatility, and has become the new state-of-the-art ex-post forecast evaluation criteria. It has been stated in [9] that realized volatility can be well explained by GARCH type models. However, realized volatility is parameter dependent, as the intraday return interval and the aggregation period need to be decided by the users. Also, high frequency data is not available to the public.

When it comes to volatility prediction models, the most widely used ones are the ARCH model and its derivatives, which are introduced in [41] and [42]. The key idea behind the models is based on volatility clustering, which means that days of high volatility

tend to be followed by days of high volatility, while days of low volatility tend to be followed by days of low volatility. More recently, a model-free approach called normalization and variance stabilization method (NoVaS) was introduced in [84], which is free from structural model assumptions and could outperform GARCH-type model in predicting the volatility of $S\&P500$ index according to [39]. One issue shared by GARCH and NoVaS is that the training window should be sufficiently long in order to acquire reliable parameter estimations. However, a long term financial time series such as stock returns tends to be non-stationary and contains embedded structural breaks. Thus it is vital to balance the two factors.

Machine learning methods, on the other hand, have been used for volatility prediction in the recent years as well. [65] makes prediction on the realized volatility of $S\&PASX200$ index of Australia using random forest algorithm, where historical averaged realized volatility estimates of different sliding window length are used as features. [55] predicts several commonly used volatility proxies using their own lags and the lag terms of another proxy. The authors found evidence from experiments on the stocks included in the $CAC40$ index that it is possible to improve the prediction of the future value of one proxy by using the information provided by other proxies.

The meritorious points of our method are listed as follows. First and foremost, most previous literatures focus on the volatility prediction of univariate time series. One possible issue for this is that the time horizon for the training process is so long that the model may learn from outdated and unrelated data. For instance, if we include data from financial crisis period into our training set and use the model for prediction during non-crisis period, then the out-of-sample prediction error of the model is likely to be high as the data in the training and testing set do not follow a similar distribution. By contrast, the proposed method trains a single model at each time term using stacked cross-sectional data among stocks within one industry, whose return series could presumably share similar

patterns. Furthermore, in our empirical study, we focused on all stocks within the financial industry that is traded in US exchanges, which contains some stocks with smaller market cap. The vast majority of related papers, on the other hand, work on the prediction problem of stocks with large market cap or market indices, whose volatilities are usually more stable and thus easier for prediction. More importantly, unlike [65], we included a truly out-of-sample data of 1.5 years in our assessment, which is completely uncontaminated from hyperparameter tuning and cross validation. In this way, we further validates the improvement of our model in unseen circumstances. As a matter of fact, no data in [65] is authentically out-of-sample as the hyperparameters in the random forest are tuned to fit the validation data. Additionally, we enables the feature sampling of volatility proxies where each of the decision tree is constructed in order to enlarge the hypothesis space of the random forest. As is suggested in [55], we uses two volatility proxies to build each decision tree.

3.2 Methodology

3.2.1 Classical Volatility Estimators

We first introduce the notations for the normalized prices.

- $c_t = lnC_t - lnO_t$: the normalized close price.
- $o_t = lnO_t - lnC_{t-1}$: the normalized open price.
- $u_t = lnH_t - lnO_t$: the normalized high price.
- $d_t = lnL_t - lnO_t$: the normalized low price.

The four classical volatility estimators we used to build our random forest predictors can then be calculated as

- Close to close estimator:

$$\hat{\sigma}_c^2(t) = (c_t - c_{t-1})^2$$

- Parkinson estimator:

$$\hat{\sigma}_{pk}^2(t) = \frac{(u_t - d_t)^2}{4ln2}$$

- Garman Klass estimator:

$$\hat{\sigma}_{gk}^2(t) = 0.511(u_t - d_t)^2 - 0.019[c_t(u_t + d_t) - 2u_t d_t] - 0.383c_t^2$$

- Rogers Satchell estimator:

$$\hat{\sigma}_{rs}^2(t) = u_t(u_t - c_t) + d_t(d_t - c_t)$$

The close-to-close estimator is the simplest and most commonly used volatility proxy, which is based purely on prices of closing auction. It is used both as one feature for our random forest and also as the prediction goal of our model. The Parkinson estimator introduced in [83] is widely believed to be the first advanced volatility estimator, which uses highest and lowest prices of the day. It assumes that the stocks in the market are traded continuously, thus underestimates the stock volatility. Additionally, the Garman Klass estimator given by [49] can be seen as an extension of Parkinson estimator, which adds open and close price information in the estimator. It achieves the minimum variance among all unbiased scale-invariant estimators in a sense explained in [49]. Additionally, [63] introduces the Rogers Satchell estimator, which could accommodate stocks whose prices follow Geometric Brownian motions with non-zero drift. This could lead to an improvement when the mean return of a stock is significantly non-zero, as the models which assume zero drift tend to overestimate the volatility.

On the other hand, overnight jumps in stock price are also taken into account by some estimators, such as the $\hat{\sigma}_6^2$ introduced in [49] and the Yang-Zhang estimator in [99], as the trading volumes tend to be higher at the beginning of trading days due to the accumulation of information overnight. However, we found in our study that $\hat{\sigma}_6^2$ actually severely overestimates the volatility. Also the Yang-Zhang estimator is calculated based on the stock prices of previous trading days. As a result, the length of the window used to construct the estimator becomes another hyperparameter to be decided. Thus we did not use them to train our ensemble model.

3.2.2 Averaging

The classical estimators introduced in 3.2.1 can be used directly as a predictor for the volatility in the following day based on the volatility clustering phenomena. That is, at day t, the predicted volatility of a stock at day $t+1$ can be represented as

$$\hat{\sigma}_{v,pred}^2(t+1) = \hat{\sigma}_v^2(t)$$

where $v \in \{c, pk, gk, rs\}$ is the indicator of volatility estimator. Note that we use a hat on top of σ for both estimation and prediction, and add $pred$ in the subscript to label the prediction value. However, this prediction method may lead to high prediction errors in that the daily volatility estimators listed in 3.2.1 are highly noisy, as only one day of data is used. Thus, the moving average of classical predictors are used to acquire denoised volatility predictors. The equally weighted version of moving average for each volatility estimator with window length p can be calculated as

$$\hat{\sigma}_{v,eq}^2(t,p) = \frac{1}{p}\sum_{k=t-p+1}^{t} \hat{\sigma}_v^2(k)$$

where $v \in \{c, pk, gk, rs\}$ is again the indicator of volatility estimators, $\hat{\sigma}_v^2(k)$ is the volatility estimator v at day k. An alternative method is to calculate exponential moving averages of

volatility estimators, considering the fact that the volatility estimates in the recent trading days are more influential than those that took place a long time ago. The exponential moving average of volatility estimators can be written as

$$\hat{\sigma}^2_{v,exp}(t) = \begin{cases} \hat{\sigma}^2_v(1) & t = 1 \\ \alpha\hat{\sigma}^2_v(t) + (1-\alpha)\hat{\sigma}^2_{v,exp}(t-1) & t > 1 \end{cases}$$

where α is the decay parameters. The weighted volatility estimates can be used as volatility predictors directly or combined together using random forest to yield more accurate predictions.

