Optimization Methods for Financial Index Tracking: From Theory to Practice

Other titles in Foundations and Trends® in Optimization

The Many Faces of Degeneracy in Conic Optimization
Dmitriy Drusvyatskiy and Henry Wolkowicz
ISBN: 978-1-68083-390-4

Multi-Period Trading via Convex Optimization
Stephen Boyd, Enzo Busseti, Steve Diamond, Ronald N. Kahn, Kwangmoo Koh, Peter Nystrup and Jan Speth
ISBN: 978-1-68083-298-3

Introduction to Online Convex Optimization
Elad Hazan
ISBN: 978-1-68083-170-2

Low-Rank Semidefinite Programming: Theory and Applications
Alex Lemon, Anthony Man-Cho So and Yinyu Ye
ISBN: 978-1-68083-136-8

Optimization Methods for Financial Index Tracking: From Theory to Practice

Konstantinos Benidis
Dept. of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong
kbenidis@connect.ust.hk

Yiyong Feng
Dept. of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong
yiyong@connect.ust.hk

Daniel P. Palomar
Dept. of Electronic and Computer Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong
palomar@ust.hk

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Optimization

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

K. Benidis and Y. Feng and D. P. Palomar. *Optimization Methods for Financial Index Tracking: From Theory to Practice*. Foundations and Trends® in Optimization, vol. 3, no. 3, pp. 171–279, 2018.

ISBN: 978-1-68083-465-9

Foundations and Trends® in Optimization
Volume 3, Issue 3, 2018

Editorial Board

Editorial Scope

Topics

Foundations and Trends® in Optimization publishes survey and tutorial articles in the following topics:

- algorithm design, analysis, and implementation (especially, on modern computing platforms
- models and modeling systems, new optimization formulations for practical problems
- applications of optimization in machine learning, statistics, and data analysis, signal and image processing, computational economics and finance, engineering design, scheduling and resource allocation, and other areas

Information for Librarians

Foundations and Trends® in Optimization, 2018, Volume 3, 4 issues. ISSN paper version 2167-3888. ISSN online version 2167-3918. Also available as a combined paper and online subscription.

Contents

Optimization Methods for Financial Index Tracking: From Theory to Practice

Konstantinos Benidis[1], Yiyong Feng[2] and Daniel P. Palomar[3]

[1] *The Hong Kong University of Science and Technology; kbenidis@connect.ust.com*
[2] *The Hong Kong University of Science and Technology*[*]*; yiyong@connect.ust.hk*
[3] *The Hong Kong University of Science and Technology; palomar@ust.com*

ABSTRACT

Index tracking is a very popular passive investment strategy. Since an index cannot be traded directly, index tracking refers to the process of creating a portfolio that approximates its performance. A straightforward way to do that is to purchase all the assets that compose an index in appropriate quantities. However, to simplify the execution, avoid small and illiquid positions, and large transaction costs, it is desired that the tracking portfolio consists of a small number of assets, i.e., we wish to create a sparse portfolio.

Although index tracking is driven from the financial industry, it is in fact a pure signal processing problem: a regression of the financial historical data subject to some portfolio constraints with some caveats and particularities. Furthermore,

[*]Dr. Yiyong Feng is currently with Three Stones Capital Limited. Dr. Feng was involved in this work when he was with the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology.

Konstantinos Benidis, Yiyong Feng and Daniel P. Palomar (2018), "Optimization Methods for Financial Index Tracking: From Theory to Practice", Foundations and Trends® in Optimization: Vol. 3, No. 3, pp 171–279. DOI: 10.1561/2400000021.

the sparse index tracking problem is similar to many sparsity formulations in the signal processing area in the sense that it is a regression problem with some sparsity requirements. In its original form, sparse index tracking can be formulated as a combinatorial optimization problem. A commonly used approach is to use mixed-integer programming (MIP) to solve small sized problems. Nevertheless, MIP solvers are not applicable for high-dimensional problems since the running time can be prohibiting for practical use.

The goal of this monograph is to provide an in-depth overview of the index tracking problem and analyze all the caveats and practical issues an investor might have, such as the frequent rebalancing of weights, the changes in the index composition, the transaction costs, etc. Furthermore, a unified framework for a large variety of sparse index tracking formulations is provided. The derived algorithms are very attractive for practical use since they provide efficient tracking portfolios orders of magnitude faster than MIP solvers.

1

Introduction

1.1 What Is a Financial Index?

An index is a number that represents the aggregate value of a group of items. In particular, a financial index is composed of a collection of assets, such as stocks or bonds, which captures the value of a specific market or a segment of it. A stock or a bond market index is effectively equivalent to a hypothetical portfolio of assets in the sense that we cannot invest directly on it, i.e., an index is not a financial instrument that we can trade. In Section 1.3 we will analyze the various ways we can gain practical access to an index.

The value of a financial index depends on all the underlying assets that compose it. However, the significance of each asset, or in other words its relative weight in the index, is different. There are two basic types of financial indices:

1. Capitalization-weighted (cap-weighted): the assets are weighted based on the ratio of their capitalization[1] to the overall capitalization of the assets that compose the index. The index value is

[1] Capitalization refers to the number of outstanding shares multiplied by share price.

Table 1.1: Example of composition of index.

Stock	Shares	Price	Cap.	Price-weighted	Cap-weighted
1	100	\$20	\$2,000	0.25	0.4
2	50	\$60	\$3,000	0.75	0.6
Total:		\$80	\$5,000	1.0	1.0

proportional to the weighted average of the capitalization of the underlying assets.

2. Price-weighted: the assets are weighted based on the ratio of their price to the sum of all of the prices of the assets that compose the index. The index value is proportional to the weighted average of the prices of the underlying assets.

To clarify the above, consider a simple example where an index is composed by only two stocks, as shown in Table 1.1. Observe that the weights of the two stocks can be very different depending on the type of index. The value of a price-weighted index would be proportional to $20 \times 0.25 + 60 \times 0.75 = 50$ and of a cap-weighted index to $2{,}000 \times 0.4 + 3{,}000 \times 0.6 = 2{,}600$.

In practice, when an index is introduced it is common to set its value to a round number such as 100 or 1,000. Therefore, the actual value of the index has to be divided by a number which is known as the index divisor. Going back to our example, if this was the first day of the index a possible divisor for the price-weighted and cap-weighted version could be \$0.5 and \$2.6, giving an initial value of 100 and 1,000 points, respectively.

Although the cap-weighted and the price-weighted indices are the most common types, the are several other variations of weighted indices. For example, an index can be equal-weighted (or unweighted), where all the assets have exactly the same importance, or volume-weighted, where the weight is based on the traded volume of the assets during some period. Another example is the Tokyo Stock Price Index (TOPIX), which transitioned from a weighting system based on the outstanding shares of each company to a weighting system based on the shares available for trading (free float).

Table 1.2: List of well known indices and their type.

Index	Type
Standard & Poor's 500 (S&P 500)	cap-weighted
Dow Jones Industrial Average (DJIA)	price-weighted
NASDAQ Composite	cap-weighted
Hang Seng Index (HSI)	cap-weighted
Financial Times Stock Exchange 100 (FTSE 100)	cap-weighted
Russell 2000	cap-weighted

In order for an index to be consistent over time it should be adjusted to capture corporate actions that affect market capitalization, such as additional share issuance, dividends and restructuring events such as mergers or spin-offs. Additionally, to remain indicative of the market that the index represents, the underlying assets that compose the index change frequently. To prevent all these corporate actions and changes in the index composition from affecting its value, the divisor of an index is adjusted appropriately so its value remains constant.

A list of well known indices is presented in Table 1.2. Since most of the major indices are cap-weighted (with the most important exception being the Dow Jones Industrial Average (DJIA) index), in all the numerical experiments we will mainly focus on the cap-weighted type, however, the algorithms in principle work for any type of index.

1.2 Why Track an Index?

Fund managers follow two basic investment strategies: active and passive. In active management strategies, the fund managers assume that the markets are not perfectly efficient and through their expertise and superior prediction methods they hope to add value by choosing high performing assets. On the contrary, the passive management strategies are based on the assumption that the market cannot be beaten in the long run. The passive managers have less flexibility and their role is to conform to a closely defined set of criteria.

Analysis of historical data has shown that the majority of the actively managed funds do not outperform the market in the long run [9, 62].

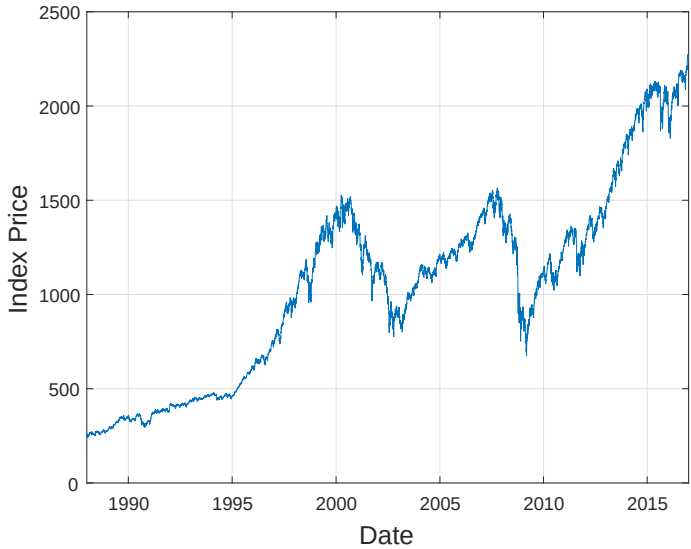
Furthermore, the stock markets have historically risen and therefore reasonable returns can be obtained without the active management's risk. These reasons have prompted the investor's interest into more passive management strategies. Index tracking, also known as index replication, is one of the most popular passive portfolio management strategies. It refers to the problem of reproducing the performance of a market index.

Apart from the direct gains that an investor could have by tracking an index, the index based exchange traded funds (ETFs), which are effectively index tracking portfolios (see Section 1.3.2), have been used widely for hedging purposes [2, 40, 3]. That is, an investor tries to minimize the risk of an investment with an index that is correlated to that investment by taking an appropriate long or short position on the index.

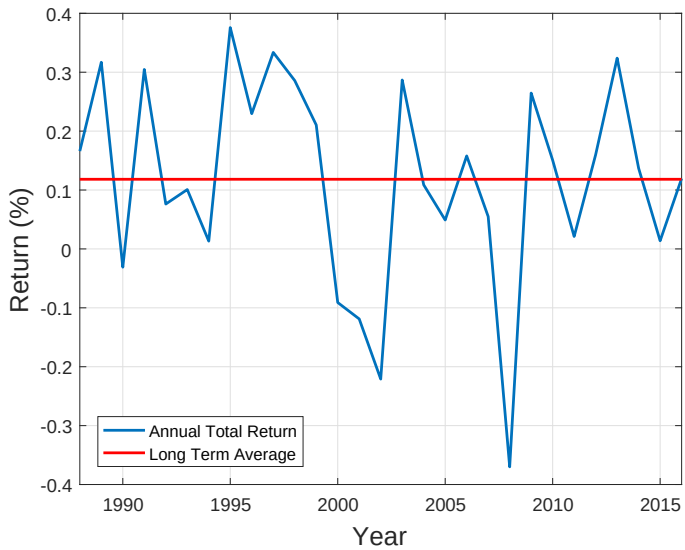
The Standard & Poor's 500 (S&P 500) is one of the world's best known (cap-weighted) indices and one of the most commonly used benchmarks for the stock market. To this end, we will use S&P 500 for illustration purposes throughout this monograph. Figure 1.1 illustrates the performance of S&P 500 over the last three decades. In Figure 1.1(b) we observe that the average annual return of the index is approximately 11%. This is a great average return considering the fact that it includes the 2000-2002 dot-com bubble and the severe crisis of 2008 where the market lost 37% of its value. Of course, an investor that was tracking this index would not have on average an 11% profit per year. This is due to various reasons such as trading costs, inflation, etc. Nevertheless, a reasonable return could be achieved by following this passive strategy.

1.3 Index Tracking

As we have already mentioned, it is not possible to trade an index directly. In order to gain access to an index we need to use other financial instruments such as options, futures, and exchange traded funds (ETFs), or create a portfolio of assets that tracks closely a given index.



(a) Price of S&P 500.



(b) Annual returns of S&P 500.

Figure 1.1: Performance of the index S&P 500 for the period 1988 - 2016.

1.3.1 Options and Futures

An option is a financial derivative since its value is linked to the price of something else. The holder of an option contract has the right, but not the obligation, to buy or sell an underlying asset at a set price on (e.g., European option) or before (e.g., American option) the expiration date of the option. Of particular interest are the index options, which give the right to buy or sell the value of an underlying index. However, note that index options are always cash settled, i.e., no actual stocks are bought or sold. Index options can be used to gain profit from general index movements or for hedging risks in a portfolio. There have been many works on the pricing of index option contracts and on their volatility estimation, for example see [32, 27, 26, 24].

An index future contract is a financial derivative that gives the holder the obligation to purchase an index at a particular price on a specified date in the future. If on that specified date the price of the index has surpassed the price that is agreed in the contract, then the holder makes a profit, and the seller suffers a loss. Futures differ from options in that a futures contract is considered an obligation, while an option is considered a right that may or may not be exercised. Index futures are a very popular way of investing in an index and many works have focus on analyzing their pricing and their relationship to the underlying index [28, 69, 46, 7, 76].

Both index options and index futures are derivative products that do not track the value of an index explicitly but rather their value is associated to the index value.

1.3.2 ETFs

Another popular way to engage in index tracking is to purchase an exchange traded fund (ETF). An ETF is like a stock but its value tracks closely a given index, e.g., see SPDR[2]. It is constructed either by using derivative products, leading to synthetic ETFs, or the underlying components of the index, leading to physical ETFs. Many physical ETFs

[2]SPDR funds are a family of ETFs. The name is an acronym for the first member of the family, the Standard & Poor's Depositary Receipts, which was later renamed to SPDR S&P 500 (ticker SPY).

use all the underlying assets of the index they are tracking, e.g., the Standard and Poor's Depositary Receipts (ticker SPY) based on the S&P 500 and the Nasdaq 100 Trust Shares (ticker QQQ) based on the Nasdaq 100. However, there are also many ETFs using a sparse construction, where representative sampling, with 80-95% of the underlying securities being used, or aggressive sampling, with only a tiny percentage being used [37, 64].

An ETF, unlike options and futures, tracks the value of an index explicitly.

1.3.3 Tracking Portfolios

Finally, we can again track the value of an index explicitly by constructing a portfolio of assets or derivatives whose value follows the value of the given index. The construction of such a tracking portfolio is important for several reasons. First, it is the building block of an ETF, i.e., in order to issue an ETF we need first to construct the corresponding portfolio that this ETF will represent. In addition, not all indices or market sectors have an ETF associated with them. Therefore, a tracking portfolio can be used to explicitly track an index where an ETF does not exist. Finally, having the tools to create such portfolios gives us the flexibility to include any partial information that we may possess or even create portfolios that try to beat the value of an index, instead of using some predetermined financial instruments that we have no freedom on adjusting.

Now, let us introduce some notation that we will use extensively throughout the monograph. Assume that an index is composed of N assets. We denote by $\mathbf{r}^b = [r_1^b, \ldots, r_T^b]^\top \in \mathbb{R}^T$ and $\mathbf{X} = [\mathbf{r}_1, \ldots, \mathbf{r}_T]^\top \in \mathbb{R}^{T \times N}$ the (arithmetic) net returns of the index and the N assets in the past T days, respectively, with $\mathbf{r}_t \in \mathbb{R}^N$ denoting the net returns of the N assets at the t-th day. Further, $\mathbf{b}_t \in \mathbb{R}^N_{++}$ denotes the normalized benchmark index weights at the t-th day, such that $\mathbf{b}_t^\top \mathbf{1} = 1$ and $\mathbf{r}_t^\top \mathbf{b}_t = r_t^b$. The prices of the assets at the t-th day are denoted by $\mathbf{p}_t = [p_{t1}, \ldots, p_{tN}]^\top$ and the number of shares of each asset as $\mathbf{n}_t = [n_{t1}, \ldots, n_{tN}]^\top$. The designed portfolio is denoted by $\mathbf{w}_t \in \mathbb{R}^N_+$, with $\mathbf{w}_t^\top \mathbf{1} = 1$.

Full Replication

The most straightforward manner to create a tracking portfolio $\mathbf{w}_t \in \mathbb{R}^N$ is by buying appropriate quantities of all the assets that compose the index, i.e., by choosing $\mathbf{w}_t = \mathbf{b}_t$. This technique is known as full replication and it requires that the true index construction weights $\mathbf{b}_t$ are available. Following this approach, a perfect tracking can be achieved.

The full replication technique has several drawbacks. First, the execution of such a portfolio may be involved since it may consist of thousands of stocks. Second, a portfolio consisting of all the assets may incorporate too many small and illiquid stocks. This translates into higher risk to investors since an illiquid stock is hard to sell if we are looking to exit and moreover it increases the costs due to slippage. Furthermore, allocating capital to all the assets increases significantly the commission fees since every asset is associated with a separate transaction. These drawbacks become more severe as we increase the rebalancing frequency of our tracking portfolio. Finally, the benchmark portfolio weight vector $\mathbf{b}_t$ and all its changes (the benchmark weight vector is consistently rebalanced by the indices providers) can be very expensive to obtain. For example, in 2006 the index sponsors S&P, Dow Jones, MSCI, and FTSE earned total revenues of $1.66 billion from the ETF providers and therefore the ETF providers were even thinking of cutting these costs by setting up their own market indices[3].

Sparse Index Tracking

A natural way to deal with the problems caused by the full replication is to use a small number of assets to (approximately) replicate an index. This leads to the construction of a sparse[4] index tracking portfolio [45, 10]. A sparse portfolio simplifies the execution of the portfolio and tends to avoid illiquid stocks that usually correspond to the assets with small weights in an index, since in a sparse setting most of these assets are

[3] See "ETF providers float idea of setting up their own market indices" published in Financial Times on 2017-05-24.

[4] If we use only a small number of assets, only a small number of weights will be nonzero, i.e., the portfolio will be sparse.

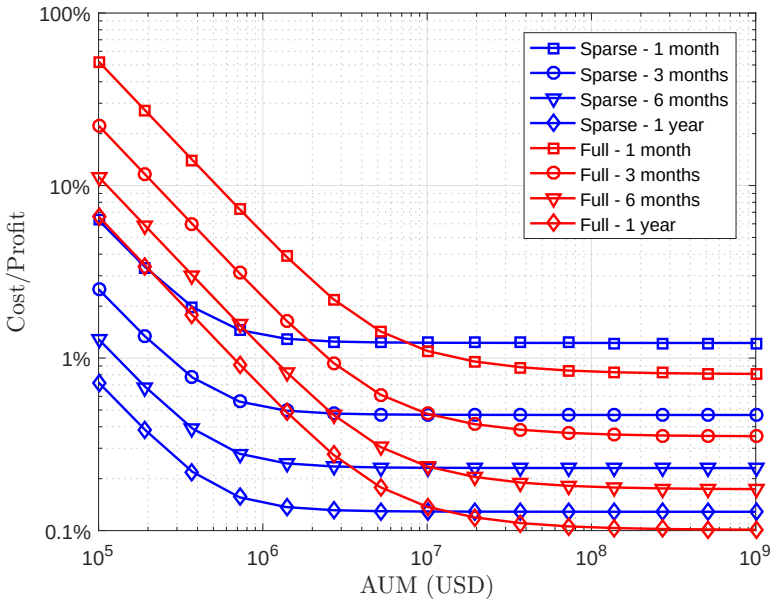


Figure 1.2: Commission fees as a percentage of profit for sparse (40 assets) and full portfolios for different AUM and rebalancing frequencies.

discarded. Furthermore, since only a small number of assets is used, the transaction costs are reduced significantly due to the reduction of the fixed (minimum) costs in the commission fees.

Now, in order to verify the advantages of sparse portfolios, let us get a more quantitative idea on the commission fees reduction that we can achieve. Consider two tracking portfolios of the index S&P 500 for a 5-year period (2011-2015): the first is sparse[5], composed of only 40 assets, while the second uses all the (approximately) 500 assets of the index. Figure 1.2 illustrates the commission fees of the two portfolios as a percentage of the profit, for a range of rebalancing frequencies and assets under management (AUM)[6]. For simplicity we do not have any leverage (i.e., leverage = 1).

[5]This portfolio was constructed using the algorithms presented in Chapter 4.

[6]AUM, also known as net asset value (NAV), measures the total market value of all the financial assets that a financial institution manages.

The first thing to observe is that there is a threshold AUM (around \$10 million) where the commission fees become equal for the two portfolios. For smaller AUM the sparse portfolio has significantly less fees while for larger AUM the full portfolio has slightly smaller fees. Although it is not straightforward why we get such a behavior, it has a simple explanation. Note that the commission fees of each transaction (given by (2.4)) depend on the number of shares Δn we buy or sell, with a fixed minimum fee. For small AUM, the number of shares Δn is not that large and therefore the fixed fee is dominating. For a sparse portfolio we need to pay this fixed fee only for a few assets whereas for a full portfolio this cost becomes significant. As the AUM grows, the fixed cost becomes less important since it is dominated by the large amount of shares we trade. Therefore, the difference of the commission fees between the two portfolios becomes more narrow. Finally, for large AUM the full portfolio has less commission fees. This is because the rebalancing of a sparse portfolio can result in a different composition of assets. This means we need to sell all the holding shares of one asset and buy many shares of another whereas for full portfolios we do not need to make such severe changes. Of course, the change in the asset composition in sparse portfolios is true for lower AUM as well. However, this effect is mitigated in lower AUM as the fixed costs are dominant.

In general, the change in the composition of the holding portfolio during rebalancing can be controlled by including a turnover penalty as we will see in Section 3.5. To illustrate the benefit of the turnover penalty, we consider the same setting as in Figure 1.2 and we focus only on the 3-month rebalancing frequency. Apart from the sparse and full portfolios, we design one more sparse portfolio that has a turnover penalty, i.e., we penalize the changes in the portfolio after rebalancing. In Figure 1.3 we observe how this penalization reduces further the transaction costs and makes the sparse portfolio with turnover penalty more cost effective that the full replication portfolio even for large AUM.

As we will see in Section 4.6, the turnover penalization can be controlled by a tuning parameter. However, we should keep in mind that there is a tradeoff between the reduction of costs and the tracking error, i.e., by enforcing only small changes in a portfolio during rebalancing can lead to a larger tracking error.

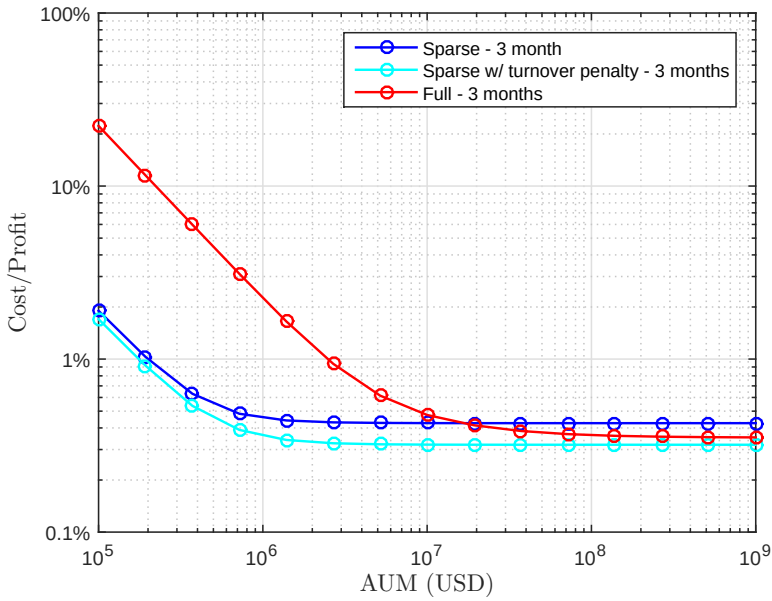


Figure 1.3: Commission fees as a percentage of profit for sparse (40 assets), sparse with turnover penalty, and full portfolios for different AUM and a rebalancing frequency of 3 months.

1.4 Goal

Due to the importance of sparse tracking portfolios, our main focus will be on deriving such portfolios with the goal of tracking an index as efficiently as possible. It is worth mentioning that in general index tracking portfolios are not efficient in the sense that they do not lie on the efficient frontier as defined by Markowitz [53]. This is expected since the goal of a sparse tracking portfolio algorithm is to find an optimal tradeoff between tracking error and sparsity and not between return and risk.

Further, the analysis and the algorithms derived in the monograph assume decisions for a single-period, i.e., the portfolio derivations do not incorporate information about future trades. For multi-period trading please refer to [16] and references therein.

Finally, although index tracking is not a real-time application, the construction time should be reasonable given the fact that extensive backtesting should be made before deploying an index tracking strategy, which requires the construction of many portfolios for a given algorithm.

1.5 Outline

The abbreviations and the notation used throughout the monograph are provided on pages 89 and 90, respectively.

In Chapter 2 we present two basic challenges we face when we engage in index tracking, namely the need for a frequent rebalancing of a tracking portfolio due to the constant changes in an index, and the transaction costs that are associated with a portfolio. As we will see, these two challenges form a natural tradeoff.

In Chapter 3 we introduce the sparse index tracking problem in its general form and we discuss the various tracking error functions and possible constraints that one could impose. We further analyze existing methods that produce sparse tracking portfolios and their drawbacks.

In Chapter 4 we derive algorithms for the sparse index tracking problem. We consider various tracking error functions and constraints. All of the possible problem variations boil down to the same effective problem that we need to solve iteratively until the algorithms converge.

In Chapter 5 we provide numerical experiments that show the performance of the derived algorithms. For illustration purposes we use the indices S&P 500 and Russell 2000.

Finally, Chapter 6 concludes the monograph.

1.6 Software

Many of the derived algorithms can be found in the R [59] software package ***sparseIndexTracking*** [12], which is available in CRAN.

2

Challenges in Index Tracking

2.1 Rebalancing Frequency

When we design an index tracking portfolio, an important issue for consideration is that the composition of an index changes frequently due to corporate actions and restructuring events, or due to underperforming stocks that are replaced in the index. For example, on average around 22 companies, or 4.4%, are added to or removed from the index S&P 500 each year. Therefore a tracking portfolio can become obsolete after a period of time if it does not take into account all of these changes.

Furthermore, the prices of all the assets, and therefore their capitalization, change constantly due to daily trading. Based on these changes, at the t-th day, the index weights $\mathbf{b}_t$ are automatically changed according to the returns of the previous day, i.e., for the j-th asset we get

$$b_{tj} = \frac{p_{tj} n_{tj}}{\mathbf{p}_t^\top \mathbf{n}_t} = \frac{p_{(t-1)j}(1 + r_{(t-1)j}) n_{(t-1)j}}{(\mathbf{p}_{t-1} \odot (\mathbf{1} + \mathbf{r}_{t-1}))^\top \mathbf{n}_{t-1}}, \tag{2.1}$$

where $\mathbf{p}_{t-1}$ and $\mathbf{p}_t$ are the opening prices of the assets at the $(t-1)$-th and t-th day, respectively. Equivalently, $\mathbf{n}_{t-1}$ and $\mathbf{n}_t$ are the number of shares of the assets[1], and $\mathbf{r}_{t-1}$ are the net returns of the $(t-1)$-th day.

[1] Depending on the index, the number of outstanding shares of each asset is usually readjusted quarterly, semi-annualy, or annualy.

Now, at the t-th day we construct an optimal[2] tracking portfolio $\mathbf{w}_t$ without getting into details on how we achieve this at the moment. The change in prices will affect the tracking portfolio $\mathbf{w}_t$ in the same manner as the hypothetical index portfolio $\mathbf{b}_t$. These changes in the weights are not a potential issue only if we use full replication, i.e., when $\mathbf{w}_t = \mathbf{b}_t$, where our portfolio is identical to the index (see Section 1.3.3). In any other case, where our portfolio is not 1-to-1 with the index, we do not wish to follow the temporal weight changes but rather keep the relative weights in our portfolio close to the computed optimal value.

In order to compensate for all of the index and price changes we need to *rebalance* our portfolio frequently. Choosing the optimal frequency is not an easy task and many factors (e.g., transaction costs, volatility, type of investment, etc.) need to be taken into account [51, 48, 4]. One of these parameters is the transaction cost, which we will analyze in the next section.

