# A descent algorithm for constrained LAD Lasso estimation with applications in portfolio selection

Yue Shi[a], Chi Tim Ng[b], Zhiguo Feng[c], Yiu, Ka-fai, Cedric[d]

[a] School of Mathematics and Systems Science, Beihang University, Beijing, China.

[b] Department of Statistics, Chonnam National University, Gwangju 500-757, Korea

[c] Department of Mathematics and Computer Science, Guangdong Ocean University, Guangdong, China.

[d] Department of Applied Mathematics, The Hong Kong Polytechnic University, Kowloon, Hong Kong.

**ARTICLE HISTORY**
Compiled December 21, 2018

**ABSTRACT**
To improve the out-of-sample performance of the portfolio, Lasso regularization is incorporated to the Mean Absolute Deviance (MAD) based portfolio selection method. It is shown that such a portfolio selection problem can be reformulated as a constrained Least Absolute Deviance (LAD) problem with linear equality constraints. Moreover, we propose a new descent algorithm based on the ideas of "nonsmooth optimality conditions" and "basis descent direction set". The resulting MAD-Lasso method enjoys at least two advantages. First, it does not involve the estimation of covariance matrix that is difficult particularly in the high-dimensional settings. Second, sparsity is encouraged. This means that assets with weights close to zero in the Markovwitz's portfolio are driven to zero automatically. This reduces the management cost of the portfolio. Extensive simulation and real data examples indicate that if the Lasso regularization is incorporated, MAD portfolio selection method is consistently improved in terms of out-of-sample performance, measured by Sharpe ratio and sparsity. Moreover, simulation results suggest that the proposed descent algorithm is more time-efficient than interior point method and ADMM algorithm.

**KEYWORDS**
MAD-Lasso; portfolio selection; constrained LAD Lasso; linear equality constraints; interior point method; ADMM; nonsmooth optimality conditions; Sharpe ratio; sparsity.

## 1. Introduction

The mean-variance framework of Markowitz [17] is the cornerstone for modern portfolio selection theory. Under this framework, in order to balance the risk and return, the portfolio variance is minimized at a given level of expected return. This entails the estimation of the mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. However, as shown in [2, 6, 9, 13], if the sample mean and sample covariance are taken as the estimation of $\boldsymbol{\mu}, \boldsymbol{\Sigma}$, the out-of-sample performance of the asset allocation is not satisfactory in practice. In the context of regression analysis, it is well known that Least Absolute Deviance (LAD) is more robust and resistant to outliers in the response compared to

CONTACT A. N. Author. Email: yue.shi@connect.polyu.hk

the usual Least Square (LS) regression, see [11, 20, 21]. The statistical properties of the constrained Lasso estimates are studied in [10, 12]. As an analogy, it is natural to believe that in portfolio selection problem, the out-of-sample performance of a portfolio can be improved if the portfolio variance is replaced by the mean absolute value. Indeed, Konno [14] proposed a Mean Absolute Deviation (MAD) based robust portfolio selection method without involving mean vector and covariance matrix explicitly.

Sparsity is also desirable in portfolio selection because it reduces the management cost. However, this cannot be achieved by applying the method of [14] directly. Though the Lasso penalty of Tibshirani [19] is introduced in the context of variable selection, it finds extensive applications in portfolio selection. For example, Brodie [3] develops a sparse and stable portfolio selection strategy by incorporating the idea of Lasso regularization. It is shown that the out-of-sample performance of the Lasso regularized method is consistently better than naive equal-weight portfolio in terms of Sharpe ratio. Further studies of regularized Markowitz's theory include, to name a few, [4, 7, 8, 24, 25]. However, all these methods are developed under the traditional mean-variance framework. The purpose of this paper is to incorporate Lasso penalty into MAD based portfolio selection method.

In this paper, we illustrate that the proposed MAD-Lasso method can be reformulated as a constrained LAD problem with linear equality constraints. In the absence of constraints, Shi [18] develop a steepest descent algorithm for the LAD Lasso problem. In the present paper, we further generalize the ideas of "nonsmooth optimality conditions" and "basis direction set" to allow equality constraints. Interior point method is a competitor of the proposed algorithm. Notice that the constrained LAD problem can be transformed into a linear programming problem and therefore can be solved by interior point method provided in the Matlab interface. However, interior point requires nested iteration that increase the tuning parameter in the outer-loop and do optimization to an approximated problem in the inner-loop. Since the solution is never exact if only finitely-many iterations are done, one needs to specify a thresholding value to determine if a component in the approximated solution equals zero. To simplify the calculation process of MAD-Lasso model, we derived the finite optimality condition theorem as the stopping rule for MAD-Lasso model, which considerably improved the time efficiency of our proposed algorithm. Extensive simulation results indicate that our proposed method can impose equivalent magnitude of number of iterations, and is much more time efficient than interior point method. The MAD-Lasso model with our proposed method performs better when dealing with heavy tailed datasets such as Asymmetric Laplace data. Alternatively, one can consider simplex method, ADMM algorithm, coordinate descent algorithm, and L-BFGS algorithm [1, 5, 16, 22, 23]. However, as mentioned in [22], interior point method possesses better worst-case complexity than simplex method (page 393). Direct application of the coordinate descent algorithm can be problematic for nonsmooth objective function like that in the MAD-Lasso problem, see [23]. Though the MAD-Lasso problem can be transformed into constrained problem with smooth objective function, the extension of the coordinate descent algorithm to the constrained cases is not trivial. L-BFGS algorithm[5, 16] involves approximations to both gradient and Hessian matrix. Therefore, it faces similar difficulties as the coordinate descent algorithm. In this paper, our proposed algorithm is compared with interior point method and ADMM algorithm only.

The paper is organized as follows. In Section 2, we propose the MAD-Lasso portfolio selection method. In Section 3, MAD-Lasso model is reformulated as constrained LAD Lasso model with linear equality constraints. New descent algorithm for the constrained LAD-Lasso problem is presented based on the ideas of zero set and ba-

sis direction set. In Section 4, extensive simulation studies and real data analysis are carried out to evaluate the performance of our methods. Results show that our proposed methodology is significantly much more time efficient than state-of-the-art linear programming solver: interior point method. Concluding remarks and future research directions are discussed in Section 6.

## 2. MAD-Lasso

Suppose that there are $n$ securities, the rate of return of $i$-th stock at time $t$ is $r_{ti}$, denote the observation matrix as $\boldsymbol{R}$ with $(t,i)$-th entry being $r_{ti}, t = 1, 2, \cdots, T, i = 1, 2, \cdots, n$. The portfolio allocation weight vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)' \in \mathbb{R}^n$ satisfies $\sum_{i=1}^n x_i = 1$. Konno [14] proposed the Mean Absolute Deviation (MAD) risk measure, defined as

$$\mathrm{MAD}\,(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T}\Big|\sum_{i=1}^{n}(r_{ti} - r_i)x_i\Big|.$$

Similar to Broadie [2], we penalize MAD portfolio selection model with Lasso penalty, and obtain the MAD-Lasso problem as

$$\begin{aligned} \boldsymbol{x}^{(\lambda)} &= \arg\min_{\boldsymbol{x}} \mathbb{E}\big\|r_0\mathbf{1}_T - \boldsymbol{R}\boldsymbol{x}\big\|_1 + \lambda\|\boldsymbol{x}\|_1 \\ \text{s.t.} \quad & \boldsymbol{x}'\boldsymbol{r} = r_0\,, \boldsymbol{x}'\mathbf{1} = 1, \end{aligned} \tag{1}$$

where $\lambda$ is a tuning parameter controlling the size of penalty. The MAD-Lasso model has the following advantages:

1. It encourages sparsity. With appropriately chosen tuning parameter $\lambda$, some components in the portfolio weight vector $\boldsymbol{x}$ shrink towards zero, resulting in sparse portfolio selection strategies.
2. It controls the shorting level of portfolio selection model. The equivalent formulation is to minimize

$$\|r_0\mathbf{1}_T - \boldsymbol{R}\boldsymbol{x}\| + 2\lambda\sum_{i:x_i\le 0}|x_i| + \lambda,$$

   where $\sum_{i:x_i\le 0}|x_i|$ controls the shorting level. The last term does not affect the optimization problem.
3. It robustify the portfolio selection problem. The $\ell_1$ norm penalty mitigate the computational difficulties related to the possible collinearity in the rates of returns of different assets. Moreover, it ameliorates the influence of financial violations and extreme cases.

**Proposition 2.1.** *We have the followings.*

*(1) For any two tuning parameters $\lambda_1 < \lambda_2$, let $\boldsymbol{x}^{(\lambda_1)}, \boldsymbol{x}^{(\lambda_2)}$ be the corresponding weight vectors. Then, we have*

$$(\lambda_1 - \lambda_2)\big(\|\boldsymbol{x}^{(\lambda_2)}\| - \|\boldsymbol{x}^{(\lambda_1)}\|\big) \ge 0.$$

*This indicates that the greater is the penalty $\lambda$, the greater is the sparsity.*

*(2) Suppose that there exists $\lambda_0$ such that all entries in $\boldsymbol{x}^{(\lambda_0)}$ are non-negative. Then, for any $\lambda \geq \lambda_0$, all entries in the solution $\boldsymbol{x}^{(\lambda)}$ are non-negative too.*

