Noname manuscript No.
(will be inserted by the editor)

A Descent Method for Least Absolute Deviation Lasso Problems

Yue Shi[1] · **Zhiguo Feng**[2,1] · **Cedric Yiu**[1]

Received: date / Accepted: date

Abstract Variable selection is an important method to analyze large quantity of data and extract useful information. Although least square regression is the most widely used scheme for its flexibility in obtaining explicit solutions, least absolute deviation (LAD) regression combined with lasso penalty becomes popular for its resistance to heavy-tailed errors in response variable, denoted as LAD-LASSO. In this paper, we consider the LAD-LASSO problem for variable selection. Based on a dynamic optimality condition of nonsmooth optimization problem, we develop a descent method to solve the nonsmooth optimization problem. Numerical experiments are conducted to confirm that the proposed method is more efficient than existing methods.

Keywords Least absolute deviation · LASSO · Nonsmooth optimization · Descent method

1 Introduction

At an era of information explosion, the extraction of useful information from massive datasets becomes an important issue. The process often involves selecting a subset of variables to explain certain observations and phenomena.

Zhiguo Feng
18281102@qq.com

Yue Shi
yue.shi@connect.polyu.hk

✉Cedric Yiu
macyiu@polyu.edu.hk

[1] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, P.R. China
[2] Department of Applied mathematics, Chongqing Normal University, Chongqing, P.R. China

It can be posed as a regression problem. Since the number of variables are not known in advance, a large dataset is often deployed in the selection process in order not to miss the key variables. In this way, the regression problem becomes a sparse fitting problem. Motivated by the non-negative garrote procedure of Breiman in [1], Tibshirani added sparsity into regression problems in [2] and constructed the Least Absolute Shrinkage and Selection Operator (LASSO) penalty. By adding a bound to the absolute sum of coefficients, LASSO could shrink some coefficients to zeroes and retain significant variables to maintain model interpretability. As a convex penalty, LASSO is solvable and flexible. Hastie et al. systematically summarized a series of lasso problems in [3], and displayed that LASSO could be extended to generalized linear models and multivariate analysis. The comprehensive advantages made lasso popular and active in engineering, finance, marketing, bioinformatics and other related fields.

In practical applications, cases with heavy-tailed errors contain outliers are ubiquitous and would deteriorate estimation accuracy significantly. As an alternative to ordinary least square regression, Least Absolute Deviation (LAD) regression maintains robustness against fat tailed errors or extreme outliers due to its connection with L_1 norm and double exponential distribution. There are several approaches combining LAD regression with certain penalty terms for variable selection problems. For example, Zeebari united the LAD regression with ridge penalty, and alleviated the multi-collinearity between variables in [4]. Wang et al. proposed a consistent tuning parameter selection technique for LAD-LASSO, and extensively studied the relative asymptotic properties in [5]. In [6], Gao studied the high dimensional LAD-LASSO problem systematically, and confirmed the corresponding asymptotic properties. In [7], Arslan introduced the weighted LAD-LASSO by adaptively adding up a weighting process to mitigate the influence of outliers against both explanatory variables and response variable. In [8], Xu introduced a two-stage method for tuning parameter selection and obtained the oracle property. Various LAD-lasso related studies have been conducted and the corresponding theoretical properties are well constructed.

Since LAD-LASSO is more robust and could be easily extended to other situations, efficient solution to this problem become imperative and necessary. Generally, LAD-LASSO could be transformed to classical linear programming problem so that they could be computed easily. As an alternative to simplex method, Koender proposed the interior point with a preprocessing step in [9]. Watson and Yiu [10] dealt with the error-in-variable l_1 norm regression using Levenberg-Marquardt method, and robust solutions are obtained accordingly. Yiu et al. [11] applied l_1-norm to beamforming design and proposed an algorithm with a set of adaptive grids to speed up the calculation process. However, existing algorithms for solving LAD-LASSO is restrictive and rely heavily on the linear programming solvers. In this paper, we study and propose a more efficient method by selecting a sequence of fastest descent directions based on dynamic optimality condition.

The rest of this paper is organized as follows. In section 2, the LAD-lasso

based linear programming problem is formulated. The optimality condition of this nonsmooth optimization problem is derived in section 3. To solve this problem, we analyze the optimality condition and develop a descent method in section 4. Simulation experiments and real data examples are given in section 5. Conclusions are given in section 6.

2 LAD-LASSO Problem

Consider linear regression problem

$$Y = X\boldsymbol{\beta} + \varepsilon, \tag{2.1}$$

where X is the $n \times p$ design matrix with row vectors $X_i \in \mathbb{R}^p, i = 1, \cdots, n$, and $Y = (y_1, \cdots, y_n)^T$ is the response vector, $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)^T$ is the parameter vector we are concerned. Generally, the LAD-LASSO regression is to minimize the l_1 norm loss function

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} |y_i - X_i\boldsymbol{\beta}|$$

subject to the constraint

$$\sum_{i=1}^{p} |\beta_i| < c,$$

where c is a positive constant.

