# Sparse Models in High-Dimensional Dependence Modelling and Index Tracking

by

Dezhao Han

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Actuarial Science

Waterloo, Ontario, Canada, 2017

## Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

**Abstract**

This thesis is divided into two parts. The first part proposes parsimonious models to the vine copula. The second part is devoted to the index tracking problem.

Vine copulas provide a flexible tool to capture asymmetry in modelling multivariate distributions. Nevertheless, the computational expense of its flexibility increases exponentially as the dimension of the joint distribution grows. To alleviate this issue, the simplifying assumption (SA) is commonly adopted in specific applications of vine copula models. In order to relax SA, Chapter 2 proposes generalized linear models (GLMs) to model parameters in conditional bivariate copulas. In the spirit of the principle of parsimony, a regularization methodology is developed to control the number of parameters. This leads to sparse vine copula models. The conventional vine copula with the SA, the proposed GLM-based vine copula and the sparse vine copula are applied to several financial datasets. Empirical results show that the proposed models in this chapter outperform the one with SA significantly in terms of the Bayesian information criterion.

Index tracking is a dominant method among passive investment strategies. It attempts to reproduce the return of stock-market indices. Chapter 3 focuses on selecting stocks to construct tracking portfolios. In order to do that, principal component analysis (PCA) is applied via a two-step procedure. In the first step, the index return is expressed as a function of the principal components (PCs) of stock returns, and a subset of PCs is selected according to Sobol's total sensitivity index. In the second step, a subset of stocks, which is most "similar" to those selected PCs, is detected. This similarity is measured by Yanai's generalized coefficient of determination, the distance correlation, or Heller-Heller-Gorfine test statistics. Given selected stocks, their weights in the tracking portfolio can be determined by minimizing a specific tracking error. Compared with existing methods, constructing tracking portfolios based on stocks selected by this PCA-based method is more computationally efficient and comparably effective at minimizing the tracking error.

When the number of index components is large, it is too computationally demanding to apply methods in Chapter 3 or most of existing methods, such as those relying on mixed-integer quadratic programming. In Chapter 4, factor models are used to describe

stock returns. Under this assumption, the tracking error is partitioned into two parts: one depends on common economic factors, and the other depends on idiosyncratic risks. According to this partition, a 2-stage method is introduced to construct tracking portfolios by minimizing the tracking error. Stage 1 relies on a mixed-integer linear programming to identify stocks that are able to reduce factors' impacts on the tracking error, and Stage 2 determines weights of identified stocks by minimizing the tracking error. This 2-stage method efficiently constructs tracking portfolios benchmarked to indices with thousands of components. It reduces out-of-sample tracking errors significantly.

In Chapter 5, the index tracking problem is solved by repeatedly solving one-period tracking problems. Each one-period tracking strategy is determined by a quadratic optimization with the $L_1$-regularization on asset weights. This formulation considers transaction costs and other practical constraints. Since the true joint distribution of financial returns is usually unknown, we solve one-period tracking problems under empirical distributions. With the $L_1$-regularization on asset weights, our one-period tracking strategy enjoys persistent properties in the high-dimensional setting. More specifically, the variable number $d = d(n) = O(n^\alpha)$, where $n$ is the sample size and $\alpha > 1$. Simulation studies are carried out to support our one-period tracking strategy's performance with finite samples. Applications on real financial data provide evidence that, in dealing with one-period tracking, this tracking strategy outperforms the $L_q$-penalty tracking method in terms of tracking performance and computational efficiency. In terms of multi-period tracking, this proposed method outperforms the full-replication strategy.

**Acknowledgements**

I am truly indebted and grateful to my supervisors Professor Ken Seng Tan and Professor Chengguo Weng for their valuable guidance and support throughout this thesis. Without their assistance, this work would not have been completed.

Thanks to my committee members Professor Matt Davison, Professor Alan Huang, Professor David Saunders, and Professor Tony Wirjanto for their insightful suggestions on this thesis.

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Motivations for Topics in this Thesis

Financial institutions usually hold myriad assets, and at the same time they undertake a vast number of risks. In order to achieve excellent business performance, financial institutions need expertise to manage a great number of assets and the exposed risks. This is also a requirement from their stakeholders.

Market regulators require financial institutions to model the dependence structure among their risks. For example, since the second Basel Accord ([13, Part 2]), banks are encouraged to maintain an economic capital which is calculated from their market risk, credit risk, and operational risk. Since each of these three major risks consists of many subcategorized risks, banks usually establish a *high-dimensional* joint distribution to quantitatively model their risks, and then economic capital is derived from this joint distribution.

From the shareholders' point of view, financial institutions are expected to increase companies' values as much as possible. Asset management plays an important role for financial institutions to meet that objective. Among different assets, such as commodities, fixed-income products, equities, real estate, *etc.*, this thesis focuses on equity investment management which is one of the key components of institutional asset management ([89, p.408]). A good equity investment relies on wise decisions on selecting stocks from *numerous* international or domestic equities and allocating funds among selected stocks.

However, it is neither worthwhile nor technically possible for financial institutions to pay detailed attention to each of their risks or each equity in the world. Due to different characteristics of their asset portfolios, financial institutions assign priorities to their major or most risky assets. Traditionally, identifying important risk drivers is based on business savvy, such as experts' experience and acumen. Nowadays, the information explosion makes these traditional methods too expensive and time-consuming. In response, financial institutions turn to embrace data-driven or quantitative methods ([11]).

This thesis is devoted to establishing sparse models for dependence modelling and portfolio management via data-driven methods. It helps financial institutions to efficiently (in terms of time and accuracy) identify influential dependence structures and select valuable equities in which to invest. More specifically, this thesis is divided into two parts. Chapter 2 improves the vine copula, a flexible method to model high-dimensional dependence structures. Chapters 3 - 5 focus on constructing tracking portfolios to reproduce returns of stock-market indices, which is a dominant method of passive equity investment strategies ([89, p.410], [108]). In subsequent parts of this thesis, investment only refers to equity investment, unless otherwise stated.

## 1.1 Sparse models in High-Dimensional Dependence Modelling

Dependence modelling plays a pivotal role in risk management, for example calculating economic capital. In most cases, the enterprise-level risk is aggregated from numerous dependent risk factors, so that an accurate modelling of the inter-relationship among these risk factors is the key to prudent risk management. The copula method is a popular approach to model dependence ([39]). One of its virtues is to model a joint distribution via two separate steps. The first step determines appropriate marginal distributions. The second step seeks an appropriate copula function to describe the dependence structure. Techniques for bivariate copulas are relatively mature, but high dimensional copulas are still under development. The multivariate Gaussian copula has been widely used in portfolio selection, credit risk management as well as many other applications in finance; see,

e.g., [25]. Despite its popularity, the Gaussian copula fails to capture some stylized facts of financial data, such as the strong tail dependence or the asymmetric dependence structure ([39]). Other elliptical copulas, particularly the Student-$t$ copula, have been proposed to capture the tail dependence, but they still fail to capture asymmetric dependence structures.

The vine copula ([9], [1]) provides a flexible tool to capture asymmetry and tail dependence in modelling multivariate distributions. Nevertheless, its flexibility is achieved at the expense of exponentially increasing the model complexity. To alleviate this issue, the simplifying assumption (SA), which is discussed later in Section 2.2, is commonly adopted in specific applications of vine copula models. In order to relax the SA, Chapter 2 proposes generalized linear models (GLMs) to describe parameters in conditional bivariate copulas. In the spirit of the principle of parsimony, a regularization methodology is developed to control the number of parameters, leading to sparse vine copula models. The conventional vine copula with the SA, the proposed GLM-based vine copula and the sparse vine copula are applied to several financial datasets. Empirical results show that proposed models in Chapter 2 outperform the one with the SA significantly in terms of the Bayesian information criterion.

## 1.2 Sparse models in Index Tracking

### 1.2.1 The Virtue of Index Tracking

In general, investment strategies can be classified as active investment strategies and passive investment strategies. Active fund managers use flexible methods to achieve high returns with low risk. Most passively managed funds, such as index funds and exchange-traded funds, aim at mimicking returns of benchmarked financial-market indices. This strategy is called index tracking. Compared with active investment strategies, passive investment strategies usually deliver higher risk-adjusted returns (in terms of Sharpe ratio or Jensen's alpha) and charge lower management fees. According to [129, p. 27], the average annual

management fee for mutual funds is 1.67 percent, while the average is 0.40 percent for exchange-traded funds.

The motivation for passive investment management originates from studies on evaluating mutual fund performance, and dates back to the introduction of Sharpe ratio ([110]) and Jensen's alpha ([72]). Empirical studies in [110] show that Sharpe ratio of the return of the Dow Jones Industrial Average is higher than the average Sharpe ratio of active mutual fund returns (before transaction costs and management expenses) studied in that paper. The outperformance of stock-market indices is reinforced in [72]. It points out that the average Jensen's alpha of active managed funds in the U.S. (both before and after transaction costs and management expenses) is negative, when these fund returns are regressed against the S&P 500 return. More granular empirical studies are carried out in [111], which show that the studied actively managed mutual funds fail to deliver significant positive relative returns on average, compared with their benchmark portfolios. According to active managers' investment style, the benchmark portfolio in [111] is a linear combination of financial indices representing different asset classes.

Even though empirical studies in the 1960s ([110], [72]) point out that stock-market indices beat the majority of active mutual funds in terms of risk-adjusted returns, stock-market indices cannot be used as investment tools. This is because they are only published numbers and do not generate any payoff themselves[1]. But ten years later, index funds came to the market in the 1970s ([89, p.412]). Empirical studies on the U.S. market in [53] show that (after expense) risk-adjusted returns of index funds tracking the S&P 500 index are higher than the average risk-adjusted return of actively managed mutual funds.

The recent boom in exchange-traded funds (ETFs) also boosts the development of index tracking methods. Due to attractive risk-adjusted returns, low management fees, and transparent objectives (which are simply tracking an index return), ETF has gained increasing popularity since it was first introduced in North America around the early 1990s ([57]). By June 2015, global ETF assets hit US$3 trillion, which has increased by 200% since 2010 ([106], [112]). Thanks to various ETFs tracking different kinds of financial

[1]Even though trading index futures could obtain index returns, but behind index futures stands the index fund to hedge them.

market indices, the idea of Sharpe's benchmark portfolios defined in [111] can be easily realized ([10]).

Index tracking plays an important role for institutional investors. Take pension funds as an example. In 2014, 50.7% of the assets managed by the Canadian Pension Plan was invested passively ([28]), 42.1% of the assets managed by the French Pensions Reserve Fund (Fonds de Reserve Pour Les Retraites) was invested passively in 2013 ([54]), and so was 86.0% of the assets managed by the Janpenese Government Pension Investment Fund in 2013 ([62]).

### 1.2.2 Constructing Tracking Portfolios via Partial Replication

Index tracking relies on a tracking portfolio to reproduce the return of a benchmark stock-market index. In order to track stock-market indices, a simple strategy is the full replication. Since information of how to calculate a stock-market index is public, at the time of construction a full replication strictly matches its asset weights to those in the index. After that, numbers of asset shares in the full replication hold still until any rebalancing. *After* construction, the full replication earning exactly the index return. However, there is always a gap between the terminal wealth of a full-replication and the terminal wealth given the initial wealth (*before* construction) earns exactly the index return. This gap is caused by the transaction cost *at* construction, and a high transaction cost leads to a large gap.

Some ETFs simply apply the full replication to track large-capitalization stock indices, such as the methodology of SPDR S&P 500 ETF, which is one of the largest ETFs benchmarked to the S&P 500 index ([119]). Stocks in the S&P 500 index are liquid large-capitalization stocks ([118]), which are easy to trade. Hence, in this case, the tracking gap of a full replication is negligible due to small transaction costs. However, small capitalization stocks are much less liquid ([80]), so that their high transaction costs usually prevent ETF managers from applying the full replication ([71]). When the full-replication is infeasible, in order to mimic an index return fund managers need to determine in which

index components to buy and the fund allocation for each selected stock ([71]). In this thesis, this methodology is called *partial replication*, which is the focus of Chapters 3-5.

Chapter 3 focuses on selecting stocks to construct tracking portfolios. Principal component analysis (PCA) is applied to select stocks via a two-step procedure. In the first step, the index return is expressed as a linear function of principal components (PCs) of stock returns, and a subset of PCs is selected according to Sobol's total sensitivity index. In the second step, a subset of stocks, which is most similar to those selected PCs, is detected. This similarity is measured by Yanai's generalized coefficient of determination, the distance correlation, or Heller-Heller-Gorfine test statistics. The weights of selected stocks in the tracking portfolio can be determined by minimizing a specific tracking error. Compared with existing methods, constructing tracking portfolios based on stocks selected by this PCA-based method is more computationally efficient and comparably effective at minimizing the tracking error.

The method of Chapter 3 is not so computationally efficient when the number of candidate stocks is very large. In order to deal with such cases, in Chapter 4 factor models are used to describe stock returns. Under this assumption, the tracking error is partitioned into two parts: one depends on common economic factors, and the other depends on idiosyncratic risks. According to this partition, a 2-stage method is introduced to construct tracking portfolios by minimizing the tracking error. Stage 1 relies on a mixed-integer linear programming to identify stocks that are able to reduce factors' impacts on the tracking error, and Stage 2 determines weights of the identified stocks by minimizing the tracking error. This 2-stage method efficiently constructs tracking portfolios benchmarked to indices with thousands of components. It reduces out-of-sample tracking error significantly.

Aiming at reducing the gap between the tracking portfolio terminal wealth and the terminal wealth given the initial wealth (*before* construction) earning exactly the index return, Chapter 5 solves the index tracking problem by repeatedly solving one-period tracking problems. Each one-period tracking strategy is determined by a quadratic optimization with the $L_1$-regularization on asset weights. This formulation addresses the stock selection and fund allocation simultaneously, and it also considers transaction costs and other practical constraints. Since the true joint distribution of financial returns is usually un-

known, this chapter solves the one-period tracking problem under empirical distributions. With the $L_1$-regularization on asset weights, the one-period tracking strategy enjoys persistent properties in the high-dimensional setting. More specifically, the variable number $d = d(n) = O(n^\alpha)$, where $n$ is the sample size and $\alpha > 1$. Simulation studies are carried out to support this one-period tracking strategy's performance with finite samples. Applications on real financial data provide evidence that, in dealing with one-period tracking, this tracking strategy outperforms the $L_q$-penalty tracking method in terms of tracking performance and computational efficiency. In terms of tracking small-capitalization stock-market indices in multi-period cases, this method outperforms the full-replication strategy.

# Chapter 2

# Vine Copula Models with GLM and Sparsity

## 2.1 Introduction

Recently, vine copulas have been proposed as powerful alternatives to classical multivariate copulas, such as multivariate elliptical copulas and Archimedean copulas. By decomposing a multivariate copula density into a product of (conditional) bivariate copula densities, the vine copula is flexible enough to capture asymmetric dependence structures as well as strong tail dependence among financial risks. The idea of vine copulas, which dates back to Joe [73] in 1996, is formally introduced by [8, 9] as a tool to organize the decomposition of a multivariate copula. Other selected works which have made important contributions to theoretical and practical aspects of vine copulas include [1] which develops a sequential estimation procedure for vine copulas; [32] which studies vine copulas in a Bayesian framework; [122] which develops a time-dependent vine copula model; [113] which proposes a vine-copula GARCH model with dynamic conditional dependence; [96] which discusses the discrete vine copulas; [64] which studies the asymptotic properties of the sequential estimators for vine copula models.

Because of the complexity of vine copula models, the simplifying assumption (SA) boosts parameter estimations of vine copulas in a more computationally efficient way. It assumes that all bivariate conditional copulas depend on the corresponding conditioning variables only through copula observations, but functional formulas of these bivariate copulas do not depend on the conditioning variables. Though some research works claim that, under certain conditions, the SA will not deteriorate the overall performance of vine copulas in describing a multivariate joint distribution ([65, 121]), numerical studies conducted by [4] suggest that SA can be too optimistic.

To relax the SA in vine copula modelling, one needs to specify a mechanism to describe the way the conditional bivariate copulas depend on those conditioning variables. One natural way is to model the copula parameters as functions of the conditioning variables. This idea is exploited by [3], where a local polynomial estimation is proposed for conditional copulas; see also [2] and [4]. Moreover, [61] estimates conditional copulas by a purely nonparametric method. While these findings signify the important role of conditioning variables, their proposed methods only work for univariate conditioning variables and extensions to the high dimensional case can be challenging due to the curse of dimensionality.

The primary objective of this chapter is to develop a parsimonious vine copula model which relaxes the SA. To accomplish this, generalized linear models (GLM) are proposed for each copula parameter to depend on the corresponding conditioning variables. Such parametric GLM based models provide an explicit way to describe how the dependence in each pair of conditioned variables relies on the conditioning variables, and the resulting models remain computationally efficient for estimation.

The flexibility of the vine copula is achieved at the expense of an exponentially increasing complexity of the resulting model. A $d$-dimensional vine copula consists of $d(d-1)/2$ (conditional) bivariate copulas and thus contains a large number of parameters for high-dimensional applications. The addition of GLM components inevitably will make a vine copula model even more complex, and thus contradicts to the principle of parsimony in statistical inference, if no further adjustment is provided.

To develop parsimonious vine copula models, this chapter develops a regularization

method to control the number of parameters, leading to sparse vine copula models. The regularization procedure relies on penalized maximum likelihood estimation (MLE) in such a way that the insignificant bivariate dependence diminishes. In this chapter, we use the penalty functions LASSO proposed by Tibshirani ([124]) and SCAD by Fan and Li ([45]), although other penalty functions can similarly be applied.

Our resulting sparse vine copula has the same function as the truncated vine copula introduced by [15], with both aiming to reduce the model complexity while retaining the most significant dependencies in a multivariate distribution. In a truncated vine copula, one needs to determine the level of tree on the vine from which the dependence is negligible and thus it is critical to explore the "significant" tree level. In our sparse vine copula, the model complexity is controlled by the tuning parameter which is associated with the penalty function used in the estimation procedure. In the specific implementation, the selection of tuning parameter can be conducted by cross-validation. As applications, the conventional vine with SA (vine-SA), sparse vine-SA, and sparse GLM-based vine (sparse vine-GLM) copulas are used to model several financial datasets. The results show that our proposed models outperform the vine-SA significantly in terms of the Bayesian information criterion.

This chapter proceeds as follows. Section 2.2 provides a brief overview about vine copulas. Section 2.3 introduces our proposed vine-GLM model and the regularization method used for developing the sparse vine copulas. Section 2.4 presents applications of the vine-SA, sparse vine-GLM, and sparse vine-SA models to several financial datasets. Section 2.5 concludes the chapter.

## 2.2 Preliminaries

A copula is a multivariate distribution $C$ with uniformly distributed marginals on $(0, 1)$. Sklar's Theorem (e.g., [94]) states that every multivariate distribution $H$ with univariate marginals $F_1, \ldots, F_d$ can be written as $H(x_1, \ldots, x_d) = C\left(F_1(x_1), \ldots, F_d(x_d)\right)$ for some appropriate $d$-dimensional copula function $C$. If $H$ is absolutely continuous and strictly increasing with univariate marginal densities $f_1, \ldots, f_d$, the chain rule implies the following

expression for its joint density function

$$h(x_1, \ldots, x_d) = c\left(F_1(x_1), \ldots, F_d(x_d)\right) \cdot \prod_{i=1}^{d} f_i(x_i), \tag{2.1}$$

where $c$ is the density of the copula $C$.

Equation (2.1) implies that the dependence structure for a random vector can be isolated from its univariate margins, and dependence modelling for a random vector boils down to specifying a joint copula function $C$ (or equivalently copula density $c$) and the appropriate forms for univariate margins. While the literature on the bivariate copula has proliferated, the research on multivariate copulas is still developing. In particular, the hierarchical copula-based structures have been recently proposed as a flexible alternative to the standard copula model. One of the most promising structures is the regular vine (R-vine) copula, of which the idea is originally proposed by Joe [73] and further explored by [8, 9, 30, 84].

An R-vine distribution entails the specification of a number of hierarchical trees where each edge is assigned with a bivariate copula. These bivariate copulas constitute the building blocks of the joint R-vine distribution. According to Definition 4.4 given in [84], an R-vine $\mathcal{V}$ on $d$ variables consists of $d-1$ trees. The $w$-th tree $T_w$ has nodes $N_w$ and edges $E_w$, where $E_w$ consists of unordered pairs of $N_w$ with no circle, $w = 1, \ldots, d-1$, satisfying three conditions:

(a) $T_1$ has nodes $N_1 = \{1, \ldots, d\}$ and edges $E_1$;

(b) For $w = 2, \ldots, d-1$, $T_w$ has nodes $N_w = \{E_{w-1}\}$ and edges $E_w$;

(c) (proximity condition) For $w = 2, \ldots, d-1$ and $\{a, b\} \in E_w$ with $a = \{a_1, a_2\}$ and $b = \{b_1, b_2\}$, it holds that $\#(a \bigcap b) = 1$.

To construct an R-vine tree with node set $\mathcal{N} = \{N_1, \ldots, N_{d-1}\}$ and edge set $\mathcal{E} = \{E_1, \ldots, E_{d-1}\}$, one associates each edge $e = \{a(e), b(e); D(e)\}$ in $E_w$ with a bivariate copula density $c_{a(e),b(e);D(e)}$, where nodes $a(e)$ and $b(e)$ are called the conditioned set, and

$D(e)$ is the conditioning set. An R-vine distribution is defined as the distribution of the random vector $\mathbf{X}$ with conditional copula density of $\left(X_{a(e)}, X_{b(e)}\right)$ given the variables $\mathbf{X}_{D(e)}$ specified as $c_{a(e),b(e);D(e)}$ for the R-vine trees with node set $\mathcal{N}$ and edge set $\mathcal{E}$. $\mathbf{X}_{D(e)}$ denotes the subvector of $\mathbf{X}$ determined by the indices in $D(e)$. Formal definitions for conditioning set and conditioned set are given in Definition 2.2 of [91].

A triplet $(\mathbf{F}, \mathcal{V}, \mathbf{B})$ is called an R-vine copula specification if $\mathbf{F} = (F_1, \ldots, F_d)$ is a vector of continuous invertible univariate distribution functions, $\mathcal{V}$ is a $d$-dimensional R-vine and $\mathbf{B} = \{B_e : e \in E_w, w = 1, \ldots, d-1\}$ is a set of copulas with $B_e$ being a bivariate copula assigned to an edge $e$ on $E_w$. According to Theorem 4.2 of [84], the joint density $h$ of $\mathbf{X}$ is uniquely determined by an R-vine copula specification as follows:

$$h(\mathbf{x}) = \prod_{i=1}^{d} f_i(x_i) \prod_{w=1}^{d-1} \prod_{e \in E_w} c_{a(e),b(e);D(e)} \left(F(x_{a(e)}|\mathbf{x}_{D(e)}),\, F(x_{b(e)}|\mathbf{x}_{D(e)})\middle|\mathbf{x}_{D(e)}\right). \tag{2.2}$$

Though the realized multivariate density $h$ is uniquely determined by a given R-vine copula specification, the representation of a multivariate density in terms of R-vine copula specification is not unique. The same multivariate density can be expressed by a large number of different vine copulas with different tree structures and orderings of variables. This follows from the fact that a multivariate distribution can be decomposed into a product of conditional bivariate distributions in a number of distinct ways; see [1] for more details and examples. Indeed, the number of possible representations increases exponentially with the dimension of the copula, among which the C-vine and D-vine structures are two particularly interesting structures commonly studied in the literature. In a C-vine structure, each tree has a root node which is linked to all the other nodes, and in a D-vine structure, nodes in any tree level can at most have two neighbours and thus every tree is flat on the vine.

As mentioned in the first section, the specific application of R-vine copula models is often accompanied with the SA, which simplifies the decomposition for the joint density

function $h(\mathbf{x})$ in (2.2) into

$$h(\mathbf{x}) = \prod_{i=1}^{d} f_i(x_i) \prod_{w=1}^{d-1} \prod_{e \in E_w} c_{a(e),b(e);D(e)} \left( F(x_{a(e)}|\mathbf{x}_{D(e)}),\, F(x_{b(e)}|\mathbf{x}_{D(e)}) \right),$$

where the original conditional copula density $c_{a(e),b(e)|D(e)}(\cdot,\, \cdot|\mathbf{x}_{D(e)})$ in (2.2) is replaced by an unconditional copula density.

It follows from the definition that the R-vine copula approach to dependence modelling involves three aspects: (1) selecting vine structure, (2) selecting bivariate copula families, and (3) estimating bivariate copula parameters. The selection of the vine structure is concerned with determining the structure of each tree on the vine. This issue is discussed in detail in [31]. In general, the basic idea is to choose an appropriate weight corresponding to each edge that measures the contribution of the associated bivariate copula to the overall dependency. A tree structure is said to be optimal if it is a maximum spanning tree in that it has the maximum sum of weights. In this chapter, we follow [31] and choose the absolute Kendall's tau as the weight variable. The maximum spanning tree can be obtained by Prim's algorithm (e.g., [27]). To determine a bivariate copula on each edge of the tree, it is common to fit the data with a set of bivariate copula candidates and choose the best one according to certain model selection criterion. Many criteria for selecting bivariate copulas in the context of vine copulas are discussed extensively in [16, Section 5.4]. The key findings of the paper are that the Akaike Information Criterion (AIC), which is defined as $\text{AIC} = 2K - 2\ln(L)$ with $K$ being the number of parameters and $L$ being the likelihood of the model, is found to be a reliable criterion. The AIC has the highest accuracy in the majority of cases, and it is even superior to the blanket goodness-of-fit test. For this reason, this chapter similarly adopts the AIC criterion for selecting the bivariate copulas.

There exist several methods for estimating a copula model. First, the conventional maximum likelihood (ML) method estimates the marginal parameters and the copula parameters simultaneously. In theory, this method gives the most efficient estimators. Nevertheless, it is commonly accompanied with a non-convex optimization over a large dimension set, and thus computationally cumbersome. Second, the so-called inference for margins (IFM) method proposed by [75] first estimates marginal parameters and then

uses the resulting parameters to estimate the copula parameters. Third, the semiparametric (SP) estimation proposed by [59] applies univariate empirical distribution functions (EDFs) to generate copula observations and then estimates copula parameters with the generated observations. The second and third methods are a two-step procedure; they separate the estimation of the copula from the univariate marginal distributions, and hence substantially reduce the computation.

In view of the complexity of a vine copula model, a two-step procedure for estimating its parameters seems more computationally tractable. [1] develops a stepwise estimation, which estimates the bivariate copulas on the same tree-level simultaneously and conducts the estimation in a top-down manner. [64] proves that the stepwise estimation is consistent and asymptotically normal, given that the copula observations are generated by the univariate EDFs. [1] also proposes a sequential estimation procedure that estimates each pair copula independently. If all the pair copulas do not share any common parameters, the sequential ML estimation is equivalent to the stepwise ML estimation. In this chapter, we will adopt the sequential ML estimation with the IFM method.

## 2.3 Vine copula with GLM and sparsity

By "vine-GLM copula" we denote as the vine copulas for which the associated conditional copulas depend on conditioning variables only via their parameters, and each copula parameter is described by a generalized linear model. The specific setup of our vine-GLM copula model is given in subsection 2.3.1, and the procedure for producing a sparse vine-GLM copula model is described in subsection 2.3.2. Subsection 2.3.3 provides some simulation studies to assess the relative efficiency of our proposed GLM-based copula models to other existing copula models.

### 2.3.1 Conditional copula with GLM

While the copulas on a vine model are all bivariate, we consider a general $d$-dimensional continuous response $\mathbf{U} = (U_1, \ldots, U_d)$ and a set of conditioning variables $\mathbf{V} = (V_1, \ldots, V_m)$.

Let $H(u_1,\ldots,u_d|\mathbf{v}) = \Pr(U_1 \le u_1,\ldots,U_d \le u_d|\mathbf{V}=\mathbf{v})$, $F_{U_i}(u_i|\mathbf{v}) = \Pr(U_i \le u_i|\mathbf{V}=\mathbf{v})$, $i=1,\ldots,d$ be the joint distribution of $\mathbf{U}|\mathbf{V}$ and marginal distribution of $U_i|\mathbf{V}$, $i=1,\ldots,d$, where $\mathbf{v}=(v_1,\ldots,v_m)$. According to Sklar's Theorem (e.g., [94, 97]), there exists a unique $d$-dimensional conditional copula $C_{\mathbf{U};\mathbf{V}}(\cdot|\cdot)$ such that

$$H(u_1,\ldots,u_d|\mathbf{v}) = C_{\mathbf{U};\mathbf{V}}\left(F_{U_1}(u_1|\mathbf{v}),\ldots,F_{U_d}(u_d|\mathbf{v})|\,\mathbf{v}\right), \quad (u_1,\ldots,u_d)\in\mathbb{R}^d.$$

Our GLM-based model assumes that the conditional copula depends on the covariates $\mathbf{V}$ via the copula parameters only, so that the joint distribution $H(u_1,\ldots,u_d|\mathbf{v})$ admits the following representation

$$H(u_1,\ldots,u_d|\mathbf{v}) = C_{\mathbf{U};\mathbf{V}}(F_{U_1|\mathbf{V}}(u_1|\mathbf{v}),\ldots,F_{U_d|\mathbf{V}}(u_d|\mathbf{v});\boldsymbol{\theta}(\mathbf{v})),$$

where $\boldsymbol{\theta}(\mathbf{v}) = (\theta_1(\mathbf{v}),\ldots,\theta_p(\mathbf{v}))$ is the conditional copula parameter vector with

$$\theta_j(\mathbf{v}) = g_j^{-1}(\eta_j(\mathbf{v})).$$

Here $g_j(\cdot)$ is a link function and $\eta_j(\cdot)$ is a calibration function for $j=1,\ldots,p$. For univariate $\mathbf{v}$, the parameter functions $\boldsymbol{\theta}(\mathbf{v})$ can be estimated by a polynomial of $\mathbf{v}$ as proposed by [3] (see also [4] and [2]). In our GLM-based conditional copulas, we consider a linear function for $\eta_j(\mathbf{v})$ with the following form

$$\eta_j(\mathbf{v}) = \beta_{0,j} + \beta_{1,j}v_1 + \cdots + \beta_{m,j}v_m, \text{ for } j=1,\ldots,p, \tag{2.3}$$

where $p$ denotes the number of parameters in the conditional copula and $\boldsymbol{\beta}_j = (\beta_{0,j},\ldots,\beta_{m,j})$ is a vector of constant coefficients. The above linear calibration function has the capability of capturing the influence of conditioning variables while still ensuring the tractability of the model. To increase the model flexibility, other conditioning variables' transformations, such as the quadratic term $v_1^2,\ldots,v_m^2$, can be incorporated to the calibration function to capture their nonlinear effects.

Let $\boldsymbol{\beta}_j = (\beta_{0,j},\beta_{1,j},\ldots,\beta_{m,j})$ for $j=1,\ldots,p$, and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_p)$ which collects all the parameters in the conditional copula $C_{\mathbf{U};\mathbf{V}}$. Given a sample $\{(\mathbf{u}_k,\mathbf{v}_k), k=1,\ldots,N\}$

for $(\mathbf{U}, \mathbf{V})$, the conditional copula model with GLM possesses a log-likelihood function of

$$l(\boldsymbol{\beta}) + \sum_{k=1}^{N} \sum_{i=1}^{d} \log \left\{ f_{U_i}(u_{i,k}|\mathbf{v}_k) \right\},$$

where $f_{U_i}(u_{i,k}|\mathbf{v}_k)$ denotes the conditional density function of the $i$th marginal for $i = 1, \ldots, d$,

$$l(\boldsymbol{\beta}) = \sum_{k=1}^{N} \log \left\{ c_{\mathbf{U};\mathbf{V}} \left( F_{U_1|\mathbf{V}}(u_{1,k}|\mathbf{v}_k), \ldots, F_{U_d|\mathbf{V}}(u_{d,k}|\mathbf{v}_k); \theta_1(\boldsymbol{\beta}_1, \mathbf{v}_k), \ldots, \theta_p(\boldsymbol{\beta}_p, \mathbf{v}_k) \right) \right\}, \quad (2.4)$$

and $\theta_j(\boldsymbol{\beta}_j, \mathbf{v}_k) = g_j^{-1}(\beta_{0,j} + \beta_{1,j} v_{1,k} + \cdots + \beta_{m,j} v_{m,k})$ for $j = 1, \ldots, p$. In principle, the MLE of $\boldsymbol{\beta}$ can be obtained by maximizing $l(\boldsymbol{\beta})$. It is worth noting that $p$ is usually smaller than or equal to two in most bivariate copulas which are usually applied in vine copula models.

Each copula parameter has a specific domain and this implies that the link function is supposed to be determined according to the same domain. Many popular bivariate copulas, as well as their parameter domains, can be found in [94] and [74]. Table 2.1 shows the choices of the link functions for the bivariate copulas that we will consider in our simulation studies and real data examples. Recall that Gaussian and Frank copulas are symmetric but without tail dependence. The Student-$t$ copula is a tail dependent symmetric copula. Clayton and Gumbel copulas have either lower or upper tail dependence. Following [95], we also consider BB1, survival BB1(sBB1), and BB7 copulas, which exhibit asymmetric tail dependence.

### 2.3.2 Sparse vine copula

Our proposed vine-GLM copula suffers from an over-fitting problem, since it has more parameters than SA-based vine copulas. In order to ensure the model complexity is kept at a reasonable level while still providing flexible dependence modelling structures, this subsection describes how sparsity can be introduced to our proposed vine-GLM copulas to attain these tradeoffs. Recall that the truncated vine copula of [15] is motivated by the

| Copula | Parameter domain | GLM link functions $g^{-1}(\cdot)$ |
|---|---|---|
| Gaussian | $\rho_G \in [-1,1]$ | $\rho_G = \tanh\left\{\frac{1}{2}(\beta_0 + \beta_1 v_1 + \cdots + \beta_m v_m)\right\}$ |
| Student-$t$ | $\rho_t \in [-1,1]$ | $\rho_t = \tanh\left\{\frac{1}{2}(\beta_{0,\rho} + \beta_{1,\rho} v_1 + \cdots + \beta_{m,\rho} v_m)\right\}$ |
| | $\nu \in (0,+\infty)$ | $\nu = \exp\left\{\beta_{0,\nu} + \beta_{1,\nu} v_1 + \cdots + \beta_{m,\nu} v_m\right\}$ |
| Clayton (strict) | $\delta \in (0,+\infty)$ | $\delta = \exp\left\{\beta_0 + \beta_1 v_1 + \cdots + \beta_m v_m\right\}$ |
| Gumbel | $\theta \in (1,+\infty)$ | $\theta = \exp\left\{\beta_0 + \beta_1 v_1 + \cdots + \beta_m v_m\right\} + 1$ |
| Frank | $\alpha \in (-\infty,+\infty)\setminus\{0\}$ | $\alpha = \beta_0 + \beta_1 v_1 + \cdots + \beta_m v_m$ |
| BB1/sBB1 | $\theta \in (0,+\infty)$ | $\theta = \exp\left\{\beta_{0,\theta} + \beta_{1,\theta} v_1 + \cdots + \beta_{m,\theta} v_m\right\}$ |
| | $\delta \in (1,+\infty)$ | $\delta = \exp\left\{\beta_{0,\delta} + \beta_{1,\delta} v_1 + \cdots + \beta_{m,\delta} v_m\right\} + 1$ |
| BB7 | $\theta \in (1,+\infty)$ | $\theta = \exp\left\{\beta_{0,\theta} + \beta_{1,\theta} v_1 + \cdots + \beta_{m,\theta} v_m\right\} + 1$ |
| | $\delta \in (0,+\infty)$ | $\delta = \exp\left\{\beta_{0,\delta} + \beta_{1,\delta} v_1 + \cdots + \beta_{m,\delta} v_m\right\}$ |

Table 2.1: The link functions for some selected bivariate copulas. The parameter $\delta$ of BB1 copula is $[1,+\infty)$, but it reduces to a Clayton copula when $\delta = 1$. Thus, only a range of $(1,+\infty)$ is assigned for the parameter $\delta$. For the same reason, a range of $(1,+\infty)$ is considered for the parameter $\theta$ in the BB7 copula.

empirical observation that the bottom trees on a vine copula model are often negligible in terms of their impact on dependence. This suggests that we can determine the "significant" tree level below which independence can be assumed.

In contrast, our sparse vine copula does not simply focus on these less significant bottom trees. Instead, it shrinks all the "insignificant" bivariate copulas on each tree-level to independent copulas, and the determination of such "insignificance" bivariate copulas is automatically carried out by a penalized estimation procedure.

The sequential estimation procedure proposed by [1] will similarly be used to develop our sparse vine copula model. The procedure estimates each bivariate copula individually. Let $\{(U_{1i}, U_{2i}), i = 1, 2, \ldots, N\}$ be independent and identically distributed (i.i.d.) observations of a bivariate copula $C(u_1, u_2; \boldsymbol{\theta})$ with copula density $c(u_1, u_2; \boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ is the vector of copula parameters. The penalized MLE is given by

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ \sum_{k=1}^{N} \ell(U_{1k}, U_{2k}; \boldsymbol{\theta}) - N \sum_{j=1}^{d} p(\theta_j) \right\}, \tag{2.5}$$

where $\ell(U_{1k}, U_{2k}; \boldsymbol{\theta}) = \ln\{c(U_{1k}, U_{2k}; \boldsymbol{\theta})\}$ is the log-likelihood, and $p(\cdot)$ is a penalty function.

Note that the above penalty function $p(\theta_j)$ is used to "detect" the insignificant parameter by shrinking the estimator $\widehat{\theta}_j$ to zero for each insignificant parameter $\theta_j$ in a linear model. Equivalently, if the penalty function is expressed as $p(\theta_j - \tilde{\theta}_j)$ for some target value $\tilde{\theta}_j$, then the estimator $\widehat{\theta}_j$ is shrunk to $\tilde{\theta}_j$.

In our numerical examples, we will use the LASSO and SCAD penalties, which are two of the most popular penalty functions in the statistical literature. The LASSO penalty is introduced by [124] for developing sparse linear regression models in a high dimensional setting. Its expression is given by

$$p_L(\theta_j) = \lambda_L |\theta_j|,$$

where the tuning parameter $\lambda_L > 0$ is imposed to control the degree to which the estimator is shrunk to zero. The SCAD penalty, which is proposed by [45], has the form

$$p_S(\theta_j) = \begin{cases} \lambda_S |\theta_j|, & |\theta_j| \leq \lambda_S, \\ -(\theta_j^2 - 2a_S\lambda_S|\theta_j| + \lambda_S^2)/\left[2(a_S - 1)\right], & \lambda_S < |\theta_j| \leq a_S\lambda_S, \\ (a_S + 1)\lambda_S^2/2, & |\theta_j| > a_S\lambda_S, \end{cases} \tag{2.6}$$

where $\lambda_S$ and $a_S$ are two tuning parameters with $\lambda_S > 0$ and $a_S > 2$.

According to [45], LASSO is better than SCAD in situations where there is too much randomness associated with the true model, while SCAD-penalized MLEs are less biased. The SCAD possesses the so-called oracle property, which roughly says that the penalized MLEs work as well as if the correct submodel were known in advance. A comprehensive review of the commonly-used penalty functions is provided in [50]. It is also worth noting that the penalized MLEs can be asymptotically normal under certain conditions as illustrated by [47]. However, in general the asymptotic normality does not apply (see [98]) and hence in our application, we will use the bootstrap method to construct the confidence intervals for the estimated parameters.

The efficiency of the penalized estimator critically depends on the choice of the tuning parameter in a penalty function since it controls the severity of the shrinkage. For a specific

application, the tuning parameters are usually determined by a cross-validation procedure, nevertheless [45] recommends $a_S = 3.7$ for the SCAD penalty. In our numerical examples, we have conducted additional studies to infer the appropriate value of $a_S$ and we similarly conclude the appropriateness of setting $a_S$ to 3.7. For this reason, we will continue to use this value for our subsequent numerical work. For other tuning parameters $\lambda_S$ or $\lambda_L$, we follow [128] and choose the tuning parameter which gives the best Bayesian information criterion (BIC) for the model. The BIC is computed by $\text{BIC} = K\ln(N) - 2\ln(L)$, where $N$ is the sample size, $K$ is the number of parameters, and $L$ is the likelihood. The BIC rule leads to a more sparse structure than the general cross-validation procedure does. As argued in [128], the general cross-validation procedure is not able to satisfactorily select the tuning parameter while the BIC-based tuning parameter is able to identify the true model consistently. For our implementations, we first conduct some pre-analysis to empirically determine the plausible ranges of the tuning parameters. Then, we obtain the penalized MLEs corresponding to each of a set of selected candidate values of $\lambda_S$ or $\lambda_L$, which will be clearly specified in each of the subsequent numerical studies. Finally, we choose $\lambda_S$ or $\lambda_L$ that gives the best BIC. For a vine copula in a large dimension, choosing the set of tuning parameters for each bivariate copula can be computationally intensive. A sub-optimal solution is to consistently use the same set of tuning parameters for all the bivariate copulas on the same level of tree.

