Sparse Index Tracking With K-Sparsity or ϵ-Deviation Constraint via ℓ_0-Norm Minimization

Xiao Peng Li, Zhang-Lei Shi, Chi-Sing Leung, *Senior Member, IEEE*, and Hing Cheung So, *Fellow, IEEE*

***Abstract*— Sparse index tracking, as one of the passive investment strategies, is to track a benchmark financial index via constructing a portfolio with a few assets in a market index. It can be considered as parameter learning in an adaptive system, in which we periodically update the selected assets and their investment percentages based on the sliding window approach. However, many existing algorithms for sparse index tracking cannot explicitly and directly control the number of assets or the tracking error. This article formulates sparse index tracking as two constrained optimization problems and then proposes two algorithms, namely, nonnegative orthogonal matching pursuit with projected gradient descent (NNOMP-PGD) and alternating direction method of multipliers for ℓ_0-norm (ADMM-ℓ_0). The NNOMP-PGD aims at minimizing the tracking error subject to the number of selected assets less than or equal to a predefined number. With the NNOMP-PGD, investors can directly and explicitly control the number of selected assets. The ADMM-ℓ_0 aims at minimizing the number of selected assets subject to the tracking error that is upper bounded by a preset threshold. It can directly and explicitly control the tracking error. The convergence of the two proposed algorithms is also presented. With our algorithms, investors can explicitly and directly control the number of selected assets or the tracking error of the resultant portfolio. In addition, numerical experiments demonstrate that the proposed algorithms outperform the existing approaches.**

***Index Terms*— Alternating direction method of multipliers (ADMM), index tracking, nonnegative orthogonal matching pursuit (NNOMP), projected gradient descent (PGD), sparse recovery.**

I. Introduction

INVESTMENT refers to selecting assets and distributing money over the selected assets. In general, there are two investment strategies, namely, active and passive [1], [2]. An active strategy seeks to beat the market performance, such as Russell 2000 and S&P 500 indexes, via actively buying and selling assets. In the last decade, some learning system-based active strategies [3], [4] were developed. For example, we can use time series forecasting methods to perform prediction [3], [4]. Active strategies might help investors to earn a higher profit than the market return. Concomitantly, the risk of active strategies is high. In addition, the management and administration fees of an active portfolio are high since fund managers and/or experienced investors are involved. In contrast, a passive strategy tends to replicate a benchmark market index. Hence, it is able to obtain market returns at low risk. The management fee or overhead of a passive fund is generally low.

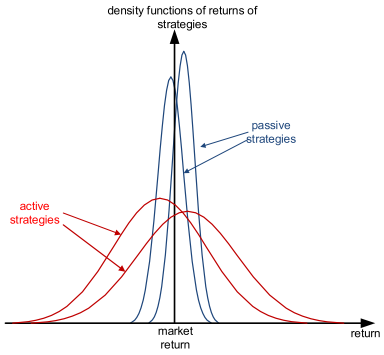

Fig. 1. Density functions of returns in active and passive investment strategies. The horizontal axis refers to the return, while the vertical axis refers to density function values. Usually, the variances (risk) of active strategies are higher than those of passive strategies.

It was reported [5] that most of the active funds are inferior to the market in the long term. Fig. 1 shows the return distributions of two kinds of strategies. The horizontal axis refers to the return, while the vertical axis refers to the density function value of the return. The center point in the horizontal axis is the market return. Usually, the returns of active strategies are with higher variance. Therefore, more and more investors prefer the passive strategy, including those holding stable funds and conservative funds.

Recently, a number of neural network-based works [6]–[8] were proposed for finance or asset management. For example, in [6], a radial basis function (RBF) approach was suggested for market trend representations. This approach improves the performance in price prediction for portfolio selection. In [7], several analog neural network methods were proposed for designing an investment portfolio. In addition, a deep neural network approach [8] was proposed to handle trading decisions for improving rewards in an unknown market environment.

Manuscript received April 28, 2021; revised November 11, 2021 and March 7, 2022; accepted April 27, 2022. *(Corresponding authors: Zhang-Lei Shi; Chi-Sing Leung.)*

Xiao Peng Li, Chi-Sing Leung, and Hing Cheung So are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China (e-mail: x.p.li@my.cityu.edu.hk; eeleungc@cityu.edu.hk; hcso@ee.cityu.edu.hk).

Zhang-Lei Shi is with the College of Science, China University of Petroleum (East China), Qingdao 266580, China (e-mail: shizhanglei2010@163.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3171819.

Digital Object Identifier 10.1109/TNNLS.2022.3171819

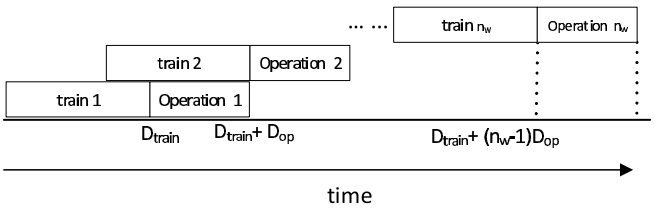

Fig. 2. Moving window concept in sparse index tracking.

Index tracking is one of the passive investment techniques for managing assets. It aims at tracking the performance of a market index by selecting some assets and determining the investment percentages of the selected assets. Of course, a perfect index tracking [9]–[11] is a full replication of a market index, which straightforwardly purchases all the assets of the target index with their true construction weights. Nevertheless, there are several drawbacks. First, high costs are caused by operating all assets when the index contains numerous assets. For instance, to replicate the Russell 2000 index, it is necessary to handle all the 2000 stocks. This implies that numerous transaction operations and fees are required, especially to track a portfolio with a high rebalancing frequency. Second, a market index might contain some illiquid stocks that will be difficult to sell.

Hence, a better approach is to select some important assets to construct a portfolio such that the performance of the portfolio closely fits that of the market index. This approach is called sparse index tracking. Formally speaking, given N assets in a market index, we would like to find a portfolio or saying a portfolio weight vector $\boldsymbol{w} \in \mathbb{R}^N$ such that the performance of the portfolio fits that of the market index, where $0 \leq w_n \leq 1$ (positive constraint) and $\sum_{n=1}^{N} w_n = 1$ (sum-to-one constraint). The nth portfolio weight w_n is the investment percentage on the nth asset. Since we would like to select dominant assets, the portfolio vector should be a sparse vector, i.e., there are a few nonzero elements in $\boldsymbol{w}$.

Regression [12]–[15] is a classical problem in neural networks. Sparse index tracking is a special form of regression problems because there are some constraints in the problem. In order to make the resultant portfolio to be adaptive to the market, index tracking should be considered as parameter learning for an adaptive system. As shown in Fig. 2, we select some important assets and estimate their investment percentages based on the data of the last training window. Afterward, we use the estimated portfolio to run the investment for an operation period. It should be noticed that there is overlapping between training windows, but there is no overlapping between operation windows.

In addition, sparse index tracking can be considered as a special form of feature selection problems [16]–[19], in which we choose a subset of relevant features for model construction. Inspired by feature selection, several sparse index tracking algorithms were proposed in [20]–[26]. In practice, many exchange-traded funds (ETFs) utilize the sparse strategy to track the market [27]–[29]. A trivial portfolio design is to keep only the large-weighted assets of the market index and exclude all remaining assets. However, its objective is not to minimize the tracking error. Alternatively, we can select assets from the whole assets based on the statistical theory. In [20] and [30], similarities between assets and the market index are first computed. Afterward, the assets with large similarities are selected. Nevertheless, the objective of this procedure is also not to minimize the tracking error.

A relatively new framework is to utilize sparse recovery theory [31], [32] to solve the sparse index tracking problem [9], [22], [33]. As shown in Fig. 2, in this idea, based on the data of a training window, we construct a sparse portfolio. Afterward, we use the constructed portfolio to run the investment for certain duration. In this framework, constructing the portfolio is formulated as an optimization problem [34], [35] that minimizes the ℓ_2-norm of tracking error and the ℓ_0-norm of the portfolio weight vector $\boldsymbol{w}$, subject to the two constraints, which are nonnegativity and sum-to-one. Here, the ℓ_0-norm is the sparsity measurement of the constructed portfolio. Since ℓ_0-norm minimization [32] is NP-hard, some works [9] proposed to approximate it by a continuous and differentiable function. They handle the resultant nonconvex formulation with the majorization–minimization (MM) method [36], [37]. However, in those works, investors cannot directly control the sparsity level or the tracking error.

The ℓ_1-norm, as a surrogate function of the ℓ_0-norm [38], [39], can be used to simplify the formulation of sparse index tracking. In [23], the least absolute shrinkage and selection operator (LASSO) [40], [41] is utilized to solve sparse index tracking. Besides, Lai *et al.* [42] leveraged the ℓ_1-norm and then converted the sum-to-one constrained problem into an unconstrained augmented Lagrangian optimization. Apart from having the same difficulty of controlling sparsity, its worst case risk cannot be controlled because negative weights are allowed. It should be noticed that investment funds are not allowed to perform the short-sell operation, i.e., negative w_n's.

Since sum-to-one and nonnegativity constraints make the ℓ_1-norm of the portfolio weight vector to be a constant, these two constraints cannot be involved in the standard LASSO for sparse index tracking. To handle this issue, some researchers suggested the reweighted ℓ_1-norm concept to make these two constraints effective in the LASSO framework [24], [25]. Nevertheless, we need to face the challenge of properly selecting the regularization parameters. In [26], a robust algorithm is proposed to handle outliers. It replaces the ℓ_2-norm by composite quantile regression [43] to minimize the tracking error.

Many existing algorithms follow the conventional regularization approach. They put the two objectives, which are tracking error and sparsity level, into a single objective function. However, this formulation may have the following issues.

1) In the conventional tasks, such as feature extraction and sparse recovery, we can focus on achieving the minimum test set error with an appropriate regularization parameter, that is, we run the algorithm a number of times with different regularization parameter values and obtain a number of models with different sparsity levels. Afterward, we select the model based on the test set.
   For sparsity index tracking, the above approach does not work. It is because the portfolio with minimum test set

error or training set error is a trivial solution, which is the full replication of the market index.

2) When we put the two objectives, tracking error and sparsity level, into a single objective function, we need to tune the regularization parameter(s) based on some trial-and-error methods such that the sparsity level or the tracking error of the resultant portfolio meets the target levels. In other words, investors cannot directly and explicitly control the sparsity or tracking error levels of the resultant portfolio. To obtain a desired sparsity level or tracking error level, for the aforementioned algorithms, we need to carefully tune the parameters. From the user's point of view, an investor may like to directly define the number of assets in the resultant portfolio or the investor may like to directly define the fitting error in the resultant portfolio.
3) Suppose that we use a modified form of the ℓ_0-norm in the regularization term. Even the resultant portfolio reaches the desired sparsity level, and the tracking error is still not optimized. It is because the objective, being optimized, is not the tracking error at the desired sparsity level. Instead, the objective, being optimized, is a combination of the tracking error and the regularization term.

The controllability on fitting error and sparsity level is usually considered in many machine learning and signal processing [44], [45] applications, especially for feature selection. To the best of our knowledge, there are a few works on sparse index tracking with the controllability on fitting error and sparsity level.

This article focuses on addressing sparse index tracking with direct and explicit control on sparsity level or tracking error level. For sparsity control, we formulate the index tracking problem as an optimization problem that minimizes the tracking error, subject to the number of selected assets less than or equal to a preset threshold. We decompose the whole problem into two consecutive subproblems, namely, asset selection and capital allocation. In the former subproblem, we use the nonnegative orthogonal matching pursuit (NNOMP) concept [46], [47] to choose a given number of assets in the index. After determining the involved assets, we allocate the weights for the selected assets based on the projected gradient descent (PGD) [48], [49]. We call the proposed algorithm as NNOMP-PGD.