This problem can be transformed into the following optimization problem:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} |y_i - X_i\boldsymbol{\beta}| + \gamma \sum_{j=1}^{p} |\beta_j|,$$

or the matrix representation

$$\min_{\boldsymbol{\beta}} \|Y - X\boldsymbol{\beta}\|_1 + \gamma\|\boldsymbol{\beta}\|_1. \tag{2.2}$$

Note that the terms in (2.2) are nonsmooth. A typical way to tackle this problem is to transform it into a linear programming problem. Denote

$$\|Y - X\boldsymbol{\beta}\|_1 = \boldsymbol{u}_1 + \boldsymbol{v}_1, \quad \|\boldsymbol{\beta}\|_1 = \boldsymbol{u}_2 + \boldsymbol{v}_2, \tag{2.3}$$

where $\boldsymbol{u}_1, \boldsymbol{v}_1, \boldsymbol{u}_2, \boldsymbol{v}_2 \geq \mathbf{0}$ and $\boldsymbol{u}_1, \boldsymbol{v}_1 \in \mathbb{R}^n$, $\boldsymbol{u}_2, \boldsymbol{v}_2 \in \mathbb{R}^p$ are defined as

$$\begin{aligned}
\boldsymbol{u}_1 &= \max\big(Y - X\boldsymbol{\beta}, \mathbf{0}\big) \\
\boldsymbol{v}_1 &= \max\big(-(Y - X\boldsymbol{\beta}\big), \mathbf{0}) \\
\boldsymbol{u}_2 &= \max(\boldsymbol{\beta}, \mathbf{0}) \\
\boldsymbol{v}_2 &= \max(-\boldsymbol{\beta}, \mathbf{0})
\end{aligned}$$

Hence

$$Y - X\boldsymbol{\beta} = \boldsymbol{u}_1 - \boldsymbol{v}_1, \boldsymbol{\beta} = \boldsymbol{u}_2 - \boldsymbol{v}_2$$

and (2.2) is equivalent to the following minimization problem:

$$\begin{aligned} \min \quad & \mathbf{0}\cdot\boldsymbol{\beta}+\boldsymbol{u}_1+\boldsymbol{v}_1+\gamma\boldsymbol{u}_2+\gamma\boldsymbol{v}_2 \\ \text{s.t.} \quad & X\boldsymbol{\beta}+\boldsymbol{u}_1-\boldsymbol{v}_1=Y \\ & \boldsymbol{\beta}-\boldsymbol{u}_2+\boldsymbol{v}_2=\mathbf{0} \\ & \boldsymbol{u}_1,\boldsymbol{v}_1,\boldsymbol{u}_2,\boldsymbol{v}_2\geq\mathbf{0} \end{aligned}$$

Denote

$$A=\begin{pmatrix} X & I & -I & \mathbf{0} & \mathbf{0} \\ I & \mathbf{0} & \mathbf{0} & -I & I \end{pmatrix}, \quad \boldsymbol{b}=\begin{pmatrix} Y \\ \mathbf{0} \end{pmatrix},$$

the optimization problem becomes

$$\begin{aligned} \min \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x}=\boldsymbol{b} \\ & \boldsymbol{u}_1,\boldsymbol{v}_1,\boldsymbol{u}_2,\boldsymbol{v}_2\geq\mathbf{0}, \end{aligned} \tag{2.4}$$

where $\boldsymbol{x}=(\boldsymbol{\beta}^T,\boldsymbol{u}_1^T,\boldsymbol{v}_1^T,\boldsymbol{u}_2^T,\boldsymbol{v}_2^T)^T, \boldsymbol{c}=(\mathbf{0},I,I,\gamma I,\gamma I)$.

Thus, (2.4) is a canonical linear programming problem and interior point method can be applied to solve it. This is currently the state-of-art technique for tackling the LAD-LASSO problem. However, when n and p become large, the computational time still grows significantly and becomes very expensive.

3 Optimality Condition

Problem (2.2) can be written as a canonical form by introducing the symbols as follows:

$$Y^*=\begin{pmatrix} Y \\ \mathbf{0} \end{pmatrix}, X^*=\begin{pmatrix} X \\ \gamma\cdot I \end{pmatrix},$$

where $\mathbf{0}$ is $p\times 1$ vector, I is p-dimensional identity matrix. Then, Problem (2.2) becomes

$$\min_{\boldsymbol{\beta}} \|Y^*-X^*\boldsymbol{\beta}\|_1. \tag{3.1}$$

For simplicity of notation, we omit the superscript $*$ and consider the canonical form

$$\min_{\boldsymbol{\beta}} \|Y-X\boldsymbol{\beta}\|_1. \tag{3.2}$$

Introducing the objective function $f(\boldsymbol{\beta})$, the optimization problem (3.2) is standardized as

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta})=\sum_{i=1}^{n}|X_i\boldsymbol{\beta}-y_i|=\sum_{i=1}^{n}f_i(\boldsymbol{\beta}), \tag{3.3}$$

where

$$f_i(\boldsymbol{\beta})=|X_i\boldsymbol{\beta}-y_i|=\begin{cases} X_i\boldsymbol{\beta}-y_i, & \text{if } X_i\boldsymbol{\beta}-y_i>0, \\ -X_i\boldsymbol{\beta}+y_i, & \text{if } X_i\boldsymbol{\beta}-y_i<0, \\ 0, & \text{if } X_i\boldsymbol{\beta}-y_i=0. \end{cases}$$

To develop an efficient method for solving Problem (3.2), the optimality conditions are needed. The derivative of f_i with respect to $\boldsymbol{\beta}$ is given by

$$\frac{\partial f_i}{\partial \boldsymbol{\beta}} = \begin{cases} X_i, & \text{if } X_i\boldsymbol{\beta} - y_i > 0, \\ -X_i, & \text{if } X_i\boldsymbol{\beta} - y_i < 0. \end{cases}$$

At the point when $X_i\boldsymbol{\beta} - y_i = 0$, it's not differentiable. However, its directional derivative exists. For a direction $d \in \mathbb{R}^n$, the directional derivative of f_i along d is defined as

$$\nabla_{d^+} f_i = \lim_{\lambda \to 0^+} \frac{|X_i(\boldsymbol{\beta} + \lambda d) - y_i| - |X_i\boldsymbol{\beta} - y_i|}{\lambda \|d\|} = \frac{|X_i d|}{\|d\|}.$$