Table 2.2 (also see Table 2.1) displays eight possible bivariate copulas which will be used to develop sparse-based vine copulas in our subsequent numerical studies. While these are not the exhaustive list of bivariate copulas, they are sufficiently representative in that they exhibit distinct distributional shapes in terms of tail dependence and asymmetry. Table 2.2 also gives the situation under which the copula degenerates to the independence structure. For example, when the target value of a copula parameter is zero, such as the Gaussian copula's $\rho$ and Clayton copula's $\delta$, the penalty term in the log-likelihood objective of (2.5) is simply $p(\theta_j)$. When the target value of a parameter is one, such as the BB1 copula's $\delta$ and Gumbel's $\theta$, the penalty term is replaced by $p(\theta_j - 1)$. For the Student-$t$ copula, the target value of $\nu$ is set to 31, a value which is large enough for the Student-$t$ copula to be close to a Gaussian copula.

To conclude this subsection, Table 2.3 summarizes the procedure for estimating sparse

| Copula | Degeneration |
|---|---|
| Student-$t$ | Student $t$ copula $\to$ Gaussian copula, as $\nu \to +\infty$. |
| Gaussian | Gaussian copula $\to$ independence copula, as $\rho_G \to 0$. |
| Clayton | Clayton copula $\to$ independence copula, as $\delta \to 0$. |
| Gumbel | Gumbel copula $\to$ independence copula, as $\theta \to 1$. |
| Frank | Frank copula $\to$ independence copula, as $\alpha \to 0$. |
| BB1 | BB1 copula $\to$ Clayton copula, as $\delta \to 1$. |
| sBB1 | sBB1 copula $\to$ independence copula, as $\theta \to 0$ and $\delta \to 1$. |
| BB7 | BB7 copula $\to$ Clayton copula, as $\theta \to 1$. |

Table 2.2: Degeneration of candidate copulas.

vine-GLM copula models. As we have pointed out in Section 2.2, we use the absolute Kendall's tau as the weight measure and apply the Prim's algorithm to obtain the maximum spanning tree at each level on the vine. Given copula observations, we apply the method given in [58] to test the independence. If the observations reject the independence assumption, we apply the AIC criterion to choose the best bivariate copula from the eight candidates in Table 2.2 for each edge of the tree, and simultaneously obtain the estimation for each selected bivariate copula by the penalized MLE procedure as outlined above.

Note that the algorithm also applies the penalized estimation scheme to a vine-GLM copula model. We continue to rely on the AIC rule for the bivariate copula selection. To compute the value of AIC, we have to estimate each candidate bivariate copula on each edge with a GLM specification, and maximize a log-likelihood with an expression similar to $l(\boldsymbol{\beta})$ given in (2.4). When the penalized estimation is applied to a vine-GLM copula model, leading to a sparse vine-GLM copula, we have two levels of shrinkage. First, we target to shrink those insignificant coefficients $\beta_s$ for $s = 1, \ldots, m$ in the GLM to be zeros to reduce model complexity. Second, given that all the coefficients $\beta_s$ for $s = 1, \ldots, m$ are indeed zeros, the intercept coefficient $\beta_0$ in the GLM is expected to be shrunk to a corresponding target so that the resulting copula parameter is attracted to a boundary value (see Table 2.2) and the underlying bivariate copula reduces to be an independent one. The specific shrinkage rule for each conditional bivariate copula-GLM is described in Table 2.4. In the table, a target value of $\beta_0$ for quite many bivariate copulas to reduce to the independence copula is $-\infty$. In our specific implementation, we replace the target value of $-\infty$ by $\log(0.001)$. Similarly, for the degrees of freedom in the Student-t copula,

we replace the target value of $+\infty$ by $\log(31)$.

| | |
|---|---|
| 1: | Input $d$-dimensional data. |
| 2: | Generate copula observations |
| 3: | **For** $w = 1, \ldots, d-1$ **do** |
| 4: | Check for proximity condition. |
| 5: | Compute empirical Kendall's tau matrix. |
| 6: | Search for the maximum spanning tree. |
| 7: | **For** each bivariate copula in the $w$-th tree level **do** |
| 8: | If tested to be independent, go to Step 16. |
| 9: | Else try a certain bivariate copula family in Table 1. |
| 10: | **For** $\lambda_L$ (or $\lambda_S$) in candidate collections **do** |
| 11: | Estimate pair copula parameters (or GLM coefficients) by the LASSO/SCAD estimators. |
| 12: | If possible, decay pair copulas according to the penalized MLEs. |
| 13: | **End for** |
| 14: | Choose $\lambda$ with the lowest BIC, take corresponding penalized MLEs as the final estimation, and compute AIC. |
| 15: | **If** all copula families have been tried, choose the one with lowest AIC, **else** go to Step 9. |
| 16: | **End for** |
| 17: | Compute pseudo observations. |
| 18: | **End for** |
| 19: | Return the density of the sparse vine-GLM specification. |

Table 2.3: The algorithm for estimation of sparse vine-GLM copula models.

### 2.3.3 Simulation studies

In order to assess the efficiency of the proposed GLM-based sparse vine copula relative to the SA-based vine copula, we consider the following carefully crafted experiments.

First we assume that the true underlying multivariate distribution is given by a five-dimensional vine copula $C$ with the corresponding tree structure depicted in Figure 2.1 and the bivariate copula families together with their parameters as shown in Table 2.5. Note that the bivariate copulas of the above vine copula comprise of Student-$t$ copulas and Frank copulas. This is motivated by their ease in generating the random samples. The

| Copula | Shrinkage targets |
|---|---|
| Gaussian | $\beta_s \to 0$ for $s = 0, 1, \ldots, m$ |
| Student-$t$ | $\beta_{s,\rho} \to 0$ for $s = 0, 1, \ldots, m$, |
| | $\beta_{s,\nu} \to 0$ for $s = 1, \ldots, m$, and $\beta_{0,\nu,0} \to +\infty$ |
| Clayton | $\beta_s \to 0$ for $s = 1, \ldots, m$, and $\beta_0 \to -\infty$ |
| Gumbel | $\beta_s \to 0$ for $s = 1, \ldots, m$, and $\beta_0 \to -\infty$ |
| Frank | $\beta_s \to 0$ for $s = 0, 1, \ldots, m$ |
| BB1/sBB1 | $\beta_{s,\theta} \to 0$ for $s = 1, \ldots, m$, and $\beta_{0,\theta} \to -\infty$, |
| | $\beta_{s,\delta} \to 0$ for $s = 1, \ldots, m$, and $\beta_{0,\delta} \to -\infty$ |
| BB7 | $\beta_{s,\theta} \to 0$ for $s = 1, \ldots, m$, and $\beta_{0,\theta} \to -\infty$, |
| | $\beta_{s,\delta} \to 0$ for $s = 1, \ldots, m$, and $\beta_{0,\delta} \to -\infty$ |

Table 2.4: Shrinkage targets for bivariate copula-GLM.

joint copula density for $(U_1, \cdots, U_5)$, as a result, has the following representation:

$$\begin{aligned} c(u_1, \ldots, u_4) \;=\; & c_{12}(u_1, u_2) c_{13}(u_1, u_3) c_{34}(u_3, u_4) c_{35}(u_3, u_5) \\ & \times c_{23;1}\big(F_{2|1}(u_2|u_1), F_{3|1}(u_3|u_1)|u_1\big) c_{14;3}\big(F_{1|3}(u_1|u_3), F_{4|3}(u_4|u_3)|u_3\big) \\ & \times c_{15;3}\big(F_{1|3}(u_1|u_3), F_{5|3}(u_5|u_3)|u_3\big) \\ & \times c_{24;13}\big(F_{2|13}(u_2|u_1, u_3), F_{4|13}(u_4|u_1, u_3)|u_1, u_3\big) \\ & \times c_{25;13}\big(F_{2|13}(u_2|u_1, u_3), F_{5|13}(u_5|u_1, u_3)|u_1, u_3\big) \\ & \times c_{45;123}\big(F_{4|123}(u_4|u_1, u_2, u_3), F_{5|13}(u_5|u_1, u_2, u_3)|u_1, u_2, u_3\big). \end{aligned} \tag{2.7}$$

A controlled data set of size 10,000 is then constructed by simulating the required samples from the above copula density. Using the procedure as outlined earlier, the SA-based vine copulas and GLM-based vine copulas are fitted to the controlled data set. Since we have complete information about the controlled data set, the efficiency of the fit can easily be gauged. The fitted tree structure is shown in Figure 2.2 and the fitted bivariate copulas (together with their fitted parameter values) are given in Tables 2.6 and 2.7 for the vine-SA model and the sparse vine-GLM model, respectively. The candidate bivariate copulas used in the estimation procedure correspond to those in Table 2.1. Note that for the GLM-based copulas, we consider both LASSO and SCAD penalty functions. Furthermore, our pre-analysis on possible tuning parameters indicates that the plausible range for the LASSO's tuning parameter $\lambda_L$ is $[10^{-6}, 10^{-4}]$ while SCAD's tuning parameter

$\lambda_S$ is $[0.05, 0.10]$. After that, ten candidates of tuning parameters are evenly selected from the respective ranges to produce ten fitted GLM-based sparse vine-copulas associated with each penalty function. The fitted set that yields the best BIC is the one that is reported in Table 2.7.

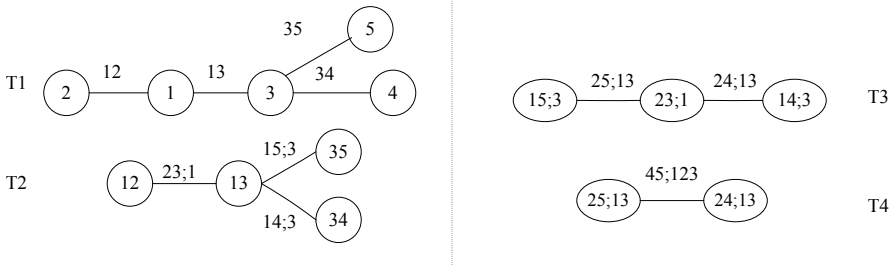



Figure 2.1: Tree structure of the true vine copula for simulation studies.

| Copula | Family | Parameters | Copula | Family | Parameters |
|---|---|---|---|---|---|
| $C_{12}$ | $t$ | $\rho_{12} = 0.45$<br>$\nu_{12} = 7.06$ | $C_{15;3}$ | Frank | $\theta_{15;3} = 0.70 - 0.4U_3$ |
| $C_{13}$ | $t$ | $\rho_{12} = 0.56$<br>$\nu_{12} = 5.75$ | $C_{14;3}$ | $t$ | $\rho_{14;3} = \tanh\left\{\frac{1}{2}(0.2254 + 0.5U_3)\right\}$<br>$\nu_{14;3} = \exp\{1.975 + 0U_3\}$ |
| $C_{34}$ | $t$ | $\rho_{12} = 0.47$<br>$\nu_{34} = 8.47$ | $C_{25;13}$ | Frank | $\theta_{25;13} = 0.45 + 0.6U_1 - 0.4U_3$ |
| $C_{35}$ | Frank | $\theta_{35} = 2.75$ | $C_{24;13}$ | Frank | $\theta_{24;13} = 1.11 + 0U_1 - 0U_3$ |
| $C_{23;1}$ | Frank | $\theta_{23;1} = 1.89 - 0.6U_1$ | $C_{45;123}$ | Frank | $\theta_{45;123} = 1.35 + 0.3U_1 + 0.4U_2 - 0.5U_3$ |

Table 2.5: Families and parameters of the bivariate copulas for vine copula simulation.

Based on the fitted results, we make the following remarks:

- It is of interest to note that all the three fitted models consistently yield the same tree structure as shown in Figure 2.2. The estimated tree structure, however, deviates from the true tree structure as depicted in Figure 2.1. The misspecification is not surprising since the method used to determine tree structures is not guaranteed to provide the true structure.
- All the three fitted models have the identical first tree in terms of the same bivariate copula families (compare Table 2.5 to Tables 2.6 and 2.7). Furthermore, the fitted parameter values are very close to their true counterparts.

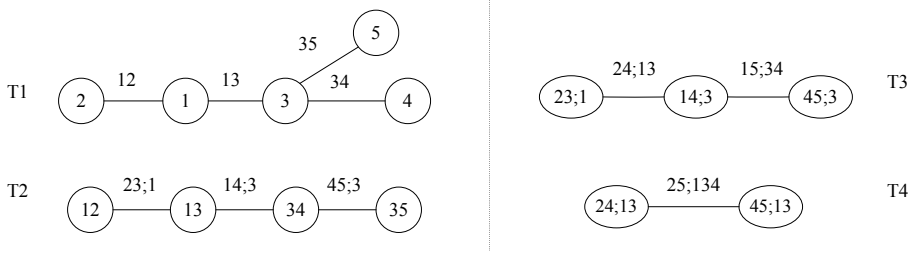



Figure 2.2: Tree structure of the fitted vine-SA and sparse vine-GLM copula.

| Copula | Family | Parameters | Copula | Family | Parameters |
|---|---|---|---|---|---|
| $C_{12}$ | $t$ | $\hat{\rho}_{12} = 0.4460$<br>$\hat{\nu}_{12} = 7.4391$ | $C_{14;3}$ | $t$ | $\hat{\rho}_{14;3} = 0.2417$<br>$\hat{\nu}_{14;3} = 7.0054$ |
| $C_{13}$ | $t$ | $\hat{\rho}_{13} = 0.5482$<br>$\hat{\nu}_{13} = 5.3925$ | $C_{45;3}$ | Frank | $\hat{\theta}_{45\|3} = 1.5908$ |
| $C_{34}$ | $t$ | $\hat{\rho}_{12} = 0.4563$<br>$\hat{\nu}_{34} = 8.0040$ | $C_{24;13}$ | Frank | $\hat{\theta}_{24;13} = 1.0638$ |
| $C_{35}$ | Frank | $\hat{\theta}_{35} = 2.7410$ | $C_{15;34}$ | Frank | $\hat{\theta}_{15;34} = 0.0472$ |
| $C_{23;1}$ | Frank | $\hat{\theta}_{23;1} = 1.5196$ | $C_{25;134}$ | Frank | $\hat{\theta}_{25;134} = 0.3474$ |

Table 2.6: Vine-SA copula parameter estimations.

The criterion based on either AIC or BIC clearly supports the superiority of the GLM-based sparse vine copulas over the SA-based vine copulas. According to [19], a difference in AIC larger than 10 is a significant support in selecting better models, and a difference between 4 and 7 provides considerable support. [79] shows that a difference in BIC larger than 5 is significant. Therefore, Table 2.8, which reports the AICs and BICs of the three fitted models, shows that both the AIC and BIC prefer our proposed model significantly to the vine-SA model. The key difference between AIC and BIC is that the latter penalizes the size of the sample data so that the larger the sample size, the heavier the penalty. They disagree when AIC chooses a more complex model than BIC does. We use AIC to select pair copulas, as a bivariate copula usually has at most two parameters. But in high-dimensional cases, such as selecting vine copula models, we focus on BIC especially when AIC and BIC prefer different models, because BIC favours a parsimonious model more than AIC in high dimensions, while AIC is likely to lead to an overfitted model. According to [34], BIC is a consistent selector that will select the true model with probability of 1 as

| | Sparse vine-GLM (LASSO) | | Sparse vine-GLM (SCAD) | |
|---|---|---|---|---|
| Copula | Family | Parameters | Family | Parameters |
| $C_{12}$ | $t$ | $\hat{\rho}_{12} = 0.4460$, $\hat{\nu}_{12} = 7.4391$ | $t$ | $\hat{\rho}_{12} = 0.4460$, $\hat{\nu}_{12} = 7.4391$ |
| $C_{13}$ | $t$ | $\hat{\rho}_{13} = 0.5482$, $\hat{\nu}_{13} = 5.3925$ | $t$ | $\hat{\rho}_{13} = 0.5482$, $\hat{\nu}_{13} = 5.3925$ |
| $C_{34}$ | $t$ | $\hat{\rho}_{34} = 0.4563$, $\hat{\nu}_{34} = 8.0040$ | $t$ | $\hat{\rho}_{34} = 0.4563$, $\hat{\nu}_{34} = 8.0040$ |
| $C_{35}$ | Frank | $\hat{\theta}_{35} = 2.7410$ | Frank | $\hat{\theta}_{35} = 2.7410$ |
| $C_{23;1}$ | Frank | $\hat{\theta}_{23;1} = 1.8802 - 0.7187U_1$ | Frank | $\hat{\theta}_{23;1} = 1.8802 - 0.7187U_1$ |
| $C_{14;3}$ | $t$ | $\hat{\rho}_{14;3} = \tanh\left\{\frac{1}{2}(0.2821 + 0.4266U_3)\right\}$ $\hat{\nu}_{14;3} = \exp\{1.9812\}$ | $t$ | $\hat{\rho}_{14;3} = \tanh\left\{\frac{0.2803+0.4302U_3}{2}\right\}$ $\hat{\nu}_{14;3} = \exp\{1.9727\}$ |
| $C_{45;3}$ | Frank | $\hat{\theta}_{45;3} = 1.7619 - 0.3429U_3$ | Frank | $\hat{\theta}_{45;3} = 1.7619 - 0.3429U_3$ |
| $C_{24;13}$ | Frank | $\hat{\theta}_{24;13} = 1.0697$ | Frank | $\hat{\theta}_{24;13} = 1.0689$ |
| $C_{15;34}$ | Frank | $\hat{\theta}_{15;34} = 0.6985 - 0.8301U_3$ | Frank | $\hat{\theta}_{15;34} = 0.6986 - 0.8306U_3$ |
| $C_{25;134}$ | Frank | $\hat{\theta}_{25;134} = 0.3348$ | Frank | $\hat{\theta}_{25;134} = 0.3371$ |

Table 2.7: Sparse vine-GLM copula estimation.

the sample size goes to infinity, while AIC might not. The legitimacy of the BIC has also been justified by [56].

Table 2.8 also shows the elapsed time of fitting the vine-SA, sparse vine-GLM with LASSO and SCAD penalties to the simulated data. All fitting procedures in this chapter are carried out with MATLAB (Version R2014a) on a PC with Intel Core i5-3210M CPU at 2.5GHz and 6.00GB memory. Fitting sparse vine-GLM models needs more time than fitting vine-SA, because 1)starting from the second tree GLM introduces more parameters to each bivariate copula, and 2) each copula type is fitted 10 times for every bivariate copula due to ten tuning parameter candidates.

In the rest of the section, we shall only focus on the LASSO penalty for fitting the sparse vine-GLM model, as the results with the SCAD are similar and almost all the same comments can be applied similarly. To provide additional insight on the fitted GLM-based sparse vine copula model, let us now focus on the fitted bivariate copula $C_{14;3}$ using the LASSO penalty. Similar comments apply to that based on the SCAD penalty.

We first examine the accuracy of the fitted parameter values relative to the true parameter values. This can be accomplished by examining the confidence intervals of the fitted values. As asymptotic normality may not apply to the present model, we resort to the bootstrap method to construct the required confidence intervals. We resample the

original controlled data set with replacement and estimate the copula parameters based on the resampled data. We repeat this procedure $1,000$ times so as to obtain $1,000$ sets of parameter estimators. We view these $1,000$ estimators as sample of the parameter estimators and use their 2.5% and 97.5% empirical quantiles to construct a 95% confidence interval. The results are shown in Table 2.9. Comparing to their true parameter values (see Table 2.5), it is reassuring that the constructed 95% confidence intervals contain the true parameter values.

Next, we are interested in the Kendall's tau of the bivariate copula $C_{14;3}$. More specifically, we are interested in how the Kendall's tau of variables $U_1$ and $U_4$ evolves along with the value of $U_3$ over the whole interval $(0, 1)$ since for such a conditional copula, the value of Kendall's tau depends on the value of the conditioning variable $U_3$. The Kendall's tau of the true model for each given value of $U_3$ can be computed based on the specified copula family and the GLM models for its parameters given in Table 2.5. The results are demonstrated by the dotted curve in Figure 2.3. To develop an estimate for such a true curve of Kendall's tau, we first fit a sparse vine-GLM model using the previously constructed data set and then compute the Kendall's tau based on the fitted parameter values for each value of $U_3$ over the interval $(0, 1)$. The results are demonstrated by the solid curve in Figure 2.3, along with the confidence bands which are similarly estimated using the bootstrap method. It is again reassuring that the true Kendall's tau falls in the 95% confidence band estimated with a sparse vine-GLM copula. The graph also reports the estimated Kendall's tau based on a vine-SA copula. In this case, the Kendall's tau is a constant and is illustrated by the dash-dot flat line since the conditional copula does not depend on the conditioning variable $U_3$.

| | number of parameters | log-likelihood | AIC | BIC | time (in seconds) |
|---|---|---|---|---|---|
| Vine-SA | 14 | 6,665.6 | -13,303 | -13,202 | 120.95 |
| Sparse vine-GLM (LASSO) | 18 | 6,698.3 | -13,361 | -13,231 | 5,772.58 |
| Sparse vine-GLM (SCAD) | 18 | 6,696.9 | -13,358 | -13,228 | 4,614.88 |

Table 2.8: Model selection: vine-SA versus sparse vine-GLM.

A final comparison is based on calculating risk measures of an investment portfolio.

| | $\hat{\beta}_{0,\rho}$ | $\hat{\beta}_{1,\rho}$ | $\hat{\beta}_{0,\nu}$ | $\hat{\beta}_{1,\nu}$ |
|---|---|---|---|---|
| Fitted value | 0.2821 | 0.4266 | 1.9812 | 0 |
| 95% CI | $(0.1931, 0.3651)$ | $(0.2803, 0.5731)$ | $(1.6693, 2.1516)$ | $(0, 0.5455)$ |

Table 2.9: 95% confidence intervals of the GLM coefficients in $C_{14;3}$.

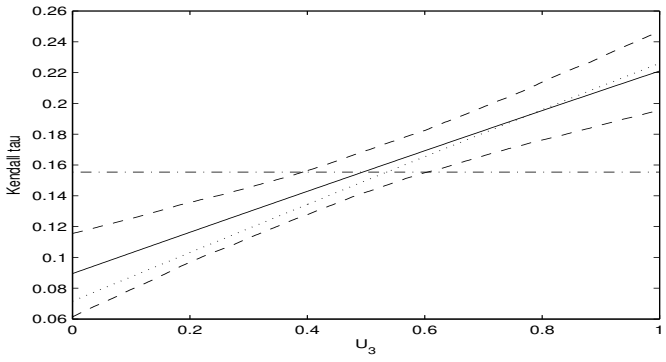



Figure 2.3: The 95% confidence band of the fitted $C_{14;3}$'s Kendall's tau. The dashed lines indicate the 95% confidence band. The solid curve is the Kendall's tau of $C_{14;3}$ in the fitted sparse vine-GLM copula, while the dash-dot line is the Kendall's tau of $C_{14;3}$ in the fitted vine-SA copula. The dotted curve is the Kendall's tau of the true model.

In particular, we are interested in Value-at-Risk (VaR) and Tail Value-at-Risk (TVaR). The VaR and TVaR of a profit-and-loss random variable $S$ at a confidence level $\alpha$ for $0 < \alpha < 1$ are defined as $\text{VaR}_\alpha(S) = \inf\{s \in \mathbb{R} : \Pr(S \leq s) \geq \alpha\}$ and $\text{TVaR}_\alpha(S) = \mathbb{E}[S|S \leq \text{VaR}_\alpha(S)]$, respectively.

Suppose that a dollar is invested in each of five (correlated) assets at time $t-1$ and that $r_{t,q}$ denotes the daily log-return of the $q$-th asset at time $t$, $q = 1, \ldots, 5$. Then, the one-day profit-and-loss variable at time $t$ of the investment portfolio is given by

$$S_t = \sum_{q=1}^{5} e^{r_{t,q}} - 5. \tag{2.8}$$

We are concerned with estimating the VaR and TVaR of the one-day profit-and-loss variable $S_t$ based on vine-SA and sparse vine-GLM (LASSO) copula models.

The Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) model is widely applied for modelling the log-return data of financial time series. In our simulation studies, each of the five daily log-return variables $r_{t,q}$ is assumed to follow a GARCH (1,1) model

$$r_{t,q} = \mu_q + \varepsilon_{t,q}, \;\; \varepsilon_{t,q} = \sigma_{t,q} z_{t,q}, \;\; \sigma^2_{t,q} = \omega_q + \alpha_q \varepsilon^2_{t-1,q} + \beta_q \sigma^2_{t-1,q}, \;\; \text{for } q = 1, \ldots, 5, \qquad (2.9)$$

with parameter values given in Table 2.10, where each innovation $z_{q,t}$ is assumed to have a Student-$t$ distribution with degree of freedom specified in the table as well. We further assume that the innovation vector $(z_{1,t}, z_{2,t}, z_{3,t}, z_{4,t}, z_{5,t})$ is subject to a dependence structure governed by the copula density (2.7) with the tree structure and bivariate copulas (both families and parameters) specified in Figure 2.1 and Table 2.5 respectively.

| | $r_{1,t}$ | $r_{2,t}$ | $r_{3,t}$ | $r_{4,t}$ | $r_{4,t}$ |
|---|---|---|---|---|---|
| $\mu_i$ | 0.0005 | 0.0002 | 0.0006 | 0.0005 | 0.006 |
| $\omega_i$ | $2.67 \times 10^{-6}$ | $3.85 \times 10^{-6}$ | $1.98 \times 10^{-6}$ | $2.52 \times 10^{-6}$ | $3.74 \times 10^{-6}$ |
| $\alpha_i$ | 0.0646 | 0.0623 | 0.0823 | 0.0767 | 0.0672 |
| $\beta_i$ | 0.9246 | 0.9346 | 0.9170 | 0.9103 | 0.9232 |
| $\varepsilon_{i,t-1}$ | 0.0074 | -0.0025 | 0.0160 | 0.0065 | 0.0079 |
| $\sigma_{i,t-1}$ | 0.0088 | 0.0170 | 0.0108 | 0.0087 | 0.0126 |
| $\nu_i$ | 4.4923 | 6.1283 | 5.8661 | 7.1731 | 6.5837 |

Table 2.10: Parameters of simulated standard-GARCH(1,1) with $t$ distributed innovation. Here, $\nu_i$ is the degree-of-freedom of a Student-$t$ distribution.

In order to evaluate the VaR and TVaR of the portfolio at time $t$, we simulate 100,000 samples of the log-return vector $(r_{1,t}, \ldots, r_{5,t})$ from model (2.9) to obtain 100,000 samples of the profit-and-loss variable $S_t$. The VaR and TVaR are then computed from these samples assuming confidence levels of 97.5% and 99%, respectively. We replicate the simulation 50 times independently and compute the average and standard deviation over these 50 estimations to produce conference intervals of the risk measures. The resulting average and the 95% confidence interval are assumed to be the correct values and are reported under the row labelled "True model" in Table 2.11. These values will be the benchmarks for the estimated VaR and TVaR from both the fitted vine-SA and sparse vine-GLM (with LASSO) models.

| | $\text{VaR}_\alpha$ | | $\text{TVaR}_\alpha$ | |
|---|---|---|---|---|
| $\alpha$ | 99% | 97.5% | 99% | 97.5% |
| Vine-SA | -0.0995 | -0.0792 | -0.1248 | -0.1026 |
| | (-0.1005, -0.0984) | (-0.0801, -0.0785) | (-0.1266, -0.1228) | (-0.1036, -0.1015) |
| Sparse vine-GLM (Lasso) | -0.1006 | -0.0799 | -0.1260 | -0.1036 |
| | ( -0.1017, -0.0997) | (-0.0806, -0.0793) | (-0.1286, -0.1239) | (-0.1051, -0.1026) |
| True model | -0.1006 | -0.0799 | -0.1260 | -0.1036 |
| | (-0.1021, -0.0994) | (-0.0807, -0.0793) | ( -0.1279, -0.1242) | ( -0.1048, -0.1025) |

Table 2.11: $\text{VaR}_\alpha$ and $\text{TVaR}_\alpha$ simulated from three models. Numbers in brackets show 95% confidence intervals.

An immediate conclusion that can be drawn from Table 2.11 is that the estimated VaR and TVaR from the sparse vine-GLM are much closer to the true values than the corresponding estimates from the vine-SA. More severely, the estimated risk measures from the vine-SA are much less negative than the corresponding true values. This implies that risk measures from the vine-SA consistently underestimate the underlying risk.

## 2.4 Application to financial data

In this section, we apply our proposed vine copula models to daily log-returns of some financial assets and compare their performance to the vine-SA copula model. We consistently use the two-step method of IFM for the estimation. The first step focuses on modelling the (parametric) univariate marginal distribution. By using the results from the first step, the second step generates the resulting vine copula observations and estimates the vine copula using the sequential estimation procedure. The general procedure of estimating the univariate marginal distributions is described in the following subsection. Subsection 2.4.2 considers an application of the sparse vine-GLM copula model to a 5-dimensional financial dataset. The impact of sparsity on vine-SA copula is illustrated in subsection 2.4.3.

### 2.4.1 Estimating univariate marginals

Determining proper univariate marginal distributions is the first and also a critical step in the IFM method since any fitting error will be carried over to fitting copulas in the second

step. We assume that the financial daily log-returns are described by the GARCH(1,1)-type models. After calibrating each marginal distribution, we use the so-called GARCH filtered transformed standardized residuals (TSRs) as the observations for vine copula estimation. Brief introductions to the GARCH(1,1)-type models and TSRs are given in Appendix A. If the univariate marginal distribution is properly estimated, we can expect that TSRs constitute a sample for a uniform random variable over $(0, 1)$. Therefore, checking uniformity of the resulting TSRs provides a reasonable way of testing the performance of the fitted univariate marginal model, as suggested by [33].

A general GARCH model consists of three components: conditional mean, conditional variance, and innovation term. In addition to the standard specification for the conditional mean and conditional variance, we consider some other more general models, as outlined in Table 2.12. Moreover, four different innovation distributions are considered as shown in Table 2.12. This setup gives 64 distinct combinations of the three components in fitting each univariate marginal data set.

| Conditional mean | Conditional variance | Innovation |
|---|---|---|
| standard | standard-GARCH(1,1) | normal |
| AR(1) | E-GARCH(1,1) | Student-$t$ |
| MA(1) | GJR-GARCH(1,1) | skewed normal |
| ARMA(1,1) | P-GARCH(1,1) | skewed-$t$ |

Table 2.12: Candidates for the three components in GARCH(1,1)-type models.

The next immediate challenge is to identify the right marginal distribution among the above 64 GARCH-type models in order to generate TSRs for estimating vine copula. Many criteria can be used to overcome this challenge. In addition to the method proposed by [33] of verifying the uniformity of the resulting TSRs, [92] suggests choosing a GARCH model by comparing certain information criteria while [67] advocates using the best forecast criterion.

In this chapter, we resort to a two-step procedure of selecting the best GARCH model. The first step applies the method of [78] to verify the non-serial correlation of TSRs. By letting $\{u_1, \ldots, u_N\}$ be the sequence of TSRs filtered from a GARCH(1,1)-type model and

$\bar{u}$ denotes its sample mean, we first calculate their autocorrelation coefficients up to 20 lags and then check whether zero falls into each of the 95% confidence intervals of these autocorrelation coefficients. We also regress each $(u_i - \bar{u})^q$ against its own 20 lags for $q = 1, 2, 3, 4$, and finally apply the so-called Lagrange multiplier test (see [78]) based on the regression results. This implies that we have two different methods of testing the non-serial correlation. While each of these tests has its own shortcomings in indicating the non-serial correlation, the combined tests could potentially enhance their performances. For the candidate models that have passed the tests from the first step, the second step involves dividing all the TSRs into 20 bins and applying the Pearson's Chi-square test to check their uniformity. The best model is selected as the one which corresponds to the smallest Chi-square test statistics. In the unlikely situation that none of these 64 candidate GARCH models passes the non-serially correlated test in the first step, we compromise to take the model that gives the best Lagrange multiplier test statistics.

| Asset | Marginal distribution |
|---|---|
| 10TNote | AR(1) - standard-GARCH(1,1) - skewed-$t$ |
| 10Bund | ARMA(1,1) - E-GARCH(1,1) - skewed-$t$ |
| Msci.world | AR(1) - E-GARCH(1,1) - skewed-$t$ |
| DAX | ARMA(1,1) - GJR-GARCH(1,1) - skewed-$t$ |
| S&P 500 | standard - GJR-GARCH(1,1) - skewed-$t$ |

Table 2.13: Fitted marginal distributions for 10TNote, 10Bund, Msci.world, DAX, and S&P 500.

### 2.4.2 Sparse vine-GLM copulas vs. vine-SA copulas

The two-step procedure described in preceding subsection is applied to the daily log-returns of the 10-year Treasure Note (10TNote) yield rate, 10-year German Bund (10Bund) yield rate, the Msci.world index, the DAX index, and the S&P 500 index for the period of January 1st, 2004 to March 4th, 2014. The data was obtained from Bloomberg. The fitted GARCH model for each log-return series is depicted in Table 2.13. The empirical log-returns exhibit asymmetric volatilities with skewed heavy tails, which are also reflected in the best fitted

GARCH model. Base on these results, the second step of the IFM method fits the vine-SA and the sparse vine-GLM copulas with LASSO and SCAD penalties. Note that this requires fitting a five-dimensional vine copula. Our pre-analysis has suggested that the reasonable ranges for the LASSO's and SCAD's tuning parameters are $\lambda_L \in [10^{-6}, 10^{-4}]$ and $\lambda_S \in [0.03, 0.08]$, respectively. We then select ten tuning parameters that are evenly distributed over the corresponding range and use these values, together with the BIC criterion, to estimate each bivariate copula on the GLM-based vine copula model. The resulting tree structure of the vine model is displayed in Figure 2.4. We first point out that all three vine-copula models (i.e. vine-SA and sparse vine-GLM copulas with LASSO and SCAD penalties) have the same tree structure. Second, the resulting structure is a D-vine copula even though our estimation procedure is conducted for a general R-vine copula.

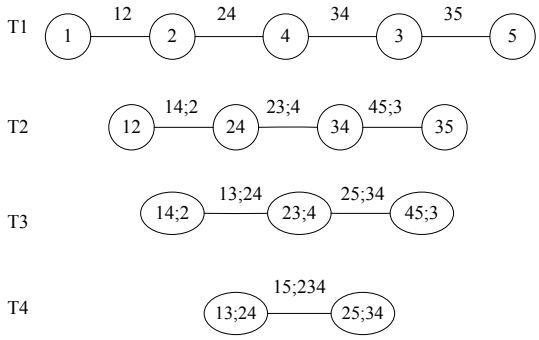



Figure 2.4: Tree structure of the fitted vine-SA and sparse vine-GLM: nodes 1, 2, 3, 4, and 5 respectively correspond to variables 10TNote, 10Bund, MSCI.world, DAX and S&P 500. $T_i$ stands for the $i$-th tree, $i = 1, \ldots, 4$.

The selected bivariate families, their parameter estimates as well as the resulting values of AICs and BICs are all reported in Tables 2.14-2.16. The comparative advantages of both GLM-based vine copula models over the vine-SA model is clearly demonstrated in the reported values of AIC and BIC in Table 2.16. Table 2.16 also shows the elapsed time of fitting these three models, and as we explained in Section 2.3.3 fitting sparse vine-GLM models needs more time. Although the vine structure is not designed for detecting economic covariates and thus in general the vine structure cannot efficiently choose a covariate with strong economic meanings, the relationship implied by the fitting results among the

| | Vine-SA | | Sparse vine-GLM (LASSO) | |
|---|---|---|---|---|
| Copula | Family | Parameters | Family | Parameters |
| $C_{12}$ | sBB1 | $\hat{\theta}_{12} = 0.2816$, $\hat{\delta}_{12} = 1.5539$ | sBB1 | $\hat{\theta}_{12} = 0.2816$, $\hat{\delta}_{12} = 1.5539$ |
| $C_{24}$ | $t$ | $\hat{\rho}_{t,24} = 0.3850$, $\hat{\nu}_{24} = 6.2217$ | $t$ | $\hat{\rho}_{t,24} = 0.3850$, $\hat{\nu}_{24} = 6.2217$ |
| $C_{34}$ | sBB1 | $\hat{\theta}_{34} = 0.4839$, $\hat{\delta}_{34} = 1.8012$ | sBB1 | $\hat{\theta}_{34} = 0.4839$, $\hat{\delta}_{34} = 1.8012$ |
| $C_{35}$ | BB1 | $\hat{\theta}_{35} = 0.7723$, $\hat{\delta}_{35} = 2.2154$ | BB1 | $\hat{\theta}_{35} = 0.7723$, $\hat{\delta}_{35} = 2.2154$ |
| $C_{14;2}$ | $t$ | $\hat{\rho}_{t,14;2} = 0.0850$, $\hat{\nu}_{14;2} = 16.2781$ | Clayton | $\hat{\delta}_{14;2} = \exp\{-2.7377 + 0.8669U_2\}$ |
| $C_{23;4}$ | $t$ | $\hat{\rho}_{t,23;4} = 0.0571$, $\hat{\nu}_{23;4} = 10.3944$ | Clayton | $\hat{\delta}_{23;4} = \exp\{-1.0945 - 6.8520U_4\}$ |
| $C_{45;3}$ | Clayton | $\hat{\delta}_{45;3} = 1.45 \times 10^{-6}$ | $t$ | $\hat{\rho}_{t,45;3} = \tanh\{\frac{1}{2}(-0.5852 + 0.3403U_3)\}$, $\hat{\nu}_{45;3} = \exp\{2.4115\}$ |
| $C_{13;24}$ | $t$ | $\hat{\rho}_{t,13;24} = 0.1308$, $\hat{\nu}_{13;24} = 9.4413$ | $t$ | $\hat{\rho}_{t,13;24} = \tanh\{\frac{0.6084-0.1991U_2-0.4994U_4}{2}\}$, $\hat{\nu}_{13;24} = \exp\{2.3564\}$ |
| $C_{25;34}$ | BB7 | $\hat{\theta}_{25;34} = 1.0273$ $\hat{\delta}_{25;34} = 0.0668$ | Clayton | $\hat{\delta}_{25;34} = \exp\{-3.3445 + 2.1530U_3 - 0.7386U_4\}$ |
| $C_{15;234}$ | $t$ | $\hat{\rho}_{15;234} = 0.1971$, $\hat{\nu}_{15;234} = 7.9043$ | $t$ | $\hat{\rho}_{t,15;234} = \tanh\{\frac{0.6184-0.2075U_2-0.1576U_3-0.0781U_4}{2}\}$ $\hat{\nu}_{15;234} = \exp\{2.1849\}$ |

Table 2.14: Fitted vine-SA and sparse vine-GLM (LASSO) models: $t$ is short for Student-$t$.

three variables of 10Bund, MSCI.world and DAX is interesting. First, both the DAX index and the 10Bond yield rate are macroeconomic indicators for the German economy, and thus they should be positively dependent. Second, as two globally important stock indices, the DAX and MSCI.world are also expected to be positively dependent. Third, based on the estimated copula $C_{23;4}$ in all the vine copulas, the DAX is chosen as the covariate. This choice makes sense since 10Bund and MSCI.world have no direct economic relationships. Fourth, though the sparse vine-GLM models with the LASSO penalty and SCAD penalty choose different bivariate copulas for the conditional pair $(2|4, 3|4)$, both models show the same effect on the dependence between the 10Bund and MSCI.world by the performance of DAX. Generally, DAX inversely affects the strength of the positive dependence between 10Bund and MSCI.world. The 10Bund and the MSCI.world exhibit a strong positive dependence when DAX performs poorly, and they exhibit a week dependence when DAX performs well. Such an observation is consistent with the asymmetric dependence in financial data.

