Quantitative Finance

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/rquf20

High-dimensional sparse index tracking based on a multi-step convex optimization approach

Fangquan Shi, Lianjie Shu, Yiling Luo & Xiaoming Huo

To cite this article: Fangquan Shi, Lianjie Shu, Yiling Luo & Xiaoming Huo (2023) High-dimensional sparse index tracking based on a multi-step convex optimization approach, Quantitative Finance, 23:9, 1361-1372, DOI: 10.1080/14697688.2023.2236158

To link to this article: https://doi.org/10.1080/14697688.2023.2236158

Quantitative Finance, 2023
Vol. 23, No. 9, 1361–1372, https://doi.org/10.1080/14697688.2023.2236158

High-dimensional sparse index tracking based on a multi-step convex optimization approach

FANGQUAN SHI†, LIANJIE SHU*‡, YILING LUO§ and XIAOMING HUO§

†School of Finance, Nanjing Audit University, Nanjing, People's Republic of China
‡Faculty of Business Administration, University of Macau, Taipa, People's Republic of China
§School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

(*Received 24 August 2022; accepted 7 July 2023; published online 3 August 2023*)

Both convex and non-convex penalties have been widely proposed to tackle the sparse index tracking problem. Owing to their good property of generating sparse solutions, penalties based on the least absolute shrinkage and selection operator (LASSO) and its variations are often suggested in the stream of convex penalties. However, the LASSO-type penalty is often shown to have poor out-of-sample performance, due to the relatively large biases introduced in the estimates of tracking portfolio weights by shrinking the parameter estimates toward to zero. On the other hand, non-convex penalties could be used to improve the bias issue of LASSO-type penalty. However, the resulting problem is non-convex optimization and thus is computationally intensive, especially in high-dimensional settings. Aimed at ameliorating bias introduced by LASSO-type penalty while preserving computational efficiency, this paper proposes a multi-step convex optimization approach based on the multi-step weighted LASSO (MSW-LASSO) for sparse index tracking. Empirical results show that the proposed method can achieve smaller out-of-sample tracking errors than those based on LASSO-type penalties and have performance competitive to those based on non-convex penalties.

Keywords: Finance; Index tracking; Sparsity; Cardinality; LASSO

1. Introduction

In recent years, index-based mutual funds and exchange-traded funds that can track a specified market benchmark have grown quickly, which have gained about 14% market shares in assets under management and about 22% in 2010 (Cremers *et al.* 2016). The growth is probably due to the empirical finding that the actively managed portfolios usually do not outperform the market in the long run after taking into account transaction and administrative costs (Sharpe 1991, Malkiel and Xu 1997, Barber and Odean 2000). Compared with actively managed funds, the index-based funds are passive investments, which are simple to construct and rebalance.

There are two different ways to construct tracking portfolios to replicate the performance of a given market benchmark. A natural way, called full replication, is to hold constitutes of the tracking portfolio in the same weights as that of the given target. The tracking portfolio constructed in this way can track the target index very closely. However, it is only suited for an index with a small number of stocks. For an index with many illiquid stocks, the full replication approach can be expensive and thus impracticable, due to high transaction and administrative costs. Thus, in order to reduce transaction and management costs, the second way known as partial index tracking is often employed in practice. The partial index tracking aims at purchasing a small number of stocks that can track the target index as closely as possible.

The partial index tracking involves the determination of the best subset of assets. This is a challenging problem as there are so many possible combinations of assets. It is nearly impossible to try out all combinations of assets and choose the best in practice even when there is a moderate number of stocks in the benchmark index. There are two tasks in partial index tracking: selecting the subset of assets and determining the optimal portfolio weights. Depending on whether these two tasks are completed separately or simultaneously, the approaches to solving the index tracking problem can be grouped into two categories: a sequential approach and a unified approach. The first category completes the two tasks in two independent steps, i.e. to first determine the subset of assets based on some intuitions and then to estimate the optimal weights by

*Corresponding author. Email: ljshu@um.edu.mo

minimizing a tracking error measure. See, for example, Jansen and Van Dijk (2002) and Montfort *et al.* (2008).

Although this two-step approach can reduce computation significantly, it is unclear how optimal the resulting tracking portfolio is Benidis *et al.* (2018). For this reason, some unified approaches have been recently proposed to index tracking, aimed at unifying the two tasks of asset selection and weights allocation. Regularization techniques have been popularly used in this category, which involves the minimization of a given tracking error measure while imposing a penalty function on the tracking portfolio weights. Both convex and non-convex penalty functions have been suggested. Owing to its good property of generating sparse solutions, the ℓ_1-penalty based on the adaptation of the least absolute shrinkage and selection operator (LASSO) by Tibshirani (1996) and its variations have been recently proposed to tackle the sparse index tracking problem. For example, Giamouridis and Paterlini (2010) introduced the ℓ_1-norm and ℓ_2-norm on the asset weights in the context of efficient hedge fund investments replication. Some recent applications include Wu *et al.* (2014), Wu and Yang (2014), and Yang and Wu (2016). Analogously, Li (2015) studied the use of the ℓ_1-norm and ℓ_2-norm penalties to obtain sparse and stable portfolios. Recently, Kremer *et al.* (2022) proposed a model based on sorted ℓ_1 penalized estimator (SLOPE) for index tracking and hedge fund replication, originating from the work of Jiang *et al.* (2022). The basic idea is to use a sequence of tuning parameters for penalizing the assets according to their rank magnitude.

The LASSO-type penalty is convex in form. This leads to a prominent advantage of using the LASSO-type penalty in index tracking. In particular, it offers the possibility to select the portfolio constituents and estimate the asset weights in a single step by solving a convex optimization problem. The resulting optimization problem can be easily solved by either quadratic programming (Tibshirani 1996), the homotopy approach (Osborne *et al.* 2000), coordinate wise optimization (Friedman *et al.* 2007), or gradient projection method (Figueiredo *et al.* 2007a). However, Fan and Li (2001) showed that the LASSO penalty tends to produce biased estimates for large coefficients. In high-dimensional sparse index tracking, when the LASSO penalty is used to generate a very sparse solution, a larger regularization parameter needs to be selected. This leads to larger biases in the estimation of asset weights. Intuitively, the large biases introduced in the asset weights could adversely affect the out-of-sample tracking performance.

To overcome this bias issue and perform well out-of-sample with much sparser solutions, non-convex penalty functions have been suggested. The most straightforward one is the cardinality penalty, which directly limits the maximum number of assets held in the tracking portfolio. However, the resulting index tracking problem is shown to be NP-hard in Huo and Ni (2007), Huo and Chen (2010), and Ruiz-Torrubiano and Suárez (2009). This presents computational challenges, especially in high-dimensional settings, due to the non-differentiability of the search space. In order to speed up the solution process with the cardinality constraint, some heuristic algorithms have been proposed. See, for example, Gilli and Këllezi (2002), Maringer and Oyewumi (2007), Krink *et al.* (2009), Scozzari *et al.* (2013), Guastaroba and Speranza (2012), Takeda *et al.* (2013), and Strub and Trautmann (2019). In addition to the cardinality penalty, some other non-convex penalties have been also discussed. A sample of research in this category includes the use of the ℓ_q penalty (Fastrich *et al.* 2014, Xu *et al.* 2015), the log-penalty (Giuzio *et al.* 2016), the modified log-penalty (Benidis *et al.* 2018), and the smoothly clipped absolute deviation (SCAD) penalty (Fastrich *et al.* 2015). Due to the complexity of the resulting optimization problems with non-convex penalties, some heuristic search algorithms are often employed to solve these problems. However, these heuristic algorithms have no guarantee of the optimality of the solution. In general, they only find an approximate solution to the optimization problem.

Aimed at countervailing bias while retaining convexity of the optimization problem, the objective of this paper is to propose a new approach based on the multi-step weighted LASSO (MSW-LASSO) for sparse index tracking in high-dimensional settings. MSW-LASSO was originally proposed in statistics by Lobo *et al.* (2007), Bühlmann and Meier (2008), Candes *et al.* (2008), and Zou and Li (2008). The proposed approach works in an iterative way. In each iteration, it uses the solution in the last iteration to update the weights in the current step. The iteration procedure stops when the solution converges. In each iteration, the solution can be obtained by solving a convex optimization problem. Moreover, the sparsity of the solution tends to increase in each iteration. Therefore, the MSW-LASSO approach is a multi-step convex optimization approach, which can maintain the computational efficiency as the traditional LASSO-based regularization methods but improves the bias issue over the latter.