Similarly, for the direction $-d$, directional derivative of f_i along $-d$ is defined as

$$\nabla_{d^-} f_i = \lim_{\lambda \to 0+} \frac{|X_i(\boldsymbol{\beta} - \lambda d) - y_i| - |X_i\boldsymbol{\beta} - y_i|}{\lambda \|d\|} = \frac{|X_i d|}{\|d\|}.$$

Hence, for the absolute linear function, we have

$$\nabla_{d^-} f_i = \nabla_{d^+} f_i.$$

Furthermore, if $X_i\boldsymbol{\beta} - y_i \neq 0$, then f_i is smooth and we have

$$\nabla_{d^-} f_i = -\nabla_{d^+} f_i.$$

Denote $X_i\boldsymbol{\beta} - y_i = u_i$, we rewrite the objective function as

$$f(\boldsymbol{\beta}) = A(\boldsymbol{\beta}) + C(\boldsymbol{\beta}),$$

where $A(\boldsymbol{\beta})$ is the smooth part of $f(\boldsymbol{\beta})$,

$$A(\boldsymbol{\beta}) = \sum_{i=1}^{n} \chi(u_i > 0)(X_i\boldsymbol{\beta} - y_i) + \sum_{i=1}^{n} \chi(u_i < 0)(-X_i\boldsymbol{\beta} + y_i) \triangleq a^T\boldsymbol{\beta} + b,$$

in which

$$\begin{aligned} \chi(\nu) &= \begin{cases} 1, \text{ if } \nu \text{ is true}, \\ 0, \text{ otherwise}, \end{cases} \\ a^T &= \sum_{i=1}^{n} \chi(u_i > 0)X_i - \sum_{i=1}^{n} \chi(u_i < 0)X_i, \\ b &= -\sum_{i=1}^{n} \chi(u_i > 0)y_i + \sum_{i=1}^{n} \chi(u_i < 0)y_i, \end{aligned}$$

and $C(\boldsymbol{\beta})$ is the nonsmooth part of $f(\boldsymbol{\beta})$.
Denote the zero set in each iteration by $\Omega_k = \{k_1, \cdots, k_m\}$, which is the set of all the indices i such that $u_i = 0$. Then

$$C(\boldsymbol{\beta}) = \sum_{i=1}^{n} \chi(u_i = 0)|X_i\boldsymbol{\beta} - y_i| = \sum_{i=1}^{m} |X_{k_i}\boldsymbol{\beta} - y_{k_i}| = \sum_{i \in \Omega_k} |X_i\boldsymbol{\beta} - y_i|.$$

Since $f(\boldsymbol{\beta})$ is the sum of n convex functions, it is convex and its local minimizer is also the global minimizer. The optimality condition of the minimizer is that any directional derivatives are greater than or equal to zero. That is, $\boldsymbol{\beta}^*$ is the optimal solution of (3.3) if and only if

$$\nabla_d f(\boldsymbol{\beta}^*) = \Big(\nabla_d A(\boldsymbol{\beta}^*) + \nabla_d C(\boldsymbol{\beta}^*)\Big) \geq 0, \quad \forall d \in \mathbb{R}^p. \tag{3.4}$$

However, it is not easy to verify this condition during computation since d is arbitrary. We should derive an equivalent condition such that it can be verified easily. Consider the function $C(\boldsymbol{\beta})$ such that

$$X_{k_i}\boldsymbol{\beta} = y_{k_i}, \quad i = 1, \cdots, m.$$

Denote

$$X_a = \begin{pmatrix} X_{k_1} \\ \vdots \\ X_{k_m} \end{pmatrix},$$

and suppose that the rank of X_a is m, we can find its generalized inverse matrix as V such that $X_a V = I_m$, where I_m is the $m \times m$ identity matrix and $V = (V_1, \cdots, V_m)$.

Consider the null space $\{V \in \mathbb{R}^p | X_a V = 0\}$. There exist $p - m$ linear independent vectors $V_j, j = m+1, \cdots, p$, which are the basis of the null space. Hence, we have

$$X_a V_j = 0, \forall j = m+1, \cdots, p.$$

Therefore, $\{V_i : i = 1, \cdots, p\}$ form a basis of $\mathbb{R}^p$ and the following orthonormality holds:

$$X_{k_i} V_j = \begin{cases} 1, \text{ when } i = j; \\ 0, \text{ when } i \neq j, \end{cases} \quad i = 1, \cdots, m, \quad j = 1, \cdots, p, \tag{3.5}$$

and we can obtain the directional derivatives of f along the vectors $\{V_j : j = 1, \cdots, p\}$. If $i \in \{1, \cdots, m\}$, we have

$$\nabla_{V_i^+} C(\boldsymbol{\beta}) = \frac{\sum_{j=1}^m |X_{k_j} V_i|}{\|V_i\|} = \frac{1}{\|V_i\|}, \quad i = 1, \cdots, m,$$
$$\nabla_{V_i^-} C(\boldsymbol{\beta}) = \frac{\sum_{j=1}^m |X_{k_j} (-V_i)|}{\|-V_i\|} = \frac{1}{\|V_i\|}, \quad i = 1, \cdots, m.$$

If $i \in \{m+1, \cdots, p\}$, we have

$$\nabla_{V_i} C(\boldsymbol{\beta}) = \frac{\sum_{j=1}^m |X_{k_j} V_i|}{\|V_i\|} = 0, i = m+1, \cdots, p.$$

Consequently, we have

$$\begin{aligned}
&\nabla_{V_i^+} f(\boldsymbol{\beta}) = \nabla_{V_i^+} A(\boldsymbol{\beta}) + \frac{1}{\|V_i\|} = (a^T V_i + 1)/\|V_i\|, \quad i = 1, \cdots, m. \\
&\nabla_{V_i^-} f(\boldsymbol{\beta}) = \nabla_{V_i^-} A(\boldsymbol{\beta}) + \frac{1}{\|V_i\|} = (-a^T V_i + 1)/\|V_i\|, \quad i = 1, \cdots, m \\
&\nabla_{V_i} f(\boldsymbol{\beta}) = \nabla_{V_i} A(\boldsymbol{\beta}) = a^T V_i/\|V_i\|, \quad i = m+1, \cdots, p.
\end{aligned} \tag{3.6}$$

An equivalent optimal condition of (3.4) is given by the following theorem.