**Proof.** (1) Suppose there are two portfolio allocation vectors $\boldsymbol{x}^{(\lambda_1)}, \boldsymbol{x}^{(\lambda_2)}$ corresponding to the tuning parameter $\lambda_1, \lambda_2$ respectively in the MAD-Lasso problem (1). We have

$$\begin{aligned} & \|r_0\mathbf{1}_T - \boldsymbol{R}\boldsymbol{x}^{(\lambda_1)}\|_1 + \lambda_1\|\boldsymbol{x}^{(\lambda_1)}\|_1 \\ \leq\ & \|r_0\mathbf{1}_T - \boldsymbol{R}\boldsymbol{x}^{(\lambda_2)}\|_1 + \lambda_1\|\boldsymbol{x}^{(\lambda_2)}\|_1 \\ =\ & \|r_0\mathbf{1}_T - \boldsymbol{R}\boldsymbol{x}^{(\lambda_2)}\|_1 + \lambda_2\|\boldsymbol{x}^{(\lambda_2)}\|_1 + (\lambda_1 - \lambda_2)\|\boldsymbol{x}^{(\lambda_2)}\|_1 \\ \leq\ & \|r_0\mathbf{1}_T - \boldsymbol{R}\boldsymbol{x}^{(\lambda_1)}\|_1 + \lambda_2\|\boldsymbol{x}^{(\lambda_1)}\|_1 + (\lambda_1 - \lambda_2)\|\boldsymbol{x}^{(\lambda_2)}\|_1 \\ =\ & \|r_0\mathbf{1}_T - \boldsymbol{R}\boldsymbol{x}^{(\lambda_1)}\|_1 + \lambda_1\|\boldsymbol{x}^{(\lambda_1)}\|_1 + (\lambda_1 - \lambda_2)\big(\|\boldsymbol{x}^{(\lambda_2)}\|_1 - \|\boldsymbol{x}^{(\lambda_1)}\|_1\big). \end{aligned}$$

This yields that

$$(\lambda_1 - \lambda_2)\big(\|\boldsymbol{x}^{(\lambda_2)}\|_1 - \|\boldsymbol{x}^{(\lambda_1)}\|_1\big) \geq 0. \tag{2}$$

(2) If all the entries of $\boldsymbol{x}^{(\lambda_0)}$ are nonnegative and some entries of $\boldsymbol{x}^{(\lambda)}$ are negative, we have $\big\|\boldsymbol{x}^{(\lambda)}\big\| \geq \sum_{i=1}^n \big|x_i^{(\lambda)}\big| = \big|\sum_{i=1}^n x_i^{(\lambda_0)}\big| = \sum_{i=1}^n \big|x_i^{(\lambda_0)}\big| = 1$. This yields that $\big\|\boldsymbol{x}^{(\lambda)}\big\| \geq \big\|\boldsymbol{x}^{(\lambda_0)}\big\|$. From (2), we have $\lambda_0 \geq \lambda$. This indicates that the all-nonnegative-entry case $\lambda_0$ corresponds to the sparsest solution. The particular solution corresponding to $\lambda_0$ is the optimal solution among all solutions for $\lambda \geq \lambda_0$.

## 3. Constrained LAD Lasso model

To solve the MAD-Lasso portfolio selection problem, we generalize the descent algorithm of Shi et al. [18] to allow linear equality constraints.

### *3.1. Problem Definition*

Consider constrained LAD-Lasso problem:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|y - A\boldsymbol{x}\|_1 + \lambda\|\boldsymbol{x}\| \\ \text{s.t.} \quad & C\boldsymbol{x} = b, \end{aligned} \tag{3}$$

where $A \in \mathbb{R}^{T\times n}$ is the covariance matrix with row vectors $A_i$, $C \in \mathbb{R}^{q\times n}, b = (b_1, \cdots, b_q)' \in \mathbb{R}^q$, and $\boldsymbol{x} = (x_1, \cdots, x_n)' \in \mathbb{R}^n$. Without loss of generality, we assume that $C$ is full rank matrix, i.e., $\text{rank}\,(C) = q$.

Problem (1) can be reformulated as special case of Problem (3) with

$$A = \begin{pmatrix} r_{11} - r_1 & r_{12} - r_2 & \cdots & r_{1n} - r_n \\ r_{21} - r_1 & r_{22} - r_2 & \cdots & r_{2n} - r_n \\ \vdots & \vdots & \vdots & \vdots \\ r_{T1} - r_1 & r_{T2} - r_2 & \cdots & r_{Tn} - r_n \end{pmatrix} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_T \end{pmatrix}, \quad y = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

$$C = \begin{pmatrix} r_1 & r_2 & \cdots & r_n \\ 1 & 1 & \cdots & 1 \end{pmatrix}, \quad b = \begin{pmatrix} r_0 \\ 1 \end{pmatrix}.$$

where $A_i$ is the $i$-th row of constraint matrix $A$, $y_i$ is the $i$-th element of $y$, and $C$ is the constraint matrix with row vectors $C_i\,, i = 1, 2, \cdots, q\,, \mathrm{rank}\,(C) = q = 2$.

### *3.2. Optimality Conditions for Feasible Direction*

Note that an arbitrary point $\boldsymbol{x}$ can be transformed to a feasible point as shown below. Suppose that $C\boldsymbol{x} - b = c_0 \neq 0$. Setting $\boldsymbol{x}_0 = \boldsymbol{x} - C^\intercal(CC^\intercal)^{-1}c_0$, then,

$$C\boldsymbol{x}_0 - b = C\boldsymbol{x} - b - CC^\intercal(CC^\intercal)^{-1}c_0 = c_0 - c_0 = 0.$$

The transformed point $\boldsymbol{x}_0$ is said to be the feasible point generated by $\boldsymbol{x}$. Thus, the initial point for the algorithm can be chosen as a feasible point. If $\boldsymbol{x}$ is a feasible point, we choose a direction $h$ such that the cost function value decreases along this direction. The choice of the direction can not be arbitrary because the constraints must be satisfied along this direction. That is, it is required that

$$C(\boldsymbol{x} + h) - b = C\boldsymbol{x} - b + Ch = Ch = 0. \tag{4}$$

**Definition 3.1.** The direction $h$ fulfilling (4) is called a feasible direction. If $h$ is a feasible direction, the corresponding directional derivative is a feasible directional derivative.

First, we have the following assumptions.

**Assumption 1.** *For any* $\boldsymbol{x}$ *and* $h \in \mathbb{R}^n$, $\lim\limits_{\lambda\to\infty} f(\boldsymbol{x} + \lambda h) = \infty$.

**Assumption 2.** *For any* $n$ *indices* $i_1, \cdots, i_n$ *in* $\{1, \cdots, T\}$, $\{A_{i_1}, \cdots, A_{i_n}\}$ *are linearly independent.*

Denote $A_i\boldsymbol{x} - y_i = u_i$ and $\Omega = \{*_1, \cdots, *_m\} = \{*_i : u_{*_i} = 0, i = 1, 2, \cdots, m\}$ as the zero set. Then, the objective function can be rewritten as summation of smooth and nonsmooth part of $f(\boldsymbol{x})$:

$$\begin{aligned}
&f(\boldsymbol{x}) = S(\boldsymbol{x}) + N(\boldsymbol{x}),\\
&S(\boldsymbol{x}) \triangleq \sum_{i=1}^{T} \mathrm{I}\,(u_i > 0)\,(A_i\boldsymbol{x} - y_i) + \sum_{i=1}^{T} \mathrm{I}\,(u_i < 0)\,(-A_i\boldsymbol{x} + y_i) \triangleq c\boldsymbol{x} + z,\\
&N(\boldsymbol{x}) \triangleq \sum_{i=1}^{T} \mathrm{I}\,(u_i = 0)\,|A_i\boldsymbol{x} - y_i| = \sum_{i=1}^{m} |A_{*_i}\boldsymbol{x} - y_{*_i}|,
\end{aligned}$$

where $\mathrm{I}\,(\cdot)$ as the indicator function and

$$c \triangleq \sum_{i=1}^{T} \mathrm{I}\,(u_i > 0)A_i - \sum_{i=1}^{T} \mathrm{I}\,(u_i < 0)A_i\,, z = -\sum_{i=1}^{T} \mathrm{I}\,(u_i > 0)y_i + \sum_{i=1}^{T} \mathrm{I}\,(u_i < 0)y_i.$$

Since $f(\boldsymbol{x})$ is convex, its local minimizer must be the global minimizer. The optimality condition of the minimizer is that any feasible directional derivatives are greater than

or equal to zero. That is, $\boldsymbol{x}^*$ is the optimal solution of (3) if and only if

$$\nabla_h f(\boldsymbol{x}^*) = \nabla_h S(\boldsymbol{x}^*) + \nabla_h N(\boldsymbol{x}^*) \geq 0, \quad \forall h \in \{h \mid h \in \mathbb{R}^n, Ch = 0\}. \tag{5}$$

However, it is not easy to verify the optimality condition (5) because there are infinitely-many feasible directions $h$. To obtain a finite representation of the optimality conditions, consider the nonsmooth part $N(\boldsymbol{x})$ with

$$A_{*_i}\boldsymbol{x} = y_{*_i}, \quad i = 1, \cdots, m.$$

If $\{A_{*_i} : i = 1, \cdots, m\}$ are independent, then $m \leq n - q$. If $m > n - q$, then the equations above are overdetermined. Denote $C_i$ as the $i$-th row of constraint matrix $C, i = 1, 2, \cdots, q$. Without loss of generality, we assume that $m \leq n - q$ and $\{A_{*_i} : i = 1, \cdots, m\} \cup \{C_1, C_2, \cdots, C_q\}$ are linearly independent. Let

$$D = \begin{pmatrix} C_1 \\ \vdots \\ C_q \\ A_{*_1} \\ \vdots \\ A_{*_m} \end{pmatrix}.$$

Generalized inverse matrix $V_D$ can be obtained such that $DV_D = I_{m+q}$, where $I_{m+q}$ is the $(m + q) \times (m + q)$ identity matrix and $V_D = (V_1, \cdots, V_{m+q})$. Consider the null space $\{V \in \mathbb{R}^n | DV = 0\}$. There exist $n - m - q$ linearly independent vectors $V_j, j = m + q + 1, \cdots, n$ that form the basis of the null space. Hence, we have $DV_j = 0, \forall j = m + q + 1, \cdots, n$. Then, an equivalent finite-representation of the optimality condition (5) is given by the following theorem.