Using the same bootstrap method as described in subsection 2.3.3, the 95% confidence

| | Sparse vine-GLM (SCAD) | |
|---|---|---|
| Copula | Family | Parameters |
| $C_{12}$ | sBB1 | $\hat{\theta}_{12} = 0.2816$, $\hat{\delta}_{12} = 1.5539$ |
| $C_{24}$ | $t$ | $\hat{\rho}_{t,24} = 0.3850$, $\hat{\nu}_{24} = 6.2217$ |
| $C_{34}$ | sBB1 | $\hat{\theta}_{34} = 0.4839$, $\hat{\delta}_{34} = 1.8012$ |
| $C_{35}$ | BB1 | $\hat{\theta}_{35} = 0.7723$, $\hat{\delta}_{35} = 2.2154$ |
| $C_{14;2}$ | $t$ | $\hat{\rho}_{t,14;2} = \tanh\left\{\frac{1}{2}\left(0.2958U_2\right)\right\}$, $\hat{\nu}_{14;2} = \exp\{3.0429 - 0.4747U_2\}$ |
| $C_{23;4}$ | $t$ | $\hat{\rho}_{t,23;4} = \tanh\left\{\frac{1}{2}\left(0.0842 - 0.0660U_4\right)\right\}$, $\hat{\nu}_{23;4} = \exp\{2.2363\}$ |
| $C_{45;3}$ | $t$ | $\hat{\rho}_{t,45;3} = \tanh\left\{\frac{1}{2}\left(-0.5844 + 0.3457U_3\right)\right\}$, $\hat{\nu}_{45;3} = \exp\{2.3959\}$ |
| $C_{13;24}$ | $t$ | $\hat{\rho}_{t,13;24} = \tanh\left\{\frac{1}{2}\left(0.5079 - 0.5060U_4\right)\right\}$, $\hat{\nu}_{13;24} = \exp\{2.3417\}$ |
| $C_{25;34}$ | Clayton | $\hat{\delta}_{25;34} = \exp\{-3.0119 + 1.1817U_3\}$ |
| $C_{15;234}$ | sBB1 | $\hat{\theta}_{15;234} = \exp\{0.3835\}$, $\hat{\delta}_{15;234} = \exp\{4.6892 - 0.7126U_2 - 1.8746U_4\} + 1$ |

Table 2.15: Fitted sparse vine-GLM (SCAD) models.

| | number of parameters | log-likelihood | AIC | BIC | time (in seconds) |
|---|---|---|---|---|---|
| Vine-SA | 19 | 4,011.0 | -7,983.9 | -7,873.1 | 46.91 |
| Sparse vine-GLM (LASSO) | 27 | 4,079.5 | -8,105.0 | -7,947.5 | 1,931.42 |
| Sparse vine-GLM (SCAD) | 26 | 4,073.0 | -8,094.0 | -7,942.3 | 1,494.82 |

Table 2.16: Estimation results of fitting vine-SA, and sparse vine-GLM copulas to the dataset with variables 10TNote, 10Bund, Msci.world, DAX and S&P 500.

intervals of the fitted GLM-based $C_{23;4}$ coefficients with the LASSO penalty are reported in Table 2.17. The 95% confidence band of the Kendall's tau is similarly demonstrated in Figure 2.5. The results for SCAD are similar and hence are omitted. We also report the Kendall's tau obtained from the fitted vine-SA model, as shown by the flat line in Figure 2.5. The figure implies that the Kendall's value based on the fitted vine-SA model falls in the 95% confidence band only when the variable $U_4$ (i.e., DAX) is within the range of about (0.1, 0.4). When DAX behaves well or extremely poorly (i.e., $U_4$ is either larger than 0.4 or below 0.1), there are significant difference in Kendall's tau between the vine-SA and the sparse vine-GLM models. For simplicity, only ranges of other conditional bivariate

copulas' Kendall's taus are given in Table 2.18. Though some lower bounds of Kendall's taus are small, they do not indicate the independence from the whole point of view.

| | $\hat{\beta}_{0,}$ | $\hat{\beta}_1$ |
|---|---|---|
| Fitted value | -1.0945 | -6.8520 |
| 95% CI | $(-1.7254, -0.4999)$ | $(-9.3520, -4.3520)$ |

Table 2.17: 95% confidence intervals of fitted GLM coefficients for $C_{23;4}$.

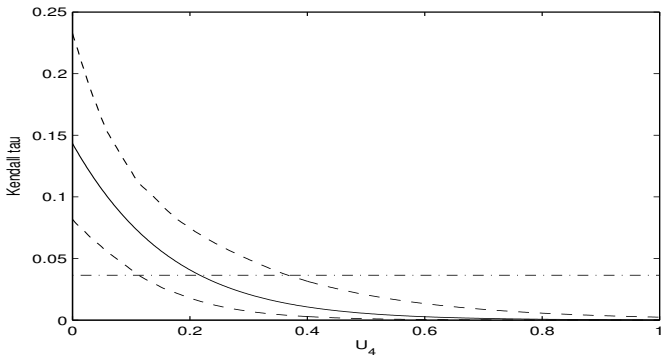



Figure 2.5: The 95% confidence band of the fitted $C_{23;4}$'s Kendall's tau. The dashed lines indicate the 95% confidence band. The solid curve is the Kendall's tau of $C_{23;4}$ in the fitted sparse vine-GLM copula, while the dash-dot line is the Kendall's tau of $C_{23;4}$ in the fitted vine-SA copula.

In order to increase the model flexibility, here quadratic terms and second-order interaction terms of conditioning variables are incorporated into the GLM calibration function. The new calibration function is modelled as a linear combination of conditioning variables and their second-order terms (quadratic and second-order interaction terms), then the 5-dimensional financial data in this section is fitted to the sparse vine-GLM models with new calibration functions. All conditioning variables are uniformly distributed on $[0, 1]$, but variances of second-order terms are different from variance of conditioning variables. In this case, before estimating penalized MLEs, we follow suggestions in [125] and standardize these second-order terms. More specifically, we scale second-order terms so that their empirical variances are equal to the variances of the conditioning variables, then the scaled second-order terms are used as variables in the new calibration function.

| Copula | Vine-SA | Sparse vine-GLM (Lasso) | Sparse vine-GLM (SCAD) |
|---|---|---|---|
| $C_{14;2}$ | 0.0542 | (0.0313, 0.0715) | (0.0000, 0.0938) |
| $C_{23;4}$ | 0.0364 | (0.0002, 0.1434) | (0.0058, 0.0268) |
| $C_{45;3}$ | $7.250 \times 10^{-5}$ | (-0.1837, -0.0778) | (-0.1834, -0.0758) |
| $C_{13;24}$ | 0.0835 | (-0.0287, 0.1907) | (0.0006, 0.1600) |
| $C_{25;34}$ | 0.0468 | (0.0084, 0.1319) | (0.0240, 0.0742) |
| $C_{15;234}$ | 0.1263 | (0.0557, 0.1938) | (0.2547, 0.2609) |

Table 2.18: Ranges of Kendall's taus of fitted conditional bivariate copulas.

The fitted tree structures with both LASSO and SCAD penalties are identical to the one in Figure 2.4, but some of the fitted bivariate copula families are different from those in Tables 2.14 and 2.15. Some other fitted results such as the number of parameters, the corresponding log-likelihood, AIC, BIC, and the elapsed time of model-fitting are shown in Table 2.19. Since sparse vine-GLM with only first-order terms in calibration functions is nested within sparse vine-GLM considering second-order terms, the fitted log-likelihood, AIC and BIC in Table 2.19 are improved compared to those in Table 2.16. However, fitting calibration functions with second-order terms are more computationally expense. In terms of this dataset, AIC and BIC in Table 2.19 are significantly better than those in Table 2.16, which suggests that adding second-order terms in calibration functions significantly improves the sparse vine-GLM model. However, for each conditional copula with $m$ conditioning variables, there are $\frac{(m+1)(m+2)}{2}$ parameters to fit in each new calibration function. Hence, considering all second-order terms in every calibration function is not feasible for high-dimensional problems where $m$ can be very large, though we carried it out in a 5-dimensional case.

| | # of parameters | log-likelihood | AIC | BIC | time (in seconds) |
|---|---|---|---|---|---|
| Sparse vine-SA (LASSO) | 31 | 4,096.8 | -8,131.6 | -7,950.7 | 4,745.96 |
| Sparse vine-SA (SCAD) | 30 | 4,094.5 | -8,128.9 | -7,953.9 | 3,672.52 |

Table 2.19: Estimation results of fitting sparse vine-GLM copulas, of which calibration functions include second-order terms, to the dataset with variables 10TNote, 10Bund, Msci.world, DAX and S&P 500.

### 2.4.3 Sparsity's impact on high-dimensional vine-SA copulas

In this subsection, we provide additional analysis on the impact of sparsity on vine copulas, especially when the dimension of the underlying copula is high. We demonstrate this by considering the log-returns of the DOW 30 companies, which make up the Dow Jones industrial average index, covering the period from January 3rd, 2005 to February 22nd, 2013. There are 25 companies for the entire coverage period, so that this example involves fitting a 25-dimensional vine copula.

We use exactly the same estimation procedure as that in the last subsection except that for simplicity we only consider the SA-based vine copulas, with and without the sparsity, and that LASSO and SCAD penalties are applied to vine-SA copulas with the appropriate ranges for the tuning parameters predetermined to be $\lambda_L \in [1 \times 10^{-5}, 5 \times 10^{-4}]$ and $\lambda_S \in [0.01, 0.15]$, respectively.

Because of the vast number of bivariate copulas and their parameter estimates, we only list the number of parameters, the corresponding log-likelihood, AIC, BIC, and the elapsed time of model-fitting in Table 2.20. The number of parameters of the sparse vine-SA models has reduced from 338 to 301 and 312, respectively, for the LASSO and SCAD penalty method. This represents at least 10% reduction in the number of parameters when compared to the conventional vine-SA model. We also remark that a majority of the bivariate copulas on the first level of tree are either the BB1 or the sBB1 copulas. Sparsity rarely appears in the first three level of trees for both the LASSO and the SCAD penalties. The sparsity becomes important at higher tree-levels. Compared with the truncated vine-SA, our parameter number reduction is due to the degradation of bivariate copulas with two parameters to those with a single parameter or even independent copulas. By using the BIC criterion, the result in Table 2.20 significantly favours the vine-SA with sparsity. As pointed at earlier, for high-dimensional model selection, it is preferred to use the BIC criterion to the AIC criterion. Fitting sparse vine-SA models needs more time than fitting vine-SA, because each copula type is fitted 10 times for every bivariate copula due to ten tuning parameter candidates.

Finally, we point out that we have similarly applied the same studies to a Stoxx 50

dataset over the period of July 8th, 2005 to December 4th, 2013. The results are similar and hence are omitted for brevity.

| | # of parameters | log-likelihood | AIC | BIC | time (in seconds) |
|---|---|---|---|---|---|
| Vine-SA | 338 | 15,667 | -30,658 | -28,756 | 775.34 |
| Sparse vine-SA (LASSO) | 301 | 15,543 | -30,488 | -28,791 | 7,267.08 |
| Sparse vine-SA (SCAD) | 312 | 15,575 | -30,526 | -28,771 | 7,231.11 |

Table 2.20: Model selection for dataset with 25 out of the Dow 30 companies.

## 2.5 Concluding remarks

The vine copula has been successfully applied in a variety of areas as a flexible tool of dependence modelling. The major technical compromise in the specific applications of vine copulas lies in the so-called simplifying assumption, which simplifies a vine model such that all the bivariate conditional copulas depend on the corresponding conditioning variables through the copula observations only, and the functional forms of these bivariate copulas do not depend on the conditioning variables. In order to relax the SA while maintaining a reasonable model complexity, we propose a generalized-linear-model-based framework to capture the effect from conditioning variables on a bivariate dependency, leading to the vine-GLM copula models. Moreover, we also develop a penalized-MLE-based regularization estimation procedure to control the complexity of vine copula models, which leads to the sparse vine copula models. Empirical studies we conducted on some financial datasets show that our proposed models with GLM and/or sparsity significantly improve the conventional vine copula model with the simplifying assumption using the criteria such as the Bayesian information criterion. In this chapter, eight bivariate copulas are considered as candidates in the specific estimation. Other bivariate copulas can be similarly analyzed with our proposed models to increase the flexibility of dependence modelling. Moreover, while the linear effect of the conditioning variables is focused on in our specific applications of the vine-GLM copula models, other transformations of the conditioning variables, such as quadratic terms, can easily be included to increase the model flexibility.

# Chapter 3

# Index Tracking using Principal Component Analysis

## 3.1 Introduction

Index tracking is a dominant method of the passive investment strategy. It constructs a tracking portfolio to reproduce the return of a benchmark stock market index. Obviously, a stock market index can be tracked by a full replication, which buys and holds all stocks that make up the index with the same weights as those in the index. When the full replication is infeasible (see more details in Section 1.2.2), many passively managed funds use a subset of stocks to construct a tracking portfolio to mimic the benchmark index return (see evidence in [71]). We refer to the problem of constructing partial replications as the *index tracking problem*.

In general, the index tracking problem should be addressed in two steps. One is identifying stocks to hold in the tracking portfolio. The other one is to compute the fund allocation to each selected stock. Focusing on minimizing the in-sample tracking error, [7] formulates the index tracking problem as a mixed-integer quadratic programming problem, and solves the "two steps" simultaneously. This paper inspires numerous studies that

explicitly exploit various mathematical optimization tools. For examples, [55] compares several tracking errors. [22] introduces another tracking error from the regression point of view, and formulates the index tracking problem as a mixed-integer linear programming problem. [105] suggests solving the index tracking problem using a hybrid programming method. For each given stock subset, stock weights are determined using quadratic programming to minimize the tracking error. The best stock subset which leads to the smallest tracking error is searched by a genetic algorithm. Most of the above methods focus on minimizing the in-sample tracking error by solving a mixed-integer quadratic programming problem, which is NP-hard (see [105]). However, it is challenging to obtain optimal solutions of a mixed-integer quadratic programming problem in an efficient way, especially when the number of index components is in the order of hundreds.

While the objective of the above-mentioned papers focuses on constructing a tracking portfolio that minimizes in-sample tracking errors, other criteria have been advocated to constructing the tracking portfolio. [102] studies the mean-variance performance of a tracking portfolio in the Markowitz framework, and this study only discusses the full replication. In terms of the partial replication, [5] studies the index tracking procedure based on the cointegration between the index level and the value of the tracking portfolio, suggesting that the value of the tracking portfolio should be cointegrated with the index level. [51] applies the clustering analysis to the index tracking problem. After stocks are clustered the authors of [51] suggest selecting one stock subjectively from each cluster. The factor model is used in [26] to address the index tracking problem. The authors of [26] suggest that the tracking portfolio should share the same factor structure with the index. However, most of these methods assume that stocks in the tracking portfolio are given, or only use naive or ad-hoc methods to select these stocks. For example, one ad-hoc approach would be to select those stocks with largest market capital.

This chapter provides a more quantitative and theoretically supported method to select stocks in tracking portfolios. In order to do so, the index return is modelled by a linear combination of stock returns plus an independent random noise. A method to identify dominant stocks is proposed based on the principal component analysis (PCA). We first decompose the index return as a function of principal components (PCs) of stock returns. According to Sobol's total sensitivity index, some essential PCs are retained to approximate

the index return, and the approximation error is controlled by Sobol's total sensitivity index. When stock returns follow a multivariate normal distribution, some analytical properties are established.

In our proposed approach, the selection of dominant stocks to construct tracking portfolios turns to be the question of choosing stocks which explain retained PCs. If the number of stocks in a tracking portfolio is pre-determined, we suggest selecting stocks that has the largest "similarity" with the retained PCs. In order to measure this similarity, [21] suggests Yanai's generalized coefficient of determination (GCD). In this chapter, we additionally recommend the distance correlation and HHG test statistics.

Given the selected stocks, determining their weights by minimizing a specific tracking error is computationally easy. When the mean square error (of the difference between the index return and the tracking portfolio return) is used as a measure of tracking error, weights are solved using quadratic programming. When the conditional value at risk (of the difference between the index return and the tracking portfolio return) is used as a measure of tracking error, weights are determined using linear programming.

The rest of this chapter is organized as follows. Section 3.2 sets up the mathematical formulation of the index tracking problem. Section 3.3 discusses the methodology to retain the significant PCs. In section 3.4, stocks in tracking portfolios are determined according to the retained PCs. In Section 3.5, some applications on real financial data are presented to support the tracking accuracy and the computational efficiency of our proposed method.

## 3.2 Formulation of the Index Tracking Problem

### 3.2.1 Introduction to Stock Market Indices

In general, a stock market index over a set of discrete times is defined as

$$I_t = \frac{1}{D} \sum_i a_i S_{t,i}, \text{ for } t = 0, 1, \ldots, \tag{3.1}$$

where $S_{t,i}$ is the price of the $i$th stock at time $t$, $D$ is the index divisor, and $a_i$ is the weight for stock $i$. If $a_i = 1$ for any $i$, this index is called a price-weighted index. Examples of price-weighted indices include the Dow Jones Industrial Average (see [116]) and the Nikkei 225. If $a_i$ is the number of outstanding shares of stock $i$, the index is called a capitalization-weighted index, such as the S&P 500 (see [117]). Based on (3.1), the index's return over the period $[t-1, t]$ is given by

$$R_t = \frac{I_t - I_{t-1}}{I_{t-1}} = \sum_i \frac{a_i S_{t,i} - a_i S_{t-1,i}}{\sum_i a_i S_{t-1,i}} = \sum_i \frac{a_i (S_{t,i} - S_{t-1,i})}{\sum_i a_i S_{t-1,i}}.$$

Let $r_{t,i} = \frac{S_{t,i} - S_{t-1,i}}{S_{t-1,i}}$ be the return of the $i$th stock over $[t-1, t]$, $R_t$ can be rewritten as

$$R_t = \sum_i \frac{a_i S_{t-1,i}}{\sum_i a_i S_{t-1,i}} r_{t,i} = \sum_i q_{t,i} r_{t,i}, \tag{3.2}$$

where $q_{t,i} = \frac{a_i S_{t-1,i}}{\sum_i a_i S_{t-1,i}}$ is the weight for stock $i$'s return at time $t$.

For many capitalization-weighted stock-market indices such as the S&P 500 index, when there is any company addition or deletion, *special* cash dividend payout, change in outstanding shares, *etc*, the index value should not jump up or drop down. In order to make the index level consistent before and after these changes, the index divisor is adjusted ([117]). Usually, these stock-market indices which include the S&P 500 index are not adjusted for *ordinary* cash dividends ([117]). Though the index divisor and outstanding shares are adjusted occasionally, they usually remain constant in several months or even one year (see [115]). In this chapter, we consider tracking a stock market index within an investment period where the index is not revised. In other words, there is no company addition/deletion, stock split, *etc*, so that $D$ and $a_i$'s do not change within the investment period.

### 3.2.2 The Index Tracking Problem

In (3.2), $R_t$ and the $r_{t,i}$'s have a linear relationship, and the weights $q_{t,i}$'s are time-varying. However, based on data of five stock-market indices (the Hang Seng index, DAX index,

FTSE index, S&P 100 index and the Nikkei 225 index) in Section 3.5, when the index return is fitted to a linear combination of its components' returns with nonnegative constant coefficients, the fitted R-squared and adjusted R-squared (see Table 3.1) are close to one. This suggests that a linear combination of stock returns with constant coefficients can explain most of the index return variance. We limit the coefficients to be nonnegative since all $q_{t,i}$'s cannot be negative. In terms of the fitting procedure, constant coefficients are estimated by least-square estimators, subject to constraints that all estimators are nonnegative and sum to 1.

| Index | R-squared | Adjusted R-squared |
|---|---|---|
| Hang Seng | 0.9958 | 0.9952 |
| DAX | 0.9416 | 0.9172 |
| FTSE | 0.9923 | 0.9889 |
| S&P100 | 0.9918 | 0.9875 |
| Nikkei 225 | 0.9982 | 0.9917 |

Table 3.1: Fitted R-squared and Adjusted R-squared for five stock-market indices

This inspires us to approximate the index return by a linear model with constant weights $q_{t,i}$ over time, but in order to compensate for the fluctuation in the original coefficients, an independent noise term is introduced. In this chapter, let $\mathbf{r}_t = (r_{t,1}, \ldots, r_{t,d})'$, and then we assume samples $(R_t, \mathbf{r}_t')$ for $t = 1, 2, \ldots, T$ are independent and identically distributed. Hence, in a generic investment period, the index return $R$ is modelled by

$$R = \mathbf{q}'\mathbf{r} + \varepsilon, \tag{3.3}$$

where the stock return vector $\mathbf{r} = (r_1, \ldots, r_d)'$ has expectation $\boldsymbol{\mu}_d$ and covariance matrix $\Sigma$. The noise $\varepsilon$ is independent of $\mathbf{r}$ and has 0 mean and finite second moment $\sigma_\varepsilon^2$. Elements of the coefficient vector $\mathbf{q} = (q_1, \ldots, q_d)'$ are constants.

The index tracking problem can be formulated as choosing a $k(\ll d)$-dimensional subset of $\mathbf{r}$, that is $\mathbf{r}^s$, and determining weights for each selected stock. The cardinality constraint, *i.e.* choosing $k$ stocks, is sometimes required by investors due to their financial budget.

The cardinality constraint is also preferred by portfolio managers, since it is impossible for them to pay detailed attention to a large number of stocks. If selected stocks are relabelled from 1 to $k$, we denote by $w_j$, $j = 1, \ldots, k$, their corresponding weights. In this chapter, we do not allow short selling stocks, that is $w_j \in [0, 1]$ for $j = 1, \ldots, k$ and $\sum_j w_j = 1$. In the U.S., there is a margin requirement for short selling stocks. The margin for short selling a stock is 50% of the market value of the borrowed stock[1], and this is a significant expense. Due to some restrictions on short-selling stocks, such as the alternative uptick rule by the U.S. Securities and Exchange Commission[2], under certain circumstances it is not easy to short sell stocks. Moreover, losses of short selling stocks are unlimited, which is too risky.

In this chapter, the selected stocks in the tracking portfolio aim at minimizing a $\rho$-distance between the index return and the tracking portfolio return, which is given by

$$\rho(R, \mathbf{w}'\mathbf{r}^s) = \sqrt{E[(R - \mathbf{w}'\mathbf{r}^s)^2]},$$

where $\mathbf{w} = (w_1, \ldots, w_k)'$. The expectation of square loss function penalizes large deviations. It is a commonly used prediction error and usually has nice analytic properties. In the following sections, we call $E[(R - \mathbf{w}'\mathbf{r}^s)^2]$ the mean square error (MSE) of a tracking portfolio. Since $\varepsilon$ is independent of $\mathbf{r}$,

$$\min_{\mathbf{w}} \rho(R, \mathbf{w}'\mathbf{r}^s) \Leftrightarrow \min_{\mathbf{w}} \rho(E[R|\mathbf{r}], \mathbf{w}'\mathbf{r}^s) \Leftrightarrow \min_{\mathbf{w}} \rho(\mathbf{q}'\mathbf{r}, \mathbf{w}'\mathbf{r}^s). \tag{3.4}$$

## 3.3 Retain Essential Principal Components

Since the variance-covariance matrix of $\mathbf{r}$, $\Sigma$, is positive semi-definite, there is a random vector $\mathbf{z} = (z_1, \ldots, z_d)'$ and a $d \times d$ orthogonal matrix $A$ such that $\Sigma = A\Lambda A'$ where $\Lambda$ is a diagonal matrix, and $\mathbf{z} = A'\mathbf{r}$. Here, $\mathbf{z}$ is called the principal components (PCs) of $\mathbf{r}$,

[1] http://www.ecfr.gov/cgi-bin/text-idx?SID=7df35b15d3a9d087dc1fbe017048f723&mc=true&node=se12.3.220_112&rgn=div8.

[2] http://www.sec.gov/news/press/2010/2010-26.htm.

and the $i$th column of $A$ is called the PC loading vector of $z_i$, for $i = 1, \ldots, d$. Diagonal elements of $\Lambda$, *i.e.* $\Lambda_1, \ldots, \Lambda_d$, are eigenvalues of $\Sigma$.

Note that we can rewrite (3.3) as $R = (\mathbf{q}'A)\mathbf{z} + \varepsilon = (\mathbf{q}^*)'\mathbf{z} + \varepsilon$. Hence,

$$\min_{\mathbf{w}} \rho(R, \mathbf{w}'\mathbf{r}^s) \Leftrightarrow \min_{\mathbf{w}} \rho(\mathbf{q}'\mathbf{r}, \mathbf{w}'\mathbf{r}^s) \Leftrightarrow \min_{\mathbf{w}} \rho\big((\mathbf{q}^*)'\mathbf{z}, \mathbf{w}'\mathbf{r}^s\big), \tag{3.5}$$

where $\mathbf{q}^* = \mathbf{q}'A$.

However, directly working on $\min_{\mathbf{w}} \rho\big((\mathbf{q}^*)'\mathbf{z}, \mathbf{w}'\mathbf{r}^s\big)$ with the cardinality constraint, *i.e.* only choosing $k$ stocks from $d$ index components, is still a mixed-integer quadratic programming problem that is challenging and computationally expensive to solve. Hence, before identifying $\mathbf{r}^s$, we first search for a vector of some PCs, $\mathbf{z}_s$, which controls $\rho((\mathbf{q}^*)'\mathbf{z}, \mathbf{b}_s'\mathbf{z}_s)$ as small as possible where $\mathbf{b}_s$ is a coefficient vector of $\mathbf{z}_s$. The quantity $\rho((\mathbf{q}^*)'\mathbf{z}, \mathbf{b}_s'\mathbf{z}_s)$ measures the distance between $(\mathbf{q}^*)'\mathbf{z}$ and the best combination of the selected subset of PCs. The selection of the subset of PCs is achieved by Sobol's total sensitivity index.

Sobol's total sensitivity index comes from the variance-based sensitivity analysis, and is used to measure how sensitive the output is to input changes. Sobol's total sensitivity index is defined based on Sobol's decomposition that is introduced in [114] with the assumption of independent uniformly distributed inputs. Sobol's decomposition is further generalized to independent inputs with any distributions by [99].

Suppose $\eta = \eta(x_1, \ldots, x_{d^*})$ is a function of independent inputs $x_i$, $i = 1, \ldots, d^*$. Sobol's decomposition states that, if $\eta = \eta(x_1, \ldots, x_{d^*})$ has finite second moments, then $\eta$ can be uniquely decomposed as $\eta = \eta(x_1, \ldots, x_{d^*}) = \sum_{v \subset \{1,\ldots,d^*\}} \eta_v(\mathbf{x}_v)$, such that

$$Var(\eta) = \sum_{v \subset \{1,\ldots,d^*\}} Var(\eta_v(\mathbf{x}_v)),$$

and the expectation of each summand, except for $\eta_0 (= \eta_\emptyset)$, is zero. Sobol's sensitivity index for $\mathbf{x}_v$, $v \subset \{1, \ldots, d^*\}$, is defined as

$$s_v = Var\big(\eta_v(\mathbf{x}_v)\big)/Var(\eta),$$

so that $0 \leq s_v \leq 1$ and $\sum_v s_v = 1$. Sobol's total sensitivity index for input $x_i$ is defined as

$$s_i^{total} = \sum_{v:i\in v} s_v = \sum_{v:i\in v} Var\big(\eta_v(\mathbf{x}_v)\big)/Var(\eta).$$

Sobol's total sensitivity index is useful to freeze unessential variables in a complicated system (see [114]).

In order to apply Sobol's total sensitivity index to search for a vector of some PCs, $\mathbf{z}_s$, that controls $\rho((\mathbf{q}^*)'\mathbf{z}, \mathbf{b}_s'\mathbf{z}_s)$, we have Proposition 3.1.

**Proposition 3.1.** Suppose $\mathbf{r} = (r_1, \ldots, r_d)$ follows a multivariate normal distribution, and elements of $\mathbf{z}$ are principal components of $\mathbf{r}$. Let $\eta = \eta(\mathbf{z}) = \mathbf{q}'\mathbf{r} = (\mathbf{q}^*)'\mathbf{z}$. Then the following results hold.

(a) Sobol's decomposition of $\eta$ is given by $\eta = \eta_0 + \sum_{i=1}^d \eta_i(z_i)$, where $\eta_0 = E[\eta]$ and $\eta_i(z_i) = q_i^* z_i + \alpha_i$ and $\alpha_i = -q_i^* E[z_i]$ for $i = 1, \ldots, d$.

(b) Write $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_s)$, and let $s_{\mathbf{z}_s}^{total} = \sum_{j:z_j\in\mathbf{z}_s} s_j^{total}$. The $\rho$-distance between $(\mathbf{q}^*)'\mathbf{z}$ and any linear combination of elements in $\mathbf{z}_1$ is larger than $\sqrt{s_{\mathbf{z}_s}^{total} \cdot Var(\eta)}$, that is

$$\rho\big((\mathbf{q}^*)'\mathbf{z}, \mathbf{a}'\mathbf{z}_1\big) \geq \sqrt{s_{\mathbf{z}_s}^{total} \cdot Var(\eta)},$$

where $\mathbf{a}$ is a column vector of constants.

(c) For any $\delta > 0$,

$$Pr\left\{\eta_i^2(z_i) < \frac{Var(\eta)}{\delta} s_i^{total}\right\} \geq 1 - \delta, \text{ for } i = 1, \ldots, d.$$

*Proof.* (a) This is a special case of the result in [87].

(b) Let $\eta = (\mathbf{q}^*)'\mathbf{z} = (\mathbf{q}_1^*)'\mathbf{z}_1 + (\mathbf{q}_s^*)'\mathbf{z}_s$. We have

$$E\left[\big((\mathbf{q}^*)'\mathbf{z} - \mathbf{a}'\mathbf{z}_1\big)^2\right] = E\left[\big((\mathbf{q}_1^*)'\mathbf{z}_1 + (\mathbf{q}_s^*)'\mathbf{z}_s - \mathbf{a}'\mathbf{z}_1\big)^2\right] = E\left[\big(\mathbf{b}'\mathbf{z}_1 + (\mathbf{q}_s^*)'\mathbf{z}_s\big)^2\right],$$

where $\mathbf{b} = \mathbf{q}_1^* - \mathbf{a}$. Let $\bar{\mathbf{z}}_i = \mathbf{z}_i - E[\mathbf{z}_i]$ for $i = 1, s$, and $c = \mathbf{b}'E[\mathbf{z}_1] + (\mathbf{q}_s^*)'E[\mathbf{z}_s]$, then

$$\begin{aligned} E\left[\left((\mathbf{q}^*)'\mathbf{z} - \mathbf{a}'\mathbf{z}_1\right)^2\right] &= E\left[\left(\mathbf{b}'\bar{\mathbf{z}}_1 + (\mathbf{q}_s^*)'\bar{\mathbf{z}}_s + c\right)^2\right] \\ &= E\left[\left(\mathbf{b}'\bar{\mathbf{z}}_1\right)^2\right] + E\left[\left((\mathbf{q}_s^*)'\bar{\mathbf{z}}_s\right)^2\right] + c^2. \end{aligned}$$

The last equation is true since $\mathbf{z}_1$ and $\mathbf{z}_s$ are independent under the assumptions of a multivariate normal distribution for $\mathbf{r}$, and $E[\bar{\mathbf{z}}_i] = 0$ for $i = 1, s$. Hence,

$$\begin{aligned} \rho^2\left((\mathbf{q}^*)'\mathbf{z}, \mathbf{a}'\mathbf{z}_1\right) &= E\left[\left((\mathbf{q}^*)'\mathbf{z} - \mathbf{a}'\mathbf{z}_1\right)^2\right] \\ &\geq E\left[\left((\mathbf{q}_s^*)'\bar{\mathbf{z}}_s\right)^2\right] = \sum_{j:\bar{z}_j \in \bar{\mathbf{z}}_s} (q_j^*)^2 Var(\bar{z}_j) \\ &= \sum_{j:z_j \in \mathbf{z}_s} (q_j^*)^2 Var(z_j). \end{aligned}$$

Note that $\eta_i(z_i) = q_i^* z_i + \alpha_i$ and $\alpha_i = -q_i^* E[z_i]$ for $i = 1, \ldots, d$. We have

$$\begin{aligned} \rho^2\left((\mathbf{q}^*)'\mathbf{z}, \mathbf{a}'\mathbf{z}_1\right) &\geq \sum_{j:z_j \in \mathbf{z}_s} \frac{Var\left(\eta_j(z_j)\right)}{Var(\eta)} Var(\eta) \\ &= s_{\mathbf{z}_s}^{total} \cdot Var(\eta). \end{aligned}$$

(c) According to Markov's inequality, for any constant $g > 0$,

$$Pr\left\{\eta_i^2(z_i) \geq g\right\} \leq \frac{E\left[\eta_i^2(z_i)\right]}{g} = \frac{Var\left(\eta_i^2(z_i)\right)}{g}.$$

That is $Pr\left\{\eta_i^2(z_i) < g\right\} \geq 1 - \frac{Var\left(\eta_i^2(z_i)\right)}{g}$. Let $g = \frac{Var\left(\eta_i(z_i)\right)}{\delta}$, we have

$$Pr\left\{\eta_i^2(z_i) < \frac{Var(\eta)}{\delta} s_i^{total}\right\} \geq 1 - \delta.$$

□

**Remark 3.1.** (a) According to part (a) of Proposition 3.1, Sobol's sensitivity index of

the PC $z_i$ is $s_i = \frac{(q_i^*)^2 Var(z_i)}{Var(\eta)}$. Since there are neither high-order terms nor intersection terms among these summands, $s_i^{total} = s_i$ for $i = 1, 2, \ldots, d$.

(b) Part (b) of Proposition 3.1 suggests that if $s_{\mathbf{z}_s}^{total}$ is large, discarding $\mathbf{z}_s$ and any linear combination of PCs leaving $\mathbf{z}_s$ out has a large deviation from $(\mathbf{q}^*)'\mathbf{z}$. Hence, PCs with large Sobol's total sensitivity indices should be retained to approximate $(\mathbf{q}^*)'\mathbf{z}$.

(c) According to part (c) of Proposition 3.1, $Var(\eta_i(z_i))$ is small if $s_i^{total}$ is sufficiently small. Note that $E[\eta_i(z_i)] = 0$, hence $\eta_i(z_i)$ as well as $z_i$ is negligible in the system $(\mathbf{q}^*)'\mathbf{z}$. In fact,

$$Pr\left\{-\frac{1}{\sigma_{\eta_i}}\sqrt{\frac{Var(\eta)}{\delta}s_i^{total}} < \frac{\eta_i(z_i)}{\sigma_{\eta_i}} < \frac{1}{\sigma_{\eta_i}}\sqrt{\frac{Var(\eta)}{\delta}s_i^{total}}\right\} > 1-\delta,$$

where $\sigma_{\eta_i}^2 = Var\big(\eta_i(z_i)\big)$.

Denote by $\alpha_{\frac{\delta}{2}}$ the $1-\frac{\delta}{2}$ quantile of a standard normal distribution. According to part (a) of Proposition 3.1, $\eta_i(z_i)$ follows a normal distribution with zero mean. Hence, $\alpha_{\frac{\delta}{2}} < \frac{1}{\sigma_{\eta_i}}\sqrt{\frac{Var(\eta)}{\delta}s_i^{total}}$, that is

$$Var(\eta_i) < \frac{Var(\eta)}{\delta\big(\alpha_{\frac{\delta}{2}}\big)^2}s_i^{total}.$$

(d) Proposition 3.1 relies on an assumption of the multivariate normal distribution. Even though stock returns do not always follow a multivariate normal distribution, analytic results in Proposition 3.1 are difficult to obtain for other more general settings. In this chapter, we use the multivariate normal distribution model as a benchmark to sort out a replicating strategy. The performance of such strategies is tested by real data in Section 3.5.

Proposition 3.1 suggests that we should retain PCs with large Sobol's total sensitivity indices to approximate $(\mathbf{q}^*)'\mathbf{z}$, and PCs with small sensitivity indices can be ignored. These retained PCs keep a large portion of $(\mathbf{q}^*)'\mathbf{z}$'s variance and the approximation error is

controlled by Sobol's total sensitivity index. Among applications in Section 3.5, we retain PCs whose corresponding sensitivity indices are larger than a threshold such as 0.001.

## 3.4 Select Variables based on Retained PCs

Suppose some PCs $\mathbf{z}_s$ are retained by comparing their Sobol's total sensitivity indices as explained in the last section. In order to identify $k$ stocks based on $\mathbf{z}_s$, a natural idea is to establish a relationship between $\rho((\mathbf{q}^*)'\mathbf{z}, \mathbf{w}'\mathbf{r}^s)$ and $\rho((\mathbf{q}^*)'\mathbf{z}, (\mathbf{b}_s)'\mathbf{z}_s)$, where $\mathbf{b}_s$ is a vector of weights for the retained PCs. However, such a relationship is rather challenging to establish. Hence, based on the retained PCs, stocks in the tracking portfolio are selected by comparing the dependence between $\mathbf{z}_s$ and $\mathbf{r}^s$.

Research works on choosing variables according to some PCs dated back to [76, 77]. The motivation is the conjecture that a portion of PCs can be very well explained by a portion of all variables that form all PCs. In [76, 77], many ad-hoc methods are compared using both artificial and real data. However, it is pointed out in [21] that these ad-hoc methods are potentially misleading in selecting subset variables to approximate retained PCs. The authors of [21] suggest selecting the variable subset by optimizing some criteria, such as Yanai's generalized coefficient of determination (GCD). Inspired by [21], in our research three criteria are considered in choosing stocks based on the retained PCs in this chapter. They are Yanai's GCD, the distance correlation and HGG test statistics.

Yanai's generalized coefficient of determination (GCD) is introduced in [132]. It is a type of the matrix correlation which is introduced in [100]. Suppose $X$ is a $n \times d$ data matrix of $\mathbf{r}$, and the $j$th column of $X$ includes samples of $r_j$ for $j = 1, \ldots, d$. Let $\mathcal{G}$ be a collection of subscripts of elements in $\mathbf{z}_s$. Here, the cardinality of $\mathcal{G}$ is denoted by $m$. Define $A_{\mathcal{G}}$ as a $d \times m$ sub-matrix of the PC loading matrix $A$. Particularly, $A_{\mathcal{G}}$ is obtained by retaining all the columns $j$ of $A$ for $j \in \mathcal{G}$. We further denote the subspace spanned by $\mathbf{z}_s$ by $G$. For the space $G$, there is an corresponding orthogonal projection matrix $P_{\mathcal{G}}(X) = XA_{\mathcal{G}}(A'_{\mathcal{G}}X'XA_{\mathcal{G}})^{-1}A'_{\mathcal{G}}X'$. Similarly, we denote by $\mathcal{K}$ a collection of subscripts of elements in $\mathbf{r}^s$. The data matrix of $\mathbf{r}^s$ is $XI_{\mathcal{K}}$. Here, $I_{\mathcal{K}}$ is an $n \times k$ sub-matrix of the

identity matrix, and $I_{\mathcal{K}}$ is obtained by keeping the $j$th column of the $d \times d$ identity matrix for $j \in \mathcal{K}$. These $k$ variables span a subspace $K$ with an orthogonal projection matrix $P_{\mathcal{K}}(X) = XI_{\mathcal{K}}(I'_{\mathcal{K}}X'XI_{\mathcal{K}})^{-1}I'_{\mathcal{K}}X'$.

Yanai's GCD of $P_{\mathcal{G}}$ and $P_{\mathcal{K}}$, which is denoted by $GCD(P_{\mathcal{G}}, P_{\mathcal{K}})$, is used in [21] to measure the "correlation" or similarity between subspaces $G$ and $K$. It is shown in [21] that

$$GCD(P_{\mathcal{G}}, P_{\mathcal{K}}) = \frac{1}{\sqrt{mk}} \sum_{j \in \mathcal{G}} (\tilde{r}_m)_j^2,$$

where $(\tilde{r}_m)_j = \sqrt{\Lambda_j}\sqrt{(\mathbf{a}_j^{\mathcal{K}})'\Sigma_{\mathcal{K}}^{-1}\mathbf{a}_j^{\mathcal{K}}}$, for $j \in \mathcal{G}$, $\Lambda_j$ is the $j$th diagonal element of $\Lambda$ which is the eigenvalue matrix of $\mathbf{r}$'s covariance matrix. Further, $\mathbf{a}_j^{\mathcal{K}}$ is the sub-vector of the $j$th column of the PC loading matrix A. The cardinality of $\mathbf{a}_j^{\mathcal{K}}$ is $k$, and each of its element corresponds to one variable in $\mathbf{r}^s$. The matrix $\Sigma_{\mathcal{K}}$ is a sub matrix of the covariance matrix $\Sigma$, involving only rows and columns corresponding to these $k$ variables in $\mathbf{r}^s$. For simplicity, we rewrite $GCD(P_{\mathcal{G}}, P_{\mathcal{K}})$ as $GCD(G, K)$.

Yanai's GCD is able to measure the similarity between two subspaces in different dimensions. The value of Yanai's GCD is between 0 and 1. If $GCD(G, K) = 1$, subspaces $G$ and $K$ coincide. That is any linear combination of data of $\mathbf{z}_s$ can be rewritten as a linear combination of data of $\mathbf{r}^s$. If $GCD(G, K) = 0$, subspaces $G$ and $K$ are mutually orthogonal. This suggests that $\mathbf{r}^s$ cannot explain any linear combinations of $\mathbf{z}_s$. Hence, the $k$ stocks that maximize $GCD(G, K)$ should be selected to explain retained PCs.