Finally, an important thing to note is that when we design a portfolio with the goal of tracking a financial index we use historical data in order to find an optimal portfolio that minimizes some appropriate tracking error. However, since the dynamics of the market constantly change, it is not wise to use data from the distant past but rather only from a recent period. In other words, historical data from the distant past would give us a significantly different optimal portfolio than recent data. This further signifies the importance of holding a tracking portfolio for a limited amount of time, where the market dynamics are relatively constant.

2.2 Transaction Costs

Up to this point we have deliberately omitted an important aspect of every financial transaction, namely the transaction costs, that could affect the overall return of our investment. Over the last decades there have been various works that incorporated the transaction costs in the optimization framework of a financial portfolio, e.g., see [58, 47, 50].

[2]Optimal in the sense that it minimizes some appropriate error function based on historical data.

The transaction costs consist of two terms, namely the commission fee and the slippage. The commission fee is effectively the payment we make to our broker for completing the transaction. There is not a universal scheme for commission fees. It differs based on the country and the specific broker. For example, U.S. has in general less commission fees than European and Asian countries. Slippage refers to the difference between the expected price of a trade and the price at which the trade is actually executed. It does not directly refer to a negative or positive movement, as any change between the expected and actual prices can qualify. In general, liquid assets have smaller slippage than illiquid assets.

The procedure of calculating the transaction costs and including them in the total wealth of our portfolio is straightforward. However, we will briefly describe it for clarity. Let us first give some definitions.

We denote by $\mathbf{w}_t \in \mathbb{R}^N$, V_t, and $r_t^w = \mathbf{w}_t^\top \mathbf{r}_t$, the holding portfolio, the wealth (budget), and the net return of the portfolio at the t-th day, respectively. Further, we denote by $c_t(\mathbf{w}_t, \mathbf{w}_{t-1})$ the transaction costs which depend on the current (targeted) and previous (holding) portfolio.

The wealth V_t at the t-th day can be calculated as:

$$V_t = V_{t-1}(1 + r_t^w) - c_t(\mathbf{w}_t, \mathbf{w}_{t-1}). \tag{2.2}$$

We can interpret this equation as follows: the wealth at the t-th day equals the wealth of the previous day multiplied by the gross return minus the costs. The transaction costs term c_t is zero if we do not rebalance the holding portfolio. We can set the initial portfolio to zero since we do not hold any assets initially, i.e., $\mathbf{w}_0 = \mathbf{0}$.

Now, let us calculate the total transaction costs. We begin by calculating the commission fees. Since we have selected the index S&P 500 as a benchmark, we use the following representative model[3] for the commission fees which is applicable in the U.S. markets:

- Every transaction is associated with only one asset and the costs apply to both selling and buying the asset.

[3] https://www.interactivebrokers.com/en/index.php?f=commission&p=stocks

- Cost per transaction: $\$0.005 \cdot n$.
- Minimum cost: \$1.
- Maximum cost: 0.5% of trade volume, which is defined as $\text{vol} = n \cdot p$.

If we decide to rebalance our portfolio, then we need to calculate the shares of each asset that we need to buy or sell in order to achieve the targeted portfolio. For the j-th asset, the shares difference[4] will be:

$$\Delta n_{tj} = |n_{tj} - n_{(t-1)j}| = \left| \frac{w_{tj} V_t}{p_{tj}} - \frac{w_{(t-1)j} V_{t-1}}{p_{(t-1)j}} \right|. \tag{2.3}$$

Based on this difference, we can calculate the commission fees from the model described above. In particular, the commission fee for the j-th asset can be compactly written as:

$$c_{tj}^{\text{cf}} \left(w_{tj}, w_{(t-1)j} \right) = \mathcal{I}_{\{\Delta n_{tj} > 0\}} \cdot \max \left(1, 0.005 \cdot \Delta n_{tj} \cdot \min \left(p_{tj}, 1 \right) \right), \tag{2.4}$$

where

$$\mathcal{I}_{\{\Delta n_{tj} > 0\}} = \begin{cases} 1, & \text{if } \Delta n_{tj} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

Accumulating this cost for all the assets will give the total cost due to commissions fees:

$$c_t^{\text{cf}} \left(\mathbf{w}_t, \mathbf{w}_{t-1} \right) = \sum_{j=1}^{N} c_{tj}^{\text{cf}} \left(w_{tj}, w_{(t-1)j} \right). \tag{2.6}$$

Figure 2.1 illustrates how the commission fees (per asset) change for a range of prices p and shares difference Δn.

Now, let us consider the costs due to slippage. In practice, slippage can be estimated from the bid-ask spread of an asset. Although one can find various models to estimate slippage, a simple one is the following:

$$\text{slippage} = \frac{\text{half of spread}}{\text{middle of spread}} = \frac{0.5(\text{bid - ask})}{0.5(\text{bid + ask})} = \frac{\text{bid - ask}}{\text{bid + ask}}. \tag{2.7}$$

[4]The number of shares should be a multiple of the lot size of a stock but we overlook this to simplify the problem.

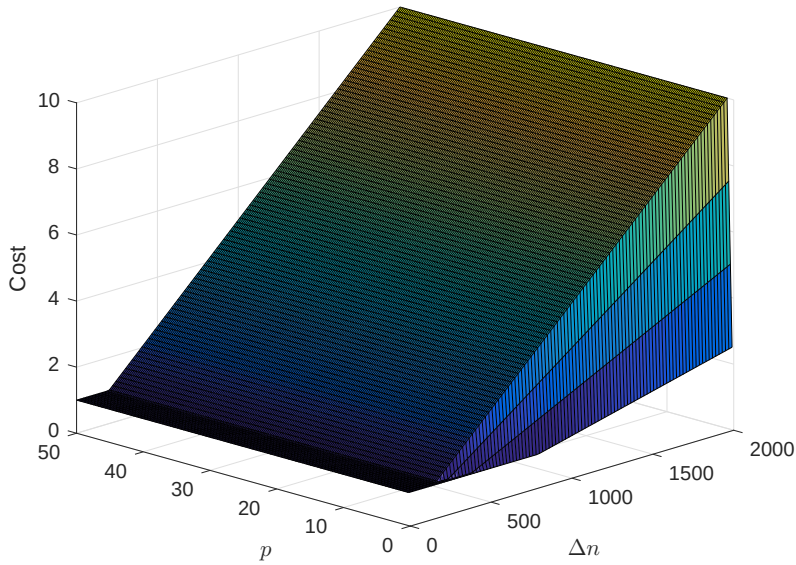


Figure 2.1: Commission fees as a function of Δn and p.

Then, the cost due to slippage for the j-th asset at the t-th day can be computed as:

$$\begin{aligned} c_{tj}^{\mathrm{sl}}\left(w_{tj}, w_{(t-1)j}\right) &= \mathrm{slippage}_{tj} \cdot \mathrm{turnover}_{tj} \\ &= \mathrm{slippage}_{tj} \cdot \Delta n_{tj} p_{tj}. \end{aligned} \tag{2.8}$$

For the total cost due to slippage we need to sum over all the assets:

$$c_t^{\mathrm{sl}}\left(\mathbf{w}_t, \mathbf{w}_{t-1}\right) = \sum_{j=1}^{N} c_{tj}^{\mathrm{sl}}\left(w_{tj}, w_{(t-1)j}\right). \tag{2.9}$$

Finally, we can aggregate the commission fees and slippage costs to obtain the total transaction costs:

$$c_t = c_t^{\mathrm{cf}} + c_t^{\mathrm{sl}}. \tag{2.10}$$

It is obvious that every time we perform a transaction we need to pay the costs for this transaction. This raises the following trade-off:

we should adjust our portfolio frequently in order to have an accurate tracking but not that frequently in order to keep the transaction costs low. As we already showed in Figure 1.2 the transaction costs become important only for small to medium investments. For large investments they become negligible while usually even better deals can be achieved with the brokers.

However, even though larger investments have lower costs, due to their magnitude they impact the market, i.e., they always move the market against them which turns out to be an execution cost as well. Thus, in that point of view, there is still a trade-off between the rebalancing frequency and the associated costs.

2.3 Granularity

When we design a portfolio, the weights correspond to the percentage of the AUM that we should allocate to each asset. However, for small investments granularity can play an important role and make the derived portfolios infeasible to implement since we can only allocate capital to each asset which is an integer multiple of the value of its lot size.

To make this clear, consider a simple example in which we would like to invest \$1000 on a portfolio that consists of two assets. The weights of the designed portfolio are $\mathbf{w} = [0.6, 0.4]^\top$ and the prices per share of each asset are $\mathbf{p} = [\$120, \$150]^\top$. Since the first asset has a weight of 0.6, we should invest $\$1000 \cdot 0.6 = \$600$ in it. This translates to the purchase $\$600/\$120 = 5$ shares, assuming that the lot size is 1 share. In a similar manner, we should allocate \$400 to the second asset which translates to $\$400/\$150 = 2.67$ shares. Obviously, we can either buy 2 or 3 shares of the second asset but not 2.67. However, this would alter significantly the weights of the implemented portfolio compared to the designed one. Also, notice that we cannot allocate exactly \$1000 in these two assets given their share prices.

The granularity restrictions depend on the AUM, the price per share of each asset, and the weights. They effectively vanish for large investments where we can easily create a portfolio very close to the designed one. However, for smaller investments we should always confirm that the designed portfolio is feasible.

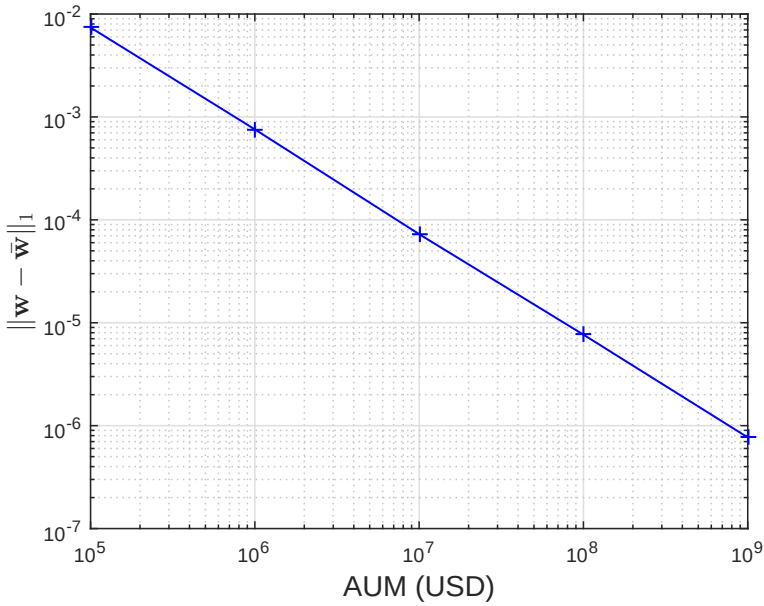


Figure 2.2: Difference of designed portfolios $\mathbf{w}$ and realized portfolios $\bar{\mathbf{w}}$ due to granularity issues.

In Figure 2.2 we illustrate how much a designed portfolio $\mathbf{w}$ diverges from the realized portfolio $\bar{\mathbf{w}}$ due to granularity issues for a range of AUM. For this, we have used historical data of the index S&P 500 and a tracking portfolio of 40 assets, assuming the lot size to be 1 share.

Notice that the granurality error $\frac{\|\mathbf{w}-\bar{\mathbf{w}}\|_1}{\|\mathbf{w}\|_1} = \|\mathbf{w} - \bar{\mathbf{w}}\|_1$ shown in Figure 2.2 is a relative measure of divergence, i.e., divergence in dollars divided by total amount of dollars invested. If instead we want an absolute measure of divergence in units of dollars, then we need to multiply each point by its corresponding AUM and the obtained divergence becomes almost constant.

3

Sparse Portfolios

In the case of full-replication we can achieve perfect tracking by selecting appropriate weights for all the assets that compose the index. However, when we use a small subset of the available assets we need to make some basic design decision, i.e., which quantity we should track, how we should measure the tracking performance, which constraints our portfolio should satisfy, etc.

3.1 Design Simplifications

Before we present all the possible design options, we need first to analyze an implicit simplification during the design process of a portfolio that is necessary to make the problem tractable.

In order to design a tracking portfolio we use historical data of the index and the underlying assets. Effectively, this is a regression problem in which we try to find a portfolio that minimizes some tracking error. In particular, consider an index composed by N assets and denote by $\mathbf{X} \in \mathbb{R}^{T \times N}$ and $\mathbf{r}^b \in \mathbb{R}^T$ the returns of the assets and the index on the last T days, respectively. If the data were perfect, i.e., there was no noise, and $T \geq N$, i.e., the problem was well-defined, we could always perfectly retrieve the true variables in a typical least-squares fitting.

In our case, this would be $\mathbf{w}^\star = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{r}^b$. Furthermore, this optimal portfolio should be equal to the true index weights (assuming that $\text{rank}(\mathbf{X}) = T$), i.e., $\mathbf{w}^\star = \mathbf{b}$. However, this is not true for the index tracking application since the weights change daily due to price changes. That is, we cannot fit perfectly a constant portfolio to a period of time since the true solution is not constant.

Albeit a simplification, in all the portfolio design formulations we will fit a single portfolio during the whole training period in order to make the problem tractable. In the following, we will focus on the design process and therefore we will drop the time index of the constructed portfolios since the portfolio is independent of time. However, when we evaluate a constructed portfolio in the numerical results during a testing period, we should track its daily changes.

3.2 Tracking Quantity

The first decision we need to make is the quantity of the index we wish to track, i.e., the returns, log-returns, prices, or log-prices. In the majority of the literature, the return of the index is the quantity of interest. However, there are some works that focus more on the tracking of log-prices.

To highlight the differences that occur depending on the selected method, we can view the index tracking problem as the comparison of two portfolios: the first is the actual index and the second is the tracking portfolio. At the t-th day (of the training period), we denote the value of the index benchmark portfolio as V_t^b and the value of a tracking portfolio $\mathbf{w}$ as V_t. In order to have a fair comparison of the performance of the two portfolios we must initialize both of them with the same value, i.e., it must hold that $V_0^b = V_0$. Note that the tracking portfolio $\mathbf{w}$ does not have a "time" index, following the design simplifications discussed in Section 3.1.

3.2.1 Returns

First we consider tracking based on the returns. The value of each portfolio at the t-th day is equivalent to the initial budget multiplied

by the gross return of the portfolio. We can calculate these values as follows:

$$V_t^b = V_0^b \prod_{i=1}^{t} \left(1 + r_i^b\right), \tag{3.1}$$

$$V_t = V_0 \prod_{i=1}^{t} \left(1 + \sum_{j=1}^{N} w_j r_{ij}\right). \tag{3.2}$$

Basically, if one can track the returns, then implicitly one is tracking the value of the index. So, the goal is to construct a portfolio that approximates well the returns of the index by combining the returns of the N assets, i.e.,

$$\sum_{j=1}^{N} w_j r_{ij} \approx r_i^b, \quad \text{for } i = 1, \ldots, t. \tag{3.3}$$

3.2.2 Log-Returns

In the case we wish to track log-returns, the log-values of the two portfolios can be computed as follows:

$$\begin{aligned} \log(V_t^b) &= \log(V_0^b) + \sum_{i=1}^{t} \log\left(1 + r_i^b\right) \\ &= \log(V_0^b) + \sum_{i=1}^{t} r_i^{\log,b}, \end{aligned} \tag{3.4}$$

and

$$\begin{aligned} \log(V_t) &= \log(V_0) + \sum_{i=1}^{t} \log\left(1 + \sum_{j=1}^{N} w_j r_{ij}\right) \\ &\approx \log(V_0) + \sum_{i=1}^{t} \left(\sum_{j=1}^{N} w_j \log(1 + r_{ij})\right) \\ &= \log(V_0) + \sum_{i=1}^{t} \left(\sum_{j=1}^{N} w_j r_{ij}^{\log}\right). \end{aligned} \tag{3.5}$$

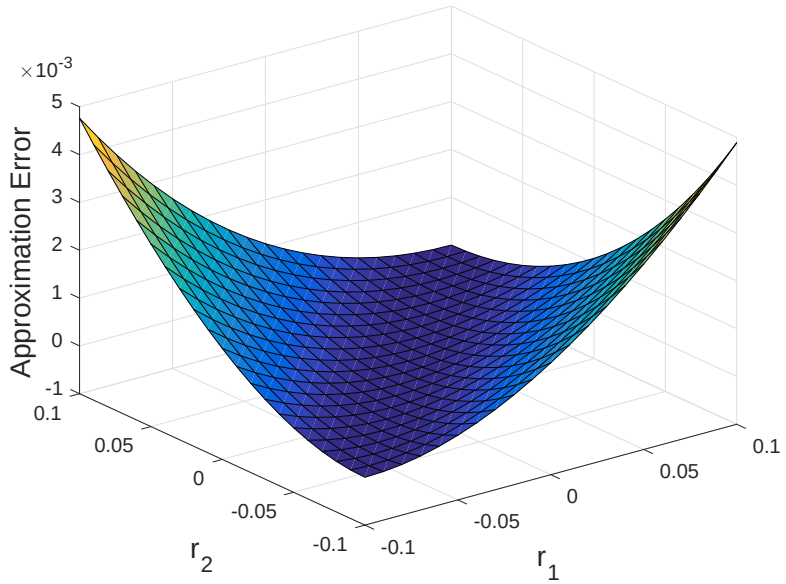


Figure 3.1: Approximation error $\log(1+\mathbf{w}^\top\mathbf{r}) - \mathbf{w}^\top \log(\mathbf{1}+\mathbf{r})$ for $\mathbf{w} = [0.4, 0.6]^\top$ and $\mathbf{r} \in [-0.1, 0.1]^2$.

In (3.5) we have used the following approximation[1]:

$$\log(1+\mathbf{w}^\top\mathbf{r}) \approx \mathbf{w}^\top \log(\mathbf{1}+\mathbf{r}), \tag{3.6}$$

where the $\log(\cdot)$ is applied elementwise. To illustrate this approximation consider a portfolio with two assets where $\mathbf{w} = [0.4, 0.6]^\top$. The error of the above approximation, i.e., $\log(1+\mathbf{w}^\top\mathbf{r}) - \mathbf{w}^\top \log(\mathbf{1}+\mathbf{r})$, is shown in Figure 3.1 for returns in the range of $[-0.1, 0.1]$. It is clear that for small enough returns, the approximation is tight. Therefore, if one tracks the log-returns, then implicitly one is tracking approximately the log-value of the index.

Thus, the goal is to construct a portfolio that approximates well the log-returns of the index by combining the log-returns of the N assets,

[1]This approximation is based on the fact that $\log(1+x) \approx x$ for small x. Applying this to both sides of (3.6) it is easy to see that $\log(1+\mathbf{w}^\top\mathbf{r}) \approx \mathbf{w}^\top\mathbf{r}$ and $\mathbf{w}^\top \log(\mathbf{1}+\mathbf{r}) \approx \mathbf{w}^\top\mathbf{r}$.

i.e.,

$$\sum_{j=1}^{N} w_j r_{ij}^{\log} \approx r_i^{\log,b}, \quad \text{for } i = 1, \ldots, t. \tag{3.7}$$

3.2.3 Prices

The values of the two portfolios when we track prices can be written as follows:

$$V_t^b = V_0^b \prod_{i=1}^{t} \left(\frac{p_i^b}{p_{i-1}^b} \right) = V_0^b \frac{p_t^b}{p_0^b}, \tag{3.8}$$

and

$$V_t = V_0 \prod_{i=1}^{t} \left(\sum_{j=1}^{N} w_j \frac{p_{ij}}{p_{(i-1)j}} \right). \tag{3.9}$$

Therefore, when we track prices we implicitly track the value of the index. In this case, the goal is to construct a portfolio that approximates well the price of the index by combining the prices of the N assets, i.e.,

$$\prod_{i=1}^{t} \left(\sum_{j=1}^{N} w_j \frac{p_{ij}}{p_{(i-1)j}} \right) \approx \frac{p_t^b}{p_0^b}, \quad \forall t > 0. \tag{3.10}$$

3.2.4 Log-Prices

Now, let us consider the case where we track log-prices. The log-values of the two portfolios are given by:

$$\begin{aligned} \log(V_t^b) &= \log(V_0^b) + \sum_{i=1}^{t} \log \left(\frac{p_i^b}{p_{i-1}^b} \right) \\ &= \log(V_0^b) + \log \left(\frac{p_t^b}{p_0^b} \right) \end{aligned} \tag{3.11}$$

and

$$\begin{aligned}\log(V_t) &= \log(V_0) + \sum_{i=1}^{t} \log\left(\sum_{j=1}^{N} w_j \frac{p_{ij}}{p_{(i-1)j}}\right) \\ &\approx \log(V_0) + \sum_{i=1}^{t}\sum_{j=1}^{N} w_j \log\left(\frac{p_{ij}}{p_{(i-1)j}}\right) \\ &= \log(V_0) + \sum_{j=1}^{N} w_j \log\left(\frac{p_{tj}}{p_{0j}}\right).\end{aligned} \tag{3.12}$$

In (3.12) we have used the following approximation:

$$\log\left(\mathbf{w}^\top \frac{\mathbf{p}_t}{\mathbf{p}_{t-1}}\right) \approx \mathbf{w}^\top \log\left(\frac{\mathbf{p}_t}{\mathbf{p}_{t-1}}\right), \tag{3.13}$$

where the $\log(\cdot)$ is applied elementwise. This is effectively the same approximation as (3.6) since $\frac{\mathbf{p}_t}{\mathbf{p}_{t-1}} = \mathbf{1} + \frac{\mathbf{p}_t - \mathbf{p}_{t-1}}{\mathbf{p}_{t-1}} = \mathbf{1} + \mathbf{r}_t$. Therefore, if one tracks the log-prices, then implicitly one is tracking approximately the log-value of the index.

Thus, the goal is to construct a portfolio that approximates well the log-prices of the index by combining the log-prices of the N assets, i.e.,

$$\sum_{j=1}^{N} w_j \log\left(\frac{p_{tj}}{p_{0j}}\right) \approx \log\left(\frac{p_t^b}{p_0^b}\right), \quad \forall t > 0. \tag{3.14}$$

3.2.5 Fitting Error

In the case of returns or log-returns, we try to fit a linear combination of asset (log-)returns that approximates the (log-)returns of the index as shown in (3.3) and (3.7). At each day the returns are independent from the previous day. Thus, a bad estimate of one day does not affect the fitting error of any other day. On the other hand, a key point when we use prices or log-prices is that the error in the fitting is accumulated. A portfolio that produces a bad (log-)price fit one day, will propagate this error until the end.

To make this more clear, let us focus on the difference between log-returns and log-prices. Following (3.7), we define the error of the log-returns fitting at the i-th day as

$$\epsilon_t^{\text{log-ret}} = r_t^{\log,b} - \mathbf{w}^\top \mathbf{r}_t^{\log}. \tag{3.15}$$

Then, for a T-period fitting, a log-return tracking approach is trying to minimize a function of the form $\sum_{t=1}^{T} f_\epsilon(\epsilon_t^{\text{log-ret}})$, where $f_\epsilon(\cdot)$ is an appropriate cost function. Equivalently, following (3.14), we define the error of the log-prices fitting at the t-th day as

$$\begin{aligned}\epsilon_t^{\text{log-pr}} &= \log\left(\frac{p_t^b}{p_0^b}\right) - \mathbf{w}^\top \log\left(\frac{\mathbf{p}_t}{\mathbf{p}_0}\right) \\ &= \log\left(\prod_{i=1}^{t}(1+r_i^b)\right) - \mathbf{w}^\top \log\left(\prod_{i=1}^{t}(\mathbf{1}+\mathbf{r}_i)\right) \\ &= \sum_{i=1}^{t} r_i^{\log,b} - \sum_{i=1}^{t} \mathbf{w}^\top \mathbf{r}_i^{\log} \\ &= \sum_{i=1}^{t} \epsilon_i^{\text{log-ret}},\end{aligned} \tag{3.16}$$

where all the operations are elementwise. Therefore, a log-price approach is trying to minimize an error function that has the form $\sum_{t=1}^{T} f_\epsilon(\sum_{i=1}^{t} \epsilon_i^{\text{log-ret}})$. It is clear that in this case the error is accumulated. Intuitively, the log-return approach is expected to have more uniform errors while the log-price approach would produce smaller errors initially and larger errors in the end.

Now, let us examine the case of returns and prices. Following (3.3), we define the error of the returns fitting at the t-th day as

$$\epsilon_t^{\text{ret}} = r_t^b - \mathbf{w}^\top \mathbf{r}_t. \tag{3.17}$$

Then, for a T-period fitting, a return tracking approach is trying to minimize a function of the form $\sum_{t=1}^{T} f_\epsilon(\epsilon_t^{\text{ret}})$. Equivalently, following (3.10), we define the error of the prices fitting at the t-th day as

$$\begin{aligned}\epsilon_t^{\text{pr}} &= \frac{p_t^b}{p_0^b} - \prod_{i=1}^{t}\left(\sum_{j=1}^{N} w_j \frac{p_{ij}}{p_{(i-1)j}}\right) \\ &= \prod_{i=1}^{t}(1+r_i^b) - \prod_{i=1}^{t}(1+\mathbf{w}^\top \mathbf{r}_i).\end{aligned} \tag{3.18}$$

Although in this case we cannot trivially decompose ϵ_t^{pr} to a simple function of $(\epsilon_1^{\text{ret}}, \ldots, \epsilon_t^{\text{ret}})$, the dependence of the past error is obvious.

3.2.6 Connection to Pairs-Trading

When we track prices or log-prices, we effectively try to find the most cointegrated portfolio to the index. This is equivalent to the pairs-selection step in a pairs-trading strategy [5, 41, 35]. For example, a simple measure for pairs-selection is the normalized price distance (NPD), defined as

$$\text{NPD} = \sum_{t=1}^{T} (\tilde{p}_{t1} - \tilde{p}_{t2})^2, \tag{3.19}$$

where $\tilde{p}_{tj}$ is the normalized price of the j-th portfolio at the t-th day. In order for two portfolios to be good candidates for a pairs-trading strategy they must have a low NPD. However, this is exactly equivalent to the portfolio goals (3.10) and (3.14) for $f_\epsilon(x) = x^2$.

To make the above connection more clear, consider the case where we track prices using the function $f_\epsilon(x) = x^2$ as an error measure. Our goal is to construct a portfolio that minimizes the tracking error given by (3.18). We can define the normalized price of the index at the t-th day as $\tilde{p}_t^b = \frac{p_t^b}{p_0^b}$ (given by the right hand side of (3.10)) and the normalized price of the portfolio as $\tilde{p}_t = \prod_{i=1}^{t} \left(\sum_{j=1}^{N} w_j \frac{p_{ij}}{p_{(i-1)j}} \right)$ (given by the left hand side of (3.10)). Then, we wish to minimize the function

$$\sum_{t=1}^{T} f_\epsilon(\epsilon_t^{\text{pr}}) = \sum_{t=1}^{T} f_\epsilon(\tilde{p}_t^b - \tilde{p}_t) = \sum_{t=1}^{T} (\tilde{p}_t^b - \tilde{p}_t)^2, \tag{3.20}$$

which is exactly equivalent to minimizing the NPD of the portfolio and the index. The case of log-prices follows in a straightforward manner.

3.3 Performance Measures

In practice, we cannot track perfectly the tracking quantity and we need to define a performance measure to quantify the tracking error. Thus, we need to define a performance measure in order to be able to design and compare different portfolios. Then, we can derive appropriate algorithms that construct "optimal" portfolios according to the selected performance measure.

A common measure of performance is the empirical tracking error (ETE) which measures how closely the tracking portfolio replicates the index [11, 45, 10, 52, 65]. It is defined as

$$\mathrm{ETE}(\mathbf{w}) = \frac{1}{T}\|\mathbf{r}^b - \mathbf{X}\mathbf{w}\|_2^2. \tag{3.21}$$

One disadvantage of the ETE is that it penalizes both the positive and negative divergence from the index. This may be a desired property if we use a tracking portfolio for hedging purposes where we wish to neither outperform or underperform the index. However, if we are tracking an index to benefit from the returns this property is not desirable.

Another performance measure is the downside risk (DR) relative to an index which is defined as [11, 38]

$$\mathrm{DR}(\mathbf{w}) = \frac{1}{T}\|(\mathbf{r}^b - \mathbf{X}\mathbf{w})^+\|_2^2, \tag{3.22}$$

that is, we are interested in minimizing the tracking error only when the index beats our portfolio.

Notice that in both (3.21) and (3.22) we have used the ℓ_2-norm to penalize the differences of the returns. However, in volatile periods where there are many extreme returns, a more robust penalty function is more appropriate. A common robust penalty function is the Huber function defined as:

$$\phi(x) = \begin{cases} x^2, & |x| \leq M, \\ M(2|x| - M), & |x| > M, \end{cases} \tag{3.23}$$

where $M > 0$ is a parameter.