For tracking error control, we formulate the index tracking problem as an optimization problem that minimizes the number of assets, subject to the tracking error less than or equal to a preset threshold. In this way, we can explicitly limit the tracking error in the resultant portfolio. Based on the alternating direction method of multipliers (ADMM) [50]–[52] concept, we derive an algorithm, called ADMM for ℓ_0-norm (ADMM-ℓ_0), to solve the problem. The ADMM-ℓ_0 algorithm is an iterative algorithm. Each iteration contains three alternating optimization steps. For the first step, even though there is an ℓ_0-norm term in the objective function of this step, we develop a closed-form solution for this step. In addition, the convergence property of the ADMM-ℓ_0 algorithm is presented. Compared with existing works, our main contributions are summarized as follows.

1) In the formulation of the NNOMP-PGD algorithm, the objective is to minimize the tracking error subject to the number of the selected assets less than or equal to a predefined value. Hence, investors can directly and explicitly control the sparsity level of the resultant portfolio.
2) In the formulation of the ADMM-ℓ_0 algorithm, the objective is to minimize the number of the selected assets subject to the tracking error less than or equal to a predefined value. Hence, investors can directly and explicitly control the tracking error of the resultant portfolio.
3) We provide the convergence analysis of the ADMM-ℓ_0. Each iteration of ADMM-ℓ_0 contains three alternating updates. One of them involves the ℓ_0-norm minimization update. We derive a global closed-form solution for this update.
4) The proposed algorithms exhibit lower tracking error than the existing algorithms based on three real-world datasets, namely, Russell 2000, S&P 500, and NASDAQ 100. Compared with ADMM-ℓ_0, the NNOMP-PGD is suitable for investors who intend to purchase a fixed number of assets. On the other hand, for investors who are more concerned about tracking error, the ADMM-ℓ_0 is more preferable.

This article is organized as follows. Related works and various formulations of sparse index tracking are reviewed in Section II. In Section III, two algorithms are derived. Convergence and computational complexity are also presented. Numerical results are then presented in Section IV. Finally, conclusions are drawn in Section V.

In this article, scalars, vectors, and matrices are represented by italic, bold lower case, and bold upper case letters, respectively. The notation $\boldsymbol{w} \succeq 0$ means that all entries of $\boldsymbol{w}$ are nonnegative. Transpose operator is denoted by $(\cdot)^T$.

II. Overview and Related Works

Consider a market index with N assets. Let $\boldsymbol{r} \in \mathbb{R}^{D_t}$ be the daily returns of the market index over the last D_t trading days. Let $\boldsymbol{A} = [\boldsymbol{a}_1 \cdots \boldsymbol{a}_N] \in \mathbb{R}^{D_t \times N}$ be the daily returns of N assets in these trading days, where $\boldsymbol{a}_n$ refers to daily returns of the nth asset. In addition, $\boldsymbol{w} = [w_1 \cdots w_N]^T \in \mathbb{R}^N$ is the portfolio weight vector, where w_n is the investment percentage assigned to the nth asset.

The replicating performance $J(\boldsymbol{w})$ of a portfolio is

$$J(\boldsymbol{w}) = \frac{1}{D}\|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{r}\|_2^2. \tag{1}$$

The sparse index tracking is a constrained regression problem, which is given by

$$\begin{aligned} &\min_{\boldsymbol{w}} J(\boldsymbol{w}) = \frac{1}{D}\|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{r}\|_2^2 \\ &\text{s.t. } \boldsymbol{w} \succeq 0, \quad \boldsymbol{w}^T\mathbf{1} = 1, \quad \text{and} \quad \boldsymbol{w} \text{ is sparse.} \end{aligned} \tag{2}$$

Its task is to search for a $\boldsymbol{w}$ that fits $\boldsymbol{r}$ under three constraints.

First, the weights must be nonnegative. This means that short selling is not allowed because it has a very high potential risk. Second, the sum of investment percentages must be equal to one. Finally, most of the weights should be zero.

The sparse index tracking problem is different from conventional sparse recovery problems or feature extraction problems. First, there are two constraints, sum-to-one constraint and nonnegative constraint, in sparse index tracking. They usually do not appear simultaneously in conventional sparse recovery problems and feature extraction problems.

Second, in sparse index tracking, there are two objectives: tracking error and sparsity level. We cannot focus on the tracking error only. It is because the solution for minimizing the training set tracking error or test set tracking error is trivial. It is because we can fully replicate the market index. This property does not appear in other sparse recovery problems.

One approach for solving (2) is to decompose the whole problem into two consecutive subproblems, namely, asset selection and capital allocation. In [20], the asset selection step computes the correlation factors [53] β_n's, which is given by

$$\beta_n = \frac{\text{Cov}(\boldsymbol{a}_n, \boldsymbol{r})}{\text{Var}(\boldsymbol{r})} \tag{3}$$

where $\text{Cov}(\boldsymbol{a}_n, \boldsymbol{r})$ denotes the covariance of $\boldsymbol{a}_n$ and $\boldsymbol{r}$ and $\text{Var}(\boldsymbol{r})$ denotes the variance of $\boldsymbol{r}$. Correlation factor β_n measures the similarity between the nth asset and the market index. When β_n is large, the trend of the nth asset is similar to the market index. After computing all β_n's, a set of assets with large values of β_n is selected. In capital allocation, the investment percentages of the selected assets are optimized by the genetic algorithm (GA). Nevertheless, in practice, the GA is not able to guarantee the global minimization of fitting error. Moreover, the mentioned stock selection process may not attain the most suitable assets to represent the target index because it considers the similarity of trend rather than the minimum of fitting error.

When the tracking error and sparsity are combined together, the sparse index tracking problem [22] can be formulated as

$$\begin{aligned} &\min_{\boldsymbol{w}} \ \frac{1}{D}\|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{r}\|_2^2 + \lambda\|\boldsymbol{w}\|_0 \\ &\text{s.t. } \boldsymbol{w} \succeq 0 \quad \text{and} \quad \boldsymbol{w}^T\boldsymbol{1} = 1 \end{aligned} \tag{4}$$

where $\|\boldsymbol{w}\|_0$ is the number of nonzero elements in $\boldsymbol{w}$. It is used to control the sparsity constraint. Here, $\lambda > 0$ is a regularization parameter to trade off fitting error and the number of assets. Since minimizing $\|\boldsymbol{w}\|_0$ is NP-hard, Benidis *et al.* [22] employed a continuous and differentiable function to approximate $\|\boldsymbol{w}\|_0$ and then used MM to deal with the resultant formulation. However, in (4), the sparsity of the portfolio is very sensitive to λ. In practice, tuning λ to achieve the desired number of assets is time-consuming.

In conventional sparse recovery algorithms, the ℓ_1-norm is considered as an effective approximation for the ℓ_0-norm. Hence, in [23], the sparse index tracking problem was formulated as

$$\begin{aligned} &\min_{\boldsymbol{w}} \ \frac{1}{D}\|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{r}\|_2^2 + \lambda\|\boldsymbol{w}\|_1 \\ &\text{s.t. } \boldsymbol{w}^T\boldsymbol{1} = 1 \end{aligned} \tag{5}$$

where (5) is the well-known LASSO formulation. In (5), there is no nonnegativity constraint. Therefore, negative weights may appear, which will lead to the possibility of short selling.

It is worth pointing out that if the sum-to-one and nonnegativity constraints of $\boldsymbol{w}$ are involved in (5), then $\|\boldsymbol{w}\|_1$ in (5) is a constant and hence becomes irrelevant. To deal with this issue, Shu *et al.* [24] utilized the reweighted ℓ_1-norm [54] to modify (6) as

$$\begin{aligned} &\min_{\boldsymbol{w}} \frac{1}{D}\|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{r}\|_2^2 + \lambda_1\left\|\boldsymbol{v}^T\boldsymbol{w}\right\|_1 + \lambda_2\|\boldsymbol{w}\|_2 + \lambda_3\|\boldsymbol{w} - \hat{\boldsymbol{w}}\|_1 \\ &\text{s.t. } \boldsymbol{w} \succeq 0 \quad \text{and} \quad \boldsymbol{w}^T\boldsymbol{1} = 1 \end{aligned} \tag{6}$$

where $\boldsymbol{v}$ and $\hat{\boldsymbol{w}} \in \mathbb{R}^N$ are reweighted and the portfolio weight vectors in previous time period vectors, respectively. Here, $\lambda_2\|\boldsymbol{w}\|_2$ is able to solve the collinear problem in LASSO [55], and $\lambda_3\|\boldsymbol{w} - \hat{\boldsymbol{w}}\|_1$ is to minimize the turnover fee, which is able to limit the number of transactions. However, the work in [24] does not provide the scheme to tune the parameters, namely, λ_1, λ_2, and λ_3. Hence, we face the challenge of properly selecting them.

It is well known that the ℓ_2-norm is highly sensitive to outliers, which may correspond to extreme returns in volatile periods. To achieve robustness and improve performance in such conditions, in [26], the composite quantile regression concept is adopted, in which the optimization problem becomes

$$\begin{aligned} &\min_{\boldsymbol{w},\boldsymbol{b}=[b_1\cdots b_M]} \sum_{m=1}^{M}\sum_{i=1}^{N} \rho_{\tau_m}\left(\boldsymbol{a}_i^T\boldsymbol{w} - b_m - y_i\right) + \lambda_1\left\|\boldsymbol{v}^T\boldsymbol{w}\right\|_1 \\ &\text{s.t. } \boldsymbol{w} \succeq 0 \quad \text{and} \quad \boldsymbol{w}^T\boldsymbol{1} = 1 \end{aligned} \tag{7}$$

where $\rho_{\tau_m}(x) = \tau_m \max(x, 0) - (1 - \tau_m)\max(-x, 0)$ and $\rho_{\tau_m}(x)$'s is called check loss functions. Parameter τ_m is set as $\tau_m = m/(M + 1)$ and b_m is the τ_m quantile of the random error. In this approach, we still need to carefully set the regularization parameter and the parameter M.

Before we present our algorithms, we would like to clarify several issues first. The formulations from (4) to (7) follow the regularization approach, in which the objective function is given by

$$\text{objective} = \text{tracking error} + \text{regularizer}. \tag{8}$$

1) One may think that we can tune regularization parameter(s) such that the resultant portfolio achieves the minimum test set tracking error. In fact, for index tracking, this tuning strategy is not appropriate. It is because the solution with the minimum test set tracking error is the full replication of the original market index.
2) The formulation of (8) does not allow us to directly and explicitly control the sparsity or the tracking error. However, the controllability on fitting error and sparsity level is usually considered in many machine learning and signal processing applications [44], [45], especially for feature selection.
3) Even we can afford the tuning process to tune the sparsity level to reach our desired level, the objective of the solution may not correspond to the tracking error reduction. Suppose that we consider a regularization term, which is a modified ℓ_1-norm term or ℓ_p-norm

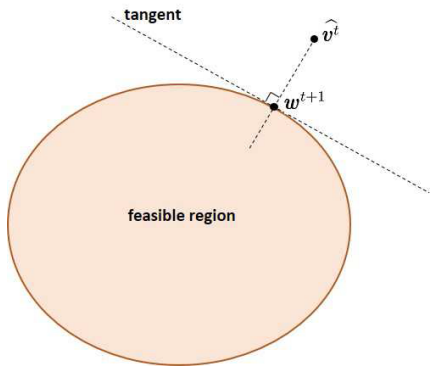

Fig. 3. Illustration of PGD.

term. After we tune the regularization parameter(s) such that the sparsity level of the solution meets the desired sparsity level, this solution is obtained by optimizing the objective in (8). The objective value at the desired sparsity level is a combination of tracking error and regularization value, rather than the tracking error only. Similarly, suppose that we tune the regularization parameter(s) such that the training tracking error of the solution meets the desired error level. This solution is obtained by optimizing the objective in (8) rather than the sparsity level.