3.2.3 Random Forest

Notation

In the first place, we introduce the notations for the algorithm.

- N_{est}: number of classical estimators to be trained by the random forest. In our case, $N_{est} = 4$ at this time.

- $x^{(j)}_{t,i}$: the volatility prediction of classical estimator j at day t of stock i. It is important to notice that $x^{(j)}_{t,i}$ is acquired based only on data up to day $t-1$, but not on day t.

- $x_{t,i} \equiv [x^{(1)}_{t,i}, x^{(2)}_{t,i}, \ldots x^{(N_{est})}_{t,i}]$: a vector of dimension $1 \times N_{est}$ which includes all classical volatility estimations of stock i at day t.

- $y_{t,i}$: the volatility proxy of stock i at day t, which is the squared return in our case. This is used as the label of $x_{t,i}$ in training and testing.

- $z_{t,i} \equiv [x_{t,i}, y_{t,i}]$: the observation vector of dimension $1 \times (N_{est} + 1)$ of stock i at day t.

- $Z_t(1)$: the observation matrix at day t after we concatenate all $z_{t,i}$ vertically for all stocks. In matrix form, it is:

$$Z_t(1) = \begin{bmatrix} z_{t,1} \\ z_{t,2} \\ \vdots \\ z_{t,I_t} \end{bmatrix}$$

  I_t is the number of stocks that present at day t, thus $Z_t(1)$ is a matrix of dimension $I_t \times (N_{est} + 1)$.

- $Z_t(p)$: the combined observation matrix after we concatenate observations from day t and $p - 1$ preceding days of day t. In matrix form, it can be represented as:

$$Z_t(p) = \begin{bmatrix} Z_t(1) \\ Z_{t-1}(1) \\ \vdots \\ Z_{t-p+1}(1) \end{bmatrix}$$

  It has a dimension of $(I_t + I_{t-1} + \ldots I_{t-p+1}) \times (N_{est} + 1)$. Typically it is a skinny matrix where the number of observations is far more than the number of features.

Random Forest for Combining Estimators

The volatility estimator based on random forest needs two operations at day t to acquire the estimates at day $t + 1$, namely training and prediction. We thus present both of them as follows.

Training: In this step, we trained a random forest based on $Z_t(p)$. The algorithm requires

two major hyperparameters, which are the number of decision trees N_{tree} and the maximum depth of decision trees, which will be further discussed in 3.2.3. When the two parameters are fixed, the algorithm proceeds by resampling the rows of $Z_t(p)$ for N_{tree} times with replacement, where the dimension of each resampled data matrix $Z_t^{(n)}(p)$ has the same dimension as $Z_t(p)$ by default. Then we can train a regression decision tree based on $Z_t^{(n)}(p)$.

We applied the classical CART algorithm to train decision tree $T_t^{(n)}$. The algorithm starts by recursively splitting the data matrix $Z_t^{(n)}(p)$ into two parts by rows based on a selected feature j and a corresponding split point s. For instance, let $x_m(A) \equiv [x_m^{(1)}(A), \dots x_m^{(N_{est})}(A)]$ be the mth row of a data matrix A in one split. Then A is split into A_1 and A_2 such that for each row $x_m(A)$,

$$x_m(A) \in \begin{cases} A_1 \text{ if } x_m^{(j)} \leq s \\ A_2 \text{ if } x_m^{(j)} > s \end{cases}$$

j, s, along with the associated outputs of the two subsets, are selected to minimize the mean square error. Further details of the algorithm can be found in [18]

Prediction: In this step, we made prediction for the volatility at day $t+1$ based on the random forest acquired in the previous step. First we calculated the classical volatility estimates $x_{t+1,i} = (x_{t+1,i}^{(1)}, x_{t+1,i}^{(2)}, \dots x_{t+1,i}^{(N_{est})})$ for each stock i at time $t+1$. Then for each trained tree $\mathbf{T}_t^{(n)}$, we started from the node to find the predicted value corresponding to $x_{t+1,i}$ until we reached a leaf with output $\hat{y}_i^{(n)}$. We then calculated the simple average of $\hat{y}_i^{(n)}$ to acquire the final estimates $\hat{y}_i$.

Model Parameters

There are four major hyperparameters to be tuned for a random forest model.

- Subsample size: the number of rows in the bootstrapped sample to build each decision tree. It turns out to be the most important hyperparameter for the model, as smaller subsample size could make the decision trees more distinct from each other, which reduces the variance of the final prediction.
- Number of features: the number of features used to build each decision tree. To create a large number of possible trees, we set it to be 2 or 3. It is important to notice that the subsample size and number of features may change the dimension of $Z_t^{(n)}(p)$.
- Number of trees: the number of decision trees to be estimated based on the bootstrapped data. A larger number of trees in the random forest is not usually detrimental to the out-of-sample prediction performance of the model.
- Maximum depth: the maximum depth of the decision trees. When the tree is split to the maximum depth, no further split is allowed. This parameter is used to control the complexity of decision trees, as over-split trees tend to have larger generalization errors.
- Loss function: the loss function to train each decision tree, which can be either absolute error or squared error.

Additionally, there are some other hyperparameters such as minimum number of observations for a further split, minimum criterion decrease for a further split, etc. We used the default values given in the scikit- learn package for these hyperparameters as the prediction power of random forest is usually less sensitive to them.

