A dual based semismooth Newton method for a class of sparse Tikhonov regularization

Ning Zhang†

January 15, 2020

Abstract. It is well known that Tikhonov regularization is one of the most commonly used methods for solving the ill-posed problems. One of the most widely applied approaches is based on constructing a new dataset whose sample size is greater than the original one. The enlarged sample size may bring additional computational difficulties. In this paper, we aim to make full use of Tikhonov regularization and develop a dual based semismooth Newton (DSSN) method without destroying the structure of dataset. From the point of view of theory, we will show that the DSSN method is a globally convergent method with at least R-superlinear rate of convergence. In the numerical computation aspect, we evaluate the performance of the DSSN method by solving a class of sparse Tikhonov regularization with high-dimensional datasets.

Key words. Semismooth Newton method; R-superlinear; dual based method; Tikhonov regularization

AMS Subject Classifications(2010): 65F22, 93B40, 90C20

1 Introduction

In order to overcome the drawbacks of the ordinary Least squares (OLS) in prediction accuracy and interpretation, Tibshirani introduced the Lasso technique [28], which is essentially an $\ell_1$ norm regularized least-square problem. With the emergence of a large number of low sample size and high dimensional data, Lasso and its variants have drawn more and more attentions. In this paper, we consider the sparse Tikhonov regularization problem in the following form

$$
\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \|Ax - b\|^2 + \frac{\lambda}{2} \|x\|^2 + \varphi(x),
$$

where $\lambda > 0$, $A \in \mathbb{R}^{m \times n}$ is a given data matrix, $\varphi : \mathbb{R}^n \to (-\infty, +\infty]$ is a closed proper convex (not necessarily smooth) function. If $\varphi(x) := \mu \|x\|_1$, $\mu > 0$, problem (1) reduces to the following elastic...
net regularized regression, which was proposed by Zou and Hastie [35] to resolve the limitations of Lasso,
\[
\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|^2 + \frac{\lambda}{2} \|x\|^2 + \mu \|x\|_1.
\] (2)

The above model has been extensively used in many fields, such as uncovering the consistent networks of functional disconnection in Alzheimer’s disease [27], estimating global bank network connectedness [6].

Inspired by the fact that the pixel values generated from image restoration problems should be nonnegative, Bai et al. [1] studied a nonnegative Tikhonov regularization as below:
\[
\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|^2 + \frac{\lambda}{2} \|x\|^2 + \mathbb{I}_{\mathbb{R}^n_+}(x),
\]
where \(\mathbb{I}_{\mathbb{R}^n_+}(x) = 0\) if \(x \geq 0\); \(\mathbb{I}_{\mathbb{R}^n_+}(x) = +\infty\) otherwise.

If \(\varphi(x) := \mu \|x\|_1 + \mathbb{I}_{\mathbb{R}^n_+}(x)\), \(\mu > 0\), problem (1) reduces to the following nonnegative elastic Lasso, which has been applied to estimate the microstructure indices from diffusion magnetic resonance data [5] and to track the market index [30]
\[
\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|^2 + \frac{\lambda}{2} \|x\|^2 + \mu \|x\|_1 + \mathbb{I}_{\mathbb{R}^n_+}(x).
\] (3)

One of the most popular way to find the solution of problem (1) is to reformulate it into the following regularized least square problem and directly apply some well-established algorithms, such as the block coordinate descent algorithm [9] and accelerated proximal gradient method [2, 3]:
\[
\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \|\tilde{A}x - \tilde{b}\|^2 + \varphi(x),
\] (4)
where
\[
\tilde{A} = \left(\frac{A}{\sqrt{\lambda}I}\right) \in \mathbb{R}^{(m+n) \times n}, \quad \tilde{b} = \left(\frac{b}{0}\right) \in \mathbb{R}^{m+n}.
\] (5)

It can be seen from (5) that the size of newly constructed data matrix \(\tilde{A}\) may increase greatly in the number of rows. This may bring additional issues by using the existing solution method.

More recently, the semismooth Newton based augmented Lagrangian algorithm (SSNAL) has shown excellent numerical performance for solving various large-scale Lasso-type problems [17, 18, 33]. From the relationship between the augmented Lagrangian algorithm and the proximal point algorithm [24, Section 4], we can observe that each subproblem of SSNAL is essentially a Tikhonov regularization problem (see Remark 3.1 for the details). Motivated by the observation, we aim to develop a dual based semismooth Newton method for solving the sparse Tikhonov regularization problem (1).

Directly from [23, Theorem 3.2], we know that the semismooth Newton method (SSN) for solving the primal formulation (1) converges locally quadratically to its unique solution \(x^*\) provided that the objective function \(f\) is strongly semismooth and every elements in \(\partial f(x^*)\) is nonsingular. It is well know that the local convergence results for convex optimization are usually not sufficient to guarantee the performance of SSN. An efficient globalization strategy has been proposed in [21].
This strategy mainly depends on a continuously differentiable convex function whose gradient is the KKT mapping (see e.g., [12, Page 625]) of the original optimization problem. However, the strategy used in [21] is no longer applicable to the primal formulation (1). The problem is due to the fact that the KKT mapping of problem (1) (see (26)) can not easily be viewed as a gradient mapping of any real valued function. On the other hand, though the smoothing Newton method studied in [10] is globally convergent, additional smoothing function should be introduced. In fact, by taking the advantage of Tikhonov regularization, we can show that the DSSN method converges globally and at least R-superlinearly.

The main contributions of this paper can be summarized as follows. Firstly, we propose a dual based semismooth Newton method (DSSN) for solving the sparse Tikhonov regularization problem (1). By fully taking advantage of the Tikhonov regularization, the propose DSSN method can avoid enlarging the scale of data matrix. Secondly, we prove that the proposed DSSN method is globally convergent and can achieve at least R-superlinear convergence rate. Finally, by deeply exploring the sparsity of the second-order information associated with the sparse regularizer, the robustness and effectiveness can be shown in the proposed method. This can be supported by the numerical results presented in section 4.