4) As there are two objectives in the sparse index tracking, more practical requirements are to have a reasonable tracking error and a reasonable sparsity level. Instead of using the loose word "reasonable," this article can consider two cases. The first one is to minimize the tracking error subject to the number of selected assets less than or equal to a predefined value. The second case is to minimize the number of selected assets subject to the tracking error less than or equal to a predefined value.

III. Proposed Algorithms

In this section, we propose two algorithms to tackle the problem stated in (2).

A. Explicitly Control Number of Selected Assets

This section derives our NNOMP-PGD algorithm. In the existing methods, they control the number of selected assets for constructing a sparse portfolio via tuning the regularization parameter, which is difficult to attain the desired number in practice.

To explicitly control the sparsity, we add an ℓ_0-norm constraint and reformulate (2) as

$$\begin{aligned} &\min_{\boldsymbol{w}} \ \|\boldsymbol{A}\boldsymbol{w}-\boldsymbol{r}\|_2^2 \\ &\text{s.t. } \boldsymbol{w}\succeq 0, \quad \boldsymbol{w}^T\mathbf{1}=1 \quad \text{and} \quad \|\boldsymbol{w}\|_0\le K \end{aligned} \tag{9}$$

where K is the investor's preference on the number of assets in the resultant portfolio. In (9), we drop the scalar $1/D$ for presentation simplicity because it does not have the effect on the solution.

Since there are three constraints in (9) and one of them involves the ℓ_0-norm, it is challenging to solve (9). We separate (9) into two subproblems, namely, asset selection and capital allocation. In asset selection, we solve the problem, which is given by

$$\begin{aligned} &\min_{\boldsymbol{w}} \ \|\boldsymbol{A}\boldsymbol{w}-\boldsymbol{r}\|_2^2 \\ &\text{s.t. } \boldsymbol{w}\succeq 0 \quad \text{and} \quad \|\boldsymbol{w}\|_0\le K. \end{aligned} \tag{10}$$

This is a nonnegative sparse recovery problem. We suggest to use the NNOMP for handling (10).

Compared with the correlation approach in [20], our formulation selects better assets since (10) minimizes the fitting error, while the correlation approach aims at searching assets with similar trend.

After determining the K assets, one may think that we can use the solution of (10) to obtain a portfolio by normalizing the solution. However, this strategy could not ensure that the normalized portfolio is optimal for the selected assets. Given the selected assets, the best way is to optimize the investment percentages again.

After determining the K assets, we remove the $(N-K)$ columns in $\boldsymbol{A}$ corresponding to the unselected items to form $\widetilde{\boldsymbol{A}}\in\mathbb{R}^{D_t\times K}$ and define a reduced weight vector $\widetilde{\boldsymbol{w}}\in\mathbb{R}^K$ for the selected assets. Afterward, we solve the capital allocation problem, which is given by

$$\begin{aligned} &\min_{\widetilde{\boldsymbol{w}}} \ \left\|\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{w}}-\boldsymbol{r}\right\|_2^2 \\ &\text{s.t. } \widetilde{\boldsymbol{w}}\succeq 0, \quad \text{and} \quad \widetilde{\boldsymbol{w}}^T\mathbf{1}=1. \end{aligned} \tag{11}$$

In (11), the aim is to optimize the weights for the selected assets such that the tracking error is minimized.

In (11), the objective function and constraints are convex. It can be solved by various methods. Based on the penalty method in convex optimization [56], [57], this article converts (11) into a unconstrained problem, which is given by

$$\min_{\widetilde{\boldsymbol{w}}\ge 0} f\left(\widetilde{\boldsymbol{w}}^t\right) := \left\|\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{w}}^t-\boldsymbol{r}\right\|_2^2+\frac{\lambda}{2}\left(\left(\widetilde{\boldsymbol{w}}^t\right)^T\mathbf{1}-1\right)^2. \tag{12}$$

In (12), we choose a simple quadratic penalty function and parameter λ is to control the fitness of the sum-to-one constraint. In our study, "λ around 10" is good enough.

The problem stated in (12) is a convex optimization problem, and hence, it can be effectively solved by the PGD concept. Its iterative equation is given by

$$\widetilde{\boldsymbol{v}}^t = \widetilde{\boldsymbol{w}}^t-\mu\nabla f\left(\widetilde{\boldsymbol{w}}^t\right) \tag{13a}$$

$$\widetilde{\boldsymbol{w}}^{t+1} = \arg\min_{\widetilde{\boldsymbol{w}}\ge 0}\left\|\widetilde{\boldsymbol{w}}-\widetilde{\boldsymbol{v}}^t\right\|^2 \tag{13b}$$

where $\nabla f(\widetilde{\boldsymbol{w}}^t)=2\widetilde{\boldsymbol{A}}^T(\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{w}}^t-\boldsymbol{r})+\lambda((\widetilde{\boldsymbol{w}}^t)^T\mathbf{1}-1)$ and $\mu>0$ is the step size. Equation (13a) is the standard gradient descent (GD) process, and Equation (13b) can be considered as a projection operation. The projection solution is $\widetilde{\boldsymbol{w}}^{t+1}=P(\widetilde{\boldsymbol{v}}^t)$, which is given by

$$P\left(\widetilde{v}_n^t\right)=\begin{cases}0, & \text{if } \widetilde{v}_n^t<0\\ \widetilde{v}_n^t, & \text{if } \widetilde{v}_n^t\ge 0\end{cases} \tag{14}$$

where $\widetilde{v}_n^t$ is the nth entry of $\widetilde{\boldsymbol{v}}^t$. Fig. 3 shows the projection operation. After determining $\boldsymbol{v}^t$ via (13a), the projection finds a point closest to $\boldsymbol{v}^t$ in the feasible region.

Algorithm 1 summarizes the procedure of the proposed NNOMP-PGD algorithm. The first iteration loop is to handle the asset selection, while the second one is to tackle the capital allocation. In Algorithm 1, Step 6 is able to greatly reduce the computational complexity in PGD due to $K \ll N$. In our study, we stop the algorithm when the condition $||\widetilde{\boldsymbol{w}}^t - \widetilde{\boldsymbol{w}}^{t-1}|| \leq 10^{-7}$ is met.

Algorithm 1 Our NNOMP-PGD

Input: $\boldsymbol{A}$, $\boldsymbol{r}$, K and μ.
Initialize: $\boldsymbol{y}^0 = \boldsymbol{r}$, $\boldsymbol{w} = \boldsymbol{0}$, index set $\mathcal{I}^0 = \emptyset$ and $\boldsymbol{A}_{\mathcal{I}^0} = \emptyset$
for $k = 1, 2, \ldots, K$ **do**
1) $i_k = \arg\max_{n \notin \mathcal{I}^{k-1}} \frac{(\boldsymbol{a}_n)^{\mathrm{T}} \boldsymbol{y}^{k-1}}{\|\boldsymbol{a}_n\|_2}$
Stop if $(\boldsymbol{a}_{i_k})^{\mathrm{T}} \boldsymbol{y}^{k-1} < 0$.
2) $\mathcal{I}^k = \mathcal{I}^{k-1} \cup i_k$
3) $\boldsymbol{A}_{\mathcal{I}^k} = [\boldsymbol{A}_{\mathcal{I}^{k-1}}, \boldsymbol{a}_{i_k}]$
4) $\boldsymbol{w}_{\mathcal{I}^k} = (\boldsymbol{A}_{\mathcal{I}^k}^{\mathrm{T}} \boldsymbol{A}_{\mathcal{I}^k})^{-1} \boldsymbol{A}_{\mathcal{I}^k}^{\mathrm{T}} \boldsymbol{y}$
5) $\boldsymbol{y}^k = \boldsymbol{r} - \boldsymbol{A}_{\mathcal{I}^k} \boldsymbol{x}_{\mathcal{I}^k}$
end for
6) compute $\widetilde{\boldsymbol{A}} = \boldsymbol{A}(\mathcal{I}^k)$ and $\widetilde{\boldsymbol{w}}^0 = \boldsymbol{w}_{\mathcal{I}^k}$
for $t = 1, 2, \ldots$ **do**
7) $\widetilde{\boldsymbol{v}}^{t-1} = \widetilde{\boldsymbol{w}}^{t-1} - \mu \nabla f(\widetilde{\boldsymbol{w}}^{t-1})$
8) $\widetilde{\boldsymbol{w}}^t = P(\widetilde{\boldsymbol{v}}^{t-1})$
Stop if stopping criterion is met.
end for
Output: $\boldsymbol{w}(\mathcal{I}^k) = \widetilde{\boldsymbol{w}}^t$

B. Explicitly Control Fitting Error

This section develops our ADMM-ℓ_0 algorithm, which can directly limit the fitting error. In the algorithm, the sparse index tracking is formulated as

$$\begin{aligned} &\min_{\boldsymbol{w}} \ \|\boldsymbol{w}\|_0 \\ &\text{s.t. } \boldsymbol{w} \succeq 0, \quad \boldsymbol{w}^T \boldsymbol{1} = 1, \ \text{and} \ \ \|\boldsymbol{A}\boldsymbol{w} - \boldsymbol{r}\|_2^2 \leq \epsilon \end{aligned} \tag{15}$$

where $\epsilon > 0$ is a user-defined tolerance parameter, which indicates the maximum affordable deviation from the target index performance. Since ℓ_0-norm minimization is subject to three conditions, (15) is an intractable problem. We suggest to use the decoupling concept to handle it. With the decoupling concept, (15) becomes

$$\begin{aligned} &\min_{\boldsymbol{w},\boldsymbol{z}} \ ||\boldsymbol{w}||_0 + \mathbb{I}(\boldsymbol{z}) \\ &\text{s.t. } \boldsymbol{w} = \boldsymbol{z} \end{aligned} \tag{16}$$

where $\mathbb{I}(\boldsymbol{z})$ is an indicator function [58], [59], which is used to report whether $\boldsymbol{z}$ is in the set

$$\mathcal{C} = \left\{\boldsymbol{z} | \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{r}\|_2^2 \leq \epsilon, \boldsymbol{z}^T \boldsymbol{1} = 1, \boldsymbol{z} \succeq 0\right\}$$

or not. If $\boldsymbol{z} \in \mathcal{C}$, then $\mathbb{I}(\boldsymbol{z}) = 0$; otherwise, $\mathbb{I}(\boldsymbol{z}) = +\infty$.

Since it is difficult to directly utilize $\mathbb{I}(\boldsymbol{z})$ to deal with (16), we approximate $\mathbb{I}(\boldsymbol{z})$ with an approximation indicator function $g(\boldsymbol{z})$ given by

$$\begin{aligned} g(\boldsymbol{z}) = \lambda_1 \Bigg(&\max\left\{0, \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{r}\|_2^2 - \epsilon\right\}^2 + \left(\boldsymbol{z}^T \boldsymbol{1} - 1\right)^2 \\ &+ \sum_{n=0}^{N} \max\{0, -z_n\}^2 \Bigg). \end{aligned} \tag{17}$$

It can be found that if $\boldsymbol{z} \in \mathcal{C}$, then $g(\boldsymbol{z}) = 0$; otherwise, $g(\boldsymbol{z}) > 0$. If λ_1 is large enough, then $g(\boldsymbol{z})$ is equivalent to $\mathbb{I}(\boldsymbol{z})$. From our experience, λ_1 around 10^4 is good enough.