**Theorem 3.2.** *Suppose that rank$(D) = m + q$, then $\boldsymbol{x}^*$ is the optimal solution if and only if the feasible directional derivatives satisfy*

$$\begin{aligned} &\nabla_{V_i} f(\boldsymbol{x}^*) = \nabla_{V_i} S(\boldsymbol{x}^*) + \nabla_{V_i} N(\boldsymbol{x}^*) \geq 0, && i = q + 1, \cdots, q + m, \\ &\nabla_{V_i^-} f(\boldsymbol{x}^*) = \nabla_{V_i^-} S(\boldsymbol{x}^*) + \nabla_{V_i^-} N(\boldsymbol{x}^*) \geq 0, && i = q + 1, \cdots, q + m, \\ &\nabla_{V_i} f(\boldsymbol{x}^*) = \nabla_{V_i} S(\boldsymbol{x}^*) = 0, && i = m + q + 1, \cdots, n. \end{aligned} \tag{6}$$

**Proof.** Note that the space of all feasible directions is spanned by $\{V_i, i = q+1, \cdots, n\}$, condition (6) is a special case of (5) and the necessary condition is obvious. Next, we establish the sufficient condition, this means that if (6) are satisfied, (5) holds. If $\boldsymbol{x}^*$ is optimal, then we have the following KKT conditions:

$$\begin{aligned} &\nabla_{V_i} f(\boldsymbol{x}^*) \geq 0, \nabla_{V_i^-} f(\boldsymbol{x}^*) \geq 0, \quad i = q + 1, \cdots, q + m, \\ &\nabla_{V_i} f(\boldsymbol{x}^*) = 0, \quad i = q + m + 1, \cdots, n. \end{aligned}$$

By orthonormality of $\{V_1, V_2, \cdots, V_n\}$, (6) can be simplified as

$$\nabla_{V_i} f(\boldsymbol{x}^*) = \nabla_{V_i} S(\boldsymbol{x}^*) = \frac{cV_i}{\|V_i\|} = 0, \quad i = q+m+1, \cdots, n,$$
$$\nabla_{V_i} f(\boldsymbol{x}^*) = \nabla_{V_i} S(\boldsymbol{x}^*) + \nabla_{V_i} N(\boldsymbol{x}^*) = \frac{cV_i}{\|V_i\|} + \frac{1}{\|V_i\|} \geq 0, \quad i = q+1, \cdots, q+m,$$
$$\nabla_{V_i^-} f(\boldsymbol{x}^*) = \nabla_{V_i^-} S(\boldsymbol{x}^*) + \nabla_{V_i^-} N(\boldsymbol{x}^*) = \frac{-cV_i}{\|V_i\|} + \frac{1}{\|V_i\|} \geq 0, \quad i = q+1, \cdots, q+m.$$

For any feasible direction $h$, there exists a weight vector $w = (w_{q+1}, w_{q+2}, \cdots, w_n)'$ such that

$$h = \sum_{i=q+1}^{n_1} w_i V_i + \sum_{i=n_1+1}^{n} w_i(-V_i).$$

Without loss of generality, we can set $w_i \geq 0, \forall i = 1, \cdots, n$. This is because when $w_i < 0$, we have $w_i V_i = (-w_i) \cdot (-V_i)$. Then, replacing $V_i$ by $-V_i$ and $w_i$ by $-w_i > 0$ yield that

$$\begin{aligned} \nabla_h N(\boldsymbol{x}^*) &= \frac{\sum_{i=1}^m |A_{*_i} h|}{\|h\|} = \frac{\sum_{i=1}^m \left| A_{*_i} \left( \sum_{j=q+1}^{n_1} w_j V_j + \sum_{j=n_1+1}^{n} w_j(-V_j) \right) \right|}{\|h\|} \\ &= \frac{\sum_{i=1}^m \left| w_i A_{*_i} V_{i+n} \right|}{\|h\|} = \frac{\sum_{i=1}^m w_i}{\|h\|}. \end{aligned}$$

We have

$$\begin{aligned} \nabla_h f(\boldsymbol{x}^*) &= \frac{\sum_{i=q+1}^n w_i c V_i}{\|h\|} + \frac{\sum_{i=1}^m w_i}{\|h\|} = \frac{1}{\|h\|} \Big( \sum_{i=1}^m w_{i+q}(cV_{i+q} + 1) + \sum_{i=m+q+1}^n w_i c V_i \Big) \\ &= \frac{1}{\|h\|} \sum_{i=1}^m w_{i+q} \nabla_{V_{i+q}} f(\boldsymbol{x}^*) \cdot \|V_{i+q}\| + \frac{1}{\|h\|} \sum_{i=m+q+1}^n w_i \nabla_{V_i} f(\boldsymbol{x}^*) \|V_i\| \geq 0. \end{aligned}$$

Then, for any feasible direction $h$, the feasible directional derivative is greater than or equals to zero. Hence, (5) holds and $\boldsymbol{x}^*$ is the optimal solution. □

### *3.3. Descent feasible directions*

The design of the algorithm is as follows. Let $\boldsymbol{x}^{(k)}$ be the approximation at the $k$-th iteration, $\Omega_k = \{k_1, k_2, \cdots, k_m\}$ as the index set of zero set. If the optimality condition (6) is satisfied, then $\boldsymbol{x}^{(k)}$ is the optimal solution. Otherwise, there exists at least a feasible direction $h$ such that the cost function decreases along this direction. These steps are repeated until (6) is satisfied.

$$\nabla_{V_i} f(\boldsymbol{x}) \geq 0, \text{ and } \nabla_{V_i^-} f(\boldsymbol{x}) \geq 0, \text{ for } i = k_1, k_2, \cdots, k_m,$$

can not be satisfied at the same time and consequently at least one of $V_i$ and $V_i^-$ is the descent direction. For an iterative point $\boldsymbol{x}^{(k)}$, with zero set $\Omega_k$, the cost function

can be rewritten as

$$f(\boldsymbol{x}^{(k)}) = c^{(k)}\boldsymbol{x}^{(k)} + \sum_{i\in\Omega_k} |A_i\boldsymbol{x}^{(k)} - y_i| + z^{(k)}. \tag{7}$$

Denote by $\Omega_k'$ the set of all the indexes such that (6) is not satisfied for $V_i$ or $V_i^-, i = k_1, k_2, \cdots, k_m$. To speed up the search, consider the indexes $i \in \Omega_k$ so that the descent directional derivatives $\nabla_{V_i} f$ or $\nabla_{V_i^-} f$ are the greatest. Suppose that $\Lambda_1 \subset \Omega_k'$ contains a proportion $\alpha$ of the indexes in $\Omega_k'$ with corresponding slowest descent directional derivatives $\nabla_{V_i} f$ or $\nabla_{V_i^-} f$. Extensive experiments show that $\alpha = 0.05$ can make a trade off between stability and convergence. Smaller $\alpha$ may fail to converge, larger $\alpha$ may decrease the convergence rate. In the following, we choose $\alpha = 0.05$. Denote

$$\Omega_k^0 = \Omega_k \backslash \Lambda_1^k.$$

Without loss of generality, assume that $k_1, k_2, \cdots, k_l \in \Omega_k^0$. This means that the indexes in $\Lambda_1^k$ are removed from the zero set. Denote

$$A_{0k} = \begin{pmatrix} A_{k_1} \\ \vdots \\ A_{k_l} \end{pmatrix}, k_1, \cdots, k_l \in \Omega_k^0.$$

Choose the descent direction $h$ in the space spanned by

$$\{V_i : i \in \Lambda_1^k\} \cup \{V_i : i = k_m + 1, \cdots, n\}$$

such that

$$h = \sum_{i\in\Lambda_1^k} t_i V_i + \sum_{i=k_m+1}^{n} t_i V_i.$$

It can be verified that

$$A_i h = 0, \quad \forall i \in \Omega_k^0.$$

Such a choice guarantees that the descent direction keep the set $\Omega_k^0$ unchanged.
Set the descent direction $h^{(k)}$ as the optimal solution to

$$\begin{aligned} \max_{v\in\mathbb{R}^p} \quad & -c^{(k)}h \\ \text{s.t.} \quad & A_i h = 0, \quad \forall i \in \Omega_k^0. \end{aligned} \tag{8}$$

It means that the solution $h$ is chosen as the vector nearest to the deepest descent direction $-c^{(k)}$. The optimal solution to Problem (8) is

$$\tilde{h} = -c^{(k)} - A_{0k}^\intercal (A_{0k} A_{0k}^\intercal)^{-1} A_{0k} \cdot (-c^{(k)}), \tag{9}$$

where $A_{0k}^\intercal (A_{0k} A_{0k}^\intercal)^{-1} A_{0k}(-c^{(k)})$ is the projected direction of $-c^{(k)}$ in the subspace $\{h : A_i h = 0, i \in \Omega_k^0\}$. Equivalently, $A_{0k}^\intercal (A_{0k} A_{0k}^\intercal)^{-1} A_{0k}(-c^{(k)}) = \text{Proj}\{-c^{(k)} \big| h :$

$A_i h = 0, i \in \Omega_k^0\}$. Then, the descent direction $h^{(k)}$ can be chosen as the normalized vector of $\tilde{h}$ with

$$h^{(k)} = \tilde{h}/\|\tilde{h}\|. \tag{10}$$

The normalized direction $h^{(k)}$ guarantees that $\gamma^{(k)}$ is the step length as defined in the following, and the zero set is updated as $\Omega_k = \Omega_k^0$.

### *3.4. Optimal Step Length*

The cost function decreases along the descent direction $h^{(k)}$. The next iteration point is generated by

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \gamma^{(k)} h^{(k)}, \quad \gamma^{(k)} > 0,$$

where $\gamma^{(k)}$ is the step length that is determined via the following optimization problem,

$$\min_{\gamma \geq 0} g(\gamma) = f(\boldsymbol{x}^{(k+1)}) = f(\boldsymbol{x}^{(k)} + \gamma h^{(k)}), \gamma \geq 0.$$

Since $f$ is convex, $g(\gamma)$ is also convex. Then, we can choose $\gamma^{(k)}$ as the optimal solution of the problem $\min_{\gamma} g(\gamma)$. This problem is equivalent to the following problem

$$\begin{aligned} \max_{\gamma \geq 0} \quad & \gamma \\ \text{s.t.} \quad & \nabla_{h^{(k)}} f(\boldsymbol{x}^{(k)} + \gamma h^{(k)}) \geq 0, \\ & \nabla_{h^{(k)-}} f(\boldsymbol{x}^{(k)} + \gamma h^{(k)}) \geq 0. \end{aligned} \tag{11}$$

For this problem, we have the following observation (see Shi et al. [18]).