Distance correlation ($dCor$), which is introduced in [123], is able to detect the dependence between random vectors in different dimensions. Distance correlation is closely linked to distance covariance. Suppose $\mathbf{x}$ is a $p$-dimensional random vector, and $\mathbf{y}$ is a $q$-dimensional random vector. The distance covariance of $\mathbf{x}$ and $\mathbf{y}$, $\mathcal{V}(\mathbf{x}, \mathbf{y})$, is defined based on characteristic functions, and it is the positive square root of

$$\mathcal{V}^2(\mathbf{x}, \mathbf{y}) = ||f_{\mathbf{xy}} - f_{\mathbf{x}}f_{\mathbf{y}}||^2 = \frac{1}{c_p c_q} \int_{\mathbb{R}^{p+q}} \frac{|f_{\mathbf{xy}}(\mathbf{t}, \mathbf{s}) - f_{\mathbf{x}}(\mathbf{t})f_{\mathbf{y}}(\mathbf{s})|^2}{|\mathbf{t}|_p^{1+p}|\mathbf{s}|_q^{1+q}} d\mathbf{t}d\mathbf{s},$$

where $f_{\mathbf{x}}$, $f_{\mathbf{y}}$, and $f_{\mathbf{xy}}$ denote characteristic functions of random vectors $\mathbf{x}$, $\mathbf{y}$, and $(\mathbf{x}, \mathbf{y})$ respectively. Constants $c_p$ and $c_q$ are defined in [123], and $|\mathbf{t}|_p$ is the Euclidean norm of $\mathbf{t}$

in $\mathbb{R}^p$. Suppose $(\mathbf{X}, \mathbf{Y}) = \{(X_i, Y_i), i = 1, \ldots, n\}$ is a collection of observed samples from the joint distribution of $(\mathbf{x}, \mathbf{y})$, the empirical distance covariance $\mathcal{V}_n(\mathbf{X}, \mathbf{Y})$ is the positive square root of $\mathcal{V}_n^2(\mathbf{X}, \mathbf{Y}) = \frac{1}{n^2}\sum_{i,l=1} A_{il}B_{il}$, where $a_{il} = |X_i - X_l|_p$, $\bar{a}_{i\cdot} = \frac{1}{n}\sum_{l=1}^n a_{il}$, $\bar{a}_{\cdot l} = \frac{1}{n}\sum_{j=1}^n a_{il}$, $\bar{a}_{\cdot\cdot} = \frac{1}{n^2}\frac{1}{n}\sum_{i,l=1}^n a_{il}$, $A_{il} = a_{il} - \bar{a}_{i\cdot} - \bar{a}_{\cdot l} + \bar{a}_{\cdot\cdot}$. Replacing $\{X_i\}$ by $\{Y_i\}$ in the calculation of $A_{il}$ leads to $B_{il}$. According to [123], $\lim_{n\to+\infty} \mathcal{V}_n(\mathbf{X}, \mathbf{Y}) = \mathcal{V}(\mathbf{x}, \mathbf{y})$ almost surely, given both $\mathbf{x}$ and $\mathbf{y}$ have finite Euclidean norms.

Distance correlation $dCor(\mathbf{x}, \mathbf{y})$ and its empirical version $dCor_n(\mathbf{x}, \mathbf{y})$ are defined by

$$
\begin{aligned}
dCor^2(\mathbf{x}, \mathbf{y}) &= \begin{cases} \frac{\mathcal{V}^2(\mathbf{x},\mathbf{y})}{\sqrt{\mathcal{V}^2(\mathbf{x})\mathcal{V}^2(\mathbf{y})}}, & \mathcal{V}^2(\mathbf{x})\mathcal{V}^2(\mathbf{y}) > 0, \\ 0, & \mathcal{V}^2(\mathbf{x})\mathcal{V}^2(\mathbf{y}) = 0, \end{cases} \\
dCor_n^2(\mathbf{X}, \mathbf{Y}) &= \begin{cases} \frac{\mathcal{V}_n^2(\mathbf{X},\mathbf{Y})}{\sqrt{\mathcal{V}_n^2(\mathbf{X})\mathcal{V}_n^2(\mathbf{Y})}}, & \mathcal{V}_n^2(\mathbf{X})\mathcal{V}_n^2(\mathbf{Y}) > 0, \\ 0, & \mathcal{V}_n^2(\mathbf{X})\mathcal{V}_n^2(\mathbf{Y}) = 0, \end{cases}
\end{aligned}
$$

Both $dCor$ and $dCor_n$ are between 0 and 1. The distance correlation equals 0 if and only if $\mathbf{x}$ and $\mathbf{y}$ are independent. If $dCor_n(\mathbf{X}, \mathbf{Y}) = 1$ then there exist a vector $a$, a nonzero real number $b$ and an orthogonal matrix $C$ such that $\mathbf{Y} = a + b\mathbf{X}C$. Returning to our variable selection problems, in order to explain given PCs $\mathbf{z}_s$, we prefer the $k$-dimensional $\mathbf{r}^s$ with the largest $dCor_n(\mathbf{z}_s, \mathbf{r}^s)$.

The HHG test, an independent test, is introduced in [70]. It can be used to describe the dependence between two random vectors in different dimensions. The idea is inspired by Pearson's independence test. Suppose $(X_i, Y_i)$ for $i = 1, \ldots, N$ are observations of random

vectors $\mathbf{x}$ and $\mathbf{y}$. For a specified distance $d(\cdot,\cdot)$ and $i \neq j$, $i, j = 1, \ldots, N$, define

$$\begin{aligned}
A_{11}(i,j) &= \sum_{k=1,k\neq i,j}^{N} I\left\{d(X_i,X_k) \leq d(X_i,X_j)\right\} I\left\{d(Y_i,Y_k) \leq d(Y_i,Y_j)\right\}, \\
A_{12}(i,j) &= \sum_{k=1,k\neq i,j}^{N} I\left\{d(X_i,X_k) \leq d(X_i,X_j)\right\} I\left\{d(Y_i,Y_k) > d(Y_i,Y_j)\right\}, \\
A_{21}(i,j) &= \sum_{k=1,k\neq i,j}^{N} I\left\{d(X_i,X_k) > d(X_i,X_j)\right\} I\left\{d(Y_i,Y_k) \leq d(Y_i,Y_j)\right\}, \\
A_{22}(i,j) &= \sum_{k=1,k\neq i,j}^{N} I\left\{d(X_i,X_k) > d(X_i,X_j)\right\} I\left\{d(Y_i,Y_k) > d(Y_i,Y_j)\right\}, \\
A_{m\cdot}(i,j) &= A_{m1}(i,j) + A_{m2}(i,j) \text{ and } A_{\cdot m}(i,j) = A_{1m}(i,j) + A_{2m}(i,j),
\end{aligned}$$

for $m = 1, 2$.

The HHG test statistics is defined as

$$T(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{N} \sum_{j=1,j\neq i}^{N} S(i,j)$$

where

$$S(i,j) = \frac{(N-2)\left[A_{12}(i,j)A_{21}(i,j) - A_{11}(i,j)A_{22}(i,j)\right]^2}{A_{1\cdot}(i,j)A_{2\cdot}(i,j)A_{\cdot 1}(i,j)A_{\cdot 2}(i,j)}$$

It is claimed in [70] that the larger the value of $S(i,j)$, the stronger the dependence between $I\left\{d(x_i,X) \leq d(x_i,x_j)\right\}$ and $I\left\{d(y_i,Y) \leq d(y_i,y_j)\right\}$. Hence, a larger $T(\mathbf{x},\mathbf{y})$ suggests stronger dependence between $\mathbf{x}$ and $\mathbf{y}$. Again, given PCs $\mathbf{z}_s$, it is better to select $\mathbf{r}^s$ that maximizes $T(\mathbf{r}^s, \mathbf{z}_s)$ to explain $\mathbf{z}_s$.

A comparison of the above three criteria to select stocks is given as follows:

- Yanai's GCD: Yanai's GCD measures the similarity between the subspace generated by the data of two random vectors. If GCD of two subspaces is 1, these two subspaces coincide. Dimensions of these two vector can be different.

- Distance correlation and HHG test statistics: Both of them can be applied to detect linear or non-linear relationships between two random vectors in different dimensions. Distance correlation has a simpler form.

Maximizing these criteria between retained PCs and $k$ stocks can be formulated as a binary programming problem. In very low dimensions, it is possible to search for the global optimal solution. However, when the dimension is large, heuristic methods should be applied to obtain a suboptimal solution.

In the end, the algorithm of our proposed variable selection for index tracking is outlined in Table 3.2.

| |
|---|
| 1: Input a $n \times (d+1)$ sample matrix of the random vector $(R, \mathbf{r})$. |
| 2: Obtain an estimator $\hat{\Sigma}$ of the covariance matrix of $\mathbf{r}$. |
| 3: Determine PCs of $\mathbf{r}$ based on $\hat{\Sigma}$, according to the eigenvalue decomposition of $\hat{\Sigma}$. |
| 4: Decompose $R$ to PCs $\mathbf{z}$ using Sobol's decomposition, which is given in the part $(a)$ of Proposition 3.1. |
| 5: Calculate Sobol's total sensitivity index for each PC. |
| 6: Retain $m$-dimensional PC subset $\mathbf{z}_s$ with Sobol's total sensitivity index larger than a certain threshold. |
| 7: Select $k$-dimensional $\mathbf{r}^s$ that maximizes GCD, $dCor_n$, or HGG test statistics. |

Table 3.2: The algorithm of variable selections for index tracking.

Given stocks to hold in the tracking portfolio, corresponding weights can be obtained by existing methods, such at those in [105] or [5]. In this chapter, we follow [105] and determine stock weights by minimizing specific tracking errors.

## 3.5 Applications to Financial Data

In this section, we apply our proposed variable selection method to real financial data. In order to evaluate its performance, tracking portfolios should be constructed, *i.e.* weights of chosen stocks should be determined, and then tracking errors should be compared.

### 3.5.1 Estimation Issues in High Dimensions

In some applications, we need to address some high-dimensional estimation issues when the sample size is smaller than the number of index components.

The estimation of the covariance matrix plays an important role in our proposed variable selection method. The sample covariance matrix is an unbiased estimator for the covariance matrix ([6]). However, in high-dimensional cases the sample covariance matrix is not of full rank, so that it is not invertible. Moreover, according to [107] eigenvalues computed from the sample covariance are not reliable in high dimensional cases. In order to overcome these shortcomings, we use the shrinkage covariance matrix, which is introduced in [86], to estimate the covariance matrix.

The shrinkage covariance matrix shrinks the sample covariance matrix towards a target matrix. Denote by $\hat{\Sigma}$ the shrinkage covariance matrix, $\hat{\Sigma}_{Sample}$ the sample covariance matrix, and $\hat{\Sigma}_T$ the target matrix, the shrinkage covariance matrix is given by

$$\hat{\Sigma} = (1-\lambda)\hat{\Sigma}_{Sample} + \lambda\hat{\Sigma}_T, \tag{3.6}$$

where $\lambda \in [0,1]$. Usually, $\hat{\Sigma}_T$ should be of full rank and have a simple form. It is suggested in [86] that $\lambda$ should be determined by minimizing the Frobenius norm of $(1-\lambda)\hat{\Sigma}_{Sample} + \lambda\hat{\Sigma}_T - \Sigma$. Here, following [107] our target matrix is a diagonal matrix of which each diagonal element is an unbiased sample variance for a corresponding stock return. The estimation of the corresponding optimal $\lambda$ is given in [107]. Given a nonzero estimation of $\lambda$, the shrinkage covariance matrix we adopted is of full rank and positive semidefinite. It also improves eigenvalue estimations (see [107]).

Another high-dimensional issue arises in estimating Sobol's sensitivity index. According to Proposition 3.1, Sobol's total sensitivity index of PC $z_i$ is $\frac{(q_i^*)^2 Var(z_i)}{Var(\eta)}$. Here, we estimate it by $\frac{(\hat{q}_i^*)^2\hat{\sigma}_{z_i}^2}{\hat{\sigma}_\eta^2}$, where $\hat{\sigma}_x^2 = \frac{1}{n-1}\sum_{i=1}^n \left(X_i - \frac{1}{n}\sum_{i=1}^n X_i\right)^2$ for any random variable $x$ with samples $\{X_i\}_{i=1}^n$. When the sample matrix of $\mathbf{z}$ is of high dimension, we turn to Lasso regression, introduced in [124], to estimate the $q_i^*$'s. Mathematically, estimators $\hat{q}_i^*$'s of the

Lasso regression are given by solving

$$\min_{\mathbf{q}^*} \sum_{t=1}^{T} \left(R_t - (\mathbf{q}^*)' Z_t\right)^2 + \lambda_L \sum_{i=1}^{d} |q_i^*|, \tag{3.7}$$

where $Z_t$, a column vector, is the sample of $\mathbf{z}$ at time $t$ which is inferred from the sample matrix of $\mathbf{r}$, $\lambda_L$ is the Lasso tuning parameter which is determined by a 10-fold cross-validation. The ordinary least square estimation (without penalty) is applied when the sample size is larger than the number of index components.

### 3.5.2 Use MSE as Tracking Error

We use the data provided in the OR-library which is used in [7] and many other papers on index tracking, and consider weekly levels of five stock market indices: the Hang Seng index, DAX index, FTSE index, S&P 100 index and the Nikkei 225 index, as well as weekly stock prices of their components. Numbers of components of Hang Seng index, DAX index, FTSE index, S&P 100 index and the Nikkei 225 index are 31, 85, 89, 100 and 225 respectively. The weekly data cover the period March 1992 to September 1997, including 291 observations.

In this section, the empirical MSE, $\frac{1}{T}\sum_{t=1}^{T}\left(R_t - \mathbf{w}'\mathbf{r}_t^s\right)^2$, is used to measure the tracking error. This tracking error is also consistent with the distance $\rho(\cdot,\cdot)$ which is defined at the end of Section 3.2. This tracking error is a standard loss function used in the industry ([53]). In order to investigate the performance of our proposed model, results reported in [105] are used as benchmarks. The methodology of [105] is briefly reviewed in Section 3.1. In order to make our results comparable with those in [105], we first obtain 290 weekly discrete-time returns, and divide them into in-sample data and out-of-sample data. Both the in-sample and the out-of-sample data include 145 weekly returns. The in-sample data are used to construct tracking portfolios, and the out-of-sample data are used to check the tracking accuracy. In dealing with the in-sample data, we use our proposed variable selection algorithm described in Table 3.2 to select stocks in the tracking portfolio, and consequently their weights are determined by minimizing the in-sample empirical MSE.

The Nikkei 225 has 225 components of which the number is larger than the size of the in-sample data (145 weekly returns). In this case, the covariance matrix is estimated by the shrinkage covariance matrix in (3.6), and $q_i^*$'s in Sobol's total sensitivity indices are estimated by Lasso regression estimators in (3.7). For the other indices, we directly apply sample covariance matrices and ordinary least square estimators. In terms of retaining PCs, we select PCs for which Sobol's total sensitivity indices are larger than 0.001.

In maximizing Yanai's GCD, we use methods proposed in [20] which is coded in the R package "subselect". In dealing with the Hang Seng index, we obtain the optimal variable subset by the function "eleaps". For other indices, it is infeasible to obtain the optimal solution, so we obtain sub-optimal solutions by a genetic algorithm that is carried out by the function "genetic" in the R package "subselect". Setting of the "genetic" function is as follows: the size of population is the maximum of 100 and 2 times the number of index components, the maximum generation number is 300, and we adopt other default settings. In maximizing $dCor_n$ for the case of the Hang Seng index, we exclusively search for the optimal solution. For the other indices, we use the Matlab built-in function "ga", which is a genetic algorithm solver. The size of population and the maximum generation number are the same as the genetic algorithm settings for maximizing Yanai's GCD, and we adopt other default settings. In calculating HHG test statistics, we use the Euclidean distance, and use the Matlab build-in function "ga" for all indices with the same settings as specified above. All reported results are averaged over 5 executions of the genetic algorithm.

Once stocks in tracking portfolios are identified, their weights are determined by solving

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{T}\sum_{t=1}^{T}\left(R_t - \mathbf{w}'\mathbf{r}_t^s\right)^2 \\ s.t. \quad & \sum_{j=1}^{k} w_j = 1, \\ & 0 \le w_j \le 1, \text{ for } j = 1, \ldots, k, \end{aligned}$$

where $\mathbf{r}_t^s$, a column vector, is a sample of $\mathbf{r}^s$ at time $t$. This proposed method of constructing tracking portfolios requires less computation than the hybrid programming method in [105], since the most time-consuming part of our method is a 0-1 integer programming problem.

Finally, in-sample and out-of-sample tracking errors (TEs) are reported in Table 3.3.

Table 3.3 shows results of tracking portfolios in which the cardinality of stock subsets varies from 5 to 10. Out-of-sample TEs should be the focus, since we are interested in the tracking ability of a tracking portfolio. According to out-of-sample TEs, our proposed method is comparable to results in [105]. It leads to better out-of-sample TEs in many scenarios, thought it does not outperform the benchmark uniformly.

### 3.5.3 Use Conditional Value at Risk as Tracking Error

Some other forms of tracking error may be used depending on fund managers' objectives. In this section, we study the index tracking problem by using the empirical conditional value at risk (CVaR) of the tracking discrepancy $R - \mathbf{w}'\mathbf{r}^s$ as a tracking error. Suppose that the loss $x$ follows a continuous distribution, the CVaR of $x$ at $\alpha$%-level, $CVaR_\alpha(x)$, is defined as $E\left[x|x > VaR_\alpha(x)\right]$, where $VaR_\alpha(x)$ is short for the Value-at-Risk of $x$ at $\alpha$%-level that is defined as the $\alpha$% quantile of $x$. CVaR is used to describe the tail behavior of the loss $x$. Compared to the empirical MSE, the CVaR tracking error leads to tracking portfolios which control the tail risk of the tracking discrepancy.

If $R-\mathbf{w}'\mathbf{r}^s$ is not required to be as close to zero as possible, a negative value is preferred since it means that the tracking portfolio has a higher return than the index return. Using the empirical CVaR of $R-\mathbf{w}'\mathbf{r}^s$ as a tracking error provides information about worse case scenarios. It helps to figure out how poorly a tracking portfolio might perform, and the resulting tracking portfolio aims at optimizing the average performance over those worse case scenarios.

Firstly, our variable selection method is applied to identify stocks in tracking portfolios. Given a set of stocks, their weights are determined by minimizing the empirical CVaR of

$R - \mathbf{w}'\mathbf{r}^s$, which according to [101], can be equivalently formulated as follows

$$\begin{aligned} \min_{(\mathbf{w},\zeta)} \quad & \zeta + \frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T}\left[(R_t - \mathbf{w}'\mathbf{r}_t^s) - \zeta\right]^+ \\ s.t. \quad & \sum_{j=1}^{k} w_j = 1, \\ & 0 \le w_j \le 1, \text{ for } j = 1, \ldots, k, \end{aligned} \tag{3.8}$$

where $\zeta$ is an auxiliary variable, $\alpha$ is the risk tolerance and set to be 0.95, $k$ is the fixed cardinality of the stock subset to hold in a tracking portfolio. Problem (3.8) can be solved using linear programming. The minimized value of the objective function is the empirical 95% CVaR of $R - \mathbf{w}'\mathbf{r}^s$.

In order to compare the performance of our variable selection method with this tracking error, a benchmark is required. Analogous to the case where the empirical MSE is used as the tracking error, the benchmark here is the tracking portfolio that is obtained by solving the variable selection and fund allocation simultaneously by minimizing the in-sample empirical 95% CVaR of $R - \mathbf{w}'\mathbf{r}^s$. Following [7], the benchmark tracking portfolio is obtained by solving

$$\begin{aligned} \min_{(\mathbf{w},\zeta,\mathbf{y})} \quad & \zeta + \frac{1}{1-\alpha}\frac{1}{T}\sum_{t=1}^{T}\left[(R_t - \mathbf{w}'\mathbf{r}_t) - \zeta\right]^+ \\ s.t. \quad & \sum_{j=1}^{d} w_j = 1, \\ & w_\varepsilon y_j \le w_j \le y_j, \text{ for } j = 1, \ldots, d, \\ & y_j \in \{0, 1\} \text{ for } j = 1, \ldots, d, \\ & \sum_{j=1}^{d} y_j = k, \end{aligned} \tag{3.9}$$

where $\mathbf{y} = (y_1, \ldots, y_d)$. Again, $\zeta$ is an auxiliary variable, $\alpha = 0.95$, $k$ is the fixed cardinality of the stock subset. In the second constraint, $w_\varepsilon$ is a positive small number to ensure that there are exactly $k$ positive elements in $\mathbf{w}$, and in this chapter we let $w_\varepsilon = 10^{-3}$. The

problem (3.9) can be cast as a mixed-integer linear programming problem of which the optimal solution is attainable efficiently by standard methods (see [22]).

Using the same data as in Section 3.5.2, given fixed subset cardinality, in-sample and out-of-sample tracking errors are obtained by our proposed method and the benchmark method respectively. Results are shown in Table 3.4.

In our specific implementation, problem (3.8) is solved by the Matlab build-in function "linprog" with default settings. The mixed-integer linear program (3.9) is solved by the Matlab built-in function "intlinprog". Settings of using this function are as follows: we use the branch-and-bound algorithm, the heuristic to a find feasible point is set to "rss", the termination criteria are TolCapRel=1e-4 and MaxTime=1500 (seconds). Both programs are run on a PC with Intel Core i5-3210M CPU at 2.5GHz and 6.00GB memory. With these termination criteria, the optimal solution to (3.9) is found in the case of the Hang Seng index. For the other indices, the program always stops at a sub-optimal solution when the maximum running time (1,500 seconds) is reached.

As shown in Table 3.4, the benchmark method leads to smaller in-sample tracking errors. However, in terms of the out-of-sample tracking errors, our method behaves comparably to the benchmark method. This suggests that our variable selection also works for controlling the tail of tracking discrepancy. Moreover, our method works much faster than the benchmark method. It takes 33.11 seconds for our method with GCD criteria to track Nikkei 225 with 10 stocks. Using $dCor$ or HHG test statistics as the criteria takes slightly more time, but the running time is as the same magnitude as that of using the GCD criteria. This is the most complicated case in Table 3.4, less time is required in other cases.

## 3.6 Discussion

In this chapter, we introduce a variable selection method for index tracking. Based on Sobol's total sensitivity index, we first select some significant PCs that well approximate the index return and explain a large portion of the index return variance. Then we search

for variables that maximize the similarity between the retained PCs and subset stocks. This similarity is measured by Yanai's GCD, distance correlation, or the HHG test statistics. Given the selected stocks, corresponding weights can be obtained by minimizing a specified tracking error. We apply our proposed variable selection to five stock-market indices. Using empirical MSE and CVaR as tracking errors, results suggest that our method is comparable with heuristic "one-step" methods in terms of out-of-sample performance, and our proposed method is more computationally efficient.

| | | GCD | | *dCor* | | HHG | | Benchmark | |
|---|---|---|---|---|---|---|---|---|---|
| | Card.($k$) | $\text{MSE}_{in}$ | $\text{MSE}_{out}$ | $\text{MSE}_{in}$ | $\text{MSE}_{out}$ | $\text{MSE}_{in}$ | $\text{MSE}_{out}$ | $\text{MSE}_{in}$ | $\text{MSE}_{out}$ |
| Hang Seng | 5 | 1.3E-04 | 1.2E-04 | 2.2E-04 | 3.5E-04 | 1.4E-04 | 1.4E-04 | 4.1E-05 | 7.2E-05 |
| (d=31) | 6 | 1.4E-04 | 9.5E-05 | 1.3E-04 | 1.7E-04 | 1.1E-04 | 7.7E-05 | 3.0E-05 | 4.7E-05 |
| | 7 | 8.8E-05 | 9.4E-05 | 1.3E-04 | 1.6E-04 | 6.0E-05 | 7.6E-05 | 2.3E-05 | 3.8E-05 |
| | 8 | 7.1E-05 | 6.4E-05 | 1.3E-04 | 1.6E-04 | 5.3E-05 | 6.6E-05 | 1.9E-05 | 2.8E-05 |
| | 9 | 7.4E-05 | 5.2E-05 | 1.1E-04 | 1.2E-04 | 5.8E-05 | 4.5E-05 | 1.6E-05 | 2.5E-05 |
| | 10 | 4.4E-05 | 5.5E-05 | 4.8E-05 | 6.6E-05 | 4.2E-05 | 5.0E-05 | 1.3E-05 | 2.0E-05 |
| DAX | 5 | 8.8E-05 | 1.7E-04 | 2.3E-05 | 1.0E-04 | 2.9E-05 | 1.1E-04 | 2.2E-05 | 1.0E-04 |
| (d=85) | 6 | 6.9E-05 | 1.3E-04 | 2.3E-05 | 9.0E-05 | 2.1E-05 | 9.0E-05 | 1.7E-05 | 8.9E-05 |
| | 7 | 6.4E-05 | 1.4E-04 | 2.0E-05 | 9.4E-05 | 2.5E-05 | 1.0E-04 | 1.3E-05 | 8.4E-05 |
| | 8 | 6.5E-05 | 1.3E-04 | 2.1E-05 | 8.6E-05 | 2.0E-05 | 1.0E-04 | 1.1E-05 | 7.9E-05 |
| | 9 | 6.3E-05 | 1.2E-04 | 1.7E-05 | 8.5E-05 | 1.3E-05 | 8.1E-05 | 9.2E-05 | 7.7E-05 |
| | 10 | 2.9E-05 | 1.2E-04 | 1.1E-05 | 7.7E-05 | 1.9E-05 | 9.1E-05 | 8.0E-05 | 7.4E-05 |
| FTSE | 5 | 2.3E-04 | 1.3E-04 | 1.1E-04 | 1.2E-04 | 1.0E-04 | 1.1E-04 | 6.4E-05 | 1.5E-04 |
| (d=89) | 6 | 1.7E-04 | 1.2E-04 | 1.1E-04 | 1.2E-04 | 1.0E-04 | 1.2E-04 | 4.9E-05 | 1.1E-04 |
| | 7 | 1.3E-04 | 1.0E-04 | 1.1E-04 | 9.3E-05 | 1.2E-04 | 1.3E-04 | 3.8E-05 | 9.0E-05 |
| | 8 | 1.3E-04 | 9.2E-05 | 8.6E-05 | 8.4E-05 | 8.6E-05 | 8.1E-05 | 2.9E-05 | 9.6E-05 |
| | 9 | 1.2E-04 | 7.4E-05 | 6.8E-05 | 9.1E-05 | 6.8E-05 | 9.2E-05 | 2.4E-05 | 8.5E-05 |
| | 10 | 1.2E-04 | 6.6E-05 | 5.6E-05 | 6.3E-05 | 8.4E-04 | 6.1E-05 | 2.1E-05 | 8.0E-05 |
| S&P100 | 5 | 1.0E-04 | 1.2E-04 | 1.4E-04 | 1.4E-04 | 2.0E-04 | 1.9E-06 | 4.4E-05 | 1.1E-04 |
| (d=100) | 6 | 1.2E-04 | 1.6E-04 | 1.5E-04 | 1.2E-04 | 1.5E-04 | 2.0E-04 | 3.3E-05 | 1.0E-04 |
| | 7 | 1.4E-04 | 1.2E-04 | 1.5E-04 | 1.4E-04 | 8.2E-05 | 1.0E-04 | 2.7E-05 | 7.7E-05 |
| | 8 | 1.1E-04 | 9.0E-05 | 9.4E-05 | 1.0E-04 | 1.0E-04 | 1.1E-04 | 2.2E-05 | 6.7E-05 |
| | 9 | 1.0E-04 | 9.4E-05 | 8.1E-05 | 8.1E-05 | 8.7E-05 | 8.8E-05 | 1.9E-05 | 5.9E-05 |
| | 10 | 9.0E-05 | 7.8E-05 | 8.6E-05 | 8.8E-05 | 7.0E-05 | 1.0E-04 | 1.6E-05 | 5.5E-05 |
| Nikkei | 5 | 9.0E-05 | 1.4E-04 | 1.5E-04 | 2.6E-04 | 1.0E-04 | 1.6E-04 | 5.4E-05 | 1.6E-04 |
| (d=225) | 6 | 6.2E-05 | 1.4E-04 | 1.4E-04 | 1.9E-04 | 9.0E-05 | 1.7E-04 | 4.0E-05 | 1.4E-04 |
| | 7 | 6.0E-05 | 1.1E-04 | 1.1E-04 | 1.6E-04 | 7.9E-05 | 2.2E-04 | 3.3E-05 | 1.3E-04 |
| | 8 | 5.2E-05 | 1.1E-04 | 6.6E-05 | 1.1E-04 | 9.0E-05 | 1.9E-04 | 2.6E-05 | 1.1E-04 |
| | 9 | 4.8E-05 | 9.6E-05 | 5.4E-05 | 1.0E-04 | 7.8E-05 | 1.7E-04 | 2.1E-05 | 9.8E-05 |
| | 10 | 4.7E-05 | 7.2E-05 | 6.5E-05 | 9.4E-05 | 6.7E-05 | 1.3E-04 | 1.7E-05 | 6.4E-05 |

Table 3.3: In-sample empirical MSE ($\text{MSE}_{in}$) and out-of sample empirical MSE ($\text{MSE}_{out}$). "GCD" refers to our method using Yanai's GCD criterion to select stocks. Similarly, "*dCor*" and "HHG" represent using the distance correlation and HHG test statistics to select stocks respectively. The last column shows published results in [105].

| | | GCD | | *dCor* | | HHG | | Benchmark | |
|---|---|---|---|---|---|---|---|---|---|
| | Card.($k$) | CVaR$_{in}$ | CVaR$_{out}$ | CVaR$_{in}$ | CVaR$_{out}$ | CVaR$_{in}$ | CVaR$_{out}$ | CVaR$_{in}$ | CVaR$_{out}$ |
| Hang Seng | 5 | 0.0239 | 0.0275 | 0.0317 | 0.0344 | 0.0261 | 0.0265 | 0.0121 | 0.0221 |
| (d=31) | 6 | 0.0188 | 0.0211 | 0.0241 | 0.0287 | 0.0251 | 0.0222 | 0.0100 | 0.0144 |
| | 7 | 0.0144 | 0.0211 | 0.0200 | 0.0282 | 0.0142 | 0.0212 | 0.0083 | 0.0159 |
| | 8 | 0.0137 | 0.0248 | 0.0213 | 0.0281 | 0.0136 | 0.0165 | 0.0065 | 0.0148 |
| | 9 | 0.0156 | 0.0161 | 0.0181 | 0.0260 | 0.0165 | 0.0190 | 0.0055 | 0.0136 |
| | 10 | 0.0124 | 0.0163 | 0.0155 | 0.0196 | 0.0136 | 0.0173 | 0.0050 | 0.0125 |
| DAX | 5 | 0.0221 | 0.0324 | 0.0128 | 0.0223 | 0.0121 | 0.0242 | 0.0085 | 0.0231 |
| (d=85) | 6 | 0.0166 | 0.0250 | 0.0112 | 0.0228 | 0.0097 | 0.0204 | 0.0073 | 0.0229 |
| | 7 | 0.0181 | 0.0242 | 0.0112 | 0.0233 | 0.0109 | 0.0273 | 0.0061 | 0.0208 |
| | 8 | 0.0152 | 0.0245 | 0.0101 | 0.0217 | 0.0107 | 0.0213 | 0.0055 | 0.0209 |
| | 9 | 0.0136 | 0.0232 | 0.0098 | 0.0207 | 0.0087 | 0.0204 | 0.0047 | 0.0186 |
| | 10 | 0.0149 | 0.0242 | 0.0082 | 0.0186 | 0.0101 | 0.0228 | 0.0046 | 0.0199 |
| FTSE | 5 | 0.0233 | 0.0263 | 0.0197 | 0.0272 | 0.0210 | 0.0212 | 0.0126 | 0.0213 |
| (d=89) | 6 | 0.0197 | 0.0280 | 0.0202 | 0.0267 | 0.0211 | 0.0289 | 0.0119 | 0.0226 |
| | 7 | 0.0214 | 0.0224 | 0.0194 | 0.0220 | 0.0200 | 0.0191 | 0.0107 | 0.0175 |
| | 8 | 0.0181 | 0.0211 | 0.0178 | 0.0218 | 0.0178 | 0.0192 | 0.0090 | 0.0163 |
| | 9 | 0.0166 | 0.0186 | 0.0153 | 0.0212 | 0.0172 | 0.0244 | 0.0087 | 0.0152 |
| | 10 | 0.0158 | 0.0175 | 0.0141 | 0.0164 | 0.0194 | 0.0191 | 0.0076 | 0.0158 |
| S&P100 | 5 | 0.0269 | 0.0259 | 0.0200 | 0.0235 | 0.0232 | 0.0317 | 0.0110 | 0.0213 |
| (d=100) | 6 | 0.0293 | 0.0320 | 0.0231 | 0.0272 | 0.0231 | 0.0300 | 0.0099 | 0.0226 |
| | 7 | 0.0266 | 0.0316 | 0.0191 | 0.0255 | 0.0191 | 0.0233 | 0.0079 | 0.0175 |
| | 8 | 0.0269 | 0.0294 | 0.0168 | 0.0279 | 0.0192 | 0.0253 | 0.0078 | 0.0163 |
| | 9 | 0.0191 | 0.0216 | 0.0157 | 0.0178 | 0.0143 | 0.0209 | 0.0064 | 0.0152 |
| | 10 | 0.0164 | 0.0246 | 0.0186 | 0.0190 | 0.0135 | 0.0234 | 0.0058 | 0.0158 |
| Nikkei | 5 | 0.0178 | 0.0278 | 0.0250 | 0.0396 | 0.0217 | 0.0323 | 0.0133 | 0.0350 |
| (d=225) | 6 | 0.0151 | 0.0249 | 0.0173 | 0.0370 | 0.0216 | 0.0276 | 0.0112 | 0.0252 |
| | 7 | 0.0148 | 0.0272 | 0.0194 | 0.0293 | 0.0169 | 0.0294 | 0.0100 | 0.0233 |
| | 8 | 0.0142 | 0.0223 | 0.0175 | 0.0282 | 0.0166 | 0.0261 | 0.0100 | 0.0229 |
| | 9 | 0.0119 | 0.0242 | 0.0162 | 0.0216 | 0.0180 | 0.0253 | 0.0090 | 0.0253 |
| | 10 | 0.0106 | 0.0269 | 0.0150 | 0.0242 | 0.0137 | 0.0215 | 0.0084 | 0.0233 |

Table 3.4: In-sample empirical 95% CVaR (CVaR$_{in}$) and out-of sample empirical 95% CVaR (CVaR$_{out}$). "GCD" refers to our method using Yanai's GCD criterion to select stocks. Similarly, "*dCor*" and "HHG" represent using the distance correlation and HHG test statistics to select stocks respectively. Here, "Benchmark" refers to results given by solving (3.9) using the Matlab built-in function "intlinprog".

# Chapter 4

# Index Tracking with Factor Models

## 4.1 Introduction

A simple way to track a benchmark index is full replication, which constructs a tracking portfolio strictly following the composition formula of the index (See Section 3.2.1 for examples). However, some small-cap stocks are only lightly traded in the stock market. Due to such liquidity issues, it is challenging to purchase enough quantity of small-cap stocks. This sometimes makes full replication infeasible, and many tracking portfolios use a portion of the index components to approximate the index return ([71]). How to construct such a partial replication is called the index tracking problem.

Denote by $R$ the index return in a generic period and $r_{tp}$ the return of a tracking portfolio. In this chapter, tracking portfolios are constructed to minimize the mean square error (MSE), which is $E[(R - r_{tp})^2]$. This is a widely accepted tracking error (see [102], [7], [105], [26]). The square loss function, $(R - r_{tp})^2$, penalizes large deviations of $r_{tp}$ from the index return $R$. According to [53], it is a standard loss function used in the industry.

In order to solve the index tracking problem, many methods such as those in [7], [82], and [105] formulate it as a mixed-integer quadratic programming problem. In this formula-

tion, stocks and corresponding weights in the tracking portfolio are determined by minimizing the in-sample empirical MSE subject to certain constraints. However, mixed-integer quadratic programming is NP-hard (see [105]), and so its optimal solution is challenging to obtain efficiently. In response, heuristic algorithms are proposed in [7], [82], and [105] to solve this mixed-integer quadratic program. Hereafter, we generally refer to these methods as heuristic methods.

When the number of index components is high, say several thousands, it is not feasible to use existing heuristic methods to construct tracking portfolios. Firstly, although tracking portfolios constructed by these heuristic methods usually lead to small in-sample tracking errors, they sometimes lead to volatile out-of-sample empirical MSEs especially when the number of index components is large. Secondly, these heuristic methods are too computationally demanding to construct tracking portfolios when the number of index components is large.

One reason of the volatile out-of-sample performance of these heuristic methods is the accumulation of estimator errors when the number of index components is high. Many important stock-market indices are composed of a large number of stocks, such as the MSCI World index (1,642 components), Russell 2000 index (2,000 components), and the Russell 3000 index (3,000 components), *etc.* For these stock-market indices, the number of index components $d$ is usually much larger than the sample size $n$ of available data. Take the Russell 3000 index as an example, we need 58 years to gather more than 3,000 weekly data. However, 58 years ago, lots of these 3,000 stocks did not even exist. Hence, it is impossible to gather more than 3,000 weekly data for all these 3,000 stocks. This is a typical problem for high-dimensional data ($d > n$), which accumulates estimation errors ([44]). As a result, without any control on estimation errors, the tracking strategy obtained by minimizing the *empirical* tracking error might result in a tracking error which substantially deviates from the minimum *true* tracking error. As far as we know, controlling this accumulation of estimation errors is seldom discussed in the existing literature on index tracking. In this chapter, we develop some theoretical foundations to assure that with some conditions the tracking strategy obtained by minimizing the empirical tracking error still leads to a tracking error which converges to the minimum true tracking error in a certain asymptotic sense.

In terms of computational issues, heuristic methods focusing on minimizing the in-sample empirical $E[(R - r_{tp})^2]$ are not practical to track indices of which the number of index components is very large. This is because they usually spend unacceptable amounts of time in practice ([22]). In response, a new tracking error in [22] is defined based on regressing the tracking portfolio return against the index return. Aiming at reducing the fitted intercept to 0 and shifting the fitted coefficient (of the index return) to 1, the index tracking problem is formulated as a mixed-integer linear programming problem. Though the authors of [22] claim that the optimal solution to a mixed-integer linear program can be obtained efficiently by some standard solvers, they do not provide sound economic reasons to support their proposed tracking error. It is also reported in [22] that the tracking portfolios constructed by this method fail to reduce the tracking error $E[(R - r_{tp})^2]$.

In this chapter, factor models are applied to describe stock returns. In general, factor models assume that each stock return is explained by a linear combination of common economic factors plus an extra idiosyncratic risk. Factor models are powerful at explaining stock returns, and they are also simple due to their linear forms. Hence, factor models are widely applied in portfolio management ([109], [103],[42], [6]).

Factor models have been applied to index tracking in [26]. Factors in that work are estimated by a portion of principal components derived from all index components. However, when the number of selected principal components is larger than that of stocks held in the tracking portfolio, the method in [26] fails to construct any tracking portfolio.

In this chapter, we consider factor models with a small number of common factors, such as Sharpe's single-index model, the characteristic line of three-moment CAPM, and the Fama-French three factor model. When stock returns are described by factor models, the MSE of a tracking portfolio can be partitioned into two parts: one only depending on common economic factors and the other depending on idiosyncratic risks from individual stocks. Based on such findings, a 2-stage method is proposed to construct tracking portfolios. Inspired by the work in [22], the first stage relies on a mixed-integer linear program to reduce factors' impacts on MSEs of tracking portfolios, and the second stage constructs a tracking portfolio that minimizes the empirical MSE based on stocks identified in Stage 1. The 2-stage method successfully and efficiently tracks stock-market indices with a large

number of components.

The organization of this chapter is as follows. Section 4.2 formulates the index tracking problem. Section 4.3 introduces factor models which are commonly applied in portfolio analysis. Section 4.4 discusses the 2-stage method to construct tracking portfolios. In Section 4.5, the 2-stage method is applied to track two stock-market indices which are made up of thousands of stocks. Empirical studies suggest that the 2-stage method significantly reduces the out-of-sample empirical MSE, and is more computationally efficient. Section 4.6 concludes this chapter.

## 4.2 Formulation of the Index Tracking Problem

Suppose a stock-market index is composed of $d$ stocks, and denote the return of the $i$th component by $r_i$ for $i = 1, 2, \ldots, d$. Let $\mathbf{w} = (w_1, \ldots, w_d)$ where $w_i$ is the weight of the $i$th index component in the tracking portfolio for $i = 1, 2, \ldots, d$. Hence, the tracking portfolio return $r_{tp}$ is given by

$$r_{tp} = \sum_{i=1}^{d} w_i r_i.$$

Over one generic period, the index tracking problem can be formulated as

$$\min_{\mathbf{w} \in \mathcal{U}} E\left[\left(R - \sum_{i=1}^{d} w_i r_i\right)^2\right], \tag{4.1}$$

where $R$ is the index return and $\mathcal{U}$ is a certain feasible set. In this chapter, $\mathcal{U}$ is defined by the following three constraints, which are usually adopted in the literature of index tracking ([7], [22], [105]).