3.3 Data Preparation

Our experiments are based on the CRSP database provided by the Wharton Research Data Services (WRDS). To reduce the computing time while maintaining a reasonable amount of stocks to justify our method, we selected all stocks in the finance sector that are traded in the US stock market, which includes NYSE, NASDAQ, ARCA and Bats. Overall there are 558 stocks in the finance sector based on on Fama-French's 12 industry portfolios over a period from Jan 1st, 2007 to Jun 30th, 2018. We chose this period as it includes the two major financial crisis in 2008 and 2011, as well as several non-crisis interim periods. The stocks we selected presumably follow similar patterns during different time periods as they are in the same sector, and we trained one machine learning model across all stocks in each trading day.

Figure 3.1 and 3.2 plot the averaged volatility of stocks using equal and market cap weights, where the squared return is used as the volatility proxy. Obviously Figure 3.2 is preferred, as it is more robust to outliers in returns, which usually take place in stocks with small market cap. The figure clearly reflects the two highly volatile periods corresponding to the financial crisis in fall 2008 and summer 2011, as well as a few shorter volatile periods.

3.4 Experiments

3.4.1 Validation and Testing

We split the entire data set into two parts, one for validation and the other one for testing. The validation set contains data from the beginning of 2007 to the end of 2016, while the

testing set contains data from the beginning of 2017 to mid 2018. We tuned the hyperparameters for experiments with data in the validation set to acquire the best performance. It should be noticed that the optimal combination of hyperparameters is selected to minimize the MAE of random forest models during the validation period. By contrast, we only report the result from the first experiment using the data in the testing set to investigate its real time prediction performance.

As to the evaluation methods for prediction, we used mean absolution error(MAE), mean squared error(MSE) and bias as criteria. The formulas for the errors are given by

- MAE: $\frac{1}{N}\sum_{i,k} w_{it}|\hat{\sigma}_{it}^2 - R_{it}^2|$
- MSE: $\frac{1}{N}\sum_{i,k} w_{it}(\hat{\sigma}_{it}^2 - R_{it}^2)^2$
- Bias: $\frac{1}{N}\sum_{i,k} w_{it}(\hat{\sigma}_{it}^2 - R_{it}^2)$

where w_{it} is the weight to calculate the averaged errors, $\hat{\sigma}_{it}^2$ is the predicted squared return of stock i at time t, R_{it}^2 is the squared return of stock i at time t, N is the number of predictions over different stocks and trading days. We used both equal and market cap weight in our experiments. The in-sample prediction result of the random forest method using the entire validation data set can be seen in Table 3.1 and 3.2, where the subsample size is set to be 10% of the training set, number of features is 2, number of trees is 50, maximum depth of the trees equals 2, and MAE is used as the loss function. For comparison, we also calculated the prediction errors of the moving average methods introduced in 3.2.2, a zero predictor which always predicts zero, GARCH(1,1) and NoVas-type models with training period equals 100 trading days.

From the result, we find that the random forest predictor achieves the minimum MAE among all predictors for both weights, which is a highly desirable result as our loss function is also set to be MAE. Despite the fact that each of the base predictors has a larger

error compared to the zero predictor, the ensemble predictor successfully beats the zero predictor. Additionally, the random forest predictor also outperforms other predictors by a larger margin in terms of MSE, which is within our expectation as both MSE and MAE are unscaled errors. Also, in terms of prediction bias, we observe that our predictor works better than the zero predictor under both weights. On the other hand, the bias of the base estimators are less robust under the two scenarios, as the size and sign of the bias change drastically with different weights.

To verify the real time performance of our method, we ran the experiment on the testing set where the combination of hyperparameters is the same as we used in the validation set. Table 3.3 and 3.4 list the prediction errors with equal and market cap weights respectively. It can be found that the random forest predictor has the least MAE and MSE among all predictors. This indicates that the random forest learns the intrinsic relationship between the base predictors and the realized volatility, and reduces the prediction errors effectively. Additionally, we find that the prediction errors in the testing set are significantly smaller compared to those in the validation set, which may probably be due to the fact that the validation set contains several high volatile periods, while the market is relatively stable in the testing set corresponding to the recent years.

3.4.2 Crisis and Non-Crisis Period

As can be seen in the previous section, the volatility level may significantly influence the prediction errors. Thus, in this section, we compare the performance of our method under financial crisis period with high volatility, and non-crisis period corresponding to low volatility.

We took the data from July 1st, 2008 to December 31st, 2008 as the financial crisis period and ran our model. The prediction errors can be found in Table 3.5 and 3.6. The

testing data in the previous section can be seen as an example for non-crisis period. The prediction result can be seen in Table 3.3 and 3.4. GARCH model and NoVas models are not used in this section as the training period occupies almost all of the crisis period. It is obvious from the table that for both periods, our ensemble method exhibits a steady improvement compared to other methods, especially in the non-crisis period. What is more, we find that the prediction errors are significantly larger in the financial crisis period.

3.5 Conclusions

As to prediction performances, we conclude that the random forest predictor outperforms all other predictors in terms of MAE and MSE, and the improvement is highly robust under different market environments. Additionally, the improvement is more significant during non-crisis period when the volatility is lower. This is reasonable as abrupt and extreme changes in volatility can take place during the financial crisis which are not predictable, but machine learning methods only work well when the in-sample and out-of-sample data are similar. Another interesting finding is that all base predictors underperform the zero predictor, which indicates the intrinsic deficiency within their design. As a matter of fact, no single base predictor works well under all circumstances, but the random forest can effectively combine them to yield a better predictor.

On the other hand, we also have some interesting findings in hyperparameter tuning. In the first place, we observed that subsample size, i.e. the number of rows used to build each decision tree, plays a key role in the performance of the method. It has been found that the predictive power of the random forest reaches the peak when the subsample size is around 10%. When the subsample size is too large, the decision trees trained in the

random forest will be similar to each other, as the number of features is rather smaller in our case. On the contrary, a decision tree trained with a dataset which is too small in size may not have a good generalization ability. Thus it is vital to find a balance point between the two issues. Second, we conclude that the number of features to build each decision tree should be chosen such that the number of different combination of features is maximized. Again this is to build less correlated trees so that the variance of the random forest predictor can be largely reduced.