**Lemma 3.3.** *There exists an optimal solution $\gamma^{(k)} > 0$ and at least one $i$ in $\{1, \cdots, n\}$ such that $A_i(\boldsymbol{x}^{(k)} + \gamma^{(k)} h^{(k)}) = y_i$, that is, $i$ is in the zero set at the point $\boldsymbol{x}^{(k)} + \gamma^{(k)} h^{(k)}$ during the $k$-th iteration.*

### *3.5. Algorithm*

Denote by $\gamma^{(k)}$ the optimum step length along the direction $h^{(k)}$ as described in Section 3.4. The cost function is updated as

$$f(\boldsymbol{x}^{(k+1)}) = c^{(k+1)} \boldsymbol{x}^{(k+1)} + \sum_{i \in \Omega_{k+1}} |A_i \boldsymbol{x}^{(k+1)}| + z^{(k+1)}.$$

Remove the indexes in $\Lambda_1^k$ from the zero set $\Omega_k$ and denote $\Omega_k^0 = \Omega_k \backslash \Lambda_1^k$. Let $\Lambda_2^k = \{i | u_i = 0\}$. Lemma 3.3 guarantees that $\Lambda_2^k$ is non-empty. In the $(k+1)$-th iteration, set

$$\Omega_{k+1} = \Omega_k^0 \cup \Lambda_2^k,$$

and

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \gamma^{(k)} h^{(k)}.$$

Continue the above process until the optimal condition (6) is satisfied.

To summarize, the algorithm is as follows:

**Algorithm 1** Descent Algorithm for MAD-Lasso model

- **Initialization:** Choose an initial point $\boldsymbol{x}^{(0)}$, compute the corresponding set $\Omega_0$, and compute the cost function $f(\boldsymbol{x}^{(0)})$. Set $k = 0$.
- **Step 1: (Terminate)**
  Generate the matrix $V$ for the zero set $\Omega_k$. If conditions (6) are satisfied within tolerance level $1e^{-6}$, then stop and return the optimal solution and value. Otherwise, go to Step 2.
- **Step 2: (Descent Direction)**
  Find the $\alpha = 0.05$ proportion fastest descent directions as $\Lambda_1^k$, where $\alpha$ denotes the percentage of selected descent directions that decrease faster than the other $1-\alpha$ directions. Set $\Omega_k^0 = \Omega_k \backslash \Lambda_1^k$, and compute the descent direction $h^{(k)}$ using (9), (10).
- **Step 3: (Optimal Step Length)** Find the best step length $\gamma^{(k)}$ by (11).
- **Step 4: (Iteration)** Update $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \gamma^{(k)} h^{(k)}$. Find $\Lambda_2^k$ and update the zero set as $\Omega_{k+1} = \Omega_k^0 \cup \Lambda_2^k$. Then we compute the cost function $f(\boldsymbol{x}^{(k+1)})$ at $(k+1)$-th iteration, and then we go to Step 1.

## 4. Numerical Experiments

In this section, several simulation studies are conducted to confirm the time efficiency and robustness of our proposed algorithm. The proposed algorithm is compared with both interior point method and ADMM algorithm. See Appendix A for the implementation of these two methods. For MAD-Lasso model, Example 1 and Example 2 show that our proposed method is significantly time efficient than interior point method and ADMM. For portfolio selection, Example 3 show that MAD-Lasso models (IP,NEW) outperform MAD model and is more robust for heavy tailed data. All the numerical experiments are performed in Matlab with an Intel (R) Core (TM) i7-4790 3.60 GHz Processor and 3.60 GHz memory. Interior point method and ADMM are implemented in the MATLAB interface (see Appendix A).

### *4.1. Simulation Studies*

**Example 1:** In the first example, we compare interior point method and our proposed method for MAD-Lasso portfolio selection models. We generate data from the following parameter settings:

$$\begin{aligned}
\text{Case (I)}: \quad & \boldsymbol{\mu} = (0.3, 0.6, 0.00001, 0.00001, 0.00001)\,, \\
& \Sigma = \mathrm{diag}(\boldsymbol{\mu}/10)\,, n = 5\,, T = 100, 200, 300, 400, 500; \\
\text{Case (II)}: \quad & \boldsymbol{\mu} = (0.2, 0.4, 0.6, 0.8, 1.0, 0.00001, 0.00001, 0.00001, 0.00001, 0.00001)\,, \\
& \Sigma = \mathrm{diag}(\boldsymbol{\mu}/10)\,, n = 10\,, T = 200, 300, 400, 500.
\end{aligned}$$

For each case, we set tuning parameter $\lambda = \sqrt{2T \log n}$ as suggested in [21]. We generate 100 replicates and averaged the estimation results to obtain final results. Then we implement interior point method, our proposed method and ADMM to evaluate the estimation performance of MAD-Lasso models. We focus on time consumption (Time), number of estimated nonzero coefficients (Degree of Freedom) and number of iterations (Iter No.). Simulation results are reported in Table 1. Figure 1-2 plot the tendency of Time and Iter No. with respect to sample size $T$.

**Table 1.** Simulation results of Example 1, Case (I) and Case (II): IP: interior point method; Proposed: our proposed method; ADMM: ADMM algorithm.

| **Case (I)** | | Time | | | Degree of Freedom | | | Number of Iterations | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $T$ | IP | Proposed | ADMM | IP | Proposed | ADMM | IP | Proposed | ADMM |
| 5 | 100 | 0.0109 | 0.0018 | 1.9321 | 3.15 | 3.10 | 3.20 | 9.40 | 4.15 | 907.75 |
| 5 | 200 | 0.0212 | 0.0022 | 10.6599 | 3.20 | 3.20 | 3.30 | 9.60 | 4.50 | 925.05 |
| 5 | 300 | 0.0312 | 0.0028 | 30.1605 | 3.15 | 3.15 | 3.05 | 9.80 | 4.60 | 964.35 |
| 5 | 400 | 0.0397 | 0.0032 | 55.5970 | 3.25 | 3.25 | 3.15 | 9.90 | 4.80 | 970.80 |
| 5 | 500 | 0.0482 | 0.0035 | 94.3329 | 3.25 | 3.30 | 3.30 | 10.00 | 4.89 | 980.58 |
| **Case (II)** | | Time | | | Degree of Freedom | | | Number of Iterations | | |
| $n$ | $T$ | IP | Proposed | ADMM | IP | Proposed | ADMM | IP | Proposed | ADMM |
| 10 | 200 | 0.0301 | 0.0081 | 16.4816 | 6.30 | 6.10 | 6.30 | 10.00 | 12.90 | 1337.80 |
| 10 | 300 | 0.0363 | 0.0098 | 46.6691 | 6.20 | 6.20 | 6.00 | 10.20 | 13.20 | 1543.40 |
| 10 | 400 | 0.0439 | 0.0103 | 92.1499 | 6.00 | 6.00 | 6.10 | 10.40 | 15.90 | 1586.30 |
| 10 | 500 | 0.0541 | 0.0140 | 153.5235 | 6.10 | 6.00 | 6.10 | 10.50 | 16.30 | 1613.50 |

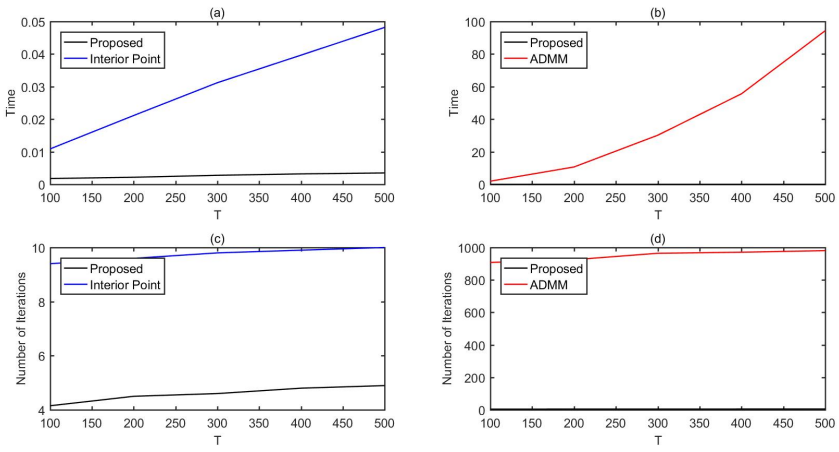


**Figure 1.** Tendency of Time and Number of Iterations w.r.t. sample size $T$ for Case (I). IP: interior point method; Proposed: our proposed method; ADMM: ADMM algorithm.

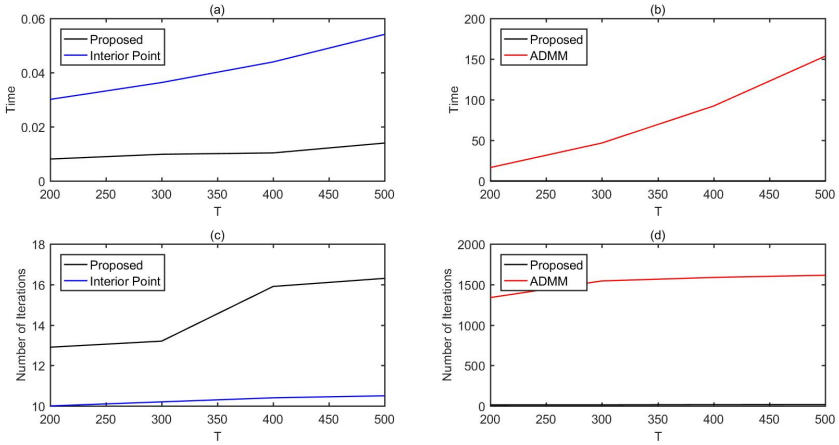


**Figure 2.** Tendency of Time and Number of Iterations tendency w.r.t. sample size $T$ for Case (II). IP: interior point method; Proposed: our proposed method; ADMM: ADMM algorithm.

Table 1 show that ADMM is quite time consuming, we then focus on comparison of interior point method and our proposed method in Example 2. With increasing sample size $T$, number of iterations (Iter No.) increases, the Degree of Freedom results show that the three methods can achieve equivalent portfolio selection results.