Based on (16) and (17), the augmented Lagrangian is established as follows:

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\gamma}) = ||\boldsymbol{w}||_0 + g(\boldsymbol{z}) + \boldsymbol{\gamma}^T (\boldsymbol{w} - \boldsymbol{z}) + \frac{\lambda_2}{2} ||\boldsymbol{w} - \boldsymbol{z}||_2^2 \tag{18}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^N$ contains Lagrange multipliers and the last term is the augmented term, which helps convexifying the original problem and the choice of $\lambda_2 > 0$ is quite flexible as long as its value is sufficiently large. From our experience, λ_2 around 10^4 is good enough.

Adopting the ADMM idea, we update $\boldsymbol{w}$, $\boldsymbol{z}$, and $\boldsymbol{\gamma}$ in an alternating and iterative manner, which is given by

$$\boldsymbol{w}^t = \arg\min_{\boldsymbol{w}} \mathcal{L}\left(\boldsymbol{w}, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) \tag{19a}$$

$$\boldsymbol{z}^t = \arg\min_{\boldsymbol{z} \in \mathcal{C}} \mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}, \boldsymbol{\gamma}^{t-1}\right) \tag{19b}$$

$$\boldsymbol{\gamma}^t = \boldsymbol{\gamma}^{t-1} + \lambda_2 \left(\boldsymbol{w}^t - \boldsymbol{z}^t\right) \tag{19c}$$

that is, this iteration process is to compute a saddle point of (18). In (19), $(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ denotes the result of the tth iteration. The details of solving (19a)–(19c) are given as follows.

Update of $\boldsymbol{w}$: Ignoring the constant terms in (19a), the update of (19a) becomes

$$\boldsymbol{w}^t = \arg\min_{\boldsymbol{w}} \ \|\boldsymbol{w}\|_0 + \frac{\lambda_2}{2} \left\|\boldsymbol{w} - \boldsymbol{b}^{t-1}\right\|_2^2 \tag{20}$$

where $\boldsymbol{b}^{t-1} = \boldsymbol{z}^{t-1} - \boldsymbol{\gamma}^{t-1}/\lambda_2$. Unlike the familiar ℓ_0-norm minimization in [60] and [61], each component in $\boldsymbol{w}$ in (20) is independent of each other. Therefore, (20) can be decomposed into N scalar subproblems, which is given by

$$\min_{w_n} \ \delta(w_n) + \frac{\lambda_2}{2} \left(w_n - b_n^{t-1}\right)^2. \tag{21}$$

where b_n^{t-1} denotes the nth element of $\boldsymbol{b}^{t-1}$) and $\delta(w_n)$ is an indicator function. If $w_n = 0$, then $\delta(w_n) = 0$; otherwise, $\delta(w_n) = 1$. The closed-form solution for (20) is then given by

$$\boldsymbol{w}^t = T_{\frac{\lambda_2}{2}}\left(\boldsymbol{b}^{t-1}\right) = \begin{cases} b_n^{t-1}, & \text{if } \left|b_n^{t-1}\right| \geq \sqrt{\dfrac{2}{\lambda_2}} \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

where $T_{\lambda_2/2}(b^{t-1})$ is a hard-thresholding operator. When b^{t-1} is less than $\sqrt{(2/\lambda_2)}$, we set it to zero; otherwise, we keep its original value. The derivation of the closed-form solution is given in Appendix A.

Update of $\boldsymbol{z}$: Since the ℓ_0-norm term in (19b) is independent of $\boldsymbol{z}$, the update of $\boldsymbol{z}$ can be simplified as

$$\begin{aligned}\boldsymbol{z}^t = \arg\min_{\boldsymbol{z}} \frac{\lambda_2}{2}\left\|\boldsymbol{w}^t - \boldsymbol{z} + \frac{1}{\lambda_2}\boldsymbol{\gamma}^{t-1}\right\|_2^2 \\ + \lambda_1\Bigg(\left(\max\left\{0, \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{r}\|_2^2 - \epsilon\right\}\right)^2 + \left(\boldsymbol{z}^T\boldsymbol{1}-1\right)^2 \\ + \sum_{n=1}^{N}(\max\{0,-z_n\})^2\Bigg)\end{aligned} \tag{23}$$

where all terms are convex and the feasible region is also convex, and hence, (23) corresponds to a convex optimization. GD [62] can be employed to tackle it. The gradient with respect to $\boldsymbol{z}$ is

$$\begin{aligned}&\nabla_z\mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}, \boldsymbol{\gamma}^{t-1}\right) \\ &= -\lambda_2\left(\boldsymbol{w}^t - \boldsymbol{z} + \frac{1}{\lambda_2}\boldsymbol{\gamma}^{t-1}\right) \\ &\quad + 2\lambda_1\Big(\delta\left(0, \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{r}\|_2^2-\epsilon\right)\left(\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{r}\|_2^2-\epsilon\right)\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{z}-\boldsymbol{r}) \\ &\qquad + \left(\boldsymbol{z}^{tT}\boldsymbol{1}-1\right)\boldsymbol{1} + \min\{\boldsymbol{0}, \boldsymbol{z}\}\Big)\end{aligned} \tag{24}$$

where $\min\{\boldsymbol{0}, \boldsymbol{z}\}$ is performed elementwise and $\varphi(0, \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{r}\|_2^2-\epsilon)$ is

$$\varphi\left(0, \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{r}\|_2^2-\epsilon\right) = \begin{cases} 1, & \text{if } \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{r}\|_2^2-\epsilon > 0 \\ 0, & \text{otherwise.}\end{cases} \tag{25}$$

Then, the update of $\boldsymbol{z}$ becomes

$$\boldsymbol{z}^l = \boldsymbol{z}^{l-1} - \mu\nabla_z\mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^{l-1}, \boldsymbol{\gamma}^{t-1}\right) \tag{26}$$

where $\mu > 0$ is the step size. Note that superscript l denotes the iteration number of the GD process, where $\boldsymbol{z}^{l=0} = \boldsymbol{z}^t$. In theory, $\boldsymbol{z}^t = \boldsymbol{z}^l$ when $\nabla_z\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^l, \boldsymbol{\gamma}^{t-1}) = \boldsymbol{0}$.

Update of $\boldsymbol{\gamma}$: For (19c), the update is quite straightforward.

Our proposed ADMM-ℓ_0 is summarized in Algorithm 2. It is worth noting that the ADMM-ℓ_0 algorithm contains two layers of iterations. The outer iteration is to update $(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ via ADMM, while the inner iteration is optimize $\boldsymbol{z}^l$ via GD. In our implementation, the inner and outer stopping conditions are $||\boldsymbol{z}^l - \boldsymbol{z}^{l-1}||_2 \le 10^{-7}$ and $||\boldsymbol{w}^t - \boldsymbol{w}^{t-1}||_2 \le 10^{-7}$, respectively.

C. Convergence Analysis

For our NNOMP-PGD, it consists of two procedures: the convergence of the NNOMP procedure is analyzed in [46] and [63] and the convergence of the PGD procedure is provided in [49] and [64]. Therefore, the convergence of our NNOMP-PGD is guaranteed because both NNOMP and PGD converge.

We now focus on analyzing the convergence of our ADMM-ℓ_0. The flow of our analysis is similar to that of the nonconvex ADMM convergence analysis [65].

First, in our ADMM-ℓ_0, we have the following two properties.

P1: For each t, there exists $\tau > 0$ such that

$$\mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t\right) - \mathcal{L}\left(\boldsymbol{w}^{t-1}, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) \le -\tau\left\|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\right\|_2^2.$$

P2: $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ is lower bounded.

Algorithm 2 Our ADMM-ℓ_0

Input: $\boldsymbol{A}$, $\boldsymbol{r}$, μ, ϵ λ_1 and λ_2
  Initialize: $\boldsymbol{\gamma}^0 = \boldsymbol{0}$ and $\boldsymbol{z}^0$ computed by NNOMP
  for $t = 1, 2, \ldots$ **do**
    1) $\boldsymbol{b}^{t-1} = \boldsymbol{z}^{t-1} - \boldsymbol{\gamma}^{t-1}/\lambda_2$
    2) $\boldsymbol{w}^t = T_{(\lambda_2/2)}(\boldsymbol{b}^{t-1})$
    3) $\boldsymbol{z}^{l=0} = \boldsymbol{z}^{t-1}$
    for $l = 1, 2, \ldots$ **do**
      4) $\boldsymbol{z}^l = \boldsymbol{z}^{l-1} - \mu\nabla_z\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}, \boldsymbol{\gamma}^{t-1})$
      Stop if stopping criterion is met.
    end for
    5) $\boldsymbol{z}^t = \boldsymbol{z}^l$
    6) $\boldsymbol{\gamma}^t = \boldsymbol{\gamma}^{t-1} + \lambda_2(\boldsymbol{w}^t - \boldsymbol{z}^t)$
    Stop if stopping criterion is met.
  end for
Output: $\boldsymbol{w}^t$

Proof: The proof of **P1** and **P2** is given in Appendix B.

From **P1** and **P2**, we have Theorem 1.

Theorem 1: Since the ADMM-ℓ_0 satisfies **P1** and **P2**, $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ converges.

Proof: **P1** indicates that $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ is monotonically nonincreasing. From **P2**, $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ is lower bounded, and thus, the convergence of $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ is guaranteed.

Furthermore, the dynamic behavior of the sequence $\{\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t\}$ is provided in Theorem 2.

Theorem 2: Based on Theorem 1, $||\boldsymbol{w}^t - \boldsymbol{w}^{t-1}||_2^2 \to 0$, $||\boldsymbol{z}^t - \boldsymbol{z}^{t-1}||_2^2 \to 0$, and $||\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1}||_2^2 \to 0$, as $t \to \infty$.

Proof: The proof is given in Appendix C.

D. Complexity Analysis

In our NNOMP-PGD, the computational complexities of Steps 1 and 4 are $\mathcal{O}(MN)$ and $\mathcal{O}(K^2M)$, respectively. Since $K \ll N$ in sparse index tracking and it is repeated at most K times, the overall complexity of NNOMP is $\mathcal{O}(KMN)$.

For PGD, the computational complexity is dominated by calculating $\nabla f(\widetilde{\boldsymbol{w}}^t)$ whose complexity is $\mathcal{O}(2KM)$. Hence, the computational complexity of PGD is $\mathcal{O}(2TKM)$, where T is the maximum iteration number. In general, $2T > N$, and therefore, NNOMP-PGD has the complexity of $\mathcal{O}(2TKM)$.

For our ADMM-ℓ_0, the computational complexity is dominated by $\nabla_z\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}, \boldsymbol{\gamma}^{t-1})$, which corresponds to $\mathcal{O}(MN)$. Hence, the overall complexity of ADMM-ℓ_0 is $\mathcal{O}(TLMN)$, where T and L are the maximum iteration numbers of inner and outer loops, respectively.

IV. Experiments

A. Datasets and Settings

Our experiments consider three real-world datasets, namely, Russell 2000, S&P 500, and NASDAQ 100. There are commonly used datasets in sparse index tracking [7], [22], [66]. According to the common practice [7], [22], [66], the stocks

TABLE I
DATASET INFORMATION

Dataset	Period	Total Day No.	D_{train}	D_{op}
Russell 2000	03/08/2015–02/06/2020	1200	200	100
S&P 500	03/08/2015–02/06/2020	1200	200	100
NASDAQ100	03/08/2015–02/06/2020	1200	200	100

that do not cover the whole period are excluded. Therefore, the used datasets extracted from Russell 2000, S&P 500, and NASDAQ 100 contain 1544, 437, and 80 stocks, respectively. The details of the datasets are listed in Table I, which contains the time period of dataset, total days D, training days D_{train}, and testing days D_{op}.