Theorem 1 *$\boldsymbol{\beta}^*$ is the optimal solution if and only if the directional derivatives satisfy*

$$\begin{aligned}
&\nabla_{V_i^+} f(\boldsymbol{\beta}^*) \geq 0, \quad i = 1, \cdots, m. \\
&\nabla_{V_i^-} f(\boldsymbol{\beta}^*) \geq 0, \quad i = 1, \cdots, m. \\
&\nabla_{V_i} f(\boldsymbol{\beta}^*) = 0, \quad i = m+1, \cdots, p.
\end{aligned} \tag{3.7}$$

Proof. Note that (3.7) is a special case of (3.4), the necessary condition is obvious. Therefore, we only proof the sufficient condition, that is, we prove that if (3.7) are satisfied, then (3.4) holds.

For any direction d, since $\{V_i : i = 1, \cdots, p\}$ is a basis of $\mathbb{R}^p$, there exists a vector λ, such that

$$d = \sum_{i=1}^{p} \lambda_i V_i. \tag{3.8}$$

Without loss of generality, we can set $\lambda_i \geq 0, \forall i = 1, \cdots, p$, because if $\lambda_i < 0$, we have $\lambda_i V_i = (-\lambda_i) \cdot V_i^-$. Then V_i^+ is replaced by V_i^-, and λ_i is replaced by $-\lambda_i > 0$.

Hence, by adjusting the order adequately, (3.8) can be reorganized as

$$d = \sum_{i=1}^{m_1} \lambda_i V_i^+ + \sum_{i=m_1+1}^{m} \lambda_i V_i^- + \sum_{i=m+1}^{p} \lambda_i V_i.$$

where $\lambda_i \geq 0, \forall i = 1, \cdots, p$. It follows from (3.6) that

$$\begin{aligned}
\nabla_d C(\boldsymbol{\beta}^*) &= \frac{\sum_{i=1}^{m} |X_{k_i} d|}{\|d\|} \\
&= \frac{\sum_{i=1}^{m} \left| X_{k_i} \left(\sum_{j=1}^{m_1} \lambda_j V_j^+ + \sum_{j=m_1+1}^{m} \lambda_j V_j^- + \sum_{j=m+1}^{p} \lambda_j V_j \right) \right|}{\|d\|} \\
&= \frac{\sum_{i=1}^{m_1} \left| \lambda_i X_{k_i} V_i^+ \right| + \sum_{i=m_1+1}^{m} \left| \lambda_i X_{k_i} V_i^- \right|}{\|d\|} = \frac{\sum_{i=1}^{m} \lambda_i}{\|d\|}.
\end{aligned}$$

Hence, by (3.6), we have

$$\begin{aligned}\nabla_d f(\boldsymbol{\beta}^*) &= \left(\sum_{i=1}^{m_1}\lambda_i a^T V_i^+ + \sum_{i=m_1+1}^{m}\lambda_i a^T V_i^- + \sum_{i=m+1}^{p}\lambda_i a^T V_i + \sum_{i=1}^{m}\lambda_i\right)\Big/\|d\| \\ &= \left(\sum_{i=1}^{m_1}\lambda_i (a^T V_i^+ + 1) + \sum_{i=m_1+1}^{m}\lambda_i (a^T V_i^- + 1) + \sum_{i=m+1}^{p}\lambda_i a^T V_i\right)\Big/\|d\| \\ &= \left(\sum_{i=1}^{m_1}\nabla_{V_i^+} f(\boldsymbol{\beta}^*)\cdot\|V_i\| + \sum_{i=m_1+1}^{m}\nabla_{V_i^-} f(\boldsymbol{\beta}^*)\cdot\|V_i\|\right)\Big/\|d\| \\ &\geq 0.\end{aligned}$$

Thus for any direction d, the directional derivative is greater than or equals to zero. Hence, (3.4) holds and $\boldsymbol{\beta}^*$ is the optimal solution. The proof is completed.

□

Remark 1 If the rank of X_a is l, and $l < m$, we can find l rows such that they are rank l. Then, the generalized inverse matrix $V = (V_1, \cdots, V_l)$ can be computed, and Theorem 1 still holds by replacing m by l. The proof is similar to that of Theorem 1.

4 Computational method

4.1 Descent direction

If the condition (3.7) is not satisfied, then there exists a direction d such that the cost function value decreases along with this direction. If the ith condition is not satisfied, that is,

$$\nabla_{V_i^+} f(\boldsymbol{\beta}) \geq 0, \text{ and } \nabla_{V_i^-} f(\boldsymbol{\beta}) \geq 0$$

can not be satisfied at the same time, then V_i^+ or V_i^- is the descent direction. For an iterative point $\boldsymbol{\beta}^{(k)}$, denote the zero set by Ω_k. The function can be rewritten as

$$f(\boldsymbol{\beta}) = a^{(k)T}\boldsymbol{\beta} + \sum_{i\in\Omega_k}|X_i\boldsymbol{\beta} - y_i| + b^{(k)}. \tag{4.1}$$

We need to find a descent direction such that (4.1) decreases along it whenever the condition (3.7) is not satisfied.