The rest of this paper is organized as follows. In section 2, we present some preliminary results that will be used for arithmetic design and numerical implementation. In section 3, we propose the dual based semismooth Newton method (DSSN) for solving the sparse Tikhonov regularization problem, and establish its convergence results. In section 4, we evaluate the numerical performance of the DSSN on UCI data sets. Finally, we conclude the paper in section 5.

**Notation:** Let $\mathbb{R}^n(\mathbb{R}_+, \mathbb{R}_-)$ be the set of all (non-negative, non-positive) $n$-vectors, $\mathbb{R}^{m \times n}$ be the set of all $m \times n$ real matrices, and $\mathbb{S}^n$ be the set of all $n \times n$ real symmetric matrices. Let $\mathcal{U}$ be a finite-dimensional real Hilbert space, $h: \mathcal{U} \to \mathbb{R} \cup \{+\infty\}$, we use $\partial_B h(u)$ and $\partial h(u)$ to denote the B-subdifferential [22, Equation (2.12)] and Clarke’s generalized Jacobian [4, Definition 2.6.1] of function $h$ at $u \in \mathcal{U}$, respectively. Let $C$ be a closed convex set in $\mathcal{U}$, we use $\Pi_C(u)$ to denote the Euclidean projection of $u \in \mathcal{U}$ onto $C$. Let $\text{Diag}(v)$ denote a diagonal matrix whose $i$-th diagonal entry is the $i$-th element of vector $v \in \mathbb{R}^n$. Let $\text{sign}(\cdot)$ denote the sign mapping on $\mathbb{R}^n$, i.e., $[\text{sign}(v)]_i = 1$, if $v_i > 0$; $[\text{sign}(v)]_i = -1$, if $v_i < 0$; $[\text{sign}(v)]_i = 0$, if $v_i = 0$, $i = 1, \ldots, n$. We denote the vector of all ones by $e$. Let index set $\mathcal{I} \subseteq \{1, \ldots, n\}$, for any matrix $A \in \mathbb{R}^{m \times n}$, we use $A_{\mathcal{I}}$ to denote $m \times |\mathcal{I}|$ sub-matrix of $A$ obtained by removing all the columns of $A$ not in $\mathcal{I}$. We use "·" to denote the Hadamard product between matrices and “◦” to denote function composition.

## 2 Preliminaries

Let $\phi: \mathcal{U} \to \mathbb{R} \cup \{+\infty\}$ be a proper, lower semicontinuous, convex function. Denote by $\Phi_\phi(u)$ the Moreau-Yosida regularization [20, 31] of $\phi$,

\[ \Phi_\phi(u) := \min_{u' \in \mathcal{U}} \left\{ \phi(u') + \frac{1}{2}\|u' - u\|^2 \right\}, \quad \forall u \in \mathcal{U}. \]
The proximal mapping associated with \( \phi \) is defined by

\[
\text{Prox}_\phi(u) := \arg \min_{u' \in \mathcal{U}} \left\{ \phi(u') + \frac{1}{2} \|u' - u\|^2 \right\}, \ \forall u \in \mathcal{U}.
\]

From e.g., [13, 16], we know that \( \Phi_\phi(\cdot) \) is a continuously differentiable convex function with its gradient being given by

\[
\nabla \Phi_\phi(u) = u - \text{Prox}_\phi(u).
\]

The following identity [25, Theorem 31.5] will be used in the subsequent analysis,

\[
\text{Prox}_\phi(u) + \text{Prox}_{\phi^*}(u) = u, \tag{6}
\]

where \( \phi^* \) is the conjugate function (for its definition, see e.g., [25, Page 104]) of \( \phi \).

**Lemma 2.1.** For any given \( u \in \mathbb{R}^n \), then the following hold:

(a) If \( \phi(x) = \mu \|x\|_1 \) and \( \mu > 0 \), then \( \phi^*(x) = 1_{\mathbb{B}_{\infty,\mu}}(x) \) with \( \mathbb{B}_{\infty,\mu} := \{ u \|u\|_\infty \leq \mu \} \) and

\[
\text{Prox}_{\phi}(u) = \text{sign}(u) \cdot \max\{ |u| - \mu e, 0 \}.
\]

(b) If \( \phi(x) = 1_{\mathbb{R}^n_+}(x) \), then \( \phi^*(x) = 1_{\mathbb{R}^n_+}(x) \) and \( \text{Prox}_{\phi}(u) = \Pi_{\mathbb{R}^n_+}(u) \).

**Lemma 2.2.** For any given \( v \in \mathbb{R}^n \), the following hold:

(a) The \( B \)-subdifferential of \( \text{Prox}_{\mu \|\cdot\|_1}(\cdot) \) at \( v \) is given by

\[
\partial_B \text{Prox}_{\mu \|\cdot\|_1}(v) = \left\{ \text{Diag}(\theta) \theta \in \mathbb{R}^n, \ \theta_i \in \left\{ \begin{array}{ll} \{1\}, & \text{if } |v_i| > \mu, \\
\{0,1\}, & \text{if } |v_i| = \mu, \ i = 1, \ldots, n \\
\{0\}, & \text{if } |v_i| < \mu, \end{array} \right. \right\}. \tag{7}
\]

(b) The \( B \)-subdifferential of \( \Pi_{\mathbb{R}^n_+}(\cdot) \) at \( v \) is given by

\[
\partial_B \Pi_{\mathbb{R}^n_+}(v) = \left\{ \text{Diag}(\theta) \theta \in \mathbb{R}^n, \ \theta_i \in \partial_B \max\{v_i, 0\}, \ i = 1, \ldots, n \right\},
\]

where for \( i = 1, \ldots, n, \)

\[
\partial_B \max\{v_i, 0\} = \left\{ \begin{array}{ll} \{1\}, & \text{if } v_i > 0, \\
\{0,1\}, & \text{if } v_i = 0, \\
\{0\}, & \text{if } v_i < 0. \end{array} \right. \tag{8}
\]

Since both \( \text{Prox}_{\mu \|\cdot\|_1}(\cdot) \) and \( \Pi_{\mathbb{R}^n_+}(\cdot) \) are piecewise linear functions, we can obtain the following results from [7, Proposition 7.47].