We adopt the moving window scheme [22], shown in Fig. 2, to test all algorithms. This scheme has several advantages. First, it is close to the practical situation, in which the portfolio is run for a duration and then is rebalanced. With the rebalance, the portfolio can adopt to the change in the market environment. Second, it is able to reduce the impact of data characteristics, namely, stable, recessionary, and bubbly markets.

We begin with the first window covering from the first day to the first $D_{\text{train}} + D_{\text{op}}$ days. The first D_{train} day is utilized to train $\boldsymbol{w}$, and then, the remaining D_{op} days are utilized to evaluate the performance. Then, we roll the window D_{op} days to get the second window. In a similar manner, a new $\boldsymbol{w}$ is obtained and is tested based on the second operation window. It can be seen that the number of days for training is $D - D_{\text{op}}$, while the number of days for testing is $D - D_{\text{train}}$.

Based on the moving window method, the magnitude of the daily tracking error (MDTE) [22] is utilized to evaluate the estimation performance, which is defined as follows:

$$\text{MDTE} = \frac{1}{D - D_{\text{train}}} \sum_{j=1}^{D-D_{\text{train}}} \left\| r_j - \boldsymbol{x}_j{}^T \boldsymbol{w}_j \right\|_2 \tag{27}$$

where r_j is the return of the market index in the jth day in the testing days, $\boldsymbol{x}_j$ is the returns of the stocks in the jth day in the testing days, and $\boldsymbol{w}_j$th is the portfolio used in the jth day in the testing days. The MDTE value is presented in basis points (bps) where 1 pbs is equivalent to 10^{-4}.

B. Convergence Behavior

In this section, we verify Theorems 1 and 2 using empirical results based on the S&P 500 and Russell 2000 datasets. Fig. 4 shows the convergence behavior of $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$. We can see that the value of $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ decreases with the number of iterations and converges within around 80 iterations.

Fig. 5 shows the dynamics of the estimated weights. It can be seen that there are no big changes in the estimated weights after around 80 iterations. Thereby, Theorem 2 is verified. In addition, all the estimated weights become greater than or equal to zero around 80 iterations.

C. Comparison With Existing Methods

In this section, seven algorithms are considered. The NNOMP-PGD and ADMM-ℓ_0 are our proposed algorithms.

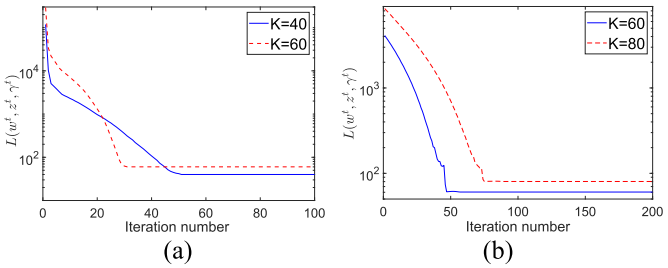

Fig. 4. Convergence of $L(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ in ADMM-ℓ_0. (a) S&P 500. (b) Russell 2000.

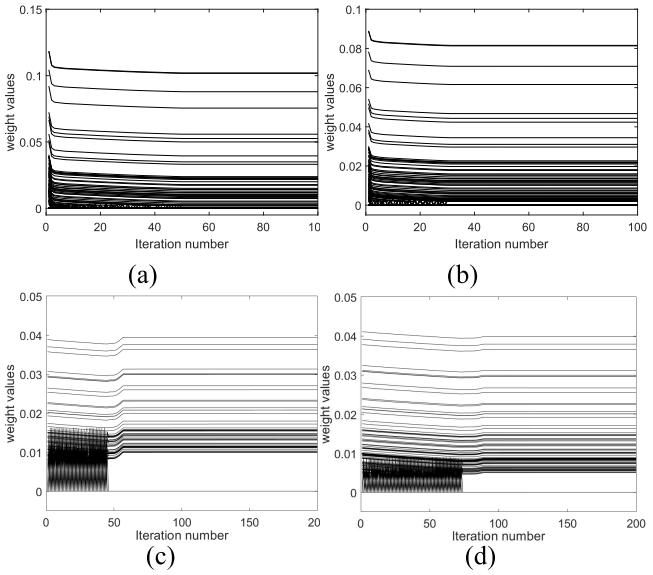

Fig. 5. Dynamics of the weight values in ADMM-ℓ_0. (a) S&P 500 ($K = 40$). (b) S&P 500 ($K = 60$). (c) Russell 2000 ($K = 60$). (d) Russell 2000 ($K = 80$).

Five existing algorithms are included for comparison. They are specialized linear approximation for index tracking (SLAIT) [22], accelerated SLAIT (ASLAIT) [22], IT-Aenet [24], LASSO [23], and a modified nonsparse (MNS) algorithm [67]. In the MNS, we first construct a nonsparse portfolio via the CVX solver based on (2). We then select the assets with the largest weights and apply the normalization. Note that we cannot use the solution of the CVX solver as the comparison algorithm because this nonsparse solution should be nearly identical to the full replication concept.

Figs. 6–8 compare the MDTE performances of various algorithms. Those figures show MDTE versus sparsity K of the portfolio.

Since the SLAIT, ASLAIT, and LASSO algorithms cannot directly control the sparsity value K and the training set fitting error, we tune their regularization parameter values such that the sparsity values of the resultant portfolios meet the target values.

For the IT-Aenet algorithm, it does not have an efficient tuning mechanism and we use the recommendation values in [24], and thus, we only have a point for IT-Aenet in the figures. Our ADMM-ℓ_0 is able to directly control the fitting error of the resultant portfolio, but it cannot directly control

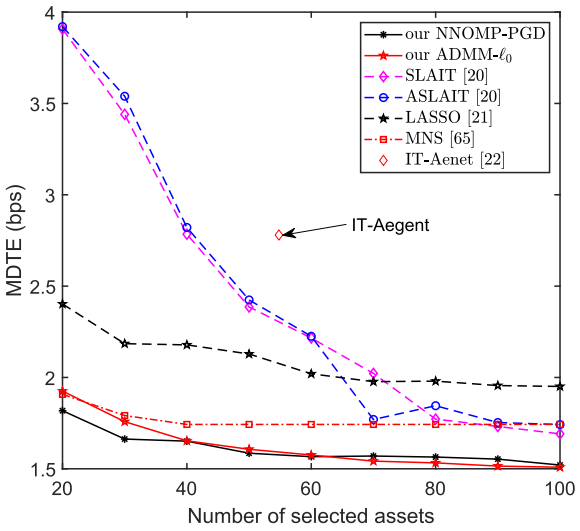

Fig. 6. MDTE of different algorithms on Russell 2000.

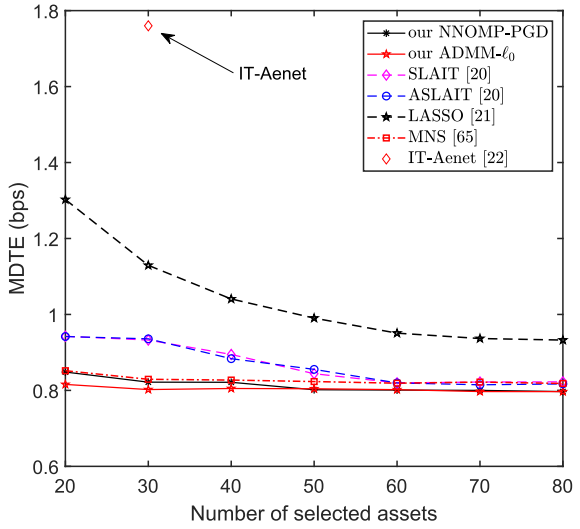

Fig. 7. MDTE of different algorithms on S&P 500.

the sparsity of the resultant portfolio. Hence, we try different fitting error values such that the sparsity of the resultant portfolio meets the target sparsity values.

From the figures, our proposed algorithms outperform SLAIT, ASLAIT, LASSO, and IT-Aenet in terms of MDTE. Under the same sparsity, the proposed algorithms obtain smaller MDTE than SLAIT, ASLAIT, LASSO, and IT-Aenet. The reason may be that our methods adopt the ℓ_0-norm, while SLAIT, ASLAIT, LASSO, and IT-Aenet employ the approximate ℓ_0-norm. For MNS, it has a smaller MDTE than ADMM-ℓ_0 with 20 assets on Russell 2000. Besides, it is better than ADMM-ℓ_0 with ten assets on NASDAQ 100. In other cases, the proposed methods are superior to MNS.

For our two proposed algorithms, it is hard to judge which is better in this experiment. Note that the two proposed algorithms serve for different investor's preferences. With the NNOMP-PGD, investors could explicitly and directly specify the number of assets and minimize the tracking error. On the other hand, with the ADMM-ℓ_0, investors could explicitly and directly specify the tracking error and aims at minimizing the number of the selected assets.

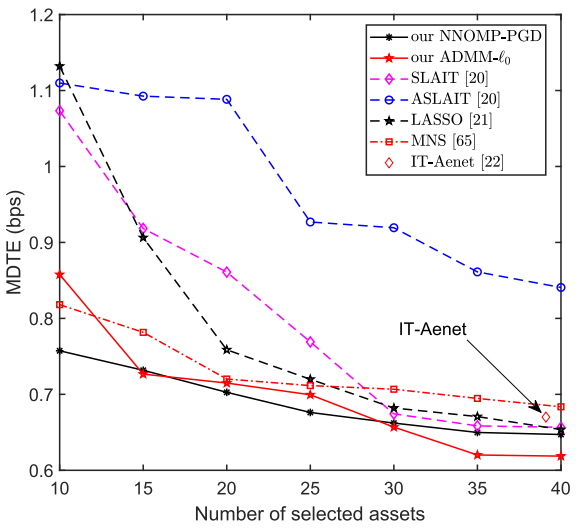

Fig. 8. MDTE of different algorithms on NASDAQ 100.

For IT-Aenet, its sparsity is determined by three penalty parameters, and hence, it is difficult to tune parameters to attain the same sparsity among all windows. Thereby, the IT-Aenet obtains different sparsity levels in different windows. Therefore, we perform a detailed comparison between the IT-Aenet and our algorithms. Table II lists the comparison results. For each training window, we tune the parameters of our proposed algorithms such that their sparsity levels are the same as those of IT-Aenet. It is seen that the proposed methods obtain lower MDTEs than IT-Aenet in all windows.

Furthermore, we study the performance of all algorithms under different market environments, namely, stabilization, recession, and bubble. Hence, we extract three different volatility periods from our datasets. Table III shows the comparison results. It can be seen that our NNOMP-PGD algorithm has smaller MDTE than SLAIT, ASLAIT, IT-Aenet, and LASSO in different datasets and investment environments.

Regarding our ADMM-ℓ_0, it is superior to SLAIT, ASLAIT, IT-Aenet, and LASSO in all conditions on the Russell 2000 dataset. It is a bit poorer than MMS for the recession environments. For S&P 500, the tracking error of our ADMM-ℓ_0 is a bit greater than that of ASLAIT for the recession situation, Also, it is a bit poorer than that of MNS for the bubble environment. For NASDAQ 100, our ADMM-ℓ_0 is a bit poorer than the MNS algorithm for the bubble environment. In general, our ADMM-ℓ_0 is superior to the competing algorithms.