Since there exists at least one $i \in \{1, \cdots, m\}$ such that condition (3.7) is not satisfied. Denote the set of all the indices k_i by Ω_k', where (3.7) is not satisfied for V_i^+ or V_i^-. Then, we can choose the descent direction d in the space spanned by

$$\{V_i : k_i \in \Omega_k'\} \cup \{V_i : i = m+1, \cdots, p\}.$$

To speed up the search, we check the descent directional derivatives $\nabla_{V_i^+} f$ or $\nabla_{V_i^-} f$, and choose the indices where they descent most. That is, we choose a subset Λ_1 of Ω_k', which is a proportional α of the indices in Ω_k' such that the corresponding descent directional derivatives $\nabla_{V_i^+} f$ or $\nabla_{V_i^-} f$ is less than the other $1-\alpha$ of the directional derivatives. Denote

$$\Omega_{0k} = \Omega_k \backslash \Lambda_1,$$

we choose the descent direction d in the space spanned by

$$\{V_i : k_i \in \Lambda_1\} \cup \{V_i : i = m+1, \cdots, p\}$$

such that

$$d = \sum_{k_i \in \Lambda_1} \lambda_i V_i + \sum_{i=m+1}^{p} \lambda_i V_i.$$

It can be verified that

$$X_i d = 0, \quad \forall i \in \Omega_{0k}.$$

Hence, the descent direction should keep the set Ω_{0k} unchanged, we set the descent direction $d^{(k)}$ as the optimal solution of

$$\begin{aligned} \max_{h \in \mathbb{R}^p} \quad & -a^{(k)} h \\ s.t. \quad & X_i h = 0, \forall i \in \Omega_{0k}. \end{aligned} \tag{4.2}$$

It means that the solution h is chosen as the vector nearest to the deepest descent direction $-a^{(k)}$, and still keep the set Ω_{0k} unchanged at the same time. The optimal solution of Problem (4.2) is

$$\tilde{d} = -a^{(k)} - X_{0k}^T (X_{0k} X_{0k}^T)^{-1} X_{0k} \cdot (-a^{(k)}), \tag{4.3}$$

where $X_{0k}^T (X_{0k} X_{0k}^T)^{-1} X_{0k} (-a^{(k)})$ is the projected direction of $-a^{(k)}$ in the subspace $\{h : X_i h = 0, i \in \Omega_{0k}\}$, and

$$X_{0k} = \begin{pmatrix} X_{k_1} \\ \vdots \\ X_{k_l} \end{pmatrix}, \quad k_1, \cdots, k_l \in \Omega_{0k}.$$

Hence, the descent direction $d^{(k)}$ can be chosen as the normalized vector of $\tilde{d}$

$$d^{(k)} = \tilde{d} / \|\tilde{d}\|, \tag{4.4}$$

and the zero set is updated as $\Omega_k = \Omega_{0k}$.

4.2 Optimum Step Length

The cost function value will decrease along the descent direction $d^{(k)}$, when the step length is small. The next iteration point will be generated by

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \lambda_k d^{(k)}, \lambda_k > 0,$$

where λ_k is the step length, which should be maximized such that the cost function value is reduced in largest magnitude. For this, we define a new problem as

$$\min_{\lambda \geq 0} g(\lambda)$$

where

$$g(\lambda) = f(\boldsymbol{\beta}^{(k+1)}) = f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)}), \quad \lambda \geq 0.$$

Since f is convex, $g(\lambda)$ is also convex, we can choose λ_k as the optimal solution of the problem $\min_{\lambda} g(\lambda)$. This problem is equivalent to the problem as follows:

$$\begin{aligned} \max_{\lambda \geq 0} \quad & \lambda \\ \text{s.t.} \quad & \nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)}) \geq 0 \\ & \nabla_{-d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)}) \geq 0. \end{aligned} \tag{4.5}$$

For this problem, we have the following observation.

Theorem 2 *There exists an optimal solution $\lambda^{(k)} > 0$ and at least one i in $\{1, \cdots, n\}$ such that $X_i(\boldsymbol{\beta}^{(k)} + \lambda^{(k)} d^{(k)}) = y_i$, that is, i is in the zero set at the point $\boldsymbol{\beta}^{(k)} + \lambda^{(k)} d^{(k)}$.*

Proof. If $\lambda = 0$, $d^{(k)}$ is a descent direction at $\boldsymbol{\beta}^{(k)}$, that is,

$$\nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)}) < 0.$$

Since $g(\lambda)$ is convex, $\frac{\partial g(\lambda)}{\partial \lambda}$ is monotonically increasing.

Note that

$$\begin{aligned} \frac{\partial g(\lambda)}{\partial \lambda} &= \lim_{\Delta\lambda \to 0} \frac{g(\boldsymbol{\beta}^{(k)} + (\lambda + \Delta\lambda) d^{(k)}) - g(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)})}{\Delta\lambda} \\ &= \|d^{(k)}\| \nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)}), \end{aligned}$$

then, the directional derivative

$$\nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)})$$

is monotonically increasing with respect to λ.

Note that each term is absolute linear function, $f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)})$ is piecewise linear and $\nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)})$ is piecewise constant. For each point where $\nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)})$ increases, there exists at least one index i such that u_i

changes from negative to positive or from positive to negative. All these indices i is in $\{1, \cdots, n\}$, which is finite. Suppose that

$$\lim_{\lambda \to +\infty} \nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)}) < 0,$$

we have

$$\lim_{\lambda \to +\infty} \nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)}) = \nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda' d^{(k)}) < 0,$$

where λ' is a sufficiently large value. Therefore

$$f(\boldsymbol{\beta}^{(k)} + \lambda' d^{(k)}) \leq f(\boldsymbol{\beta}^{(k)}) + \nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda' d^{(k)}) \to -\infty.$$

This contracts to the fact that $f \geq 0$, which is impossible. Thus we must have

$$\lim_{\lambda \to +\infty} \nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)}) \geq 0.$$

Since $\nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)})$ is piecewise linear, we can find a point λ' such that $\nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda' d^{(k)})$ becomes positive or zero in the first time. That is,

$$\begin{aligned} \nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)}) < 0, \quad \lambda < \lambda', \\ \nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)}) \geq 0, \quad \lambda \geq \lambda'. \end{aligned}$$

Hence, $\boldsymbol{\beta}^{(k)} + \lambda' d^{(k)}$ is the minimum point of $f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)})$.