Figure 3.2 illustrates how Huber compares to the ℓ_2-norm. We observe that it penalizes linearly extreme values and this is the reason it is more robust in the presence of outliers.

Combining the Huber penalty with the above tracking errors we can get the following robust performance measures [11]:

$$\mathrm{HETE}(\mathbf{w}) = \frac{1}{T}\mathbf{1}^\top \boldsymbol{\phi}\left(\mathbf{r}^b - \mathbf{X}\mathbf{w}\right), \tag{3.24}$$

$$\mathrm{HDR}(\mathbf{w}) = \frac{1}{T}\mathbf{1}^\top \boldsymbol{\phi}\left((\mathbf{r}^b - \mathbf{X}\mathbf{w})^+\right), \tag{3.25}$$

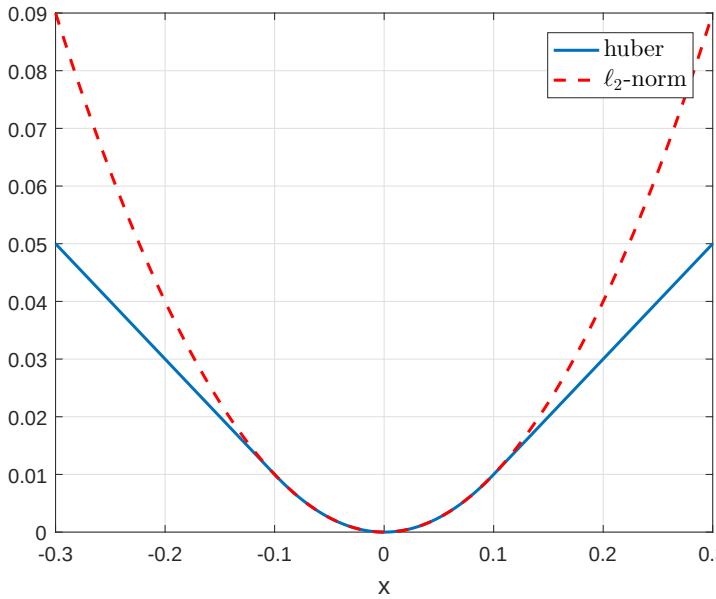


Figure 3.2: Huber function for $M = 0.1$ compared to the ℓ_2-norm.

where $\boldsymbol{\phi}(\mathbf{x}) = [\phi(x_1), \ldots, \phi(x_T)]^\top$.

Other possible performance measures are the Value-at-Risk (VaR) relative to an index [39], and the conditional Value-at-Risk (CVaR) relative to an index [61]. If the returns follow an elliptical distribution, VaR and CVaR are convex functions and in fact admit a closed-form expression [49]. In the case of non-elliptical distributions, CVaR remains convex (although VaR does not), and can be approximated by sampling the distribution [61]. In general, CVaR is preferred more in practice since it is a coherent measure of risk while VaR is not [8].

Note that all of the above tracking measures use the returns of the index and the assets as the tracking quantity of interest. However, this quantity can be replaced with log-returns, prices and log-prices as we showed earlier.

3.4 Problem Formulation

3.4.1 Two-Step Approaches

The first approach of sparse index tracking is to decompose the task into two steps, namely asset selection, where we select a subset of the assets, and capital allocation, where we distribute the capital among the selected assets.

Various asset selection methods have been proposed. A simple and naive approach is to randomly select $K < N$ stocks [30]. This method is usually used as a benchmark. A widely used method, especially for a market capitalization weighted index, is to select the largest K assets according to their market capitalizations [56]. Another approach is to select the assets that have similar return performances as the index, i.e., the K most correlated assets to the index [15, 30]. Finally, a selection based on cointegration has been proposed where the idea is to select K assets so that there exists a linear combination of their log-prices cointegrated well with the value of the index [1, 15, 31].

Once a subset of K assets has been selected we need to assign the capital in a proper manner. A naive allocation is to distribute the capital among the selected assets proportional to the original weights with their summation equal to 1, that is,

$$\mathbf{w} = \frac{\mathbf{b} \odot \mathbf{s}}{\mathbf{1}^T(\mathbf{b} \odot \mathbf{s})}, \tag{3.26}$$

where $\odot$ denotes Hadamard (elementwise) product, and

$$s_j = \begin{cases} 1 & \text{if the } j\text{-th asset is selected} \\ 0 & \text{otherwise,} \end{cases} \tag{3.27}$$

with $\mathbf{s}^\top \mathbf{1} = K$.

Although this naive allocation is easy and fast, it is not optimized and it further requires the index weight vector $\mathbf{b}$ that may not be available. To this end, we can use the following optimized allocation

that does not make any use of the index weight vector $\mathbf{b}$ [56]:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w} \odot \mathbf{s}) \\ \text{subject to} \quad & (\mathbf{w} \odot \mathbf{s})^\top \mathbf{1} = 1, \\ & \mathbf{w} \geq \mathbf{0}, \end{aligned} \tag{3.28}$$

where TE denotes a general tracking error measure (such as ETE, DR, etc.). The constraints ensure that there is no short-selling, i.e., all the weights are positive, and that they sum up to one (budget constraint). This is effectively the minimization of the tracking error using only the selected assets. To make the solution more robust, one can remove a few assets each time and apply this idea iteratively for several times to achieve enough sparsity in the portfolio [45].

3.4.2 Joint Approaches

The previous methods allocate the capital in two steps (i.e., asset selection and capital allocation) and it is not clear how optimal the resulting tracking portfolio is. Another approach that unifies these two steps is to directly penalize the cardinality of the tracking portfolio [45, 35, 11]:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda \|\mathbf{w}\|_0 \\ \text{subject to} \quad & \mathbf{w}^\top \mathbf{1} = 1, \\ & \mathbf{w} \geq \mathbf{0}, \end{aligned} \tag{3.29}$$

where $\|\mathbf{w}\|_0$ is the ℓ_0-"norm"[2] of $\mathbf{w}$ that measures the number of its nonzero elements, and $\lambda \geq 0$ is a parameter that controls the sparsity of the portfolio, i.e., we get sparser solutions for larger values of λ. Notice that this problem is highly non-convex due to the ℓ_0-"norm" term.

A widely used convex approximation of the ℓ_0-"norm" that promotes sparsity is the ℓ_1-norm, i.e., the "LASSO" technique [72]. Although "LASSO" has been used in portfolio optimization problems [18, 29, 34], it is not directly applicable to index tracking with long only constraints

[2] Although ℓ_p with $p < 1$ is not a norm, it is a common abuse of notation to call it as such. To highlight this difference we will use quotation marks when we are dealing with an ℓ_p with $p < 1$, i.e., ℓ_p-"norm".

since

$$\|\mathbf{w}\|_1 = \sum_{j=1}^{N} |w_j| = \sum_{j=1}^{N} w_j = 1, \tag{3.30}$$

that is, the ℓ_1-norm reduces to a constant and is therefore irrelevant.

Common approaches to deal with (3.29) is to approximate the ℓ_0-"norm" with the non-convex ℓ_p-"norm" [45, 75], where $p < 1$, or other non-convex functions [25], and use heuristic algorithms to solve the optimization problem. Finally, if we do not restrict to long only positions then the ℓ_1-norm is not constant and therefore can be also used to approximate the ℓ_0-"norm" [18].

The index tracking problem can be interpreted as a typical regression problem, i.e., we wish to minimize some appropriate cost function, augmented by a regularization term, like the ℓ_0-"norm", that promotes sparsity. In this scope, Bertsimas et al. [14] developed discrete first-order algorithms to warm-start the mixed-integer quadratic programming (MIQP) computation. The proposed algorithm finds in minutes near optimal solutions for samples in the order of hundreds and predictors in the order of thousands. In [33], Fan et al. discussed a family of variable selection methods that adopt a penalized likelihood approach, as well as more recent methods like the bridge regression, the "LASSO", and the smoothly clipped absolute deviation (SCAD) method, providing a unified algorithm that solves a sequence of quadratic approximations of the original problem. In [77] , Zhang et al. bridge the gap between the ℓ_1 and ℓ_0 penalties. This is done via using non-convex penalties that are a better surrogate (in the sense of approximating the penalty) to ℓ_0 over the ℓ_1. They also produce less biased estimates than those produced by the ℓ_1 ("LASSO") penalized solutions.

3.5 Portfolio Constraints

Apart from the constraints mentioned already, i.e., budget ($\mathbf{w}^\top \mathbf{1} = 1$) and no short selling ($\mathbf{w} \geq \mathbf{0}$), there are a few more possible constraints that we could impose on a tracking portfolio.

First, we may wish to mitigate the weights for some groups of stocks (e.g., sectors), or respect the index sector partition for the selected

subset of stocks. These are all linear constraints and easy to include in an optimization formulation.

In order to control the transaction costs when we rebalance a portfolio we may include a turnover constraint, defined as [65, 38]:

$$\|\mathbf{w} - \mathbf{w}_0\|_1 \leq C_1. \tag{3.31}$$

Here, $\mathbf{w}_0$ refers to the tracking portfolio in the previous time period, and C_1 is an upper bound on the total change in the portfolio between two consecutive periods.

The turnover constraint (3.31) restricts the overall change in a portfolio during rebalancing. However, it does not restrict the number of assets that are rebalanced, i.e., it does not distinguish between large changes in a few assets and small changes in many assets. In practice, the former is preferable due to the fixed cost that each asset is associated with. To overcome this, we can include a sparse turnover constraint, defined as

$$\|\mathbf{w} - \mathbf{w}_0\|_0 \leq C_0, \tag{3.32}$$

where C_0 is an upper bound on the total number of assets that change.

Finally, it is common for the fund managers to impose some holding constraints to avoid extreme positions or brokerage fees for very small orders, which translates into non-convex constraints. In this case, the optimization problem (3.29) takes the following form:

$$\begin{aligned} \underset{\mathbf{w},\mathbf{s}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda \mathbf{s}^\top \mathbf{1} \\ \text{subject to} \quad & \mathbf{w}^\top \mathbf{1} = 1, \\ & \mathbf{l} \odot \mathbf{s} \leq \mathbf{w} \leq \mathbf{u} \odot \mathbf{s}, \\ & \mathbf{s} \in \{0,1\}^N, \end{aligned} \tag{3.33}$$

where s_j plays the role of the indicator function, i.e., $s_j = \mathcal{I}_{\{w_j > 0\}}$, but here is a variable and not fixed as in (3.28). Further, $\mathbf{l}, \mathbf{u} \in \mathbb{R}_+^N$, with $\mathbf{0} \leq \mathbf{l} \leq \mathbf{u}$, are the lower and upper holding bounds, respectively, for the selected stocks.

The optimization problem (3.33) is NP-hard due to the binary variable $\mathbf{s}$. Several different approaches have been used to deal with it. Since (3.33) is a mixed integer programming (MIP) problem, one can

use commercial solvers like GUROBI and CPLEX that can deal with small to medium size MIP problems [19, 65, 71]. However, the worst-case complexity of an MIP solver is exponential, which makes it impractical for higher dimensional problems. To this end, genetic algorithms [6, 10, 65, 56, 22] and differential evolution [6, 52, 63] heuristics have been proposed. However, these methods are not able to prove optimality of the solution and, in general, they have inferior performance compared to an MIP solver [65].

3.6 Sector Information

Sectors are areas of economy in which businesses share related products or services. Investors use sectors to allocate assets and other investments into categories such as technology, health care, energy, utilities and telecommunications, etc. Each of these sectors has a unique risk profile and unique characteristics.

Every index is composed by assets that belong to various sectors, where each sector has a different weight equal to the sum of the weights of the assets of the index that belong to that sector. For example, based on the global industry classification standard (GICS), S&P 500 has assets from 11 sectors: consumer discretionary, consumer staples, energy, financials, health care, industrials, information technology, materials, telecommunications, utilities, and real estate.

Now, consider an index consisting of N assets that are distributed in S sectors. We define the binary mask matrix $\mathbf{M} \in \mathbb{R}^{S\times N}$, where $m_{ij} = 1$ if the j-th asset belong to the i-th sector and $m_{ij} = 0$ otherwise. An issue when we use a sparse tracking portfolio is that the selected assets may correspond only to a few of the original sectors, i.e., we do not select at all assets from some of the sectors. By doing this, our portfolio might not be well diversified to different risk factors. Therefore, we may wish to track the benchmark index with a sparse portfolio $\mathbf{w} \in \mathbb{R}^N_+$ that keeps the sector diversification. This problem can be formulated

as follows:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathrm{TE}(\mathbf{w}) + \lambda \left\| \mathbf{M} \mathcal{I}_{\{\mathbf{w}>\mathbf{0}\}} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{w}^\top \mathbf{1} = 1, \\ & \mathbf{w} \geq \mathbf{0}. \end{aligned} \tag{3.34}$$

The above formulation tends to add more sparsity in large sectors. An alternative formulation that overcomes this drawback is to use $\tilde{\mathbf{M}} = \mathrm{Diag}(\mathbf{M}\mathbf{1}_N)^{-1}\mathbf{M}$ as the mask matrix, i.e., a normalized version of $\mathbf{M}$:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathrm{TE}(\mathbf{w}) + \lambda \left\| \tilde{\mathbf{M}} \mathcal{I}_{\{\mathbf{w}>\mathbf{0}\}} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{w}^\top \mathbf{1} = 1, \\ & \mathbf{w} \geq \mathbf{0}. \end{aligned} \tag{3.35}$$

These two sector formulations produce sparse portfolios but simultaneously keep the diversification among sectors so the resulting portfolios are well diversified.

4

Optimization Algorithms

Index tracking requires efficient algorithms for the construction of tracking portfolios. This is a challenging task since the need for sparsity to reduce the costs, the requirement for low tracking error, and the practical constraint for low running time are in general opposing goals and hard to combine. In this section we derive efficient algorithms for the high-dimensional sparse index tracking problem that possess all of these characteristics.

Before we jump into the algorithm derivations we will first introduce the majorization-minimization (MM) framework which is going to be the basis for all the algorithms. Further, since the ℓ_0-"norm" is not continuous, something essential for the MM framework, we briefly review some continuous approximate functions that are broadly used in the literature.

4.1 Majorization-Minimization

Mathematical models are used in the vast majority of science and engineering branches in order to explain a system, make predictions or give insight about the behavior of a component. Due to the theoretical and technological advances of the last decades, the models have become

significantly more complex and the associated optimization problems very challenging, even for modern computational systems. For example, in some applications the optimization problem may be highly non-convex, having a discontinuous non-convex objective function or a non-convex constraint set. Classical algorithms cannot be applied in this case. Furthermore, in many modern applications we are dealing with high-dimensional non-convex problems that are not separable in the variables and therefore parallel computing cannot be used. The computational resources required for this type of problems is usually prohibitive, especially when a real-time response is required. These obstacles show the necessity of abandoning the idea of a general-purpose solver and adopting a more problem-oriented approach, where we can derive optimization algorithms that exploit the specific structure of each problem. An efficient way to achieve this is through the MM framework [44, 70].

The MM algorithm is a way to handle optimization problems that are too difficult to face directly. The basic idea of MM is fairly simple: instead of minimizing the original objective function $f(x)$, at the (k)-th iteration one can minimize a surrogate function $g(x|x^{(k)})$ that is an upper bound of $f(x)$ and touches $f(x)$ at $x^{(k)}$. It can be seen easily that

$$f\left(\mathbf{x}^{(k+1)}\right) \leq g\left(\mathbf{x}^{(k+1)}|\mathbf{x}^{(k)}\right) \leq g\left(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}\right) = f\left(\mathbf{x}^{(k)}\right)$$

holds, indicating that the sequence $\{f(x^{(k)})\}$ generated by MM is nonincreasing.

Although straightforward as an idea, it is not a trivial task to derive an efficient MM algorithm. The difficulty lies in the construction of the surrogate function. In general, in order for a surrogate function to be meaningful, it should have some properties such as separability in variables, convexity, smoothness, closed-form solution, etc. Thus, instead of solving the complicated original problem we sequentially solve a series of much simpler problems. However, in order to achieve fast convergence, the surrogate function should be sufficiently complicated in order to follow the shape of the original function. These two opposite goals make the construction of a surrogate function very challenging.

Note that we only present the majorization-minimization framework that fits for minimization problems. The minorization-maximization

framework for maximization problems works in an equivalent way, such that in each update $f\left(\mathbf{x}^{(k+1)}\right) \geq f\left(\mathbf{x}^{(k)}\right)$ holds.

4.1.1 The MM Algorithm

Consider a general optimization problem

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{4.1}$$

where $\mathcal{X}$ is a closed set. We say that the function $f(\mathbf{x})$ is majorized at a given point $\mathbf{x}^{(k)}$ by the surrogate function $g\left(\mathbf{x}|\mathbf{x}^{(k)}\right)$ if the following properties are satisfied:

$$\begin{aligned} &\textbf{(A1)} \quad g\left(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}\right) = f\left(\mathbf{x}^{(k)}\right), \\ &\textbf{(A2)} \quad g\left(\mathbf{x}|\mathbf{x}^{(k)}\right) \geq f(\mathbf{x}), \ \forall \mathbf{x} \in \mathcal{X}, \\ &\textbf{(A3)} \quad \nabla g\left(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}\right) = \nabla f\left(\mathbf{x}^{(k)}\right). \end{aligned}$$

With an initial point $\mathbf{x}_0 \in \mathcal{X}$, the MM algorithm generates a sequence of feasible points as follows:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x} \in \mathcal{X}} g\left(\mathbf{x}|\mathbf{x}^{(k)}\right). \tag{4.2}$$

We have already shown that the sequence $\{f(\mathbf{x}^{(k)})\}$ is nonincreasing. Since $\mathcal{X}$ is closed and $f(\mathbf{x})$ continuous, $f(\mathbf{x}^{(k)})$ converges to a finite limit.

In practice, the main difficulty in the derivation of an MM algorithms is to find an appropriate surrogate function such that the minimizer of $g\left(\mathbf{x}|\mathbf{x}^{(k)}\right)$ can be easily found or even have a closed-form solution. This is not a trivial task since there is no systematic way of constructing surrogate functions and the derivation depends highly on the structure and the special characteristics of each problem [70].

4.1.2 Convergence of the MM Algorithm

To claim convergence of $\{\mathbf{x}^{(k)}\}$ we need to make the following assumptions:

(A4) $f(\mathbf{x})$ is continuous and $\mathcal{X}$ convex,

(A5) $g\left(\mathbf{x}|\mathbf{x}^{(k)}\right)$ is continuous in both $\mathbf{x}$ and $\mathbf{x}^{(k)}$,

(A6) $f'\left(\mathbf{x}^{(k)};\mathbf{d}\right) = g'\left(\mathbf{x}^{(k)};\mathbf{d}|\mathbf{x}^{(k)}\right),\ \forall \mathbf{x}^{(k)}+\mathbf{d}\in\mathcal{X},$ where $f'\left(\mathbf{x}^{(k)};\mathbf{d}\right)$ stands for the directional derivative.

Theorem 4.1. Any limit point of the sequence $\{\mathbf{x}_t\}$ is a stationary point of (4.1) if assumptions (A1)-(A6) are satisfied. Further, if

(A7) the initial level set, defined as $\mathcal{X}^0 \triangleq \{\mathbf{x}|f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$, is compact,

then

$$\lim_{t\to+\infty} d\left(\mathbf{x}^{(k)}, \mathcal{X}^\star\right) = 0,$$

where $\mathcal{X}^\star$ is the set of all stationary points of (4.1), and $d\left(\mathbf{x}^{(k)}, \mathcal{X}^\star\right) = \inf_{\mathbf{x}\in\mathcal{X}^\star} \|\mathbf{x}-\mathbf{x}\|_2$ [60, 70].

4.2 ℓ_0-"norm" Approximate Function

Consider the sparse index tracking problem in its general form:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda\|\mathbf{w}\|_0 \\ \text{subject to} \quad & \mathbf{w}\in\mathcal{W}, \end{aligned} \tag{4.3}$$

where $\mathcal{W}$ is a set of possible convex portfolio constraints. We assume that $\{\mathbf{w}|\mathbf{w}\geq\mathbf{0}, \mathbf{w}^\top\mathbf{1}=1\}\subseteq\mathcal{W}$.

The ℓ_0-"norm" is hard to handle directly since it is a non-convex, discontinuous and non-differentiable function. Due to its complexity, a common approach is to approximate it with another, more tractable function. A natural choice is the convex envelope of the ℓ_0-norm, i.e., the ℓ_1-norm, that became widely used after the LASSO was introduced [72]. The ℓ_1-norm regularization has proved to work well in a variety of problems like face recognition [74], compressed sensing [20], sparse

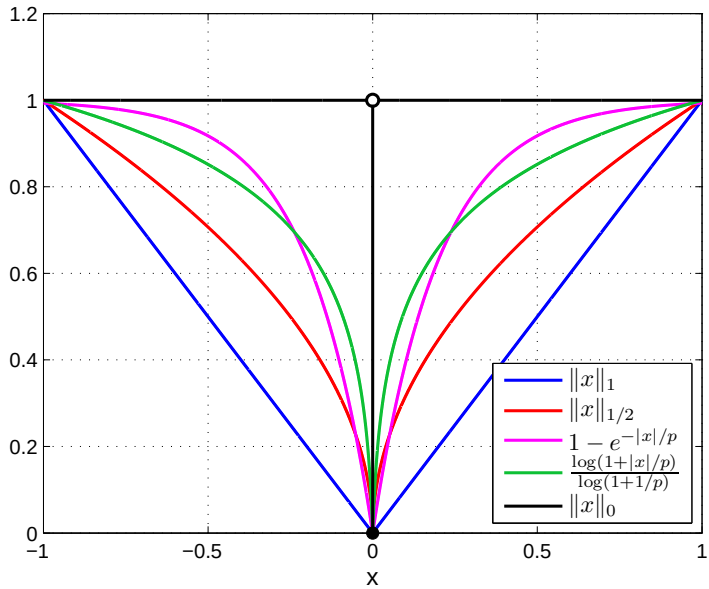


Figure 4.1: Approximate functions of the ℓ_0-"norm" for $p = 0.1$.

PCA [78], etc. However, it is not suitable for applications like index tracking since it reduces to a constant (see Section 3.4.2). Further, other approximations may lead to easier optimization problems when combined with other terms of the objective function, or to an easier construction of surrogate functions when MM is used.

Some of the (non-convex) functions that have been widely used in the literature in order to approximate the ℓ_0-"norm" are [42, 21, 67, 13, 11, 68, 23]:

- ℓ_p-"norm", with $0 < p < 1$,
- $1 - e^{-|x|/p}$ (elementwise), with $p > 0$,
- $\frac{\log(1+|x|/p)}{\log(1+1/p)}$ (elementwise), with $p > 0$.

Note that the approximation in all of these functions is controlled by the parameter p. In Figure 4.1 we observe that all of these approximate

functions are a compromise between the ℓ_0-"norm"[1] and the ℓ_1-norm.

In practice, all of the aforementioned approximate functions have similar performance and can be used interchangeably with an appropriate parameter tuning. To this end, throughout this manuscript we will use the following approximate function:

$$\rho_p(w) = \frac{\log(1 + |w|/p)}{\log(1 + 1/p)}, \tag{4.4}$$

where $0 < p \ll 1$ and $\rho_p(w) \to \mathcal{I}_{\{w \neq 0\}}$ as $p \to 0$. However, all the algorithm derivations can be easily adjusted to the other approximate functions [67].

The function $\rho_p(w)$ is a good approximation of the ℓ_0-"norm" when $w \in [0, 1]$. In many cases we are interested in approximating the indicator function in other intervals, e.g., in the interval $[0, u]$ where $u \leq 1$ is an upper bound of the index weights, or in the interval $[0, l]$, where l is a lower bound of the index weights. The use of the latter will become clear where we discuss the non-convex lower holding constraints. To this end, we use a modified version of the function $\rho_p(w)$ defined as:

$$\rho_{p,\gamma}(w) = \frac{\log(1 + |w|/p)}{\log(1 + \gamma/p)}, \tag{4.5}$$

where $\gamma > 0$. Notice that $\rho_p(w) = \rho_{p,1}(w)$. The function $\rho_{p,\gamma}(w)$ is a good approximation of the indicator function in the interval $[0, \gamma]$.

In the following we set $\gamma = u$, where u is a possible upper bound of the weights specified in the constraint set $\mathcal{W}$. If there is not an upper bound constraint then implicitly $u = 1$[2].

For the multivariate case, it is convenient to define:

$$\boldsymbol{\rho}_{p,u}(\mathbf{w}) = [\rho_{p,u}(w_1), \ldots, \rho_{p,u}(w_N)]^\top. \tag{4.6}$$

Now, we can approximate problem (4.3) as follows:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda \mathbf{1}^\top \boldsymbol{\rho}_{p,u}(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}. \end{aligned} \tag{4.7}$$

[1] More precisely, Figure 4.1 shows the scalar case which is essentially the approximation of the indicator function since $\|\mathbf{w}\|_0 = \sum_{i=1}^N \mathcal{I}_{\{w_i \neq 0\}}$.

[2] For simplicity we assume that $\mathbf{u} = u\mathbf{1}$. However, all the derivations hold for a general upper bound $\mathbf{u}$.

This problem is still not convex since the function $\rho_{p,u}(w)$ is concave (for $w \geq 0$). However, now it has become continuous and differentiable.

4.3 Basic Formulation

In this section we focus on the basic (approximate) sparse index tracking formulation, i.e.,

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda \mathbf{1}^\top \boldsymbol{\rho}_{p,u}(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}. \end{aligned} \tag{4.8}$$

First, in Section 4.3.1 we will derive algorithms for the various tracking errors and a general constraint set $\mathcal{W}$. Then, in Section 4.3.2 we will restrict $\mathcal{W}$ and derive semi-closed-form iterative algorithms based on the MM framework.

4.3.1 General Algorithms

To deal with the non-convexity of (4.8) we will use the MM framework. Since $\boldsymbol{\rho}_{p,\gamma}(\mathbf{w})$ is separable[3] we just need to find a majorization function for the univariate case, i.e., $\rho_{p,\gamma}(w)$. Following [21], an upper bound of $\rho_{p,\gamma}(w)$ at each iteration point $w^{(k)}$ is given by its first-order Taylor expansion.

Lemma 4.2. The function $\rho_{p,\gamma}(w)$, with $w \geq 0$, is majorized at $w^{(k)}$ by $g_{p,\gamma}(w, w^{(k)}) = d_{p,\gamma}(w^{(k)})w + c_{p,\gamma}(w^{(k)})$, where

$$d_{p,\gamma}(w^{(k)}) = \frac{1}{\log(1+\gamma/p)(p+w^{(k)})} \tag{4.9}$$

and

$$c_{p,\gamma}(w^{(k)}) = \frac{\log\left(1+w^{(k)}/p\right)}{\log(1+\gamma/p)} - \frac{w^{(k)}}{\log(1+\gamma/p)(p+w^{(k)})}. \tag{4.10}$$

Proof. See Appendix A.1. □

[3]Here, separable means that each entry $\rho_{p,\gamma}(w_j)$ of $\boldsymbol{\rho}_{p,\gamma}(\mathbf{w})$, as defined in (4.6), is a function only of w_j.

Algorithm 1 LAIT - Linear Approximation for Index Tracking problem (4.8)

1: Set $k = 0$, choose $\mathbf{w}^{(0)} \in \mathcal{W}$
2: **repeat**:
3: Compute $\mathbf{d}_{p,u}^{(k)}$ according to (4.11) and (4.9)
4: Solve (4.13) and set the optimal solution as $\mathbf{w}^{(k+1)}$
5: $k \leftarrow k + 1$
6: **until** convergence
7: **return** $\mathbf{w}^{(k)}$

In the following, it is convenient to define:

$$\mathbf{d}_{p,u}^{(k)} = \left[d_{p,u}(w_1^{(k)}), \ldots, d_{p,u}(w_N^{(k)}) \right]^\top, \tag{4.11}$$

$$\mathbf{c}_{p,u}^{(k)} = \left[c_{p,u}(w_1^{(k)}), \ldots, c_{p,u}(w_N^{(k)}) \right]^\top. \tag{4.12}$$

Now, discarding the constant terms, at the $(k+1)$-th iteration we can solve the following convex optimization problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathrm{TE}(\mathbf{w}) + \lambda {\mathbf{d}_{p,u}^{(k)}}^\top \mathbf{w} \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}. \end{aligned} \tag{4.13}$$

For convex tracking errors such as (3.21), (3.22), (3.24) and (3.25), and for a convex constraint set $\mathcal{W}$, which is an appropriate assumption for index tracking, problem (4.13) is a convex optimization problem and therefore any off-the-shelf solver can be used [17, 43, 54]. Notice that the computational cost of the MM algorithm can be greatly reduced even with generic convex solvers through warm-starting or factorization caching.