The proposed approach has a close relationship with the existing approaches. It is a generalization of the traditional one-step LASSO approach for index tracking. Clearly, it includes the traditional LASSO and adaptive LASSO (ALASSO) as special cases when one-step and two-step iterations are performed, respectively. Also, it can be viewed as an iterative algorithm based on local linear approximation for solving index tracking problems with non-convex penalties. The weight in each iteration is equal to the derivative of the non-convex penalty function. In this sense, the proposed approach can bridge the gap between the methods based on convex and non-convex penalties in index tracking.

2. The multi-step weighted LASSO approach

The sparse index tracking aims at replicating a target index with a smaller number of constituents. The common way is to tackle the problem with a cardinality constraint, which can be formulated as Fastrich *et al.* (2014)

$$\min_{\beta} \quad \frac{1}{n}\|y - X\beta\|_2^2, \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{p} \beta_j = 1, \tag{2}$$

$$\beta_j \geq 0, \quad (3)$$

$$\#\Gamma \leq k_{\text{act}}, \quad (4)$$

where $y = [y_1, \ldots, y_n]^T$ is the $n \times 1$ vector of index returns, $X = (x_{ij})_{n\times p}$ is the $n \times p$ matrix of returns on the p index constituents in n time periods, and $\beta = [\beta_{1,\ldots,}\beta_p]^T$ is the $p \times 1$ weight vector to be determined for minimizing the tracking error in (1). Constraints in (2) and (3) denote the full investment condition and no-short sale constraint, respectively. The cardinality constraint in (4) restricts the number of active positions $\#\Gamma(\Gamma = \{j \in \{1, 2, \ldots, p\} | \beta_j > 0\})$ to be at most equal to $k_{\text{act}} \in \{1, 2, \ldots, p\}$.

The above problem can be reformulated as the mixed 0–1 programming problem below:

$$\begin{aligned}
\min_{\beta} \quad & \frac{1}{n}\|y - X\beta\|_2^2, \\
\text{s.t.} \quad & \sum_{j=1}^{p} \beta_j = 1, \\
& \sum_{j=1}^{p} z_j \leq k_{\text{act}}, \\
& 0 \leq \beta_j \leq z_j, \\
& z_j \in \{0, 1\},
\end{aligned}$$

which can be solved using solvers like Gurobi. However, the cardinality constraint makes the above problem NP-hard. This leads to computational challenges, especially when p is relatively large.

As an alternative to obtain a sparse solution, some regularization techniques have been suggested for sparse index tracking by imposing a penalty of the portfolio weights, $\sum_{j=1}^{p} \text{pen}(\beta_j)$. Due to the nice properties of variable selection and computational efficiency, the convex penalty based on the LASSO penalty, also known as the ℓ_1-norm penalty, has been widely discussed. The objective function with the LASSO penalty becomes

$$\min_{\beta} \quad \frac{1}{n}\|y - X\beta\|_2^2 + \sum_{j=1}^{p} p_{\lambda_j}(\beta_j) = \frac{1}{n}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1,$$

where $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$ is the ℓ_1-norm of β. In principle, the penalty $p_{\lambda_j}(\beta_j)$ can be different for different components. For ease of presentation, here we consider $p_{\lambda_j}(\beta_j) = p_\lambda(\beta_j)$, implying that the same penalty function is applied to every component of β.

It is interesting to note that under the constraints $\sum_{j=1}^{p} \beta_j = 1$ and $\beta_j \geq 0$, $\sum_{j=1}^{p} |\beta_j| = 1$. Therefore, the ℓ_1-norm of the portfolio weights is a constant of one in this case, implying that the regularization parameter λ cannot control the sparsity of the solution. For this reason, the classical LASSO penalty is inefficient in promoting sparsity in index tracking with both the full investment and no-short selling constraints. Although Wu *et al.* (2014) discussed LASSO for index tracking, their study is limited to the case with no-short selling constraint only. The full investment constraint was not considered in their study.

In order to make the LASSO-type penalty generate sparser solutions in index tracking, a variation needs to be made. In particular, one can use the ALASSO penalty, which is a weighted ℓ_1-norm of the portfolio weights. The ALASSO penalty is defined as

$$\sum_{j=1}^{p} p_\lambda(\beta_j) = \lambda \sum_{j=1}^{p} \hat{w}_j |\beta_j|.$$

Zou (2006) suggested choosing the weight on $|\beta_j|$ as $\hat{w}_j = \frac{1}{|\hat{\beta}_j^{\text{init}}|^\nu}$, where ν is usually set to be 1, and $\hat{\beta}^{\text{init}}$ is the initial estimate of β that can be computed by solving (1)–(3). Clearly, ALASSO includes LASSO as a special case with $\hat{w}_j = 1$ for $j = 1, \ldots p$. For more details about the ALASSO, one can refer to Zou (2006).

Although the ALASSO approach can generate sparse solutions in index tracking, the regularization typically needs a large regularization parameter to shrink many of the near-to-zero weights towards zero, especially in high-dimensional settings. But a large regularization parameter implies substantial shrinkage to zero, leading to larger biases in estimating the tracking portfolio weights. This in turn could deteriorate the out-of-sample tracking performance. In order to reduce biases in the tracking portfolio weight estimation, we suggest the MSW-LASSO approach to generalize the ALASSO approach.

The MSW-LASSO procedure consists of the following steps:

(1) Initialize the weights $w_j^{(0)} = w_0$ for $j = 1, \ldots, p$;
(2) For each iteration $k = 1, 2, \ldots, M$, solve the following index tracking problem:

$$\hat{\beta}^{(k)} = \arg\min_{\beta} \left\{ \|y - X\beta\|_2^2 + \sum_{j=1}^{p} w_j^{(k-1)} |\beta_j| \right\}, \quad (5)$$

$$\text{s.t.} \quad \sum_{j=1}^{p} \beta_j = 1, \quad (6)$$

$$\beta_j \geq 0, \quad (7)$$

where $w_j^{(k-1)}$ is the weight on $|\beta_j|$ in the $(k-1)$th iteration.
(3) Repeat above steps until the solution converges.

The problem (5)–(7) is essentially a quadratic programming problem. General algorithms for convex optimization could be applied to solve it. In this paper, we use the interior point method for solving it, which is quite efficient and can solve the optimization problem in polynomial time. In particular, the R package, **LowRankQP**, was employed to solve the problem (5)–(7). Moreover, following Candes *et al.* (2008), the initial weights are set to be $w_j^{(0)} = w_0 = 1$ $(j = 1, \ldots, p)$ without any prior knowledge about $\beta_j^{(0)}$.

Proposition 1 below shows that the MSW-LASSO procedure with $w_j^{(k-1)} = |p'_\lambda(\hat{\beta}_j^{(k-1)})|$ can be viewed as an iterative algorithm based on the local linear approximation for minimizing the optimization problem with a penalty $\sum_{j=1}^{p} p_\lambda(\beta_j)$, where $p'_\lambda(\beta_j)$ denotes the derivative of the penalty $p_\lambda(\beta_j)$. The proof of Proposition 1 is shown in Appendix.

PROPOSITION 1 *Consider a concave penalty function $p_\lambda(\beta_j)$ defined on $[0, \infty)$, the MSW-LASSO procedure with the weighting function $w_j^{(k-1)} = p'_\lambda(\hat{\beta}_j^{(k-1)})$ is equivalent to solving the following optimization problem*

$$\min_{\beta} \quad f(\beta) = \left\{ \|Y - X\beta\|_2^2 + \sum_{j=1}^{p} p_\lambda(\beta_j) \right\}, \tag{8}$$

$$\text{s.t.} \quad \beta \in \mathcal{B} = \left\{ \beta : \beta \geq 0, \sum_{j=1}^{p} \beta_j = 1 \right\}, \tag{9}$$

where each step of the MSW-LASSO performs an iterative local linear approximation with the penalty $\sum_{j=1}^{p} p_\lambda(\beta_j)$.