(a) The no short-selling constraint. In the U.S., there is a margin requirement for short selling stocks. The margin for short selling a stock is 50% of the market value of the

borrowed stock[1], and this is a significant expense. Due to some restrictions on short-selling stocks, such as the alternative uptick rule by the U.S. Securities and Exchange Commission[2], under certain circumstances it is not easy to short sell stocks. Moreover, losses of short selling stocks are unlimited, which is too risky. Thus, we assume that all stock weights in the tracking portfolio are non-negative, that is

$$w_i \geq 0 \text{ for } i = 1, \ldots, d. \tag{4.2}$$

(b) The budget constraint. Since the tracking portfolio only consists of stocks, all stock weights must sum up to 1, that is

$$\sum_{i=1}^{d} w_i = 1. \tag{4.3}$$

(c) The cardinality constraint. Due to their financial budget, sometimes investors are only willing to invest into *at most* $k(\ll d)$ index components. The cardinality constraint is sometimes preferred by portfolio managers, since it is impossible for them to pay detailed attention to a large number of stocks. Mathematically speaking, the cardinality constraint is expressed as

$$\sum_{i=1}^{d} I\{w_i \neq 0\} \leq k, \tag{4.4}$$

where $I\{\cdot\}$ is the indicator function.

Let $\mathbf{w}^*$ be the optimal solution to (4.1) subject to (4.2)-(4.4). Then, positive elements of $\mathbf{w}^*$ indicate stocks to hold in the tracking portfolio and their corresponding weights. However, the joint distribution of $(R, \mathbf{r}')$ is usually unknown, where $\mathbf{r} = (r_1, \ldots, r_d)'$. Thus, $\mathbf{w}^*$ cannot be solved directly. Instead, most literature on index tracking searches for the tracking strategy $\hat{\mathbf{w}}^*$ which minimizes the *empirical* tracking error while subject to (4.2)-(4.4).

[1] http://www.ecfr.gov/cgi-bin/text-idx?SID=7df35b15d3a9d087dc1fbe017048f723&mc=true&node=se12.3.220_112&rgn=div8.

[2] http://www.sec.gov/news/press/2010/2010-26.htm.

Before presenting a quantitative description of the empirical tracking error, it is necessary to introduce an assumption on the relationship between the sample size $n$ and the number of index components $d$. As discussed in Section 4.1, in this chapter we focus on tracking stock-market indices for which $d$ is usually larger than $n$. In order to derive some theoretical results which are friendly to such high-dimensional datasets where $d > n$, we posit the following assumption.

*Assumption 4.1 Let* $d = d(n) = O(n^\alpha)$ *as* $n \to \infty$, *where* $\alpha > 1$.

This order of $d(n)$ in Assumption 4.1 is inherited from [63] to prove Proposition 4.1 in the following. With Assumptions 4.1, the empirical tracking error is given by

$$\frac{1}{n}\sum_{t=1}^{n}\left(R_t - \sum_{i=1}^{d(n)} w_i r_{t,i}\right)^2, \tag{4.5}$$

where $(R_t, r_{t,1}, \ldots, r_{t,d(n)})$ are samples of $(R, \mathbf{r}')$ at time $t$ for $t = 1, \ldots, n$. Note that in the high-dimensional setting $\mathbf{w} = (w_1, \ldots, w_{d(n)})$. Let

$$L_{F^{(n)}}(\mathbf{w}) = E\left[\left(R - \sum_{i=1}^{d(n)} w_i r_i\right)^2\right] \text{ and } L_{\hat{F}^{(n)}}(\mathbf{w}) = \frac{1}{n}\sum_{t=1}^{n}\left(R_t - \sum_{i=1}^{d(n)} w_i r_{t,i}\right)^2, \tag{4.6}$$

where $F^{(n)}$ is the joint distribution of $(R, r_1, \ldots, r_{d(n)})$. Hence, $\mathbf{w}^*$ and $\hat{\mathbf{w}}^*$ can be rewritten as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathcal{U}^{(n)}} L_{F^{(n)}}(\mathbf{w}) \text{ and } \hat{\mathbf{w}}^* = \arg\min_{\mathbf{w}\in\mathcal{U}^{(n)}} L_{\hat{F}^{(n)}}(\mathbf{w}), \tag{4.7}$$

where, according to (4.2)-(4.4),

$$\begin{aligned}\mathcal{U}^{(n)} = \Bigg\{\mathbf{w} = (w_1, \ldots, w_{d(n)}) : \quad & 0 \le w_i \le 1, \text{ for } i = 1, \ldots, d(n), \\ & \sum_{i=1}^{d(n)} w_i = 1, \\ & \sum_{i=1}^{d(n)} I\{w_i \neq 0\} \le k\Bigg\},\end{aligned} \tag{4.8}$$

and $k(\ll d(n))$ is a fixed positive integer.

In general, there is no direct relationship between $\mathbf{w}^*$ and $\hat{\mathbf{w}}^*$, since they are optimal solutions to minimize different objective functions. However, the performance of the tracking strategy $\mathbf{w}^*$ must be evaluated by $L_{F^{(n)}}(\hat{\mathbf{w}}^*)$. In order to quantify the gap between $L_{F^{(n)}}(\hat{\mathbf{w}}^*)$ and $L_{F^{(n)}}(\mathbf{w}^*)$, Proposition 4.1 shows that under certain conditions $L_{F^{(n)}}(\hat{\mathbf{w}}^*) - L_{F^{(n)}}(\mathbf{w}^*)$ converges to 0 in probability as $n \to \infty$.

**Proposition 4.1.** In addition to Assumptions 4.1, let us further assume that

(a) random vectors $(R_t, r_{t,1}, \ldots, r_{t,d(n)})$ at different time $t$ for $t = 1, 2, \ldots, n$ are independent and identically distributed (i.i.d.) samples of the random vector $(R_t, r_1, \ldots, r_{d(n)})$, which follows a joint distribution $F^{(n)}$.

(b)

$$E\left[\left(\max_{1\le i,j\le d(n)} |r_i r_j - \sigma_{ij}|\right)^2\right] \le \tilde{M} < \infty \text{ and } E\left[\left(\max_{1\le i\le d(n)} |Rr_i - \sigma_i|\right)^2\right] \le \tilde{M} < \infty,$$

where $\sigma_{ij} = E[r_i r_j]$, $\sigma_i = E[Rr_i]$, and $\tilde{M}$ is a constant.

Then

$$L_{F^{(n)}}(\hat{\mathbf{w}}^*) - L_{F^{(n)}}(\mathbf{w}^*) \xrightarrow{p} 0, \text{ as } n \to \infty,$$

where $L_{F^{(n)}}(\mathbf{w})$ is defined in (4.6), $\mathbf{w}^*$ and $\hat{\mathbf{w}}^*$ are defined in (4.7).

*Proof.* In general, let us define

$$\begin{aligned} \mathbf{w}^*_{\mathcal{B}} &= \arg\min_{\mathbf{w}\in\mathcal{B}} L_{F^{(n)}}(\mathbf{w}), \\ \hat{\mathbf{w}}^*_{\mathcal{B}} &= \arg\min_{\mathbf{w}\in\mathcal{B}} L_{\hat{F}^{(n)}}(\mathbf{w}), \end{aligned}$$

where $\mathcal{B}$ is a general feasible set. Further, let $||\mathbf{x}||_1 = \sum_i |x_i|$ for any vector $\mathbf{x} = (x_1, \ldots, x_{d(n)})$. According to Theorem 3 in [63], under the assumptions stated in this proposition, as long as

$$\mathcal{B} \subset B^n_{b(n)} = \left\{\mathbf{w} : ||\mathbf{w}||_1 \le b(n)\right\},$$

where $b(n) = o\big((\frac{n}{\log(n)})^{1/4}\big)$, we have

$$L_{F^{(n)}}(\hat{\mathbf{w}}^*_{\mathcal{B}}) - L_{F^{(n)}}(\mathbf{w}^*_{\mathcal{B}}) \xrightarrow{p} 0 \text{ as } n \to \infty.$$

In order to finish this proof, it suffices to show that $\mathcal{U}^{(n)} \subset B^n_{b(n)}$ where $\mathcal{U}^{(n)}$ is defined in (4.8). Note that for any $\mathbf{w} \in \mathcal{U}^{(n)}$, it holds that $\sum_{i=1}^{d(n)} w_i = 1$ and $0 \leq w_i \leq 1$ for $i = 1, \ldots, d(n)$. Hence,

$$||\mathbf{w}||_1 = \sum_{i=1}^{d(n)} |w_i| = \sum_{i=1}^{d(n)} w_i = 1 \leq b(n) = o\big((\frac{n}{\log(n)})^{1/4}\big)$$

for any sufficiently large $n$. Hence, $\mathcal{U}^{(n)} \subset B^n_{b(n)}$, and this completes the proof. □

Although financial data might demonstrate serial dependence, we use Assumption $(a)$ in Proposition 4.1 as a benchmark. Due to the complexity of our index tracking formulation in the high-dimensional setting, it is challenging to develop any meaningful theory without this i.i.d assumption.

Proposition 4.1 provides a motivation to search for $\hat{\mathbf{w}}^*$ that minimizes the empirical tracking error. Under the assumptions in Proposition 4.1, $\hat{\mathbf{w}}^*$ leads to a tracking error $L_{F^{(n)}}(\hat{\mathbf{w}}^*)$ which is close to the true minimum tracking error $L_{F^{(n)}}(\mathbf{w}^*)$ when the sample size $n$ is sufficiently large, even $d = d(n) = O(n^\alpha)$. As far as we know, such a motivation is seldom discussed in the existing literature on index tracking.

However, $\hat{\mathbf{w}}^*$ defined in (4.7) is a solution to a mixed-integer quadratic program ([7]). In practice, when the number of index components $d$ is large, it is challenging to obtain $\hat{\mathbf{w}}^*$ efficiently. Rather than directly solving a mixed-integer quadratic program to obtain $\hat{\mathbf{w}}^*$, in this chapter we describe $r_i$ for $i = 1, \ldots, d$ by factor models. Under this assumption, a 2-stage method is introduced to construct tracking portfolios. The 2-stage method is computationally efficient, and it significantly reduces out-of-sample empirical MSEs.

## 4.3 Factor Models in Portfolio Analysis

Factor models assume that a stock return is determined by common economic factors as well as the company's individual business performance, or idiosyncratic risk, which is independent of economic factors. Specifically, in a factor model the return of stock $i$, $r_i$, is given by

$$r_i = \alpha_i + \sum_j \gamma_{ij} F_j + \varepsilon_i, \text{ for } i = 1, \ldots, d,$$

where $\alpha_i$ is a constant, $F_j$ is the $j$-th factor, the weight of $j$-th factor for stock $i$ is $\gamma_{ij}$ for all $i, j$. The idiosyncratic risk of stock $i$ is $\varepsilon_i$, and this is a random variable with zero mean and finite variance $\sigma^2_{\varepsilon_i}$ for all $i$. Idiosyncratic risks are independent of factors, and $cov(\varepsilon_{i_1}, \varepsilon_{i_2}) = 0$ for $i_1 \neq i_2$.

One of the simplest factor models in portfolio analysis is Sharpe's single-index model introduced in [109]. In this model, the stock return only depends on one factor, which is the return of a stock-market index. Specifically, the single-index model is given by

$$r_i = \alpha_i + \beta_i R + \varepsilon_i, \text{ for } i = 1, \ldots, d, \tag{4.9}$$

where $R$ is a stock-market index return, and $\beta_i$ is a constant coefficient for stock $i$. The relationship in (4.9) is attractive in the mean-variance framework, since it simplifies the mathematical and computational analysis but also explains a large portion of stock return variations (see [109]).

The quadratic characteristic line of three-moment CAPM, which is introduced in [81], is another factor model in portfolio analysis. In addition to the market portfolio return $r_M$, the quadratic characteristic line includes $r^2_M$ as another factor in the model. [81] assumes that the investor's expected utility is defined over the first three moments of the probability distribution of end of period wealth, and the return of a portfolio that consists of risky assets is asymmetric. Also, under a series of other assumptions, the authors of [81] derive the three-moment CAPM, and the quadratic characteristic line of three-moment CAPM is

given by

$$r_i - r_f = c_{0i} + c_{1i}(r_M - r_f) + c_{2i}(r_M - E[r_M])^2 + \varepsilon_i, \text{ for } i = 1, \ldots, d, \tag{4.10}$$

where $r_f$ is the return of the risk-free asset, and $c_{0i}, c_{1i}$, and $c_{2i}$ are constant coefficients. After some algebra, equation (4.10) can be rewritten as

$$r_i = \tilde{c}_{0i} + \tilde{c}_{1i} r_M + \tilde{c}_{2i} r_M^2 + \varepsilon_i, \text{ for } i = 1, \ldots, d,$$

where $\tilde{c}_{0i}, \tilde{c}_{1i}$, and $\tilde{c}_{2i}$ are constant coefficients. In practice, the market portfolio return in the three-moment CAPM is usually replaced by a stock-market index ([81] and [68]). Replacing the market portfolio by the stock-market index, the characteristic line of the three-moment CAPM is given by

$$r_i = \tilde{c}_{0i} + \tilde{c}_{1i} R + \tilde{c}_{2i} R^2 + \varepsilon_i, \text{ for } i = 1, \ldots, d. \tag{4.11}$$

Empirical tests in [81] suggest that the three-moment CAPM explains stock return variations well.

Another popular factor model in portfolio analysis is the Fama-French three factor model in [42]. The authors of [42] emphasize the importance of a company's size (or equivalently the market capitalization) and book-to-market ratio in explaining its stock return, and suggest that stock returns can be well explained by the stock-market index return, the size factor, and the book-to-market ratio factor. The size factor is mimicked by the return of a portfolio which longs small size stocks and shorts large size stocks, and the book-to-market ratio factor is mimicked by the return of a portfolio which longs high book-to-market ratio stocks and shorts low book-to-market ratio stocks. Denoting by $SMB$ the size factor and $HML$ the book-to-market ratio factor, the Fama-French three factor model is given by

$$r_i = \alpha_i + \beta_i R + \gamma_{i1} SMB + \gamma_{i2} HML + \varepsilon_i, \text{ for } i = 1, \ldots, d. \tag{4.12}$$

Many empirical studies that support the Fama-French three factor model are carried out in [42]. According to [43], the Fama-French three factor is widely used in empirical research.

Since all of (4.9), (4.11) and (4.12) have the factor $R$, in this chapter the general form of factor models is rewritten as

$$r_i = \alpha_i + \beta_i R + \sum_{j=1}^{q} \gamma_{ij} F_j + \varepsilon_i, \text{ for } i = 1, \ldots, d, \tag{4.13}$$

where $q$ is a fixed positive integer.

## 4.4 A 2-Stage Method to Construct Tracking Portfolios

### 4.4.1 Decomposition of The Tracking Error

Assume that stock returns are explained by (4.13). Note that $r_{tp} = \sum_{i=1}^{d} w_i r_i$. In a generic period, the difference between the index return and the portfolio return, $R - r_{tp}$, is given by

$$R - r_{tp} = -\left[\sum_{i=1}^{d} w_i \alpha_i + R\Big(\sum_{i=1}^{d} w_i \beta_i - 1\Big) + \sum_{j=1}^{q} F_j \Big(\sum_{i=1}^{d} w_i \gamma_{ij}\Big) + \sum_{i=1}^{d} w_i \varepsilon_i\right].$$

Therefore, the tracking error can be written as

$$E[(R - r_{tp})^2] = MSE_F + MSE_I, \tag{4.14}$$

where

$$MSE_F = E\left[\Big(\sum_{i=1}^{d} w_i \alpha_i + R\big(\sum_{i=1}^{d} w_i \beta_i - 1\big) + \sum_{j=1}^{q} F_j \big(\sum_{i=1}^{d} w_i \gamma_{ij}\big)\Big)^2\right], \tag{4.15}$$

$$MSE_I = \sum_{i=1}^{d} w_i^2 \sigma_{\varepsilon_i}^2. \tag{4.16}$$

Again, the joint distribution of the $R$, $F_j$'s, and $\varepsilon_i$'s for all $i, j$ is usually unknown,

so that we cannot directly obtain the tracking strategy $\mathbf{w}^*$ by minimizing $E[(R - r_{tp})^2]$ in (4.14) subject to $\mathcal{U}^{(n)}$ in (4.8). Instead, we turn to search for $\hat{\mathbf{w}}^*$ by minimizing the empirical tracking error under factor models subject to $\mathcal{U}^{(n)}$.

In order to present the empirical tracking error within the setting of high-dimensional data, we still posit Assumption 4.1 with factor models. Hence, in this chapter the empirical tracking error, or empirical MSE, with factor models is given by

$$\frac{1}{n}\sum_{t=1}^{n}\left(\sum_{i=1}^{d(n)} w_i\hat{\alpha}_i + R_t\Big(\sum_{i=1}^{d(n)} w_i\hat{\beta}_i - 1\Big) + \sum_{j=1}^{q} F_{t,j}\Big(\sum_{i=1}^{d(n)} w_i\hat{\gamma}_{ij}\Big)\right)^2 + \sum_{i=1}^{d} w_i^2\hat{\sigma}_{\varepsilon_i}^2, \tag{4.17}$$

where $(R_t, F_{t,1}, \ldots, F_{t,q})$ are samples of $(R, F_1, \ldots, F_q)$ observed at time $t$ for $t = 1, \ldots, n$, quantities $\hat{\alpha}_i$, $\hat{\beta}_i$, $\hat{\gamma}_{ij}$, $\hat{\sigma}^2_{\varepsilon_i}$ are ordinary least square (OLS) estimators of their counterparts in (4.15) and (4.16) for all $i, j$.

Analogous to Proposition 4.1, we quantify the gap between $L_{F^{(n)}}(\hat{\mathbf{w}}^*)$ and $L_{F^{(n)}}(\mathbf{w}^*)$ under the assumption of factor models in Proposition 4.2. Before doing that, it is necessary to simplify some notations.

Suppose the number of $F_j$'s, *i.e.* $q$, is a fixed finite number. Let $r_{d(n)+1} = R$, $\tilde{\mathbf{r}} = (r_1, \ldots, r_{d(n)+1})'$, $w_{d(n)+1} = -1$, $\boldsymbol{\xi} = (w_1, \ldots, w_{d(n)+1})'$, and $\mathbf{f} = (1, R, F_1, \ldots, F_q)'$. For simplicity, the general factor model in (4.13) is rewritten as

$$\tilde{\mathbf{r}} = B\mathbf{f} + \boldsymbol{\varepsilon}, \tag{4.18}$$

where $B = (\mathbf{b}_1, \ldots, \mathbf{b}_{d(n)+1})'$, $\mathbf{b}_i = (\alpha_i, \beta_i, \gamma_{i1}, \ldots, \gamma_{iq})'$ for $i = 1, \ldots, d(n)$, $b_{d(n)+1} = (0, 1, 0, \ldots, 0)'$, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_{d(n)}, 0)'$. Consequently, we can rewrite $L_{F^{(n)}}(\mathbf{w})$ defined in (4.6) as

$$\begin{aligned} L_{F^{(n)}}(\mathbf{w}) &= E\left[\left(R - \sum_{i=1}^{d(n)} w_i r_i\right)^2\right] = E\left[\left(\sum_{i=1}^{d(n)+1} w_i r_i\right)^2\right] \\ &= \boldsymbol{\xi}' E\left[\tilde{\mathbf{r}}\tilde{\mathbf{r}}'\right]\boldsymbol{\xi} \\ &= \boldsymbol{\xi}'(B\Sigma_f B' + \Sigma_\varepsilon)\boldsymbol{\xi}, \end{aligned} \tag{4.19}$$

where $F^{(n)}$ is the joint distribution of $(R, F_1, \ldots, F_q, r_1, \ldots, r_{d(n)})$, $\boldsymbol{\xi} = (w_1, \ldots, w_{d(n)}, -1)$, $\Sigma_f = E[\mathbf{f}\mathbf{f}']$, $\Sigma_\varepsilon = E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}']$.

Denote the sample matrix of $\tilde{\mathbf{r}}$ by $Y = (Y_1, \ldots, Y_{d(n)+1})$ and $Y_i = (Y_{1i}, \ldots, Y_{ni})'$ for $i = 1, \ldots, d(n)+1$. Let the sample matrix of $\mathbf{f}$ be $X$, where $X = (\mathbf{e}, \mathbf{R}, \mathbf{F}_1, \ldots, \mathbf{F}_q)$, $\mathbf{e}$ is a $n$-column vector of 1's, $\mathbf{R} = (R_1, \ldots, R_n)'$, $\mathbf{F}_j = (\mathbf{F}_{1,j}, \ldots, \mathbf{F}_{n,j})'$ for $j = 1, \ldots, q$. With factor models, according to (4.17), $L_{\hat{F}^{(n)}}(\mathbf{w})$ defined in (4.6) can be rewritten as

$$L_{\hat{F}^{(n)}}(\mathbf{w}) = \boldsymbol{\xi}'(\hat{B}\hat{\Sigma}_f\hat{B}' + \hat{\Sigma}_\varepsilon)\boldsymbol{\xi}, \tag{4.20}$$

where $\hat{\Sigma}_f = \frac{1}{n}X'X$, $\hat{B} = (\hat{\mathbf{b}}_1, \ldots, \hat{\mathbf{b}}_{d(n)+1})'$, and $\hat{\Sigma}_\varepsilon$ is a diagonal matrix with diagonal elements $\hat{\sigma}^2_{\varepsilon_1}, \ldots, \hat{\sigma}^2_{\varepsilon_{d(n)+1}}$. For $i = 1, \ldots, d(n)+1$, the vector of OLS estimators $\hat{\mathbf{b}}_i$ is defined by $\hat{\mathbf{b}}_i = (X'X)^{-1}X'Y_i$, and the OLS estimator of $\sigma^2_{\varepsilon_i}$ is $\hat{\sigma}^2_{\varepsilon_i} = \frac{1}{n-(q+1)-1}(Y_i - X\hat{\mathbf{b}}_i)'(Y_i - X\hat{\mathbf{b}}_i)$. We note that the above formulas for OLS estimators are also valid for $i = d(n)+1$ because

$$Y_{d(n)+1} = \mathbf{R} = X\mathbf{b}_{d(n)+1},$$

so that $\hat{\mathbf{b}}_{d(n)+1} = \mathbf{b}_{d(n)+1}$ and $\hat{\sigma}^2_{\varepsilon_{d(n)+1}} = \sigma^2_{\varepsilon_{d(n)+1}} = 0$.

Under the factor model assumption, suppose $\mathbf{w}^*$ and $\hat{\mathbf{w}}^*$ follow the definition in (4.7). In order to measure the gap between $L_{F^{(n)}}(\hat{\mathbf{w}}^*)$ and $L_{F^{(n)}}(\mathbf{w}^*)$, Proposition 4.2 show that under some assumptions

$$L_{F^{(n)}}(\hat{\mathbf{w}}^*) - L_{F^{(n)}}(\mathbf{w}^*) \xrightarrow{p} 0 \text{ as } n \to \infty.$$

**Proposition 4.2.** In addition to Assumption 4.1, let us further assume that

(a) random vectors $(R_t, F_{t,1}, \ldots, F_{t,q}, r_{t,1}, \ldots, r_{t,d(n)})$ at time $t$ for $t = 1, \ldots, n$ are i.i.d. samples from the random vector $(R, F_1, \ldots, F_q, r_1, \ldots, r_{d(n)})$, which follows a joint distribution $F^{(n)}$.

(b) stock returns follows the factor model (4.18),

(c) there exists a constant $\tilde{M} > 0$, such that $E[r_i^2] < \tilde{M}$ for any $i = 1, \ldots, d(n)$ and

$E[f_j^2] < \tilde{M}$ for any $f_j \in \mathbf{f}$, each element in $\mathbf{b}_i$ is less than or equal to $\tilde{M}$ for $i = 1, \ldots, d(n)$. The matrix $\Sigma_f = E[\mathbf{f}\mathbf{f}']$ is positive definite,

(d) there exist constants $\psi_1 > 0$ and $\phi_1 > 0$ such that for any $\delta > 0$ and any $f_j \in \mathbf{f}$,

$$Pr(|f_j| > \delta) \leq \exp\big(-(\delta/\phi_1)^{\psi_1}\big),$$

(e) $E[\varepsilon_i] = 0$, $\sigma^2_{\varepsilon_i} = Var(\varepsilon_i) < \infty$, the random noise $\varepsilon_i$ is independent of common factors for $i = 1, \ldots, d(n)$, and $cov(\varepsilon_i, \varepsilon_j)$=0 for $i \neq j$,

(f) there exist $\psi_2 > 0$ and $\phi_2 > 0$, such that for any $\delta > 0$

$$Pr(|\varepsilon_i| > \delta) \leq \exp\big(-(\delta/\phi_2)^{\psi_2}\big), \text{ for } i = 1, \ldots, d(n),$$

(g) ordinary least square (OLS) estimators are used to estimate $\hat{\mathbf{b}}_i$ and $\hat{\sigma}_{\varepsilon_i}$ for $i = 1, 2, \ldots, d(n)+1$.

(h) $||\frac{1}{d(n)}B'B - \Omega|| = o(1)$ for some $q \times q$ positive definite matrix $\Omega$, where $||A|| = \lambda^{1/2}_{\max}(A'A)$ and $\lambda_{\max}(D)$ denotes the largest eigenvalue of a matrix $D$.

Suppose that $L_{F^{(n)}}(\mathbf{w})$ and $L_{\hat{F}^{(n)}}(\mathbf{w})$ are defined in (4.19) and (4.20) respectively, and $\mathbf{w}^*$ and $\hat{\mathbf{w}}^*$ are defined in (4.7). Then, $\forall \varepsilon > 0$,

$$Pr\left\{|L_{F^{(n)}}(\hat{\mathbf{w}}^*) - L_{F^{(n)}}(\mathbf{w}^*)| > 2\varepsilon\right\} \leq O\left(\frac{1}{n^{2\alpha}} + \frac{1}{n^2}\right),$$

which implies

$$L_{F^{(n)}}(\hat{\mathbf{w}}^*) - L_{F^{(n)}}(\mathbf{w}^*) \xrightarrow{p} 0 \text{ as } n \to \infty.$$

*Proof.* Note that, by definition, $L_{\hat{F}_n}(\hat{\mathbf{w}}^*) - L_{\hat{F}_n}(\mathbf{w}^*) < 0$. Let $\Sigma = B\Sigma_f B' + \Sigma_\varepsilon$ and

$\hat{\Sigma} = \hat{B}\hat{\Sigma}_f\hat{B}' + \hat{\Sigma}_\varepsilon$. We have

$$\begin{aligned}
0 &\leq L_{F_n}(\hat{\mathbf{w}}^*) - L_{F_n}(\mathbf{w}^*) \\
&= L_{F_n}(\hat{\mathbf{w}}^*) - L_{\hat{F}_n}(\hat{\mathbf{w}}^*) + L_{\hat{F}_n}(\hat{\mathbf{w}}^*) - L_{\hat{F}_n}(\mathbf{w}^*) + L_{\hat{F}_n}(\mathbf{w}^*) - L_{F_n}(\mathbf{w}^*) \\
&\leq |L_{F_n}(\hat{\mathbf{w}}^*) - L_{\hat{F}_n}(\hat{\mathbf{w}}^*)| + |L_{\hat{F}_n}(\mathbf{w}^*) - L_{F_n}(\mathbf{w}^*)| \\
&\leq 2 \sup_{\mathbf{w}\in\mathcal{U}^{(n)}} |L_{F_n}(\mathbf{w}) - L_{\hat{F}_n}(\mathbf{w})| \\
&= 2 \sup_{\mathbf{w}\in\mathcal{U}^{(n)}} |\boldsymbol{\xi}'(\Sigma - \hat{\Sigma})\boldsymbol{\xi}| \\
&\leq 2||\Sigma - \hat{\Sigma}||_{\max} \sup_{\mathbf{w}\in\mathcal{U}^{(n)}} |\sum_{i,j} \xi_i\xi_j| \\
&\leq 2||\Sigma - \hat{\Sigma}||_{\max} \sup_{\mathbf{w}\in\mathcal{U}^{(n)}} ||\boldsymbol{\xi}||_1^2,
\end{aligned}$$

where $||A||_{\max} = \max_{i,j}\{a_{ij}\}$ for any matrix $A = (a_{ij})_{i,j}$. Then for any $\varepsilon > 0$, we have

$$Pr\left\{|L_{F_n}(\hat{\mathbf{w}}^*) - L_{F_n}(\mathbf{w}^*)| > 2\varepsilon\right\} \leq Pr\left\{||\Sigma - \hat{\Sigma}||_{\max} \sup_{\mathbf{w}\in\mathcal{U}^{(n)}} ||\boldsymbol{\xi}||_1^2 > \varepsilon\right\}. \tag{4.21}$$

Given any constant $\psi \in (0, \frac{1}{2})$, we have $\varepsilon > \frac{1}{n^\psi}$ for sufficiently large $n$, so that

$$Pr\left\{||\Sigma - \hat{\Sigma}||_{\max} \sup_{\mathbf{w}\in\mathcal{U}^{(n)}} ||\boldsymbol{\xi}||_1^2 > \varepsilon\right\} \leq Pr\left\{||\Sigma - \hat{\Sigma}||_{\max} \sup_{\mathbf{w}\in\mathcal{U}^{(n)}} ||\boldsymbol{\xi}||_1^2 > \frac{1}{n^\psi}\right\}. \tag{4.22}$$

With the assumptions in this proposition, we can apply Theorem 3.2 from [46], in conjunction with the assumption $d(n) = O(n^\alpha)$ where $\alpha > 1$, to obtain

$$Pr\left\{||\Sigma - \hat{\Sigma}||_{\max} \geq \left(\frac{C(\alpha\log(n) + q^2\log(n))}{n}\right)^{1/2}\right\} = O\left(\frac{1}{n^{2\alpha}} + \frac{1}{n^2}\right),$$

where $C > 0$ is a constant. Hence, as long as $\sup_{\mathbf{w}\in\mathcal{U}^{(n)}} ||\boldsymbol{\xi}||_1 \leq \left(\frac{n^{1-2\psi}}{C(\alpha\log(n)+q^2\log(n))}\right)^{1/4}$ for

sufficiently large $n$, we have

$$\begin{aligned} & Pr\left\{||\Sigma-\hat{\Sigma}||_{\max}\sup_{\mathbf{w}\in\mathcal{U}^{(n)}}||\boldsymbol{\xi}||_1^2>\frac{1}{n^{\psi}}\right\} \\ \leq\ & Pr\left\{||\Sigma-\hat{\Sigma}||_{\max}\left(\frac{n^{1-2\psi}}{C(\alpha\log(n)+q^2\log(n))}\right)^{1/2}>\frac{1}{n^{\psi}}\right\} \\ =\ & Pr\left\{||\Sigma-\hat{\Sigma}||_{\max}\geq\left(\frac{C(\alpha\log(n)+q^2\log(n))}{n}\right)^{1/2}\right\} \\ =\ & O\left(\frac{1}{n^{2\alpha}}+\frac{1}{n^2}\right). \end{aligned} \tag{4.23}$$

According to (4.21), (4.22), and (4.23), as long as $\sup_{\mathbf{w}\in\mathcal{U}^{(n)}}||\boldsymbol{\xi}||_1\leq\left(\frac{n^{1-2\psi}}{C(\alpha\log(n)+q^2\log(n))}\right)^{1/4}$, we have

$$Pr\left\{|L_{F_n}(\hat{\mathbf{w}}^*)-L_{F_n}(\mathbf{w}^*)|>2\varepsilon\right\}\to 0,\ \text{as } n\to\infty.$$

Hence, in order to complete the proof, it suffices to show that

$$\sup_{\mathbf{w}\in\mathcal{U}^{(n)}}||\boldsymbol{\xi}||_1\leq\left(\frac{n^{1-2\psi}}{C(\alpha\log(n)+q^2log(n))}\right)^{1/4}.$$

By the definition of $\boldsymbol{\xi}$, we have $||\boldsymbol{\xi}||_1=1+\sum_{i=1}^{d(n)}|w_i|$. According to the definition of $\mathcal{U}^{(n)}$ in (4.8), for any $\mathbf{w}\in\mathcal{U}^{(n)}$,

$$||\boldsymbol{\xi}||_1=1+\sum_{i=1}^{d(n)}|w_i|=1+\sum_{i=1}^{d(n)}w_i=2,$$

so that

$$\sup_{\mathbf{w}\in\mathcal{U}^{(n)}}||\boldsymbol{\xi}||_1\leq\left(\frac{n^{1-2\psi}}{C(\alpha\log(n)+q^2\log(n))}\right)^{1/4},$$

for any sufficiently large $n$. This completes the proof. □

Proposition 4.2 points out that, under the assumptions of factor models, $\hat{\mathbf{w}}^*$ leads to a

tracking error $L_{F^{(n)}}(\hat{\mathbf{w}}^*)$ which is close to the true minimum tracking error $L_{F^{(n)}}(\mathbf{w}^*)$ when the sample size $n$ is sufficiently large, even though $d = d(n) = O(n^\alpha)$. However, $\hat{\mathbf{w}}^*$ defined in (4.7) with factor models is also a solution to a mixed-integer quadratic programming. In practice, it is challenging to obtain $\hat{\mathbf{w}}^*$ in an efficient way when the number of index component is large.

Nevertheless, the decomposition of MSE in (4.14) inspires our 2-stage method that is described later in Section 4.4.2. According to (4.14), the tracking error is partitioned into two parts: $\text{MSE}_F$ which only depends on common economic factors, and $\text{MSE}_I$ that only depends on idiosyncratic risks. Rather than searching for $\mathbf{w}^*$ (or $\hat{\mathbf{w}}^*$) by directly minimizing the tracking error (or the empirical tracking error), we control $\text{MSE}_F$ and $\text{MSE}_I$ individually by our 2-stage method.

In the 2-stage method, we focus on controlling $\text{MSE}_F$ in (4.15). There are some relevant studies to support that $\text{MSE}_F$ contributes the majority of $E[(R - r_{tp})^2]$. Empirical studies in [22] provide evidence that $\text{MSE}_F$ explains a large portion of $E[(R - r_{tp})^2]$. Some studies also show that large portfolios significantly reduce the idiosyncratic risk, which suggests that $\text{MSE}_I$ is less important in our case. It is observed in [41] that the standard deviation of an equally weighted portfolio return, of which stocks are randomly selected, decreases rapidly to a positive asymptote as the number of stocks in the portfolio increases. Moreover, 10 randomly selected stocks are usually able to construct a well diversified portfolio, and adding extra stocks does not have significant improvement on reducing the portfolio return's standard deviation. Studies in [38], [126], and [120] suggest that more stocks are needed to diversify a portfolio, but all these studies imply that 30 to 40 stocks are necessary to construct a well-diversified portfolio. In the framework of factor models, diversifiable risk in a portfolio is $\text{MSE}_I$ (see [41]). Note that this chapter aims at tracking stock-market indices with a large number of components, usually larger than 40 (see examples in [22]). Hence, $\text{MSE}_I$ of a tracking portfolio is small if the tracking portfolio is benchmarked to an index with a large number of components.

Our 2-stage method relies on an upper bound of $\text{MSE}_F$. Applying Minkowski's inequal-

ity to $\text{MSE}_F$ leads to

$$\sqrt{\text{MSE}_F} \leq \left| \sum_{i=1}^{d(n)} w_i \alpha_i \right| + \left| \sum_{i=1}^{d(n)} w_i \beta_i - 1 \right| \sqrt{E[R^2]} + \sum_{j=1}^{q} \left| \sum_{i=1}^{d(n)} w_i \gamma_{ij} \right| \sqrt{E[F_j^2]}. \tag{4.24}$$

In Stage 1, $\text{MSE}_F$ is controlled by minimizing a generalized form of the right hand side of (4.24) by a mixed-integer linear program. The optimal solution to a mixed-integer linear program can be obtained in an efficient way by standard solvers ([22]). Based on the stocks identified in Stage 1, Stage 2 determines weights of the selected stocks by minimizing the structured empirical MSE in (4.17) via a standard quadratic program. In practice, when the number of index components is large, this 2-stage method is more computationally efficient than constructing a tracking portfolio by solving a mixed-integer quadratic program.

### 4.4.2 2-Stage Method

Based on analysis from Section 4.4.1, a 2-stage method is introduced to construct tracking portfolios in this section.

*Stage 1*

We avoid controlling $\text{MSE}_F$ via a mixed-integer quadratic program, because it is computationally difficult. Inspired by (4.24), one way to control $\text{MSE}_F$ is to minimize the upper bound on $\text{MSE}_F$ in (4.24), which is linear with respect to $(w_1, \ldots, w_{d(n)})$. However, the true values of $E[R^2]$ and the $E[F_j^2]$'s are unknown in general. Plugging any estimators of $E[R^2]$ and the $E[F_j^2]$'s into the right hand side of (4.24) and minimizing it usually leads to suboptimal solutions. Inspired by [22], we control a generalized form of the right hand

side of (4.24) by a tracking strategy $\mathbf{w}_{f1}^*(\boldsymbol{\lambda})$, which is a solution to the following problem:

$$\min_{\mathbf{w},\mathbf{y}} \quad \lambda_\alpha \left| \sum_{i=1}^{d(n)} w_i \alpha_i \right| + \lambda_\beta \left| \sum_{i=1}^{d(n)} w_i \beta_i - 1 \right| + \sum_{j=1}^{q} \lambda_j \left| \sum_{i=1}^{d(n)} w_i \gamma_{ij} \right|, \tag{4.25}$$

$$s.t. \quad 0 \leq w_i \leq y_i, \text{ for } i = 1, \ldots, d(n), \tag{4.26}$$

$$\sum_{i=1}^{d(n)} w_i = 1, \tag{4.27}$$

$$y_i \in \{0, 1\}, \text{ for } i = 1, \ldots, d(n), \tag{4.28}$$

$$\sum_{i=1}^{d(n)} y_i = k, \tag{4.29}$$

where $\lambda_\alpha$, $\lambda_\beta$, and the $\lambda_j$'s are positive numbers given exogenously as tuning parameters, $\boldsymbol{\lambda}$ is the vector of these tuning parameters, and $\mathbf{y} = (y_1, \ldots, y_{d(n)})$. Constraints (4.26)-(4.29) is a reformulation of the feasible set $\mathcal{U}^{(n)}$ in (4.8) ([7]). Working with empirical data, it is necessary to replace $\alpha_i$'s, $\beta_i$'s, and $\gamma_{ij}$'s for all $i, j$ by their OLS estimators. Hence, we can obtain $\hat{\mathbf{w}}_{f1}^*(\boldsymbol{\lambda})$ which is given by

$$\min_{\mathbf{w},\mathbf{y}} \lambda_\alpha \left| \sum_{i=1}^{d(n)} w_i \hat{\alpha}_i \right| + \lambda_\beta \left| \sum_{i=1}^{d(n)} w_i \hat{\beta}_i - 1 \right| + \sum_{j=1}^{q} \lambda_j \left| \sum_{i=1}^{d(n)} w_i \hat{\gamma}_{ij} \right|, \tag{4.30}$$

subject to (4.26)-(4.29). This is a mixed-integer linear program, of which the optimal solution can be obtained by standard solvers in an efficient way ([22]).

Details on how to determine tuning parameters are discussed later in Section 4.4.3.

*Stage 2*

The most important contribution of Stage 1 is that the positive weights in $\hat{\mathbf{w}}_{f1}^*(\boldsymbol{\lambda})$ identify a subset of stocks that are capable of controlling $\text{MSE}_F$. Hence, we can use these identified stocks to construct a tracking portfolio, and their final weights in the tracking portfolio are determined in Stage 2 by minimizing the empirical structured MSE (4.17) subject to (4.2) and (4.3). This is just a quadratic program.