In some situations, investors not only want to track market index but also want to obtain more profits than the market index. We investigate the earning performance of the designed sparse portfolios computed by different algorithms. Figs. 9 and 10 show the change of accumulated returns over time, that is, 1 dollar is invested, and then, the revenue varies with time going on the assumption of no transaction fee. The ground truth is calculated by the market index, namely, S&P

TABLE II

MDTEs OF THE PROPOSED ALGORITHMS AND IT-AENET IN DIFFERENT WINDOWS ON DIFFERENT DATASETS. THE SPARSITY LEVEL OF NNOMP-PGD AND ADMM-ℓ_0 IN EACH WINDOW IS SET WITH REFERENCE TO THE SPARSITY LEVEL OF THE SOLUTION COMPUTED BY IT-AENET

Dataset		Window	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	Overall
Russell 2000		Sparsity	84	85	83	68	67	51	44	16	9	22	54.8
	MDTE value	NNOMP-PGD	5.49	4.83	4.25	3.95	5.13	3.60	6.31	5.44	5.83	17.53	1.76
		ADMM-ℓ_0	6.78	5.39	4.64	3.99	5.15	4.03	7.05	5.80	7.21	14.39	1.45
		IT-Aenet	9.58	9.08	7.34	6.67	8.76	6.39	13.71	9.44	9.87	26.98	2.78
S&P 500		Sparsity	34	32	27	13	17	48	36	40	33	18	30.0
	MDTE value	NNOMP-PGD	3.74	2.48	1.80	3.21	2.84	1.64	2.30	1.82	2.18	9.09	0.91
		ADMM-ℓ_0	4.56	2.26	2.02	2.37	2.69	1.66	2.62	2.52	1.82	8.30	0.83
		IT-Aenet	8.53	4.24	2.97	4.88	4.96	7.07	11.81	6.73	6.55	17.48	1.76
NASDAQ 100		Sparsity	42	41	39	42	35	36	39	41	42	34	39.1
	MDTE value	NNOMP-PGD	1.28	1.81	1.89	1.66	1.88	1.29	1.79	1.29	1.49	6.18	0.61
		ADMM-ℓ_0	1.32	1.88	1.61	1.86	2.03	1.96	1.94	1.31	1.75	6.24	0.63
		IT-Aenet	1.99	2.08	2.24	2.85	3.22	2.65	2.89	2.43	2.12	6.70	0.67

TABLE III

PERFORMANCE COMPARISON OF DIFFERENT ALGORITHMS IN DIFFERENT VOLATILITY CONDITIONS

Dataset	Volatility Condition		Stabilization	Recession	Bubble
Russell 2000	Sparsity	K	41	34	35
	our NNOMP-PGD	MDTE	3.63456	5.28631	4.13764
	our ADMM-ℓ_0	MDTE	3.71665	5.90655	4.05581
	ASLAIT [20]	MDTE	4.73555	9.66602	5.91326
	SLAIT [20]	MDTE	4.95069	9.81745	5.78104
	IT-Aenet [22]	MDTE	8.84680	20.1863	12.3010
	LASSO [21]	MDTE	7.04236	11.8809	9.89371
	MNS [65]	MDTE	3.74590	5.78365	4.07817
S&P 500	Sparsity	K	31	38	83
	our NNOMP-PGD	MDTE	1.32599	2.31485	1.05152
	our ADMM-ℓ_0	MDTE	1.33188	2.59954	1.17222
	ASLAIT [20]	MDTE	1.64346	2.53092	1.28659
	SLAIT [20]	MDTE	1.68770	2.72004	1.28083
	IT-Aenet [22]	MDTE	2.77384	6.15658	3.81617
	LASSO [21]	MDTE	3.03492	6.84620	1.84955
	MNS [65]	MDTE	1.43451	3.13376	1.06403
NASDAQ 100	Sparsity	K	10	39	28
	our NNOMP-PGD	MDTE	2.20643	1.76343	1.64927
	our ADMM-ℓ_0	MDTE	2.33211	1.73263	1.95529
	ASLAIT [20]	MDTE	2.50001	1.92163	2.40761
	SLAIT [20]	MDTE	3.05776	1.92274	2.48873
	IT-Aenet [22]	MDTE	3.40804	3.54744	2.22207
	LASSO [21]	MDTE	3.48859	3.94443	2.57729
	MNS [65]	MDTE	2.76912	2.74553	1.84726

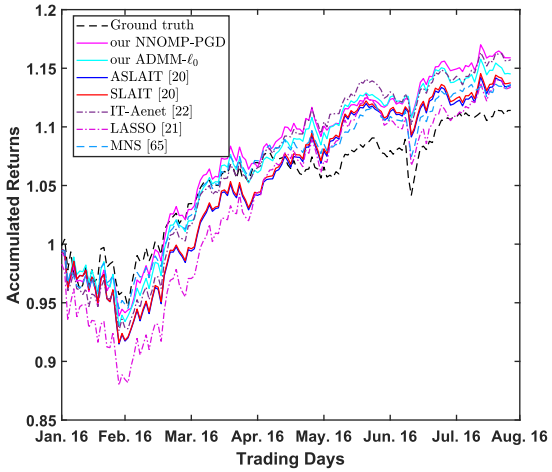

Fig. 9. S&P 500: trading days versus accumulated returns by the proposed algorithms with ASLAIT, SLAIT, IT-Aenet, LASSO, and MNS with the number of selected assets and training days being 25 and 110, respectively.

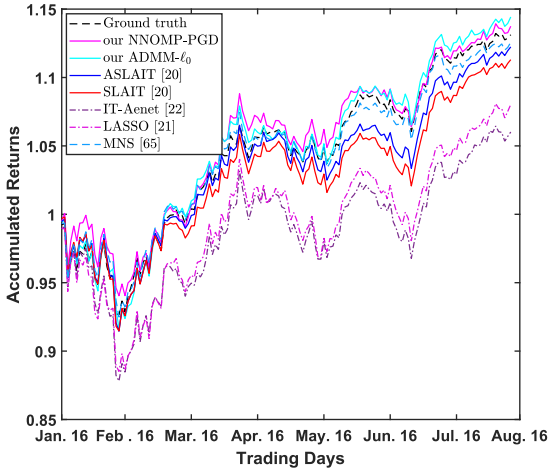

Fig. 10. NASDAQ 100: trading days versus accumulated returns by the proposed algorithms with ASLAIT, SLAIT, IT-Aenet, LASSO, and MNS with the number of selected assets and training days being 25 and 110, respectively.

500 in Fig. 9 and NASDAQ 100 in Fig. 10. The other lines represent the accumulated returns of designed sparse portfolios on different datasets. In this experiment, the first 110 days are utilized to train $\boldsymbol{w}$ with 25 assets, and then, the revenues of 150 days are plotted. In Fig. 9, it is observed that all designed sparse portfolios have more returns than the ground truth. It is clear that our algorithms attain more profits than SLAIT, ASLAIT, LASSO, and MNS. Besides, IT-Aenet is superior to ADMM-ℓ_0 but inferior to our NNOMP-PGD. On NASDAQ 100, the proposed algorithms outperform the ground truth. In contrast, SLAIT, ASLAIT, IT-Aenet, LASSO, and MNS are below the ground truth and have a slightly large daily MDTE.

Remark: It should be noticed that the SLAIT, ASLAIT, IT-Aenet, and LASSO do not have the ability to control the sparsity and the fitting error. On the other hand, our NNOMP-PGD is designed to directly control the sparsity

of the resultant portfolio and our ADMM-ℓ_0 is designed to control the fitting error of the resultant portfolio.

V. Conclusion

We have derived two effective algorithms based on the ℓ_0-norm to deal with the sparse index tracking problem. The NNOMP-PGD considers explicitly and directly controlling the number of assets. The ADMM-ℓ_0 considers explicitly and directly controlling the tracking error. In addition, the convergence of the proposed algorithms has been reported. Numerical experiments using real-world datasets have demonstrated that the proposed algorithms outperform several existing approaches. In general, the proposed algorithms have lower tracking errors. Also, the proposed algorithms provide better controllability on the sparsity or tracking error of the resultant portfolio.

Traditional sparse recovery usually does not consider sparsity, sum-to-one, and nonnegativity requirements simultaneously. Hence, one research direction is to develop more efficient and effective algorithms with high controllability.

In addition, there are several possible future works. Currently, we only consider two cases of controllability, either sparsity and tacking error. It is interesting to develop an algorithm to allow investors to simultaneously specify the predefined values on sparsity and tracking error. In this case, sparse index tracking becomes a constraint satisfaction problem. Also, to provide more flexibility for investors, it is interesting to set upper and lower bounds on the investment percentages on the selected assets. Last but not least, most of the existing index tracking algorithms do not consider the variance of the tracking error or the risk of the portfolio. Hence, it is suggested to add constraints to reduce the variance of the tracking error and the risk of the portfolio.

Appendix A
Closed-Form Solution of $\boldsymbol{w}$ in ADMM-ℓ_0

Optimization of $\boldsymbol{w}$ in (20) is equivalent to minimizing the sum of N scalar subproblems

$$\begin{aligned}\min_{\boldsymbol{w}} V(\boldsymbol{w}) &= \|\boldsymbol{w}\|_0 + \frac{\lambda_2}{2}\left\|\boldsymbol{w}-\boldsymbol{b}^{t-1}\right\|_2^2 \\ &= \sum_n^N \left(\delta(w_n) + \frac{\lambda_2}{2}\left(w_n^2 - 2w_n\left(b^{t-1}\right)_n + \left(b^{t-1}\right)_n^2\right)\right) \\ &= \sum_n^N \upsilon(w_n). \end{aligned} \tag{28}$$

Hence, it is sufficient to show that the minimum value of (28) can be attained by minimizing with respect to each w_n. Since the value of $\delta(w_n)$ is either 1 or 0, to calculate the minimizer of (28), we discuss two cases for each w_n, namely, $w_n = 0$ and $w_n \neq 0$. It is clear that if $w_n = 0$, then $\upsilon(w_n) = \lambda_2/2(b^{t-1})_n^2$. For $w_n \neq 0$, we know that $w_n = (b^{t-1})_n$ leads to the minimum value $g(w_n) = 1$. Therefore, the minimum of $\upsilon(w_n)$ in different cases is

$$\min \upsilon(w_n) = \begin{cases} \frac{\lambda_2}{2}\left(b^{t-1}\right)_n^2, & \text{if } w_n = 0 \\ 1, & \text{if } w_n = \left(b^{t-1}\right)_n. \end{cases} \tag{29}$$

If w_n is not zero, $(\lambda_2/2)(b^{t-1})_n^2$ must be less than or equal to 1, that is, $|(b^{t-1})_n| \geq (2/\lambda_2)^{1/2}$. Hence, the solution (22) has been proven. It is worth noting that the optimal solution to (20) is not unique when there exists $(b^{t-1})_n = (2/\lambda_2)^{1/2}$. The reason is that $\upsilon((b^{t-1})_n) = 0$ for $(b^{t-1})_n = (2/\lambda_2)^{1/2}$. Hence, (22) is the unique solution with $(b^{t-1})_n \neq (2/\lambda_2)^{1/2}$ and one of optimal solutions with $(b^{t-1})_n = (2/\lambda_2)^{1/2}$.