Figure 1-2 plot the tendency of time consumption (Time) and number of iterations (Iter No.) for interior point method, our proposed method and ADMM algorithm. Extensive simulation studies confirmed that our proposed method is more time efficient and impose equivalent magnitude number of iterations with respect to interior point method.

**Example 2.** Since Example 1 confirmed the fact that ADMM is time inefficient, we compare our proposed method and interior point method for MAD-Lasso model with $n = 20$ and $n = 50$. Simulation results of time consumption (Time), number of iterations (Iter No.) and Degree of Freedom are reported in Table 2, tendency of time consumption and number of iterations w.r.t. sample size $T$ are depicted in Figure 3.

**Table 2.** Simulation results of time consumption, number of iterations, degree of freedom for $n = 20$ and $n = 50$. IP: interior point method, Proposed: our proposed algorithm.

| $\boldsymbol{n = 20}$ | | Time | | Iter No. | | Degree of Freedom | |
|---|---|---|---|---|---|---|---|
| $n$ | $T$ | IP | Proposed | IP | Proposed | IP | Proposed |
| 20 | 400 | 0.0579 | 0.0215 | 10.60 | 31.24 | 7.02 | 7.01 |
| 20 | 600 | 0.0781 | 0.0275 | 10.80 | 36.36 | 7.03 | 7.02 |
| 20 | 800 | 0.0937 | 0.0333 | 11.10 | 41.28 | 7.01 | 7.03 |
| 20 | 1000 | 0.1365 | 0.0445 | 11.60 | 44.16 | 7.04 | 7.02 |
| $\boldsymbol{n = 50}$ | | Time | | Iter No. | | Degree of Freedom | |
| $n$ | $T$ | IP | Proposed | IP | Proposed | IP | Proposed |
| 50 | 500 | 0.1101 | 0.0754 | 11.84 | 87.13 | 8.77 | 8.65 |
| 50 | 600 | 0.1342 | 0.0931 | 11.96 | 97.25 | 8.88 | 8.86 |
| 50 | 700 | 0.1646 | 0.1076 | 11.99 | 103.61 | 9.55 | 9.43 |
| 50 | 800 | 0.1978 | 0.1294 | 12.02 | 109.54 | 9.88 | 9.86 |
| 50 | 900 | 0.2308 | 0.1352 | 12.16 | 103.62 | 10.02 | 10.04 |
| 50 | 1000 | 0.2988 | 0.1843 | 12.34 | 109.45 | 10.11 | 10.08 |

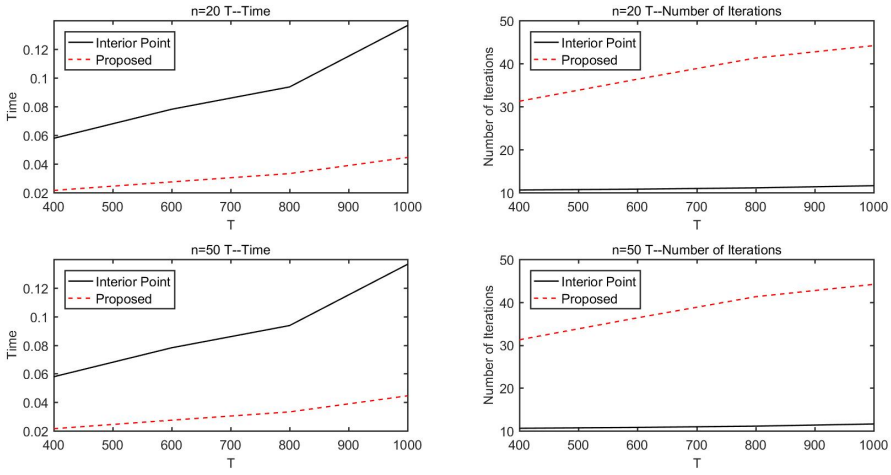


**Figure 3.** Tendency of time consumption and number of iterations with $n = 20$ and $n = 50$. IP: interior point method; Proposed: our proposed algorithm.

Table 2 show that though our proposed method require larger number of iterations than interior point method, it is significantly more time efficient than interior point method. This extensively verifies the time efficiency of our proposed method for MAD-Lasso model. Figure 3 also verifies the time efficiency of our algorithm and the equivalent magnitude of number of iterations with respect to interior point method.

**Example 3.** To explore the portfolio selection performance of our proposed method and ADMM, we compare portfolio selection performance of Gaussian data and Asymmetric Laplace (AL) data (see Kotz [15]). The Gaussian data and AL data are generated from the following parameter settings:

$$\boldsymbol{\mu} = (0.20, 0.40, 0.60, 0.80, 1.0, \underbrace{0.00001, \cdots, 0.00001}_{p-5}), \quad \Sigma = \mathrm{diag}(\boldsymbol{\mu}/10),$$

with $(n, T) = (10, 200), (20, 500), (50, 1000)$. Simulation results are reported in Table 3, the corresponding boxplots are illustrated in Figure 4-6.

**Table 3.** Simulation results of Gauss Data and AL Data for Setting I,II,III.

| **Setting I** | | | Gaussian Data | | | AL Data | | |
|---|---|---|---|---|---|---|---|---|
| Model | $n$ | $T$ | Time | DF | Iter No. | Time | DF | Iter No. |
| MAD Model | 10 | 200 | 0.0360 | 10.00 | 7.22 | 0.0361 | 10.00 | 7.27 |
| MAD-Lasso(IP) | 10 | 200 | 0.0394 | 6.15 | 10.61 | 0.0409 | 5.89 | 11.15 |
| MAD-Lasso(NEW) | 10 | 200 | 0.0072 | 6.13 | 14.71 | 0.0066 | 5.89 | 13.81 |
| **Setting II** | | | Gaussian Data | | | AL Data | | |
| Model | $n$ | $T$ | Time | DF | Iter No. | Time | DF | Iter No. |
| MAD Model | 20 | 500 | 0.0608 | 20.00 | 8.37 | 0.0605 | 20.00 | 8.49 |
| MAD-Lasso(IP) | 20 | 500 | 0.0665 | 6.81 | 11.39 | 0.0685 | 6.41 | 11.76 |
| MAD-Lasso(NEW) | 20 | 500 | 0.0220 | 6.79 | 34.53 | 0.0179 | 6.39 | 30.12 |
| **Setting III** | | | Gaussian Data | | | AL Data | | |
| Model | $n$ | $T$ | Time | DF | Iter No. | Time | DF | Iter No. |
| MAD Model | 50 | 1000 | 0.2103 | 49.99 | 9.41 | 0.2082 | 50.00 | 9.29 |
| MAD-Lasso(IP) | 50 | 1000 | 0.2667 | 9.68 | 12.07 | 0.2645 | 8.23 | 11.86 |
| MAD-Lasso(NEW) | 50 | 1000 | 0.1638 | 9.60 | 129.50 | 0.1180 | 8.17 | 94.72 |

Table 3 show that our proposed algorithm is more time efficient than interior point for both Gaussian Data and Asymmetric Laplace (AL) Data. For AL Data, the Degree of Freedom is much more close to 5, i.e. the number of true significant variables $(X_1, \cdots, X_5)$ than for Gaussian data. The special connection between MAD risk measure and Asymmetric Laplace distribution yields that the MAD-Lasso model is more stable for heavy tailed data such as AL Data. For MAD-Lasso models with varying dimension scales, number of iterations of our proposed method is greater than interior point method; However, Time index show that our proposed method is time efficient. Thus our proposed method performs better than interior point method overall.

Figure 4-6 display the time consumption (Time), Degree of Freedom, number of iterations (Iter No.) for Gaussian Data and AL Data with varying sample sizes. MAD-Lasso model with our proposed method outperforms other methods, especially for AL data. Simulation results of our proposed method are more robust against AL data and time efficient.

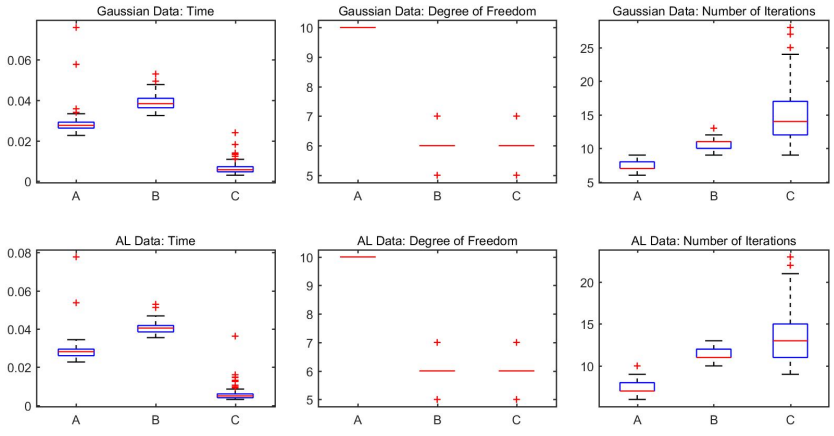


**Figure 4.** Boxplots of Gaussian Data and AL Data with $n = 10, T = 200$. A: MAD model; B: MAD-Lasso model with interior point method; C: MAD-Lasso Model with proposed method.

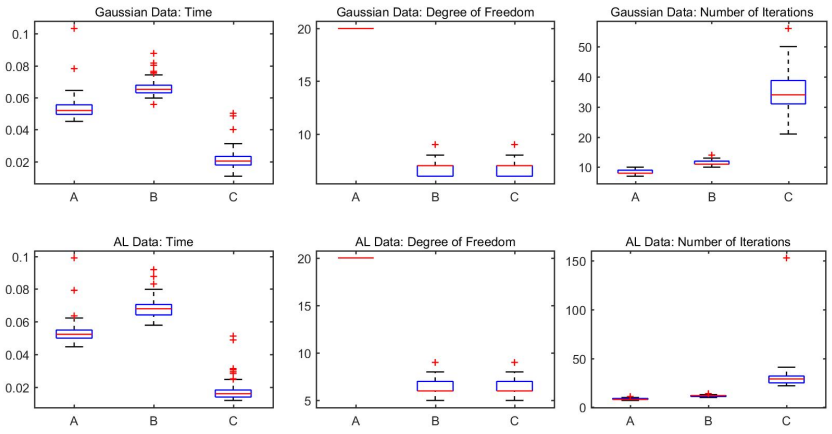


**Figure 5.** Boxplots of Gaussian Data and AL Data with $n = 20, T = 500$. A: MAD model; B: MAD-Lasso model with interior point method; C: MAD-Lasso Model with proposed method.