Specifically, suppose that there are $k'(\leq k)$ positive weights in $\hat{\mathbf{w}}^*_{f1}(\boldsymbol{\lambda})$, that is $k'$ stocks are identified in Stage 1. We relabel those identified $k'$ stocks as $1, \ldots, k'$, and their final weights in the tracking portfolio should be determined by

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{n}\sum_{t=1}^{n}\Big(\sum_{i=1}^{k'} w_i\hat{\alpha}_i + R_t\big(\sum_{i=1}^{k'} w_i\hat{\beta}_i - 1\big) + \sum_{j=1}^{q} F_{t,j}\big(\sum_{i=1}^{k'} w_i\hat{\gamma}_{ij}\big)\Big)^2 + \sum_{i=1}^{k'} w_i^2\hat{\sigma}^2_{\varepsilon_i}, \quad (4.31) \\ s.t. \quad & 0 \leq w_i \leq 1, \text{ for } i = 1, \ldots, k', \\ & \sum_{i=1}^{k'} w_i = 1. \end{aligned}$$

### 4.4.3 Determine Tuning Parameters

Tuning parameters in (4.30) play a critical role in the 2-stage method. One choice for the tuning parameter values is to let $\lambda_\alpha = 1$ and $\lambda_\beta$, $\lambda_j$'s be the sample means of $R^2$ and the $F_j^2$'s that are denoted by $\hat{E}[R^2]$ and $\hat{E}[F_j^2]$ for $j = 1, 2, \ldots, q$. This set of values corresponds to the right hand side of (4.24). However, our empirical studies suggest that these choices, compared to tuning parameters given by the following method, do not lead to smaller tracking errors. Our empirical studies show that the range and magnitudes of $\{\hat{\alpha}\}_{i=1}^d$ are much smaller than those of $\left\{\hat{\beta}_i - 1\right\}_{i=1}^d$ and $\{\hat{\gamma}_{ij}\}_{i=1}^d$. Hence, the choice $\lambda_\alpha = 1$ falsely puts too much weights on $\left|\sum_{i=1}^{d(n)} w_i\hat{\alpha}_i\right|$ in the objective function (4.30), while $\lambda_\beta = \hat{E}[R^2]$ and $\lambda_j = \hat{E}[F_j^2]$ for all $j$, which are around 0.0001, assign overly small weights to other summands in the objective function. This suggests that, in order to control $\text{MSE}_F$ and the structured MSE (4.14) in a better way, $\lambda_\alpha$ in (4.30) should be small compared to $\lambda_\beta$ and $\lambda_j$ for $j = 1, 2, \ldots, q$. This argument is also supported by our empirical studies.

With different $\boldsymbol{\lambda}$, the 2-stage method can identify different stocks. In terms of model selection, it is suggested in [35] that the in-sample MSE underestimates the MSE calculated from out-of-sample data, but the cross-validation error is a good substitution. Hence, ideally the tuning of $\boldsymbol{\lambda}$ should be carried out by minimizing the cross validation error which

is given later in (4.32). However, when there is more than one tuning parameter in (4.30), it is too computationally expensive to determine their values by the the cross-validation method. In order to simplify this problem and make it implementable, we propose to let these tuning parameters sum up to 1, and let $\lambda_\beta$ and the $\lambda_j$'s be proportional to the sample means of the corresponding factors with the same rate, *i.e.*,

$$\lambda_\beta = (1-\lambda_\alpha)\frac{\sqrt{\hat{E}[R^2]}}{\sqrt{\hat{E}[R^2]}+\sum_j\sqrt{\hat{E}[F_j^2]}}, \text{ and } \lambda_j = (1-\lambda_\alpha)\frac{\sqrt{\hat{E}[F_j^2]}}{\sqrt{\hat{E}[R^2]}+\sum_j\sqrt{\hat{E}[F_j^2]}} \text{ for all } j.$$

Hence, given observed samples of $R$ and the $F_j$'s, all these tuning parameters are determined by the value of $\lambda_\alpha$. This tuning procedure is ad-hoc, but it has the advantage of reducing the number of tuning parameters in (4.30) to one, and a cross-validation for determining a single tuning parameter is easy to carry out. It also leads to acceptable in-sample and out-of-sample MSEs as confirmed by results from the real data applications in Section 4.5.

The tuning parameter $\lambda_\alpha$ is determined by an $M$-fold cross validation. More specifically, we randomly partition $\{(R_t, F_{1t}, F_{2t}, \dots)\}_{t=1}^n$ into $M$ equal parts, $D_1, \dots, D_M$. For $m = 1, \dots, M$, choose $D_m$ as the validation data, and use the remaining $M-1$ parts as the training data. Given a value of $\lambda_\alpha$, applying the 2-stage method to the training data leads to a tracking portfolio. When the constructed tracking portfolio is applied to the validation data $D_m$, we can compute $CV_m(\lambda_\alpha)$, which is given by

$$\sum_{t:(R_t,r_{t,1},\dots,r_{t,d})\in D_m}\left(R_t - \sum_{i=1}^{k'} w_i r_{t,i}\right)^2.$$

Define the $M$-fold cross validation error as

$$CV(\lambda_\alpha) = \frac{1}{n}\sum_m CV_m(\lambda_\alpha), \tag{4.32}$$

and take $\arg\min_{\lambda_\alpha} CV(\lambda_\alpha)$ as the tuning parameter. In this chapter, we do not search for $\lambda_\alpha$ by directly minimizing the cross-validation error. Instead, we always consider a set of

candidate values of $\lambda_\alpha$ and choose the value that gives the smallest cross-validation error.

## 4.5 Application

### 4.5.1 Data

In this section, the 2-stage method is applied to track two capitalization-weighted stock-market indices, the Russell 2000 and the Russell 3000. The Russell 3000 index consists of 3,000 publicly held US companies, which represent 98% of the investable US equity market. The Russell 2000 index is comprised of 2,000 small-cap stocks that are the bottom 2,000 components of the Russell 3000. Weekly levels of the Russell 2000 and Russell 3000, as well as their components' weekly prices, are downloaded from Bloomberg. The weekly data cover the period from October 2nd, 2009 to April 24th, 2015, and include 291 observations for each index level and the stock price of each component. Corresponding weekly data of the Fama-French factors SMB and HML are obtained from the Fama-French Data Library[3].

Since there are missing data for many index components, we construct and track synthetic Russell 2000 and Russell 3000 indices. In doing that, stocks (listed as an index components on April 24th, 2015) with any missing weekly data are deleted, so that the synthetic Russell 2000 consists of 1,306 stocks and the synthetic Russell 3000 consists of 2,137 stocks. Synthetic capitalization-weighted stock-market indices are constructed according to

$$I = \frac{1}{D} \sum_{i=1}^{d} a_i S_i,$$

where $I$ is the index level, $S_i$ is the stock price for stock $i$, $a_i$ is the number of outstanding shares for stock $i$, and $D$ is the index divisor. Over the whole discussed period, the number of outstanding shares for each stock remains the same as that on April 24th, 2015, and the index divisor of the Russell 2000 (3000) is the number at which the synthetic index level on April 24th, 2015 equals to the real Russell 2000 (3000) level observed on that day.

[3] http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html

Weekly discrete-time returns of stocks and synthetic indices are calculated and partitioned into the in-sample data, including 145 weekly returns, and out-of-sample data, including 145 weekly returns. In the rest of this section, we always refer to the synthetic Russell 2000 (3000) index as the Russell 2000 (3000) index.

### 4.5.2 Results

In these applications, each of the Russell 2000 and Russell 3000 is tracked by at most 50, 100, and 150 index components respectively, using our 2-stage method. Sharpe's single-index model (SIM), the characteristic line of the three-moment CAPM (3CAPM), and the Fama-French 3 factor (FF3F) model are used to describe asset returns. OLS estimators are used to estimate $\alpha_i$, $\beta_i$, $\gamma_{ij}$, and $\sigma^2_{\varepsilon_i}$ for all $i, j$ in the general factor model (4.13). The mixed-integer linear program (4.30) is solved by the Matlab built-in function "intlinprog". Settings of using this function are as follows: the branch-and-bound algorithm is used, the heuristic to find a feasible point is set to "rss", and the termination criteria are TolCapRel=1e-4 and MaxTime=1500. With those termination criteria, optimal solutions to (4.30) are found for all cases. The quadratic program (4.31) is solved by the Matlab built-in function "quadprog" with default settings.

In terms of tuning in (4.30), the tuning parameter $\lambda_\alpha$ is determined by a 5-fold cross validation. Based on some pre-analysis (see Section 4.4.3), we find that $\lambda_\alpha$ should be a small number for both the Russell 2000 and Russell 3000. A candidate set of $\lambda_\alpha$, that is $\{10^{-5}, 5\times10^{-5}, 10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}\}$, is proposed, and the tuning parameter $\lambda_\alpha$ is the value which gives the smallest 5-fold cross-validation error as defined in (4.32). Figure 4.1 shows ranks (by magnitude) of cross-validation errors, out-of-sample empirical MSEs (4.5) and out-of-sample empirical structured MSEs (4.17), at different values of $\lambda_\alpha$, in the case that Russell 3000 is tracked by at most 50 stocks with the FF3F model. Though focusing on tuning $\lambda_\alpha$ is ad-hoc, Figure 4.1 suggests that this tuning method can lead to small out-of-sample MSEs. Also, out-of-sample MSEs have a similar trend with the cross-validation error. Similar trends among cross-validation errors and empirical MSEs can also be observed while tracking the Russell 2000 and Russell 3000 by other factor models.

Table 4.1 shows the in-sample and out-of-sample empirical structured MSEs (4.17) of tracking the Russell 2000 and Russell 3000 indices with at most 50, 100, and 150 stocks. In all applications, the number of selected stocks $k'$ always equals to the upper bound $k$. This is because, in each generic period, the index return is a linear combination of all components. Moreover, since the transaction cost is not considered, the more stocks included in the tracking portfolio, the smaller the MSE should be. Table 4.1 suggests that the tracking procedure with FF3F leads to the best in-sample and out-of-sample structured MSE, and the tracking procedure with 3CAPM improves that with SIM.

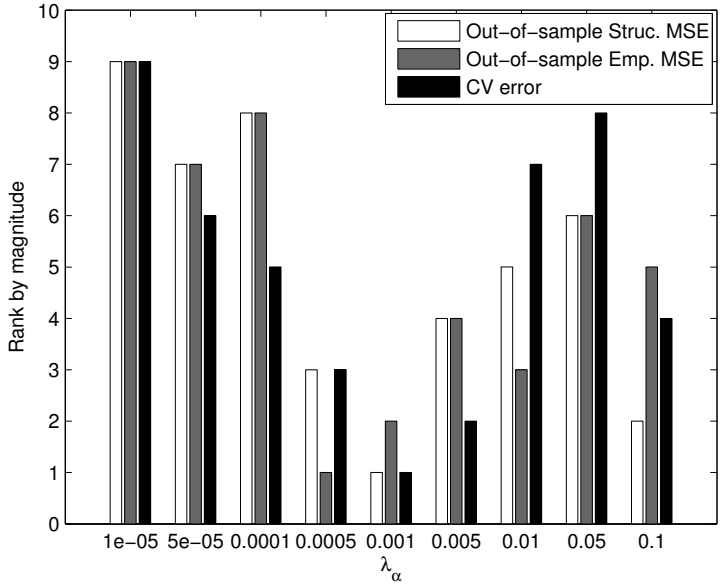



Figure 4.1: Ranks, by magnitude, of the cross-validation (CV) error, out-of-sample structured (Struc.) MSE, and out-of-sample empirical (Emp.) MSE at different values of $\lambda_\alpha$.

In Table 4.2, we report the cross-validation error, and in-sample and out-of-sample empirical MSEs (4.5) of each tracking portfolio in Table 4.1. By investigating out-of-sample empirical MSE, we can compare the 2-stage method with heuristic methods which directly minimize the in-sample empirical MSE. In doing so, the method in [66] is used as a benchmark. More specifically, the benchmark uses the GCD criterion to select stocks, and these stocks construct a tracking portfolio that minimizes the empirical in-sample tracking error. This GCD method usually leads to comparable out-of-sample empirical MSEs with

| Index | Number of stocks | Factor Model | In-sample Struc. MSE | Out-of-sample Struc. MSE |
|---|---|---|---|---|
| Russell 2000 | 50 | SIM | 3.8373E-05 | 3.7015E-05 |
| | | 3CAPM | 3.7635E-05 | 3.2631E-05 |
| | | FF3F | **3.1372E-05** | **3.0178E-05** |
| | 100 | SIM | 1.8051E-05 | 1.7695E-05 |
| | | 3CAPM | 1.7480E-05 | 1.6909E-05 |
| | | FF3F | **1.7101E-05** | **1.6558E-05** |
| | 150 | SIM | 1.4929E-05 | 1.4818E-05 |
| | | 3CAPM | 1.4643E-05 | 1.4501E-05 |
| | | FF3F | **1.3391E-05** | **1.2845E-05** |
| Rusell 3000 | 50 | SIM | 1.9520E-05 | 1.9515E-05 |
| | | 3CAPM | 1.5869E-05 | 1.5606E-05 |
| | | FF3F | **1.5195E-05** | **1.4980E-05** |
| | 100 | SIM | 9.5698E-06 | 9.3923E-06 |
| | | 3CAPM | 8.1331E-06 | 7.9520E-06 |
| | | FF3F | **6.8338E-06** | **6.7810E-06** |
| | 150 | SIM | 5.5745E-06 | 5.5592E-06 |
| | | 3CAPM | 5.4059E-06 | 5.3568E-06 |
| | | FF3F | **4.3777E-06** | **4.3622E-06** |

Table 4.1: In-sample structured (Struc.) MSEs and out-of sample structured MSEs of tracking the Russell 2000 and Russell 3000 by at most 50, 100, and 150 stocks.

those of heuristic methods on index tracking (see [66]).

In order to compare how significant the difference between out-of-sample empirical MSEs is, we carry out an asymptotic Z-test. Let $g_s = (R - \mathbf{w}_s'\mathbf{r})^2$ be the squared tracking discrepancy evaluated at $\mathbf{w}_s$, for $s = 1, 2$. Because of the assumption that $(R_t, \mathbf{r}_t')$ for $t = 1, \ldots, n$ are i.i.d samples, squared tracking discrepancies, $g_{s,t} = (R_t - \mathbf{w}_s'\mathbf{r}_t)^2$ for $s = 1, 2$, at different times are i.i.d. Let $e = g_1 - g_2$, $\mu_e = E[e]$, and $\sigma_e^2 = \mathrm{Var}(e)$. Then due to the central limit theorem we have

$$\frac{\bar{e} - \mu}{\sqrt{\sigma_e^2/n}} \to N(0, 1), \text{ as } n \to \infty,$$

where $\bar{e}$ is the sample mean of $\{e_t\}_{t=1}^n$. In our applications, the out-of-sample size is

145 and $\sigma_e^2$ is replace by its unbiased sample estimate. For each 2-stage method with either SIM, 3CAPM, or FF3F factor model, let $e$ be its squared tracking discrepancy less the squared tracking discrepancy given by the GCD method. A Z-test with the null hypothesis $H_0 : E[e] = 0$ and alternative hypothesis $H_1 : E[e] < 0$ is carried out to evaluate how significant the difference between out-of-sample MSEs is, and the $p$-values are given in the 6th column in Table 4.2.

According to the results in Table 4.2, the GCD method leads to small in-sample empirical MSEs, but out-of-sample empirical MSEs are relatively large. Focusing on the out-of-sample performance, the 2-stage method can significantly reduce the empirical MSE (in terms of reported p-value), and the method with FF3F usually yields the smallest out-of-sample empirical MSE. The last column shows the running time of each method when it is carried out on a PC with Intel Core i5-3210M CPU at 2.5GHz and 6.00GB memory. It provides evidence that the 2-stage method is much more computationally efficient.

## 4.6 Discussion

In this chapter, factor models are introduced to construct tracking portfolios. With the assumption of factor models, the tracking error can be partitioned into two parts. One only depends on economic factors, and the other depends on idiosyncratic risks. A 2-stage method is proposed to construct tracking portfolios. The first stage identifies stocks that are capable of reducing factors' impact on the tracking error. The second stage determines stock weights by exactly minimizing the tracking error with the selected stocks.

If the number of index components is large, compared with existing heuristic methods used in the index tracking literature, the 2-stage method significantly reduces the out-of-sample tracking error, and it is more computationally efficient.

| Index | Number of stocks | Method | CV error | In-sample Emp. MSE | Out-of-sample Emp. MSE | p-value | Time |
|---|---|---|---|---|---|---|---|
| Russell 2000 | 50 | GCD | - | **5.41E-05** | 7.67E-05 | - | 1.21*h.* |
| | | SIM | 7.80E-05 | 9.37E-05 | 8.39E-05 | 0.74 | 25.73 *s.* |
| | | 3CAPM | **5.68E-05** | 7.57E-05 | **5.82E-05** | 0.02 | 31.65 *s.* |
| | | FF3F | 6.49E-05 | 5.85E-05 | 6.65E-05 | 0.13 | 51.19 *s.* |
| | 100 | GCD | - | **4.71E-05** | 6.31E-05 | - | 3.27 *h.* |
| | | SIM | 6.98E-05 | 8.46E-05 | 6.77E-05 | 0.71 | 34.07 *s.* |
| | | 3CAPM | 5.10E-05 | 6.06E-05 | 4.93E-05 | 0.03 | 41.87 *s.* |
| | | FF3F | **4.61E-05** | 5.66E-05 | **4.66E-05** | 0.00 | 61.34 *s.* |
| | 150 | GCD | - | **2.34E-05** | 6.98E-05 | - | 5.25 *h.* |
| | | SIM | 6.13E-05 | 6.28E-05 | 4.76E-05 | 0.00 | 40.15 *s.* |
| | | 3CAPM | 4.80E-05 | 5.92E-05 | 4.66E-05 | 0.00 | 105.70 *s.* |
| | | FF3F | **4.51E-05** | 5.12E-05 | **4.36E-05** | 0.00 | 312.38 *s.* |
| Russell 3000 | 50 | GCD | - | 5.31E-05 | 5.32E-05 | - | 1.94 h. |
| | | SIM | 3.29E-05 | 4.09E-05 | 3.37E-05 | 0.01 | 36.53 *s.* |
| | | 3CAPM | 2.87E-05 | 3.73E-05 | 2.85E-05 | 0.00 | 46.35 *s.* |
| | | FF3F | **2.18E-05** | **1.91E-05** | **2.17E-05** | 0.00 | 54.60 *s.* |
| | 100 | GCD | - | 1.91E-05 | 3.17E-05 | - | 4.47 *h.* |
| | | SIM | 2.80E-05 | 3.20E-05 | 2.75E-05 | 0.18 | 61.20 *s.* |
| | | 3CAPM | 2.69E-05 | 2.88E-05 | 2.53E-05 | 0.09 | 64.17 *s.* |
| | | FF3F | **1.31E-05** | **1.30E-05** | **1.43E-05** | 0.00 | 76.48 *s.* |
| | 150 | GCD | - | **8.86E-06** | 4.02E-05 | - | 6.07 *h.* |
| | | SIM | 2.33E-05 | 2.78E-05 | 2.28E-05 | 0.01 | 93.67 *s.* |
| | | 3CAPM | 2.01E-05 | 2.45E-05 | 2.08E-05 | 0.00 | 102.48 *s.* |
| | | FF3F | **1.04E-05** | 1.19E-05 | **1.26E-05** | 0.00 | 316.17 *s.* |

Table 4.2: In-sample empirical MSE (Emp. MSE), out-of sample empirical MSE, and cross-validation (CV) errors of tracking the Russell 2000 and Russell 3000 by at most 50, 100, and 150 stocks. In the last column, *h.* is short for *hours*, and *s.* is short for seconds.

# Chapter 5

# $L_1$-regularization for Index Tracking with Transaction Costs

## 5.1 Introduction

In order to track stock-market indices, a simple strategy is the full replication. At the time of construction, a full replication strictly follows the composition formula of the index (see Section 3.2.1 for examples.). After that, numbers of asset shares in the full replication remain constant until any rebalancing. A mathematical formulation of full replication can be found in Section 5.5.3. Starting from any time *after* construction, the full replication earns exactly the index return. However, there is always a gap between the terminal wealth of a full-replication and the terminal wealth given the initial wealth (*before* construction) earns exactly the index return. This gap is caused by the transaction cost *at* construction, and a high transaction cost leads to a large gap.

Transaction costs primarily consist of explicit costs and implicit costs ([80]). Explicit costs usually refer to broker commissions, which brokers charge for their executions of trading orders. Implicit costs usually refer to the deviation of the transaction price from the unperturbed price, which is observed before the trade. Findings in [80] show that,

in general, stock transaction costs are inversely related to stocks' market capitalizations. For the exchange-listed stocks studied in [80], the ratio of transaction costs to the traded wealth (excluding transaction costs) varies from 0.31% for large-capitalization stocks to 2.35% for small-capitalization stocks..

Some exchange traded funds (ETFs) simply apply the full replication to track large-capitalization stock indices, such as the methodology of SPDR S&P 500 ETF, which is one of the largest ETF benchmarked to the S&P 500 index. In this case, the gap of a full replication is negligible due to small transaction costs of trading large-capitalization stocks ([119]). However, small capitalization stocks are more illiquid, and their high transaction costs usually prevent ETF managers from applying full replication ([71]). When the full-replication is infeasible, in order to mimic an index return fund managers need to determine which index components to invest and how to allocate assets to each selected stock ([71]). This methodology is called the partial replication.

There is a rich literature on partial replication. Many methods such as those in [7], [82], and [105] formulate partial replication problem as a mixed-integer quadratic program. These methods determine stocks and corresponding weights in the tracking portfolio simultaneously. However, mixed-integer quadratic programming is NP-hard (see [105]), so the optimal solution is challenging to obtain efficiently. Heuristic methods are proposed in [7], [82], [105], and [49] to solve this optimization problem, but they usually lead to suboptimal solutions. These methods usually require an upper bound of the number of selected stocks, but different upper bounds drastically impact the tracking performance ([22]). Due to the computational complexity of these heuristic methods, it is challenging to efficiently determine the "right" upper bound which leads to a small tracking error. So these methods cannot address the stock selection problem efficiently, especially when the number of index components is large.

Statistical regularization methods, such as [52], [124], become popular in portfolio management to select stocks. The $L_1$-regularization is one of the most popular regularization methods, due to its computational efficiency and capability to generate sparse structures. Empirical results in [18] show that adding $L_1$-regularization to Markowitz's framework improves Sharpe's ratio. The $L_1$-regularization is applied in [48] to construct minimum-

variance portfolios, and it is found that imposing $L_1$-regularization reduces the estimation error. In terms of applying statistical regularizations to index tracking, [17] briefly discusses constructing a tracking portfolio by minimizing the tracking error plus a $L_q$ penalty, where $q$ is a positive number close to 0. The non-negative Lasso was applied to index tracking in [130], and some properties on non-negative Lasso are discussed. However, in [130] portfolio weights are not required to sum up to 1, so that the method provides little guide on constructing tracking portfolios.

Index tracking with $L_q$-penalty ($0 < q < 1$) is revisited by [49] and [131]. With a no-short-selling constraint, the $L_1$-norm of stock weights is always 1, so that authors of [49] advocate the $L_q$-penalty method to promote sparsity. Because the $L_q$-penalized tracking error is a non-convex function, authors of [49] introduce a hybrid heuristic method to minimize the objective function. The optimal number of selected stocks in a tracking portfolio is also discussed in [49]. The authors of [131] focus on developing algorithms to solve for tracking strategies with the $L_{1/2}$-regularization.

Aiming at reducing the gap between the tracking portfolio terminal wealth and the terminal wealth given the initial wealth (before construction) earns exactly the index return, we formulate the index tracking problem into an optimization problem. In general, this is a multi-period tracking problem. In this chapter, the multi-period tracking problem is tackled by repeatedly solving one-period tracking problems with the $L_1$-regularization on asset weights as a sub-optimal solution. Our formulation takes into account transaction costs and other practical issues, such as the budget constraint, no-short-selling stock constraint, *etc.*. Besides index components, in the tracking portfolio we also include a money market account which earns an interest rate. Including one more asset is expected to improve the tracking performance, and we allow borrowing money from the money market to invest into stocks. Another motivation of including the money market account is to vitalize the $L_1$-regularization on stock weights with the no-short-selling stock constraint. The $L_1$-norm on stock weights can be adjusted by changing the money market account weight. Moreover, the $L_1$-norm on asset weights is more flexible and it is not the constant 1 any more, when a negative position in the money market account is allowed.

Since the true joint distribution of financial returns is usually unknown, in this chapter

the index tracking problem is solved by minimizing the empirical tracking error. This analysis is carried out in a high-dimensional statistical setting. More specifically, the number of parameters $d$ is larger than the sample size $n$ and also grows with $n$. In the stock market, there are usually too many stocks to gather enough historical data for a classical statistical analysis, which requires $d < n$. For example, the U.S. stock-market index Russell 3,000 consists of 3,000 components. In order to gather more than 3,000 weekly data, we need 58 years. However, 58 years ago, only a few of these 3,000 stocks existed, so that it is impossible to gather more than 3,000 weekly data for all these 3,000 stocks. For such a big $d$ small $n$ dataset, it is more suitable to apply the high-dimensional statistical inference setting where $d$ is viewed as a function of $n$, and the inference is based on asymptotic results as $n \to \infty$.

A tracking strategy obtained by minimizing the empirical tracking error is not necessarily relevant to a tracking strategy that minimizes the true tracking error. However, under some assumptions, our $L_1$-regularization tracking strategy obtained by minimizing the empirical tracking error is persistent. The definition of *persistent* is introduced in [63] and can be found in Section 5.3 of this chapter. The persistent property guarantees that our tracking strategy obtained by minimizing the empirical tracking error leads to a tracking error which asymptotically converges to the minimum true tracking error. The persistence is an asymptotic property. In oder to verify the performance of our $L_1$-regularization tracking strategy with finite samples, we carry out a simulation study. It shows that the tracking error of our strategy gets more stable and approaches to the minimized true tracking error as $n$ gets larger. When it is applied to real financial data in Section 5.5, Our tracking strategy outperforms other methods from the relevant literature in terms of tracking accuracy and computational efficiency.

The organization of this chapter is as follows. Section 5.2 formulates the index tracking problem with transaction costs and basic practical constraints. In Section 5.3 the $L_1$-regularization and the persistence property of our one-step tracking strategy is discussed. Simulation studies in Section 5.4 verify the performance of our one-step $L_1$-regularization tracking method with finite samples. In Section 5.5, applications with financial data provide evidence that our $L_1$-regularization tracking method has better tracking performance than other methods, such as the $L_q$-penalty method and full-replication. Section 5.6 con-

cludes this chapter.

## 5.2 Formulations of Index Tracking with Transaction Costs

In this section, a tracking strategy is introduced to reduce the gap between the tracking portfolio terminal wealth and the terminal wealth accumulated by the index return from the same initial investment as the tracking portfolio.

Following [80], we assume that the transaction cost is proportional to the traded stock wealth throughout this chapter. The proportional rate is denoted by $\theta(\geq 0)$ which is the same for both buying and selling stocks. We further assume that there is no transaction cost for trading over money market accounts.

### 5.2.1 Some Notation

The following notation is necessary to proceed to our formulation of the index tracking problem.

- Denote by $d$ the number of index components, and the money market account is labelled as the 0-th asset in the tracking portfolio.
- For $t = 1, 2, 3, \ldots$, the index return from time $t-1$ to $t$, $R_t$, is given by

$$R_t = \frac{I_t - I_{t-1}}{I_{t-1}},$$

  where $I_t$ is the index level at time $t$.

- Similarly, the return of the $i$-th asset from time $t-1$ to time $t$, $r_{t,i}$, is given by

$$r_{t,i} = \frac{S_{t,i} - S_{t-1,i}}{S_{t-1,i}},$$

  where $S_{t,i}$ is the $i$-th asset price at time $t$ for $i = 0, 1, \ldots, d$.

- For $t = 1, 2, 3, \ldots$, the time before rebalancing at time $t-1$ is denoted by $(t-1)^-$, and the time after rebalancing is denoted by $(t-1)^+$.

- For $i = 0, 1, \ldots, d$, denote by $x_i^{(t-1)^-}$ (or $x_i^{(t-1)^+}$) the dollar value of the $i$-th asset in the tracking portfolio before (or after) rebalancing at time $t-1$.

- At time $(t-1)^-$, denote the tracking portfolio wealth by $W_{(t-1)^-}$, so that

$$W_{(t-1)^-} = \sum_{i=0}^{d} x_i^{(t-1)^-}.$$

- Further, let

$$\mathbf{x}^{(t-1)^-} = \left(x_0^{(t-1)^-}, x_1^{(t-1)^-}, \ldots, x_d^{(t-1)^-}\right)', \text{ and } \mathbf{x}^{(t-1)^+} = \left(x_0^{(t-1)^+}, x_1^{(t-1)^+}, \ldots, x_d^{(t-1)^+}\right)'.$$

- At time $(t)^-$ for $t = 1, 2, \ldots, T$, given $\mathbf{x}^{(t-1)^+}$, the wealth of the tracking portfolio can be written as

$$W_{(t)^-} = \sum_{i=0}^{d} (1 + r_{t,i}) x_i^{(t-1)^+}. \tag{5.1}$$

- Suppose the investment horizon is $T$, which is a positive integer. Let $W^I_{(T)^-}$ be the terminal wealth if a portfolio starts from an initial investment of $W_{(0)^-}$ and earns the index return over the period $[0, T]$, so that

$$W^I_{(T)^-} = \left(W_{(0)^-}\right) \prod_{s=1}^{T} (1 + R_s).$$

After time $T$, the tracking portfolio can be sold and converted to cash, kept in the

trading account, or merged to other portfolios. In order to avoid discussing transaction costs (if any) charged *at* time $T$, in this chapter we only focus on the terminal wealth up to time $(T)^-$.

### 5.2.2 Formulations of the Index Tracking Problem

Aiming at reducing the gap between the tracking portfolio terminal wealth and the terminal wealth accumulated by the index return from the same initial investment as the tracking portfolio, in general the index tracking problem can be formulated as

$$\min_{\mathbf{x}} E\left[\left(W^I_{(T)^-} - W_{(T)^-}\right)^2 \middle| \mathbf{x}^{(0)^-}\right] \tag{5.2}$$

$$s.t. \quad \mathbf{x} \in \mathcal{U}, \tag{5.3}$$

where $\mathbf{x} = (\mathbf{x}^{(0)^+}, \mathbf{x}^{(1)^+}, \ldots, \mathbf{x}^{(T-1)^+})$, and $\mathcal{U}$ is a certain feasible set due to some practical considerations. The objective function (5.2) adopts the square-loss $\left(W^I_{(T)^-} - W_{(T)^-}\right)^2$, since the square-loss penalizes large deviations between $W^I_{(T)^-}$ and $W_{(T)^-}$. Also, the square loss is a popular loss-function in practice ([53]).

However, it is challenging to solve such a multi-period index tracking problem (5.2), even when $\mathcal{U}$ in (5.3) is formed by basic practical constraints, such as the budget, no-short-selling, or transaction costs constraints ([29]). Instead, in this chapter the multi-period index tracking problem is tackled by repeatedly solving one-period index tracking problems through time 0 to $T-1$ as a sub-optimal solution. Our formulation of a one-period index tracking problem is presented as follows.

Consider the period from time $t-1$ to $t$ for $t = 1, 2, \ldots, T$. If $W_{(t-1)^-}$ earns the index return $R_t$ throughout this period, at time $(t)^-$ the wealth $W_{(t-1)^-}$ grows to $(1+R_t)W_{(t-1)^-}$. Meanwhile, at time $(t)^-$, the tracking portfolio wealth $W_{(t)^-}$ is given in (5.1). In a one-period index tracking problem, the tracking portfolio is rebalanced at time $(t-1)^+$ to minimize the expectation of the squared discrepancy between $W_{(t)^-}$ in (5.1) and $(1 + R_t)W_{(t-1)^-}$. Hence, the optimal tracking strategy at time $(t-1)^+$, which is determined by solving a one-period index tracking problem, is supposed to be

$$
\begin{aligned}
\tilde{\mathbf{x}}^{(t-1)^+} &= \arg\min_{\mathbf{x}^{(t-1)^+}} E\left[\left((1+R_t)(W_{(t-1)^-}) - W_{(t)^-}\right)^2 \middle| \mathbf{x}^{(t-1)^-}\right] \\
&= \arg\min_{\mathbf{x}^{(t-1)^+}} E\left[\left((1+R_t)(W_{(t-1)^-}) - \sum_{i=0}^{d}(1+r_{t,i})x_i^{(t-1)^+}\right)^2 \middle| \mathbf{x}^{(t-1)^-}\right], \quad (5.4) \\
s.t. \quad & \mathbf{x}^{(t-1)^+} = \left(x_0^{(t-1)^+}, \ldots, x_d^{(t-1)^+}\right)' \in \mathcal{U}_{t-1},
\end{aligned}
$$

where $\mathcal{U}_{t-1}$ is a certain feasible set of $\mathbf{x}^{(t-1)^+}$. In this chapter, four basic practical constraints are considered to define $\mathcal{U}_{t-1}$. They are:

(a) The budget constraint. Once investors have initially invested an amount of money, they are reluctant or sometimes unable to raise addition funds to enlarge their portfolio wealth[1] during subsequent portfolio rebalancings. Hence, we assume that there is no money injected into the portfolio during the rebalancing.

Also, a portfolio is supposed to utilize investments at hand as much as possible. Thus, the ideal budget constraint should be self-financing, which requires that the portfolio wealth after rebalancing plus transaction costs at rebalancing equals the portfolio wealth before rebalancing, *i.e.*

$$
\sum_{i=0}^{d} x_i^{(t-1)^+} + \sum_{i=1}^{d} \theta \left| x_i^{(t-1)^-} - x_i^{(t-1)^+} \right| = W_{(t-1)^-} \ .
$$

However, the above self-financing constraint with transaction costs usually leads to a non-convex feasible set and thus complicates the optimization procedure for a solution. In order to avoid a non-convex feasible set, we follow [88] to relax the self-financing constraint as

$$
\sum_{i=0}^{d} x_i^{(t-1)^+} + \sum_{i=1}^{d} \theta \left| x_i^{(t-1)^-} - x_i^{(t-1)^+} \right| \leq W_{(t-1)^-} \ . \quad (5.5)
$$

The inequality in (5.5) usually results in a convex feasible set, for which optimal solu-

[1]Buying stock via borrowed money does not increase the portfolio wealth, so that this is different from raising money to enlarge the portfolio total amount.

tions can be obtained efficiently. If a solution leaves the constraint (5.5) non-binding, it leads to a withdrawal of some money from the portfolio during the rebalancing. Empirical applications in Section 5.5.3 show that, even though withdrawing money is allowed via an inequality such as (5.5), it accounts for only a small portion (around 0.1% on average) of the total wealth before rebalancing. Since this tiny amount of money contributes little to any returns, the withdrawn money (if any) is not reinvested in following periods.

(b) No-shorting-selling stocks. In the U.S., there is a margin requirement for short selling stocks. The margin for short selling a stock is 50% of the market value of the borrowed stock[2], and this is a significant expense. Due to some restrictions on short-selling stocks, such as the alternative uptick rule by the U.S. Securities and Exchange Commission[3], under certain circumstances it is not easy to short sell stocks. Moreover, losses of short selling stocks are unlimited, which is too risky. Thus, we assume that tracking portfolios do not have short positions in any stocks. Specifically, the no-short-selling constraint is given by

$$x_i^{(t-1)^+} \geq 0, \text{ for } i = 1, \ldots, d. \tag{5.6}$$

However, we allow borrowing money to buy stocks which may result in a negative position in the money market account.

(c) Limit on borrowed money. In our formulation, $x_0^{(t-1)^+}$ is allowed to be negative for $t = 1, 2, \ldots$, which allows investors to borrow money to invest into stocks. However, in practice, the amount of borrowed money is seldom too large compared with the total portfolio wealth $W_{(t-1)^-}$ before rebalancing. Some studies such as [93] and [85] also warn of the disadvantages of a high leverage ratio, defined as the ratio of debt to total asset. Hence, in our formulation it is always required that

$$-cW_{(t-1)^-} \leq x_0^{(t-1)^+}, \tag{5.7}$$

[2] http://www.ecfr.gov/cgi-bin/text-idx?SID=7df35b15d3a9d087dc1fbe017048f723&mc=true&node=se12.3.220_112&rgn=div8.

[3] http://www.sec.gov/news/press/2010/2010-26.htm.

where $c$ is a constant parameter. In order to control the amount of borrowed money, $c$ should be non-negative and the quantity $cW_{(t-1)^-}$ puts an upper bound on the amount of borrowed money. A sufficiently large $c$ could disqualify the constraint (5.7). When $c$ is negative, $|cW_{(t-1)^-}|$ puts a lower bound on the wealth invested in the money market account. According to (5.5) and (5.6), if negative, $c$ cannot be smaller than $-1$.

(d) Limit on the total transaction cost. Under some circumstances, investors prefer to limit their transaction costs spent on constructing or rebalancing their portfolios. In order to meet this requirement, an upper bound is introduced to restrict the total transaction cost. Its mathematical formulation is given by

$$\sum_{i=1}^{d} \theta \left| x_i^{(t-1)^-} - x_i^{(t-1)^+} \right| \leq \gamma W_{(t-1)^-} \tag{5.8}$$

where $\theta$ is the proportional rate of transaction costs, $\gamma(\geq 0)$ controls this upper bound. A sufficiently large $\gamma$ could disqualify the constraint (5.8). The constraint (5.8) controls the upper bound of how much the wealth allocation can be adjusted in each stock during rebalancing, so that the constraint (5.8) is capable of stabilizing the tracking portfolio during rebalancing.

In summary, a heuristic method is introduced to solve the multi-period tracking portfolio problem (5.2) subject to the budget and no-short-selling constraints, as well as limits on borrowed money and transaction costs. In this heuristic method, the tracking portfolio is rebalanced at each time $(t-1)$ for $t = 1, \ldots, T$, in a way that the tracking strategy after each rebalancing $\tilde{\mathbf{x}}^{(t-1)^+}$ is derived by solving a one-period index tracking problem. More specifically, the vector $\tilde{\mathbf{x}}^{(t-1)^+}$ is given by (5.4) subject to (5.5)-(5.8) for $t = 1, \ldots, T$.

For convenience, we rewrite the one-period index tracking problem (5.4) subject to (5.5)-(5.8) in the following way. Let

$$Y_t = (1 + R_t), \;\; X_{t,i} = 1 + r_{t,i} \text{ for } i = 0, 1, \ldots, d, \tag{5.9}$$

and $\mathbf{X}_t = (X_{t,0}, X_{t,1}, \ldots, X_{t,d})'$ for $t = 1, \ldots, T$. Assume that $\left(W_{(t-1)^-}\right) > 0$, and let

$$\boldsymbol{\beta}^{(t-1)^-} = \frac{\mathbf{x}^{(t-1)^-}}{W_{(t-1)^-}}, \quad \boldsymbol{\beta}^{(t-1)^+} = \frac{\mathbf{x}^{(t-1)^+}}{W_{(t-1)^-}}.$$

Then the one-period index tracking formulation (5.4) subject to (5.5)-(5.8), can be simplified to

$$\tilde{\boldsymbol{\beta}}^{(t-1)^+} = \arg\min_{\boldsymbol{\beta}^{(t-1)^+}} E\left[\left(Y_t - \left(\boldsymbol{\beta}^{(t-1)^+}\right)'\mathbf{X}_t\right)^2 \middle| \boldsymbol{\beta}^{(t-1)^-}\right] \tag{5.10}$$

$$s.t. \quad \sum_{i=0}^{d} \beta_i^{(t-1)^+} + \sum_{i=1}^{d} \theta \left|\beta_i^{(t-1)^-} - \beta_i^{(t-1)^+}\right| \leq 1\ , \tag{5.11}$$

$$\beta_i^{(t-1)^+} \geq 0, \text{ for } i = 1, \ldots, d, \tag{5.12}$$

$$-c \leq \beta_0^{(t-1)^+}, \tag{5.13}$$

$$\sum_{i=1}^{d} \theta \left|\beta_i^{(t-1)^-} - \beta_i^{(t-1)^+}\right| \leq \gamma, \tag{5.14}$$

for $t = 1, \ldots, T$.

In this chapter, at the time of construction, *i.e.* time 0, the tracking portfolio is always assumed to be constructed from a pure cash position. That is $x_0^{(0)^-} = W_{(0)^-}$ and $x_i^{(0)^-} = 0$ for $i = 1, \ldots, d$. Correspondingly, $\beta_0^{(0)^-} = 1$ and $\beta_i^{(0)^-} = 0$ for $i = 1, \ldots, d$.

For $t = 1, \ldots, T-1$, suppose that at time $(t-1)^+$ a tracking strategy $\tilde{\boldsymbol{\beta}}^{(t-1)^+}$ (or equivalently $\tilde{\mathbf{x}}^{(t-1)^+}$) has been determined by (5.10)-(5.14). In the next period, the same procedure in (5.10)-(5.14) is repeated for the rebalancing at time $t$. Note that before rebalancing at time $t$ it is necessary to figure out $\beta_i^{(t)^-}$ (or equivalently the dollar amount of the $i$th asset $x_i^{(t)^-}$) for $i = 0, 1, \ldots, d$. In fact, $x_i^{(t)^-}$ is given by

$$x_i^{(t)^-} = (1 + r_{t,i})\tilde{x}_i^{(t-1)^+} = (1 + r_{t,i})\tilde{\beta}_i^{(t-1)^+} W_{(t-1)^-}, \text{ for } i = 0, 1, \ldots, d.$$

Thus, the portfolio value $W_{(t)^-}$ at time $t$ before rebalancing is given by

$$W_{(t)^-} = \sum_{i=0}^{d} x_i^{(t)^-} = \left(W_{(t-1)^-}\right) \sum_{i=0}^{d} (1 + r_{t,i}) \tilde{\beta}_i^{(t-1)^+}, \tag{5.15}$$

and then

$$\beta_i^{(t)^-} = \frac{x_i^{(t)^-}}{W_{(t)^-}} = \frac{(1 + r_{t,i}) \tilde{\beta}_i^{(t-1)^+}}{\sum_{i=0}^{d} (1 + r_{t,i}) \tilde{\beta}_i^{(t-1)^+}}, \text{ for } i = 0, 1, \ldots, d. \tag{5.16}$$

Repeating the one-period method (5.10)-(5.14) may not be the optimal strategy to (5.2) subject to the budget, no-short-selling stocks, borrowed money limit, and transaction costs limit constraints. However, it provides one computationally feasible solution to the multi-period index tracking problem. Some empirical implementations of repeating the one-period method in Section 5.5.3 show evidence that this strategy works better than the full-replication under some circumstance.