In view of the smaller bias achieved by the non-convex penalties over the convex penalties, we suggest some non-convex penalty functions for $w_j^{(k-1)} = |p'_\lambda(\hat{\beta}_j^{(k-1)})|$ in equation (5) of the MSW-LASSO procedure. A number of non-convex penalty functions have been suggested in the literature. For example, Fan and Li (2001) proposed the SCAD penalty of the form

$$p_{\lambda,\mathrm{SCAD}}(\beta_j) = \begin{cases} \lambda|\beta_j| & |\beta_j| \leq \lambda, \\ -(\beta_j^2 - 2a\lambda|\beta_j| + \lambda^2)/ & \\ \quad [2(a-1)] & \lambda < |\beta_j| \leq a\lambda, \\ (a+1)\lambda^2/2 & |\beta_j| > a\lambda, \end{cases}$$

where $a > 2$, and the usual choice for the a parameter is $a = 3.7$. The SCAD penalty coincides with the LASSO penalty until $|\beta_j| = \lambda$, then smoothly changes to a quadratic function until $|\beta_j| = a\lambda$, after which it remains a constant. Benidis *et al.* (2018) considered a modified log-penalty function to approximate the ℓ_0-penalty function, given by

$$p_{\lambda,\mathrm{Log-M}}(\beta_j) = \lambda \frac{\log(1 + |\beta_j|/\varepsilon)}{\log(1 + 1/\varepsilon)},$$

where $0 < \varepsilon \ll 1$ is a parameter. Zhang (2010) suggested the minimax concave penalty (MCP) of the form

$$p_{\lambda,\mathrm{MCP}}(\beta_j) = \begin{cases} \lambda|\beta_j| - \frac{\beta_j^2}{2b}, & |\beta_j| \leq b\lambda, \\ \frac{1}{2}b\lambda^2, & |\beta_j| > b\lambda, \end{cases}$$

where $b > 1$. Fastrich *et al.* (2014) considered the ℓ_q penalty (or bridge penalty) for index tracking, given by

$$p_{\lambda,\ell_q}(\beta_j) = \lambda|\beta_j|^q,$$

where $0 < q < 1$.

For illustration, the penalty functions $\sum_{i=1}^{2} p_\lambda(\beta_j)$ based on SCAD, MCP, Log-M, and ℓ_q with constraints $\beta_1 \geq 0, \beta_2 \geq 0$, and $\beta_1 + \beta_2 = 1$ are displayed in figure 1. Notice that the ℓ_1 penalty increases linearly with $|\beta_j|$. To obtain sparser solutions, larger values of the tuning parameter are required. This would assign undesirable large penalties onto relatively large coefficients that are unlikely to be zero, leading to increased bias in the solution. In contrast, the non-convex penalties assign non-linear penalties onto β_j. As can be seen from figure 1, the SCAD, MCP, Log-M, and ℓ_q penalties increase very quickly for close-to-zero coefficients. Different from the log-M and ℓ_q penalties, both SCAD and MCP penalties remain unchanged after the coefficients increase to a certain threshold.

The derivatives of SCAD, Log-M, MCP, and ℓ_q penalty functions with respect to β_j ($\beta_j \neq 0$) are given by

$$p'_{\lambda,\mathrm{SCAD}}(\beta_j) = \begin{cases} \mathrm{sign}(\beta_j)\lambda, & |\beta_j| \leq \lambda, \\ \mathrm{sign}(\beta_j)\frac{a\lambda - |\beta_j|}{a-1}, & \lambda < |\beta_j| \leq a\lambda, \\ 0, & |\beta_j| > a\lambda, \end{cases}$$

$$p'_{\lambda,\mathrm{Log-M}}(\beta_j) = \lambda \frac{\mathrm{sign}(\beta_j)}{\varepsilon(1 + |\beta_j|/\varepsilon)\log(1 + 1/\varepsilon)},$$

$$p'_{\lambda,\mathrm{MCP}}(\beta_j) = \begin{cases} \mathrm{sign}(\beta_j)\frac{b\lambda - |\beta_j|}{b}, & |\beta_j| \leq b\lambda, \\ 0, & |\beta_j| > b\lambda, \end{cases}$$

and

$$p'_{\lambda,\ell_q}(\beta_j) = \lambda \mathrm{sign}(\beta_j)|\beta_j|^{q-1}1(\beta_j \neq 0),$$

respectively. Note that the above penalties are not differentiable at $\beta_j = 0$. In each iteration of the MSW-LASSO procedure with $\beta_j^{(k-1)} = 0$, one can set $w_j^{(k-1)} = \lambda$ for SCAD, $w_j^{(k-1)} = \lambda/(\varepsilon\log(1 + 1/\varepsilon))$ for Log-M, $w_j^{(k-1)} = \lambda$ for MCP, and $w_j^{(k-1)} = \infty$ for ℓ_q penalty, respectively. As for ℓ_q penalty, if $w_j^{(k-1)} = \infty$, then $\beta_j^{(k)}$ is set to zero. For more details about the solution procedure, please refer to Section 2.8.6 of Bühlmann and Van De Geer (2011).

Based on the updating rule of the MSW-LASSO approach, we see that it reduces to the classical LASSO and ALASSO approaches when it performs one-step iteration and two-step iterations, respectively. When $k = 1$, $w_j^{(k-1)} = 1$ and the penalty on the tracking portfolio weights reduces to the LASSO penalty, as can be seen from problem (5)–(7). When $k = 2$, the penalty on the tracking portfolio weights reduces to the ALASSO penalty.

The following simulation demonstrates that MSW-LASSO with weights derived from the MCP penalty can improve the bias issue over ALASSO. We use the daily return data of the S&P 100 data set during the period 01/01/2016 to 31/12/2016 to generate the simulated index return. In particular, the true weights, β^*, for the 99 stocks in the S&P 100 data set are assigned with the scaled weights of constituents in the S&P 100 index. The target index return can then be computed based on

$$y_t = \beta^* r_t$$

where r_t is the 99×1 vector of returns on the constituents. Two cases with $k_{\mathrm{act}} = 10$ and 20 active stocks selected were considered.

Figures 2(a,b) plot the number of stocks selected and bias in the tracking portfolio weights obtained based on the cardinality constraint approach, ALASSO, and MSW-LASSO when $k_{\mathrm{act}} = 10$ and 20, respectively. The bias in the estimated portfolio weights is measured by the ℓ_1 distance between the estimated and true portfolio weights, namely, $\|\hat{\beta} - \beta^*\|_1 =$