Note that $\nabla_{d^{(k)}} f(\boldsymbol{\beta}^{(k)} + \lambda d^{(k)})$ is discontinuous at λ', there exists at least one index i in $\{1, \cdots, n\}$ such that

$$X_i(\boldsymbol{\beta}^{(k)} + \lambda' d^{(k)}) = 0.$$

The proof is completed. Hence, we can find the optimum step length in each iteration.

□

4.3 Algorithm

We denote λ_k as the optimum step length along the direction $d^{(k)}$. By using the step length λ_k, the cost function becomes

$$f(\boldsymbol{\beta}^{(k+1)}) = (a^{(k+1)})^T \boldsymbol{\beta}^{(k+1)} + \sum_{i \in \Omega_{k+1}} |X_i \boldsymbol{\beta}^{(k+1)}| + b^{(k+1)},$$

and the kth iteration terminated and moved to the $(k+1)$th iteration. For this update, the indices in Λ_1 have been removed from the zero set Ω_k. It follows from Theorem 2 that some indices move to the zero set. We denote all these indices by Λ_2, then a new zero set at $(k+1)$th iteration is generated as

$$\Omega_{k+1} = \Omega_k \cup \Lambda_2.$$

Hence, we find a new iterate as $\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \lambda_k d^{(k)}$, and the zero set is updated as Ω_{k+1}. We continue the iteration until the optimal condition (3.7) is satisfied. In summary, the algorithm is as follows:

Algorithm 1

Initialization: Choose an initial point $\boldsymbol{\beta}^{(0)}$, compute the corresponding set Ω_0, and compute the cost function $f(\boldsymbol{\beta}^{(0)})$. Set $k = 0$.
Step 1: (Terminate)
Generate the matrix V for the zero set Ω_k. If the condition (3.7) is satisfied, then stop and return the optimal solution and value. Otherwise, go to Step 2.
Step 2: (Descent Direction)
Find the α fastest descent directions as Λ_1, where α denotes the percentage of selected descent directions that decrease faster than the other $1-\alpha$ directions. Set $\Omega_{0k} = \Omega_k \backslash \Lambda_1$, and compute the descent direction $d^{(k)}$ using (4.4).
Step 3: (Optimal Step Length) Find the best step length λ_k by (4.5).
Step 4: (Iteration) Update $\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \lambda_k d^{(k)}$. Find Λ_2 and update the zero set as $\Omega_{k+1} = \Omega_{0k} \cup \Lambda_2$. Then we compute the cost function $f(\boldsymbol{\beta}^{(k+1)})$, let $k = k+1$ and go to Step 1.

5 Numerical Examples

In this section, Algorithm 1 is implemented to solve the LAD-LASSO problem, where parameter α controls the percentage of directions selected from the descent direction set. Too small or too large α values may result in unsteadiness or time inefficiency. Here α is set as 0.05 to reach a balance between stability and time consumption. We compare our proposed method with Interior Point method and Gurobi based on Matlab platform, where the default solver of function `linprog` is Interior Point method.

To solve LAD-LASSO problem, a key consideration is the tuning parameter selection. In [12], Wang focused on the high dimensional penalized least absolute deviation problem, and a tuning parameter selection procedure is given. Denote $\boldsymbol{x}_i$ as the ith column vector of design matrix X, we first scaled the dataset such that $\|\boldsymbol{x}_i\|_2^2 = n, i = 1, \cdots, p$, and choose $\lambda = \sqrt{2n \log p}$, which is rate consistent.

5.1 Simulation Study

Similar to Gao and Huang [6], we consider 4 simulation examples with data generated by

$$Y = X\boldsymbol{\beta} + \varepsilon, \quad \varepsilon \sim N(0, 1),$$

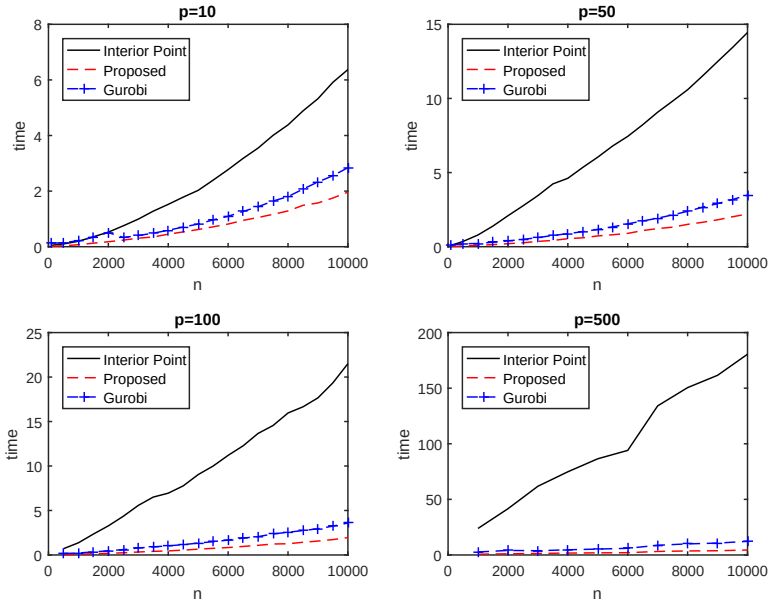
where design matrix X follows multivariate Gaussian distribution with zero mean vector and covariance matrix Σ, the elements of Σ is given by $(\Sigma)_{ij} = 0.5^{|i-j|}$ such that the correlation between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is $0.5^{|i-j|}$. For simplicity, the true coefficient $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta} = (2, 2, 2, 2, 2, 0, \cdots, 0),$$

where the first five elements equal to 2 and the remaining $p-5$ elements are zeroes, thus there are 5 nonzero components.