## 5.3 The $L_1$-regularization and Persistence

In this chapter, we assume that the random vectors $(R_t, r_{t,0}, r_{t,1}, \ldots, r_{t,d})'$ are independent and identically distributed (i.i.d.) at different times $t$ for $t = 1, 2, \ldots$. Although financial data might demonstrate serial dependence, we use such an i.i.d. assumption as a benchmark. Due to the complexity of the model, it is challenging to develop any meaningful theory without such an i.i.d assumption. Correspondingly, the random vector $(Y_t, X_{t,0}, X_{t,1}, \ldots, X_{t,d})'$, of which elements are defined in (5.9), at different times $t$ are i.i.d. samples from a random vector $(Y, X_0, X_1, \ldots, X_d)'$. Under this assumption, properties of the one period index tracking problem (5.10)-(5.14) are homogeneous for $t = 1, 2, \ldots$, except for different parameters $\boldsymbol{\beta}^{(t-1)^-}$. Hence, in this section, we only focus on properties of rebalancing at time 0. Properties discussed in this section still hold for the rebalancing at time $t = 1, 2, \ldots, T-1$.

If the true joint distribution of the vector $(Y, \mathbf{X}')'$ with $\mathbf{X} = (X_0, X_1, \ldots, X_d)'$ is given, at time 0 the problem (5.10)-(5.14) is a quadratic programming problem. However, the

true joint distribution is usually unknown, so that it is impossible to solve (5.10)-(5.14) directly. In this chapter, we minimize empirical tracking errors. Specifically, at time 0 the tracking strategy after rebalancing $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ is given by

$$\begin{aligned} \hat{\boldsymbol{\beta}}_n^{(0)^+} &= \arg \min_{\boldsymbol{\beta}=(\beta_0,\ldots,\beta_d)'} \hat{E}\left[\left(Y-\boldsymbol{\beta}'\mathbf{X}\right)^2 \middle| \boldsymbol{\beta}^{(0)^-}\right] \\ s.t. \quad & (5.11),\ (5.12),\ (5.13),\ (5.14), \end{aligned} \tag{5.17}$$

where

$$\hat{E}\left[\left(Y-\boldsymbol{\beta}'\mathbf{X}\right)^2 \middle| \boldsymbol{\beta}^{(0)^-}\right] = \frac{1}{n}\sum_{s=1}^{n}(Y_s-\boldsymbol{\beta}'\mathbf{X}_s)^2,$$

and $\mathbf{X}_s = (X_{s,0}, X_{s,1}, \ldots, X_{s,d})'$ for $s = 1, \ldots, n$, and $n$ is the number of available samples of $(Y, \mathbf{X}')'$ at time 0. For simplicity, in (5.17) and the following parts of this chapter, we use $\boldsymbol{\beta}$ to replace the original decision variables $\boldsymbol{\beta}^{(0)^+}$.

As discussed in Section 5.1, in order to derive some statistical properties which are suitable to high-dimensional data where $d > n$, we posit the problem in a setting where the number of index components $d$ grows as the sample size $n$. More specifically, we let $d = d(n) = O(n^\alpha)$ with $\alpha > 1$. This order of $d(n)$ is inherited from [63] to prove the following Theorem 5.1.

Denote the true distribution of $(Y, X_0, X_1, \ldots, X_{d(n)})'$ by $F_n$, and its empirical distribution by $\hat{F}_n$. Let

$$L_{F_n}(\boldsymbol{\beta}) = E\left[\left(Y-\boldsymbol{\beta}'\mathbf{X}\right)^2 \middle| \boldsymbol{\beta}^{(0)^-}\right] \text{ and } L_{\hat{F}_n}(\boldsymbol{\beta}) = \hat{E}\left[\left(Y-\boldsymbol{\beta}'\mathbf{X}\right)^2 \middle| \boldsymbol{\beta}^{(0)^-}\right].$$

Suppose at time 0, a feasible set $\mathcal{U}_0(n)$ is defined by (5.11)-(5.14). More specifically, $\mathcal{U}_0(n)$

is given by

$$\mathcal{U}_0(n) = \Bigg\{ \boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_{d(n)})' : \quad \sum_{i=0}^{d(n)} \beta_i + \sum_{i=1}^{d(n)} \theta \Big| \beta_i^{(0)^-} - \beta_i \Big| \leq 1, \\ \beta_i \geq 0, \text{ for } i = 1, \ldots, d(n), \\ -c \leq \beta_0, \\ \sum_{i=1}^{d(n)} \theta \Big| \beta_i^{(0)^-} - \beta_i \Big| \leq \gamma \Bigg\}. \tag{5.18}$$

Let

$$\tilde{\boldsymbol{\beta}}_n^{(0)^+} = \arg \min_{\boldsymbol{\beta} \in \mathcal{U}_0(n)} L_{F_n}(\boldsymbol{\beta}), \text{ and } \hat{\boldsymbol{\beta}}_n^{(0)^+} = \arg \min_{\boldsymbol{\beta} \in \mathcal{U}_0(n)} L_{\hat{F}_n}(\boldsymbol{\beta}).$$

In general, $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ has no obvious relationship with $\tilde{\boldsymbol{\beta}}_n^{(0)^+}$, since they are optimal solutions to minimize different objective functions. However, it is investigated in the following that as long as $\mathcal{U}_0(n)$ satisfies some conditions (which are discussed later), it leads to

$$L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0^+)}) - L_{F_n}(\tilde{\boldsymbol{\beta}}_n^{(0^+)}) \xrightarrow{p} 0 \text{ as } n \to \infty, \tag{5.19}$$

where $\xrightarrow{p}$ stands for convergence in probability. If the relationship in (5.19) holds, then $\hat{\boldsymbol{\beta}}_n^{(0^+)}$ is called *persistent* with respect to (w.r.t.) $\mathcal{U}_0(n)$ in [63]. Suppose that the index tracking strategy $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ is persistent, then it leads to an actual risk $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$ which is close to the minimum true risk (or true tracking error) $L_{F_n}(\tilde{\boldsymbol{\beta}}^{(0^+)})$ for sufficiently large $n$.

In the following, we show that as long as $c$ in (5.7) satisfies $|c| = |c_n| = o\big(\big(\frac{n}{\log(n)}\big)^{1/4}\big)$ then $\mathcal{U}_0(n)$ defined in (5.18) leads to the persistence of $\hat{\boldsymbol{\beta}}_n^{(0)^+}$. This discussion relies on the following theorem in [63].

**Theorem 5.1** (Greenshtein and Ritov (2004))**.** Assume that

(a) $d = d(n) = O(n^\alpha)$ where $\alpha > 1$,

(b) $(Y_s, X_{s,0}, X_{s,1}, \ldots, X_{s,d(n)})'$ for $s = 1, \ldots, n$ are independent and identically distributed samples of the random vector $(Y, X_0, X_1, \ldots, X_{d(n)})'$ which follows a joint distribution $F_n$.

(c) $E\left[\left(\max_{0\le i,j\le d(n)}|X_iX_j-\sigma_{ij}|\right)^2\right]\le M<\infty$ and $E\left[\left(\max_{0\le i\le d(n)}|YX_i-\sigma_i|\right)^2\right]\le M<\infty$, where $\sigma_{ij}=E[X_iX_j]$, $\sigma_i=E[YX_i]$, and $M$ is a constant.

Let

$$\tilde{\boldsymbol{\beta}}_n=\arg\min_{\boldsymbol{\beta}\in B^{(n)}}L_{F_n}(\boldsymbol{\beta}),\ \text{and}\ \hat{\boldsymbol{\beta}}_n=\arg\min_{\boldsymbol{\beta}\in B^{(n)}}L_{\hat{F}_n}(\boldsymbol{\beta}),$$

where $B^{(n)}$ is a certain feasible set, then

(1) $\forall\delta>0$,

$$P_{F_n}\left\{L_{F_n}(\hat{\boldsymbol{\beta}}_n)-L_{F_n}(\tilde{\boldsymbol{\beta}}_n)\ge\delta\right\}\le\frac{C}{\delta}\sup_{\boldsymbol{\beta}\in B^{(n)}}||\boldsymbol{\beta}||_1^2\sqrt{\frac{\log(n)}{n}},\tag{5.20}$$

where $C$ is a positive constant, and

(2) for any sequence

$$B^{(n)}_{b(n)}=\left\{\boldsymbol{\beta}:||\boldsymbol{\beta}||_1\le b_n=o\Big(\big(\frac{n}{\log(n)}\big)^{1/4}\Big)\right\},\tag{5.21}$$

where $||\boldsymbol{\beta}||_1$ is the $L_1$-norm of $\boldsymbol{\beta}$, *i.e.* $||\boldsymbol{\beta}||_1=\sum_{i=0}^{d(n)}|\beta_i|$, there exists a persistent sequence indexed by $n$. One persistent sequence is given by

$$\hat{\boldsymbol{\beta}}_n=\arg\min_{\boldsymbol{\beta}\in B^n_{b(n)}}L_{\hat{F}_n}(\boldsymbol{\beta}).$$

*Proof.* See [63]. □

Theorem 5.1 implies that, under assumptions (a)-(c) in Theorem 5.1, if the feasible set $\mathcal{U}_0(n)$ is $L_1$-regulated at the order of $o\Big(\big(\frac{n}{\log(n)}\big)^{1/4}\Big)$, in other words $\mathcal{U}_0(n)\subset B^{(n)}_{b(n)}$ which is defined in (5.21), then $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ is persistent w.r.t. $\mathcal{U}_0(n)$. Actually, by its definition (5.18), $\mathcal{U}_0(n)$ is $L_1$-regulated at the order of $o\big(\big(\frac{n}{\log(n)}\big)^{1/4}\big)$, as long as the quantity $c$ in (5.13) satisfies $|c|=|c_n|=o\big(\big(\frac{n}{\log(n)}\big)^{1/4}\big)$. In fact, for any $\boldsymbol{\beta}\in\mathcal{U}_0(n)$,

(i) if $\beta_0\ge 0$, $||\boldsymbol{\beta}||_1$ must be less than or equal to 1 due to (5.11) and (5.12). Hence, $||\boldsymbol{\beta}||_1$ is $L_1$-regulated at the order of $o\big(\big(\frac{n}{\log(n)}\big)^{1/4}\big)$,

(ii) if $\beta_0 < 0$, then (5.13) implies $c > 0$, and further (5.11) and (5.12) lead to

$$\begin{aligned} ||\boldsymbol{\beta}||_1 &= \sum_{i=0}^{d(n)} |\beta_i| = |\beta_0| + \sum_{i=1}^{d(n)} \beta_i \\ &\leq |\beta_0| + 1 - \theta \sum_{i=1}^{d(n)} \left|\beta_i^{(0)^-} - \beta_i\right| - \beta_0 \\ &\leq |\beta_0| + 1 + |\beta_0| \\ &\leq 2c + 1. \end{aligned}$$

Given $c = c_n = o\big(\big(\frac{n}{\log(n)}\big)^{1/4}\big)$, we have $||\boldsymbol{\beta}||_1 = o\big(\big(\frac{n}{\log(n)}\big)^{1/4}\big)$.

In summary, as discussed in Section 5.2.2 $c = c_n > -1$, so that as long as $|c| = |c_n| = o\big(\big(\frac{n}{\log(n)}\big)^{1/4}\big)$, $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ is persistent to $\tilde{\boldsymbol{\beta}}_n^{(0)^+}$.

It is worth echoing the significant role that the constraint (5.7) plays in our one-step index tracking method. Besides avoiding a high leverage ratio described in Section 5.2.2, it induces an $L_1$-regularization on the feasible set to make the solution $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ persistent. In practice, it is sufficient to trigger the persistence by controlling the amount of borrowed money at a fixed level or allowing it to increase at a certain rate of the sample size at hand. However, the above theoretical analysis does not point out a way to determine any accurate value of $c$. In applications, we follow [48] to determine $c$ by data-driven methods, such as cross-validation or bootstrapping methods (see Section 5.5).

## 5.4 Simulation Study

The persistence property of $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ in (5.17) holds asymptotically. However, it is impossible to obtain infinite samples in reality. In this section, we carry out simulation studies to investigate how $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0^+)})$ is close to $L_{F_n}(\tilde{\boldsymbol{\beta}}_n^{(0^+)})$ with finite samples.

### 5.4.1 Simulation Methodology

In this simulation study, we first design a joint distribution of a large number of stocks. The number of stocks $d$ is designed to be an increasing function of the number of simulated scenarios $n$. More specifically, we let $d = d(n) = \lfloor n^{\alpha} \rfloor$ with $\alpha > 1$ and $\lfloor \cdot \rfloor$ is the floor function of any real number. Then, these stocks construct an equally weighted stock-market index. An equally weighted stock-market index is used since it is easy to construct and analyze. Given the true joint distribution of stock returns and the index return, an analytical form of $L_{F_n}(\boldsymbol{\beta})$ can be obtained. The performance of our index tracking method with $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ can be investigated by comparing the gap between $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$ and the minimized true risk $L_{F_n}(\tilde{\boldsymbol{\beta}}_n^{(0)^+})$.

For simplicity, from time 0 to time 1 we assume stock returns, as well as the return on the money market account (0-th asset), follow Sharpe's single-index model ([109]) which is given by

$$\mathbf{r} = \mathbf{a} + \mathbf{b}R^M + \boldsymbol{\varepsilon}, \tag{5.22}$$

where $\mathbf{r} = (r_0, r_1, \ldots, r_{d(n)})'$ is the vector of asset returns, $\mathbf{a} = (a_0, a_1, \ldots, a_{d(n)})'$ and $\mathbf{b} = (b_0, b_1, \ldots, b_{d(n)})'$ are vectors of constant coefficients. We assume that $R^M$ is a market portfolio return which follows a normal distribution with mean $\mu_{R^M}$ and variance $\sigma^2_{R^M}$. The vector of random noises $\boldsymbol{\varepsilon} = (\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{d(n)})'$ follows a multivariate normal distribution (MVN) with mean $\mathbf{0}$ and covariance matrix $D_\varepsilon$, which is denoted by MVN$(\mathbf{0}, D_\varepsilon)$. The matrix $D_\varepsilon$ is diagonal with positive diagonal elements $\sigma^2_{\varepsilon_i}$ for $i = 0, 1, \ldots, d(n)$. Hence, $\mathbf{r}$ follows MVN$\big(\mathbf{a} + (\mu_{R^M})\mathbf{b}, \sigma^2_{R^M}\mathbf{b}\mathbf{b}' + D_\varepsilon\big)$.

Let $\mathbf{e}_0 = (0, 1, \ldots, 1)'$ be a $(1 + d(n))$-column vector. Then the return of an equally-weighted index $R$, which consists of $r_1, r_2, \ldots, r_{d(n)}$, is given by

$$R = \frac{\mathbf{e}_0'\mathbf{r}}{d(n)},$$

then

$$Y = \frac{\mathbf{e}_0'\mathbf{X}}{d(n)}, \tag{5.23}$$

where according to (5.9) $\mathbf{X} = \mathbf{e} + \mathbf{r}$ and $\mathbf{e} = (1, \ldots, 1)'$ is a $(1 + d(n))$ column vector.

Given initial wealth $W_{(0)^-}$ and dollar amounts for each asset $x_i^{(0)^-}$ or equivalently $\beta_i^{(0)^-}$ for $i = 0, 1, \ldots, d(n)$ at time 0, we have $L_{F_n}(\boldsymbol{\beta}) = E\left[\left(Y - \boldsymbol{\beta}'\mathbf{X}\right)^2 | \boldsymbol{\beta}^{(0)^-}\right] = E\left[\left(Y - \boldsymbol{\beta}'\mathbf{X}\right)^2\right]$. Hence, the true risk (or true tracking error) of $\boldsymbol{\beta}$ is given by

$$\begin{aligned} L_{F_n}(\boldsymbol{\beta}) &= E\left[\left(Y - \boldsymbol{\beta}'\mathbf{X}\right)^2\right] = (-1, \boldsymbol{\beta}')E\left[\begin{pmatrix} Y \\ \mathbf{X} \end{pmatrix}(Y', \mathbf{X}')\right]\begin{pmatrix} -1 \\ \boldsymbol{\beta} \end{pmatrix} \\ &= (-1, \boldsymbol{\beta}')\begin{pmatrix} E[YY'] & E[Y\mathbf{X}'] \\ E[\mathbf{X}Y'] & E[\mathbf{X}\mathbf{X}'] \end{pmatrix}\begin{pmatrix} -1 \\ \boldsymbol{\beta} \end{pmatrix} \\ &= \boldsymbol{\beta}' E[\mathbf{X}\mathbf{X}']\boldsymbol{\beta} - 2E[Y\mathbf{X}']\boldsymbol{\beta} + E[YY'], \end{aligned} \tag{5.24}$$

where

$$\begin{aligned} E[YY'] &= Var(Y) + E[Y]^2 = \frac{1}{\left(d(n)\right)^2}\mathbf{e}_0'\Sigma_X\mathbf{e}_0 + \mu_Y^2, \\ E[\mathbf{X}Y'] &= Cov(\mathbf{X}, Y) + E[\mathbf{X}]E[Y]' = \frac{1}{d(n)}\Sigma_X\mathbf{e}_0 + \mu_X\mu_Y, \\ E[\mathbf{X}\mathbf{X}'] &= Cov(\mathbf{X}, \mathbf{X}) + E[\mathbf{X}]E[\mathbf{X}]' = \Sigma_X + \mu_X\mu_X', \end{aligned}$$

and $\Sigma_X = \sigma_{R^M}^2\mathbf{b}\mathbf{b}' + D_\varepsilon$, $\mu_Y = \frac{1}{d(n)}\mu_X'\mathbf{e}_0$, $\mu_X = \mathbf{e} + \mathbf{a} + (\mu_{R^M})\mathbf{b}$. Since $\Sigma_X$ is positive definite, so is $E[\mathbf{X}\mathbf{X}']$. Hence, given any fixed $c_n$, we can efficiently obtain the optimal solution $\tilde{\boldsymbol{\beta}}_n^{(0)^+}$ defined in (5.10)-(5.14) via quadratic programming solvers.

Given $n$ and the parameters in (5.22), we can simulate samples of $\left\{(R_s^M, \boldsymbol{\varepsilon}_s)\right\}_{s=0}^n$, and then generate an in-sample dataset

$$\mathcal{T}^{Sim} = \left\{(Y_s, X_{s,0}, X_{s,1}, \ldots, X_{s,d(n)})' : s = 1, 2, \ldots, n\right\}$$

according to (5.22) and (5.23). Based on $\mathcal{T}^{Sim}$, the one-period index tracking strategy $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ at time $(0)^+$ is given by (5.17) subject to (5.11)-(5.14). The actual risk $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$ can be computed by plugging $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ into (5.24).

The performance of our tracking strategy $\hat{\boldsymbol{\beta}}_n^{(0)^+}$ in finite samples can be investigated by repeating the simulation $S$ times. Based on sufficiently many repetitions, we can construct a confidence interval of $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$ to evaluate how stable $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$ is. The gap

between the averaged $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$ and $L_{F_n}(\tilde{\boldsymbol{\beta}}_n^{(0)^+})$ shows on average how close $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$ is to $L_{F_n}(\tilde{\boldsymbol{\beta}}_n^{(0)^+})$.

### 5.4.2 An Implementation of the Simulation Study

In order to simulate samples, we let $\alpha = 1.25$, and let $\mu_{R^M}$ and $\sigma^2_{R^M}$ be the sample mean and sample variance of the Russell 3000 index weekly returns described in Section 5.5. For $i = 1, 2, \ldots, d(n)$, coefficients $a_i$, $b_i$, and $\sigma^2_{\varepsilon_i}$ are ordinary least square (OLS) estimators of regressing the $i$-th Russell 3000 component weekly return on the Russell 3000 index return. Parameters $a_0$, $b_0$, and $\sigma^2_{\varepsilon_0}$ are OLS estimators of regressing weekly interest rates against the Russell 3000 index return. All these parameter estimators are obtained from data in the recovery environment described in Section 5.5.

Based on $\mathcal{T}^{Sim}$, we construct a tracking portfolio at time 0 given $\beta_0^{(0)^-} = 1$ and $\beta_i^{(0)^-} = 0$ for $i = 1, \ldots, d(n)$. In this implementation, we assume the proportional rate of transaction cost is around the middle of the range $[0.31\%, 2.35\%]$ described in Section 5.1, and let $\theta = 1\%$. Further, let the transaction cost limit $\gamma$ be 1%. We increase the value of $n$ from 100 to 200, and then 450 to evaluate how the tracking strategy behaves as $n$ grows. Results are summarized in Figures 5.1-5.3 respectively. Each of Figures 5.1-5.3 shows the true risk (solid line) $L_{F_n}(\tilde{\boldsymbol{\beta}}_n^{(0)^+})$ *i.e.* true mean square error (MSE), the average of actual risks $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$'s (dash-dot line) and corresponding 90% confidence band of $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$'s (dashed lines) which are obtained from 30 repetitions of simulations.

In all Figures 5.1-5.3, $c_n$ varies from $-1$ to 3, and the minimized true risk $L_{F_n}(\tilde{\boldsymbol{\beta}}_n^{(0)^+})$ decreases as $c_n$ gets larger, which results from enlarged feasible sets. Once $c_n$ is sufficiently large, the true risk does not change. This is because $c_n$ is large enough to disqualify the constraint (5.7). Moreover, the minimized true risk is uniformly smaller than the actual risk $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$, which is true by definition.

Figures 5.1-5.3 show that as $n$ increases the confidence band gets narrower, and the averaged actual risk approaches to the true risk. This verifies the persistence result

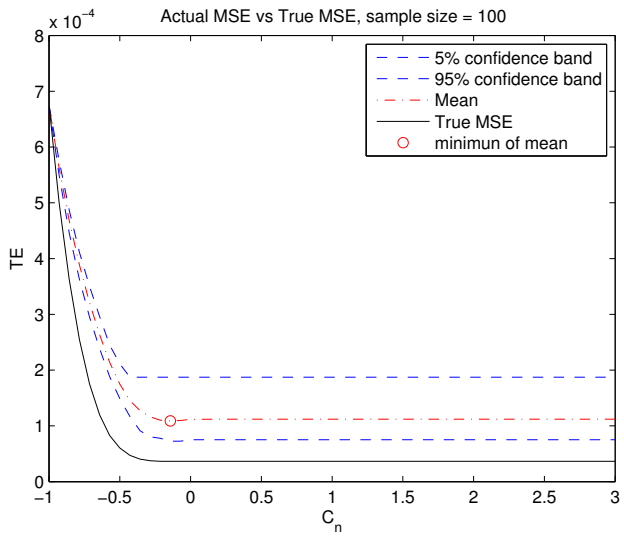



Figure 5.1: Minimized True Risk *vs.* Actual Risk: $n = 100$, #stock=316.

in Theorem 5.1. Moreover, for any fixed $n$, the confidence band gets wider as $c_n$ increases. This verifies the result (5.20) in Theorem 5.1, which says that the upper bound of $P_{F_n}\left\{L_{F_n}(\hat{\boldsymbol{\beta}}_n) - L_{F_n}(\boldsymbol{\beta}_n^*) \geq \delta\right\}$ becomes larger as $\sup_{\boldsymbol{\beta}\in B^{(n)}} ||\boldsymbol{\beta}||_1$ gets bigger.

All Figures 5.1-5.3 indicate a tradeoff between the magnitude of the actual risk $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$ and its stableness. When $c_n$ is small, the averaged actual risk is close to the minimized true risk $L_{F_n}(\tilde{\boldsymbol{\beta}}_n^{(0)^+})$ and the confidence band is narrow, but the minimized true risk is relatively large. When $c_n$ is large, the minimized true risk is small, but the averaged actual risk deviates from the minimized true risk due to larger estimation errors, which is suggested by wider confidence intervals. In practice, it is necessary to choose a $c_n$ to implement the tracking strategy. In this chapter, we follow the suggestion in [69, p.221] and choose the value of $c_n$ which leads to the minimum of the averaged actual risk.

## 5.5 Application with Financial Data

In this section, our index-tracking method with $L_1$-regularization, which is repeatedly solving (5.17) subject to (5.11)-(5.14), is applied to real financial data. Firstlly, we consider a

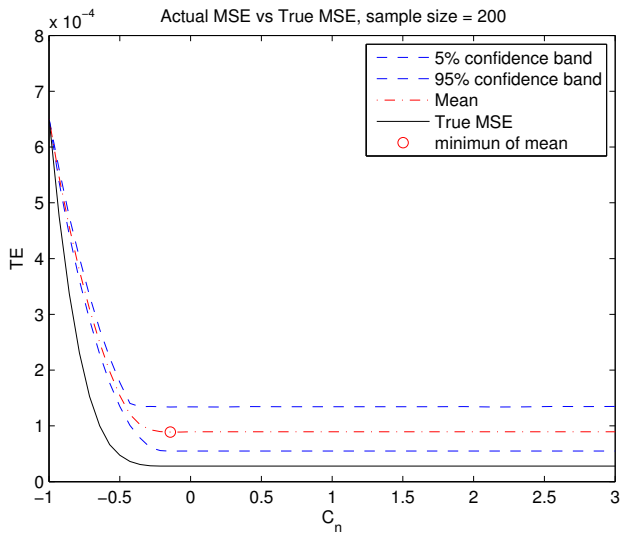



Figure 5.2: Minimized True Risk *vs.* Actual Risk, $n = 200$, #stock=752.

one-period problem, where the $L_q$-penalty ($0 < q < 1$) method in [49] is used as a benchmark. In this case, our method with $L_1$-regularization has better tracking performance. Secondly, our one-period tracking method is repeated to solve a multi-period tracking problem. More specifically, the one-period method is rebalanced period by period in a rolling window setting. Compared to the full-replication strategy, our method has a better tracking performance.

### 5.5.1 Data

We consider two U.S. capitalization-weighted stock-market indices, the Russell 2000 and Russell 3000, of which the majority of index components are small-cap stocks. Weekly levels of Russell 2000 (3000), as well as their components' weekly prices, are downloaded from the Bloomberg terminal. We study weekly data in two economic environments, the recession environment from March 5th, 2004 to September 25th, 2009 (which covers the 2008 financial crisis) and the recovery environment from October 2nd, 2009 to April 24th, 2015. Each recession/recovery environment includes 291 weekly observations of both index levels and stock prices. Corresponding weekly interest rates are calibrated from the 1-

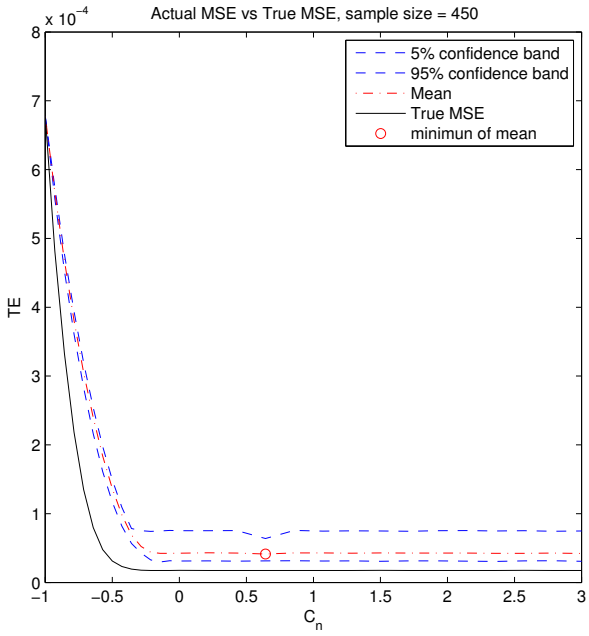



Figure 5.3: Minimized True Risk *vs.* Actual Risk, $n = 450$, #stock=2,072.

month T-bill discount rate. More specifically, suppose $y_{tb}$ is the annual discount rate of a 1-month T-bill, then $r_0 = \left[(1 + y_{tb})^{1/12} - 1\right]/4$.

Since there are missing data for many index components, we construct and track synthetic versions of the Russell 2000 (3000). In doing that, stocks (listed as index components on April 24th, 2015) with any missing weekly data deleted. Numbers of components of synthetic Russell 2000 (3000) in the recession and recovery environments are illustrated in Table 5.1. Synthetic capitalization-weighted stock-market indices are constructed according to

$$I_t = \frac{1}{D} \sum_{i=1}^{d} a_i S_{t,i}, \text{ for } t = 1, 2, \ldots, \tag{5.25}$$

where $I_t$ is the index level at time $t$, $S_{t,i}$ is the stock price for stock $i$ at time $t$, $a_i$ is the number of outstanding shares for stock $i$, and $D$ is the index divisor. Over the recession environment, the $a_i$'s in the synthetic Russell 2000 (3000) remain the same as those on September 25th, 2009, and the index divisor of the Russell 2000 (3000) is the number

that equals the synthetic index level on September 25th, 2009 to the real Russell 2000 (3000) level observed on that day. A similar procedure is used for the case of the recovery environment, and the $a_i$'s and index divisor are determined by the outstanding shares and the index level on April 24th, 2015.

Table 5.1: The number of components of synthetic indicies

| Synthetic indicies | Russell 2000 (Recession) | Russell 3000 (Recession) | Russell 2000 (Recovery) | Russell 3000 (Recovery) |
|---|---|---|---|---|
| Number of components | 907 | 1,601 | 1,306 | 2,137 |

Weekly discrete-time returns of stocks and synthetic indices in the recession (or recovery) environment are partitioned into the in-sample data $\mathcal{T}_{Recession}$ (or $\mathcal{T}_{Recovery}$) with 200 weekly returns, and out-of-sample data $\mathcal{V}_{Recession}$ (or $\mathcal{V}_{Recovery}$) with 90 weekly returns. In the following, we always refer to the synthetic Russell 2000 (3000) as the Russell 2000 (3000).

Since the majority of Russell 2000 (3000) components are small-capitalization stocks, in all applications in this section we let $\theta = 1\%$ which is around the middle of the transaction cost range $[0.31\%, 2.35\%]$ discussed in Section 5.1. We also let $\gamma = 1\%$.

### 5.5.2 One Period Performance

In this subsection, we only show results on tracking the Russell 2000 index in the recovery environment, since tracking the Russell 2000 in the recession environment, as well as tracking the Russell 3000 index in the recession and recovery environments, yields similar results. Based on $\mathcal{T}_{recovery}$, our one-period $L_1$-regularization tracking method is applied to determine a tracking strategy $\hat{\boldsymbol{\beta}}_n^{(0)^+}$, given $\beta_0^{(0)^-} = 1$ and $\beta_i^{(0)^-} = 0$ for $i = 1, \ldots, d$.

In order to investigate the performance of $\hat{\boldsymbol{\beta}}_n^{(0)^+}$, we use the $L_q$-penalty method in [49] as a benchmark. A brief description of this method can be found in Section 5.1. The

$L_q$-penalty tracking method is formulated into an optimization problem with a non-convex objective function, which is given by

$$\begin{aligned} \hat{\boldsymbol{\beta}}_n^q &= \arg\min_{\boldsymbol{\beta}} \hat{E}\left[\left(Y-\boldsymbol{\beta}'\mathbf{X}\right)^2 \middle| \boldsymbol{\beta}^{(0)^+}\right] + \lambda\Big(\sum_{i=0}^{d}|\beta|^q\Big)^{1/q} \qquad (5.26) \\ s.t. \quad & \sum_{i=0}^{d}\beta_i + \sum_{i=1}^{d}\theta|\beta_i^{(0)^-} - \beta_i| \le 1 \\ & \beta_i \ge 0, \text{ for } i = 1,\ldots,d, \\ & \theta\sum_{i=1}^{d}|\beta_i^{(0)^-} - \beta_i| \le \gamma. \end{aligned}$$

In the above formulation, the constraint $-c \le \beta_0$ is ignored since the $L_q$ penalty in the objective function is able to generate sparsity. Further, the persistency is not discussed in the $L_q$-penalty method in [49], so that it is not necessary to force the constraint $-c \le \beta_0$. This relaxation expands the feasible set.

Following [49], we let $q = 0.5$ in our implementation. Following algorithms in [50] and [49], we carry out the hybrid heuristic algorithm to solve (5.26) with corresponding constraints. This takes around 2 hours for each implementation. All applications in this chapter are carried out on a PC with Intel Core i5-3210M CPU at 2.5GHz and 6.00GB memory. Since the hybrid heuristic algorithm usually leads to suboptimal solutions, we always repeat it three times with the same inputs and report the averaged result.

Results of $L_q$-penalty methods are sensitive to $\lambda$ in (5.26), so that tuning $\lambda$ is important to this $L_q$-penalty method. Due to the computational burden, we cannot try too many different $\lambda$'s. Instead, we vary $\lambda$ in a carefully chosen candidate set {1.00E-9, 1.00E-7, 1.00E-6, 5.00E-6, 1.00E-5, 1.00E-4}, which covers a sufficiently large range (see discussions of Table 5.2).

Since the true joint distribution is unknown in real applications, we cannot evaluate the tracking performance by comparing the actual risk $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$. However, it can be estimated by the out-of-sample MSE, *i.e.* $\frac{1}{N^{out}}\sum_{s=1}^{N^{out}}(Y_s - \big(\hat{\boldsymbol{\beta}}_n^{(0)^+}\big)_n'\mathbf{X}_s)^2$, where $(Y_s, \mathbf{X}_s')' \in \mathcal{V}_{recovery}$ and $N^{out}$ is the sample size of $\mathcal{V}_{recovery}$. According to the weak law of large numbers,

the out-of-sample MSE converges to $L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})$ in probability under the assumption of i.i.d. data. Thus, the out-of-sample MSE is used as a criterion to evaluate the tracking performance of $\hat{\boldsymbol{\beta}}_n^{(0)^+}$, which is obtained from $\mathcal{T}_{recovery}$.

Actually, in order to evaluate the tracking performance with finite in-sample data, it is more suitable to compare an estimator of the expectation $E\left[L_{F_n}(\hat{\boldsymbol{\beta}}_n^{(0)^+})\right]$ with that of $E\left[L_{F_n}(\hat{\boldsymbol{\beta}}_n^{q})\right]$. Suggested by [69, p.254], they can be estimated by a $K$-fold cross-validation or the averaged out-of-sample MSE given by bootstrapped in-sample data. However, the bootstrapping and $K$-fold cross-validation methods require extra computational efforts. Even though they can be applied to our $L_1$-regularization method due to the efficiency of solving a quadratic program, they are too computationally expensive for the $L_q$-penalty method. For the $L_q$-penalty method, each implementation of a $K$-fold cross-validation requires $(K+1) \times \#\lambda \times 3 \times 2$ hours, and each implementation of a bootstrapping method takes around $B \times \#\lambda \times 3 \times 2$ hours where $B$ is the number of bootstrapped scenarios and $\#\lambda$ is the number of candidate $\lambda$'s. Hence, we only compare the $L_1$-regularization method with the $L_q$-penalty method according to the out-of-sample MSE based on $\mathcal{T}_{recovery}$.

Figure 5.4 summaries the results of our $L_1$-regularization tracking method. The solid line is the out-of-sample MSE based on the original $\mathcal{T}_{recovery}$. In order to evaluate how stable the out-of-sample MSE is, we bootstrapped the in-sample data 1,000 times. Dashed lines show the 90% confidence band given by the bootstrapping percentile method ([37]), and the dash-dot line represents the averaged out-of-sample MSE given by bootstrapped in-sample data.

Figure 5.4 shows that both the original out-of-sample MSE and the averaged out-of-sample MSE decrease as $c_n$ increases at the beginning. After they reach their minima, they curve up until they turn flat. When $c_n$ is small, the out-of-sample MSE is quite stable since the confidence band is narrow. However, when $c_n$ is large the confidence band is much wider, so that the out-of-sample MSE is not stable, and it is enlarged by estimation errors. A sufficiently large $c_n$ disqualifies the constraint (5.6), so the curves turn flat.

The value of $c_n$, to be applied in the $L_1$-regularization tracking strategy, is the number where the solid curve reaches its minimum. At that point, the out-of-sample MSE based

on $\mathcal{T}_{recovery}$ is 1.2238E-4. At this value of $c_n$, the $L_1$-regularization strategy selects 265 out of 1,306 components to construct a tracking portfolio.

Table 5.2 shows results of the $L_q$-regularization tracking strategy based on $\mathcal{T}_{recovery}$. Each result in the second and third columns is an averaged result of three implementations. The last column shows the total elapsed time to obtain averaged results. Table 5.2 shows that the out-of-sample MSE gets smaller when $\lambda$ decreases from 1.00E-4. After it reaches the minimum at $\lambda$ =1.00E-07, it gets larger due to estimation errors. The range of candidate $\lambda$'s appears to be large enough since the number of selected stocks (column three in Table 5.2) varies from 12 to $1,035$ which almost reaches the total number of Russell 2000 components in the recovery environment.

According to the out-of-sample MSE, the $L_1$-regularization tracking method outperforms the $L_q$-penalty method. In Figure 5.4, the minimum value of the solid curve is 1.2238E-4, while Table 5.2 shows that the minimum out-of-sample MSE of the latter one is 1.2330E-04. Moreover, the $L_1$-regularization method is much more computationally efficient. Generating each path of the out-of-sample MSE (such as the solid curve) in Figure 5.4 takes around 20 minutes, while generating Table 5.2 takes 36 hours. Another advantage of the $L_1$-regularization method is its persistent property. As far as we know, whether the $L_q$-penalty method is persistent or not is still an open question in the large $d$ small $n$ setting.

Table 5.2: Results of applying the $L_q$-penalty method to track the Russell 2000

| $\lambda$ | Out-of-sample MSE | # stocks | Time (seconds) |
|---|---|---|---|
| 1.00E-9 | 1.2522E-04 | 1,035 | 22,890 |
| 1.00E-7 | **1.2330E-04** | 518 | 22,293 |
| 1.00E-6 | 1.7049E-04 | 77 | 22,431 |
| 5.00E-6 | 2.0231E-04 | 21 | 22,515 |
| 1.00E-5 | 2.0845E-04 | 19 | 22,536 |
| 1.00E-4 | 2.7722E-04 | 12 | 22,488 |

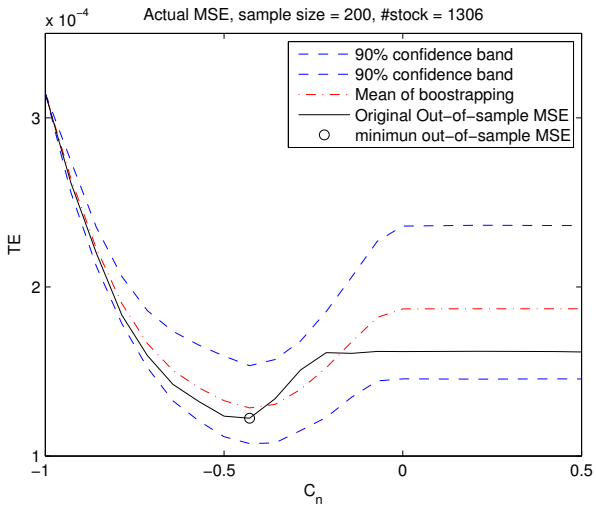



Figure 5.4: Results of the $L_1$-regularization method to track the Russell 2000

### 5.5.3 Multiple Period Performance

The one-period $L_1$-regularization tracking method is repeated period by period to solve a multi-period tracking problem. More specifically, we carry out a rolling-window method. In the first period, starting with initial wealth $W_{(0)^-}$, a tracking portfolio is constructed from a pure cash position based on the first 200 in-sample data. Using the 201st data point, we compute $(W^I_{(1)^-} - W_{(1)^-})^2$. In the next period, the in-sample window is moved one-step further, and the tracking portfolio is rebalanced based on the 2nd to the 201st data points. Using the 202nd data point, we compute $(W^I_{(2)^-} - W_{(2)^-})^2$, and so on. In this subsection, the Russell 2000 and the Russell 3000 are tracked by a 30-period rolling window method.