Appendix B
Proof of **P1** and **P2**

A. Proof of **P1**

For $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$, we have

$$\begin{aligned}&\mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t\right) - \mathcal{L}\left(\boldsymbol{w}^{t-1}, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) \\ &\quad= \mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) - \mathcal{L}\left(\boldsymbol{w}^{t-1}, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) \leftarrow \boldsymbol{w}\text{-update} \\ &\qquad+ \mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^{t-1}\right) - \mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) \leftarrow \boldsymbol{z}\text{-update} \\ &\qquad+ \mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t\right) - \mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^{t-1}\right) \leftarrow \boldsymbol{\gamma}\text{-update}. \end{aligned} \tag{30}$$

Because $\boldsymbol{w}^t$ is the optimum solution of $\mathcal{L}(\boldsymbol{w}, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1})$ based on (22) with fixed $\boldsymbol{z}^{t-1}$ and $\boldsymbol{\gamma}^{t-1}$, resulting in

$$\mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) - \mathcal{L}\left(\boldsymbol{w}^{t-1}, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) \leq 0. \tag{31}$$

On the other hand, since $\mathcal{L}(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\gamma})$ is strongly convex with respect to $\boldsymbol{z}$, the relationship between $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1})$ and $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^{t-1})$ is

$$\begin{aligned}\mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) \geq\ & \mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^{t-1}\right) \\ &+ \nabla_z \mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^{t-1}\right)^T \left(\boldsymbol{z}^{t-1} - \boldsymbol{z}^t\right) \\ &+ \frac{m}{2}\left\|\boldsymbol{z}^{t-1} - \boldsymbol{z}^t\right\|_2^2. \end{aligned} \tag{32}$$

As $\boldsymbol{z}^t$ is the optimum solution of $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}, \boldsymbol{\gamma}^{t-1})$ based on (26), which leads to $\nabla_z \mathcal{L}|_{(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^{t-1})} = \boldsymbol{0}$, then we attain

$$\mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^{t-1}\right) - \mathcal{L}\left(\boldsymbol{w}^t, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}\right) \leq -\frac{m}{2}\left\|\boldsymbol{z}^{t-1} - \boldsymbol{z}^t\right\|_2^2. \tag{33}$$

From (17), we know that $g(\boldsymbol{z}) = \lambda_1(\max\{0, \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{r}\|_2^2 - \epsilon\})^2 + (\boldsymbol{z}^T\boldsymbol{1} - 1)^2 + \sum_n^N (\max\{0, -z_n\})^2)$, and then, the gradient of $g(\boldsymbol{z})$ is

$$\begin{aligned}\nabla g(\boldsymbol{z}) = 2\lambda_1 \Big(&\delta\left(0, \|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{r}\|_2^2 - \epsilon\right)\left(\|\boldsymbol{A}\boldsymbol{z} - \boldsymbol{r}\|_2^2 - \epsilon\right)\boldsymbol{A}^T \\ &\times (\boldsymbol{A}\boldsymbol{z} - \boldsymbol{r}) + \left(\boldsymbol{z}^T\boldsymbol{1} - 1\right)\boldsymbol{1} + \min\{\boldsymbol{0}, \boldsymbol{z}\}\Big). \end{aligned} \tag{34}$$

Meanwhile, we have

$$\boldsymbol{\gamma}^t = \boldsymbol{\gamma}^{t-1} + \lambda_2\left(\boldsymbol{w}^t - \boldsymbol{z}^t\right) \tag{35}$$

$$\nabla_z \mathcal{L}|_{(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^{t-1})} = -\lambda_2\left(\boldsymbol{w}^t - \boldsymbol{z}^t + \frac{1}{\lambda_2}\boldsymbol{\gamma}^{t-1}\right) + \nabla g\left(\boldsymbol{z}^t\right) = \boldsymbol{0}. \tag{36}$$

Hence, we can attain $\boldsymbol{w}^t - \boldsymbol{z}^t = (1/\lambda_2)(\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1})$ from (35) and $\nabla g(\boldsymbol{z}^t) = \lambda_2(\boldsymbol{w}^t - \boldsymbol{z}^t + (1/\lambda_2)\boldsymbol{\gamma}^{t-1})$ from (36). Furthermore, based on these two equations, we obtain $\nabla g(\boldsymbol{z}^t) = \boldsymbol{\gamma}^t$.

Therefore, for updating $\boldsymbol{\gamma}$

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t) - \mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^{t-1}) \\
&\quad = (\boldsymbol{\gamma}^t)^T(\boldsymbol{w}^t - \boldsymbol{z}^t) - (\boldsymbol{\gamma}^{t-1})^T(\boldsymbol{w}^t - \boldsymbol{z}^t) \\
&\quad = (\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1})^T(\boldsymbol{w}^t - \boldsymbol{z}^t) \\
&\quad = \frac{1}{\lambda_2}\|\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1}\|_2^2 \\
&\quad = \frac{1}{\lambda_2}\|\nabla g(\boldsymbol{z}^t) - \nabla g(\boldsymbol{z}^{t-1})\|_2^2 \\
&\quad \le \frac{C}{\lambda_2}\|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2^2 \qquad (37)
\end{aligned}$$

where C is the Lipschitz continuous constant, and the last inequality is based on the property of Lipschitz continuous gradient. Combining (31), (33), and (37), we attain

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t) - \mathcal{L}(\boldsymbol{w}^{t-1}, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}) \\
&\qquad \le \left(\frac{C}{\lambda_2} - \frac{m}{2}\right)\|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2^2. \qquad (38)
\end{aligned}$$

This means that $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ is monotonically nonincreasing with $\lambda_2 \ge (2C/m)$.

B. Proof of P2

Based on **P1**, we get

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t) \\
&\quad = ||\boldsymbol{w}^t||_0 + (\boldsymbol{\gamma}^t)^T(\boldsymbol{w}^t - \boldsymbol{z}^t) + \frac{\lambda_2}{2}||\boldsymbol{w}^t - \boldsymbol{z}^t||_2^2 \\
&\qquad + \lambda_1\Bigg(\left(\max\left\{0, \|\boldsymbol{A}\boldsymbol{z}^t - \boldsymbol{r}\|_2^2 - \epsilon\right\}\right)^2 \\
&\qquad\quad + \left((\boldsymbol{z}^t)^T\boldsymbol{1} - 1\right)^2 + \sum_{n=1}^{N}\left(\max\{0, -(z^t)_n\}\right)^2\Bigg) \qquad (39) \\
&\quad = ||\boldsymbol{w}^t||_0 + \nabla g(\boldsymbol{z}^t)^T(\boldsymbol{w}^t - \boldsymbol{z}^t) + \frac{\lambda_2}{2}||\boldsymbol{w}^t - \boldsymbol{z}^t||_2^2 \\
&\qquad + \lambda_1\Bigg(\left(\max\left\{0, \|\boldsymbol{A}\boldsymbol{z}^t - \boldsymbol{r}\|_2^2 - \epsilon\right\}\right)^2 \\
&\qquad\quad + \left((\boldsymbol{z}^t)^T\boldsymbol{1} - 1\right)^2 \\
&\qquad\quad + \sum_{n=1}^{N}\left(\max\{0, -(z^t)_n\}\right)^2\Bigg). \qquad (40)
\end{aligned}$$

Since $g(\boldsymbol{z})$ is convex and Lipschitz continuous with $0 \preceq \boldsymbol{z} \preceq 1$, we obtain

$$g(\boldsymbol{w}) - g(\boldsymbol{z}) \le \nabla g(\boldsymbol{z})^T(\boldsymbol{w} - \boldsymbol{z}) + \frac{C}{2}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2 \qquad (41)$$

where C is the Lipschitz continuous constant, resulting in

$$g(\boldsymbol{z}) + \nabla g(\boldsymbol{z})^T(\boldsymbol{w} - \boldsymbol{z}) \ge g(\boldsymbol{w}) - \frac{C}{2}\|\boldsymbol{w} - \boldsymbol{z}\|_2^2. \qquad (42)$$

Plugging (42) into (39) yields

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t) \ge\ & ||\boldsymbol{w}^t||_0 + \left(\frac{\lambda_2}{2} - \frac{C}{2}\right)||\boldsymbol{w}^t - \boldsymbol{z}^t||_2^2 \\
& + \lambda_1\Bigg(\left(\max\left\{0, \|\boldsymbol{A}\boldsymbol{w}^t - \boldsymbol{r}\|_2^2 - \epsilon\right\}\right)^2 \\
& + \sum_{n=1}^{N}\left(\max\{0, -(w^t)_n\}\right)^2 \\
& + \left((\boldsymbol{w}^t)^T\boldsymbol{1} - 1\right)^2\Bigg) \qquad (43)
\end{aligned}$$

where if $\lambda_2 \ge C$, then we have $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t) \ge 0$. In addition, combining the conclusion of **P1** that $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ is monotonically nonincreasing with $\lambda_2 \ge (2C/m)$, then $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ converges if $\lambda_2 \ge \max((2C/m), C)$.

Appendix C
Proof of Theorem 2

For $\boldsymbol{z}^t$, based on (38), we have

$$\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t) - \mathcal{L}(\boldsymbol{w}^{t-1}, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1}) \le -\tau\|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2^2 \qquad (44)$$

where $\tau > 0$. Then, we get

$$\|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2^2 \le -\frac{1}{\tau}\left(\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t) - \mathcal{L}(\boldsymbol{w}^{t-1}, \boldsymbol{z}^{t-1}, \boldsymbol{\gamma}^{t-1})\right). \qquad (45)$$

Since $\mathcal{L}(\boldsymbol{w}^t, \boldsymbol{z}^t, \boldsymbol{\gamma}^t)$ has been proven to be convergent, we obtain $\|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2^2 \to 0$ with $t \to +\infty$.

Regarding $\boldsymbol{\gamma}^t$, according to (37), we attain

$$\|\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1}\|_2^2 \le C\|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2^2. \qquad (46)$$

Thus, $\|\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1}\|_2^2 \to 0$ with $t \to +\infty$ can be obtained because $\|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2^2 \to 0$ with $t \to +\infty$.

From (19c), we have $\boldsymbol{w}^t = (1/\lambda_2)(\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1}) + \boldsymbol{z}^t$. Thus, we have

$$\begin{aligned}
\|\boldsymbol{w}^t - \boldsymbol{w}^{t-1}\|_2^2 =\ & \Bigg\|\frac{1}{\lambda_2}(\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1}) + \boldsymbol{z}^t \\
& - \frac{1}{\lambda_2}(\boldsymbol{\gamma}^{t-1} - \boldsymbol{\gamma}^{t-2}) - \boldsymbol{z}^{t-1}\Bigg\|_2^2 \qquad (47) \\
\le\ & \Bigg(\left\|\frac{1}{\lambda_2}(\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1})\right\|_2 + \|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2 \\
& + \left\|\frac{1}{\lambda_2}(\boldsymbol{\gamma}^{t-1} - \boldsymbol{\gamma}^{t-2})\right\|_2\Bigg)^2 \qquad (48) \\
\le\ & 3\Bigg(\left\|\frac{1}{\lambda_2}(\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1})\right\|_2^2 + \|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2^2 \\
& + \left\|\frac{1}{\lambda_2}(\boldsymbol{\gamma}^{t-1} - \boldsymbol{\gamma}^{t-2})\right\|_2^2\Bigg). \qquad (49)
\end{aligned}$$

Inequality (48) comes from triangle inequality. Inequality (49) comes from the fact that $2ab \le a^2 + b^2$ for any real a and b. Since $\lim_{t\to 0}\|\boldsymbol{z}^t - \boldsymbol{z}^{t-1}\|_2^2 = 0$ and $\lim_{t\to 0}\|\boldsymbol{\gamma}^t - \boldsymbol{\gamma}^{t-1}\|_2^2 = 0$, (49) means that $\|\boldsymbol{w}^t - \boldsymbol{w}^{t-1}\|_2^2 \to 0$, as $t \to 0$.

REFERENCES

[1] R. J. Fuller, B. Han, and Y. Tung, "Thinking about indices and 'passive' versus active management," *J. Portfolio Manag.*, vol. 36, no. 4, pp. 35–47, 2010.

[2] B. G. Malkiel, "Passive investment strategies and efficient markets," *Eur. Financial Manage.*, vol. 9, no. 1, pp. 1–10, Mar. 2003.

[3] L. Bai, L. Cui, Z. Zhang, L. Xu, Y. Wang, and E. R. Hancock, "Entropic dynamic time warping kernels for co-evolving financial time series analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jul. 21, 2021, doi: 10.1109/TNNLS.2020.3006738.