3.6 Exhibits for Chapter Three

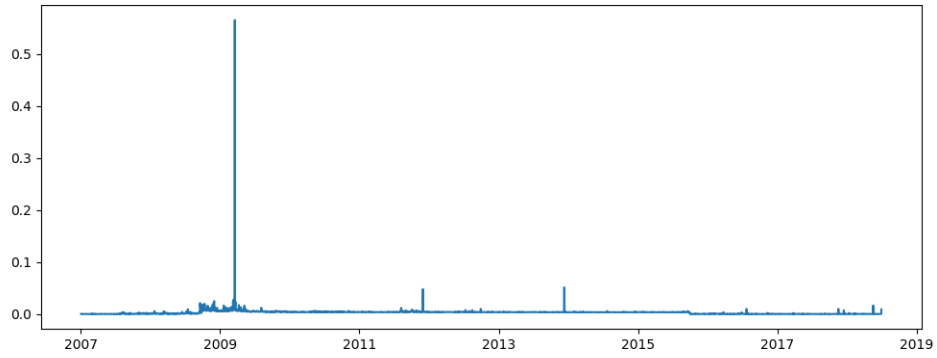


Figure 3.1: Averaged Volatility of Stocks in the Finance Sector with Equal Weights

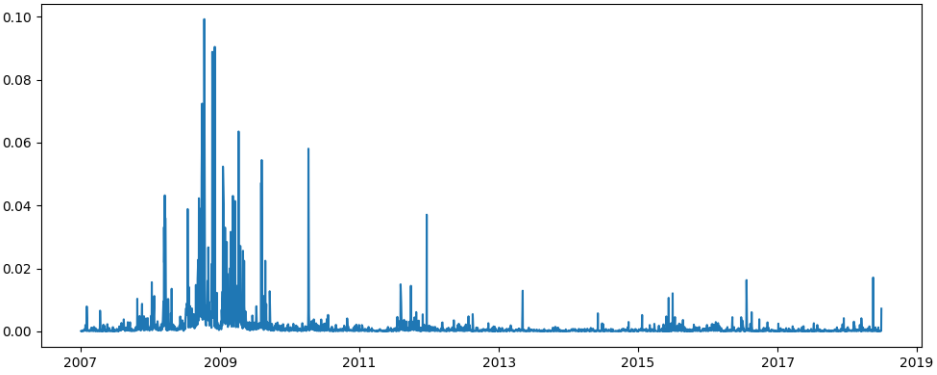


Figure 3.2: Averaged Volatility of Stocks in the Finance Sector with Market Cap Weights

Table 3.1: Prediction Errors with Equal Weights: Validation Set

Criterion	MAE	MSE	Bias
Parkinson	0.001601	0.001460	0.000161
Garman Klass	0.001460	0.001385	-8.238e-05
Rogers Satchell	0.001636	0.005169	0.0001009
Random Forest	0.001138	2.786e-05	-0.0008442
Zero predictor	0.001227	3.116e-05	-0.001227
GARCH(1,1)	>1	>1	>1
Simple NoVas	>1	>1	>1
Exponential NoVas	0.001306	5.020e-05	-0.001305

Table 3.2: Prediction Errors with Market Cap Weights: Validation Set

Criterion	MAE	MSE	Bias
Parkinson	0.001679	0.0005352	-0.0001722
Garman Klass	0.001734	0.001032	-0.0001410
Rogers Satchell	0.001887	0.003748	2.171e-06
Random Forest	0.001453	4.388e-05	-0.001111
Zero predictor	0.001601	5.1360e-05	-0.001601
GARCH(1,1)	0.8249	>1	0.8207
Simple NoVas	>1	>1	>1
Exponential NoVas	0.002258	0.0001457	-0.00225

Table 3.3: Prediction Errors with Equal Weights: Testing Set

Criterion	MAE	MSE	Bias
Squared Return	0.0006646	7.629e-06	0.0001177
Parkinson	0.0005493	9.677e-06	3.511e-05
Garman Klass	0.0005397	8.321e-06	-0.0001410
Rogers Satchell	0.0005686	8.614e-06	2.171e-06
Random Forest	0.0004012	6.400e-06	-0.001111
Zero predictor	0.0004416	7.351e-06	-0.004416
GARCH(1,1)	0.01569	>1	0.1560
Simple NoVas	>1	>1	>1
Exponential NoVas	0.0004562	1.5875e-05	-0.0004558

Table 3.4: Prediction Errors with Market Cap Weights: Testing Set

Criterion	MAE	MSE	Bias
Squared Return	0.0003210	1.688e-06	6.3911e-05
Parkinson	0.0002375	1.004e-06	-3.177e-05
Garman Klass	0.0002359	9.137e-07	-3.231e-05
Rogers Satchell	0.0002388	9.265e-07	-3.002e-05
Random Forest	0.0002046	7.953e-07	-0.0001298
Zero predictor	0.0002202	9.978e-07	-0.0002202
GARCH(1,1)	0.01395	>1	0.0132
Simple NoVas	>1	>1	>1
Exponential NoVas	0.0003538	5.866e-06	-0.0003538

Table 3.5: Prediction Errors with Equal Weights: 2008 Financial Crisis

Criterion	MAE	MSE	Bias
Squared Return	0.006293	0.0004478	0.0006723
Parkinson	0.007625	0.0008291	0.002022
Garman Klass	0.005675	0.0001965	-0.0003067
Rogers Satchell	0.006061	0.0002603	0.0001162
Random Forest	0.004777	0.0001497	-0.002677
Zero predictor	0.005001	0.0001801	-0.0005001

Table 3.6: Prediction Errors with Market Cap Weights: 2008 Financial Crisis

Criterion	MAE	MSE	Bias
Squared Return	0.01056	0.0006915	0.002198
Parkinson	0.008465	0.0003620	-0.0003795
Garman Klass	0.008775	0.0004217	-0.0002237
Rogers Satchell	0.009133	0.0004762	7.502e-05
Random Forest	0.007622	0.0002845	-0.004537
Zero predictor	0.008208	0.0003663	-0.008208

APPENDIX A

MODELS FOR CHAPTER 1

A.1 Group lasso and sparse group lasso

The group lasso, first introduced by [100], considers a general regression problem with J factors:

$$Y = \sum_{j=1}^{J} X_j \beta_j + \epsilon$$

where Y is an $n \times 1$ vector, $\epsilon \sim N_n(0, \sigma^2 I)$, X_j is an $n \times p_j$ matrix corresponding to the j-th factor and β_j is a coefficient vector of size p_j. Under high dimensional settings where the number of parameters is far larger than the number of observations, sparse estimations of the regression parameters are often required. A solution of sparse estimation for a special case was proposed by [95] when $p_1 = \cdots p_J = 1$, which is the famous lasso estimator:

$$\hat{\beta}^{LASSO}(\lambda) = \arg\min_{\beta} \|Y - X\beta\|_2^2 + \lambda \|\beta\|_{L_1}$$

The λ here is the tuning parameter, which is usually determined by cross validation; and $\|\|_{L_1}$ represents the L_1 norm of a vector. Several methods can be utilized to find the solution for the lasso optimization problem, such as subgradient method, least angel square and proximal gradient method. Despite the fact that the implementation of lasso can be fast and efficient, it can only select individual predictors, but not choose factors composing several predictors.