Algorithm 1 summarizes the above iterative procedure. We refer to it as LAIT. It is obvious that LAIT is very general since it can be used for any convex tracking error measure TE, and any convex constraint set $\mathcal{W}$.

Polishing

Even if the optimal solution of (4.13) is sparse, solving (4.13) with a solver will not produce in general a sparse result due to the accuracy of

the solver, rounding errors, etc. Therefore Algorithm 1 will not give an explicit subset of assets where we allocate weight. However, many of the weights will be extremely small and therefore we can use a threshold (e.g., 10^{-6}) to distinguish the true support. Finally, having identified the sparsity pattern, we can polish the result by solving (3.28).

The polishing procedure can be beneficial for all the subsequent algorithms in this monograph[4], especially the ones that require a solver. The specialized algorithms produce sparse results since they admit a closed-form solution and therefore polishing is not as important.

4.3.2 Specialized Algorithms

In the previous section we derived an iterative algorithm that works for any convex tracking error measure and convex constraint set. One step of the algorithm is to solve (4.13), which is convex and therefore it can be done by any standard solver. Nevertheless, since the problem has to be solved several times during the MM procedure, depending on the tracking measure, the constraint set and the dimension of the problem, the computational cost can be significant.

Interestingly, for specific tracking measures and constraint sets we can derive algorithms that have a closed-form update and therefore do not require a solver. In particular, we consider the tracking measures ETE, DR, HETE and HDR defined in (3.21), (3.22), (3.24) and (3.25), respectively, and the following convex set parametrized by u:

$$\mathcal{W}_u = \{\mathbf{w} | \mathbf{w}^\top \mathbf{1} = 1, \mathbf{0} \leq \mathbf{w} \leq u\mathbf{1}\}, \tag{4.14}$$

that is, we require the weights to be nonnegative, to have an upper bounds u, and their summation to be equal to one.

First, we state two propositions that will be useful in the derivation of the closed-form update algorithms. Consider an optimization problem of the following form:

$$\begin{array}{ll} \underset{\mathbf{w}}{\text{minimize}} & \mathbf{w}^\top \mathbf{w} + \mathbf{q}^\top \mathbf{w} \\ \text{subject to} & \mathbf{w} \in \mathcal{W}_u, \end{array} \tag{4.15}$$

[4]Depending on the algorithm, the optimization problem (3.28) may vary, e.g., we may need to impose different constraints.

where $\mathbf{q} \in \mathbb{R}^N$. The following propositions provide a waterfilling structured solution of (4.15), considering two special cases, namely, $u = 1$ and $u < 1$ [57, 11].

Proposition 4.1. The optimal solution of the optimization problem (4.15) with $u = 1$ is

$$\mathbf{w}^\star = \left(-\frac{1}{2}(\mu\mathbf{1} + \mathbf{q})\right)^+, \tag{4.16}$$

with

$$\mu = -\frac{\sum_{i\in\mathcal{A}} q_i + 2}{\text{card}(\mathcal{A})}, \tag{4.17}$$

and

$$\mathcal{A} = \{j|\mu + q_j < 0\}, \tag{4.18}$$

where $\mathcal{A}$ can be determined in $O(\log(N))$ steps.

Proof. See Appendix A.2. □

We refer to the iterative procedure of Proposition 4.1 as $\text{AS}_1(\mathbf{q})$ (Active-Set for $u = 1$).

Proposition 4.2. The optimal solution of the optimization problem (4.15) with $u < 1$ is

$$\mathbf{w}^\star = \left(\min\left(-\frac{1}{2}(\mu\mathbf{1} + \mathbf{q}), u\mathbf{1}\right)\right)^+, \tag{4.19}$$

with

$$\mu = -\frac{\sum_{j\in\mathcal{B}_2} q_j + 2 - \text{card}(\mathcal{B}_1)2u}{\text{card}(\mathcal{B}_2)}, \tag{4.20}$$

and

$$\mathcal{B}_1 = \{j|\mu + q_j \leq -2u\}, \tag{4.21}$$
$$\mathcal{B}_2 = \{j| -2u < \mu + q_j < 0\}, \tag{4.22}$$

where $\mathcal{B}_1$ and $\mathcal{B}_2$ can be determined in $O(N\log(N))$ steps.

Proof. See Appendix A.3. □

We refer to the iterative procedure of Proposition 4.2 as $\mathrm{AS}_u(\mathbf{q})$ (Active-Set for general $u < 1$). Again, for simplicity we have considered the simple case where $\mathbf{u} = u\mathbf{1}$, however the extension to a general $\mathbf{u}$ is trivial and the differences are mentioned in Appendix A.3.

Notice that if we set $u = 1$, AS_1 and AS_u do not become the same algorithm although they will return the same optimal solution. Further, a good practice would be to use a smart initialization of the sets $\mathcal{B}_1$ and $\mathcal{B}_2$ based on the solution of AS_1. A more extensive discussion about the iterative algorithms in Propositions 4.1 and 4.2 can be found in Appendices A.2 and A.3, respectively. We will illustrate the benefit of AS_1 and AS_u (with and without initialization) for solving (4.15) in the numerical results in Chapter 5.

We are now ready to consider problem (4.13) for different choices of $\mathrm{TE}(\mathbf{w})$, namely, ETE, DR, HETE, and HDR.

ETE

Let us consider first the tracking measure ETE, defined in (3.21). The optimization problem (4.13) takes the following form:

$$\begin{array}{ll} \underset{\mathbf{w}}{\text{minimize}} & \dfrac{1}{T}\left\|\mathbf{X}\mathbf{w} - \mathbf{r}^b\right\|_2^2 + \lambda \mathbf{d}_{p,u}^{(k)\top}\mathbf{w} \\ \text{subject to} & \mathbf{w} \in \mathcal{W}_u. \end{array} \tag{4.23}$$

By expanding the norm of the objective function and dropping the constants, we can rewrite the optimization problem as follows:

$$\begin{array}{ll} \underset{\mathbf{w}}{\text{minimize}} & \dfrac{1}{T}\mathbf{w}^\top\mathbf{X}^\top\mathbf{X}\mathbf{w} + \left(\lambda \mathbf{d}_{p,u}^{(k)} - \dfrac{2}{T}\mathbf{X}^\top\mathbf{r}^b\right)^\top \mathbf{w} \\ \text{subject to} & \mathbf{w} \in \mathcal{W}_u. \end{array} \tag{4.24}$$

In order to get a closed-form update algorithm we need to majorize the quadratic term and decouple the variables.

Lemma 4.3. Let $\mathbf{L}$ be a real symmetric matrix and $\mathbf{M}$ another real symmetric matrix such that $\mathbf{M} \succeq \mathbf{L}$. Then, for any point $\mathbf{w}^{(k)} \in \mathbb{R}^N$, the quadratic function $\mathbf{w}^\top\mathbf{L}\mathbf{w}$ is majorized at $\mathbf{w}^{(k)}$ by $\mathbf{w}^\top\mathbf{M}\mathbf{w} + 2\mathbf{w}^\top(\mathbf{L} - \mathbf{M})\mathbf{w}^{(k)} + \mathbf{w}^{(k)\top}(\mathbf{M} - \mathbf{L})\mathbf{w}^{(k)}$.

Proof. See [66, Lemma 1]. □

Based on Lemma 4.3, if we set

$$\mathbf{L}_1 = \frac{1}{T}\mathbf{X}^\top\mathbf{X}, \tag{4.25}$$

and

$$\mathbf{M}_1 = \lambda_{\max}^{(\mathbf{L}_1)}\mathbf{I}, \tag{4.26}$$

then $\mathbf{M}_1 \succeq \mathbf{L}_1$ holds and the quadratic term of (4.24) can be majorized at $\mathbf{w}^{(k)}$ by:

$$\begin{aligned}\mathbf{w}^\top\mathbf{L}_1\mathbf{w} \leq & \mathbf{w}^\top\mathbf{M}_1\mathbf{w} + 2\mathbf{w}^\top\left(\mathbf{L}_1 - \mathbf{M}_1\right)\mathbf{w}^{(k)} + const \\ = & \lambda_{\max}^{(\mathbf{L}_1)}\mathbf{w}^\top\mathbf{w} + 2\mathbf{w}^\top\left(\mathbf{L}_1 - \lambda_{\max}^{(\mathbf{L}_1)}\mathbf{I}\right)\mathbf{w}^{(k)} + const\end{aligned}$$

Now, after dropping the constant terms, the new optimization problem at the $(k+1)$-th iteration can be written in the form of (4.15), where

$$\mathbf{q} = \mathbf{q}_{\text{ETE}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_1)}}\left(2\left(\mathbf{L}_1 - \lambda_{\max}^{(\mathbf{L}_1)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda\mathbf{d}_{p,u}^{(k)} - \frac{2}{T}\mathbf{X}^\top\mathbf{r}^b\right) \tag{4.27}$$

is a constant depending on $\mathbf{w}^{(k)}$. This problem can be solved via $\text{AS}_{1|u}\left(\mathbf{q}_{\text{ETE}}^{(k)}\right)$, where $\text{AS}_{1|u}(\cdot)$ means $\text{AS}_1(\cdot)$ or $\text{AS}_u(\cdot)$.

DR

The optimization problem (4.13) for the DR tracking error measure, defined in (3.22), takes the following form:

$$\begin{aligned}\underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{T}\left\|(\mathbf{r}^b - \mathbf{X}\mathbf{w})^+\right\|_2^2 + \lambda{\mathbf{d}_{p,u}^{(k)}}^\top\mathbf{w} \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}_u.\end{aligned} \tag{4.28}$$

Due to the max function $(\cdot)^+$ we cannot simply expand the norm as in the ETE case. To this end, we use the following lemma to transform the objective in a similar form as ETE.

Lemma 4.4. The function $\text{DR}(\mathbf{w})$ defined in (3.22) is majorized at $\mathbf{w}^{(k)}$ by the quadratic convex function $\frac{1}{T}\left\|\mathbf{r}^b - \mathbf{X}\mathbf{w} - \mathbf{y}^{(k)}\right\|_2^2$, where

$$\mathbf{y}^{(k)} = -\left(\mathbf{X}\mathbf{w}^{(k)} - \mathbf{r}^b\right)^+. \tag{4.29}$$

Proof. See Appendix A.4. □

Based on the results of Lemma 4.4, the optimization problem at the $(k+1)$-th iteration becomes:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{T}\|\mathbf{r}^b - \mathbf{X}\mathbf{w} - \mathbf{y}^{(k)}\|_2^2 + \lambda \mathbf{d}_{p,u}^{(k)\top}\mathbf{w} \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}_u. \end{aligned} \tag{4.30}$$

Now, we can follow the same procedure as in the ETE case, i.e., expand the ℓ_2-norm and majorize the quadratic term based on Lemma 4.3. The derivation is similar to the ETE case and therefore we omit it.

The optimization problem at the $(k+1)$-th iteration can be written in the form of (4.15), where

$$\begin{aligned} \mathbf{q} = \mathbf{q}_{\text{DR}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_1)}} \Bigg(& 2\left(\mathbf{L}_1 - \lambda_{\max}^{(\mathbf{L}_1)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} \\ & + \frac{2}{T}\mathbf{X}^\top(\mathbf{y}^{(k)} - \mathbf{r}^b)\Bigg). \end{aligned} \tag{4.31}$$

Thus, it can be solved efficiently via $\text{AS}_{1|u}\left(\mathbf{q}_{\text{DR}}^{(k)}\right)$.

HETE

The optimization problem (4.13) for the HETE tracking error measure, defined in (3.24), takes the following form:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{T}\mathbf{1}^\top \boldsymbol{\phi}\left(\mathbf{X}\mathbf{w} - \mathbf{r}^b\right) + \lambda \mathbf{d}_{p,u}^{(k)\top}\mathbf{w} \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}_u. \end{aligned} \tag{4.32}$$

Lemma 4.5. The function $\phi(x)$ is majorized at $x^{(k)}$ by $f(x|x^{(k)}) = a(x^{(k)})x^2 + b(x^{(k)})$, where

$$a(x^{(k)}) = \begin{cases} 1, & |x^{(k)}| \le M, \\ \frac{M}{|x^{(k)}|}, & |x^{(k)}| > M, \end{cases} \tag{4.33}$$

and

$$b(x^{(k)}) = \begin{cases} 0, & |x^{(k)}| \le M, \\ M(|x^{(k)}| - M), & |x^{(k)}| > M. \end{cases} \tag{4.34}$$

Proof. See Appendix A.5. □

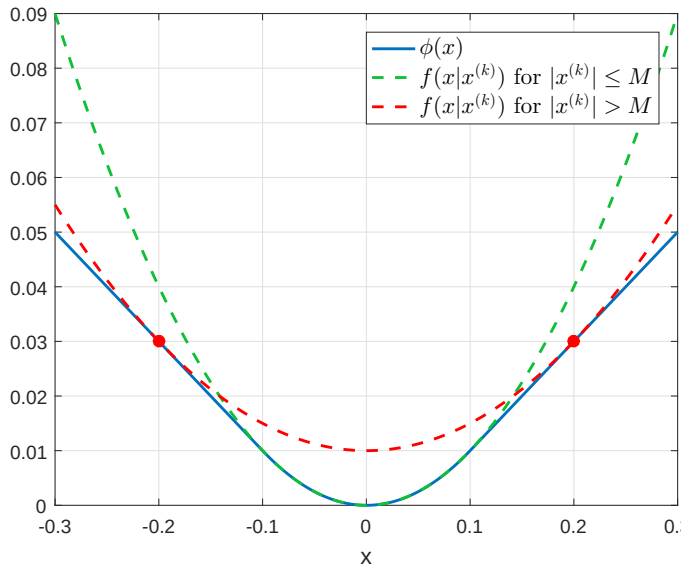


Figure 4.2: Majorization of the Huber penalty $\phi(x)$ with the quadratic function $f(x|x^{(k)})$, for $M = 0.1$.

The Huber function and the two majorization cases are shown in Figure 4.2.

Now, combining Lemma 4.5 and 4.3, we can transform (4.32) to (4.15), where

$$\begin{aligned}\mathbf{q} = \mathbf{q}_{\text{HETE}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_2)}} \Big(2 \left(\mathbf{L}_2 - \lambda_{\max}^{(\mathbf{L}_2)} \mathbf{I} \right) \mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} \\ - \frac{2}{T} \mathbf{X}^\top \text{Diag} \left(\mathbf{a}_{\text{HETE}}^{(k)} \right) \mathbf{r}^b \Big),\end{aligned} \tag{4.35}$$

with

$$\mathbf{L}_2 = \frac{1}{T} \mathbf{X}^\top \text{Diag} \left(\mathbf{a}_{\text{HETE}}^{(k)} \right) \mathbf{X}, \tag{4.36}$$

and

$$\mathbf{a}_{\text{HETE}}^{(k)} = [a([\mathbf{r}^b - \mathbf{X}\mathbf{w}^{(k)}]_1), \ldots, a([\mathbf{r}^b - \mathbf{X}\mathbf{w}^{(k)}]_T)]^\top, \tag{4.37}$$

with $a(\cdot)$ given by (4.33). Again, this problem can be solved efficiently via $\text{AS}_{1|u} \left(\mathbf{q}_{\text{HETE}}^{(k)} \right)$.

HDR

The optimization problem (4.13) for the HDR tracking error measure, defined in (3.25), takes the following form:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{T}\mathbf{1}^\top \boldsymbol{\phi}\left(\left(\mathbf{r}^b - \mathbf{X}\mathbf{w}\right)^+\right) + \lambda \mathbf{d}_{p,u}^{(k)\top}\mathbf{w} \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}_u, \end{aligned} \tag{4.38}$$

where the Huber penalty combined with the downside risk is defined as

$$\phi((x)^+) = \begin{cases} 0, & x \leq 0, \\ x^2, & 0 < x \leq M, \\ M(2|x| - M), & x > M, \end{cases} \tag{4.39}$$

with $M > 0$ a parameter.

Lemma 4.6. The function $\phi((x)^+)$ is majorized at $x^{(k)}$ by $f(x|x^{(k)}) = a(x^{(k)})(x - c(x^{(k)}))^2 + b(x^{(k)})$, where

$$a(x^{(k)}) = \begin{cases} \frac{M}{M-2x^{(k)}}, & x^{(k)} < 0, \\ 1, & 0 \leq x^{(k)} \leq M, \\ \frac{M}{|x^{(k)}|}, & x^{(k)} > M, \end{cases} \tag{4.40}$$

$$b(x^{(k)}) = \begin{cases} 0, & x^{(k)} \leq M, \\ M(|x^{(k)}| - M), & x^{(k)} > M, \end{cases} \tag{4.41}$$

and

$$c(x^{(k)}) = \begin{cases} x^{(k)}, & x^{(k)} < 0, \\ 0, & x^{(k)} \geq 0. \end{cases} \tag{4.42}$$

Proof. See Appendix A.6. □

The Huber penalty combined with the downside risk and the three majorization cases are shown in Figure 4.3.

Combining Lemma 4.6 and 4.3, we can transform (4.38) to (4.15), where

$$\begin{aligned} \mathbf{q} = \mathbf{q}_{\text{HDR}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_3)}} \Big(& 2\left(\mathbf{L}_3 - \lambda_{\max}^{(\mathbf{L}_3)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} \\ & + \frac{2}{T}\mathbf{X}^\top \text{Diag}\left(\mathbf{a}_{\text{HDR}}^{(k)}\right)\left(\mathbf{c}_{\text{HDR}}^{(k)} - \mathbf{r}^b\right)\Big), \end{aligned} \tag{4.43}$$

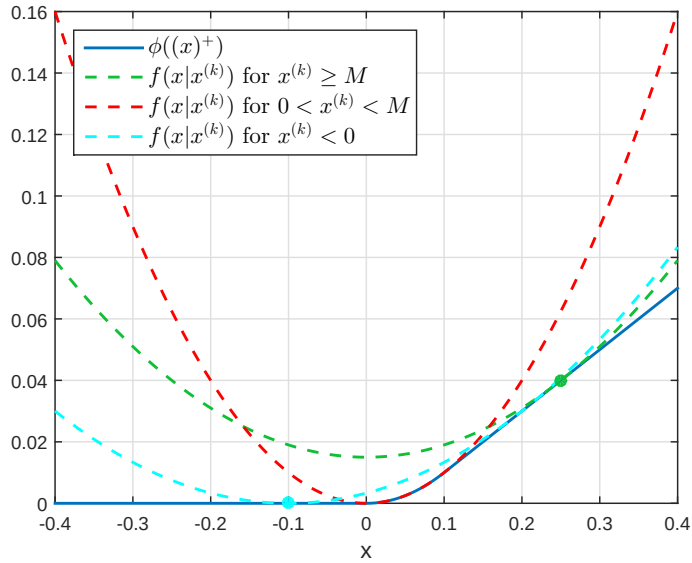


Figure 4.3: Majorization of the Huber penalty $\phi((x)^+)$ with the quadratic function $f(x|x^{(k)})$, for $M = 0.1$.

with

$$\mathbf{L}_3 = \frac{1}{T}\mathbf{X}^\top \text{Diag}\left(\mathbf{a}_{\text{HDR}}^{(k)}\right)\mathbf{X}, \tag{4.44}$$

$$\mathbf{a}_{\text{HDR}}^{(k)} = [a([\mathbf{r}^b - \mathbf{X}\mathbf{w}^{(k)}]_1), \ldots, a([\mathbf{r}^b - \mathbf{X}\mathbf{w}^{(k)}]_T)]^\top, \tag{4.45}$$

$$\mathbf{c}_{\text{HDR}}^{(k)} = [c([\mathbf{r}^b - \mathbf{X}\mathbf{w}^{(k)}]_1), \ldots, c([\mathbf{r}^b - \mathbf{X}\mathbf{w}^{(k)}]_T)]^\top, \tag{4.46}$$

and $a(\cdot)$, $c(\cdot)$ are given by (4.40) and (4.42), respectively. This problem can be solved via $\text{AS}_{1|u}\left(\mathbf{q}_{\text{HDR}}^{(k)}\right)$.

Algorithm 2 summarizes the overall iterative procedure to solve (4.8) with $\mathcal{W} = \mathcal{W}_u$. We refer to it as SLAIT.

All the variations of Algorithm 2 are included in the R software package ***sparseIndexTracking*** [12].

4.4 Holding Constraints

In this section we revisit problem (3.33) assuming a general set of convex constraints $\mathcal{W}$ and the non-convex holding constraints that

Algorithm 2 SLAIT - Specialized LAIT for (4.8) with $\mathcal{W} = \mathcal{W}_u$

Set $k = 0$, choose $\mathbf{w}^{(0)} \in \mathcal{W}_u$
Compute $\lambda_{\max}^{(\mathbf{L}_1)}$ for ETE/DR or $\lambda_{\max}^{(\mathbf{L}_2)}$ for HETE/HDR
repeat:
    Compute $\mathbf{q}$ according to (4.27) for ETE, (4.31) for DR, (4.35)for HETE or (4.43) for HDR
    Solve (4.15) with $\mathrm{AS}_{1|u}(\mathbf{q})$ from Propositions 4.1,4.2 and set the optimal solution as $\mathbf{w}^{(k+1)}$
    $k \leftarrow k + 1$
until convergence
return $\mathbf{w}^{(k)}$

we indicate separately. Again, we approximate the ℓ_0-"norm" by the continuous differentiable function $\rho_{p,\gamma}$ defined in (4.5). The problem can be reformulated as follows:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathrm{TE}(\mathbf{w}) + \lambda \mathbf{1}^\top \boldsymbol{\rho}_{p,u}(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}, \\ & \mathbf{l} \odot \boldsymbol{\mathcal{I}}_{\{\mathbf{w}>\mathbf{0}\}} \leq \mathbf{w} \leq \mathbf{u} \odot \boldsymbol{\mathcal{I}}_{\{\mathbf{w}>\mathbf{0}\}}. \end{aligned} \tag{4.47}$$

Here, we have used the notation $\boldsymbol{\mathcal{I}}_{\{\mathbf{w}>\mathbf{0}\}} = [\mathcal{I}_{\{w_1>0\}}, \ldots, \mathcal{I}_{\{w_N>0\}}]^\top$. Notice that the upper bound constraint $\mathbf{w} \leq \mathbf{u} \odot \boldsymbol{\mathcal{I}}_{\{\mathbf{w}>\mathbf{0}\}}$ is equivalent to $\mathbf{w} \leq \mathbf{u}$ and therefore it can be included in $\mathcal{W}$.

In the special case where $l = 0$, this problem becomes equivalent to (4.8) and the algorithms proposed in Section 4.3 can be used to solve it. Thus, we assume that $l > 0$.

4.4.1 Penalization of Constraint Violations

The lower bound constraint of (4.47) is non-convex and hard to deal with directly. Thus, instead of this constraint we can include an additional term in the objective that penalizes all the non-zero w_j's that are less than l. Since the lower bound constraint is separable for each w_j we can use a penalty function that penalizes each w_j independently. A suitable penalty function for a general entry w is the following:

$$f_l(w) = \left(\mathcal{I}_{\{0<w<l\}} \cdot l - w\right)^+. \tag{4.48}$$

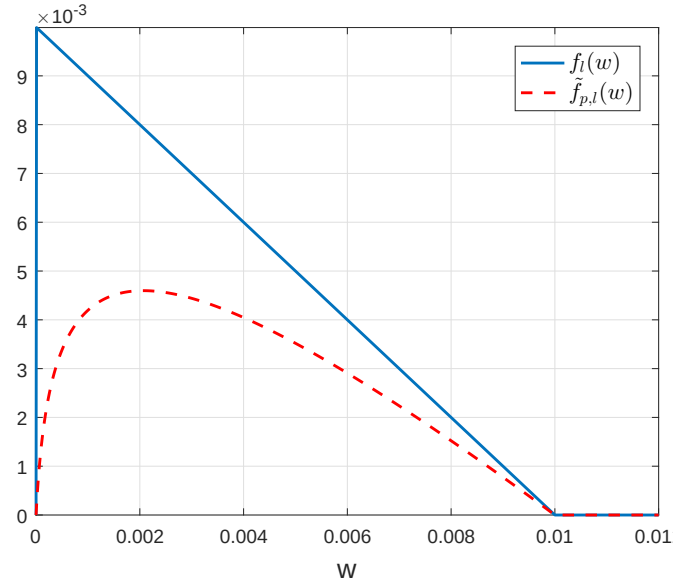


Figure 4.4: Penalty functions $f_l(w)$ and $\tilde{f}_{p,l}(w)$ for $l = 0.01$ and $p = 10^{-4}$.

Again, we can approximate the indicator function with $\rho_{p,\gamma}(w)$, given in (4.5). Since we are interested for the interval $[0, l]$ we select $\gamma = l$. We define the approximate penalty function as:

$$\tilde{f}_{p,l}(w) = (\rho_{p,l}(w) \cdot l - w)^+ . \tag{4.49}$$

In Figure 4.4 we illustrate $f_l(w)$ and $\tilde{f}_{p,l}(w)$ for $l = 0.01$.

Now, we include the additional penalty term in the objective and the new optimization problem becomes:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda \mathbf{1}^\top \boldsymbol{\rho}_{p,u}(\mathbf{w}) + \boldsymbol{\nu}^\top \tilde{\mathbf{f}}_{p,l}(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}, \end{aligned} \tag{4.50}$$

where the parameter $\boldsymbol{\nu} \in \mathbb{R}_+^N$ controls the penalization of the weights that violate the lower bound, and $\tilde{\mathbf{f}}_{p,l}(\mathbf{w}) = [\tilde{f}_{p,l}(w_1), \ldots, \tilde{f}_{p,l}(w_N)]^\top$. This problem is not convex since $\rho_{p,u}(w)$ is concave and $\tilde{f}_{p,l}(w)$ is neither convex nor concave as shown in Figure 4.4.

4.4.2 General Algorithms

Let us first focus on the third term of the objective of (4.50), i.e., the function $\tilde{f}_{p,l}(\mathbf{w})$. Again, since the function is separable we need to deal with the univariate case only.

Lemma 4.7. The function $\tilde{f}_{p,l}(w)$ is majorized at $w^{(k)} \in [0, u]$ by the convex function

$$h_{p,l}(w, w^{(k)}) = \left(\left(d_{p,l}(w^{(k)}) \cdot l - 1 \right) w + c_{p,l}(w^{(k)}) \cdot l \right)^+, \tag{4.51}$$

where $d_{p,l}(w^{(k)})$ and $c_{p,l}(w^{(k)})$ are given by (4.9) and (4.10), respectively.

Proof. See Appendix A.7. □

The second term of (4.50) can be majorized with the linear majorization function presented in Lemma 4.2. The optimization problem at the $(k+1)$-th iteration becomes:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda \mathbf{d}_{p,u}^{(k)\top} \mathbf{w} \\ & + \boldsymbol{\nu}^\top \left(\text{Diag}\left(\mathbf{d}_{p,l}^{(k)} \odot \mathbf{l} - \mathbf{1} \right) \mathbf{w} + \mathbf{c}_{p,l}^{(k)} \odot \mathbf{l} \right)^+ \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}, \end{aligned} \tag{4.52}$$

where $\mathbf{d}_{p,u}^{(k)}$, $\mathbf{d}_{p,l}^{(k)}$ are given by (4.11), and $\mathbf{c}_{p,l}^{(k)}$ by (4.12), respectively. The optimization problem (4.52) is convex and it can be solved using any off-the-shelf solver.

Algorithm 3 summarizes the above iterative procedure. We refer to it as LAITH.

4.4.3 Specialized Algorithms with Holding Constraints

As in the case presented in Section 4.3.2, if we restrict the constraint set $\mathcal{W}$ for the problem (4.52) we can derive algorithms that have a closed-form update and therefore do not require a solver. Here, we consider the same set $\mathcal{W}_u$ given by (4.14).