[4] J. F. L. de Oliveira, E. G. Silva, and P. S. G. de Mattos Neto, "A hybrid system based on dynamic selection for time series forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 29, 2022, doi: 10.1109/TNNLS.2021.3051384.

[5] B. M. Barber and T. Odean, "Trading is hazardous to your wealth: The common stock investment performance of individual investors," *J. Finance*, vol. 55, no. 2, pp. 773–806, Dec. 2002.

[6] Z.-R. Lai, D.-Q. Dai, C.-X. Ren, and K.-K. Huang, "Radial basis functions with adaptive input and composite trend representation for portfolio selection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6214–6226, Dec. 2018.

[7] M.-F. Leung and J. Wang, "Minimax and biobjective portfolio selection based on collaborative neurodynamic optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 2825–2836, Jul. 2021.

[8] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 3, pp. 653–664, Mar. 2016.

[9] K. Benidis, Y. Feng, and D. P. Palomar, "Optimization methods for financial index tracking: From theory to practice," *Found. Trends Optim.*, vol. 3, no. 3, pp. 171–279, 2018.

[10] P. Das, N. Johnson, and A. Banerjee, "Online lazy updates for portfolio selection with transaction costs," in *Proc. Nat. Conf. Artif. Intell.*, Washington, DC, USA, Jul. 2013, pp. 1–7.

[11] Z.-R. Lai, L. Tan, X. Wu, and L. Fang, "Loss control with rank-one covariance estimate for short-term portfolio optimization," *J. Mach. Learn. Res.*, vol. 21, pp. 1–37, Jun. 2020.

[12] X. Yan and X. Su, *Linear Regression Analysis: Theory and Computing*. Singapore: World Scientific, 2009.

[13] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[14] S. Sharma, S. Chaudhury, and C. T. Jayadeva, "Block sparse variational Bayes regression using matrix variate distributions with application to SSVEP detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 1, pp. 351–365, Jan. 2022.

[15] R. Wang, N. Xiu, and C. Zhang, "Greedy projected gradient-Newton method for sparse logistic regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 527–538, Feb. 2020.

[16] X. Wu, X. Xu, J. Liu, H. Wang, B. Hu, and F. Nie, "Supervised feature selection with orthogonal regression and feature weighting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 1831–1838, May 2020.

[17] Y. Han, Y. Yang, Y. Yan, Z. Ma, N. Sebe, and X. Zhou, "Semisupervised feature selection via spline regression for video semantic recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 252–264, Feb. 2015.

[18] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[19] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[20] K. J. Oh, T. Y. Kim, and S. Min, "Using genetic algorithm to support portfolio optimization for index fund management," *Expert Syst. Appl.*, vol. 28, no. 2, pp. 371–379, Feb. 2005.

[21] C. Dose and S. Cincotti, "Clustering of financial time series with application to index and enhanced index tracking portfolio," *Phys. A, Stat. Mech. Appl.*, vol. 355, no. 1, pp. 145–151, Sep. 2005.

[22] K. Benidis, Y. Feng, and D. P. Palomar, "Sparse portfolios for high-dimensional financial index tracking," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 155–170, Jan. 2018.

[23] L. R. Sant'Anna, J. F. Caldeira, and T. P. Filomena, "Lasso-based index tracking and statistical arbitrage long-short strategies," *North Amer. J. Econ. Finance*, vol. 51, Jan. 2020, Art. no. 101055.

[24] L. Shu, F. Shi, and G. Tian, "High-dimensional index tracking based on the adaptive elastic net," *Quant. Finance*, vol. 20, no. 9, pp. 1513–1530, 2020.

[25] Y. Zheng, T. M. Hospedales, and Y. Yang, "Diversity and sparsity: A new perspective on index tracking," 2018, *arXiv:1809.01989*.

[26] N. Li, "Efficient sparse portfolios based on composite quantile regression for high-dimensional index tracking," *J. Stat. Comput. Simul.*, vol. 90, no. 8, pp. 1466–1478, Feb. 2020.

[27] A. A. P. Santos, "Beating the market with small portfolios: Evidence from Brazil," *Economia*, vol. 16, no. 1, pp. 22–31, Jan. 2015.

[28] K. Naumenko and O. Chystiakova, "An empirical study on the differences between synthetic and physical ETFs," *Int. J. Econ. Finance*, vol. 7, no. 3, pp. 24–35, Feb. 2015.

[29] M. Kosev and T. Williams, "Exchange-traded funds," *RBA Bull.*, pp. 51–59, Mar. 2011.

[30] E. A. Sandoval and R. N. Saens, "The conditional relationship between portfolio beta and return: Evidence from Latin America," *Cuadernos de Economía*, vol. 41, no. 122, pp. 65–89, Apr. 2004.

[31] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Two-stage nonnegative sparse representation for large-scale face recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 35–46, Jan. 2013.

[32] M. Hyder and K. Mahata, "An approximate l0 norm minimization algorithm for compressed sensing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 3365–3368.

[33] Y. Feng and D. P. Palomar, *A Signal Processing Perspective of Financial Engineering*, vol. 9. Delft, The Netherlands: Now Publishers, 2016.

[34] B. Shi, X. Bai, W. Liu, and J. Wang, "Face alignment with deep regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 183–194, Jan. 2016.

[35] X. Zhen *et al.*, "Multitarget sparse latent regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1575–1586, May 2018.

[36] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statist.*, vol. 58, no. 1, pp. 30–37, Feb. 2004.

[37] D. R. Hunter, "MM algorithms for generalized Bradley–Terry models," *Ann. Statist.*, vol. 32, no. 1, pp. 384–406, 2004.

[38] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[39] Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, "Low-rank structure learning via nonconvex heuristic recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 383–396, Mar. 2013.

[40] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B, Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[41] F. E. Harrell, Jr., *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis*. New York, NY, USA: Springer, 2015.

[42] Z.-R. Lai, P.-Y. Yang, L. Fang, and X. Wu, "Short-term sparse portfolio optimization based on alternating direction method of multipliers," *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 2547–2574, Oct. 2018.

[43] H. Zou and M. Yuan, "Composite quantile regression and the Oracle model selection theory," *Ann. Statist.*, vol. 36, no. 3, pp. 1108–1126, Jun. 2008.

[44] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.

[45] Y. Chen, Y. Ye, and M. Wang, "Approximation hardness for a class of sparse optimization problems," *J. Mach. Learn. Res.*, vol. 20, no. 38, p. 127, Feb. 2019.

[46] M. Yaghoobi, D. Wu, and M. E. Davies, "Fast non-negative orthogonal matching pursuit," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1229–1233, Sep. 2015.

[47] T. T. Nguyen, J. Idier, C. Soussen, and E.-H. Djermoune, "Non-negative orthogonal greedy algorithms," *IEEE Trans. Signal Process.*, vol. 67, no. 21, pp. 5643–5658, Nov. 2019.

[48] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Comput.*, vol. 19, no. 10, pp. 2756–2779, Aug. 2007.

[49] P. H. Calamai and J. J. Moré, "Projected gradient methods for linearly constrained problems," *Math. Program.*, vol. 39, no. 1, pp. 93–116, Sep. 1987.

[50] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Delft, The Netherlands: Now Publishers, 2011.

[51] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.

[52] H. Liu, Y. Liu, and F. Sun, "Robust exemplar extraction using structured sparse coding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1816–1821, Aug. 2015.

[53] W. F. Sharpe, "Capital asset prices: A theory of market equilibrium under conditions of risk," *J. Finance*, vol. 19, no. 3, pp. 425–442, Sep. 1964.

[54] H. Zou, "The adaptive lasso and its Oracle properties," *J. Amer. Stat. Assoc.*, vol. 101, no. 476, pp. 1418–1429, Jan. 2012.

[55] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statist. Soc. B, Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, Mar. 2005.
[56] R. K. Arora, *Optimization: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2015.
[57] I. Babuška, "The finite element method with penalty," *Math. Comput.*, vol. 27, no. 122, pp. 221–228, 1973.
[58] K. Q. Ye, "Indicator function and its application in two-level factorial designs," *Ann. Statist.*, vol. 31, no. 3, pp. 984–994, Jun. 2003.
[59] Ö. Yeniay, "Penalty function methods for constrained optimization with genetic algorithms," *Math. Comput. Appl.*, vol. 10, no. 1, pp. 45–56, Apr. 2005.
[60] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[61] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
[62] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[63] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.
[64] J. C. Dunn, "Global and asymptotic convergence rate estimates for a class of projected gradient processes," *SIAM J. Control Optim.*, vol. 19, no. 3, pp. 368–400, May 1981.
[65] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, Jan. 2019.
[66] G. Guastaroba and M. G. Speranza, "Kernel search: An application to the index tracking problem," *Eur. J. Oper. Res.*, vol. 217, no. 1, pp. 54–68, Feb. 2012.
[67] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming, version 2.1," Tech. Rep., 2014. [Online]. Available: Available: http://cvxr.com/cvx

Xiao Peng Li received the B.Eng. degree in electronic science and technology from Yanshan University, Qinhuangdao, China, in 2015, and the M.Sc. degree in electronic information engineering from the City University of Hong Kong, Hong Kong, China, in 2018, where he is currently pursuing the Ph.D. degree in electrical engineering, supervised by Prof. So Hing Cheung.

From 2018 to 2019, he was a Research Assistant with the Department of Information Engineering, Shenzhen University, Shenzhen, China. His research interests include deep neural networks, sparse recovery, matrix processing, tensor processing and their applications on image recovery, video restoration, hyperspectral unmixing, as well as stock market analysis.

Zhang-Lei Shi received the Ph.D. degree from the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, in 2021.

He is currently a Lecturer with the College of Science, China University of Petroleum (East China), Qingdao, China. His current research interests include neural networks and machine learning.

Chi-Sing Leung (Senior Member, IEEE) received the Ph.D. degree in computer science from The Chinese University of Hong Kong, Hong Kong, in 1995.

He is currently a Professor with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He has published over 120 journal articles in the areas of digital signal processing, neural networks, and computer graphics. His research interests include neural computing and computer graphics.

Dr. Leung was a member of the Organizing Committee of ICONIP 2006. He was the Program Chair of ICONIP 2009 and ICONIP 2012. In 2005, he received the 2005 IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award for his paper titled, "The Plenoptic Illumination Function." He serves as an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and *Neural Processing Letters*. He is/was a Guest Editor of several journals, including *Neural Computing and Applications*, *Neurocomputing*, and *Neural Processing Letters*. He is a Governing Board Member of the Asian Pacific Neural Network Society (APNNS) and the Vice President of APNNS.

Hing Cheung So (Fellow, IEEE) was born in Hong Kong. He received the B.Eng. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 1990, and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong, Hong Kong, in 1995.

From 1990 to 1991, he was an Electronic Engineer with the Research and Development Division, Everex Systems Engineering Ltd., Hong Kong. From 1996 to 1999, he was a Research Assistant Professor with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, where he is currently a Professor. His research interests include detection and estimation, adaptive algorithms, robust signal processing, source localization, and sparse approximation.

Dr. So was on the Editorial Board of *IEEE Signal Processing Magazine* from 2014 to 2017 and IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2010 to 2014. He has been on the Editorial Board of *Signal Processing* since 2010 and *Digital Signal Processing* since 2011. He was a Lead Guest Editor of IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING—Special Issue on Advances in Time/Frequency Modulated Array Signal Processing in 2017. In addition, he was an elected member of the Signal Processing Theory and Methods Technical Committee, IEEE Signal Processing Society, from 2011 to 2016, where he was the Chair of the Awards Subcommittee from 2015 to 2016.