**Lemma 2.3.** For given \( \mu > 0 \), both \( \text{Prox}_{\mu \|\cdot\|_1}(\cdot) \) and \( \Pi_{\mathbb{R}^n_+}(\cdot) \) are strongly semismooth.

The proof of the following results can be obtained by using the same routine of [33, Proposition 2.1] and [17, Theorem 2]. In order to make this paper more readable, we will provide a sketch of the proof of the following lemma.
Lemma 2.4. For any given $\mu > 0$, let $\phi(x) := \mu \|x\|_1 + \Pi_{R^n_+}(x)$. Then it holds that

$$\text{Prox}_\phi(z) = \text{Prox}_{\mu \|\cdot\|_1} \circ \Pi_{R^n_+}(z), \ \forall z \in \mathbb{R}^n.$$  

Furthermore, define a set-valued function $\hat{\partial}_B \text{Prox}_\phi : \mathbb{R}^n \rightrightarrows \mathbb{R}^{n \times n}$ as

$$\hat{\partial}_B \text{Prox}_\phi(u) = \left\{ W = \Theta \Xi : \Theta \in \partial_B \text{Prox}_{\mu \|\cdot\|_1}(v), \ \Xi \in \partial_B \Pi_{R^n_+}(u), \ v = \Pi_{R^n_+}(u) \right\},$$

then the set-valued function $\hat{\partial}_B \text{Prox}_\phi$ is a nonempty compact valued upper-semicontinuous multifunction and for any $W \in \hat{\partial}_B \text{Prox}_\phi(u)$, $W$ is a symmetric positive semidefinite matrix and for $u' \to u$, it holds that

$$\text{Prox}_\phi(u') - \text{Prox}_\phi(u) - W(u' - u) = O(\|u' - u\|^2), \ \forall W \in \hat{\partial}_B \text{Prox}_\phi(u').$$

Proof. By [32, Theorem 1], in order to obtain equation (9), it is sufficient to show that

$$\partial \Pi_{R^n_+}(z) \subseteq \partial \Pi_{R^n_+}(\text{Prox}_{\mu \|\cdot\|_1}(z)), \ \forall z \in \mathbb{R}^n,$$

which can be obtained directly from the definition of Clarke’s generalized Jacobian and Lemma 2.2. Furthermore, it follows from Lemma 2.3, (9) and [7, Theorem 7.5.17] that the set-valued function $\hat{\partial}_B \text{Prox}_\phi$ is a nonempty compact valued upper-semicontinuous multifunction and equation (11) holds. Besides, the result that any elements in $\hat{\partial}_B \text{Prox}_\phi(u)$ is symmetric and positive semidefinite can be obtained from Lemma 2.2. The proof is completed.

Remark 2.1. In the numerical experiment, motivated by the idea in [18], for any given $v \in \mathbb{R}^n$, we choose the following element in $\partial_B \text{Prox}_{\mu \|\cdot\|_1}(v)$:

$$\Theta = \text{Diag}(\theta) \text{ with } \theta_i = \begin{cases} 1, & \text{if } |v_i| > \mu, \\ 0, & \text{if } |v_i| \leq \mu, \ i = 1, \ldots, n, \end{cases}$$

and choose one element $\Theta \in \partial_B \Pi_{R^n_+}(v)$ as below:

$$\Theta = \text{Diag}(\theta) \text{ with } \theta_i = \begin{cases} 1, & \text{if } v_i > 0, \\ 0, & \text{if } v_i \leq 0, \ i = 1, \ldots, n. \end{cases}$$

3 A Dual based Semismooth Newton Method

By introducing an auxiliary variable, problem (1) can be reformulated as

$$\min_{x,y} \frac{1}{2} \|y\|^2 + \frac{\lambda}{2} \|x\|^2 + \varphi(x) \quad \text{s.t.} \quad Ax - y - b = 0.$$  

(14)

The Lagrangian function associated with the above problem is given by

$$\mathcal{L}(x, y, z) = \frac{1}{2} \|y - z\|^2 + \frac{\lambda}{2} \|z\|^2 + \varphi(x) + (Ax - y - b, z) = \frac{1}{2} \|y - z\|^2 + \varphi(x) + \frac{\lambda}{2} \|x + \lambda^{-1} A^T z\|^2 - \frac{1}{2} \|\lambda^{-1} A^T z\|^2 - b^T z.$$  

5
Then, we can obtain that
\[
\min_{x,y} \mathcal{L}(x, y, z) = \min_y \left\{ \frac{1}{2} \|y - z\|^2 - \frac{1}{2} \|z\|^2 \right\} + \min_x \left\{ \varphi(x) + \frac{\lambda}{2} \|x + \lambda^{-1} A^T z\|^2 - \frac{\lambda}{2} \|\lambda^{-1} A^T z\|^2 \right\} - b^T z
\]
\[
= -\frac{1}{2} \|z\|^2 + \Phi_{\lambda^{-1}}(-\lambda^{-1} A^T z) - \frac{\lambda}{2} \|\lambda^{-1} A^T z\|^2 - b^T z.
\]
Consequently, the dual problem (minimization form) of (14) takes the following form:
\[
\min_z h(z) := \frac{1}{2} \|z\|^2 - \Phi_{\lambda^{-1}}(-\lambda^{-1} A^T z) + \frac{\lambda}{2} \|\lambda^{-1} A^T z\|^2 + b^T z, \tag{15}
\]
where \(\Phi_{\lambda^{-1}}(\cdot)\) is the Moreau-Yosida regularization of \(\lambda^{-1} \varphi\). And then, the optimal solution \(x^*\) of problem (1) can be obtained by
\[
x^* = \text{Prox}_{\lambda^{-1}}(-\lambda^{-1} A^T z^*), \tag{16}
\]
where \(z^*\) is the unique optimal solution of problem (15).