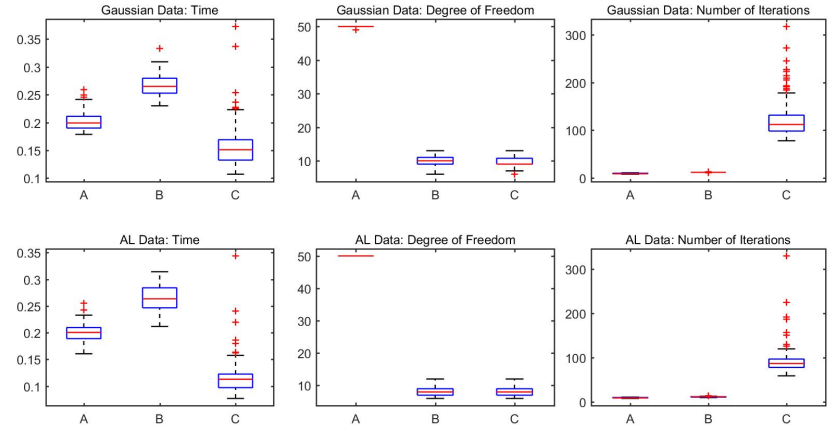


**Figure 6.** Boxplots of Gaussian Data and AL Data with $n = 50, T = 1000$. A: MAD model; B: MAD-Lasso model with interior point method; C: MAD-Lasso Model with proposed method.

## 5. Portfolio Selection Applications

To evaluate portfolio selection performance of MAD model and MAD-Lasso models, extensive simulation studies and real data analysis are carried out to compare (1) MAD, (2) MAD-Lasso with the proposed algorithm, and (3) MAD-Lasso with interior point (IP) method, see Appendix A for description of interior point method. The comparison is based on computational efficiency and performance of the portfolio selection under the following risk measures:
(1) Expected Return (Mean): Mean $= \boldsymbol{w}'\boldsymbol{\mu}$.
(2) Sharpe Ratio: Sharpe $= \boldsymbol{w}'\boldsymbol{\mu}/\sqrt{\boldsymbol{w}'\boldsymbol{\Sigma}\boldsymbol{w}}$.
(3) Sparsity: number of nonzero entries of $\boldsymbol{w}$ (see [26]).
(4) Time: time consumption.
(5) Standard Deviation (StD): $\sigma = \sqrt{\boldsymbol{w}'\boldsymbol{\Sigma}\boldsymbol{w}}$.
(6) Mean Absolute Deviation (MAD): $\frac{1}{T}\sum_{t=1}^{T}\left|\sum_{i=1}^{n}(r_{ti} - r_i)w_i\right|$.

(7) Value at Risk (VaR): $\text{VaR}_\alpha = \mu + \sigma\Phi^{-1}(1-\alpha)$.
(8) Expected Shortfall (ES): $\text{ES}_\alpha = \mu + \sigma\frac{\psi(\Phi^{-1}(1-\alpha))}{\alpha}$.

### *5.1. Simulation Study*

**Example 4:** To investigate the estimation performance of different portfolio selection methods, we consider two cases that the datasets are generated from multivariate Gaussian distribution and multivariate Asymmetric Laplace distribution with the following parameter settings:

$$\boldsymbol{\mu} = (0.00001\,, 0.00002\,, 0.00003\,, 0.00004\,, 0.00005\,, 0.60\,, 0.70\,, 0.80\,, 0.90\,, 1.00).$$

$$\boldsymbol{\Sigma} = \text{diag}\,(\boldsymbol{\mu}/10).$$

In both cases, (1) MAD, (2) MAD-Lasso with interior point method (IP, see Appendix A), and (3) MAD-Lasso with the proposed method (NEW) as described in Section 3 are used for portfolio selection. The tuning parameter $\lambda$ is chosen as $\lambda = \sqrt{2T\log n}$, as suggested in [21]. We conduct 100 replicates for both Gaussian Data and Asymmetric Laplace Data cases. Portfolio selection simulation results are reported in Table 4. Figure 7-8 further display the boxplots of Gaussian Data and AL Data, respectively.

**Table 4.** Simulation results of Gaussian Data and AL Data for portfolio selection.

| **Gaussian Data** | $n$ | $T$ | Time | Sparsity | Iter No.. | Mean | StD | Sharpe | VaR | CVaR |
|---|---|---|---|---|---|---|---|---|---|---|
| MAD Model | 10 | 100 | 0.0178 | 10.00 | 6.66 | 0.30 | 0.17 | 1.80 | 0.58 | 0.40 |
| MAD-Lasso(IP) | 10 | 100 | 0.0129 | 6.09 | 10.10 | 0.30 | 0.17 | 1.77 | 0.58 | 0.40 |
| MAD-Lasso(NEW) | 10 | 100 | 0.0058 | 6.08 | 13.34 | 0.30 | 0.17 | 1.77 | 0.58 | 0.40 |
| **AL Data** | $n$ | $T$ | Time | Sparsity | Iter. No. | Mean | StD | Sharpe | VaR | CVaR |
| MAD Model | 10 | 100 | 0.0175 | 10.00 | 6.75 | 0.30 | 0.33 | 0.91 | 0.85 | 0.59 |
| MAD-Lasso(IP) | 10 | 100 | 0.0127 | 5.71 | 10.83 | 0.30 | 0.34 | 0.89 | 0.86 | 0.60 |
| MAD-Lasso(NEW) | 10 | 100 | 0.0055 | 5.69 | 12.70 | 0.30 | 0.34 | 0.89 | 0.86 | 0.60 |

Table 4 confirmed that the MAD-Lasso models outperform MAD model in terms of Sharpe ratio, Sparsity and risk measures (StD, MAD, $\text{VaR}_{0.01}$, $\text{CVaR}_{0.01}$). Under the above-mentioned indicators, the performance of MAD-Lasso is similar for interior point method and the proposed descent algorithm. However, in terms of computational time, the proposed descent algorithm significantly outperforms interior point method. Figure 7-8 further plot the boxplots of the simulation results.

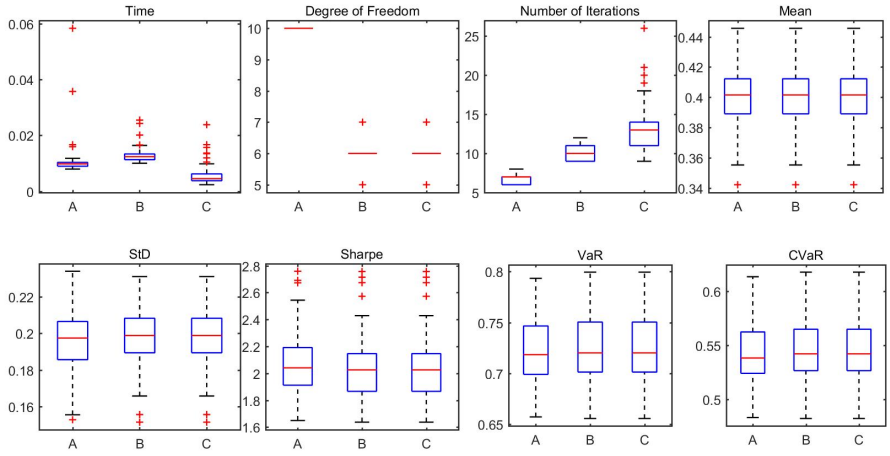


**Figure 7.** Simulation results of Gaussian Data for portfolio selection: A. MAD model; B. MAD-Lasso with interior point method (IP); C. MAD-Lasso with proposed method (NEW).

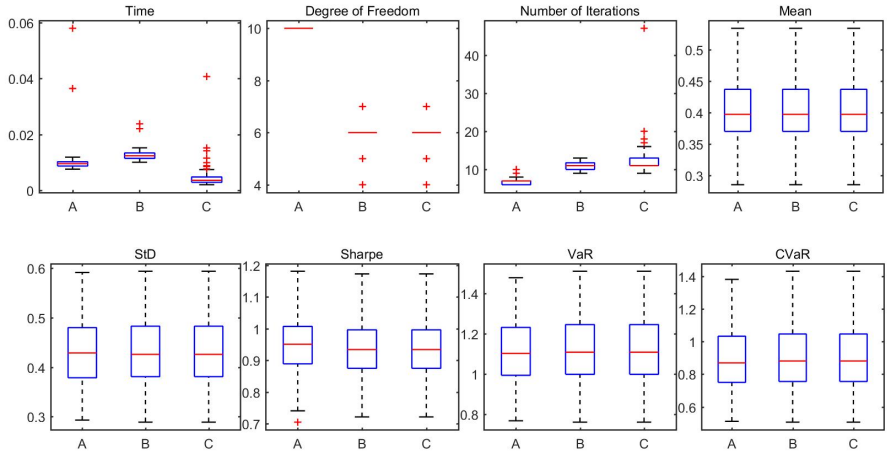


**Figure 8.** Simulation results of AL Data for portfolio selection: A. MAD model; B. MAD-Lasso with interior point method (IP); C. MAD-Lasso with proposed method (NEW).

### *5.2. Real Data Analysis*

Consider the datasets complied by Fama and French. Portfolios involving 48 industry sectors are obtained from both daily and monthly data (abbreviated to FF48d, FF48m) from June 1976 to June 2006. In both FF48d and FF48m datasets, the portfolios are constructed at the end of each June. Denote by $r_{ti}$ the annualized return in time $t$ of $i$-th industry, $i = 1, 2, \cdots, 48$.