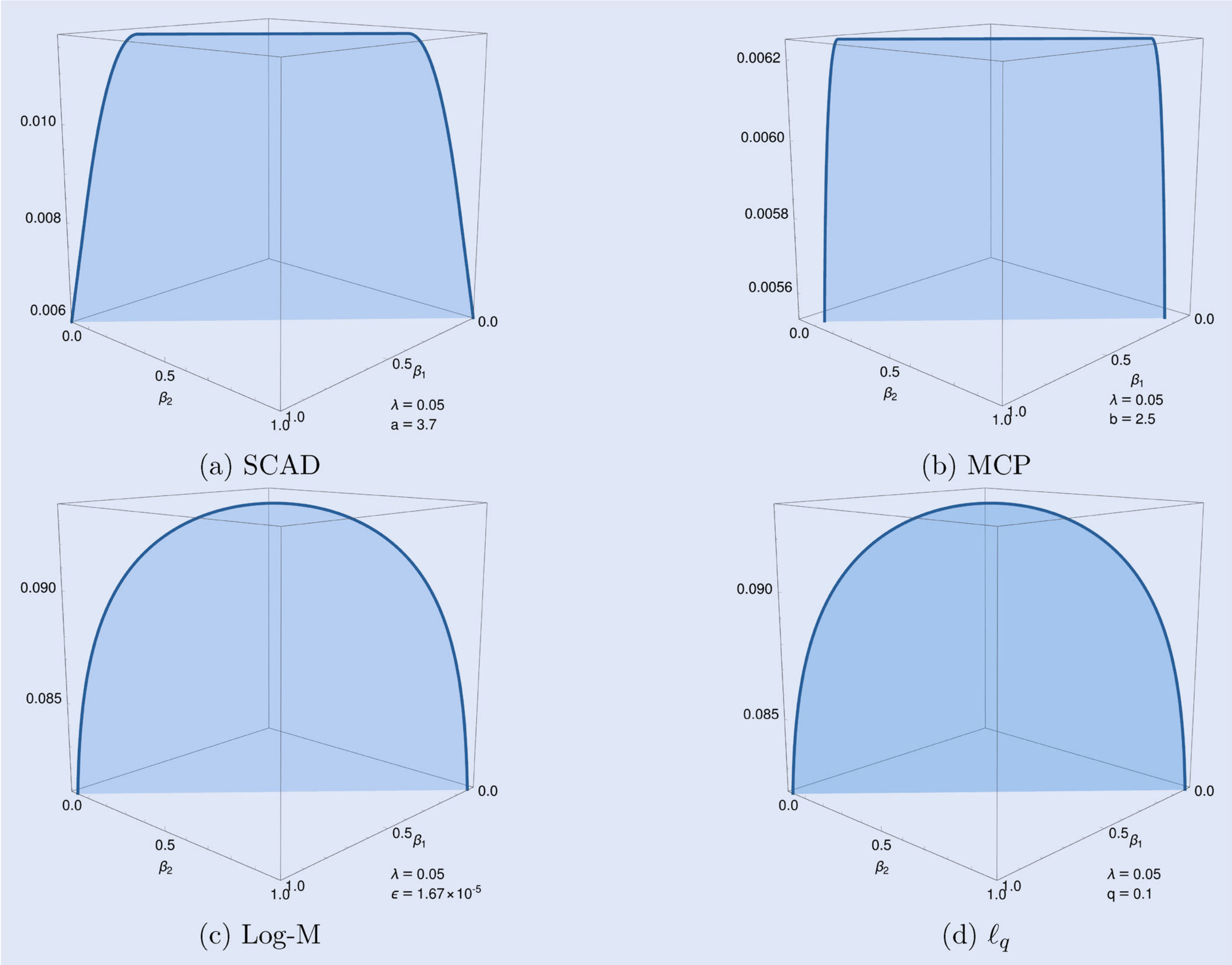

Figure 1. The penalty function $p_\lambda(\beta_1) + p_\lambda(\beta_2)$ with constraints $\beta_1 \geq 0$, $\beta_2 \geq 0$, and $\beta_1 + \beta_2 = 1$ based on (a) SCAD; (b) MCP; (c) Log-M; and (4) ℓ_q.

$\sum_{j=1}^{p} |\beta_j - \beta_j^*|$. To simplify the discussion, the weight for MSW-LASSO is based on the derivative of the MCP penalty function. From figure 2(b), when there are 20 stocks are selected to replicate the target index performance, MSW-LASSO tends to select fewer stocks in each iteration and stabilizes the stock selection after 5 iterations. In contrast, both ALASSO and the cardinality constraint approach select the required number of stocks within one step. However, the bias in the estimated portfolio weights based on MSW-LASSO is smaller than those based on ALASSO and the cardinality constraint approach.

3. Experimental setup

3.1. Data sets and performance metrics

To evaluate the out-of-sample performance of the proposed method, some public data sets that widely used in the current literature are used. In particular, we consider daily returns of four stock market indices and their constituents, including FTSE 100 (UK), S&P 100 (USA), Nikkei 225 (Japan), and S&P 500 (USA). The daily closing prices of these indices and their constituents during the period from 01/01/2016 to 31/12/2022 are downloaded from Yahoo Finance. The stocks that have more than 5 consecutively missing prices are removed from the data sets. Other missing prices are imputed by the linear interpolation approach. We compute the daily log-return of asset j in day t by

$$x_{t,j} = \log\left(\frac{P_{t,j}}{P_{t-1,j}}\right), \quad t = 1, \ldots, T,$$

where T is the total number of periods in a data set, and $P_{t,j}$ is the daily price of asset j in day t. Table 1 shows the descriptive statistics of the financial index returns. Clearly, the index returns exhibit typical negative skewness and fat tails.

The out-of-sample performance of a tracking portfolio is often assessed based on the following metrics: (i) tracking error (*TE*) and (ii) tracking portfolio turnover (*TO*). The tracking error measures how closely the tracking portfolio replicates the index. The turnover measures the stability of the tracking portfolio. Lower turnover means lower transaction costs.

Similar to the method of Fastrich *et al.* (2014), a moving time window approach was employed to compare index

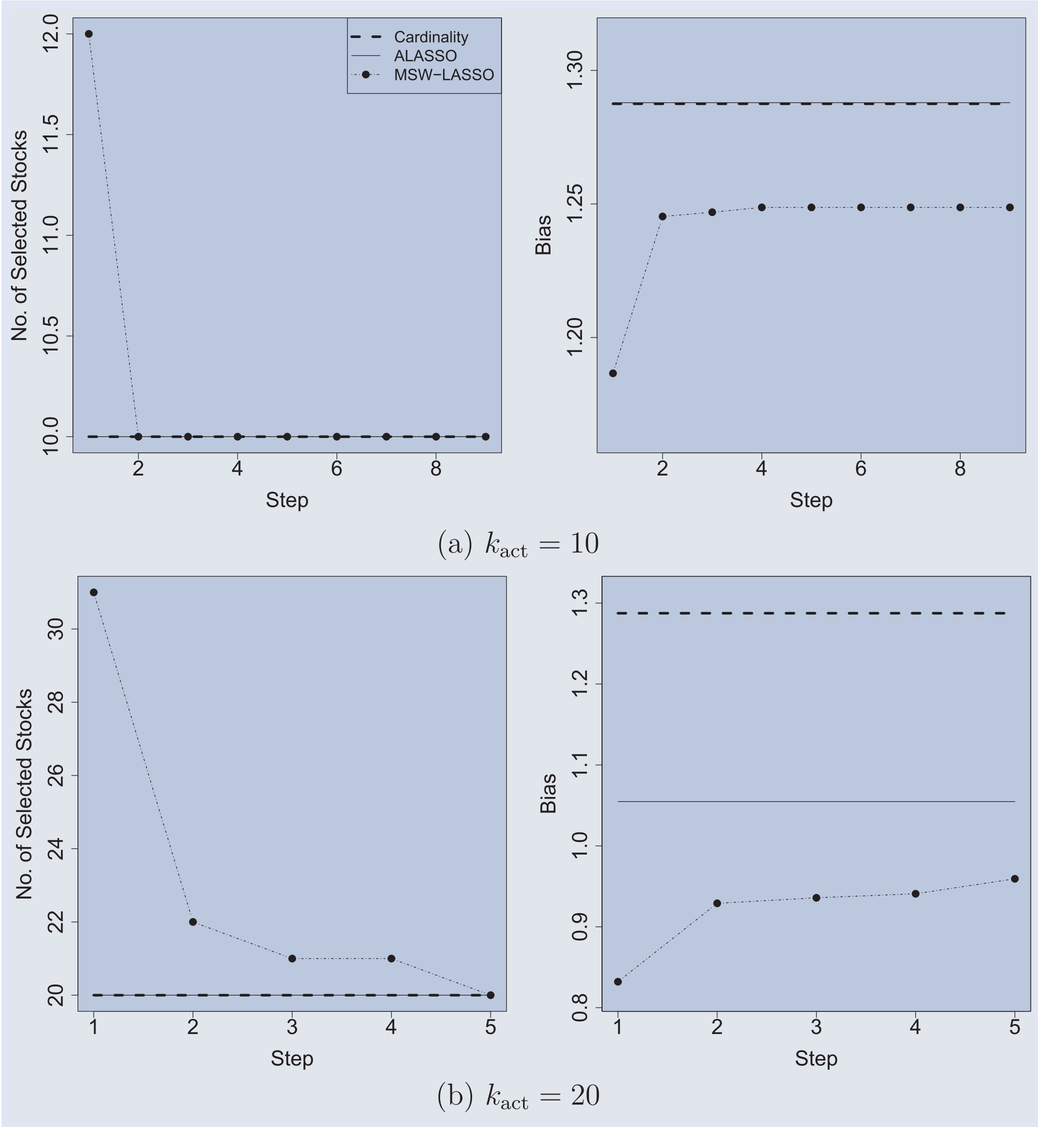

Figure 2. The number of stocks selected and biases in the estimated tracking portfolio weights based on the cardinality constraint approach, ALASSO, and MSW-LASSO with weights derived from the MCP penalty: (a) $k_{\text{act}} = 10$; and (b) $k_{\text{act}} = 20$.

Table 1. Descriptive statistics of the indices' daily log-returns (%).