[100] solve this limitation of lasso by considering the following group lasso estimation:

$$\hat{\beta}^{G-LASSO}(\lambda) = \arg\min_{\beta} \frac{1}{2} \|Y - \sum_{j=1}^{J} X_j \beta_j\|_2^2 + \lambda \sum_{j=1}^{J} \gamma_j \|\beta_j\|_2$$

where γ_j is the size of j-th predictor group, and $\|\|_2$ is the L_2 norm penalty. Note that if the group size is one, then the problem is reduced to simple lasso problem. The group

lasso allows the users to select non-zero coefficient on the group level, zeroing out groups of coefficients in the linear model.

However, one notable deficiency of group lasso is that it cannot accommodate within group sparsity estimation. That is, if one of the coefficient within a group is selected to be non-zero, then the coefficients within the entire group are non-zero. This can lead to overfitting and severe error in estimation and prediction, especially when the group sizes are large. To solve this problem, [91] proposes the sparse group lasso which add a L_1 regularization term of the coefficients behind, and weigh the two penalties with a parameter α. The objective function thus becomes

$$\hat{\beta}^{SG-LASSO}(\lambda) = \arg\min_{\beta} \frac{1}{2}\|Y - \sum_{j=1}^{J} X_j\beta_j\|_2^2 + (1-\alpha)\lambda\sum_{j=1}^{J}\gamma_j\|\beta_j\|_2 + \alpha\lambda\|\beta\|_1$$

The model can be fitted by blockwise descent method, which is guaranteed to converge to the global optimum.

A.2 Existing penalty methods for VARX

Here we list the penalty functions on the AR coefficient that are compared as comparison. It is important to notice that the penalty method in the function $\mathcal{P}_x(\beta)$ corresponds to the penalty function $\mathcal{P}_y(\Phi)$.

Penalty name	$\mathcal{P}_y(\Phi)$	$\mathcal{P}_x(\beta)$
Lag	$\|\Phi\|_1$	$\|\beta\|_1$
Own/Other	$\sqrt{k}\sum_{l=1}^{p}\|\Phi_{on}^{(l)}\|_F + \sqrt{k(k-1)}\sum_{l=1}^{p}\|\Phi_{off}^{(l)}\|_F$	$\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\beta_{:,i}^{j}\|_F$
Sparse Own/other	$(1-\alpha)(\sqrt{k}\sum_{l=1}^{p}\|\Phi_{on}^{(l)}\|_F + \sqrt{k(k-1)}\sum_{l=1}^{p}\|\Phi_{off}^{(l)}\|_F) + \alpha\|\Phi\|_1$	$(1-\alpha)\sqrt{k}\sum_{j=1}^{s}\sum_{i=1}^{m}\|\beta_{:,i}^{j}\|_F + \alpha\|\beta\|_1$

For the Lag penalty, it is essentially a lasso problem, and the solution can be found by coordinate descent method. For the Own/Other penalty, the diagonal elements and off-diagonal elements of each lags in Φ are separated as independent groups. As to the

penalty function of exogenous variables, each variable are treated separately in one group for one time period. This method is analogous to group lasso and its solution can be found by block coordinate descent method. Additionally, the Sparse Own/Other penalty adapts sparse group lasso allowing for within group sparsity, and its objective function can be minimized via proximal gradient descent procedure.

APPENDIX B

PRACTICAL ISSUES

It is widely believed among professional investment practitioners that it takes a significant amount of work to transform a theoretical model into a real world strategy. Academic work needs to be supported by database maintenance, risk management, cost control, etc, in order to be used in production. This chapter discusses three practical issues in quantitative investment which are commonly encountered in the industry. The first part is about issues on financial data, while the second part concentrates on the interpretation of machine learning models for finance. Last but not least, trading constraint is discussed in the third part.

The chapter is primarily based on the author's work experience in the fall of 2019. One crucial point to be noticed is that the content of this chapter is based on the reflection of the author and by no means represents the views of the entities he is affiliated with. Additionally, no investment advice is given in this chapter.

B.1 Issues in Financial Data

Data is one of the most important parts for quantitative research and trading. Data problems may cause false discovery in backtesting, and more severely, great loss in the real time strategy. Data issues include but are not subjected to missing data, outliers, data revision, etc, and it is important to treat them with great care.

B.1.1 Missing Data

There are a variety of reasons for missing data in finance. For instance, macroeconomic data on the country level may be missing due to the unwillingness of the government to release the data or exchange holidays. On the single-name stock level, missing market data may be due to trading halt or delisting of a stock, while missing fundamental data may relate to the type of the company. Additionally, mistake from data vendors is also a significant reason for missing data in reality. Even prominent data providers like Bloomberg can have data delay or send data with wrong dates. Padding is the most widely used method for missing slow-moving data. For fast-moving data, we may use the median of corresponding data from related assets to fill the missing ones.

B.1.2 Outliers

Another commonly seen data issue in practice is outliers. An outlier may occur due to a tail event or simply a data collection error. Usually a data point is marked as outlier when its corresponding Z-score surpasses a certain threshold. Outliers can be extremely harmful for machine learning-related strategy, as the trained model can be severely biased towards the outliers, which are not representative for the entire population. One way to solve the problem is to clip the data with upper and lower bound. Although simple, this method brings about additional hyperparameters into the model and may increase the risk of overfitting. Another method to tackle the issue is to develop new algorithms that are robust to outliers. The RANSAC algorithm, for instance, is one of the most widely used methods of this type in the recent years. The algorithm trains a same model on a series of bootstrapped datasets and picks the result with the best fit. It can achieve satisfying performances when the model to be fitted is simple, such as linear regression, and the training data is small in size. However, the algorithm can be computationally

infeasible with complicated base models or large datasets. Additionally, it can only filter out outliers with a certain probability, as the number of outliers is unknown and the bootstrapped datasets are usually not exhaustive. Other robust regression methods can be found in [31].