Since \(h(z)\) is a strongly convex and continuously differentiable function, we know that the unique solution of (15) can be obtained by solving the following linear equation:
\[
\nabla h(z) = 0, \tag{17}
\]
where \(\nabla h(z) = z + b - A \text{Prox}_{\lambda^{-1}}(-\lambda^{-1} A^T z)\). Furthermore, define
\[
\hat{\partial}_B(\nabla h)(z) := \left\{ I + \lambda^{-1} A W A^T \in \mathbb{R}^{m \times m} | W \in \hat{\partial}_B \text{Prox}_{\lambda^{-1}}(-\lambda^{-1} A^T z) \right\}, \tag{18}
\]
where \(\hat{\partial}_B \text{Prox}_{\lambda^{-1}}\) denotes the generalized Jacobian of \(\text{Prox}_{\lambda^{-1}}\). If \(\varphi(\cdot) = \mu \|\cdot\|_1\) or \(\varphi(\cdot) = \mathbb{I}_{\mathbb{R}_+^n}(\cdot)\), the generalized Jacobian \(\hat{\partial}_B \text{Prox}_{\lambda^{-1}}\) is exactly the B-subdifferential given by (7) or (8). If \(\varphi(\cdot) = \mu \|\cdot\|_1 + \mathbb{I}_{\mathbb{R}_+^n}(\cdot)\), the generalized Jacobian \(\hat{\partial}_B \text{Prox}_{\lambda^{-1}}\) is defined by (10).

In order to design an implementable semismooth Newton method for solving linear system (17), we introduce the following proposition which shows that \(\hat{\partial}_B(\nabla h)\) defined by (18) can be used as a surrogate generalized Jacobian of \(\nabla h\). Moreover, the following result also plays an important role in establishing the convergence results of the dual based semismooth Newton method (Algorithm 1) for solving problem (1).

**Proposition 3.1.** The set-valued function \(\hat{\partial}_B(\nabla h)\) defined by (18) is a nonempty compact valued upper-semicontinuous multi-function and for any \(V \in \hat{\partial}_B(\nabla h)(z)\), \(V\) is a symmetric positive definite matrix and for \(z' \to z\), it holds that
\[
\nabla h(z') - \nabla h(z) - V(z' - z) = O(\|z' - z\|^2), \quad \forall V \in \hat{\partial}_B(\nabla h(z')). \tag{19}
\]

**Proof.** From the properties of the Moreau-Yosida regularization, we know that the function \(h\) defined in (15) is continuously differentiable with Lipschitz continuous gradient. This together with Lemma 2.3, Lemma 2.4 and [17, Proposition 7] implies equation (19) holds. The proof is completed. \(\square\)
Now, we are ready to describe the dual based semismooth Newton (DSSN) method, which is essentially the semismooth Newton method for solving (17).

**Algorithm 1** Dual Based Semismooth Newton (DSSN) Method for Solving (1)

Given \( \varrho \in (0, 1/2), \bar{\eta} \in (0, 1), \zeta \in (0, 1], \) and \( \beta \in (0, 1). \) Choose \( z^0 \in \mathbb{R}^m. \) For \( j = 0, 1, \ldots, \) perform the following steps in each iteration.

**Step 1.** (Newton direction) Choose one specific matrix \( V_j \in \hat{\partial} (\nabla h)(z^j). \) Specifically, we choose an element \( W_j \in \partial B \text{Prox}_{\lambda - 1} \varphi - (\lambda - 1)A^T z^j \) and then \( V_j := I + (\lambda - 1)A W_j A^T \in \hat{\partial} (\nabla h)(z^j). \)

Solve the following linear system
\[
V_j d = -\nabla h(z^j) \quad (20)
\]
exactly or by the conjugate gradient (CG) algorithm to find \( d \) such that \( \|V_j d + \nabla h(z^j)\| \leq \min(\bar{\eta}, \| \nabla h(z^j) \|^1 + \zeta). \)

**Step 2.** (Line search) Set \( \alpha_j = \beta^l_j, \) where \( l_j \) is the smallest nonnegative integer \( l \) for which
\[
h(z^j + \beta^l d^j) \leq h(z^j) + \varrho \beta^l \langle \nabla h(z^j), d^j \rangle. \quad (21)
\]

**Step 3.** Set \( z^{j+1} = z^j + \alpha_j d^j \) and \( x^{j+1} = \text{Prox}_{\lambda - 1} \varphi - (\lambda - 1)A^T z^{j+1}. \)

The following theorem provides the global convergence of the sequence \( \{(z^j, x^j)\}, \) the local superlinear convergence rate of \( \{z^j\}, \) and the local R-superlinear convergence rate of \( \{x^j\}. \)

**Theorem 3.1.** Let \( \{(z^j, x^j)\} \) be the sequence generated by Algorithm 1. Then the following hold:

(a) The sequence \( \{z^j\} \) is well-defined and converges globally to the unique solution \( z^* \) of (17).

Moreover, the convergence rate is at least superlinear:
\[
\|z^{j+1} - z^*\| = O(\|z^j - z^*\|^1 + \zeta), \quad (22)
\]
where \( \zeta \in (0, 1] \) is the parameter given in Algorithm 1.

(b) The sequence \( \{x^j\} \) converges globally to the unique solution \( x^* \) of problem (1) with at least \( R \)-superlinear convergence rate.