	T	No. of Stocks	Mean	Standard deviation	Skewness	Kurtosis	Min	Max
FTSE 100	1766	93	0.011	1.053	– 0.965	17.252	– 11.512	8.666
S&P 100	1761	99	0.037	1.230	– 0.724	17.308	– 12.295	9.646
Nikkei 225	1709	220	0.020	1.275	– 0.166	7.838	– 8.253	7.731
S&P 500	1761	483	0.037	1.220	– 0.861	18.953	– 12.765	8.968

tracking investment strategies in all the experiments. In particular, a training window of size T_{train} ($T_{\text{train}} < T$) is first selected to determine the optimal tracking portfolio. Then the tracking portfolio weights were held unchanged and applied to the subsequent T_{test} out-of-sample trading days to compute the out-of-sample performance. At the end of this testing period, we need to redesign the new tracking portfolio. For this, we move the training window forward by T_{test} days, and use the last T_{train} days to design and the subsequent T_{test} days to evaluate the new portfolio. Following the setting of Fastrich *et al.* (2014), we choose $T_{\text{train}} = 250$ days and $T_{\text{test}} = 21$ days.

Define $N = (T - T_{\text{train}})/T_{\text{test}}$ as the total number of rolling windows. Based on the sequence of β_i for $i = 1, \ldots, N$, the turnover is computed by

$$TO = \frac{1}{N-1} \sum_{i=1}^{N-1} \sum_{j=1}^{p} (|\beta_{i+1,j} - \beta_{i+,j}|),$$

where $\beta_{i+1,j}$ is the desired weight of asset j at the $(i+1)$th window (after rebalancing), and $\beta_{i+,j}$ denotes the weight of asset j at the $(i+1)$th window before rebalancing,

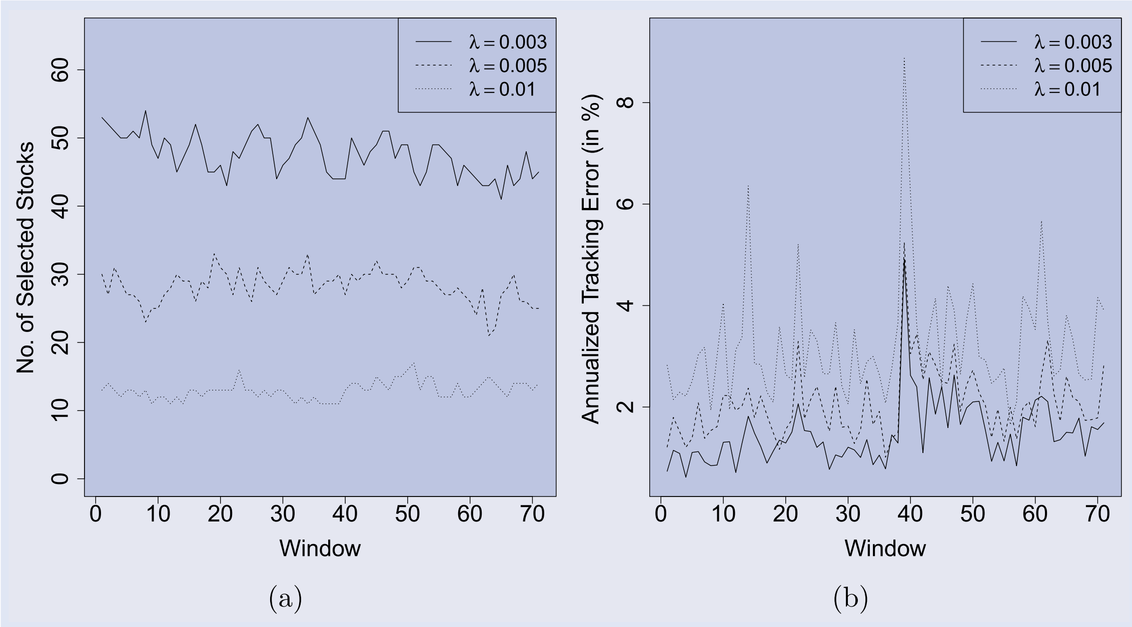

Figure 3. (a) The number of selected stocks and (b) the corresponding out-of-sample tracking error of MSW-LASSO over each window under different values of λ based on the S&P 100 data set.

given by

$$\beta_{i+,j} = \frac{b_{i,j}}{\sum_{j=1}^{p} b_{i,j}} \text{ and } b_{i,j} = \beta_{i,j}\left(\prod_{m=T_{\text{train}}+i\times T_{\text{test}}+1}^{m=T_{\text{train}}+i\times T_{\text{test}}+T_{\text{test}}} (1+x_{m,j})\right).$$

The turnover measure can be interpreted as the average percentage of wealth traded across the p available assets over the $N-1$ trading periods.

Based on the portfolio weights determined in the first training window, β_1, the out-of-sample return r_t^{os} at time t ($t = T_{\text{train}}+1,\ldots,T_{\text{train}}+T_{\text{test}}$) is calculated as

$$r_t^{\text{os}} = \frac{W_t}{W_{t-1}} - 1,$$

where $W_t = \sum_{j=1}^{p} \beta_{1,j} \prod_{i=T_{\text{train}}+1}^{t} (1+x_{i,j})$ is the relative wealth at time t. The out-of-sample tracking error for the first testing window is given by

$$TE_1^{\text{os}} = \sqrt{\frac{1}{T_{\text{test}}} \sum_{t=T_{\text{train}}+1}^{T_{\text{train}}+T_{\text{test}}} (r_t^{\text{os}} - y_t)^2},$$

where y_t is the index return at time t. The tracking error for the subsequent training and test windows can be computed in the same way.

3.2. Descriptives for the regularization parameter

The regularization parameter λ of MSW-LASSO controls the sparsity level of the tracking portfolio. The larger the value of λ is, the sparser the resulting tracking portfolio. Given the moving-window approach for evaluating the out-of-sample performance, one may consider using cross-validation to select λ, i.e. selecting λ to minimize the cross-validated tracking error over each window. However, this leads to a relatively small value of λ selected, which in turn leads to a non-sparse tracking portfolio. This is because a tracking portfolio with a larger number of selected stocks tends to have smaller out-of-sample tracking errors. This phenomenon has been observed in the finance literature (Fastrich *et al.* 2014, 2015).

For illustration, figure 3 plots the number of stocks selected and the corresponding tracking error over each window for MSW-LASSO with weights derived from the MCP penalty under different values of λ based on the S&P 100 data set. As can be seen from figure 3(a), when λ increases, the number of selected stocks tends to decrease over each window as expected. Also, when λ decreases, the tracking error tends to be smaller over each window, as can be seen from figure 3(b).

To make fair comparisons among different index tracking approaches, we need to maintain the same sparsity level of tracking portfolios, i.e.selecting the same or nearly the same number of stocks over each time window. For this reason, we select the regularization parameter to provide a desired number of active stocks, k_{act}.

4. Empirical results

In this section, we compare the out-of-sample performance between the MSW-LASSO and some competitive approaches, including the ALASSO, cardinality and SLOPE approaches. To utilize the grouping ability of SLOPE, Kremer *et al.* (2022) further suggested a variation called SLOPE-SLC. To simplify the discussion, we focus on SLOPE-SLC only but not SLOPE. The ALASSO penalty is convex while the cardinality penalty is non-convex. They are typical representatives of regularization techniques based on convex and non-convex penalties, respectively. The LASSO approach is not considered here as it is inefficient in promoting sparsity under both no-short selling and full investment conditions as mentioned above.

Table 2. Annualized out-of-sample tracking errors (in %) for different approaches based on various data sets.