In each period, we need to determine the tuning parameter $c_n$, and this is carried out by a 5-fold cross validations. The 5-fold cross validation is adopted, since computing the bootstrapped averaged out-of-sample MSE is too computationally expensive in a rolling window setting. Implementing the $L_1$-regularization tracking method in each period takes around 1 hour.

The multi-period tracking performance is evaluated by the normalized tracking error

at time $T$, $TE(T)$, which is given by

$$\mathrm{TE}(T) = \frac{\left(W^I_{(T)^-} - W_{(T)^-}\right)^2}{(W_{(0)^-})^2}.$$

We further compare $TE(T)$ of different tracking methods in tracking different indices. The full-replication strategy is used as the benchmark in multi-period cases. We give up using the $L_q$-penalty method as a benchmark since it is too computationally expensive in the rolling window setting. Repeating the $L_q$-penalty method to solve a 30-period problem requires $30 \times K \times \#\lambda \times 3 \times 2$ hours, when a $K$-fold cross-validation is applied to tune $\lambda$ in each period.

The methodology of a full-replication strategy is as follows. At time 0, $W_{(0)^-}$ in cash is used to construct a tracking portfolio according to the full replication strategy. For $i = 1, \ldots, d$, denote by $N_i^{tp}$ the number of shares for each stock bought at time 0, and $S_{0,i}$ the stock price at time 0. Suppose $N_i^{tp}$ can be fractions, then

$$\begin{cases} \sum_{i=1}^d N_i^{tp} S_{0,i} + \sum_{i=1}^d \theta |N_i^{tp} S_{0,i}| = W_{(0)^-}. \\ N_i^{tp} \geq 0, \text{ for } i = 1, \ldots, d. \end{cases}$$

Thus, $\sum_{i=1}^d N_i^{tp} S_{0i} = \frac{1}{1+\theta} W_{(0)^-}$. Since stock weights of the full-replication match those in the index, we have

$$\frac{N_i^{tp} S_{0i}}{\frac{1}{1+\theta} W_{(0)^-}} = \frac{\frac{a_i}{D} S_{0i}}{I_0}, \text{ for } i = 1, \ldots, d,$$

where $I_0 = \sum_{i=1}^d \frac{a_i}{D} S_{0i}$. Then $N_i^{tp} = \frac{1}{1+\theta} \cdot \frac{W_{(0)^-}}{I_0} \cdot \frac{a_i}{D}$. At time $(t)^-$ for $t = 1, \ldots, T$, note the definition of $I_t$ in (5.25), then the tracking portfolio value $W^{full}_{(t)^-}$ is

$$\begin{aligned} W^{full}_{(t)^-} &= \sum_{i=1}^d N_i^{tp} S_{t,i} = \frac{1}{1+\theta} \cdot \frac{W_{(0)^-}}{I_0} \sum_{i=1}^d \frac{a_i S_{t,i}}{D} \\ &= \frac{1}{1+\theta} \cdot W_{(0)^-} \frac{I_t}{I_0} = \frac{1}{1+\theta} W^I_{(t)^-}. \end{aligned}$$

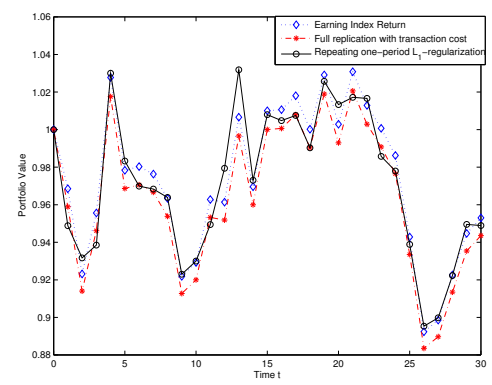
The normalized tracking error of a full replication portfolio at time $(T)^-$ is given by

$$\Big(\frac{\big|W^I_{(T)^-} - W^{full}_{(T)^-}\big|}{W_{(0)^-}}\Big)^2 = \Big(\frac{\theta}{1+\theta}\frac{|I_T|}{I_0}\Big)^2 = \Big(\frac{\theta}{1+\theta}\Big)^2 \prod_{s=1}^{T}\big(1+R_s\big)^2.$$

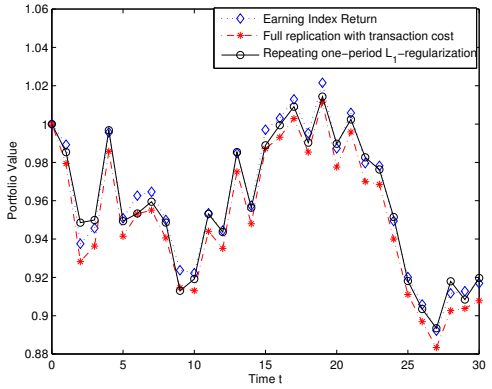
Hence, the normalized tracking error of a full-replication strategy depends on the magnitude of $\theta$. It is expected that the (normalized) tracking error of a full-replication is small when it tracks a large-capitalization stock-market index, in which case the transaction costs are small. However, the full-replication strategy could suffer a large (normalized) tracking error when it is benchmarked to a small-capitalization stock-market index, in which case the transaction costs are large.

Figure 5.5 shows paths of tracking portfolio values benchmarked to the synthetic Russell 2000 and Russell 3000 indices in both the recession and recovery environments, as well as the corresponding index level paths. We assume that both the initial index level and initial portfolio wealth start from 1. All four sub-figures in Figure 5.5 indicate that the wealth of full replication is uniformly below the synthetic index level but exactly follows the trend of the index level since time 1. The gap between the synthetic index level and the full-replication portfolio value is induced by transaction costs at construction. Overall, the tracking portfolio constructed by repeating the $L_1$-regularization method is closer to the index level. Figure 5.6 provides more convincing evidence. It shows normalized tracking errors of the full replication and the portfolio constructed by repeating the $L_1$-regularization method from time 1 to 30. In most cases, repeating the $L_1$-regularization method leads to smaller normalized tracking errors.

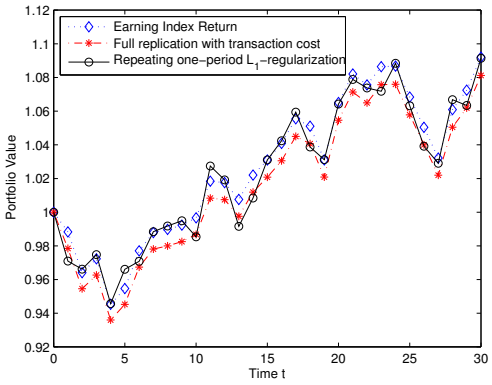
In tracking both the Russell 2000 and the Russell 3000, the number of selected stocks is small at the first several rebalancings for the $L_1$-regularization tracking strategy. For example, in the recovery environment, the $L_1$-regularization tracking strategy selects 118 (161) stocks to track the Russell 2000 (3000) in the first period. In general, this number increases gradually at each rebalancing. Eventually, all Russell 2000 (3000) components are included within 30 rebalancings. Due to the nature of stock-market indices, a good tracking portfolio is expected to include as many components as possible. Similar changes can be found in the recession environment.



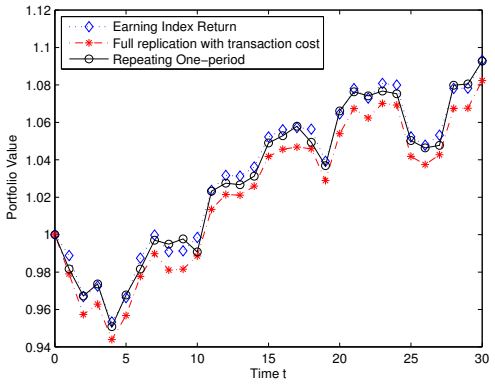

(a) Russell 2000 (Recession)




(b) Russell 3000 (Recession)



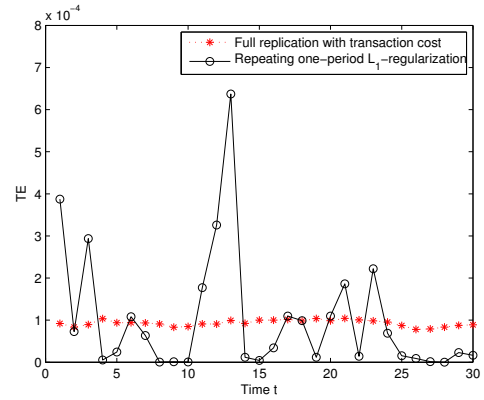

(c) Russell 2000 (Recovery)



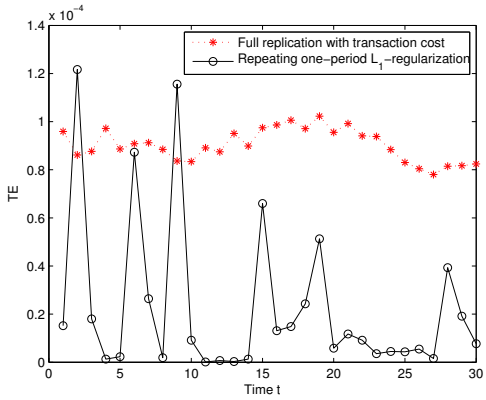

(d) Russell 3000 (Recovery)

Figure 5.5: Tracking portfolio values *vs.* index level

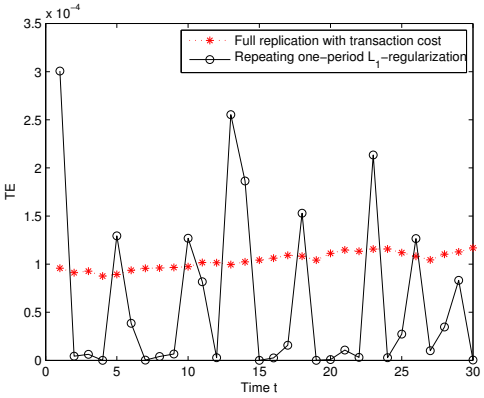
As discussed in Section 5.2.2, it is possible to withdraw money from $W_{(t-1)^-}$ while rebalancing the $L_1$-regularization tracking strategy at time $t-1$ for $t = 1, 2, \ldots$. During 30 rebalancings of tracking the Russell 2000 in the recovery environments, the ratios of withdrawn money to $W_{(t-1)^-}$ vary from 8.20E-08 to 2.53E-3, and the average ratio is 1.52E-3. During 30 rebalancings of tracking the Russell 3000 in the recovery environments, ratios vary from 6.65E-7 to 1.68E-3 with an average of 1.14E-3. In all cases, the withdrawn money takes a little portion of the portfolio wealth before rebalancing. Similar magnitudes of withdraw money can be observed in the recession environment.



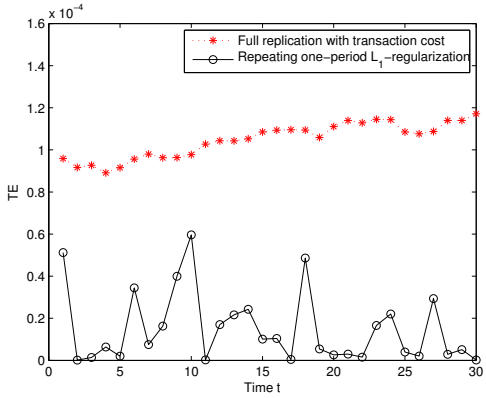

(a) Russell 2000 (Recession)




(b) Russell 3000 (Recession)




(c) Russell 2000 (Recovery)




(d) Russell 3000 (Recovery)

Figure 5.6: Normalized tracking errors of tracking portfolios

Among 30 rebalancings to track the Russell 2000 or Russell 3000 in both the recession and recovery environments, transaction costs of the $L_1$-regularization method decrease quickly at the beginning and remain stable thereafter. This is because constructing tracking portfolios with a pure cash position spends a large initial transaction cost compared with transaction costs spent at following rebalancings.

## 5.6 Conclusion

In this chapter, aiming at reducing the gap between the tracking portfolio terminal wealth and the terminal wealth given the initial wealth (before construction) earns exactly the index return, a multi-period index tracking problem is solved by repeatedly solving one-period index tracking problems. Transaction costs and other practical constraints are considered in our index tracking formulation. Since the true distribution of financial returns is usually unknown, the one-period index tracking strategy is obtained by minimizing the empirical tracking error. With an $L_1$-regularization on asset weights, our one-period tracking strategy enjoys persistent properties in the high-dimensional setting. Simulation studies are carried out to support our one-period tracking strategy's performance with finite samples. Applications on real financial data provide evidence that, under certain circumstances, our tracking strategy outperforms benchmark methods in the one-period and multi-period cases.

In this chapter, we estimate the true covariance matrix by the sample covariance matrix. Improved covariance matrix estimators, such as shrinkage methods in [86] and [104], are very likely to improve the tracking performance. It is interesting to investigate the persistence property with improved covariance matrix estimators, and the order of the $L_1$-regularization might be extended. Inspired by results in [48], the sample covariance matrix, obtained from the assumption that stock returns follow factor models, is very likely to improve the tracking performance, especially when the number of index component is large.

More interesting future works include generalizing the i.i.d. assumption of financial returns to the world of jointly stationary and ergodic processes. Following studies in [127] and [24], it is possible to prove persistency of $L_1$-regularization under that assumption. Another direction of future work, in the time series setting, is to construct co-integration systems ([40]) to reproduce the index level. Even though some studies have been carried out in this direction ([5]), it would be exciting if some results on co-integration were obtained in the high-dimensional setting.

# Chapter 6

# Future Works

This chapter states some potential directions for future research about sparse models on vine copula and index tracking.

## 6.1 Potential Directions for Vine Copulas

The vine copula is a flexible tool to describe multivariate dependence structures. Unfortunately, its complexity grows exponentially along with the number of variables in the model. In addition to the sparse vine introduced in Chapter 2, another promising method for simplifying vine copulas is the factor copula model proposed in [83], where the random variables of interest are assumed to depend on several common latent factors, and their dependence can be modelled by a vine copula truncated at several levels. We are interested in the applications of factor copula models for modelling financial asset returns with observable factor variables. In the literature, there are a lot of works searching for what factors (mostly macro-economic variables rather than latent factors) are most important to explain stock returns. If a few macro-economic variables are chosen as factors, vine copulas used to describe dependence among stock returns can be truncated at several levels. Since factor models are very successful in explaining stock returns, such a factor-based

vine copula is concise and promising for describing the dependence structure among stock returns. Compared with linear factor models described in Section 4.3, factor-based vine copulas are more flexible at describing dependence structures. In terms of searching for important factors among numerous macro-economic variables, variable selection methods such as Lasso provide many helpful tools.

## 6.2 Potential Directions for Index Tracking

In Chapters 4 and 5, most tracking strategies are obtained by minimizing the empirical tracking error, such as the mean square error. Properties of those tracking strategies are closely related to properties of linear regression estimators. Recently, the Dantzig selector is introduced in [23] to obtain linear regression estimators in the high-dimensional statistical setting. The Dantzig selector can be expressed as minimizing a linear objective function subject to the $L_1$ regularization on estimators ([36]). The authors of [23] claim that, under some conditions, using the Dantzig selector it "is possible nearly to select the best subset of variables". It is proved in [12] that, under some assumptions, the Dantzig selector estimator is persistent to the optimal parameters with respect to minimizing the true mean square error.

As one can see from the previous chapters, minimizing the empirical mean square error (MSE) in an index tracking problem with the cardinality constraint usually relies on mixed-integer quadratic programming. This is what we try to avoid in Chapters 3-5, especially when the number of index components is large. An idea we can borrow from the Dantzig selector to control the mean square tracking error is to construct tracking strategies by minimizing the Dantzig selector linear objective function. With such a linear objective function, it is computationally tractable even considering the cardinality constraint. In this case, the index tracking problem boils down to a mixed-integer linear program which can be solved efficiently. By minimizing the Dantzig selector linear objective function subject to the cardinality constraint, it is very likely to obtain a tracking strategy which leads to a tracking error (or MSE) that is close to the true minimum tracking error (or MSE). A recent working paper ([90]) discusses some relevant theoretical properties of the

linear regression estimators, which are obtained by minimizing the Dantzig selector linear objective function subject to the $L_0$-regularization.

All theoretical results in Chapters 3-5 rely on the assumptions that financial asset returns at different time are independent and identically distributed. However, it is more realistic to assume that financial returns are stationary. Even though it is more challenging to derive theoretical results in a setting with serial dependence, some works have been done in this direction to investigate properties of linear model parameter estimators in the high-dimensional statistical setting.

Persistent or consistent properties of tracking strategies play a key role in index tracking methods, given that tracking strategies are derived by minimizing the empirical tracking error. In Chapters 4 and 5, persistent properties of tracking strategies that minimize the empirical mean square error are derived under the assumption of i.i.d. financial returns. However, some relevant theoretical results are obtained in [24] with the assumption of serial dependence. The authors of [24] prove that the ridge regression estimators are consistent in a high-dimensional statistical setting under the assumption of stationarity. Even though the consistency or persistency of Lasso estimators is not discussed in [24], it is very likely to prove the persistency (with respect to MSE) of Lasso estimators under the same assumptions. Persistency of Lasso estimators, if obtained, directly applies to tracking strategies that minimize the empirical tracking error with $L_1$ regularizations on asset weights.

Another direction of future research, still in the time series setting, is to reproduce index levels by co-integration systems ([40]). Chapters 3-5 concentrate on mimicking one-period index returns. In a one-period problem, mimicking index returns is equivalent to matching index levels. However, they are not equivalent in a multi-period tracking problem. Co-integration methods provide a tool to directly approximate index levels, which is an intuitive multi-period tracking target. Some studies on index tracking have been carried out along this direction ([5]). However, it could be of interest if some results on co-integration are obtained in the high-dimensional setting.

# References


[1] Aas, K., C. Czado, A. Frigessi, and H. Bakken (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics 44*(2), 182–198.

[2] Abegaz, F., I. Gijbels, and N. Veraverbeke (2012). Semiparametric estimation of conditional copulas. *Journal of Multivariate Analysis 110*, 43–73.

[3] Acar, E., V. Craiu, and F. Yao (2011). Dependence calibration in conditional copulas: a nonparametric approach. *Biometrics 67*, 445–453.

[4] Acar, E., C. Genest, and J. Néslehova (2012). Beyond simplified pair-copula constructions. *Journal of Multivariate Analysis 110*, 74–90.

[5] Alexander, C. (1999). Optimal hedging using cointegration. *Philosophical Transactions of the Royal Society of London Series A - Mathematical Physical and Engineering Sciences 357*, 2039–2058.

[6] Bai, J. and S. Shi (2011). Estimating high dimensional covariance matrices and its applications. *Annals of Economics and Finance 12*(2), 199–215.

[7] Beasley, J., N. Meade, and T.-J. Chang (2003). An evolutionary heuristic for the index tracking problem. *European Journal of Operational Research 148*, 621–643.

[8] Bedford, T. and R. Cooke (2001). Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial intelligence 32*, 245–268.

[9] Bedford, T. and R. Cooke (2002). Vines - a new graphical model for dependent random variables. *Annals of Statistics 30*, 1031–1068.

[10] Bender, J., R. Briand, G. Nielsen, and D. Stefek (2010). Portfolio of risk premia: A new approach to diversification. *The Journal of Portfolio Management 36*(2), 17–25.

[11] Berman, D. (2016, February). Scotiabank embraces analytics as it shifts to data-driven strategy. http://www.theglobeandmail.com/report-on-business/scotiabank-embraces-analytics-as-it-shifts-to-data-driven-strategy/article28519206/.

[12] Bickel, P., Y. Ritov, and A. Tsybakov (2009). Simultanesous analysis of Lasso and Dantzig selector. *The Annals of Statistics 37*(4), 1705–1732.

[13] BIS (2006, June). *Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework - Comprehensive Version*. Basel Committee on Banking Supervision International.

[14] Brechmann, E. and C. Czado (2013). Risk management with high-dimensional vine copulas: An analysis of the euro stoxx 50. *Statistics & Risk Modeling 30(4)*, 307–342.

[15] Brechmann, E., C. Czado, and K. Aas (2012). Truncated regular vines in high dimensions with applications to financial data. *Canadian Journal of Statistics 40*, 68–85.

[16] Brechmann, E. C. (2010). Truncated and simplified regular vines and their applications. *Technische Universität München Diplom Thesis*.

[17] Brodie, J., I. Daubechies, C. De Mol, D. Giannone, and I. Loris (2002). Optimal benchmark tracking with small portfolios. *The Journal of Portfolio Management 28*(2), 33–39.

[18] Brodie, J., I. Daubechies, C. De Mol, D. Giannone, and I. Loris (2009). Sparse and stable Markowitz portfolios. *Proceedings of National Academy of Sciences of the United States of America 106*(30), 12267–12272.

[19] Burnham, K. and D. Anderson (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological Methods and Research 33*, 261–304.

[20] Cadima, J., J. Cerdeira, and M. Minhoto (2004). Computational aspects of algorithms for variable selection in the context of principal components. *Computational Statistics & Data Analysis 47*, 225–236.

[21] Cadima, J. and I. Jolliffe (2001). Variables selection and the interpretation of principal subspaces. *Journal of Agricultural, Biological, and Environmental Statistics 6*, 62–79.

[22] Canakgoz, N. and J. Beasley (2008). Mixed-integer programming approaches for index tracking and enhanced indexation. *European Journal of Operational Research 196*, 384–399.

[23] Candes, E. and R. Tao (2007). The Dantzig selector: Statistical estimation when $p$ is much larger than $n$. *The Annals of Statistics 35*(6), 2313–2351.

[24] Carrasco, M. and B. Rossi (2016). In-sample inference and forecasting in misspecified factor models. *Journal Of Business & Economic Statistics 34*(3), 313–338.

[25] Cherubini, U., E. Luciano, and W. Vecchiato (2004). *Copula Methods in Finance*. England: John Wiley & Sons.

[26] Corielli, F. and M. Marcellino (2006). Factor based index tracking. *Journal of Banking and Finance 30*, 2215–2233.

[27] Cormen, T., C. Leiserson, R. Rivest, and C. Stein (2001). *Introduction to Algorithms, 2nd Edition*. Cambridge: The MIT Press.

[28] CPP (2014). 2014 annual report. CPP Investment Board.

[29] Cvitanic, J. and I. Karatzas (1992). Convex duality in constrained portfolio optimization. *The Annals of Applied Probability 2*(4), 767–818.

[30] Czado, C., E. Brechmann, and L. Gruber (2013). Selection of vine copulas. In F. D. P. Jaworski and W. K. Hardle (Eds.), *Copulae in Mathematical and Quantitative Finance: Proceedings of the Workshop Held in Cracow, 10-11 July 2012*. Springer.

[31] Czado, C., S. Jeske, and M. Hofmann (2012). Selection strategies for regular vine copulae. Submitted to Soumis au Journal de la Societe Francaise de Staatistique.

[32] Czado, C. and A. Min (2011). Bayesian inference for d-vines: Estimation and model selection. In D. Kurowicka and H. Joe (Eds.), *Handbook on Vines*. World Scientific.

[33] Diebold, F., T. Gunther, and A. Tay (1998). Evaluating density forecasts with applications to financial risk management. *International Economic Review 39*(4), 863–883.

[34] Dziak, J., D. Coffman, and R. Li (2012). Sensitivity and specificity of information criteria. The Pennsylvania State University Technical Report Series #12-119, College of Health and Human Development, The Pennsylvania State University.

[35] Efron, B. (2004). The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association 99*(467), 619–642.

[36] Efron, B., T. Hastie, and R. Tibshirani (2007). Discussion: The Dantzig selector: Statistical estimation when $p$ is much larger than $n$. *The Annals of Statistics 35*(6), 2358–2364.

[37] Efron, B. and R. Tibshirani (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science 1*(1), 54–77.

[38] Elton, E. and M. Gruber (1977). Risk reduction and portfolio size: An analytical solution. *The Journal of Business 50*, 415–437.

[39] Embrechts, P., A. McNeil, and D. Strauman (2002). Correlation and dependence in risk management: Properties and pitfalls. In M. A. H. Dempster (Ed.), *Risk Management: Value at Risk and Beyond*, pp. 176–223. Cambridge University Press.

[40] Engle, R. and C. Granger (1987). Co-integration and error correction: Representation, estimation, and testing. *Econometrica 55*(2), 251–276.

[41] Evans, J. and S. Archer (1968). Diversification and the reduction of dispersion: An empirical analysis. *The Journal of Finance 23*, 761–767.

[42] Fama, E. and K. French (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics 33*, 3–56.

[43] Fama, E. and K. French (2004). The capital asset pricing model: Theory and evidence. *Journal of Economic Perspectives 18*(3), 25–46.

[44] Fan, J., F. Han, H. Liu, and B. Vickers (2016). Robust inference of risks of large portfolios. *Journal of Econometrics 194*, 298–308.

[45] Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association 96*, 1348–1360.

[46] Fan, J., Y. Liao, and M. Mincheva (2011). High-dimensional covariance matrix estimation in approximate factor models. *The Annals of Statistics 39*, 3320–3356.

[47] Fan, J. and J. Lv (2011). Non-concave penalized likelihood with np-dimensionality. *IEEE – Information Theory 57*, 5467–5484.

[48] Fan, J., J. Zhang, and K. Yu (2012). Vast portfolio selection with gross-exposure constraints. *Journal of the American Statistical Association 107*(498), 592–606.

[49] Fastrich, B., S. Paterlini, and P. Winker (2014). Cardinality versus q-norm constraints for index tracking. *Quantitative Finance 14*(11), 2019–2032.

[50] Fastrich, B. and P. Winker (2012). Robust portfolio optimization with a hybrid heuristic algorithm. *Computational Management Science 9*(1), 63–88.

[51] Focardi, S. and F. Fabozzi (2004). A methodology for index tracking based on time-series clustering. *Quantitative Finance 4*, 417–425.

[52] Frank, I. and J. Friedman (1993). A statistical view of some chemometrics regression tools. *Technometrics 35*(2), 109–135.

[53] Frino, A. and D. Gallagher (2001). Tracking S&P 500 index funds. *The Journal of Portfolio Management 28*, 44–55.

[54] FRR (2013). 2013 annual report. Fonds de Reserve Pour Les Retraites.

[55] Gaivoronski, A., S. Krylov, and N. van der Wijst (2005). Optimal portfolio selection and dynamic benchmark tracking. *European Journal of Operational Research 163*, 115–131.

[56] Gao, X. and P.-K. Song (2010). Composite likelihood bayesian information criteria for model selection in high-dimensional data. *Journal of the American Statistician Association 105*(492), 1531–1540.

[57] Gastineau, G. (2010). *The Exchange-Traded Funds Manual, 2nd edition*. Wiley.

[58] Genest, C. and A. Favre (2007). Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of Hydrologic Engineering 12*(4), 347–368.

[59] Genest, C., K. Ghoudi, and L. Rivest (1995). A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika 82*, 543–552.

[60] Ghalanos, A. (2014). Introduction to the rugarch package (version 1.3-3). Technical report.

[61] Gijbels, I., N. Veraverbeke, and M. Omelka (2011). Conditional copulas, association measures and their applications. *Computational Statistics and Data Analysis 55*, 1919–1932.

[62] GPIF (2013). Review of operations in fiscal 2013. Government Pension Investment Fund, Japan.

[63] Greenshtein, E. and Y. Ritov (2004). Persistence in high-dimensional linear predictor selection and the virtue of overparametriztion. *Bernoulli 10*(6), 971–988.

[64] Haff, I. (2013). Parameter estimation for pair-copula constructions. *Bernoulli 19*, 462–491.

[65] Haff, I., K. Aas, and A. Frigessi (2010). On the simplified pair-copula construction – simply useful or too simplistic. *Journal of Multivariate Analysis 101*, 1296–1310.

[66] Han, D., K. Tan, and C. Weng (2015). Variable selection for index tracking using principal component analysis. Working Paper, University of Waterloo.

[67] Hansen, P. and A. Lunde (2005). A forecast comparison of volatility models: Does anything beat a GARCH(1,1)? *Journal of Applied Econometrics 20*, 873–889.

[68] Harvey, C. and A. Siddique (2000). Conditional skewness in asset pricing tests. *The Journal of Finance 55*(3), 1263–1295.

[69] Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning - Data Mining, Inference, and Prediction, 2nd Ed.* Springer-Verlag New York.

[70] Heller, R., Y. Heller, and M. Gorfine (2013). A consistent multivariate test of association based on ranks of distances. *Biometrika 100*(2), 503–510.

[71] iShares (2015, August). *2015 Prospectus: iShares Russell 2000 ETF — IWM — NYSE ARCA*. BlackRock.

[72] Jensen, M. (1968). The performance of mutual funds in the period 1945-1964. *Journal of Finance 23*(2), 389–416.

[73] Joe, H. (1996). Families of $m$-variate distributions with given margins and $m(m-1)/2$ bivariate dependence parameters. In L. Ruschendorf, B. Schweizer, and M. Taylor (Eds.), *Distributions with Fixed Marginals and Related Topics*. Institute of Mathematical Statistics, Hayward.

[74] Joe, H. (1997). *Multivariate Models and Multivariate Dependence Concepts*. Chapman and Hall/CRC.

[75] Joe, H. and J. Xu (1996). The estimation method of inference functions for margins for multivariate models. Technical report, Technical Report no. 166, Department of Statistics, University of British Columbia.

[76] Jolliffe, I. (1972). Discarding variables in a principal component analysis. i: Artificial data. *Journal of the Royal Statistical Society, Series C (Applied Statistics) 21*(2), 160–173.

[77] Jolliffe, I. (1973). Discarding variables in a principal component analysis. ii: Real data. *Journal of the Royal Statistical Society, Series C (Applied Statistics) 22*(1), 21–31.

[78] Jondeau, E. and M. Rockinger (2006). The copula-garch model of conditional dependencies: An international stock market application. *Journal of International Money and Finance 25*, 827–853.

[79] Kass, R. and A. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*(430), 773–795.

[80] Keim, D. and A. Madhavan (1997). Transactions costs and investment style: an inter-exchange analysis of institutional equity trades. *Journal of Financial Economics 46*, 265–292.

[81] Kraus, A. and R. Litzenberger (1976). Skewness preference and the valuation of risk assets. *The Journal of Finance 31*(4), 1085–1100.

[82] Krink, T., S. Mittnik, and S. Paterlini (2009). Differential evolution and combinatorial search for constrained index-tracking. *Annals of Operations Research 172*, 153–176.

[83] Krupskii, P. and H. Joe (2013). Factor copula models for multivariate data. *Journal of Multivariate Analysis 120*, 85–101.

[84] Kurowicka, D. and R. Cooke (2006). *Uncertainty Analysis with High Dimensional Dependence Modelling*. Chichester: John Wiley.

[85] Lang, L., E. Ofek, and R. Stulz (1996). Leverage, investment, and firm growth. *Journal of Financial Economics 40*, 3–29.

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

[87] Li, G. and H. Rabitz (2014). Analytical HDMR formulas for functions expressed as quadratic polynomials with a multivariate normal distribution. *Journal of Mathematical Chemistry 52*(8), 2052–2073.

[88] Lobo, M., M. Fazel, and S. Boyd (2007). Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research 152*(1), 341–365.

[89] Maginn, J., D. Tuttle, and D. McLeavey (2007). *Managing Investment Portfolios: A Dynamic Process, 3rd. edition*. Wiley.

[90] Mazumder, R. and P. Radchenko (2016). The discrete Dantzig selector: Estimating sparse linear models via mixed integer linear optimization. Working Paper.

[91] Dißmann, J., E. Brechmann, C. Czado, and D. Kurowicka (2013). Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics and Data Analysis 59*, 52–69.

[92] Mitchell, H. and M. Mckenzie (2003). Garch model selection criteria. *Quantitative Finance 3*, 262–284.

[93] Myers, S. (1977). Determinants of corporate borrowing. *Journal of Financial Economics 5*, 147–175.

[94] Nelsen, R. B. (2006). *An Introduction to Copulas, 2nd Edition*. New York: Springer.

[95] Nikoloulopoulos, A., H. Joe, and H. Li (2012). Vine copulas with asymmetric tail dependence and applications to financial return data. *Computational Statistics and Data Analysis 56*, 3659–3673.

[96] Panagiotelis, A., C. Czado, and H. Joe (2012). Pair copula constructions for multivariate discrete data. *Journal of the American Statistical Association 107*, 1063–1072.

[97] Patton, A. (2006). Modelling asymmetric exchange rate dependence. *International Economic Review 47*(2), 527–556.

[98] Pötscher, B. (2009). On the distribution of penalized maximum likelihood etimators: The lasso, scad, and thresholding. *Journal of Multivariate Analysis 100*, 2065–2082.

[99] Rabitz, H. and O. F. Alis (1999). General foundations of high-dimensional model representations. *Journal of Mathematical Chemistry 25*, 197–233.

[100] Ramsay, J., J. Berge, and G. Styan (1984). Matrix correlation. *Psychometrika 49*(3), 403–423.

[101] Rockafellar, T. R. and S. Uryasev (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance 26*, 1443–1471.

[102] Roll, R. (1992). A mean/variance analysis of tracking error. *Journal of Portfolio Management 18*(4), 13–22.

[103] Ross, S. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory 13*, 341–360.

[104] Rothman, A. (2012). Positive definite estimators of large covariance matrices. *Biometrika 99*(3), 733–740.

[105] Ruiz-Torrubiano, R. and A. Suarez (2009). A hybrid optimization approach to index tracking. *Annals of Operations Research 166*, 57–71.

[106] Salisbury, I. (2010, January). ETF assets top $1 trillion. http://www.wsj.com/articles/SB10001424052748704675104575001043625253702.

[107] Schafer, J. and K. Strimmer (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical applications in Genetics and Molecular Biology 4(1)*. Article 32.

[108] SEC. Fast answers: Exchange-traded funds (ETFs). https://www.sec.gov/answers/etf.htm.

[109] Sharpe, W. F. (1963). A simplified model for portfolio analysis. *Management Science 9*(2), 277–293.

[110] Sharpe, W. F. (1966). Mutual fund performance. *Journal of Business 39*(1), 119–138.

[111] Sharpe, W. F. (1992). Asset allocation: Management style and performance measurement. *The Journal of Portfolio Management 18*(2), 7–19.

[112] Shriber, T. (2015, June). ETFs top the $3 trillion milestone. http://www.etftrends.com/2015/06/etfs-top-the-3-trillion-milestone/.

[113] So, M. and C. Yeung (2014). Vine-copula garch model with dynamic conditional dependence. *Computational Statistics and Data Analysis 76*, 655–671.

[114] Sobol, I. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical Modeling & Computational Experiment 1*(4), 407–414.

[115] SPDJI (2014a). *Dow Jones Industrial Average Historical Divisor Changes*. McGraw Hill Financial.

[116] SPDJI (2014b). *S&P Dow Jones Indices: Index Methodology–Dow Jones Averages Methodology*. McGraw Hill Financial.

[117] SPDJI (2016a). *S&P Dow Jones Indices: Index Mathematics Methodology*. S&P Global.

[118] SPDJI (2016b). *S&P U.S. Indices Methodology*. S&P Global.

[119] SPDR (2015, January). *SPDR S&P 500 ETF Trust 2015 prospectus*. SPDR.

[120] Statman, M. (1987). How many stocks make a diversified portfolio? *Journal of Financial and Quantitative Analysis 22*, 353–363.

[121] Stöber, J., H. Joe, and C. Czado (2013). Simplified pair copula constructions - limitations and extensions. *Journal of Multivariate Analysis 119*, 101–118.

[122] Stöeber, J. and C. Czado (2011). Detecting regime switches in the dependence structure of high dimensional financial data. Forthcoming in Computational Statistics and Data Analysis.

[123] Székely, G. J., M. L. Rizzo, and N. K. Bakirov (2007). Measuring and testing independence by correlation of distances. *Annals of Statistics 35*(6), 2769–2794.

[124] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 58*, 267–288.

[125] Tibshirani, R. (1997). The lasso method for variable selection in the Cox model. *Statistics in Medicine 16*, 385–395.

[126] Tole, T. (1982). You can't diversify without diversifying. *The Journal of Portfolio Management 8*, 5–11.

[127] Wang, H., G. Li, and C. Tsai (2007a). Regression coefficient and autoregressive order shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 69*(1), 63–78.

[128] Wang, H., R. Li, and C. Tsai (2007b). Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika 94*, 553–568.

[129] Wild, R. (2007). *Exchange-Traded Funds For Dummies*. Wiley.

[130] Wu, L., Y. Yang, and H. Liu (2014). Nonnegative-lasso and application in index tracking. *Computational Statistics and Data Analysis 70*, 116–126.

[131] Xu, F., Z. Xu, and H. Xue (2015). Sparse index tracking based on $l_{1/2}$ model and algorithm. Working Paper.

[132] Yanai, H. (1974). Unification of various techniques of multivariate analysis by means of generalized coefficient of determination. *Journal of Behaviormetrics 1*, 45–54.

# APPENDICES

# Appendix A

# GARCH(1,1)-Type Models

## A.1 GARCH model

A general GARCH model consists of three components: conditional mean, conditional variance, and innovation term. A standard GARCH(1,1) model for a discrete time series $\{r_t, t = 0, 1 \ldots\}$ is given by

$$r_t = \mu + \varepsilon_t \,, \tag{A.1}$$

$$\varepsilon_t = \sigma_t z_t \,, \tag{A.2}$$

$$\sigma_t^2 = \omega + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2 \,, \tag{A.3}$$

where $\mu$ is the drift term, $\varepsilon_t$ is the diffusion term, $\sigma_t$ is the volatility at time $t$, and $z_t$ is the innovation at time $t$. Here, $\omega$, $\alpha$ and $\beta$ are model parameters. Popular innovation distributions used in GARCH models for modelling financial data include the normal, Student-$t$, generalized error distribution, skewed normal and skewed Student-$t$ distributions.

A common extension of the conditional mean equation (A.1) is to replace (A.1) by an AutoRegressive-Moving-Average (ARMA) model, leading to the so-called ARMA-GARCH model. The ARMA(1,1) is one of the most popular models for the conditional mean in

modelling financial time series. It admits the following formulation:

$$r_t = \mu + ar_{t-1} + \varepsilon_t + b\varepsilon_{t-1}, \;\; t = 1, 2, \ldots, \tag{A.4}$$

where $a$ and $b$ are parameters. The AutoRegressive process of order one AR(1) is retrieved if we set $a \neq 0$ and $b = 0$, and the Moving-Average of order one MA(1) is obtained if we let $a = 0$ and $b \neq 0$.

One common feature with financial log-return data is called the "leverage effect", which refers to the generally negative correlation between an asset return and its changes of volatility. In other words, a negative shock leads to a higher volatility than a positive shock on average. To capture the leverage effect, some asymmetric conditional variance models are proposed in the literature to replace equation (A.3), where the volatility responds symmetrically to both positive and negative shocks. Three prominent examples are as follows.

- Exponential-GARCH (E-GARCH):

$$\ln(\sigma_t^2) = \omega + \alpha_1 \frac{|\varepsilon_{t-1}| + \gamma_1 \varepsilon_{t-1}}{\sigma_{t-1}} + \beta_1 \ln(\sigma_{t-1}^2).$$

- Glosten-Jagannathan-Runkle GARCH (GJR-GARCH):

$$\sigma_t^2 = \omega + \alpha_1 \varepsilon_{t-1}^2 + \gamma_1 I\left\{\varepsilon_{t-1} < 0\right\} \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2.$$

- Power-GARCH (P-GARCH):

$$\sigma_t^\delta = \omega + \alpha_1 (|\varepsilon_{t-1}| - \gamma_1 \varepsilon_{t-1})^\delta + \beta_1 \sigma_{t-1}^\delta, \;\; \delta > 0.$$

More details about these generalized GARCH models can be found in [60].

## A.2 Transformed standardized residuals

Following [14, section 4.1], the transformed standardized residuals (TSRs) are obtained by filtering and normalizing the original data with the model estimation for a time series. We take the standard GARCH(1,1) as an example and the TSRs can be obtained in a similar way for the other models. Let $\hat{\mu}$, $\hat{\omega}$, $\hat{\alpha}$ and $\hat{\beta}$ denote the estimators for GARCH(1,1) parameters fitted with time series data $\{r_t, t \geq 0\}$. We first calculate residuals $\hat{\varepsilon}_t = r_t - \hat{\mu}$, and estimate volatilities

$$\hat{\sigma}_t = \sqrt{\hat{\omega} + \hat{\alpha}\hat{\epsilon}_{t-1}^2 + \hat{\beta}\hat{\sigma}_{t-1}^2}, \quad t = 1, 2, \ldots,$$

where $\hat{\epsilon}_0 = 0$ and $\hat{\sigma}_0$ is equal to the standard deviation of squared sample residuals. The standardized residuals are subsequently obtained as

$$\hat{z}_t = \frac{\hat{\epsilon}_t}{\hat{\sigma}_t}, \quad t = 0, 1, \ldots.$$

In the end, the TSRs are obtained as the values of the EDF of $\hat{z}_t$ evaluated at each point of $\{\hat{z}_t, t \geq 0\}$.