We consider four cases of p as 10, 50, 100, 500, respectively. For each p, the value of n increases from 500 (100 for $p=10$ case) to 10000 gradually. For each p and n, the data X and Y are simulated 100 times. Interior point method, the proposed method and Gurobi are applied to these problems for comparison. The running time of these methods are depicted in Figure 1. It can be seen that the proposed method is more efficient than the other methods, especially when n increases. That is, the larger n/p is, the more efficient the proposed method becomes.

Fig. 1 p=10,50,100,500 n–time plot



Several representative simulation results are listed in Table 1 - 4, where Running Time denotes the average time taken; MSE evaluates the average prediction error; Degree of Freedom (Zou[13]) refers to the number of nonzero components of the estimator; Correctly Fitted Ratio indicates accurate estima-

tion of nonzero component locations relative to the total simulation. Results show that Running Time, MSE and Correctly Fitted Ratio are same for all methods, which indicate that they have converged to the same optimal solution. Thus, our proposed method achieves both time efficiency and estimation accuracy.

Table 1 Simulation Results of $p = 10$:

		TIME			MSE		
n	p	Interior Point	Proposed	Gurobi	Interior Point	Proposed	Gurobi
100	10	0.0456	0.0103	0.8684	0.0680	0.0680	0.0680
500	10	0.0928	0.0355	0.8877	0.0110	0.0110	0.0110
1000	10	0.1845	0.0722	0.9540	0.0055	0.0055	0.0055
2000	10	0.4740	0.1771	1.2126	0.0027	0.0027	0.0027
5000	10	2.2176	0.7675	1.7595	0.0010	0.0010	0.0010
10000	10	6.5670	2.3630	3.8824	0.0005	0.0005	0.0005
		Degree of Freedom			Correctly Fitted Ratio		
n	p	Interior Point	Proposed	Gurobi	Interior Point	Proposed	Gurobi
100	10	5.19	5.19	5.19	0.83	0.83	0.83
500	10	5.29	5.29	5.29	0.77	0.77	0.77
1000	10	5.31	5.31	5.31	0.74	0.74	0.74
2000	10	5.23	5.23	5.23	0.79	0.79	0.79
5000	10	5.25	5.25	5.25	0.76	0.76	0.76
10000	10	5.18	5.18	5.18	0.83	0.83	0.83

Table 2 Simulation Results of $p = 50$:

		TIME			MSE		
n	p	Interior Point	Proposed	Gurobi	Interior Point	Proposed	Gurobi
100	50	0.0990	0.0182	1.0301	0.1044	0.1044	0.1044
1000	50	0.7803	0.1039	1.0755	0.0065	0.0065	0.0065
2000	50	2.0035	0.2363	1.2457	0.0033	0.0033	0.0033
5000	50	6.7602	1.0177	2.3907	0.0014	0.0014	0.0014
10000	50	15.2296	3.2660	4.8684	0.0007	0.0007	0.0007
		Degree of Freedom			Correctly Fitted Ratio		
n	p	Interior Point	Proposed	Gurobi	Interior Point	Proposed	Gurobi
100	50	5.21	5.21	5.21	0.80	0.80	0.80
1000	50	5.23	5.23	5.23	0.80	0.80	0.80
2000	50	5.35	5.35	5.35	0.69	0.69	0.69
5000	50	5.20	5.20	5.20	0.81	0.81	0.81
10000	50	5.25	5.25	5.25	0.81	0.81	0.81

Table 3 Simulation Results of $p = 100$:

		TIME			MSE		
n	p	Interior Point	Proposed	Gurobi	Interior Point	Proposed	Gurobi
500	100	1.3169	0.0770	1.2476	0.0161	0.0161	0.0161
1000	100	2.5999	0.1487	1.4152	0.0081	0.0081	0.0081
2000	100	5.2977	0.3179	1.8193	0.0036	0.0036	0.0036
5000	100	12.5036	1.0418	2.8586	0.0015	0.0015	0.0015
10000	100	29.2864	3.1812	6.2155	0.0008	0.0008	0.0008
		Degree of Freedom			Correctly Fitted Ratio		
n	p	Interior Point	Proposed	Gurobi	Interior Point	Proposed	Gurobi
500	100	5.33	5.33	5.33	0.75	0.75	0.75
1000	100	5.17	5.17	5.17	0.85	0.85	0.85
2000	100	5.29	5.29	5.29	0.74	0.74	0.74
5000	100	5.25	5.25	5.25	0.77	0.77	0.77
10000	100	5.39	5.39	5.39	0.67	0.67	0.67

Table 4 Simulation Results of $p = 500$:

		TIME			MSE		
n	p	Interior Point	Proposed	Gurobi	Interior Point	Proposed	Gurobi
1000	500	17.8460	0.5508	2.6924	0.0089	0.0089	0.0089
2000	500	32.9152	0.8235	3.4965	0.0047	0.0047	0.0047
5000	500	75.6648	1.8116	5.2965	0.0017	0.0017	0.0017
8000	500	118.5086	3.0154	8.4263	0.0011	0.0011	0.0011
10000	500	152.5622	4.3109	11.2252	0.0009	0.0009	0.0009
		Degree of Freedom			Correctly Fitted Ratio		
n	p	Interior Point	Proposed	Gurobi	Interior Point	Proposed	Gurobi
1000	500	5.26	5.26	5.26	0.77	0.77	0.77
2000	500	5.22	5.22	5.22	0.82	0.82	0.82
5000	500	5.23	5.23	5.23	0.77	0.77	0.77
8000	500	5.24	5.24	5.24	0.78	0.78	0.78
10000	500	5.31	5.31	5.31	0.72	0.72	0.72

5.2 Real Data Examples

In this section, we have selected 5 different real datasets for numerical experiment. Again, we compare our method with the interior point method and the Gurobi method. The datasets are as follows:

1. Prostate Cancer Data, which is studied by Stamey et al. [14] dealing with the correlation of 9 predictors and prostate specific antigen (lpsa).
2. Boston Housing Data, which is derived from Harrison and Rubinfeld [15] focussing on the 14 predictors that affect medv (median value of owner-occupied homes in $1000s).

3. Bardet Data, which is the simplified gene expression data presented by Scheetz et al. [17], where design matrix X is a 120×100 matrix expanded from the expression levels of 20 filtered genes. The objective is to discover the correlation between 100 predictors and the expression level of gene TRIM32 that causes Bardet-Biedl syndrome.
4. Diabetes Data, which is studied by Efron [16] containing 442 patients with 10 clinical measures: age, sex, body mass index(bmi), average blood pressure(map), and six blood serum measurements. The aim is to find the correlation between response y and the above 10 predictors.
5. China Stock Data, which considered by Wang [5] exploring the relationship of Return on Equity (ROE_{t+1}) and other 9 predictors.

Since all three methods found the same result, we focus on the execution time. Table 5 shows the running results for the 5 datasets:

Table 5 Time Comparison for real datasets

Name	n	p	Interior Point	Proposed	Gurobi
Prostate Cancer	97	8	0.0243	0.0067	0.6855
Boston Housing	506	13	0.0878	0.0382	0.7414
Bardet	120	100	0.1304	0.0514	0.6988
Diabetes	442	10	0.5127	0.0113	0.6989
China Stock	1946	9	0.2632	0.1163	1.3954

For the 5 datasets, time comparison of Interior Point method, our proposed method and Gurobi are summarized in Table 5, again our proposed method is faster than other methods.

6 Conclusion

In this paper, we have studied the LAD-LASSO problem for variable selection. For this nonsmooth optimization, we have derived the optimality condition and have developed a descent algorithm such that the nonsmooth optimization problem can be optimized directly. Numerical experiments with both simulated and real data have been employed to demonstrate that our proposed method is more efficient than the traditional interior point method and the state-of-the-art LP solver Gurobi.

Acknowledgement

This paper is partially supported by RGC Grant PolyU. 152200/14E and PolyU 4-ZZGS.

References

1. Breiman, L.: Better subset regression using the nonnegative garrote. Technometrics **37**(4), 373-384 (1995)
2. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), pp.267-288 (1996)
3. Hastie, T., Tibshirani, R., Wainwright, M.: Statistical learning with sparsity. CRC press (2015)
4. Zeebari, Z.: A Simulation Study on the Least Absolute Deviations Method for Ridge Regression. forthcoming in Communications in Statistics Theory and Methods (2012)
5. Wang, H., Li, G., Jiang, G.: Robust regression shrinkage and consistent variable selection through the LAD-Lasso. Journal of Business & Economic Statistics **25**(3), 347-355 (2007)
6. Gao, X., Huang, J.: Asymptotic analysis of high-dimensional LAD regression with LASSO. Statistica Sinica, pp.1485-1506 (2010)
7. Arslan, O.: Weighted LAD-LASSO method for robust parameter estimation and variable selection in regression. Computational Statistics & Data Analysis **56**(6), 1952-1965 (2012)
8. Xu, J., Ying, Z.: Simultaneous estimation and variable selection in median regression using Lasso-type penalty. Annals of the Institute of Statistical Mathematics **62**(3), 487-514 (2010)
9. Portnoy, S., Koenker, R.: The Gaussian hare and the Laplacian tortoise: computability of squared-error versus absolute-error estimators. Statistical Science **12**(4), 279-300 (1997)
10. Watson, G.A., Yiu, K.F.C.: On the solution of the errors in variables problem using the l_1 norm. BIT Numerical Mathematics **31**(4), 697-710 (1991)
11. Yiu K.F.C., Yang X., Nordholm S., et al. Near-field broadband beamformer design via multidimensional semi-infinite-linear programming techniques. IEEE Transactions on Speech and Audio processing **11**(6), 725-732 (2003)
12. Wang, L.: The L_1 penalized LAD estimator for high dimensional linear regression. Journal of Multivariate Analysis **120**, 135-151 (2013)
13. Zou, H., Hastie, T., Tibshirani, R.: On the degrees of freedom of the lasso. The Annals of Statistics **35**(5), 2173-2192 (2007)
14. Stamey, T.A., Kabalin, J.N., McNeal, J.E., Johnstone, I.M., Freiha, F., Redwine, E.A., Yang, N.: Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients. The Journal of urology **141**(5), 1076-1083 (1989)
15. Harrison, D., Rubinfeld, D.L.: Hedonic housing prices and the demand for clean air. Journal of environmental economics and management **5**(1), 81-102 (1978)
16. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. The Annals of statistics **32**(2), 407-499 (2004)
17. Scheetz, T.E., Kim, K.Y.A., et al.: Regulation of gene expression in the mammalian eye and its relevance to eye disease. Proceedings of the National Academy of Sciences **103**(39), 14429-14434 (2006)