**Proof.** Since the function \( h \) is strongly convex, we can obtain result (a) from Proposition 3.1 by following the same routine in the proof of [17, Theorem 3]. Now, we come to result (b). It holds from the definition of \( x^j \) and (16) that
\[
\|x^{j+1} - x^*\| = \|\text{Prox}_{\lambda - 1} \varphi - (\lambda - 1)A^T z^{j+1} - \text{Prox}_{\lambda - 1} \varphi - (\lambda - 1)A^T z^*\|
\leq \lambda^{-1} \sigma_{\max}(A) \|z^{j+1} - z^*\|,
\]
where $\sigma_{\text{max}}(A)$ is the largest singular value of $A$. This together with Result (a) and (22) implies that the sequence $\{x^j\}$ converges globally to the unique solution $x^*$ and there exists a positive scaler $\kappa$ such that

$$
\|x^{j+1} - x^*\| \leq \kappa\|(z^j - z^*)^{1+\zeta}\|.
$$

Consequently, for sufficient large $j$, there exists $\kappa' \in (0, 1)$ such that

$$
\|x^{j+1} - x^*\| \leq \kappa'\|z^j - z^*\|.
$$

This completes the proof. \hfill \Box

**Remark 3.1.** We mentioned in the introduction that solving the sparse Tikhonov regularization problem can be viewed as solving a subproblem of the SSNAL. Here, we give some comments on this point by taking the elastic net Lasso (2) for example. Observe that, for the elastic net Lasso, problem (15) can be viewed as a subproblem corresponding to each iteration of SSNAL which was proposed in [18]. Specifically, from the discussions presented in [24, Section 4], we know that the outer iteration scheme of SSNAL is equivalent to the proximal point algorithm (PPA) described as follows: given positive scalars $\sigma_k \uparrow \sigma_\infty \leq \infty$, the $k$-th iteration of PPA for Lasso problem is given by

$$
x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|^2 + \mu\|x\|_1 + \frac{1}{2\sigma_k}\|x - x^k\|^2.
$$

Therefore, the optimization problem corresponding to the $k$-th iteration of PPA can reduce to the elastic net model when $x^k = 0$ and $\sigma_k = \lambda^{-1}$.

### 3.1 Techniques for Linear System

The linear system (20) in Algorithm 1 is the most time-consuming part. We take the nonnegative elastic net (3) for example, i.e.,

$$
\varphi(x) = \mu\|x\|_1 + \mathbb{I}_{\mathbb{R}_+^n}(x).
$$

In this case, for any given $\tilde{z} \in \mathbb{R}^n$, the element $\Theta \in \partial \text{Prox}_\varphi(\tilde{z})$ is chosen as

$$
\Theta = \Theta_1\Theta_2,
$$

where $\Theta_1$ and $\Theta_2$ are given by (12) and (13), respectively. Since both $\Theta_1$ and $\Theta_2$ are diagonal matrices, the matrix $\Theta$ is a diagonal matrix, i.e. $\Theta := \text{Diag}(\theta)$. Then, we define

$$
\mathcal{I}(\tilde{z}) := \{i : \theta_i = 1\}, \quad \mathcal{I}_0(\tilde{z}) := \{i : \theta_i = 0\},
$$

where $\tilde{z} := -\lambda^{-1}A^Tz$. Therefore, the linear system (20) can be simplified as follow:

$$
\left( I_m + \lambda^{-1}A_{\mathcal{I}(\tilde{z})}A_{\mathcal{I}(\tilde{z})}^T \right) d = -\nabla h(z),
$$

where $A_{\mathcal{I}(\tilde{z})} \in [\mathbb{R}^{m \times |\mathcal{I}(\tilde{z})|}]$. 

8
Since the coefficient matrix of equation (20) is symmetric and positive definite, there are many solution methods for finding the solution of the linear system, see e.g., [11, 15]. Next, we recall some techniques that can be used to increase the computation efficiency for finding the solution of (25).

- If \( m << |\mathcal{I}(\hat{z})| \) and \( m \) is moderate, the linear equations (25) can be solved by using Cholesky factorization [11, Theorem 4.2.7] of the coefficient matrix.

- If \( |\mathcal{I}(\hat{z})| \ll m \) and \( |\mathcal{I}(\hat{z})| \) is moderate, the Sherman-Morrison-Woodbury formula [11, Page 50] can be used to reduce the computational efforts. Specifically,
  \[
  \left( I_m + \lambda^{-1} A_{\mathcal{I}(\hat{z})} A^T_{\mathcal{I}(\hat{z})} \right)^{-1} = I_m - \lambda^{-1} A_{\mathcal{I}(\hat{z})} (I + \lambda^{-1} A^T_{\mathcal{I}(\hat{z})} A_{\mathcal{I}(\hat{z})})^{-1} A^T_{\mathcal{I}(\hat{z})}.
  \]

Therefore, the computation cost of solving linear system (25) can be reduced by applying Cholesky factorization of the matrix \( I + \lambda^{-1} A^T_{\mathcal{I}(\hat{z})} A_{\mathcal{I}(\hat{z})} \in S^{\mathcal{I}(\hat{z})} \).

- If both \( |\mathcal{I}(\hat{z})| \) and \( m \) are large, the practical conjugated gradient algorithm [11, Algorithm 10.2.1] can be applied to solve the linear system (25) approximately.

4 Numerical Experiments

In this section, we compare the numerical performance of the dual based semismooth Newton method (DSSN) and the primal based semismooth Newton method (PSSN) on the instances from UCI data repository [19]. In the numerical experiments, the data sets \( mgp7, pyrim5 \), and \( bodyfat7 \) are the expended data sets by using the polynomial basis function. For the details of data processing, we refer to [14, 18].

The first order optimal condition of problem (1) can be written as

\[
\Psi(x) := x - \text{Prox}_\varphi (x - \lambda x - A^T (Ax - b)) = 0.
\]

Based on the above condition, we measure the accuracy of the approximate solution of problem (1) by using the following residual:

\[
\eta = \|x - \text{Prox}_\varphi (x - \lambda x - A^T (Ax - b))\|.
\]

For a given tolerance \( \varepsilon = 10^{-6} \), all the tested algorithms will be stopped when \( \eta \leq \varepsilon \) or the maximum number 200 of iterations is reached. Inspired by the parameter used in [35], we choose \( \lambda \in \{100, 10, 1, 0.1, 0.01\} \). For testing purpose, the parameter \( \mu \) is chosen as

\[
\mu = \mu_c \|A^T b\|_\infty.
\]

All our numerical results are obtained by running MATLAB R2018b on a desktop (4-core, i5-7300 CPU @2.60GHz, 8.00GB of RAM).
4.1 Primal based Semismooth Newton Method

Note that $\Psi(x)$ defined by (26) is a locally Lipschitz continuous function and one can employ the following semismooth Newton method (SSN):

$$x^{k+1} = x^k - V_k^{-1}(x^k),$$

where $V_k \in \partial_B \Psi(x^k)$ with the mapping $\partial_B \Psi : \mathbb{R}^n \to \mathbb{S}^n$ being defined by

$$\partial_B \Psi(\bar{x}) := \left\{ I - \Theta((1 - \lambda)I - A^TA) \in \mathbb{S}^n : \Theta \in \partial_B \text{Prox}_\varphi(\bar{x}) \right\}, \quad \forall \bar{x} \in \mathbb{R}^n.$$

Inspired by the smoothing Newton method (see e.g., [7, 26]) and the solution method used in [34], we define the merit function

$$r(x) = \|\Psi(x)\|^2. \quad (27)$$

Therefore, the primal based semismooth Newton method for solving problem (1) can be described by Algorithm 2.