B.1.3 Data Revision

Data revision usually refers to the situation where the data vendor sends the data on one day but changes the value subsequently. Another type of data revision is data delay, where the data is originally missing but added in a later time. Although it is often acceptable for small data revisions, it is crucial to notice that backtesting with final version of the data may contain look-ahead bias. For instance, an upward revision of the profit of a company may boom its stock price. The real-time investors, however, can only use the old value to make investment decisions prior to the revision date. On the other hand, one may use the revised value before it is available to the public in backtesting, which could lead to spurious outperformance of the corresponding strategies.

B.2 Interpretation of Machine Learning Models

It is important for researchers to understand how decisions are made by machine learning models in finance. Academic scholars, on the one hand, can benefit from model interpretation by finding new economic theories. On the other hand, investors can be more confident in the decisions made by machines as the underlying economic logic behind the model is clarified. This chapter will discuss the effect of a single feature and interaction among features, as well as the interpretation of machine learning models as a whole.

B.2.1 Interpretation of Features

There are multiple ways to understand the role a single feature plays in a machine learning model. For instance, in linear regression the p-value is often used to determine if a feature has significant effect on the prediction of the label. Additionally, feature importance has been introduced to describe the importance of features under machine learning context. One can calculate as feature importance the amount of reduction a feature has brought about in loss function with in-sample data. By contrast, researchers can also train two different models, one with the original feature and the other with a randomly permuted one, and test their difference in predictive power with out-of-sample data. If the two models have similar performances then one may conclude that the feature does not promote the prediction power of the model.

Interaction effect among features, on the other hand, should also be paid sufficient attention to. For one thing, one feature may have predictive effect on the label with the presence of another feature. In this case, the interaction term should be included in the model. For another thing, correlation among features could affect their importance measurement. In the case of linear regression, if one feature is highly dependent on others then the variance estimate of its corresponding coefficient should be adjusted upwards. Otherwise the significance level of the feature tends to be overestimated. The interaction effect between features can be measured by H-statistics introduced in [44]. Another interesting case involves random forest, where two correlated features may be selected with similar probability in different decision trees. Consequently the importance of the features are underestimated.

B.2.2 Interpretation of Complex Models

Complex machine learning model is often viewed as black box, as the actual prediction mechanism is too complicated to be understood by human being. For instance, random forest, support vector machine and modern deep learning models are usually considered black-box. By contrast, traditional machine learning models, such as linear model, decision tree, etc, are considered interpretable or white-box as it is easy to figure out how the predicted values are calculated given the features. In the field of investment, interpretable models are more preferred by investors in general as the economic intuition behind the models are more recognizable. However, complex models can describe more complicated patterns and have the potential to achieve higher out-of-sample prediction accuracy. Thus it is valuable to find connections between the two.

The most commonly used method to interpret a complex model is to approximate it with a simpler white-box model, which is sometimes called surrogate model. On the one hand, a surrogate model can be trained to approximate the entire complex model, where the columns in the original design matrix are used as features and the predictions from the complex model are taken as labels. The R-squared statistic can then be used to measure how well the surrogate model replicates the black-box model. On the other hand, one can also find an interpretable approximation for a specific observation. For this purpose, a new dataset should be simulated in the first place based on the perturbation of the observation interested, then the predicted values are acquired from the black-box model as labels. The next step is to train a white-box model based on the features and labels, where sample weights should be assigned to the perturbed observations based on their distances to the original observation.

When the number of features is large, then even the simplest linear model becomes less interpretable. To control the number of features to be included in the model, we may

apply regularization methods such as lasso and L_0 norm.

B.3 Trading Constraints

Trading constraint is an important issue to be considered for institutional investors, yet often ignored by financial scholars. As a matter of fact, many strategies proposed in academic journals may not be realist or achieve the claimed excess returns after trading constraints are imposed.

In this section, two classes of constraints are to be discussed. The first class of constraints is related to transaction cost, which usually includes commission fee, spread cost, and market impact. The first two types of fee are easy to be estimated, as brokers often charge a flat fee plus a percentage fee based on the transaction value. However, it is challenging to estimate the market impact of large transactions. Market impact can dramatically hurt the profitability of strategies for large fund, as large buy or sell market orders can drive the stock price higher or lower before the transactions are completed. One common way for large fund to reduce the cost from market impact is to add daily trading constraint and gradually reach its desired position. Besides, the second class of constraints involve risk control, which is often achieved via diversification. For instance, an experienced fund manager may impose limitation on the maximum notional position and risk contribution of a single asset, as well as the exposure of his portfolio to any single risk factor. A good example for risk exposure constraint is the beta neutral strategy, which aims to control the correlation between the market return and portfolio return. Also, it is also a wise idea to set risk target for portfolios based on ex ante volatility prediction.

BIBLIOGRAPHY

[1] D. Sornette A. Johansen. Critical crashes. *Risk*, 12(1):91–94, 1999.

[2] D. Sornette A. Johansen, O. Ledoit. Crashes as critical points. *International Journal of Theoretical and Applied Finance*, 3:219–255, 20.

[3] O. Ledoit A. Johansen, D. Sornette. Predicting financial crashes using discrete scale invariance. *Journal of Risk*, 1(4):5–32, 1999.

[4] K. Hornik A. Trapletti. *tseries: Time Series Analysis and Computational Finance*, 2018. R package version 0.10-46.

[5] Yakov Amihud. Illiquidity and stock returns: cross-section and time-series effects. *Journal of Financial Markets*, 5(1):31–56, 2002.

[6] Hamed Amini, Rama Cont, and Andreea Minca. Stress testing the resilience of financial networks. *International Journal of Theoretical and applied finance*, 15(01):1250006, 2012.

[7] Hamed Amini, Rama Cont, and Andreea Minca. Resilience to contagion in financial networks. *Mathematical Finance*, 2013.

[8] Hamed Amini and Andreea Minca. Inhomogeneous financial networks and contagious links. *Available at SSRN*, 2014.

[9] T. Andersen and T. Bollerslev. Answering the skeptics: Yes, standard volatility models do provide accurate forecasts. *International Economic Review*, 73(4), 1998.

[10] Marta Bańbura, Domenico Giannone, and Lucrezia Reichlin. Large bayesian vector auto regressions. *Journal of Applied Econometrics*, 25(1):71–92, 2010.

[11] Morten L Bech and Enghin Atalay. The topology of the federal funds market. *Physica A: Statistical Mechanics and its Applications*, 389(22):5223–5246, 2010.