**Example 1.** In this example, we compare the out-of-sample performances of MAD model (mad), Naive Evenly model (naive), and MAD-Lasso models with interior point and our proposed method (IP, NEW). For such a purpose, all portfolios are constructed by fixing the expected return at $r_0 = \bar{\boldsymbol{r}}$, where the target return $r_0$ as the average return achieved by the naive, evenly-weighted portfolio, computed from either the entire daily

data or the entire monthly data. Consider the sequence of increasing tuning parameters $\lambda = 2^{-5:1:5}\sqrt{2T\log n}$ with $\lambda_1 = 2^{-5}\sqrt{2T\log n} = \frac{1}{32}\sqrt{2T\log n}\,, \lambda_2 = 2^{-4}\sqrt{2T\log n} = \frac{1}{16}\sqrt{2T\log n}\,, \cdots, \lambda_{10} = 2^5\sqrt{2T\log n} = 32\sqrt{2T\log n}$.

For both FF48d and FF48m datasets, we compare MAD model (mad), Naive evenly-weighted model (naive), and MAD-Lasso (IP, NEW). The comparisons are based on computational time, Sparsity, Sharpe, MAD, $\text{VaR}_{0.01}$ and $\text{ES}_{0.01}$. Estimation results are reported in Table 5, 6.

**Table 5.** Portfolio selection results of FF48d data:$(T, n) = (7573, 48)$; ExpRet: $r_0 = 0.0550$.

| | Time | | Sharpe | | Sparsity | |
|---|---|---|---|---|---|---|
| | mad 3.8550 | naive 0.0064 | mad 0.0841 | naive 0.0632 | mad 47 | naive 48 |
| | MAD-Lasso (IP) | MAD-Lasso (NEW) | MAD-Lasso (IP) | MAD-Lasso (NEW) | MAD-Lasso (IP) | MAD-Lasso (NEW) |
| $\lambda_1$ | 2.9822 | 2.3658 | 0.0852 | 0.0852 | 47 | 47 |
| $\lambda_2$ | 3.4228 | 2.0967 | 0.0852 | 0.0852 | 47 | 47 |
| $\lambda_3$ | 3.3466 | 2.0494 | 0.0851 | 0.0851 | 45 | 45 |
| $\lambda_4$ | 3.4036 | 2.1634 | 0.0848 | 0.0848 | 36 | 36 |
| $\lambda_5$ | 3.3688 | 2.4207 | 0.0843 | 0.0843 | 32 | 32 |
| $\lambda_6$ | 3.6359 | 1.8651 | 0.0830 | 0.0830 | 25 | 24 |
| $\lambda_7$ | 3.9089 | 2.8070 | 0.0811 | 0.0811 | 18 | 18 |
| $\lambda_8$ | 4.2888 | 2.8444 | 0.0801 | 0.0801 | 13 | 13 |
| $\lambda_9$ | 4.1552 | 2.8232 | 0.0801 | 0.0801 | 13 | 13 |
| $\lambda_{10}$ | 4.6611 | 2.8770 | 0.0801 | 0.0801 | 13 | 13 |
| | MAD | | $\text{VaR}_{0.01}$ | | $\text{ES}_{0.01}$ | |
| | mad 0.4617 | naive 0.6069 | mad 1.5746 | naive 2.0784 | mad 1.7960 | naive 2.3732 |
| | MAD-Lasso (IP) | MAD-Lasso (NEW) | MAD-Lasso (IP) | MAD-Lasso (NEW) | MAD-Lasso (IP) | MAD-Lasso (NEW) |
| $\lambda_1$ | 0.4457 | 0.4457 | 1.5556 | 1.5556 | 1.7742 | 1.7742 |
| $\lambda_2$ | 0.4458 | 0.4458 | 1.5559 | 1.5559 | 1.7745 | 1.7745 |
| $\lambda_3$ | 0.4461 | 0.4460 | 1.5576 | 1.5576 | 1.7765 | 1.7765 |
| $\lambda_4$ | 0.4471 | 0.4471 | 1.5620 | 1.5620 | 1.7815 | 1.7815 |
| $\lambda_5$ | 0.4501 | 0.4501 | 1.5716 | 1.5716 | 1.7925 | 1.7925 |
| $\lambda_6$ | 0.4576 | 0.4576 | 1.5948 | 1.5948 | 1.8191 | 1.8191 |
| $\lambda_7$ | 0.4696 | 0.4696 | 1.6313 | 1.6313 | 1.8609 | 1.8609 |
| $\lambda_8$ | 0.4770 | 0.4770 | 1.6517 | 1.6517 | 1.8843 | 1.8843 |
| $\lambda_9$ | 0.4770 | 0.4770 | 1.6517 | 1.6517 | 1.8843 | 1.8843 |
| $\lambda_{10}$ | 0.4770 | 0.4770 | 1.6517 | 1.6517 | 1.8843 | 1.8843 |

Simulation results of Table 5-6 show that with increasing tuning parameter $\lambda$, we can achieve higher level Sparsity and smaller Sharpe ratio. Moreover, the values of MAD, $\text{VaR}_{0.01}$, $\text{ES}_{0.01}$ increases accordingly. For both datasets, MAD model (mad) outperforms Naive model (naive) in terms of Sharpe ratio and risk measures. MAD-Lasso models with smaller tuning parameters can achieve better performance than MAD model. For MAD-Lasso models, results show that our proposed algorithm (NEW) is much more time efficient than interior point method (IP). With properly chosen tuning parameters, MAD-Lasso models can achieve higher Sharpe ratio and smaller risk than MAD model (mad) and Naive model (naive). On the other hand, with increasing tuning parameter $\lambda$, MAD-Lasso models can achieve better performance than MAD Model by sacrificing a little bit Sharpe ratio and risk.

**Table 6.** Portfolio selection results of FF48m data: $(T, n) = (361, 48)$; ExpRet: $r_0 = 1.2606$.

| | Time | | Sharpe | | Sparsity | |
|---|---|---|---|---|---|---|
| | mad 0.9094 | naive 0.0001 | mad 0.4332 | naive 0.2450 | mad 48 | naive 48 |
| | MAD-Lasso(IP) | MAD-Lasso(NEW) | MAD-Lasso(IP) | MAD-Lasso(NEW) | MAD-Lasso(IP) | MAD-Lasso(NEW) |
| $\lambda_1$ | 0.3121 | 0.2698 | 0.4311 | 0.4324 | 48 | 48 |
| $\lambda_2$ | 0.2449 | 0.1782 | 0.4317 | 0.4304 | 47 | 48 |
| $\lambda_3$ | 0.2245 | 0.1363 | 0.4314 | 0.4316 | 47 | 48 |
| $\lambda_4$ | 0.2763 | 0.1646 | 0.4308 | 0.4308 | 42 | 44 |
| $\lambda_5$ | 0.2742 | 0.1154 | 0.4260 | 0.4265 | 40 | 39 |
| $\lambda_6$ | 0.2485 | 0.0977 | 0.4189 | 0.4177 | 29 | 27 |
| $\lambda_7$ | 0.2599 | 0.1042 | 0.3986 | 0.3985 | 19 | 19 |
| $\lambda_8$ | 0.2856 | 0.1217 | 0.3734 | 0.3734 | 15 | 15 |
| $\lambda_9$ | 0.2343 | 0.0932 | 0.3473 | 0.3473 | 10 | 10 |
| $\lambda_{10}$ | 0.2590 | 0.0945 | 0.3473 | 0.3481 | 10 | 10 |
| | MAD | | $\text{VaR}_{0.01}$ | | $\text{ES}_{0.01}$ | |
| | mad 2.1255 | naive 3.7957 | mad 8.0304 | naive 13.2301 | mad 9.0166 | naive 14.9736 |
| | MAD-Lasso(IP) | MAD-Lasso(NEW) | MAD-Lasso(IP) | MAD-Lasso(NEW) | MAD-Lasso(IP) | MAD-Lasso(NEW) |
| $\lambda_1$ | 2.0243 | 2.0247 | 8.0625 | 8.0419 | 9.0533 | 9.0298 |
| $\lambda_2$ | 2.0251 | 2.0261 | 8.0542 | 8.0738 | 9.0438 | 9.0662 |
| $\lambda_3$ | 2.0262 | 2.0281 | 8.0583 | 8.0557 | 9.0485 | 9.0455 |
| $\lambda_4$ | 2.0341 | 2.0347 | 8.0675 | 8.0686 | 9.0590 | 9.0603 |
| $\lambda_5$ | 2.0604 | 2.0600 | 8.1440 | 8.1359 | 9.1467 | 9.1374 |
| $\lambda_6$ | 2.1144 | 2.1236 | 8.2615 | 8.2806 | 9.2813 | 9.3032 |
| $\lambda_7$ | 2.2383 | 2.2389 | 8.6172 | 8.6197 | 9.6888 | 9.6916 |
| $\lambda_8$ | 2.4150 | 2.4150 | 9.1145 | 9.1146 | 10.2586 | 10.2586 |
| $\lambda_9$ | 2.6162 | 2.6162 | 9.7053 | 9.7053 | 10.9354 | 10.9354 |
| $\lambda_{10}$ | 2.6162 | 2.6169 | 9.7053 | 9.6843 | 10.9354 | 10.9113 |

**Example 2.** In this example, MAD-Lasso with interior point method (IP) and our proposed method (NEW) are taken into comparison. For FF48d and FF48m Datasets with tuning parameters chosen as $\lambda = (0.05 : 0.05 : 8) \cdot \sqrt{2T \log n}$ with multiply ratio $0.05\sqrt{2T \log n}$. Simulation results of MAD-Lasso models are displayed in Figure 9-10.