				MSW-LASSO			
k_{act}	CARD	ALASSO	SLOPE-SLC	SCAD	MCP	Log-M	ℓ_q
			FTSE 100				
10	**3.96**	5.81***	5.79***	4.44	4.31	4.22	4.18
20	**2.63**	3.31***	3.50***	2.79	2.73	2.65	2.66
30	**2.06**	2.44**	2.52***	2.15	2.11	2.12	2.12
40	**1.80**	2.03	1.95	1.82	1.81	1.83	1.84
50	1.67	1.78	1.68	1.66	**1.65**	1.67	1.68
			S&P 100				
10	3.93	4.86***	7.20***	3.99	3.86	**3.78**	3.83
20	2.58	2.99***	4.17***	2.53	**2.54**	2.59	2.58
30	**1.94**	2.29**	2.76***	2.00	2.00	1.98	1.99
40	**1.59**	1.89**	1.93**	1.65	1.66	1.61	1.62
50	**1.40**	1.59**	1.49	1.42	1.41	1.41	1.42
			Nikkei 225				
20	**3.23**	3.95***	4.59***	3.32	3.27	3.34	3.36
40	**2.35**	2.65**	2.71**	2.40	2.40	2.42	2.42
60	2.03	2.16	2.05	**2.02**	2.03	2.04	2.04
80	1.90	1.95	**1.85**	1.90	1.90	1.91	1.91
100	1.85	1.86	**1.81**	1.84	1.84	1.84	1.84
			S&P 500				
40	2.33	2.68**	7.20***	2.34	**2.31**	2.33	2.36
80	1.71	1.85	4.17*	**1.69**	**1.69**	1.72	1.74
120	1.46	1.54	2.76*	1.45	1.45	**1.43**	**1.43**
160	1.36	1.37	1.93*	**1.32**	**1.32**	**1.32**	**1.32**
200	1.31	1.29	1.49	**1.28**	**1.28**	**1.28**	**1.28**

Note: The t-test is performed to test the differences in tracking errors between MSW-LASSO with MCP penalty and other approaches. The statistically significant differences with confidence levels 10%, 5%, and 1% are indicated by *, **, and ***, respectively.

Table 2 compares the annualized out-of-sample tracking errors among the cardinality, ALASSO, SLOPE-SLC, and MSW-LASSO approaches. Four different weight functions based on the derivative of SCAD, MCP, Log-M and ℓ_q penalties are considered. The shapes of these penalty functions depend on the parameter. For the sake of simplicity, we set the shape parameter of these penalty functions to be the popular value widely suggested in the literature. In particular, we set $a = 3.7$ for SCAD, $b = 2.5$ for MCP, $\varepsilon = 1.67 \times 10^{-5}$ for Log-M, and $q = 0.1$ for ℓ_q penalty.

Compared to ALASSO, MSW-LASSO employs a multi-step iteration scheme to shrink near-to-zero weights towards zero. Therefore, the latter is expected to improve bias of ALASSO in portfolio weight estimation. It is pronounced to observe from table 2 that the out-of-sample tracking error based on MSW-LASSO is uniformly smaller than that based on ALASSO for all the data sets considered here. Moreover, the differences in the tracking errors based on ALASSO and MSW-LASSO with MCP penalty are statistically significant when k_{act} is small. Clearly, MSW-LASSO outperforms ALASSO. This illustrates the benefit of the MSW-LASSO approach in increasing the tracking accuracy by curtailing biases in parameter estimation of ALASSO.

Compared to SLOPE-SLC, MSW-LASSO can usually achieve reduction of tracking error, especially when k_{act} is small and the number of assets is large. For example, under the data set S&P 500, MSW-LASSO with MCP weights produces uniformly smaller tracking error than SLOPE-SLC across a wide range values of k_{act}. The difference is substantial for $k_{act} \leq 160$.

The cardinality penalty is non-convex, compared to ALASSO. As shown in table 2 the cardinality approach usually has smaller out-of-sample tracking errors than ALASSO except for the S&P 500 data set when $k_{\text{act}} = 200$. Its performance is also very competitive to MSW-LASSO. However, compared to the cardinality approach, MSW-LASSO dramatically reduces the computational cost as it essentially solves a convex optimization problem in each iteration. To illustrates this, figure 4 compares the running time among the cardinality constraint approach, ALASSO, SLOPE-SLC, MSW-LASSO with weights derived from the MCP, SCAD, Log-M, and ℓ_q penalties against the number of stocks selected for the first rolling window. To speed up the solution procedure of the cardinality constraint approach, we set the maximum running time as 1200 seconds. Clearly, the cardinality constraint approach usually reaches the time capping of 1200 seconds. The computation time for ALASSO and SLOPE-SLC is less than one second. Compared to ALASSO, the running time of MSW-LASSO approaches usually only increase marginally. The computation time of ALASSO, SLOPE-SLC, and MSW-LASSO is substantially less than that of the cardinality approach.

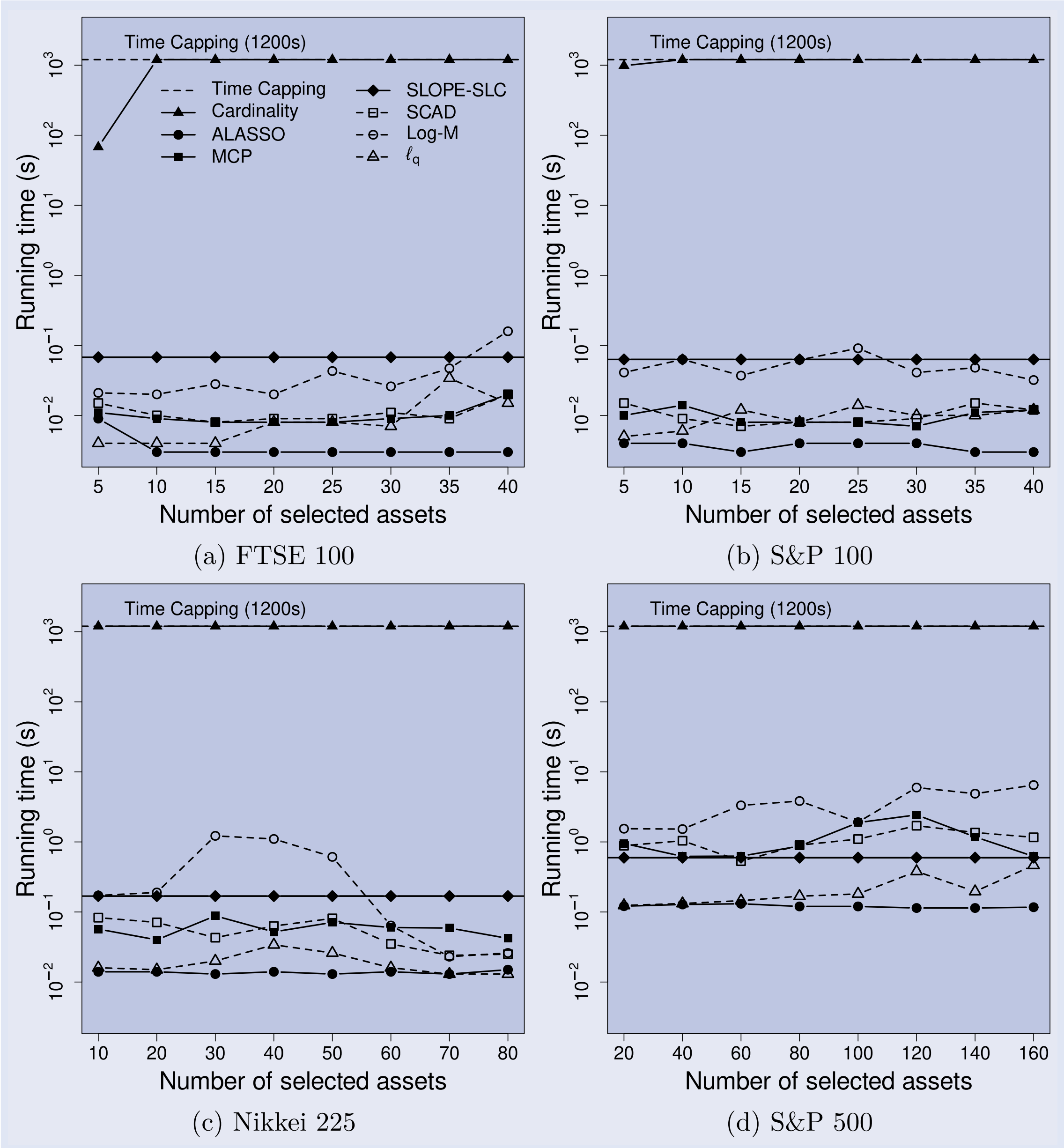

Figure 4. Comparison of running time among the cardinality constraint approach, ALASSO, SLOPE-SLC, and MSW-LASSO with weights derived from the MCP, SCAD, Log-M, and ℓ_q penalties over the first training window in each data set.