Algorithm 2 Primal based Semismooth Newton (PSSN) Method for Solving (1)

Given $\varrho \in (0, 1/2)$, and $\beta \in (0, 1)$. Choose $y^0 \in \mathbb{R}^m$. For $k = 0, 1, \ldots$, perform the following steps in each iteration.

**Step 1.** (Newton direction) Choose $\Theta \in \partial_B \text{Prox}_\varphi(y)$ with $y = x^j - \lambda x^j - A^T(Ax^j - b)$. Compute $d^j$ by solving

$$(I - \Theta((1 - \lambda)I - A^TA))d = -\Psi(x^k). \quad (28)$$

**Step 2.** (Line search) Set $\alpha_j = \beta^l_j$, where $l_j$ is the smallest nonnegative integer $l$ for which

$$r(x^j + \beta^l_j d^j) \leq r(x^j) + \varrho \beta^l_j \langle \nabla r(x^j), d^j \rangle.$$

**Step 3.** Set $x^{j+1} = x^j + \alpha_j d^j$.

Similar to Algorithm 1, the linear system (28) in Algorithm 2 is the most time-consuming part. We also take the nonnegative elastic net for example. In this case, for any given $y \in \mathbb{R}^n$, the element $\Theta \in \partial_B \text{Prox}_\varphi(y)$ is chosen as (23). Define

$$\mathcal{I}(y) := \{i : \theta_i = 1\}, \quad \mathcal{I}_0(y) := \{i : \theta_i = 0\}, \quad (29)$$

where

$$y = x - \lambda x - A^T(Ax - b).$$

Therefore, the linear system (28) can be simplified as follow:

$$d_{\mathcal{I}_0(y)} = -\Psi_{\mathcal{I}_0(y)}(x^k),$$

$$(\lambda I_{\mathcal{I}(y)} + A_{\mathcal{I}(y)}^TA_{\mathcal{I}(y)})d_{\mathcal{I}(y)} = -\Psi_{\mathcal{I}(y)}(x^k) - A_{\mathcal{I}(y)}^TA_{\mathcal{I}_0(y)}d_{\mathcal{I}_0(y)}. \quad (30)$$
Note that the linear system (30) can be solved by using the similar techniques presented in subsection 3.1.

Remark 4.1. It follows from [23, Theorem 3.2] that the PSSN method without line search converges locally quadratically to the unique solution of problem (1). The numerical experience shows that the line search step (Step 2 in Algorithm 2) can increase the convergence speed. However, since the smoothness of the merit function \( r(\cdot) \) defined by (27) is unclear, it is still a challenge problem that whether the PSSN method can converge globally [8, Theorem 8.3.15].

4.2 Elastic Net Lasso

Motivated by [29], the elastic net Lasso problem (2) can be reformulated into the following form which is exactly the Lasso problem [28]:

\[
\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{2} \| \tilde{A} x - \tilde{b} \|^2 + \mu \| x \|_1,
\]

where \( \tilde{A} \) and \( \tilde{b} \) are given by (5). Therefore, the semismooth Newton based augmented Lagrangian method (SSNAL) proposed by [18] can be used to solve problem (4). The high efficiency of SSNAL\(^1\) for solving the Lasso problem has been convincingly demonstrated by various data sets by [18]. Therefore, we take the performance of SSNAL for solving the (31) as a benchmark.

In this section, we present the performance of the DSSN, PSSN, and SSSNAL for solving the elastic net problem (2) on the UCI datasets that mentioned at the beginning of this section. In order to test the robustness of algorithms with respect to the parameter \( \lambda \), for each \( \mu_c \), we choose a sequence of parameters \( \{ \lambda_k \} \) which run from 1 to 0.1 and solve a sequence of optimization problems by using PSSN, DSSN and SSSNAL. It is worth mentioning that, for each \( \mu_c \), the optimal solution of optimization problem with \( \lambda_k \) is taken as the initial point of the problem with \( \lambda_{k+1} \). These results are reported in Table 1. In Table 2, we select a grid of values of \( \lambda \in \{ 100, 10, 1, 0.1, 0.01 \} \) and \( \mu = k \mu_c(2) \) (here, \( \mu_c(2) \) is the second \( \mu_c \) that used in Table 1 for each data set) with \( k = 100 : -1 : 1 \).

In this table, for each data set, we only need to choose the initial point once and take optimal solution of the \( l \)-th problem (say \( x^{l*} \)) as the initial point of the next optimization problem.

From Tables 1 and 2, we can see that the convergence speed of DSSN of all the test examples are superior to that of PSSN. In order to obtain the index sets (29), which can be used to reduce the computational effort, the \( u = Ax \) and \( A^T u \) have to be computed. The computation of \( A^T A u \) is at the cost of \( O(mn^2) \). However, for the DSSN, the main cost is in computing \( A^T z \) at \( O(mn) \). For most cases, especially for the higher dimensional problems, DSSN is also outperform the SSSNAL for solving problem (31). The most possible reason is the formulation (31) overlooks the advantage of the Tikhonov regularization in designing numerical algorithm and destroys the low-sample size and high-dimensional structure of data matrix \( A \).