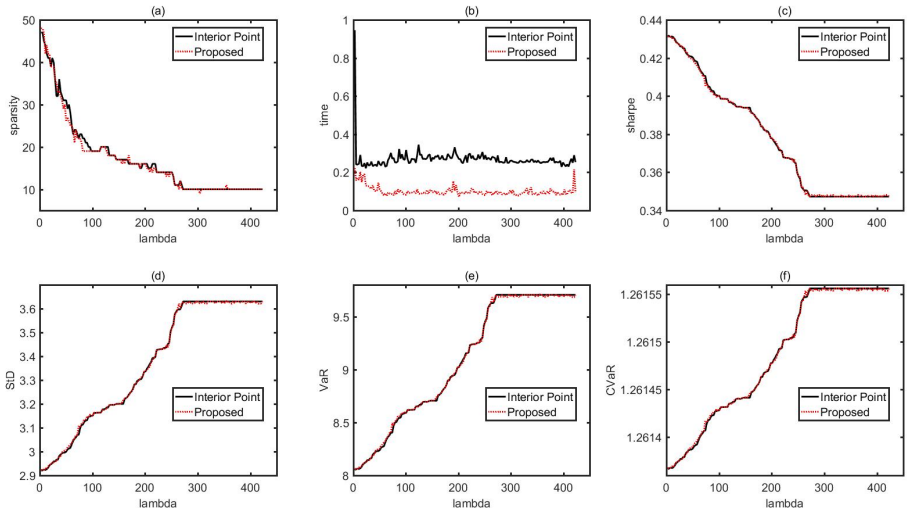


**Figure 9.** Portfolio selection tendency of FF48d data with increasing tuning parameter $\lambda$s.

Figure 9-10 show that our proposed method impose larger sparsity and is more time efficient than interior point method with larger tuning parameter $\lambda$. Moreover, after some point when $\lambda$ is large enough, the curves of Sharpe ratio, StD, $\text{VaR}_\alpha$ and $\text{CVaR}_\alpha$

with respect to $\lambda$ behave like horizontal lines, this confirmed the fact that the sparsity fixed with tuning parameter large enough, this coincide with Proposition 2.1 (2).

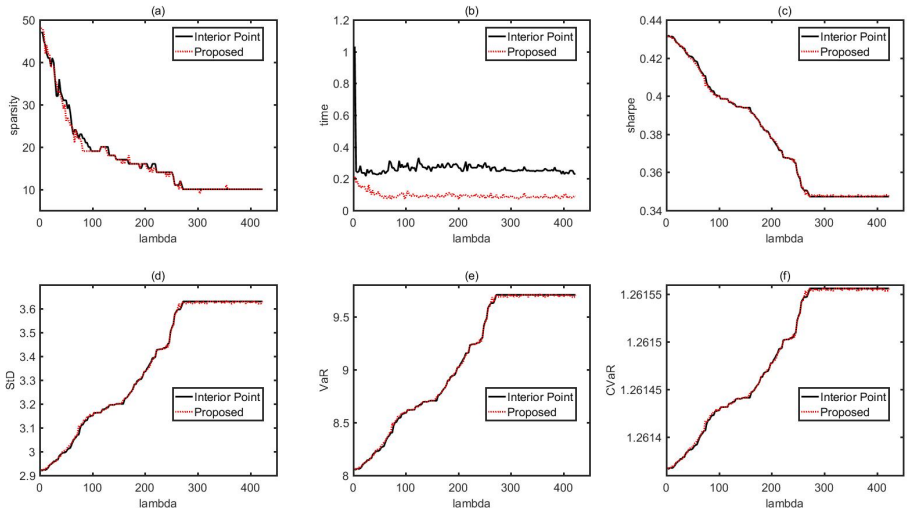


**Figure 10.** Portfolio selection tendency of FF48m data with increasing tuning parameter $\lambda$s.

## 6. Conclusion and Prospects

In this article, we proposed the MAD-Lasso portfolio selection strategy that can be reformulated as a constrained LAD Lasso problem with linear equality constraints. Based on the idea of nonsmooth optimality conditions, we derive a new descent algorithm that updates descent directions iteratively from a basis direction set. Under specific tolerance level and finite stopping conditions, extensive simulation studies and real data analysis verifies that our method is much more time efficient than interior point method and ADMM. The new method process equivalent magnitude of number of iterations with respect to interior point method. It is shown that MAD-Lasso models encourage sparsity and are robust against heavy tailed datasets, thus is recommended for portfolio selection. Future research directions include partial index tracking, portfolio hedging, and portfolio adjustment under MAD-Lasso framework.

## References


[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. *Distributed optimization and statistical learning via the Alternating Direction Method of Multipliers*. Foundations and Trends in Machine Learning Vol. 3.1 (2011), pp. 1–122.

[2] M. Broadie. *Computing efficient frontiers using estimated parameters*. Annals of Operations Research Vol. 45.1 (1993), pp. 21–58.

[3] J. Brodie, I. Daubechies, C. De Mol, D. Giannone, and I. Loris. *Sparse and stable Markowitz portfolios*. Proceedings of the National Academy of Sciences Vol. 106.30 (2009), pp. 12267–12272.

[4] M. Carrasco, and N. Noumon. *Optimal portfolio selection using regularization*. Discussion Paper (2011).

[5] A. Coppola, and B.M. Stewart. *lbfgs: Efficient l-bfgs and owl-qn optimization in r*, (2014).

[6] V. DeMiguel, L. Garlappi, and R. Uppal. *Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy*? The Review of Financial Studies Vol. 22.5 (2007), pp. 1915–1953.

[7] B. Fastrich, S. Paterlini, and P. Winker. *Penalized least squares for optimal sparse portfolio selection*. In COMPSTAT 2014 Conference Proceedings, International Conference on Computational Statistics, (2014).

[8] M. Fernandes, G. Rocha, and T. Souza. *Regularized minimum-variance portfolios using asset group information* (2012), pp. 1–28.

[9] P.A. Frost, and J.E. Savarino. *For better performance: constrain portfolio weights*. The Journal of Portfolio Management Vol. 15.1 (1988), pp. 29–34.

[10] B.R. Gaines, J. Kim and H. Zhou. *Algorithms for fitting the constrained lasso*. Journal of Computational and Graphical Statistics (2018), (just-accepted).

[11] X. Gao, and J. Huang. *Asymptotic analysis of high-dimensional LAD regression with LASSO*. Statistica Sinica Vol. 20.4 (2008), pp. 1485–1506.

[12] G.M. James, C. Paulson, and P. Rusmevichientong. *Penalized and constrained regression*. Technical Report Vol. 15 (2013).

[13] J.D. Jobson, and B. Korkie. *Estimation for Markowitz efficient portfolios*. Journal of the American Statistical Association Vol. 75.371 (1980), pp. 544–554.

[14] H. Konno, H. Yamazaki, *Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market*. Management Science Vol. 37.5 (1991), pp. 519–531.

[15] S. Kotz, T. Kozubowski, and K. Podgorski. *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media (2012).

[16] D.C. Liu, J. Nocedal. On the Limited Memory BFGS Method for Large Scale Optimization. Mathematical Programming Vol. 45.1-3 (1989), pp. 503C-528.

[17] H. Markowitz. *Portfolio selection*. The Journal of Finance Vol. 7.1 (1952), pp. 77–91.

[18] Y. Shi, Z. Feng, and K.F.C. Yiu. *A descent method for least absolute deviation lasso problems*. Optimization Letters (2017), pp. 1–17.

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

[20] H. Wang, G. Li, and G. Jiang. *Robust regression shrinkage and consistent variable selection through the LAD-Lasso*. Journal of Business & Economic Statistics Vol. 25.3 (2007), pp. 347–355.

[21] L. Wang. *The $L_1$ penalized LAD estimator for high dimensional linear regression*. Journal of Multivariate Analysis Vol. 120 (2013), pp. 135–151.

[22] S. Wright, and J. Nocedal. *Numerical optimization*. Springer Science Vol. 35 (1999), pp.67–68, 7.

[23] T.T. Wu, and K. Lange. *Coordinate descent algorithms for lasso penalized regression*. The Annals of Applied Statistics Vol. 2.1 (2008), pp. 224–244.

[24] Y.M. Yen. *A note on sparse minimum variance portfolios and coordinate-wise descent algorithms*. arXiv preprint arXiv:1005.5082 (2010).

[25] Y.M. Yen, and T.J. Yen. *Solving norm constrained portfolio optimization via coordinate-wise descent algorithms*. Computational Statistics & Data Analysis Vol. 76 (2014), pp. 737–759.

[26] H. Zou, T. Hastie, and R. Tibshirani. *On the degrees of freedom of the Lasso*. The Annals of Statistics Vol. 35.5 (2007), pp. 2173–2192.

## Appendix A. Interior Point Method and ADMM Algorithm

Reparameterize Model (3) with

$$\boldsymbol{y}^* \triangleq \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0}_n \end{pmatrix}, \boldsymbol{A}^* \triangleq \begin{pmatrix} \boldsymbol{A} \\ \lambda I_n \end{pmatrix}, T^* \triangleq T + n.$$

Model (3) becomes

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{A}^*\boldsymbol{x} - \boldsymbol{y}^*\|_1 \\ \text{s.t.} \quad & \boldsymbol{C}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0}. \end{aligned}$$

For convenience, we dropped the $*$ sign and obtain

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\| \\ \text{s.t.} \quad & \boldsymbol{C}\boldsymbol{x} = \boldsymbol{b}. \end{aligned}$$

where $A \in \mathbb{R}^{T\times n}, y \in \mathbb{R}^n, C \in \mathbb{R}^{q\times n}, b \in \mathbb{R}^q$.

To solve Problem (3), consider the following transformations. Let $\boldsymbol{u}^+, \boldsymbol{u}^-$ be the positive part and negative part of $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}$, and $\boldsymbol{x}^+, \boldsymbol{x}^-$ be the positive part and negative part of $\boldsymbol{x}$, then Problem (3) becomes

$$\begin{aligned} \arg\min_{\boldsymbol{x}} \quad & \boldsymbol{u}^+ + \boldsymbol{u}^- + \lambda\boldsymbol{x}^+ + \lambda\boldsymbol{x}^- \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{u}^+ - \boldsymbol{u}^- = \boldsymbol{y}, \\ & \boldsymbol{x} - \boldsymbol{x}^+ + \boldsymbol{x}^- = \boldsymbol{0}, \\ & \boldsymbol{C}\boldsymbol{x} = \boldsymbol{b}. \end{aligned} \tag{A1}$$

Problem (A1) is a general linear programming problem that can be solved using `linprog` provided in Matlab, the stopping conditions of interior point method is that the algorithm iterate until it reaches a point that satisfies the constraints to within tolerances and meanwhile the relative step length are small, and the tolerance is $1e^{-6}$. ADMM can be implemented in Matlab as described in Boyd et al.[1] with applications in ADMM, the stopping condition is that the primal and dual residuals below tolerance $1e^{-6}$.