To get a closed-form update algorithm we need to majorize the objective of (4.52) and transform it to an appropriate form. Let us begin with the majorization of the third term of the objective. This

Algorithm 3 LAITH - Linear Approximation for the Index Tracking problem with Holding constraints (4.50)

1: Set $k = 0$, choose $\mathbf{w}^{(0)} \in \mathcal{W}$
2: **repeat**:
3: Compute $\mathbf{d}_{p,l}^{(k)}, \mathbf{d}_{p,u}^{(k)}$ according to (4.11) and (4.9)
4: Compute $\mathbf{c}_{p,l}^{(k)}$ according to (4.12) and (4.10)
5: Solve (4.52) and set the optimal solution as $\mathbf{w}^{(k+1)}$
6: $k \leftarrow k + 1$
7: **until** convergence
8: **return** $\mathbf{w}^{(k)}$

term is separable and therefore we need to focus only in the univariate case, i.e., in the function $h_{p,l}(w, w^{(k)})$ as defined in Lemma 4.7. However, $h_{p,l}(w, w^{(k)})$ is not smooth due to the max operator. Thus, a majorization function cannot be defined at the point where $h_{p,l}(w, w^{(k)})$ is not smooth due to the non-differentiability of $h_{p,l}(w, w^{(k)})$ [36]. To this end, we will use a smooth approximation of the function $(x)^+$ defined as:

$$\frac{x + \sqrt{x^2 + \epsilon^2}}{2}, \tag{4.53}$$

where $0 < \epsilon \ll 1$ controls the approximation. Applying this result in $h_{p,l}(w, w^{(k)})$, we can define its smooth version as:

$$\tilde{h}_{p,\epsilon,l}(w, w^{(k)}) = \frac{\alpha^{(k)} w + \beta^{(k)} + \sqrt{(\alpha^{(k)} w + \beta^{(k)})^2 + \epsilon^2}}{2}, \tag{4.54}$$

where

$$\alpha^{(k)} = d_{p,l}(w^{(k)}) \cdot l - 1, \tag{4.55}$$

$$\beta^{(k)} = c_{p,l}(w^{(k)}) \cdot l. \tag{4.56}$$

Lemma 4.8. The function $\tilde{h}_{p,\epsilon,l}(w, w^{(k)})$ is majorized at $w^{(k)}$ by the quadratic convex function $q_{p,\epsilon,l}(w, w^{(k)}) = a_{p,\epsilon,l}(w^{(k)})w^2 + b_{p,\epsilon,l}(w^{(k)})w + c_{p,\epsilon,l}(w^{(k)})$, where

$$a_{p,\epsilon,l}(w^{(k)}) = \frac{(\alpha^{(k)})^2}{4\sqrt{(\alpha^{(k)} w^{(k)} + \beta^{(k)})^2 + \epsilon^2}}, \tag{4.57}$$

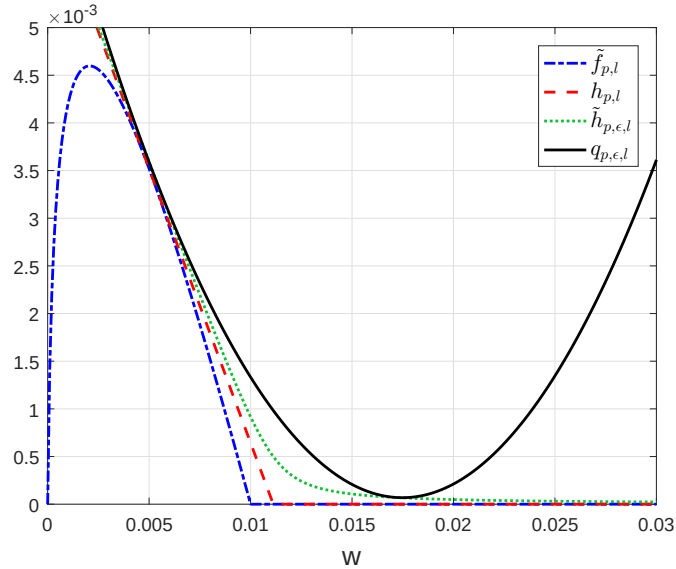


Figure 4.5: Penalty function $\tilde{f}_{p,l}$, linear ($h_{p,l}$) and smooth linear ($\tilde{h}_{p,\epsilon,l}$) upper bounds of $\tilde{f}_{p,l}$, and quadratic upper bound $q_{p,\epsilon,l}$, for $l = 0.01$, $\epsilon = 10^{-3}$, $p = 10^{-4}$.

$$b_{p,\epsilon,l}(w^{(k)}) = \frac{\alpha^{(k)}\beta^{(k)}}{2\sqrt{(\alpha^{(k)}w^{(k)} + \beta^{(k)})^2 + \epsilon^2}} + \frac{\alpha^{(k)}}{2}, \tag{4.58}$$

and

$$c_{p,\epsilon,l}(w^{(k)}) = \frac{(\alpha^{(k)}w^{(k)})^2 + 2\alpha^{(k)}\beta^{(k)}w^{(k)} + {\beta^{(k)}}^2 + \epsilon^2}{4\sqrt{(\alpha^{(k)}w^{(k)} + \beta^{(k)})^2 + \epsilon^2}} + \frac{\beta^{(k)}}{2} \tag{4.59}$$

is an optimization irrelevant constant.

Proof. See Appendix A.8. □

In Figure 4.5 we illustrate this majorization procedure with all the intermediate steps. Now, using the quadratic majorizer of Lemma 4.8 for the third term, problem (4.52) becomes:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda {\mathbf{d}_{p,u}^{(k)}}^\top \mathbf{w} \\ & + \mathbf{w}^\top \text{Diag}\left(\mathbf{a}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\right) \mathbf{w} + \left(\mathbf{b}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\right)^\top \mathbf{w} \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}_u, \end{aligned} \tag{4.60}$$

where

$$\mathbf{a}_{p,\epsilon,l}^{(k)} = [a_{p,\epsilon,l}(w_1^{(k)}), \ldots, a_{p,\epsilon,l}(w_N^{(k)})]^\top, \tag{4.61}$$

$$\mathbf{b}_{p,\epsilon,l}^{(k)} = [b_{p,\epsilon,l}(w_1^{(k)}), \ldots, b_{p,\epsilon,l}(w_N^{(k)})]^\top. \tag{4.62}$$

We will next consider different choices of TE($\mathbf{w}$), namely, ETE, DR, HETE, and HDR.

ETE

Again, let us consider first the tracking measure ETE. By expanding the norm of the objective function, dropping the constants and rearranging the terms we can rewrite the optimization problem as follows:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathbf{w}^\top \left(\frac{1}{T}\mathbf{X}^\top\mathbf{X} + \text{Diag}\left(\mathbf{a}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\right)\right)\mathbf{w} \\ & + \left(\lambda \mathbf{d}_{p,u}^{(k)} - \frac{2}{T}\mathbf{X}^\top\mathbf{r}^b + \mathbf{b}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\right)^\top \mathbf{w} \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}_u, \end{aligned} \tag{4.63}$$

In order to get a closed-form update algorithm we need to majorize the quadratic term of the objective and decouple the variables. Following similar arguments as in Section 4.3.2, and based on Lemma 4.3, we set:

$$\mathbf{L}_4 = \frac{1}{T}\mathbf{X}^\top\mathbf{X} + \text{Diag}\left(\mathbf{a}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\right), \tag{4.64}$$

$$\mathbf{M}_4 = \lambda_{\max}^{(\mathbf{L}_4)}\mathbf{I}. \tag{4.65}$$

The optimization problem at the $(k+1)$-th iteration can be written in the form of (4.15), where

$$\begin{aligned} \mathbf{q} = \mathbf{q}_{\text{ETE-h}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_4)}} \Bigg(& 2\left(\mathbf{L}_4 - \lambda_{\max}^{(\mathbf{L}_4)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} \\ & - \frac{2}{T}\mathbf{X}^\top\mathbf{r}^b + \mathbf{b}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\Bigg). \end{aligned} \tag{4.66}$$

Thus, it can be solved efficiently via $\text{AS}_{1|u}\left(\mathbf{q}_{\text{ETE-h}}^{(k)}\right)$.

DR

A similar procedure can be followed for the DR tracking error measure. First we use Lemma 4.4 to majorize DR with a quadratic function, as in Section 4.3.2. By rearranging the terms and using Lemma 4.3, the optimization problem at the $(k+1)$-th iteration can be written in the form of (4.15), where

$$\begin{aligned}\mathbf{q} = \mathbf{q}^{(k)}_{\text{DR-h}} = \frac{1}{\lambda^{(\mathbf{L}_4)}_{\max}} \Big(2 \left(\mathbf{L}_4 - \lambda^{(\mathbf{L}_4)}_{\max} \mathbf{I} \right) \mathbf{w}^{(k)} + \lambda \mathbf{d}^{(k)}_{p,u} \\ + \frac{2}{T} \mathbf{X}^\top (\mathbf{y}^{(k)} - \mathbf{r}^b) + \mathbf{b}^{(k)}_{p,\epsilon,l} \odot \boldsymbol{\nu} \Big).\end{aligned} \tag{4.67}$$

Thus, it can be solved efficiently via $\text{AS}_{1|u}\left(\mathbf{q}^{(k)}_{\text{DR-h}}\right)$.

HETE

Following similar arguments for the HETE tracking error the optimization problem at the $(k+1)$-th iteration can be written in the form of (4.15), where

$$\begin{aligned}\mathbf{q} = \mathbf{q}^{(k)}_{\text{HETE-h}} = \frac{1}{\lambda^{(\mathbf{L}_5)}_{\max}} \Big(2 \left(\mathbf{L}_5 - \lambda^{(\mathbf{L}_5)}_{\max} \mathbf{I} \right) \mathbf{w}^{(k)} + \lambda \mathbf{d}^{(k)}_{p,u} \\ - \frac{2}{T} \mathbf{X}^\top \text{Diag}\left(\mathbf{a}^{(k)}_{\text{HETE}} \right) \mathbf{r}^b + \mathbf{b}^{(k)}_{p,\epsilon,l} \odot \boldsymbol{\nu} \Big),\end{aligned} \tag{4.68}$$

where

$$\mathbf{L}_5 = \frac{1}{T} \mathbf{X}^\top \text{Diag}\left(\mathbf{a}^{(k)}_{\text{HETE}} \right) \mathbf{X} + \text{Diag}\left(\mathbf{a}^{(k)}_{p,\epsilon,l} \odot \boldsymbol{\nu} \right). \tag{4.69}$$

HDR

Finally, for the HDR tracking error, the parameter $\mathbf{q}$ is given by:

$$\begin{aligned}\mathbf{q} = \mathbf{q}^{(k)}_{\text{HDR-h}} = \frac{1}{\lambda^{(\mathbf{L}_6)}_{\max}} \Big(2 \left(\mathbf{L}_6 - \lambda^{(\mathbf{L}_6)}_{\max} \mathbf{I} \right) \mathbf{w}^{(k)} + \lambda \mathbf{d}^{(k)}_{p,u} \\ + \frac{2}{T} \mathbf{X}^\top \text{Diag}\left(\mathbf{a}^{(k)}_{\text{HDR}} \right) \left(\mathbf{c}^{(k)}_{\text{HDR}} - \mathbf{r}^b \right) + \mathbf{b}^{(k)}_{p,\epsilon,l} \odot \boldsymbol{\nu} \Big),\end{aligned} \tag{4.70}$$

where

$$\mathbf{L}_6 = \frac{1}{T} \mathbf{X}^\top \text{Diag}\left(\mathbf{a}^{(k)}_{\text{HDR}} \right) \mathbf{X} + \text{Diag}\left(\mathbf{a}^{(k)}_{p,\epsilon,l} \odot \boldsymbol{\nu} \right). \tag{4.71}$$

Algorithm 4 SLAITH - Specialized LAITH for (4.50) with $\mathcal{W} = \mathcal{W}_u$

1: Set $k = 0$, choose $\mathbf{w}^{(0)} \in \mathcal{W}_u$
2: Compute $\lambda_{\max}^{(\mathbf{L}_2)}$
3: **repeat**:
4: Compute $\mathbf{q}$ according to (4.66) for ETE, (4.67) for DR, (4.68) for HETE or (4.70) for HDR
5: Solve (4.15) with $\mathrm{AS}_{1|u}(\mathbf{q})$ from Propositions 4.1,4.2 and set the optimal solution as $\mathbf{w}^{(k+1)}$
6: $k \leftarrow k + 1$
7: **until** convergence
8: **return** $\mathbf{w}^{(k)}$

Algorithm 4 summarizes the overall iterative procedure to solve (4.52) with $\mathcal{W} = \mathcal{W}_u$. We refer to it as SLAITH.

4.5 Algorithms Summary

In this section we summarize all the results in a compact table. Effectively, regardless of the chosen tracking error and whether we impose lower bound constraints or not, in every iteration of the MM algorithm we need to solve the following optimization problem:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathbf{w}^\top \mathbf{w} + {\mathbf{q}^{(k)}}^\top \mathbf{w} \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}_u, \end{aligned} \tag{4.72}$$

which has a semi-closed-form solution. The difference between the different design choices lies in the specific form of $\mathbf{q}^{(k)}$. Table 4.1 summarizes all the different variations of $\mathbf{q}^{(k)}$, where the exact form of all the parameter dependencies are given in each corresponding subsection.

Table 4.1: Parameters $\mathbf{q}^{(k)}$ for different tracking error measures and w/ or w/o the presence of lower bound (LB) constraints.

TE	LB	$\mathbf{q}^{(k)}$
ETE	No	$\mathbf{q}_{\text{ETE}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_1)}} \left(2\left(\mathbf{L}_1 - \lambda_{\max}^{(\mathbf{L}_1)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} - \frac{2}{T}\mathbf{X}^\top \mathbf{r}^b\right)$
DR	No	$\mathbf{q}_{\text{DR}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_1)}} \left(2\left(\mathbf{L}_1 - \lambda_{\max}^{(\mathbf{L}_1)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} + \frac{2}{T}\mathbf{X}^\top (\mathbf{y}^{(k)} - \mathbf{r}^b)\right)$
HETE	No	$\mathbf{q}_{\text{HETE}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_2)}} \left(2\left(\mathbf{L}_2 - \lambda_{\max}^{(\mathbf{L}_2)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} - \frac{2}{T}\mathbf{X}^\top \text{Diag}\left(\mathbf{a}_{\text{HETE}}^{(k)}\right)\mathbf{r}^b\right)$
HDR	No	$\mathbf{q}_{\text{HDR}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_3)}} \left(2\left(\mathbf{L}_3 - \lambda_{\max}^{(\mathbf{L}_3)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} + \frac{2}{T}\mathbf{X}^\top \text{Diag}\left(\mathbf{a}_{\text{HDR}}^{(k)}\right)\left(\mathbf{c}_{\text{HDR}}^{(k)} - \mathbf{r}^b\right)\right)$
ETE	Yes	$\mathbf{q}_{\text{ETE-h}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_4)}} \left(2\left(\mathbf{L}_4 - \lambda_{\max}^{(\mathbf{L}_4)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} - \frac{2}{T}\mathbf{X}^\top \mathbf{r}^b + \mathbf{b}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\right)$
DR	Yes	$\mathbf{q}_{\text{DR-h}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_4)}} \left(2\left(\mathbf{L}_4 - \lambda_{\max}^{(\mathbf{L}_4)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} + \frac{2}{T}\mathbf{X}^\top (\mathbf{y}^{(k)} - \mathbf{r}^b) + \mathbf{b}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\right)$
HETE	Yes	$\mathbf{q}_{\text{HETE-h}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_5)}} \left(2\left(\mathbf{L}_5 - \lambda_{\max}^{(\mathbf{L}_5)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} - \frac{2}{T}\mathbf{X}^\top \text{Diag}\left(\mathbf{a}_{\text{HETE}}^{(k)}\right)\mathbf{r}^b + \mathbf{b}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\right)$
HDR	Yes	$\mathbf{q}_{\text{HDR-h}}^{(k)} = \frac{1}{\lambda_{\max}^{(\mathbf{L}_6)}} \left(2\left(\mathbf{L}_6 - \lambda_{\max}^{(\mathbf{L}_6)}\mathbf{I}\right)\mathbf{w}^{(k)} + \lambda \mathbf{d}_{p,u}^{(k)} + \frac{2}{T}\mathbf{X}^\top \text{Diag}\left(\mathbf{a}_{\text{HDR}}^{(k)}\right)\left(\mathbf{c}_{\text{HDR}}^{(k)} - \mathbf{r}^b\right) + \mathbf{b}_{p,\epsilon,l}^{(k)} \odot \boldsymbol{\nu}\right)$

4.6 Turnover Constraint

Consider the general (approximate) sparse index problem (4.8), with additional turnover constraint as defined in (3.31), i.e.,

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathrm{TE}(\mathbf{w}) + \lambda \mathbf{1}^\top \boldsymbol{\rho}_{p,u}(\mathbf{w}) \\ \text{subject to} \quad & \|\mathbf{w}_0 - \mathbf{w}\|_1 \leq C_1, \\ & \mathbf{w} \in \mathcal{W}. \end{aligned} \tag{4.73}$$

The above problem can be majorized with the techniques shown in the previous sections, and solved efficiently by any solver since the turnover constraint is convex. However, even when $\mathcal{W}$ has a simple form, the turnover constraint prevents a closed-form solution. We can overcome this using a similar technique as in the holding constraints, i.e., penalize the turnover constraint in the objective. In that case, the optimization problem becomes:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathrm{TE}(\mathbf{w}) + \lambda \mathbf{1}^\top \boldsymbol{\rho}_{p,u}(\mathbf{w}) + \nu \|\mathbf{w} - \mathbf{w}_0\|_1 \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}, \end{aligned} \tag{4.74}$$

where ν is a regularization parameter.

Since the ℓ_1-norm is not differentiable (which is essential for the MM algorithm derivations) we can use the following approximate function that aims at smoothening the ℓ_1-norm (elementwise) around zero by a quadratic function [67]:

$$g_\epsilon(x) = \begin{cases} \frac{x^2}{2\epsilon}, & |x| \leq \epsilon, \\ |x| - \frac{\epsilon}{2}, & |x| > \epsilon, \end{cases} \tag{4.75}$$

where $\epsilon > 0$ controls the approximation.

Lemma 4.9. The function $g_\epsilon(x)$ is majorized at $x^{(k)}$ by $f(x|x^{(k)}) = a(x^{(k)})x^2 + b(x^{(k)})$, where

$$a(x^{(k)}) = \begin{cases} \frac{1}{2\epsilon}, & |x^{(k)}| \leq \epsilon, \\ \frac{1}{2|x^{(k)}|}, & |x^{(k)}| > \epsilon, \end{cases} \tag{4.76}$$

and

$$b(x^{(k)}) = \begin{cases} 0, & |x^{(k)}| \leq \epsilon, \\ \frac{1}{2}(|x^{(k)}| - \epsilon), & |x^{(k)}| > \epsilon. \end{cases} \tag{4.77}$$

Proof. The smooth ℓ_1-norm function (4.75) is actually a scaled version of the Huber function defined in (3.23). Therefore we can majorize the (smooth) turnover term with a quadratic function as in the Huber case. We ommit the derivation since it is identical to the one in Appendix A.5. □

Similarly to (4.74), we can penalize the sparse turnover constraint, defined in (3.32), as

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda \mathbf{1}^\top \boldsymbol{\rho}_{p,u}(\mathbf{w}) + \nu \|\mathbf{w} - \mathbf{w}_0\|_0 \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}. \end{aligned} \tag{4.78}$$

The term $\mathbf{w} - \mathbf{w}_0$ takes both positive and negative values. Therefore, approximating the ℓ_0-"norm" with (4.4) will lead to a non-differentiable function at $\mathbf{w} = \mathbf{w}_0$. In this case, we need to approximate the ℓ_0-"norm" with a smooth continuous function defined as [67, 13]

$$\rho_p^\epsilon(x) = \begin{cases} \frac{x^2}{2\epsilon(p+\epsilon)\log(1+1/p)}, & |x| \leq \epsilon, \\ \frac{\log\left(\frac{p+|x|}{p+\epsilon}\right) + \frac{\epsilon}{2(p+\epsilon)}}{\log(1+1/p)}, & |x| > \epsilon, \end{cases} \tag{4.79}$$

with $0 < p \leq 1$ and $0 < \epsilon \ll 1$, which is effectively a smooth version of (4.4). Figure 4.6 illustrates the smooth approximate function (4.79) compared to (4.4).

Lemma 4.10. [67] The function $\rho_p^\epsilon(x)$ is majorized at $x^{(k)}$ by $f(x|x^{(k)}) = d(x^{(k)})x^2 + c(x^{(k)})$, where

$$d(x^{(k)}) = \begin{cases} \frac{1}{2\epsilon(p+\epsilon)\log(1+1/p)}, & |x^{(k)}| \leq \epsilon, \\ \frac{1}{2\log(1+1/p)|x^{(k)}|\left(|x^{(k)}|+p\right)}, & |x^{(k)}| > \epsilon, \end{cases} \tag{4.80}$$

and $c(x^{(k)})$ is an optimization irrelevant constant.

Based on Lemmas 4.9 and 4.10, both the turnover and the sparse turnover penalty term can be majorized with a quadratic function. Therefore, following the steps of Section 4.3.2 we can easily derive a semi-closed-form update algorithm including these terms.

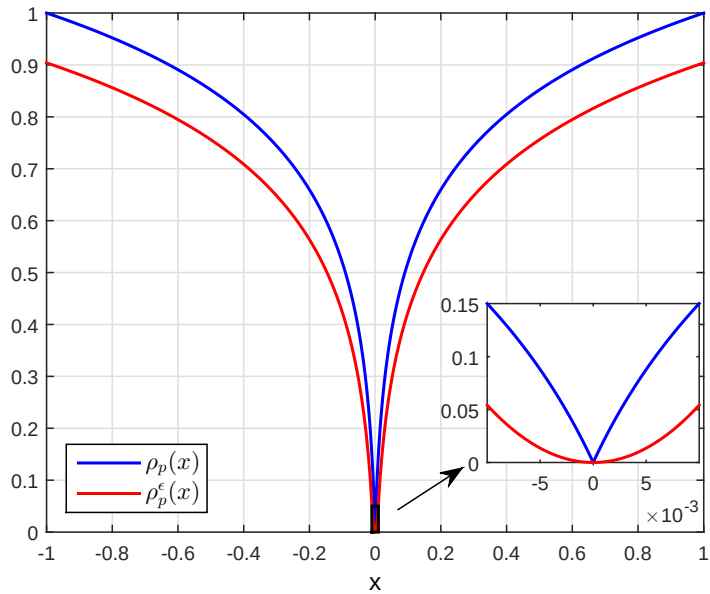


Figure 4.6: ℓ_0-"norm" approximate function $\rho_p(x)$ (4.4) and smooth approximate function $\rho_p^\epsilon(x)$ (4.79), for $p = \epsilon = 10^{-2}$.

4.7 Sector Formulation

Now, let us revisit the sparse index tracking problem with sector information, i.e.,

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda \|\tilde{\mathbf{M}} \mathcal{I}_{\{\mathbf{w}>\mathbf{0}\}}\|_2^2 \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}. \end{aligned} \tag{4.81}$$

First, we approximate the indicator function with (4.5), as in the previous formulations. The approximate problem can be written as follows:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \text{TE}(\mathbf{w}) + \lambda \|\tilde{\mathbf{M}} \boldsymbol{\rho}_{p,u}(\mathbf{w})\|_2^2 \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}. \end{aligned} \tag{4.82}$$

Based on Lemma 4.2, the second term of the objective can be majorized as

$$\|\tilde{\mathbf{M}} \boldsymbol{\rho}_{p,u}(\mathbf{w})\|_2^2 \leq \|\tilde{\mathbf{M}} \mathbf{D}^{(k)} \mathbf{w} + \tilde{\mathbf{M}} \mathbf{c}^{(k)}\|_2^2, \tag{4.83}$$

where $\mathbf{D}^{(k)} = \mathrm{Diag}(\mathbf{d}_{p,u}^{(k)})$.

Now, the optimization problem at the $(k+1)$-th iteration becomes:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \mathrm{TE}(\mathbf{w}) + \lambda \left\| \tilde{\mathbf{M}} \mathbf{D}^{(k)} \mathbf{w} + \tilde{\mathbf{M}} \mathbf{c}^{(k)} \right\|_2^2 \\ \text{subject to} \quad & \mathbf{w} \in \mathcal{W}, \end{aligned} \tag{4.84}$$

which is convex. Furthermore, by expanding the second term of the objective we get a quadratic and a linear term, which can fit nicely with any of the tracking errors we examined in order to produce closed-form solution algorithms following similar techniques as in the previous sections.

4.8 Computational Complexity

In this section we study the computational complexity of the proposed algorithms.

First we consider the LAIT algorithm. In every iteration we need to compute the vector $\mathbf{d}_{p,u}^{(k)}$, which can be done in $O(N)$ operations. Then, in the general case, we need to solve a quadratically constrained quadratic program (QCQP) (depending on the constraints it may reduce to a quadratic program (QP)) [17]. These problems can be reformulated as second order cone programs (SOCP) with complexity $O(N^{3.5}\log(1/\delta))$ per iteration, using the Nesterov-Todd (NT) direction [55], where δ is the accepted duality gap. Thus, keeping only the higher order terms, LAIT has an overall complexity of $O\left(N_{\text{iter}} N^{3.5}\log(1/\delta)\right)$, where N_{iter} is the number of iterations.

In the case of the SLAIT algorithm we do not need to call a solver. First, we need to compute $\mathbf{L}_1/\mathbf{L}_2$ and $\lambda_{\max}^{(\mathbf{L}_1)}/\lambda_{\max}^{(\mathbf{L}_2)}$ in $O(N^2T)$ and $O(N^2)$ operations, respectively. Then, in every iteration we need to compute $\mathbf{d}_{p,u}^{(k)}$ $(O(N))$, perform a matrix-vector multiplication $(O(N^2))$ and finally some vector additions $(O(N))$ in order to obtain $\mathbf{q}^{(k)}$. The last step is to find the next iterate point which can be efficiently computed by the proposed algorithms AS_1 or AS_u in $O(\log(N))$ or $O(N\log(N))$ steps, where the complexity of each step is $O(N)$. Thus, again keeping the higher order terms, the overall complexity of the algorithm is $O\left(N^2T + N_{\text{iter}} N^2\log(N)\right)$.

Table 4.2: Complexity of the proposed algorithms.

Algorithm	Complexity
LAIT/LAITH	$O\left(N_{\text{iter}}N^{3.5}\log(1/\delta)\right)$
SLAIT/SLAITH	$O\left(N^2T + N_{\text{iter}}N^2\log(N)\right)$

The overall complexity of LAITH is the same as the complexity of LAIT, i.e., $O\left(N_{\text{iter}}N^{3.5}\log(1/\delta)\right)$, since in each iteration we need to compute $\mathbf{d}_{p,l}^{(k)}, \mathbf{d}_{p,u}^{(k)}$ and $\mathbf{c}_{p,l}^{(k)}$ in $O(N)$, and then solve an SOCP. Similarly, the complexity of SLAITH is the same as the complexity of SLAIT, i.e., $O\left(N^2T + N_{\text{iter}}N^2\log(N)\right)$. It requires some more vector computations and vector-vector multiplications but these are lower order operations that do not affect the complexity order.

In Table 4.2 we summarize the complexity of all the proposed algorithms.

4.9 Convergence

The majorization-minimization (MM) algorithm is a way to handle optimization problems that are too difficult to deal with directly [44, 70]. Consider a general optimization problem

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & f\left(\mathbf{x}\right) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X}, \end{aligned}$$

where $\mathcal{X}$ is a closed set. We say that the function $f\left(\mathbf{x}\right)$ is majorized at a given point $\mathbf{x}^{(k)}$ by the surrogate function $g\left(\mathbf{x}|\mathbf{x}^{(k)}\right)$ if the following properties are satisfied:

$$g\left(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}\right) = f\left(\mathbf{x}^{(k)}\right), \tag{4.85}$$

$$g\left(\mathbf{x}|\mathbf{x}^{(k)}\right) \geq f\left(\mathbf{x}\right), \forall \mathbf{x} \in \mathcal{X}, \tag{4.86}$$

$$\nabla g\left(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}\right) = \nabla f\left(\mathbf{x}^{(k)}\right). \tag{4.87}$$

Then, $\mathbf{x}$ is iteratively updated (with k denoting iterations) as:

$$\mathbf{x}^{(k+1)} = \arg\min_{\mathbf{x}\in\mathcal{X}} g\left(\mathbf{x}|\mathbf{x}^{(k)}\right). \tag{4.88}$$

With this scheme, it is easy to prove that the objective value is decreased monotonically at each iteration, i.e.,

$$f\left(\mathbf{x}^{(k+1)}\right) \leq g\left(\mathbf{x}^{(k+1)}|\mathbf{x}^{(k)}\right) \leq g\left(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}\right) \leq f\left(\mathbf{x}^{(k)}\right). \tag{4.89}$$

All the algorithms presented in this paper are based on the MM framework. Given the sequence of points $(\mathbf{x}^{(k)})_{k\in\mathbb{N}}$ generated by the algorithm, we know that the sequence of objective values evaluated at these points is non-increasing. Since the constraint sets in our problems are compact, the sequence of objective values is bounded. Thus, it is guaranteed to converge to a finite value. We will analyze the convergence property of the sequence $(\mathbf{x}^{(k)})_{k\in\mathbb{N}}$ generated by the algorithms.