Table 3 further compares the turnover of tracking portfolios based on different approaches. The turnover tends to decrease as k_{act} increases. For a fixed k_{act}, the cardinality approach often yields the highest turnover, while SLOPE-SLC yields the lowest turnover. The turnover of MSW-LASSO is in-between the cardinality and SLOPE-SLC approaches. Moreover, as can be seen from table 3, the weight functions based on the derivatives of SCAD and MCP penalty functions can generate smaller turnover than the weight functions based on Log-M, and ℓ_q penalty functions for these four data sets.

It is also of interest to compare the diversity level of different tracking portfolios. For this purpose, table 4 compares the Herfindahl-Hirschman Index (HHI) of different tracking portfolios based on the above approaches for the FTSE 100, S&P 100, Nikkei 225, and S&P 500 data sets. The HHI is computed by $HHI = \sum_{i=1}^{p} \beta_i^2$, which measures the concentration level of the tracking portfolio. The larger the HHI, the more concentrated (or less diversified) the tracking portfolio. From table 4, we can observe that the HHI value decreases as the number of selected stocks increases. This is because the more stocks selected in the portfolio, the more diversified the portfolio is. Moreover, a general tendency can be observed. In particular, the cardinality and MSW-LASSO approaches can yield portfolios with very similar diversification, which are much more diversified than the tracking portfolio based on ALASSO. Under the S&P 500 data set, SLOPE-SLC yields a much more concentrated tracking portfolio, compared to MSW-LASSO.

5. Conclusion

The challenge of sparse index tracking is to select a subset of assets while keeping the tracking error low. Aimed

Table 3. Out-of-sample turnover for different approaches based on various data sets.

k_{act}	CARD	ALASSO	SLOPE-SLC	MSW-LASSO			
				SCAD	MCP	Log-M	ℓ_q
FTSE 100							
10	0.76	0.23	**0.15**	0.34	0.29	0.46	0.45
20	0.57	0.26	**0.16**	0.35	0.33	0.44	0.43
30	0.39	0.27	**0.18**	0.32	0.31	0.36	0.36
40	0.31	0.26	**0.19**	0.27	0.27	0.29	0.28
50	0.23	0.24	**0.18**	0.23	0.23	0.23	0.23
S&P 100							
10	0.88	0.29	**0.17**	0.38	0.37	0.45	0.45
20	0.80	0.32	**0.21**	0.43	0.42	0.55	0.56
30	0.67	0.34	**0.22**	0.43	0.42	0.53	0.51
40	0.52	0.35	**0.22**	0.39	0.39	0.44	0.44
50	0.39	0.32	**0.20**	0.32	0.32	0.37	0.37
Nikkei 225							
20	0.77	0.37	**0.27**	0.47	0.48	0.57	0.57
40	0.58	0.40	**0.30**	0.45	0.46	0.52	0.52
60	0.43	0.39	**0.31**	0.40	0.40	0.42	0.42
80	0.35	0.36	**0.31**	0.35	0.35	0.36	0.36
100	0.32	0.33	**0.30**	0.32	0.32	0.32	0.32
S&P 500							
40	1.17	0.70	**0.17**	0.84	0.85	0.96	0.97
80	0.96	0.74	**0.21**	0.81	0.81	0.88	0.88
120	0.75	0.71	**0.22**	0.71	0.71	0.75	0.75
160	0.66	0.65	**0.22**	0.61	0.62	0.64	0.64
200	0.62	0.57	**0.20**	0.56	0.56	0.57	0.57

Table 4. HHI for different approaches based on various data sets.

k_{act}	CARD	ALASSO	SLOPE-SLC	MSW-LASSO			
				SCAD	MCP	Log-M	ℓ_q
FTSE 100							
10	**0.1101**	0.1773	0.1150	0.1208	0.1116	0.1151	0.1154
20	0.0597	0.0796	0.0667	0.0591	**0.0578**	0.0621	0.0630
30	0.0451	0.0567	0.0508	0.0435	**0.0432**	0.0466	0.0470
40	0.0385	0.0460	0.0425	0.0377	**0.0376**	0.0396	0.0398
50	0.0356	0.0394	0.0377	0.0353	**0.0352**	0.0362	0.0364
S&P 100							
10	**0.1096**	0.1600	0.1249	0.1221	0.1130	0.1171	0.1175
20	**0.0585**	0.0867	0.0694	0.0594	0.0588	0.0616	0.0619
30	0.0427	0.0611	0.0494	0.0418	**0.0417**	0.0446	0.0448
40	0.0352	0.0470	0.0391	**0.0343**	**0.0343**	0.0366	0.0368
50	0.0310	0.0383	0.0329	**0.0302**	**0.0302**	0.0318	0.0320
Nikkei 225							
20	**0.0579**	0.0842	0.0753	0.0601	0.0588	0.0607	0.0610
40	0.0360	0.0459	0.0436	**0.0357**	**0.0357**	0.0373	0.0376
60	**0.0299**	0.0338	0.0333	**0.0299**	**0.0299**	0.0306	0.0307
80	**0.0279**	0.0294	0.0290	**0.0279**	**0.0279**	0.0282	0.0282
100	0.0272	0.0275	**0.0270**	**0.0270**	0.0271	0.0272	0.0272
S&P 500							
40	0.0311	0.0460	0.1249	0.0315	**0.0305**	0.0325	0.0330
80	0.0189	0.0243	0.0694	0.0184	**0.0183**	0.0198	0.0199
120	0.0153	0.0176	0.0494	**0.0151**	**0.0151**	0.0158	0.0158
160	0.0141	0.0147	0.0391	**0.0138**	**0.0138**	0.0141	0.0141
200	0.0137	0.0135	0.0329	**0.0134**	**0.0134**	**0.0134**	**0.0134**

at reducing bias while preserving convexity of the optimization problem, this paper suggested a new approach based on MSW-LASSO for sparse index tracking in high-dimensional settings. The proposed method is a generalization of the traditional LASSO and ALASSO approach. By using multi-step iterations, the sparsity of the solution tends to increase, and the bias in the parameter estimation tends to improve as compared to the traditional LASSO-based approaches. In each iteration, the solution can be obtained by solving a convex optimization problem. Therefore, the proposed approach can maintain the computational efficiency as the traditional convex regularization methods and improve the bias issue over the latter.

The empirical results show that MSW-LASSO yields smaller out-of-sample tracking errors than ALASSO and SLOPE-SLC. Compared to the cardinality approach, MSW-LASSO can produce very competitive out-of-sample performance. However, MSW-LASSO requires much less running time to solve the sparse index tracking problem than the nonconvex regularization based on the cardinality penalty. This is very attractive in practice as the computational time in high-dimensional settings is always an important concern.

Note that the same tuning parameter is used in each iteration of MSW-LASSO. It is possible to generalize MSW-LASSO with different tuning parameters for each iteration step. Some future work will be carried out in this way.

Acknowledgments

The authors thank the editor, managing editor, and three referees for their valuable comments.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work of Dr. Shi was supported by the National Natural Science Foundation of China (No. 72101119). The work of Dr. Shu was supported by the Department of Science and Technology of Guangdong Province (EF020/FBA-SLJ/2022/GDSTC), and the University of Macau Research Committee (MYRG2020-00081-FBA, MYRG2022-00017-FBA).

References

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

Benidis, K., Feng, Y. and Palomar, D.P., Sparse portfolios for high-dimensional financial index tracking. *IEEE. Trans. Signal. Process.*, 2018, **66**, 155–170.

Bühlmann, P. and Meier, L., Discussion: "One-step sparse estimates in nonconcave penalized likelihood models" (auths H. Zou and R. Li). *Ann. Stat.*, 2008, **36**(4), 1534–1541.

Bühlmann, P. and Van De Geer, S., *Statistics for High-Dimensional Data: Methods, Theory and Applications*, 2011 (Springer Science & Business Media: New York).

Candes, E.J., Wakin, M.B. and Boyd, S.P., Enhancing sparsity by reweighted ℓ_1 minimization. *J. Fourier Anal. Appl.*, 2008, **14**, 877–905.

Cremers, M., Ferreira, M.A., Matos, P. and Starks, L., Indexing and active fund management: International evidence. *J. Financ. Econ.*, 2016, **120**(3), 539–560.

Fan, J. and Li, R., Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.*, 2001, **96**(456), 1348–1360.

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

Fastrich, B., Paterlini, S. and Winker, P., Constructing optimal sparse portfolios using regularization methods. *Comput. Manage. Sci.*, 2015, **12**(3), 417–434.