Furthermore, from the convergence result Theorem 3.1, we know that a well-chosen initial point can greatly speed up the convergence. Based on the results presented in Table 2, two instances are

\(^1\)Matlab code is available at: http://www.math.nus.edu.sg/~mattohkc/SuiteLasso.html
chosen to illustrate that the DSSN has the advantage over PSSN. Figure 1 shows that the solution gap between the two neighbouring dual optimization problems and the iteration numbers have a similar performance. Compare to the performance of PSSN, both the solution gap and the iteration numbers are relatively small and stable, especially from the 80-th to the 100-th optimization problems.

Table 1: The performances of DSSN(D), PSSN(P) and SNNAL(L) on selected instances from UCI data repository. λ = 1 : −0.1 : 0.1

| problem name | $\mu_c$ | mzn | time | error |
|--------------|---------|------|------|-------|
|              | (m, n)  | (max,min) | $D \mid P \mid L$ | $D \mid P \mid L$ |
| Leukaemia    | 1e-02   | (104, 39) | 0.21 | 0.41 | 5.42 | 2.23e-13 | 1.21e-13 | 5.31e-07 |
|              | 1e-03   | (422, 105) | 0.16 | 0.38 | 5.34 | 1.13e-13 | 1.69e-13 | 4.30e-07 |
|              | 1e-04   | (1924, 423) | 0.24 | 0.44 | 5.10 | 3.01e-13 | 2.26e-08 | 4.36e-07 |
| duke         | 1e-02   | (100, 48) | 0.11 | 0.36 | 5.83 | 3.29e-13 | 6.62e-14 | 5.82e-07 |
|              | 1e-03   | (387, 102) | 0.14 | 0.42 | 5.56 | 5.76e-13 | 1.15e-13 | 5.09e-07 |
|              | 1e-04   | (1872, 388) | 0.28 | 0.51 | 6.04 | 4.26e-12 | 3.00e-08 | 5.89e-07 |
| colon-cancer | 1e-02   | (73, 63) | 0.06 | 0.20 | 2.57 | 2.74e-12 | 8.46e-14 | 4.65e-07 |
|              | 1e-03   | (317, 82) | 0.11 | 0.19 | 3.59 | 6.55e-12 | 1.34e-13 | 4.86e-07 |
|              | 1e-04   | (1086, 318) | 0.24 | 0.26 | 3.19 | 3.81e-11 | 9.15e-13 | 4.56e-07 |
| mpg7         | 1e-02   | (15, 14) | 0.16 | 0.75 | 1.95 | 1.00e-08 | 1.23e-12 | 2.43e-07 |
|              | 1e-03   | (78, 51) | 0.17 | 0.57 | 7.26 | 4.86e-10 | 3.61e-12 | 6.12e-07 |
|              | 1e-04   | (516, 214) | 0.54 | 0.96 | 11.68 | 1.29e-10 | 7.82e-12 | 5.85e-07 |
| pyrim5       | 1e-03   | (422, 166) | 3.41 | 18.24 | 152.50 | 2.19e-12 | 1.95e-08 | 5.16e-07 |
|              | 1e-04   | (2677, 610) | 3.62 | 10.43 | 270.93 | 9.45e-12 | 1.09e-07 | 6.58e-07 |
|              | 1e-05   | (16306, 2803) | 5.14 | 9.35 | 424.01 | 6.28e-11 | 8.49e-08 | 5.31e-07 |
| bodyfat7     | 1e-03   | (12, 3) | 2.39 | 9.02 | 28.45 | 3.10e-12 | 8.99e-14 | 4.85e-07 |
|              | 1e-04   | (196, 16) | 2.89 | 11.10 | 75.92 | 1.30e-13 | 9.75e-14 | 4.96e-07 |
|              | 1e-05   | (2734, 206) | 3.19 | 12.90 | 129.51 | 2.95e-13 | 3.57e-08 | 5.80e-07 |
| triazines4   | 1e-01   | (321, 321) | 76.34 | 478.34 | 25.39 | 1.62e-09 | 6.26e-10 | 1.01e-09 |
|              | 1e-02   | (416, 387) | 67.53 | 314.15 | 748.72 | 3.85e-09 | 2.89e-09 | 5.19e-07 |
|              | 1e-03   | (1535, 853) | 61.10 | 340.99 | 2630.55 | 2.85e-09 | 2.61e-09 | 6.54e-07 |
Table 2: The performances of DSSN(D), PSSN(P) and SSNAL(L) on selected instances from UCI data repository.