[12] Itzhak Ben-David, Francesco Franzoni, and Rabih Moussawi. Hedge fund stock trading in the financial crisis of 2007–2009. *Review of Financial Studies*, 25(1):1–54, 2012.

[13] Ben S Bernanke, Jean Boivin, and Piotr Eliasz. Measuring the effects of monetary policy: a factor-augmented vector autoregressive (favar) approach. *The Quarterly Journal of Economics*, 120(1):387–422, 2005.

[14] Annika Birch, Zijun Liu, and Tomaso Aste. A counterparty risk study of the UK banking system. *Available at SSRN 2599891*, 2014.

[15] Michael Boss, Helmut Elsinger, Martin Summer, and Stefan Thurner 4. Network topology of the interbank market. *Quantitative Finance*, 4(6):677–684, 2004.

[16] Yann Braouezec and Lakshithe Wagalath. Strategic fire-sales and price-mediated contagion in the banking system. *Working paper*, 2017.

[17] Anton Braverman and Andreea Minca. Networks of common asset holdings: Aggregation and measures of vulnerability. *Working paper, available at SSRN*, 2014.

[18] L. Breiman, J. Friedman, and R. Olshen. Classification and regression trees. 1984.

[19] Christian Brownlees, Eulalia Nualart, and Yucheng Sun. Realized networks. *Forthcoming in Journal of Applied Econometrics*, 2017.

[20] Fabio Caccioli, Munik Shrestha, Cristopher Moore, and J Doyne Farmer. Stability analysis of financial contagion due to overlapping portfolios. *Journal of Banking & Finance*, 46:233–245, 2014.

[21] Neil Calkin and Marcos Lopez de Prado. Stochastic flow diagrams. *Algorithmic Finance*, 3(1):21–42, 2014.

[22] Mark M. Carhart. On persistence in mutual fund performance. *The Journal of Finance*, 52(1):57–82, 1997.

[23] Xi Chen, Seyoung Kim, Qihang Lin, Jaime G Carbonell, and Eric P Xing. Graph-structured multi-task regression and an efficient optimization method for general fused lasso. *arXiv preprint arXiv:1005.3579*, 2010.

[24] Rama Cont, Amal Moussa, et al. Network structure and systemic risk in banking systems. *Edson Bastos e, Network Structure and Systemic Risk in Banking Systems (December 1, 2010)*, 2010.

[25] Rama Cont and Lakshithe Wagalath. Running for the exit: distressed selling and endogenous correlation in financial markets. *Mathematical Finance*, 23(4):718–741, 2013.

[26] Rama Cont and Lakshithe Wagalath. Institutional investors and the dependence structure of asset returns. *Available at SSRN 2402006*, 2014.

[27] R. Cumby, S. Figlewski, and J. Hasbrouck. Forecasting volatility and correlations with egarch models. *Journal of Derivatives*, pages 51–63, 1993.

[28] E. Said D. Dickey. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71:599–607, 1984.

[29] W. Fuller D. Dickey. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74:427–431, 1979.

[30] W. Yan W. Zhou D. Sornette, R. Woodard. Clarifications to questions and criticisms on the johansen-ledoit-sornette financial bubble model. *Physica A*, 392:4417–4428, 2013.

[31] David Matteson David Ruppert. *Statistics and Data Analysis for Financial Engineering with R Examples*. Springer, 2015.

[32] Richard A Davis, Pengfei Zang, and Tian Zheng. Sparse vector autoregressive modeling. *Journal of Computational and Graphical Statistics*, 25(4):1077–1096, 2016.

[33] Marco. Lopez de Prado. *Advances in Financial Machine Learning*. Wiley, 2018.

[34] Marcos Lopez de Prado. Building diversified portfolios that outperform out-of-sample. *Journal of Portfolio Management*, 42:59–69, 2016.

[35] Victor DeMiguel, Francisco J. Nogales, and Raman Uppal. Stock return serial dependence and out-of-sample portfolio performance. *The Review of Financial Studies*, 27(4):1031, 2014.

[36] T.N. Dinh and M.T. Thai. Community detection in scale-free networks: Approximation algorithms for maximizing modularity. *Selected Areas in Communications, IEEE Journal on*, 31(6):997–1006, June 2013.

[37] Carlo Drago and Germana Scepi. Time series clustering from high dimensional data. In *Revised Selected Papers of the First International Workshop on Clustering High–Dimensional Data*, pages 72–86, 2012.

[38] Fernando Duarte and Thomas M. Eisenbach. Fire-sale spillovers and systemic risk. *FRB of New York Staff Report*, 645, 2015.

[39] H. Emec E. Gulay. Comparison of forecasting performances: Does normalization and variance stabilization method beat garch(1,1)-type models? empirical evidence from the stock market. *Journal of Forecasting*, 37:133–150, 2018.

[40] David Easley and Jon Kleinberg. *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[41] R. Engle. Autoregressive conditional heteroscedasticity with estimates of variance of united kingdom inflation. *Econometrica*, 50(4):987–1008, 1982.

[42] R. Engle. Arch: selected readings. *Oxford University Press*, 50, 1995.

[43] G.W. Evans. Pitfalls in testing for explosive bubbles in asset prices. *The American Economic Review*, 81:922–30, 1991.

[44] B. Popescu F. Jerome. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, pages 916–954, 2008.

[45] Eugene F. Fama and Kenneth R. French. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3 – 56, 1993.

[46] S. Figlewski. Forecasting volatility. *Journal of Finance*, 50:507–528, 1995.

[47] William Fung and David Hsieh. Extracting portable alphas from equity long-short hedge funds. *working paper*, 2004.

[48] G. Wozniak G. Larson. Market timing can work in the real world. *Journal of Portfolio Management*, 21, 1995.

[49] M. Garman and M. Klass. On the estimation of price volatility from historical data. *Journal of Business*, pages 67–78, 1980.

[50] Gopinath and A Ramesh. *Introduction to wavelets and wavelet transforms :*. Prentice-Hall, 1998.

[51] C.W.J. Granger and P. Newbold. Spurious regressions in econometrics. *Journal of Econometrics*, 1974.

[52] Robin Greenwood, Augustin Landier, and David Thesmar. Vulnerable banks. *Journal of Financial Economics*, 115(3):471–485, 2015.

[53] L.C. Hsieh. Investment analysis under the p s y bubble. *Dissertations Paper, National Chung Hsing University, National Digital Library of Thesis and Dissertations in Taiwan*, 2017.