Figueiredo, M.A., Nowak, R.D. and Wright, S.J., Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE. J. Sel. Top. Signal. Process.*, 2007a, **1**(4), 586–597.

Figueiredo, M.A., Bioucas-Dias, J.M. and Nowak, R.D., Majorization–minimization algorithms for wavelet-based image restoration. *IEEE. Trans. Image. Process.*, 2007b, **16**(12), 2980–2991.

Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., Pathwise coordinate optimization. *Ann. Appl. Stat.*, 2007, **1**(2), 302–332.

Giamouridis, D. and Paterlini, S., Regular (ized) hedge fund clones. *J. Financ. Res.*, 2010, **33**(3), 223–247.

Gilli, M. and Këllezi, E., The threshold accepting heuristic for index tracking. In *Financial Engineering, E-Commerce and Supply Chain*, pp. 1–18, 2002 (Springer: Neuenkirch).

Giuzio, M., Eichhorn-Schott, K., Paterlini, S. and Weber, V., Tracking hedge funds returns using sparse clones. *Ann. Oper. Res.*, 2016, **266**(1), 1–23.

Guastaroba, G. and Speranza, M.G., Kernel search: An application to the index tracking problem. *Eur. J. Oper. Res.*, 2012, **217**(1), 54–68.

Huo, X. and Chen, J., Complexity of penalized likelihood estimation. *J. Stat. Comput. Simul.*, 2010, **80**(7), 747–759.

Huo, X. and Ni, X., When do stepwise algorithms meet subset selection criteria?. *Ann. Stat.*, 2007, **35**(2), 870–887.

Jansen, R. and Van Dijk, R., Optimal benchmark tracking with small portfolios. *J. Portf. Manage.*, 2002, **28**(2), 33–39.

Jiang, W., Bogdan, M., Josse, J., Majewski, S., Miasojedow, B., Ročková, V. and Group, T., Adaptive Bayesian SLOPE: Model selection with incomplete data. *J. Comput. Graph. Stat.*, 2022, **31**(1), 113–137.

Kremer, P.J., Brzyski, D., Bogdan, M. and Paterlini, S., Sparse index clones via the sorted ℓ_1-Norm Sparse index clones via the sorted $\ell_1$1-norm. *Quant. Finance*, 2022, **22**(2), 349–366.

Krink, T., Mittnik, S. and Paterlini, S., Differential evolution and combinatorial search for constrained index-tracking. *Ann. Oper. Res.*, 2009, **172**(1), 153.

Li, J., Sparse and stable portfolio selection with parameter uncertainty. *J. Bus. Econ. Stat.*, 2015, **33**(3), 381–392.

Lobo, M.S., Fazel, M. and Boyd, S., Portfolio optimization with linear and fixed transaction costs. *Ann. Oper. Res.*, 2007, **152**(1), 341–365.

Malkiel, B.G. and Xu, Y., Risk and return revisited. *J. Portf. Manage.*, 1997, **23**(3), 9.

Maringer, D. and Oyewumi, O., Index tracking with constrained portfolios. *Intel. Syst. Account. Finance Manage. Int. J.*, 2007, **15**(1-2), 57–71.

Montfort, K. v., Visser, E. and van Draat, L.F., Index tracking by means of optimized sampling. *J. Portf. Manage.*, 2008, **34**(2), 143–152.

Osborne, M.R., Presnell, B. and Turlach, B.A., A new approach to variable selection in least squares problems. *IMA J. Numer. Anal.*, 2000, **20**(3), 389–403.

Ruiz-Torrubiano, R. and Suárez, A., A hybrid optimization approach to index tracking. *Ann. Oper. Res.*, 2009, **166**(1), 57–71.

Scozzari, A., Tardella, F., Paterlini, S. and Krink, T., Exact and heuristic approaches for the index tracking problem with UCITS constraints. *Ann. Oper. Res.*, 2013, **205**(1), 235–250.

Sharpe, W.F., The arithmetic of active management. *Financ. Anal. J.*, 1991, **47**(1), 7–9.

Strub, O. and Trautmann, N., A two-stage approach to the UCITS-constrained index-tracking problem. *Comput. Oper. Res.*, 2019, **103**, 167–183.

Takeda, A., Niranjan, M., Gotoh, J y. and Kawahara, Y., Simultaneous pursuit of out-of-sample performance and sparsity in index tracking portfolios. *Comput. Manage. Sci.*, 2013, **10**(1), 21–49.

Tibshirani, R., Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodol.)*, 1996, **58**(1), 267–288.

Wu, L. and Yang, Y., Nonnegative elastic net and application in index tracking. *Appl. Math. Comput.*, 2014, **227**, 541–552.

Wu, L., Yang, Y. and Liu, H., Nonnegative-lasso and application in index tracking. *Comput. Stat. Data. Anal.*, 2014, **70**, 116–126.

Xu, F., Xu, Z. and Xue, H., Sparse index tracking based on $L\_\{1/2\}$ model and algorithm. arXiv preprint arXiv:1506.05867, 2015.

Yang, Y. and Wu, L., Nonnegative adaptive lasso for ultra-high dimensional regression models and a two-stage method applied in financial modeling. *J. Stat. Plan. Inference.*, 2016, **174**, 52–67.

Zhang, C.H.., Nearly unbiased variable selection under minimax concave penalty. *Ann. Stat.*, 2010, **38**(2), 894–942.

Zou, H., The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.*, 2006, **101**(476), 1418–1429.

Zou, H. and Li, R., One-step sparse estimates in nonconcave penalized likelihood models. *Ann. Stat.*, 2008, **36**(4), 1509.

Appendix. Proof of Proposition thm1

Consider a general optimization problem

$$\begin{aligned} \min_x \quad & g(x), \\ \text{s.t.} \quad & x \in \mathcal{X}, \end{aligned}$$

where set $\mathcal{X}$ is a convex set, and $g(x)$ is concave on $\mathcal{X}$. One way to optimize such an objective is by the Majorization-Minimization (MM) algorithm (Figueiredo *et al.* 2007b). Given a feasible solution $x^{(k-1)} \in \mathcal{X}$, Candes *et al.* (2008) further improved the algorithm by minimizing a linearization of g around $x^{(k-1)}$. This leads to the following iteration

$$x^{(k)} = \arg\min_x g(x^{(k-1)}) + \langle \nabla g(x^{(k-1)}), x - x^{(k-1)} \rangle, \quad \text{s.t. } x \in \mathcal{X}. \tag{A1}$$

Since the updating scheme (A1) iteratively minimizes a local linear approximation of the objective function $g(\cdot)$, this algorithm is often known as the iterative local linear approximation algorithm.

Note that for $\beta \geq 0$, the penalty in (8), $\sum_{j=1}^{p} p_\lambda(\beta_j)$, is concave. Given the current estimator $\hat{\beta}^{(k-1)}$, a local linear approximation to the penalty function leads to

$$\begin{aligned} \hat{\beta}^{(k)} = \arg\min_{\beta \in \mathcal{B}} \quad & \|Y - X\beta\|_2^2 + \sum_{j=1}^{p} p_\lambda(\hat{\beta}_j^{(k-1)}) \\ & + \sum_{j=1}^{p} p'_\lambda(\hat{\beta}_j^{(k-1)}) * (\beta_j - \hat{\beta}_j^{(k-1)}), \end{aligned}$$

which is equivalent to

$$\hat{\beta}^{(k)} = \arg\min_{\beta \in \mathcal{B}} \quad \|Y - X\beta\|_2^2 + \sum_{j=1}^{p} p'_\lambda(\hat{\beta}_j^{(k-1)}) * \beta_j. \tag{A2}$$

Since $\beta \in \mathcal{B}$ requires that $\beta \geq 0$, we have $\beta_j = |\beta_j|$. Then problem (A2) is the same as

$$\hat{\beta}^{(k)} = \arg\min_{\beta \in \mathcal{B}} \quad \|Y - X\beta\|_2^2 + \sum_{j=1}^{p} p'_\lambda(\hat{\beta}_j^{(k-1)}) * |\beta_j|, \tag{A3}$$

which is exactly the MSW-LASSO with weight $w_j^{(k-1)} = p'_\lambda(\hat{\beta}_j^{(k-1)})$. Therefore, the MSW-LASSO can be viewed as an iterative local linear approximation to the problem (8) with penalty $\sum_{j=1}^{p} p_\lambda(\beta_j)$.