A unified convergence proof can be established given that all the optimization problems satisfy a minimum set of conditions. In particular, we require that all the conditions (4.85)-(4.87) and (4.88) hold, that the objective function f is continuous and bounded below, and the constraint set is convex. These conditions are met by all the optimization problems we focused on.

Further, consider the following assumptions [70][Section II.C]:

1. The sublevel set $\text{lev}_{\leq f(\mathbf{x}_0)} f := \{\mathbf{x} \in \mathcal{X} | f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ is compact given that $f(\mathbf{x}_0) < +\infty$.

2. $f(\mathbf{x})$ and $g(\mathbf{x}|\mathbf{x}^{(k)})$ are continuously differentiable with respect to $\mathbf{x}$.

3. For all $\mathbf{x}^{(k)}$ generated by the algorithm, there exists $\gamma \geq 0$ such that $\forall \mathbf{x} \in \mathcal{X}$, we have
$$(\nabla g(\mathbf{x}|\mathbf{x}^{(k)}) - \nabla g(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}))^\top (\mathbf{x} - \mathbf{x}^{(k)}) \leq \gamma \|\mathbf{x} - \mathbf{x}^{(k)}\|^2.$$

Given this minimum set of requirements and assumptions, the following are guaranteed:

1. The sequence of points $(\mathbf{x}^{(k)})_{k\in\mathbb{N}}$ produced by the MM algorithm converges.

2. The objective value f is non-increasing and converges to a limit $f^\star$, where $f^\star$ is a stationary value.

Therefore, it is guaranteed that all the algorithms presented in this paper converge to a stationary point.

5

Numerical Experiments

In this section we evaluate the performance of the proposed algorithms using historical data of the indices S&P 500 (Bloomberg ticker SPX:IND) and Russell 2000 (Bloomberg ticker RTY:IND). For both indices we use a rolling window where a training period T_{tr} is used to derive the optimal tracking portfolio and a testing period T_{tst} to approximate the index movement with the derived portfolio. The details of each index, i.e., the window sizes T_{tr} and T_{tst}, and the total data period T are presented in Table 5.1.

All the experiments were performed on a PC with a 3.20GHz i5-4570 CPU and 8GB RAM.

5.1 Implementation

5.1.1 Acceleration

The derivation of all the proposed algorithms is based on the majorization-minimization framework. In order to obtain surrogate functions that can be easily solved in closed-form many terms of the original functions were majorized twice. This can possibly lead to loose bounds that translates into a significantly large number of iterations for the MM algorithms to

Table 5.1: Index information.

Index	Data Period	T_{tr}	T_{tst}
S&P 500	01/01/10 - 31/12/15	252	252
Russell 2000	01/06/06 - 31/12/15	1000	252

convergence. Thus, in this section we describe an acceleration scheme, called SQUAREM, that can improve significantly the convergence speed of the proposed algorithms.

SQUAREM was originally proposed in [73] to accelerate EM algorithms. Since MM is a generalization of EM and the update rule of MM is just a fixed-point iteration like EM, we can easily apply the SQUAREM acceleration method to MM algorithms with minor modifications. Without loss of generality we will present the accelerated version only of Algorithm 2 (SLAIT). The accelerated version of Algorithm 4 follows in a straightforward manner.

We denote by $F_{\text{SLAIT}}(\cdot)$ the fixed-point iteration map of the SLAIT algorithm, that is, $\mathbf{w}^{(k+1)} = F_{\text{SLAIT}}(\mathbf{w}^{(k)})$, and by $\text{SLAIT}(\mathbf{w}^{(k)})$ the value of the objective function of (4.8) at the (k)-th iteration. The general SQUAREM method can cause two possible problems to the MM algorithms. First, the updated point may violate the constraints. To solve this issue we can project to the feasible set, which is equivalent to solving the following optimization problem:

$$\begin{aligned} \underset{\mathbf{z}}{\text{minimize}} \quad & \|\mathbf{z} - \mathbf{w}\|_2^2 \\ \text{subject to} \quad & \mathbf{z} \in \mathcal{W}_u, \end{aligned} \tag{5.1}$$

where $\mathbf{w}$ is the updated point and $\mathbf{z}$ the projected. By expanding the norm, the objective can be rewritten as $\mathbf{z}^T\mathbf{z} - 2\mathbf{w}^T\mathbf{z}$. This problem is in the form of (4.15) and can be solved efficiently by $\text{AS}_{1|u}(-2\mathbf{w})$. The second problem is that the acceleration may violate the descent property of the MM algorithm. Thus, a backtracking step is adopted halving the distance of the step-length η and -1. As $\eta \to -1$, $\text{SLAIT}(\mathbf{w}^{(k+1)}) \leq \text{SLAIT}(\mathbf{w}^{(k)})$ is guaranteed to hold. The accelerated SLAIT is summarized in Algorithm 5.

Algorithm 5 Accelerated SLAIT

Set $k = 0$, choose $\mathbf{w}^{(0)} \in \mathcal{W}_u$
repeat:
  $\mathbf{w}_1 = F_{\text{IMRP}}(\mathbf{w}^{(k)})$
  $\mathbf{w}_2 = F_{\text{IMRP}}(\mathbf{w}_1)$
  $\mathbf{r} = \mathbf{w}_1 - \mathbf{w}^{(k)}$
  $\mathbf{v} = \mathbf{w}_2 - \mathbf{w}_1 - \mathbf{r}$
  Compute the step-length $\eta = -\frac{\|\mathbf{r}\|_2}{\|\mathbf{v}\|_2}$
  $\mathbf{w} = \mathbf{w}^{(k)} - 2\eta\mathbf{r} + \eta^2\mathbf{v}$
  $\mathbf{w} = \text{AS}_{1|u}(-2\mathbf{w})$ (projection)
  while SLAIT($\mathbf{w}$) > SLAIT($\mathbf{w}^{(k)}$)
    $\eta \leftarrow (\eta - 1)/2$
    $\mathbf{w} = \mathbf{w}^{(k)} - 2\eta\mathbf{r} + \eta^2\mathbf{v}$
    $\mathbf{w} = \text{AS}_{1|u}(-2\mathbf{w})$ (projection)
  end while
  $\mathbf{w}^{(k+1)} = \mathbf{w}$
  $k \leftarrow k + 1$
until convergence
return $\mathbf{w}^{(k)}$

5.1.2 Sequential Decreasing Scheme

Throughout the paper we have used the function ρ_p as a proxy of the ℓ_0-"norm". The approximation is controlled by the parameter p and in particular, as $p \to 0$ we get $\rho_p \to \ell_0$. However, by setting a very small value to p it is likely that the algorithm will get stuck to a local minimum [21]. To solve this issue, we start with a large value of p, i.e., a "loose" approximation, and solve the corresponding optimization problem. Then, we sequentially decrease p, i.e., we "tighten" the approximation, and solve the problem again using the previous solutions as an initial value (warm-start). Note that in practice we are interested only in the solution of the last, "tightest" problem.

A similar approach is followed for the penalty term presented in (4.49) and the parameter $\boldsymbol{\nu}$. We start with a small value of $\boldsymbol{\nu}$ and solve the corresponding optimization problem. If there are constraint violations we increase the value of $\boldsymbol{\nu}$ and solve the problem again using the previous solutions as an initial value. The algorithms terminate when there are no constraint violations.

5.2 Sparse Index Tracking

In this first experiment we compare the performance of the proposed algorithms LAIT and SLAIT. The solution to the optimization problem (3.29) given by the Gurobi solver for MIP problems, denoted as MIP_{Gur}, serves as the principal benchmark[1]. We further compare the proposed methods with the Hybrid Half-Thresholding algorithm [75], denoted as HHT, and the Diversity Method [45], denoted as $\text{DM}_{1/2}$, where the ℓ_p-"norm" approximation is used, with $p = 1/2$.

All the optimization steps of LAIT are evaluated using the MOSEK solver (SLAIT does not require a solver). For the MIP we chose the Gurobi solver since it is known for its good performance in mixed integer problems. The HHT algorithm (requires the minimization of a QP) is implemented using the function "quadprog" of Matlab, which is also used by the authors of [75]. The ℓ_p-"norm" approximation of the diversity method is evaluated using the built-in function "fmincon" of Matlab. We keep the minimum constraint set $\mathcal{W}_u$ as defined in (4.14) for all algorithms, with $u = 0.05$. Finally, for practical reasons we have set the maximum running time of all algorithms to 1200 seconds.

Initially, we use the first T_{tr} days to design the portfolios, while we evaluate their performance in the next T_{tst} days. In the end of this testing period we need to rebalance our portfolios. For this, we roll the training window and use the last T_{tr} days to design, and the next T_{tst} to evaluate the new portfolios. This scheme is shown pictorially in Figure 5.1. The total number of testing days is $T - T_{\text{tr}}$, i.e., we remove from the total data period T the initial window that we use only for training. Note that we do not rebalance the portfolios during the test periods which means that the portfolios are constant in shares, however, the portfolios are changing daily in dollars due to the daily price changes. To this end, for notational convenience we stack the time-varying tracking portfolios of all the testing days columnwise in a matrix[2] $\mathbf{W} \in \mathbb{R}_+^{N \times (T - T_{\text{tr}})}$ so that he achieved returns can be written in

[1] The MIP solution is optimal if the algorithm runs until full convergence. However, in practice one has to stop the MIP after a certain amount of time so the final solution may be suboptimal as seen later in the numerical results.

[2] The use of a matrix is just to present (5.2) in a compact form. It is always

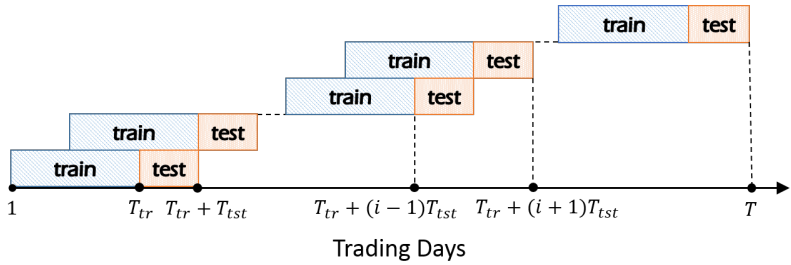


Figure 5.1: Illustration of the rolling training and testing windows.

a compact form as $\text{diag}(\mathbf{XW})$, where $\mathbf{X} \in \mathbb{R}^{(T-T_{\text{tr}})\times N}$.

First, we measure how close the proposed algorithms can replicate a given index. To this end, for a given sparsity level we compute the magnitude of the daily tracking error (MDTE) defined as:

$$\text{MDTE} = \frac{1}{T - T_{\text{tr}}} \left\| \text{diag}(\mathbf{XW}) - \mathbf{r}^b \right\|_2, \tag{5.2}$$

where $\mathbf{r}^b \in \mathbb{R}^{T-T_{\text{tr}}}$. All the tracking error results are presented in basis points (bps)[3]. Apart from the tracking error, we further compute the average[4] running time of each algorithm for the different sparsity levels.

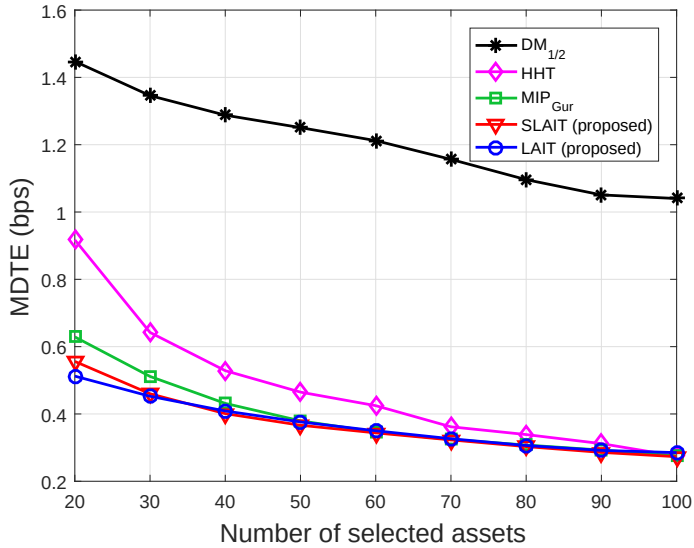
In Figures 5.2(a) and 5.3(a) we compare the tracking error of all the algorithms using the daily returns of the S&P 500 and Russell 2000, respectively. We observe that the proposed algorithms outperform significantly the HHT and $\text{DM}_{1/2}$ algorithms in terms of tracking error. Compared to MIP_{Gur}, the proposed algorithms perform slightly better for small cardinalities and have effectively the same MDTE for larger cardinalities.

The average running time of the algorithms is presented in Figures 5.2(b) and 5.3(b) for the two indices, respectively. We observe that all the algorithms apart from MIP_{Gur} need only a few seconds to converge. On the other hand, the MIP_{Gur} algorithm consumes always all the allowed running time.

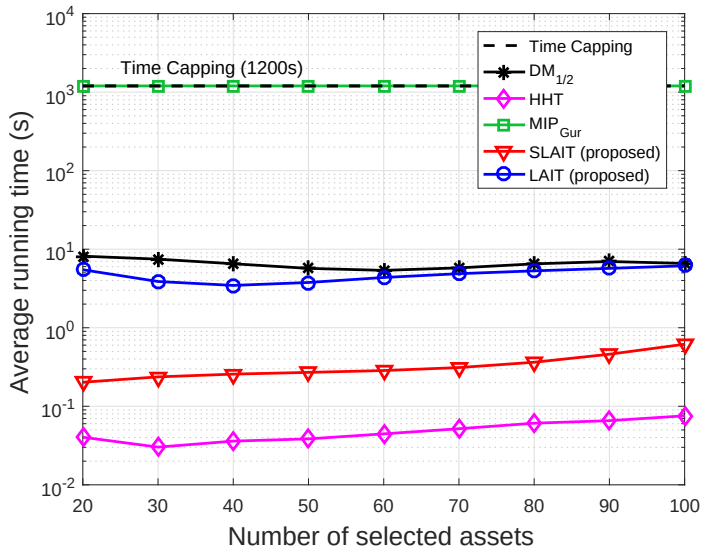
possible to denote the portfolio at the t-th day as $\mathbf{w}_t$ and use an appropriate summation.

[3]One basis point is equal to 0.01%.

[4]For a fixed sparsity level, we need to design $\lceil \frac{T-T_{\text{tr}}}{T_{\text{tst}}} \rceil$ portfolios, one for each testing window. The time averaging is over these portfolios.

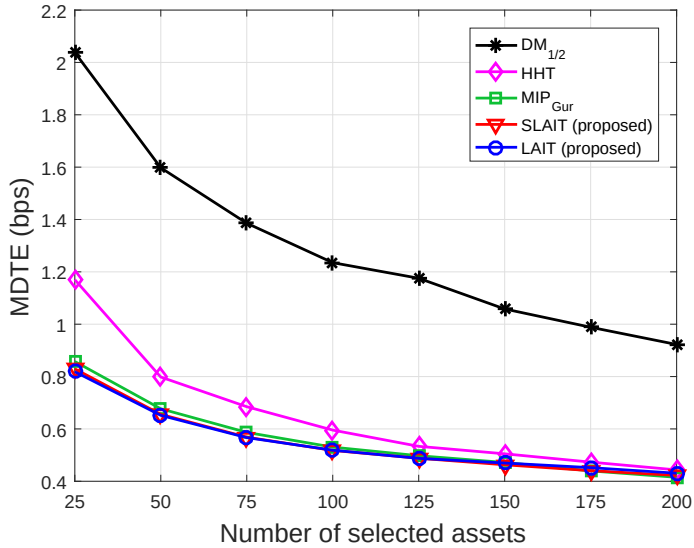


(a) Magnitude of daily tracking error.

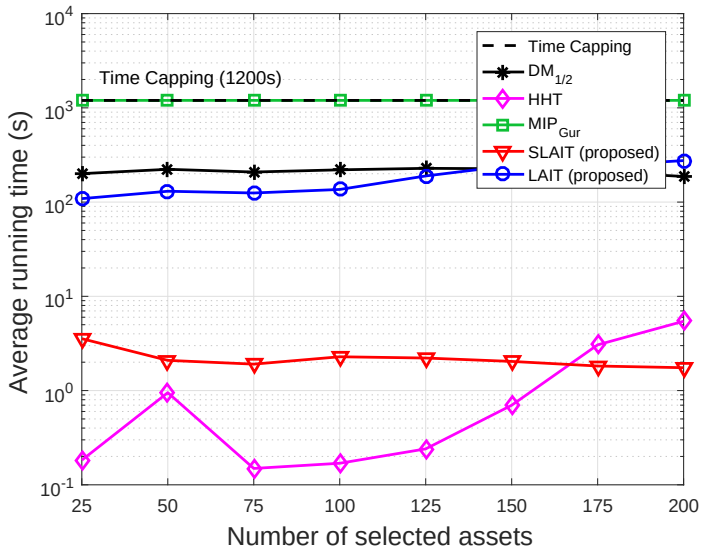


(b) Average running time.

Figure 5.2: S&P 500: Comparison of the proposed algorithms LAIT, ALAIT with the benchmark algorithms MIP_{Gur}, HHT and $\text{DM}_{1/2}$.



(a) Magnitude of daily tracking error.



(b) Average running time.

Figure 5.3: Russell 2000: Comparison of the proposed algorithms LAIT, ALAIT with the benchmark algorithms MIP_{Gur}, HHT and $\text{DM}_{1/2}$.

5.3 Sparse Index Tracking with Holding Constraints

Now, we consider the case where we have the non-convex holding constraints and we compare the algorithms LAITH and SLAIT. We use as a benchmark a direct MIP implementation of the optimization problem (3.33), denoted as $\text{MIP}_{\text{Gur-h}}$. Again, we keep the minimum constraint set $\mathcal{W}_u$ as defined in (4.14) for LAITH and SLAITH, with $u = 0.05$. For $\text{MIP}_{\text{Gur-h}}$ we further include the non-convex lower bound constraint with $l = 0.001$. The propose algorithms take into account these constraints through the proposed penalty term in the objective.

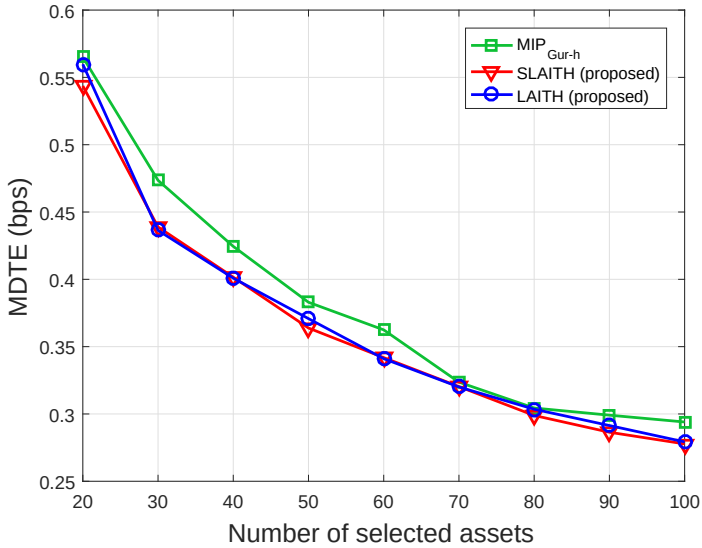
Again, in Figures 5.4 and 5.5 we observe that the proposed algorithms match or outperform $\text{MIP}_{\text{Gur-h}}$ in terms of tracking error, while they are orders of magnitude faster.

5.4 Comparison of Tracking Measures

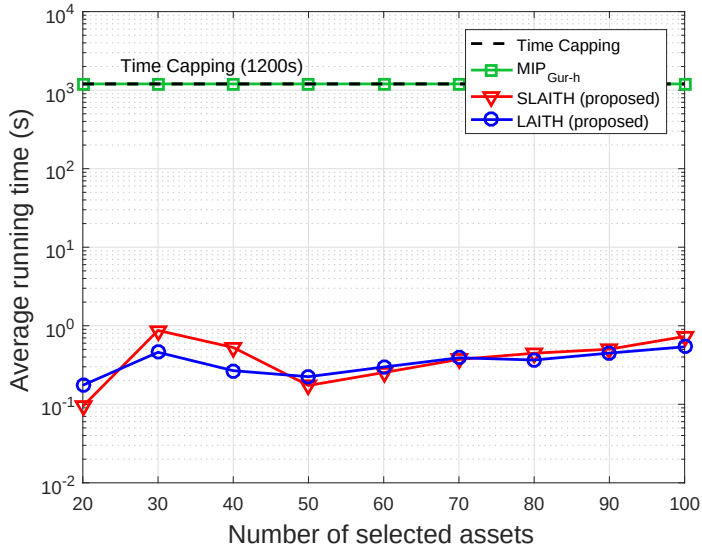
Now, we examine how the various tracking measures perform under periods of different volatility. For this, we use three different periods of the index S&P 500. First, we consider the period 2010-2015 which was relatively stable. Then, we examine the period before and after the great recession (2006-2012), and finally the period around the dot-com bubble (1999-2004). In Table 5.2 we summarize the information about these three periods.

Further, we examine the effect of different rebalancing frequencies. In particular, we fix the training window to be 126 working days ($\sim$ 6 months) and we use a rebalancing frequency (testing window) of 67 days ($\sim$ 3 months) and 22 days ($\sim$ 1 month). Finally, we examine different return frequencies, i.e., we track the daily and monthly returns of the considered periods. All the designed portfolios consist of 40 assets.

Figures 5.6-5.17 illustrate the index movements for the three periods presented in Table 5.2, compared to the movement of the various tracking portfolios. The black vertical dashed lines denote the rebalancing dates of the portfolios. Throughout the whole tracking we keep the notional value of the portfolios fixed, i.e., at each rebalancing date we invest the same amount of money, say \$1. This means that if before rebalancing we had less than \$1 (our investment lost money), we have a loss and we

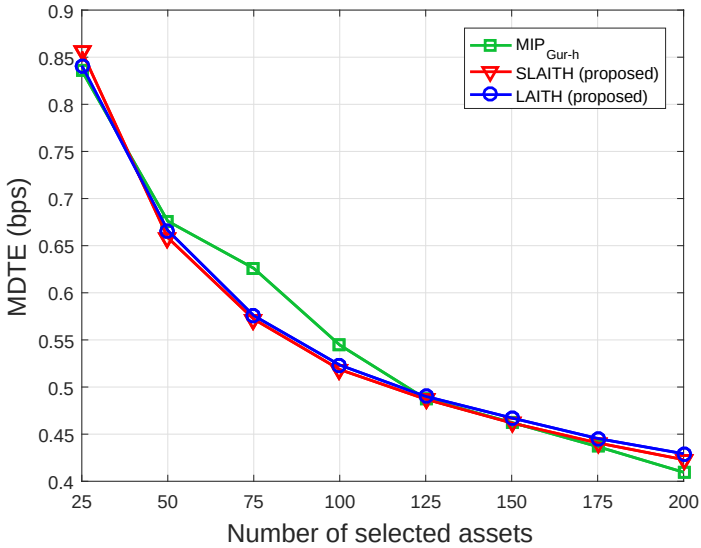


(a) Magnitude of daily tracking error.

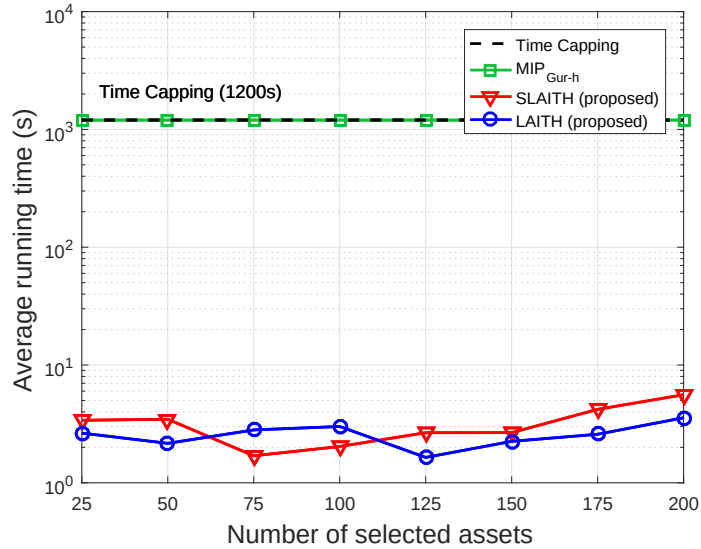


(b) Average running time.

Figure 5.4: S&P 500: Comparison of the proposed algorithms LAITH, ALAITH with the benchmark algorithm $\text{MIP}_{\text{Gur-h}}$.



(a) Magnitude of daily tracking error.



(b) Average running time.

Figure 5.5: Russell 2000: Comparison of the proposed algorithms LAITH, ALAITH with the benchmark algorithm $\text{MIP}_{\text{Gur-h}}$.

Table 5.2: S&P 500 periods.

Data Period	Volatility	Remark
01/01/10 - 31/12/15	Low	Stable market
01/01/06 - 31/12/12	High	Great recession
01/01/99 - 31/12/04	Medium	Dot-com bubble

need to compensate for it. On the other hand, if we had more than \$1 we collect the difference. By accumulating the profits and losses (P&L) at each rebalancing date we get the total P&L for a given period. Excess P&L refers to the P&L of a tracking portfolio if we subtract the P&L of the index. That is, if the excess P&L of a portfolio is zero, it means that it has exactly the same performance as the index.

From Figures 5.6-5.17 we observe that in general monthly returns lead to better results compared to daily returns, something expected since we use a rebalancing frequency in the scale of months. That is, we want our portfolio to be equal to (or beat) the index when we rebalance it regardless of what happened in between.

Moreover, we observe the effect of the rebalancing frequency during the crisis periods where the market is more volatile. In particular, we see that rebalancing more often leads to better results, Figures 5.10-5.17, whereas during a stable period, Figures 5.6-5.9, we could keep the tracking portfolios for longer periods without diverging significantly from the index.

Finally, during the great recession, Figures 5.10-5.13, where there are many extreme returns, the effect of a robust tracking measure is clear. Both ETE and DR with the ℓ_2-norm diverge from their goal. However, the excess P&L of HETE stays around zero, while the excess P&L of HDR clearly beats the index.

5.5 Computational Complexity of $\mathbf{AS}_{1|u}$

For the advanced algorithms SLAIT and SLAITH we have presented two closed-form update algorithms, i.e., AS_1 and AS_u, that solve the

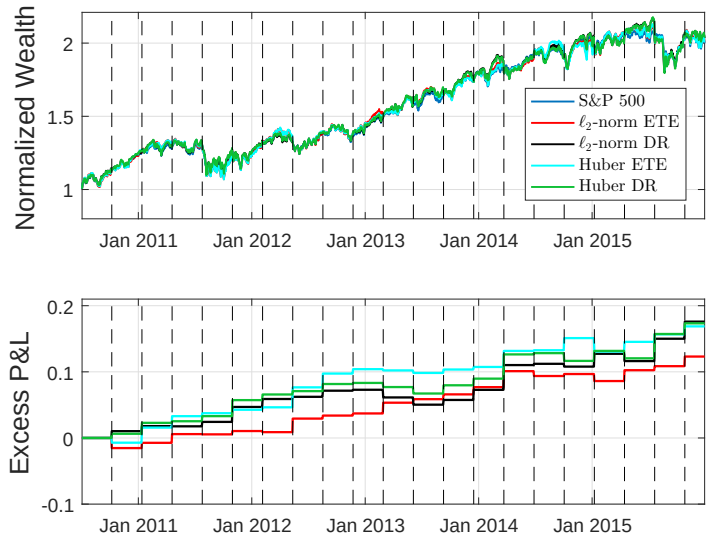


Figure 5.6: Stable market - Returns: daily - Train/Test days: 126/67.