[54] Nan-Jung Hsu, Hung-Lin Hung, and Ya-Mei Chang. Subset selection for vector autoregressive processes using lasso. *Computational Statistics & Data Analysis*, 52(7):3645–3657, 2008.

[55] etc J. Stefani, O. Caelen. Machine learning for multi-step ahead foresting of volatility proxies. *Mining data for financial applications (Conference Paper)*, 2017.

[56] S. Chan J. Wang. Stock market trading rule discovery using two-layer bias decision tree. *Expert Systems with Applications*, 30:605–611, 2006.

[57] Narasimhan Jegadeesh. Evidence of predictable behavior of security returns. *The Journal of Finance*, 45(3):881–898, 1990.

[58] Narasimhan Jegadeesh and Sheridan Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance*, 48(1):65–91, 1993.

[59] P. Jorion. Predicting volatility in the foreign exchange market. *Journal of Finance*, 50:507–528, 1995.

[60] Hyun-Joo Kim, Youngki Lee, Byungnam Kahng, and In mook Kim. Weighted scale-free network in financial correlations. *Journal of the Physical Society of Japan*, 71(9):2133–2136, 2002.

[61] Gary M Koop. Forecasting with medium and large bayesian vars. *Journal of Applied Econometrics*, 28(2):177–203, 2013.

[62] A. Kumar. Who gambles in the stock market? *Journal of Finance*, 64(4), 2009.

[63] Rogers L.C.G and S. Satchell. Estimating variance from high, low, and closing prices. *Annals of Applied Probability*, pages 504–512, 1991.

[64] Robert B Litterman et al. Techniques of forecasting using vector autoregressions. Technical report, 1979.

[65] C. Luong and N. Dokuchaev. Forecasting of realised volatility with the random forests algorithm. *Journal of Risk and Financial Management*, 11(4), 2018.

[66] et.al M. Ahmad. Performance of moving average investment timing strategy in uk stock market: Individual stocks versus portfolios. *Journal of Economic and Social Studies*, 7(2), 2018.

[67] N. Hespeels R. Gryp M. Balling, D.V.d. Poel. Evaluating multiple classifiers for stock price direction prediction. *Journal of Expert Systems with Applications*, November:7046–7056, 2015.

[68] L. Jin M. Yiu, J. Yu. Detecting bubbles in hong kong residential property market. *Journal of Asian Economics*, 28:115–124, 2012.

[69] G S Maddala and Shaowen Wu. A comparative study of unit root tests with panel data and a new simple test. *Oxford Bulletin of Economics and Statistics*, 61(0):631–52, Special I 1999.

[70] A Minca. *Mathematical modeling of default contagion*. PhD thesis, Universite Paris VI (Pierre et Marie Curie), 2011.

[71] Andreea Minca. Networks of common asset holdings: Aggregation and measures of vulnerability. *Available at SSRN 2379669*, 2014.

[72] Camelia Minoiu and Javier A Reyes. Network analysis of global banking: 1978-2009. 2011.

[73] Pablo Montero and José A. Vilar. TSclust: An R package for time series clustering. *Journal of Statistical Software*, 62(1):1–43, 2014.

[74] S. Moore. Busting the myth of market timing. *Forbes*, 2016.

[75] Vito M. R. Muggeo. Estimating regression models with unknown break-points. *Statistics in Medicine*, 22(19):3055–3071, 2003.

[76] Vito M. R. Muggeo. segmented: An r package to fit regression models with broken-line relationships. *R News*, 8(1):20–25, 2008.

[77] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[78] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[79] W Nicholson, D Matteson, and J Bien. Varx-l: Structured regularization for large vector autoregressions with exogenous variables. *International Journal of Forecasting*, 2017.

[80] et.al P. Philips. Testing for multiple bubbles: Historical episodes of exuberance and collapse in the s&p 500. *International Economic Review*, 56(4):1043–1078, 2015.

[81] E. S. Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.

[82] E. S. Page. Cumulative sum charts. *Technometrics*, 3:1–9, 1961.

[83] M. Parkinson. The extreme value method for estimating the variance of the rate of return. *Journal of Business*, pages 61–65, 1980.

[84] D. Politis. Model-free and model-based volatility prediction. *Journal of Financial Econometrics*, 5:358–359, 2007.

[85] et.al P.Philips. Explosive behavior in the 1990s nasdaq: When did exuberance escalate asset values? *International Economic Review*, 52:201–226, 2011.

[86] J. Li S. Ge Q. Qin, Q. Wang. Linear and nonlinear trading models with gradient boosted random forests and application to singapore stock market. *Journal of Intelligent Learning Systems and Applications*, 5:1–10, 2013.

[87] Sadka. Liquidity risk and the cross-section of hedge-fund returns. *Journal of Financial Economics*, 98(1):54 – 71, 2010.

[88] Eric Schaanning. *Fire sales and systemic risk in financial networks*. PhD thesis, Imperial College London, 2016.

[89] P. Shen. Market timing strategies that worked. *Journal of Portfolio Management*, 29(2), 2003.

[90] Ali Shojaie and George Michailidis. Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523, 2010.

[91] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[92] Song Song and Peter J Bickel. Large vector auto regressions. *arXiv preprint arXiv:1106.3915*, 2011.

[93] D. Sornette. *Why Stock Markets Crash*. Princeton University, 2003.

[94] J. Klein T. Feldman, A. Jung. Buy and hold versus timing strategies: The winner is... *Journal of Portfolio Management*, 42(1), 2015.

[95] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[96] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[97] D. Sornette W. Yan, R. Woodard. Diagnosis and prediction of market rebounds in financial markets. *Physica A*, 391:1361–1380, 2012.

[98] Wolf Wagner. Diversification at financial institutions and systemic crises. *Journal of Financial Intermediation*, 19(3):373 – 386, 2010.

[99] D. Yang and Q. Zhang. Drift-independent volatility estimation based on high, low, open, and close prices. *Journal of Business*, 73:477–491, 2000.

[100] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[101] S. Wang Z. Liu, D. Han. Testing bubbles: Exuberance and collapse in the shanghai a-share stock market. *China's New Sources of Economic Growth: Vol. 1: Reform, Resources and Climate Change*, pages 247–270, 2016.

[102] Hui Zhang, Tu Bao Ho, Yang Zhang, and Mao Song Lin. Unsupervised feature extraction for time series clustering using orthogonal wavelet transform. *Informatica*, 30(3):305–319, 2006.