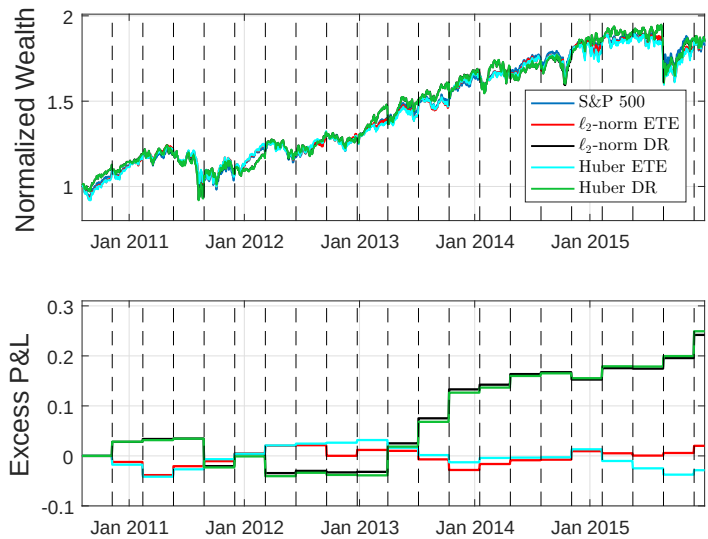


Figure 5.7: Stable market - Returns: monthly - Train/Test days: 126/67.

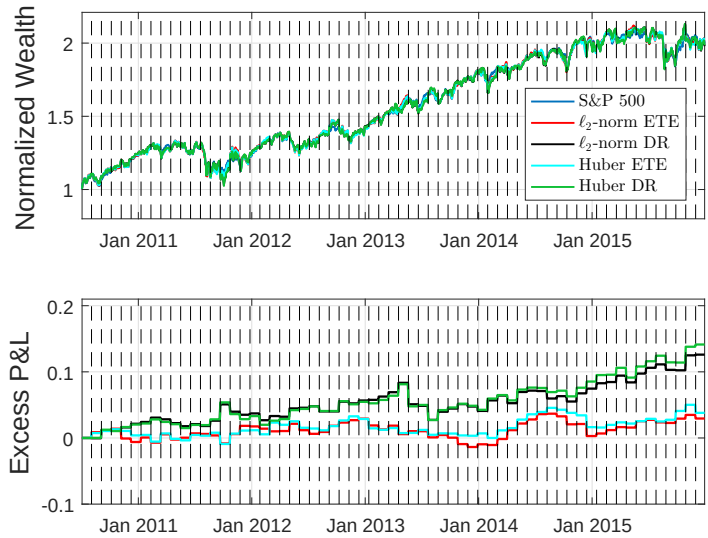


Figure 5.8: Stable market - Returns: daily - Train/Test days: 126/22.

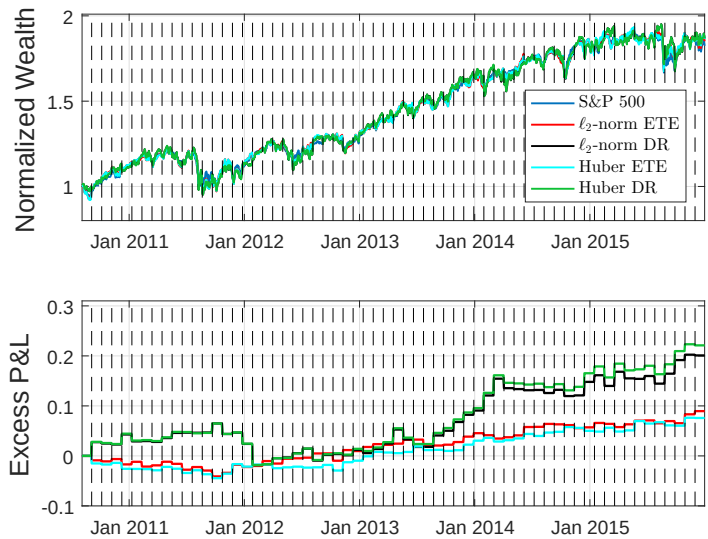


Figure 5.9: Stable market - Returns: monthly - Train/Test days: 126/22.

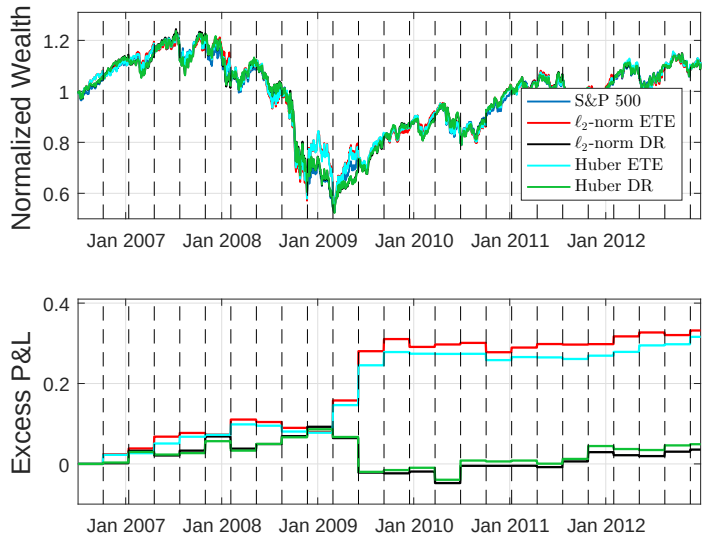


Figure 5.10: Great recession - Returns: daily - Train/Test days: 126/67.

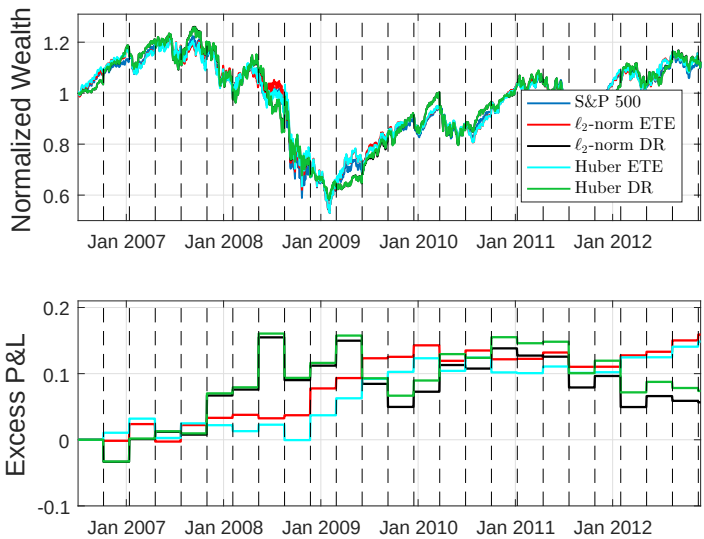


Figure 5.11: Great recession - Returns: monthly - Train/Test days: 126/67.

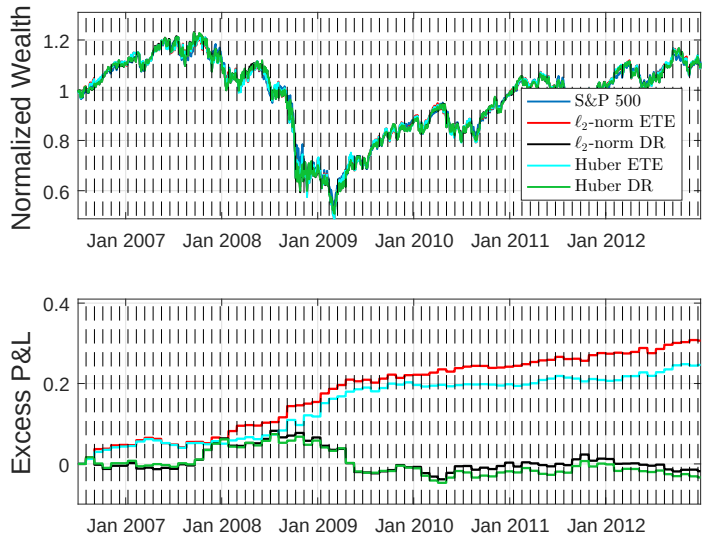


Figure 5.12: Great recession - Returns: daily - Train/Test days: 126/22.

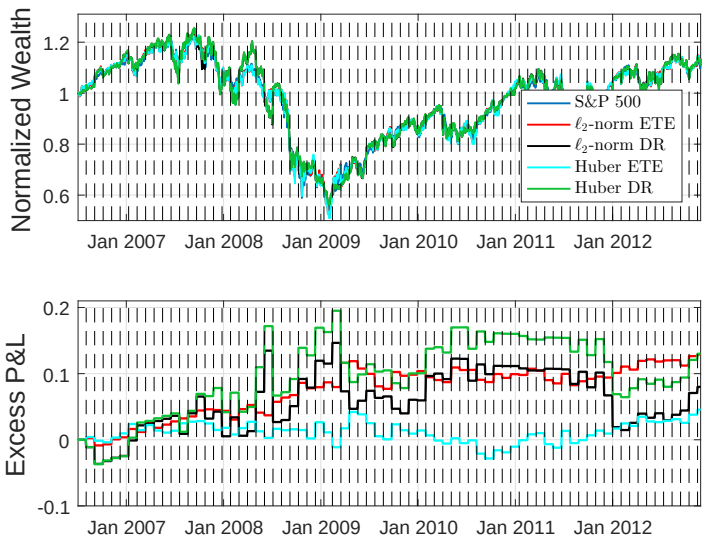


Figure 5.13: Great recession - Returns: monthly - Train/Test days: 126/22.

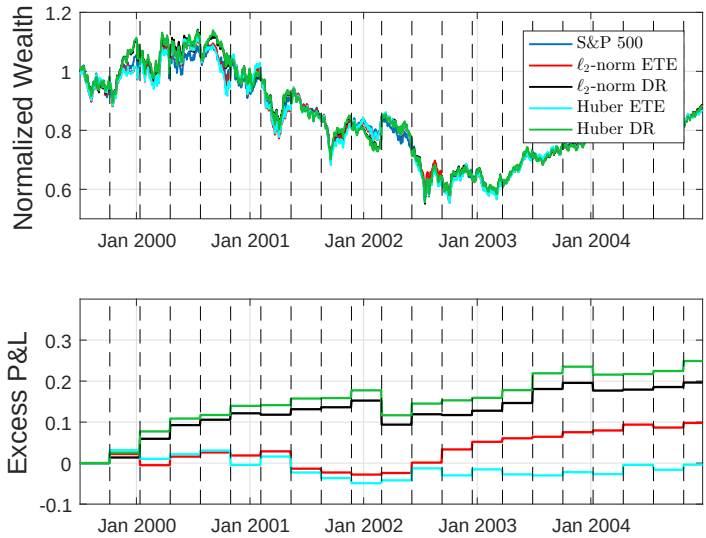


Figure 5.14: Dot-com bubble - Returns: daily - Train/Test days: 126/67.

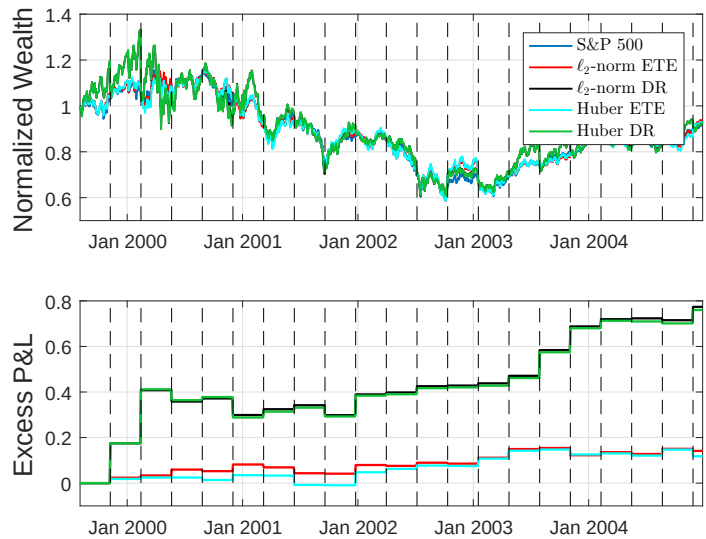


Figure 5.15: Dot-com bubble - Returns: monthly - Train/Test days: 126/67.

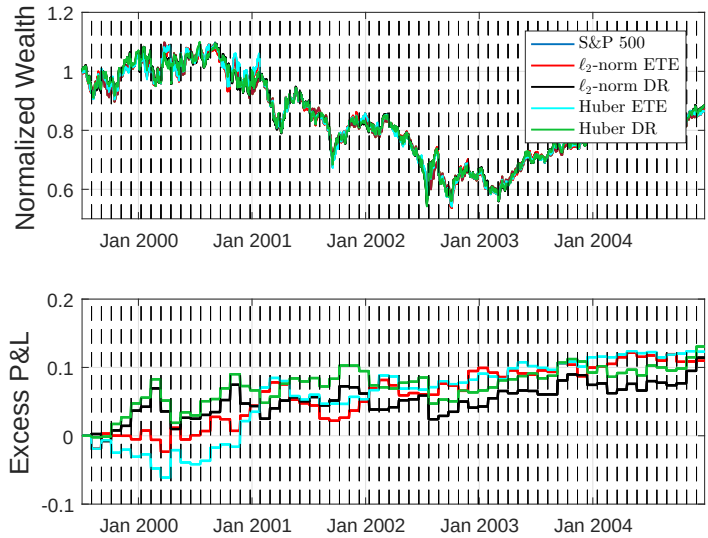


Figure 5.16: Dot-com bubble - Returns: daily - Train/Test days: 126/22.

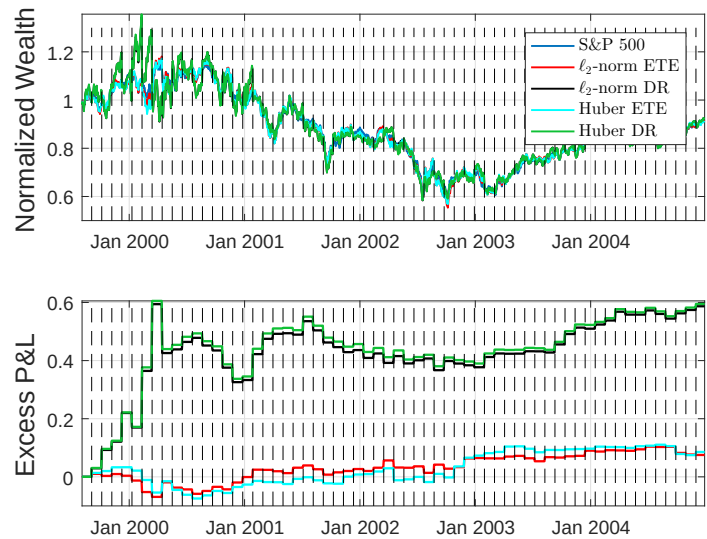


Figure 5.17: Dot-com bubble - Returns: monthly - Train/Test days: 126/22.

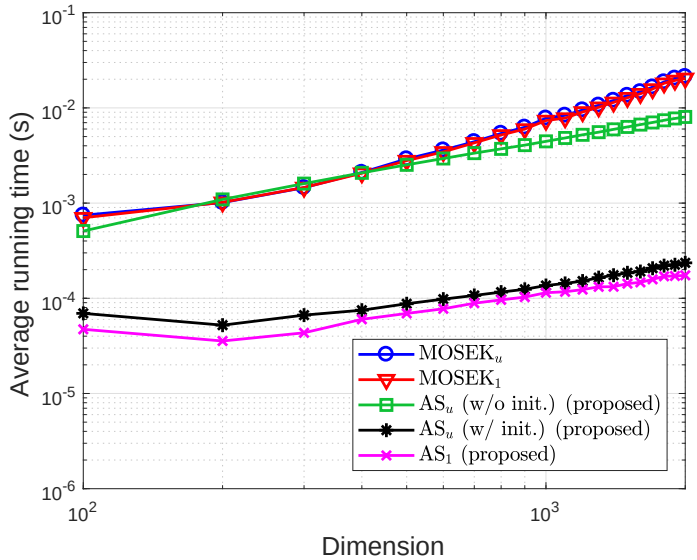


Figure 5.18: Average running time of AS_1 and AS_u. Comparison with the algorithms MOSEK_1 and $\mathrm{MOSEK}_\mathrm{u}$ that correspond to a direct implementation of (4.15) using the MOSEK solver for the cases where $u = 1$ and $u < 1$, respectively. Each curve is an average of 500 random trials.

inner optimization problem (4.15). Here, we compare the performance of these algorithms in terms of average running time with a direct implementation of (4.15) using the MOSEK solver.

For a given problem dimension N, we randomly generate 500 vectors $\mathbf{q} \in \mathbb{R}^N$. To test AS_1 we consider the constraint set $\mathcal{W}_u$ with $u = 1$, while for AS_u we set $u = 20/N$. We sequentially increase the problem dimension from $N = 100$ to $N = 2000$ to examine the scalability of the algorithms.

Figure 5.18 illustrates the average running time of AS_1 and AS_u (with and without initialization) compared to the MOSEK solver. It is clear that the proposed algorithms are more than one order of magnitude faster. Further, they scale well (linearly) with dimension since their average running time increased less than half order of magnitude for $N = 100$ to $N = 2000$.

6

Conclusions

In this monograph we have presented an in-depth analysis of the financial index tracking problem. Index tracking, in all of its forms, requires efficient algorithms for the construction of tracking portfolios. This is a challenging task since the need for sparsity to simplify the execution and to reduce the costs, the requirement for low tracking error, and the practical constraint for low running time are in general opposing goals and hard to combine.

In the first two chapters we have provided a detailed background, providing details about how an index is constructed, the various methods we can track an index and the motivation of an investor to track an index. We have further analyzed two challenges that an investor faces when engaging in index tracking, i.e., the need for frequent rebalancing and the transaction costs, which form a natural trade-off.

In the third chapter we have introduced several design choices that we need to make when we want to create a portfolio, i.e., which quantity we should track and what are possible tracking error measures and portfolio constraints. Finally, we have provided several optimization formulations for the index tracking problem, and various existing methods that solve different variations of the index tracking problem.

In the fourth chapter we have provided new algorithms based on the MM framework. We have shown that regardless of the design choices, we effectively need to solve the same optimization problem, which has a semi-closed-form solution.

Finally, we have performed numerical experiments using historical data of the indices S&P 500 and Russell 2000. We have shown the superiority of the proposed algorithms since they combine two key attributes: they match or outperform (especially in the case of holding constraints) existing benchmarks in terms of tracking error and require a minimal running time to converge. These attributes, combined with the flexibility in tracking measures and constraints, make the proposed algorithms very attractive for practical use.

Abbreviations

AUM	Assets Under Management
CVaR	Conditional Value-at-Risk
DR	Downside Risk
EM	Expectation Maximization
ETE	Empirical Tracking Error
ETF	Exchange Traded Fund
HDR	Huber Downside Risk
HETE	Huber Empirical Tracking Error
KKT	Karush Kuhn Tucker
LASSO	Least Absolute Shrinkage and Selection Operator
MIP	Mixed Integer Programming
MIQP	Mixed Integer Quadratic Programming
MM	Majorization - Minimization
NAV	Net Asset Value
PCA	Principal Component Analysis
QCQP	Quadratically Constrained Quadratic Programming
QP	Quadratic Programming
SOCP	Second Order Cone Programming
TE	Tracking Error
VaR	Value-at-Risk

Notation

$\mathbf{a}$	Vector.
a_i	The i-th entry of $\mathbf{a}$.
a	Scalar or a general entry of vector $\mathbf{a}$.
A	Scalar.
$\mathbf{A}$	Matrix.
$A_{i,j}$	The (i, j)-th element of matrix $\mathbf{A}$.
$\mathcal{A}$	Set.
$(\cdot)^\top$	Transpose.
$\mathbb{N}$	The set of nonnegative integers.
$\mathbb{R}$	The set of real numbers.
$\mathbb{R}^m$, $\mathbb{R}^m_+$, $\mathbb{R}^m_{++}$	The set of real, nonnegative real, and positive real vectors of size m, respectively.
$\mathbb{R}^{m\times n}$, $\mathbb{R}^{m\times n}_+$	The set of $m \times n$ matrices with real and nonnegative real-valued entries, respectively.
$\operatorname{Diag}(\mathbf{a})$	Diagonal matrix with $\mathbf{a}$ as its principal diagonal.
$\operatorname{diag}(\mathbf{A})$	Vector consisting of the diagonal elements of $\mathbf{A}$.
$\mathbf{1}$, $\mathbf{1_n}$	Vector with all elements being 1 of implicit dimension and dimension n, respectively.
$\mathbf{I}$, $\mathbf{I}_n$	Identity matrix of implicit dimension and dimension $n \times n$, respectively.
$\mathcal{I}_{\{\text{condition}\}}$	Indicator function. Equals 1 if the conditions is true and 0 otherwise.

$\mathrm{Tr}(\mathbf{A})$	Trace of matrix $\mathbf{A}$.
$\mathrm{vec}\,(\mathbf{A})$	Vector consisting of all the columns of $\mathbf{A}$ stacked.
$[\mathbf{a}]_{m\times n}$	Matrix of dimension $m\times n$ such that $\mathrm{vec}\,([\mathbf{a}]_{m\times n}) = \mathbf{a}$.
$\lambda_{\max}^{(\mathbf{A})}$	Maximum eigenvalue of matrix $\mathbf{A}$.
$\mathbf{A} \succeq \mathbf{B}$,	$\mathbf{A} - \mathbf{B}$ is positive semidefinite.
$\mathbf{A}^{1/2}$	Hermitian square root of the positive semidefinite matrix $\mathbf{A}$.
$\lvert a\rvert$	Absolute value of a.
$\lVert\mathbf{a}\rVert_p$	ℓ_p-norm of $\mathbf{a}$.
$\lVert\mathbf{a}\rVert_0$	Number of nonzero elements of $\mathbf{a}$.
$\mathrm{rank}(\mathbf{A})$	Rank of matrix $\mathbf{A}$.
$\mathrm{card}(\mathcal{A})$	Cardinality of $\mathcal{A}$.
$\mathcal{A} \cup \mathcal{B}$	Union of $\mathcal{A}$ and $\mathcal{B}$.
$\mathcal{A} \setminus \mathcal{B}$	Difference of$\mathcal{A}$ and $\mathcal{B}$.
$\mathcal{A} \subseteq \mathcal{B}$	$\mathcal{A}$ is a subset of $\mathcal{B}$.
$[i : j]$	Set of all integers between (and including) i and j. For $i > j$ it is the empty set.
$O(\cdot)$	Big-O notation.
$\otimes$	Kronecker product.
$\odot$	Hadamard product.
$\nabla f\,(\mathbf{a})$	Gradient of function $f\,(\mathbf{x})$ evaluated at $\mathbf{x} = \mathbf{a}$.
$\partial f(\mathbf{x})$	Partial derivative of $f(\mathbf{x})$.
max	Maximum.
min	Minimum.
$(a)^+$	$\max(a, 0)$.
$\log\,(\cdot)$	Natural logarithm.

Acknowledgements

The work of Konstantinos Benidis, Yiyong Feng, and Daniel P. Palomar was supported by the Hong Kong Research Grants Council under research grant 16208917.

Appendices

A

Proofs

A.1 Proof of Lemma 4.2

The function $\rho_{p,\gamma}(w)$ is concave for $w \geq 0$. Thus, an upper bound at any point $w^{(k)} \geq 0$ is its first-order Taylor approximation [21, 35]:

$$\begin{aligned}\rho_{p,\gamma}(w) &= \frac{\log(1+w/p)}{\log(1+\gamma/p)} \\ &\leq \frac{1}{\log(1+\gamma/p)}\left[\log\left(1+w^{(k)}/p\right) + \frac{1}{p+w^{(k)}}(w-w^{(k)})\right] \\ &= d_{p,\gamma}(w^{(k)})w + b_{p,\gamma}(w^{(k)}),\end{aligned}$$

where

$$d_{p,\gamma}(w^{(k)}) = \frac{1}{\log(1+\gamma/p)(p+w^{(k)})}$$

and

$$b_{p,\gamma}(w^{(k)}) = \frac{\log\left(1+w^{(k)}/p\right)}{\log(1+\gamma/p)} - \frac{w^{(k)}}{\log(1+\gamma/p)(p+w^{(k)})}.$$

A.2 Proof of Proposition 4.1

In the case where $u = 1$ we can drop the constraint $\mathbf{w} \leq \mathbf{u}$ since it becomes implicit from the other two constraints, i.e., $\mathbf{w}^\top \mathbf{1} = 1$ and

$\mathbf{w} \geq \mathbf{0}$. Further, without loss of generality, throughout this proof we will assume that $\mathbf{q}$ is sorted in ascending order, i.e., $q_i \leq q_j$ for $i < j$.

With this simplification, the Lagrangian of (4.15) is:

$$\mathcal{L}(\mathbf{w}, \mu, \boldsymbol{\nu}) = \mathbf{w}^\top \mathbf{w} + \mathbf{q}^\top \mathbf{w} + \mu(\mathbf{w}^\top \mathbf{1} - 1) - \boldsymbol{\nu}^\top \mathbf{w}.$$

The KKT conditions are:

$$\frac{\partial \mathcal{L}}{\partial w_i} = 2w_i + q_i + \mu - \nu_i = 0, \tag{A.1}$$

$$\mathbf{w}^\top \mathbf{1} = 1, \tag{A.2}$$

$$\mathbf{w} \geq \mathbf{0}, \tag{A.3}$$

$$\boldsymbol{\nu} \geq \mathbf{0}, \tag{A.4}$$

$$\nu_i w_i = 0, \quad \forall i. \tag{A.5}$$

From the derivative of the Lagrangian we get:

$$w_i = \frac{1}{2}(\nu_i - \mu - q_i). \tag{A.6}$$

We identify three cases:

a) $\mu + q_i > 0$: It must hold that $\nu_i \geq \mu + q_i > 0$ since $w_i \geq 0$ (primal feasibility). Further, if $\nu_i > 0$ then necessarily $w_i = 0$ (complementary slackness).

b) $\mu + q_i < 0$: It must hold that $w_i > 0$ since $\nu_i \geq 0$ (dual feasibility). Further, since $w_i > 0$, it holds that $\nu_i = 0$ (complementary slackness) and therefore $w_i = -(\mu + q_i)/2$ (from (A.6)).

c) $\mu + q_i = 0$: The only solution is $w_i = \nu_i = 0$ (complementary slackness).

We can state this result more compactly as follows:

$$w_i = \begin{cases} 0, & \text{if } \mu + q_i \geq 0, \quad \text{(A.7a)} \\ -(\mu + q_i)/2, & \text{if } \mu + q_i < 0. \quad \text{(A.7b)} \end{cases}$$

Thus, for a given μ, only the w_i's corresponding to the smaller q_i's are not zero. Also, if $w_i > 0$, then $w_j > 0$ for all $j < i$.

Now, we need to find the optimal value of the dual variable μ. Assume we know that K_{opt} weights are positive, i.e., $\mathbf{w}_{[1:K_{\text{opt}}]} > \mathbf{0}$ and

$\mathbf{w}_{[K_{\text{opt}}+1:N]} = \mathbf{0}$. From $\mathbf{w}^\top \mathbf{1} = 1$ (primal feasibility), substituting $\mathbf{w}$ given by (A.7a) and (A.7a) we get:

$$\mathbf{w}^\top \mathbf{1} = 1 \quad \Longrightarrow -\sum_{i=1}^{K_{\text{opt}}} (\mu + q_i)/2 = 1.$$

With some trivial term rearrangements we get the value of μ:

$$\mu = -\frac{\sum_{i=1}^{K_{\text{opt}}} q_i + 2}{K_{\text{opt}}}. \tag{A.8}$$

Therefore, a straightforward way to find the optimal solution is to start with $K = 1$ non-zero weights, compute μ from (A.8) and check the conditions (A.7a) and (A.7b). If they hold then $K = K_{\text{opt}}$, else we need to increase K.

However, it is not hard to prove that if $K < K_{\text{opt}}$, then $\mu + q_{K+1} < 0$ which violates (A.7a). Similarly if $K > K_{\text{opt}}$, then $\mu + q_K > 0$ which condition (A.7b)[1]. Thus, by knowing if K should be increased or decreased we can do a binary search to find K_{opt} that terminates in at most $\log(N)$ steps.

A.3 Proof of Proposition 4.2

Without loss of generality, we will assume that $\mathbf{c} = \mathbf{q} + 2\mathbf{u}$ is ordered in ascending order for a general upper bound $\mathbf{u}$, i.e., $c_i \leq c_j$ for $i < j$. Since this proof follows similar steps to the proof of Proposition 4.1 we will skip the details.

The Lagrangian of (4.15) is:

$$\begin{aligned}\mathcal{L}(\mathbf{w}, \mu, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2) =& \mathbf{w}^\top \mathbf{w} + \mathbf{q}^\top \mathbf{w} + \mu(\mathbf{1}^\top \mathbf{w} - 1) \\ & - \boldsymbol{\nu}_1^\top \mathbf{w} + \boldsymbol{\nu}_2^\top (\mathbf{w} - \mathbf{u}).\end{aligned}$$

The KKT conditions are:

$$\frac{\partial \mathcal{L}}{\partial w_i} = 2w_i + q_i + \mu - \nu_{1,i} + \nu_{2,i} = 0, \tag{A.9}$$

[1]Start from $K = 1$. If $K < K_{\text{opt}}$ it should hold that $\mu + q_2 \geq 0$ (else $K_{\text{opt}} = 1$). Use this condition for $K = 2$ and proceed in the same way. It can be easily seen that if $K < K_{\text{opt}}$, (A.7b) always holds however (A.7a) cannot hold. The intuition for the case $K > K_{\text{opt}}$ is the same.

$$\mathbf{w}^\top \mathbf{1} = 1, \tag{A.10}$$
$$\mathbf{w} \geq \mathbf{0}, \tag{A.11}$$
$$\mathbf{w} \leq \mathbf{u}, \tag{A.12}$$
$$\boldsymbol{\nu}_1 \geq \mathbf{0}, \tag{A.13}$$
$$\boldsymbol{\nu}_2 \geq \mathbf{0}, \tag{A.14}$$
$$\nu_{1,i} w_i = 0, \quad \forall i, \tag{A.15}$$
$$\nu_{2,i}(w_i - u) = 0, \quad \forall i. \tag{A.16}$$

From the derivative of the Lagrangian we get:

$$w_i = \frac{1}{2}(\nu_{1,i} - \nu_{2,i} - \mu - q_i). \tag{A.17}$$

Considering the different cases we get the following:

$$w_i = \begin{cases} 0, & \text{if } \mu + c_i \geq 2u_i, & \text{(A.18a)} \\ -(\mu + q_i)/2, & \text{if } 0 < \mu + c_i < 2u_i, & \text{(A.18b)} \\ u_i, & \text{if } \mu + c_i < 0\ . & \text{(A.18c)} \end{cases}$$

These conditions state the following: for a given μ, a subset of w_i's that correspond to the smallest c_i's will take the maximum possible value u_i. Another subset of w_i's with small enough c_i's will take some non-zero value less than u_i. Finally, the w_i's that correspond to the larger c_i's will be zero.

Now, we need to determine the value of μ. Assume we know that $K_{1,\text{opt}}$ weights are equal to u_i and $K_{2,\text{opt}}$ weights are positive and less than u_i, i.e., $\mathbf{w}_{[1:K_{1,\text{opt}}]} = \mathbf{u}_{[1:K_{1,\text{opt}}]}$, $\mathbf{0} < \mathbf{w}_{[K_{1,\text{opt}}+1:K_{1,\text{opt}}+K_{2,\text{opt}}]} < \mathbf{u}_{[K_{1,\text{opt}}+1:K_{1,\text{opt}}+K_{2,\text{opt}}]}$ and $\mathbf{w}_{[K_{2,\text{opt}}+1:N]} = \mathbf{0}$. Then, from $\mathbf{w}^\top \mathbf{1} = 1$ (primal feasibility), the value of μ is:

$$\mu = -\frac{\sum_{i=K_{1,\text{opt}}+1}^{K_{1,\text{opt}}+K_{2,\text{opt}}} q_i - 2\sum_{i=1}^{K_{1,\text{opt}}} u_i + 2}{K_{2,\text{opt}}}. \tag{A.19}$$

In order to evaluate μ we need to determine $K_{1,\text{opt}}$ and $K_{2,\text{opt}}$. In a similar manner as in the proof of Proposition 4.1, we can start with $K = 1$ non-zero weights and sequentially increase its value until we find the optimal one. For a given value of K we can do a binary search to identify K_1 weights with maximum value and K_2 positive weights with

a value less than the upper bound, where $K = K_1 + K_2$. Unfortunately, if the conditions (A.18a)-(A.18c) are not satisfied for the derived K_1 and K_2 we cannot determine if we need to increase or decrease the total number of non-zero weights K. This leads to one linear search with a binary search in each step with a combined complexity $O(N \log(N))$.

A better approach is to set $K = K_{\text{opt}}$, where K_{opt} is the number of positive weights in the case where $u = 1$ (see Appendix A.2), since by imposing an upper bound constraint there will be at least K_{opt} weights that are not zero.

An interesting point is that $K_{2,\text{opt}}$ cannot be zero since it is the denominator in (A.19). We know that $K_{1,\text{opt}} + K_{2,\text{opt}} > 0$ must hold, i.e., there is at least one non-zero weight, since $\mathbf{w}^\top \mathbf{1} = 1$. However, for $K_{2,\text{opt}} = 0$ we get $\mu = +\infty$ and from the conditions (A.18a)-(A.18c) we see that all the weights should be zero. Therefore, this observation shows that it is not possible all the non-zero weights to be equal to their upper bound u_i (since then $K_{2,\text{opt}} = 0$). Although theory shows that we cannot get a solution that all the non-zero weights are equal to their upper bounds, in practice this could happen if we have a group of some extremely negative c_i's while the rest of c_i's are much larger. In this case, due to roundoff errors and finite precision, a solution where all the non-zero weights are equal to their upper bounds is possible and it needs a special consideration.

Finally, in the special case where $\mathbf{u} = u\mathbf{1}$, μ becomes:

$$\mu = -\frac{\sum_{i=K_{1,\text{opt}}+1}^{K_{1,\text{opt}}+K_{2,\text{opt}}} q_i - 2K_{1,\text{opt}}u + 2}{K_{2,\text{opt}}}. \tag{A.20}$$

Further, note that in this case, sorting according to $\mathbf{c}$ is equivalent to sorting according to $\mathbf{q}$.

A.4 Proof of Lemma 4.4

For convenience set $\mathbf{z} = \mathbf{r}^b - \mathbf{X}\mathbf{w}$. Then:

$$\text{DR}(\mathbf{w}) = \frac{1}{T}\|(\mathbf{z})^+\|_2^2 = \frac{1}{T}\sum_{i=1}^{T} \tilde{z}_i^2,$$

where

$$\tilde{z}_i = \begin{cases} z_i, & \text{if } z_i > 0, \\ 0, & \text{if } z_i \leq 0. \end{cases}$$

Now, we can majorize each $\tilde{z}_i^2$ term to get an upper bound for $\mathrm{DR}(\mathbf{z})$. We need to consider two cases, i.e., majorization on a point $z_i^{(k)} > 0$ and on a point $z_i^{(k)} \leq 0$.

1. For a point $z_i^{(k)} > 0$, $f_1(z_i|z_i^{(k)}) = z_i^2$ is an upper bound of $\tilde{z}_i^2$, with $f_1(z_i^{(k)}|z_i^{(k)}) = \left(z_i^{(k)}\right)^2 = \left(\tilde{z}_i^{(k)}\right)^2$.

2. For a point $z_i^{(k)} \leq 0$, $f_2(z_i|z_i^{(k)}) = \left(z_i - z_i^{(k)}\right)^2$ is an upper bound of $\tilde{z}_i^2$, with $f_2(z_i^{(k)}|z_i^{(k)}) = \left(z_i^{(k)} - z_i^{(k)}\right)^2 = 0 = \left(\tilde{z}_i^{(k)}\right)^2$.

For both cases the proofs are straightforward and they are easily shown pictorially. Figure A.1 illustrates these two cases.

Now, we can majorize $\tilde{z}_i^2$ at any point $z_i^{(k)}$ as follows:

$$\begin{aligned} \tilde{z}_i^2 \leq & \begin{cases} f_1(z_i|z_i^{(k)}), & \text{if } z_i^{(k)} > 0, \\ f_2(z_i|z_i^{(k)}), & \text{if } z_i^{(k)} \leq 0, \end{cases} \\ = & \begin{cases} (z_i - 0)^2, & \text{if } z_i^{(k)} > 0, \\ (z_i - z_i^{(k)})^2, & \text{if } z_i^{(k)} \leq 0, \end{cases} \\ = & (z_i - y_i^{(k)})^2, \end{aligned}$$

where

$$\begin{aligned} y_i^{(k)} = & \begin{cases} 0, & \text{if } z_i^{(k)} > 0, \\ z_i^{(k)}, & \text{if } z_i^{(k)} \leq 0, \end{cases} \\ = & -(-z_i^{(k)})^+. \end{aligned}$$

Thus, $\mathrm{DR}(\mathbf{z})$ is majorized as follows:

$$\mathrm{DR}(\mathbf{w}) = \frac{1}{T}\sum_{i=1}^{T} \tilde{z}_i^2 \leq \frac{1}{T}\sum_{i=1}^{T}(z_i - y_i^{(k)})^2 = \frac{1}{T}\|\mathbf{z} - \mathbf{y}^{(k)}\|_2^2.$$

Substituting back $\mathbf{z} = \mathbf{r}^b - \mathbf{X}\mathbf{w}$, we get

$$\mathrm{DR}(\mathbf{w}) \leq \frac{1}{T}\|\mathbf{r}^b - \mathbf{X}\mathbf{w} - \mathbf{y}^{(k)}\|_2^2.$$

where $\mathbf{y}^{(k)} = -(-\mathbf{z}^{(k)})^+ = -(\mathbf{X}\mathbf{w}^{(k)} - \mathbf{r}^b)^+$. This completes the proof.

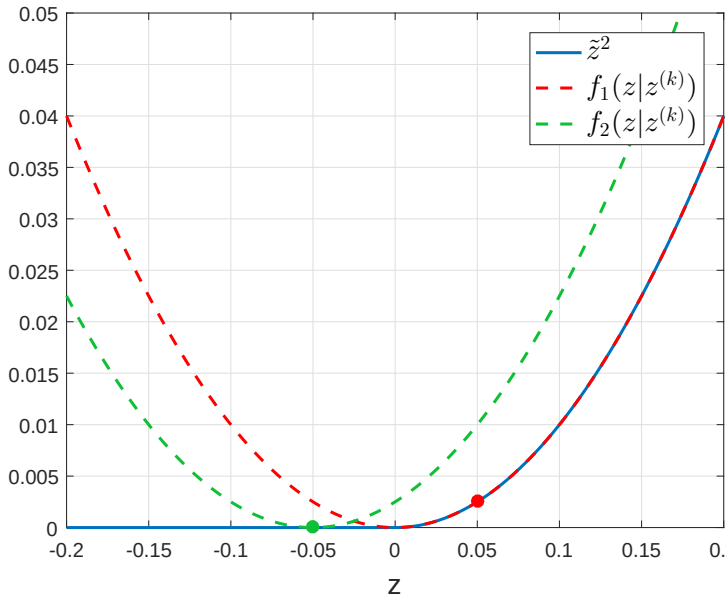


Figure A.1: Majorization cases of $\tilde{z}^2$: a) $z^{(k)} = 0.05 > 0$ and b) $z^{(k)} = -0.05 \leq 0$.

A.5 Proof of Lemma 4.5

We consider the two cases separately.

1. $|x^{(k)}| \leq M$: The majorization function becomes $f(x|x^{(k)}) = x^2$. It is straightforward that this is a majorizer of $\phi(x)$ since for $|x| \leq M$ we have $f(x|x^{(k)}) = \phi(x)$ and for $|x| > M$ it holds that $f(x|x^{(k)}) > \phi(x)$.

2. $|x^{(k)}| > M$: We bound the linear part of Huber with a quadratic function of the form $g(x) = ax^2 + b$. At any point of majorization $x^{(k)} > M$ it should hold that $\phi(x^{(k)}) = g(x^{(k)})$ and $\phi'(x^{(k)}) = g'(x^{(k)})$. From these two conditions it is easy to derive the results of Lemma 4.5.

A.6 Proof of Lemma 4.6

The proof for the cases $0 \leq x^{(k)} \leq M$ and $x^{(k)} > M$ follow similar arguments as the proof of Lemma 4.5 and therefore are omitted. To this end, we will provide a sketch of the proof for the case $x^{(k)} < 0$.

We consider a majorization function of the form $a(x-c)^2 + b$. We require that the majorizer is equal to the function $\phi((x)^+)$, defined in (4.39), at $x^{(k)} < 0$ and at one point $z \geq M$ so the bound is tight. We further require that the derivative of the majorizer, i.e., $2a(x-c)$, is equal to the derivative of (4.39) at $x^{(k)} < 0$ and $z \geq M$.

It is fairly easy to show that the only point $z \geq M$ that satisfies the above conditions is $z = M - x^{(k)}$. Taking this into account, it is straightforward to derive the result of Lemma 4.6.

A.7 Proof of Lemma 4.7

From Lemma 4.2 we have that $\rho_{p,l}(w) \leq d_{p,l}(w^{(k)})w + c_{p,l}(w^{(k)})$ for $w \geq 0$. Thus, for $\tilde{f}_{p,l}(w)$ we get:

$$\begin{aligned}\tilde{f}_{p,l}(w) &= \max\left(\rho_{p,l}(w)\cdot l - w, 0\right)\\ &\leq \max\left((d_{p,l}(w^{(k)})w + c_{p,l}(w^{(k)}))\cdot l - w, 0\right)\\ &= \max\left(\left(d_{p,l}(w^{(k)})\cdot l - 1\right)w + c_{p,l}(w^{(k)})\cdot l, 0\right).\end{aligned}$$

This function is convex since it is the maximum of two convex (actually affine) functions, i.e., $f_1 = \left(d_{p,l}(w^{(k)})\cdot l - 1\right)w + c_{p,l}(w^{(k)})\cdot l$ and $f_2 = 0$.

A.8 Proof of Lemma 4.8

Consider the concave function $f(x) = \sqrt{x}$ for $x \in [0, u]$. An upper bound of a concave function at any point x_0 is its first-order Taylor approximation, i.e.,

$$\sqrt{x} \leq \sqrt{x_0} + \frac{1}{2\sqrt{(x_0)}}(x - x_0).$$

By setting $x = (\alpha w + \beta)^2 + \epsilon^2$ we get the following bound:

$$\sqrt{(\alpha w + \beta)^2 + \epsilon^2}$$

$$\begin{aligned}
&\leq \sqrt{(\alpha w_0 + \beta)^2 + \epsilon^2} + \frac{\alpha^2(w^2 - w_0^2) + 2\alpha\beta(w - w_0)}{2\sqrt{((\alpha w_0 + \beta)^2 + \epsilon^2)}} \\
&= \frac{\alpha^2 w^2 + 2\alpha\beta w}{2\sqrt{((\alpha w_0 + \beta)^2 + \epsilon^2)}} + const.
\end{aligned}$$

By majorizing the square root term of $\tilde{h}_{p,\epsilon,l}(w, w^{(k)})$ following the aforementioned approach, the result of Lemma 4.8 is straightforward.

References

[1] C. Alexander. "Optimal hedging using cointegration". *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. 357(1758): 2039–2058, Aug. 1999.

[2] C. Alexander and A. Barbosa. "Effectiveness of minimum-variance hedging". *Journal of Portfolio Management*. 33(2): 46, Jan. 2007.

[3] C. Alexander and A. Barbosa. "Hedging index exchange traded funds". *Journal of Banking & Finance*. 32(2): 326–337, Feb. 2008.

[4] H. Almadi, D. Rapach, and A. Suri. "Return predictability and dynamic asset allocation: How often should investors rebalance?" *Journal of Portfolio Management*. 40(4): 16–27, July 2014.

[5] S. Andrade, V. Di Pietro, and M. Seasholes. "Understanding the profitability of pairs trading". *Unpublished working paper, UC Berkeley, Northwestern University*, 2005.

[6] K. Andriosopoulos, M. Doumpos, N. C. Papapostolou, and P. K. Pouliasis. "Portfolio optimization and index tracking for the shipping stock and freight markets using evolutionary algorithms". *Transportation Research Part E: Logistics and Transportation Review*. 52(June): 16–34, June 2013.

[7] A. Antoniou, G. Koutmos, and A. Pericli. "Index futures and positive feedback trading: Evidence from major stock exchanges". *Journal of Empirical Finance*. 12(2): 219–238, Mar. 2005.

[8] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. "Coherent measures of risk". *Mathematical Finance*. 9(3): 203–228, July 1999.

[9] B. M. Barber and T. Odean. "Trading is hazardous to your wealth: The common stock investment performance of individual investors". *The Journal of Finance*. 55(2): 773–806, Apr. 2000.

[10] J. E. Beasley, N. Meade, and T.-J. Chang. "An evolutionary heuristic for the index tracking problem". *European Journal of Operational Research*. 148(3): 621–643, Aug. 2003.

[11] K. Benidis, Y. Feng, and D. P. Palomar. "Sparse portfolios for high-dimensional financial index tracking". *IEEE Transactions on Signal Processing*. 66(1): 155–170, Jan. 2018.

[12] K. Benidis and D. P. Palomar. *sparseIndexTracking: Design of Portfolio of Stocks to Track an Index*. R package version 0.1.0. URL: https://CRAN.R-project.org/package=sparseIndexTracking.

[13] K. Benidis, Y. Sun, P. Babu, and D. P. Palomar. "Orthogonal sparse PCA and covariance estimation via Procrustes reformulation". *IEEE Transactions on Signal Processing*. 64(23): 6211–6226, Dec. 2016.

[14] D. Bertsimas, A. King, and R. Mazumder. "Best subset selection via a modern optimization lens". *The Annals of Statistics*. 44(2): 813–852, 2016.

[15] D. Bianchi and A. Gargano. "High-dimensional index tracking with cointegrated assets using an hybrid genetic algorithm". *Available at: http://ssrn.com/abstract*. 1785908, 2011.

[16] S. Boyd, E. Busseti, S. Diamond, R. N. Kahn, K. Koh, P. Nystrup, J. Speth, *et al.* "Multi-period trading via convex optimization". *Foundations and Trends® in Optimization*. 3(1): 1–76, Aug. 2017.

[17] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press.

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

[19] N. A. Canakgoz and J. E. Beasley. "Mixed-integer programming approaches for index tracking and enhanced indexation". *European Journal of Operational Research*. 196(1): 384–399, July 2009.

[20] E. J. Candès and M. B. Wakin. "An introduction to compressive sampling". *IEEE Signal Processing Magazine*. 25(2): 21–30, Mar. 2008.

[21] E. J. Candès, M. B. Wakin, and S. P. Boyd. "Enhancing sparsity by reweighted ℓ_1 minimization". *Journal of Fourier Analysis and Applications*. 14(5-6): 877–905, Dec. 2008.

[22] T.-J. Chang, N. Meade, J. E. Beasley, and Y. M. Sharaiha. "Heuristics for cardinality constrained portfolio optimisation". *Computers & Operations Research*. 27(13): 1271–1302, Nov. 2000.

[23] R. Chartrand and W. Yin. "Iteratively reweighted algorithms for compressive sensing". In: 3869–3872.

[24] S.-H. Chu and S. Freund. "Volatility estimation for stock index options: a GARCH approach". *The Quarterly Review of Economics and Finance*. 36(4): 431–450, Dec. 1996.

[25] T. F. Coleman, Y. Li, and J. Henniger. "Minimizing tracking error while restricting the number of assets". *Journal of Risk*. 8(4): 33, July 2006.

[26] G. M. Constantinides, J. C. Jackwerth, and S. Perrakis. "Mispricing of S&P 500 index options". *The Review of Financial Studies*. 22(3): 1247–1277, Feb. 2008.

[27] R. Cont and T. Kokholm. "A consistent pricing model for index options and volatility derivatives". *Mathematical Finance*. 23(2): 248–274, Apr. 2013.

[28] B. Cornell and K. R. French. "The pricing of stock index futures". *Journal of Futures Markets*. 3(1): 1–14, Mar. 1983.

[29] V. DeMiguel, L. Garlappi, F. J. Nogales, and R. Uppal. "A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms". *Management Science*. 55(5): 798–812, May 2009.

[30] C. Dose and S. Cincotti. "Clustering of financial time series with application to index and enhanced index tracking portfolio". *Physica A: Statistical Mechanics and its Applications*. 355(1): 145–151, Sept. 2005.

[31] C. L. Dunis and R. Ho. "Cointegration portfolios of European equities for index tracking and market neutral strategies". *Journal of Asset Management*. 6(1): 33–52, June 2005.

[32] J. Evnine and A. Rudd. “Index options: The early evidence”. *The Journal of Finance*. 40(3): 743–756, July 1985.

[33] J. Fan and R. Li. “Variable selection via nonconcave penalized likelihood and its oracle properties”. *Journal of the American Statistical Association*. 96(456): 1348–1360, Dec. 2001.

[34] B. Fastrich, S. Paterlini, and P. Winker. “Constructing optimal sparse portfolios using regularization methods”. *Computational Management Science*. 12(3): 417–434, July 2015.

[35] Y. Feng and D. P. Palomar. “A Signal Processing Perspective on Financial Engineering”. *Foundations and Trends® in Signal Processing*. 9(1–2): 1–231, Aug. 2016.

[36] M. A. T. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak. “Majorization–minimization algorithms for wavelet-based image restoration”. *IEEE Transactions on Image Processing*. 16(12): 2980–2991, Dec. 2007.

[37] S. L. Fuller. “The evolution of actively managed exchange-traded funds”. *The Review of Securities and Commodities Regulation*. 41(8): 89–96, Apr. 2008.

[38] A. A. Gaivoronski, S. Krylov, and N. Van der Wijst. “Optimal portfolio selection and dynamic benchmark tracking”. *European Journal of Operational Research*. 163(1): 115–131, May 2005.

[39] A. A. Gaivoronski and G. Pflug. “Value-at-risk in portfolio optimization: properties and computational approach”. *Journal of Risk*. 7(2): 1–31, Dec. 2004.

[40] G. L. Gastineau and C. Weber. “Hedging exchange traded mutual funds or other portfolio basket products”. US Patent 7,099,838.

[41] E. Gatev, W. N. Goetzmann, and K. G. Rouwenhorst. “Pairs trading: Performance of a relative-value arbitrage rule”. *The Review of Financial Studies*. 19(3): 797–827, Feb. 2006.

[42] I. F. Gorodnitsky and B. D. Rao. “Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm”. *IEEE Transactions on Signal Processing*. 45(3): 600–616, Mar. 1997.

[43] M. Grant, S. Boyd, and Y. Ye. “CVX: Matlab software for disciplined convex programming”, 2008.

[44] D. R. Hunter and K. Lange. "A tutorial on MM algorithms". *The American Statistician*. 58(1): 30–37, Feb. 2004.

[45] R. Jansen and R. Van Dijk. "Optimal benchmark tracking with small portfolios". *The Journal of Portfolio Management*. 28(2): 33–39, Jan. 2002.

[46] I. G. Kawaller, P. D. Koch, and T. W. Koch. "The temporal price relationship between S&P 500 futures and the S&P 500 index". *The Journal of Finance*. 42(5): 1309–1329, Dec. 1987.

[47] H. Kellerer, R. Mansini, and M. G. Speranza. "Selecting portfolios with fixed costs and minimum transaction lots". *Annals of Operations Research*. 99(1-4): 287–304, Dec. 2000.

[48] D. Kuhn and D. G. Luenberger. "Analysis of the rebalancing frequency in log-optimal portfolio selection". *Quantitative Finance*. 10(2): 221–234, Feb. 2010.

[49] Z. M. Landsman and E. A. Valdez. "Tail conditional expectations for elliptical distributions". *North American Actuarial Journal*. 7(4): 55–71, Oct. 2003.

[50] M. S. Lobo, M. Fazel, and S. Boyd. "Portfolio optimization with linear and fixed transaction costs". *Annals of Operations Research*. 152(1): 341–365, July 2007.

[51] A. W. Lynch and P. Balduzzi. "Predictability and transaction costs: The impact on rebalancing rules and behavior". *The Journal of Finance*. 55(5): 2285–2309, Oct. 2000.

[52] D. Maringer and O. Oyewumi. "Index tracking with constrained portfolios". *Intelligent Systems in Accounting, Finance and Management*. 15(1-2): 57–71, Jan. 2007.

[53] H. Markowitz. "Portfolio selection". *The Journal of Finance*. 7(1): 77–91, Mar. 1952.

[54] A. Mosek. "The MOSEK optimization toolbox for MATLAB manual", 2015.

[55] Y. E. Nesterov and M. J. Todd. "Self-scaled barriers and interior-point methods for convex programming". *Mathematics of Operations Research*. 22(1): 1–42, Feb. 1997.

[56] K. J. Oh, T. Y. Kim, and S. Min. "Using genetic algorithm to support portfolio optimization for index fund management". *Expert Systems with Applications*. 28(2): 371–379, Feb. 2005.

[57] D. P. Palomar and J. R. Fonollosa. "Practical algorithms for a family of waterfilling solutions". *IEEE Transactions on Signal Processing*. 53(2): 686–695, Feb. 2005.

[58] N. R. Patel and M. G. Subrahmanyam. "A simple algorithm for optimal portfolio selection with fixed transaction costs". *Management Science*. 28(3): 303–314, Mar. 1982.

[59] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: http://www.R-project.org/.

[60] M. Razaviyayn, M. Hong, and Z.-Q. Luo. "A unified convergence analysis of block successive minimization methods for nonsmooth optimization". *SIAM Journal on Optimization*. 23(2): 1126–1153, June 2013.

[61] R. T. Rockafellar and S. Uryasev. "Optimization of conditional value-at-risk". *Journal of Risk*. 2(Apr.): 21–42, Apr. 2000.

[62] G. G. Rompotis. "Active vs. passive management: New evidence from exchange traded funds", 2009.

[63] R. Ruiz-Torrubiano and A. Suárez. "A hybrid optimization approach to index tracking". *Annals of Operations Research*. 166(1): 57–71, Feb. 2009.

[64] A. A. P. Santos. "Beating the market with small portfolios: Evidence from Brazil". *EconomiA*. 16(1): 22–31, Jan. 2015.

[65] A. Scozzari, F. Tardella, S. Paterlini, and T. Krink. "Exact and heuristic approaches for the index tracking problem with UCITS constraints". *Annals of Operations Research*. 205(1): 235–250, May 2013.

[66] J. Song, P. Babu, and D. P. Palomar. "Optimization methods for designing sequences with low autocorrelation sidelobes". *IEEE Transactions on Signal Processing*. 63(15): 3998–4009, Aug. 2015.

[67] J. Song, P. Babu, and D. P. Palomar. "Sparse generalized eigenvalue problem via smooth optimization". *IEEE Transactions on Signal Processing*. 63(7): 1627–1642, Apr. 2015.

[68] B. K. Sriperumbudur, D. A. Torres, and G. R. G. Lanckriet. "A majorization-minimization approach to the sparse generalized eigenvalue problem". *Machine Learning*. 85(1-2): 3–39, Oct. 2011.

[69] A. Subrahmanyam. "A theory of trading in stock index futures". *The Review of Financial Studies*. 4(1): 17–51, Jan. 1991.

[70] Y. Sun, P. Babu, and D. P. Palomar. "Majorization - minimization algorithms in signal processing, communications, and machine learning". *IEEE Transactions on Signal Processing*. 65(3): 794–816, Feb. 2016.

[71] A. Takeda, M. Niranjan, J.-Y. Gotoh, and Y. Kawahara. "Simultaneous pursuit of out-of-sample performance and sparsity in index tracking portfolios". *Computational Management Science*. 10(1): 21–49, Feb. 2013.

[72] R. Tibshirani. "Regression shrinkage and selection via the lasso". *Journal of the Royal Statistical Society. Series B (Methodological)*. Jan.: 267–288, Jan. 1996.

[73] R. Varadhan and C. Roland. "Simple and globally convergent methods for accelerating the convergence of any EM algorithm". *Scandinavian Journal of Statistics*. 35(2): 335–353, June 2008.

[74] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. "Robust face recognition via sparse representation". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 31(2): 210–227, Feb. 2009.

[75] F. Xu, Z. Xu, and H. Xue. "Sparse Index Tracking Based On $L_{1/2}$ Model And Algorithm". *arXiv preprint arXiv:1506.05867*, 2015.

[76] J. Yang, Z. Yang, and Y. Zhou. "Intraday price discovery and volatility transmission in stock index and stock index futures markets: Evidence from China". *Journal of Futures Markets*. 32(2): 99–121, Feb. 2012.

[77] C.-H. Zhang. "Nearly unbiased variable selection under minimax concave penalty". *The Annals of Statistics*. 38(2): 894–942, 2010.

[78] H. Zou, T. Hastie, and R. Tibshirani. "Sparse principal component analysis". *Journal of Computational and Graphical Statistics*. 15(2): 265–286, June 2006.