| problem name       | λ      | nnz      | time     | error   |
|--------------------|--------|----------|----------|---------|
|                    | (m, n) | (max, min) | D | P | L | D | P | L |
| Leukaemia          | 1e+02  | (6685, 1813) | 3.49 | 4.82 | 0.38 | 2.21e-09 | 1.29e-08 | 6.44e-07 |
| (38,7129)          | 1e+01  | (5336, 413)  | 1.64 | 3.85 | 0.72 | 2.56e-13 | 1.10e-08 | 4.45e-07 |
|                    | 1e+00  | (1924, 104)  | 0.83 | 3.14 | 1.13 | 1.28e-13 | 1.08e-09 | 5.76e-07 |
|                    | 1e-01  | (423, 38)    | 0.80 | 3.59 | 1.66 | 2.65e-13 | 8.62e-10 | 6.54e-07 |
|                    | 1e-02  | (105, 34)    | 1.76 | 9.22 | 2.49 | 3.30e-12 | 1.34e-13 | 6.47e-07 |
| duke               | 1e+02  | (6624, 1791) | 4.55 | 5.70 | 2.53 | 7.71e-13 | 2.61e-08 | 6.67e-07 |
| (44,7129)          | 1e+01  | (5215, 381)  | 1.96 | 3.95 | 2.61 | 3.65e-13 | 1.92e-08 | 7.47e-07 |
|                    | 1e+00  | (1872, 100)  | 1.01 | 3.61 | 2.68 | 2.90e-13 | 8.08e-09 | 6.99e-07 |
|                    | 1e-01  | (388, 47)    | 0.86 | 4.16 | 2.77 | 7.86e-13 | 2.11e-13 | 6.32e-07 |
|                    | 1e-02  | (102, 44)    | 1.66 | 8.93 | 2.87 | 5.96e-12 | 9.21e-13 | 5.96e-07 |
| colon-cancer       | 1e+02  | (1918, 988)  | 2.83 | 3.43 | 2.67 | 5.54e-13 | 5.16e-10 | 4.93e-07 |
| (62,2000)          | 1e+01  | (1792, 288)  | 1.59 | 1.95 | 2.47 | 1.35e-12 | 2.76e-08 | 5.30e-07 |
|                    | 1e+00  | (1086, 73)   | 0.83 | 1.39 | 2.24 | 3.04e-12 | 3.90e-09 | 5.75e-07 |
|                    | 1e-01  | (318, 63)    | 0.72 | 1.42 | 2.01 | 1.08e-11 | 4.11e-09 | 5.02e-07 |
|                    | 1e-02  | (84, 56)     | 1.41 | 2.48 | 1.86 | 8.68e-11 | 6.75e-13 | 5.15e-07 |
| mpg7               | 1e+02  | (2557, 76)   | 1.90 | 4.56 | 2.11 | 7.32e-12 | 5.56e-12 | 5.42e-07 |
| (392,3432)         | 1e+01  | (1234, 20)   | 1.06 | 4.04 | 3.19 | 1.16e-11 | 4.02e-12 | 3.99e-07 |
|                    | 1e+00  | (516, 14)    | 0.93 | 4.08 | 4.40 | 3.38e-10 | 3.08e-12 | 2.50e-07 |
|                    | 1e-01  | (214, 14)    | 1.04 | 4.76 | 6.28 | 1.92e-08 | 2.77e-12 | 3.97e-07 |
|                    | 1e-02  | (142, 13)    | 1.61 | 6.87 | 9.11 | 6.69e-09 | 2.79e-12 | 4.87e-07 |
| pyrim5             | 1e+02  | (143811, 12077) | 69.50 | 121.99 | 44.24 | 8.72e-09 | 1.07e-07 | 5.67e-07 |
| (74,201376)        | 1e+01  | (74869, 2021) | 31.26 | 70.58 | 109.45 | 3.88e-12 | 7.21e-08 | 6.00e-07 |
|                    | 1e+00  | (16306, 422) | 18.93 | 77.04 | 159.03 | 3.81e-12 | 5.30e-08 | 5.92e-07 |
|                    | 1e-01  | (2803, 166)  | 18.26 | 106.93 | 201.44 | 7.33e-12 | 1.83e-08 | 5.52e-07 |
|                    | 1e-02  | (558, 86)    | 25.07 | 402.05 | 274.29 | 2.59e-11 | 3.82e-08 | 4.05e-07 |
| bodyfat7           | 1e+02  | (94185, 3455) | 39.44 | 86.42 | 249.40 | 1.63e-09 | 1.87e-07 | 3.03e-07 |
| (252,116280)       | 1e+01  | (28754, 142) | 18.83 | 70.91 | 198.32 | 6.25e-09 | 4.39e-08 | 4.02e-07 |
|                    | 1e+00  | (2734, 11)   | 12.53 | 58.21 | 163.72 | 2.96e-13 | 7.50e-14 | 4.65e-07 |
|                    | 1e-01  | (206, 3)     | 10.96 | 56.01 | 133.28 | 5.39e-12 | 6.01e-14 | 4.15e-07 |
|                    | 1e-02  | (35, 2)      | 11.79 | 73.53 | 75.97 | 2.44e-10 | 3.52e-14 | 4.44e-07 |
| triazines4         | 1e+02  | (17281, 345) | 145.23 | 826.04 | 410.37 | 5.20e-11 | 4.19e-09 | 4.61e-07 |
| (186,635376)       | 1e+01  | (4050, 321)  | 96.41 | 623.63 | 850.76 | 3.75e-10 | 7.84e-10 | 3.76e-07 |
|                    | 1e+00  | (1535, 321)  | 119.17 | 801.84 | 1535.02 | 3.19e-09 | 1.37e-08 | 3.78e-07 |
|                    | 1e-01  | (853, 321)   | 190.86 | 1703.73 | 2467.18 | 3.34e-08 | 1.19e-07 | 3.95e-07 |
|                    | 1e-02  | (653, 321)   | 573.12 | 6200.19 | 3722.94 | 2.73e-07 | 6.05e-01 | 4.38e-07 |
Figure 1: Number of iterations and solution gap $\|x^l - x^{(l-1)*}\|$, $l \geq 10$, on the Leukaemia and bodyfat7 data sets. $\lambda = 0.01$.

4.3 Nonnegative Elastic Net Lasso

In this section, we present the performance of the DSSN and PSSN for solving the nonnegative elastic net problem (3) on the UCI datasets. These numerical results are presented in Table 3 and Table 4. Both tables show that the convergence speed of DSSN of all the test examples are superior to that of PSSN. Based on the results presented in Table 4, two instances Leukaemia and bodyfat7 are chosen to illustrate that the DSSN has the advantage over PSSN and the illustration is presented by Figure 2. We can observe that the performance of DSSN for solving the nonnegative elastic net Lasso is similar to that of DSSN for solving the elastic net Lasso.

Due to the nonnegative constraints, the package SuiteLasso is can not be applied directly.
Table 3: The performances of DSSN(D) and PSSN(P) on selected instances from UCI data repository. $\lambda = 1 : -0.01 : 0.1$

| problem name     | $\mu$  | nnz (max,min) | time | error   |
|------------------|--------|---------------|------|---------|
|                  |        |               |      | $P \mid D$ | $P \mid D$ |
| (m, n):          | $\mu$  | (m, n):       |      |          |
|                  |        | (m, n):       |      |          |
|                  | 1e-02  | (94, 44)      | 1.03 | 0.15    | 2.69e-14 | 1.39e-13 |
| Leukaemia        | 1e-03  | (369, 97)     | 1.47 | 0.33    | 1.72e-08 | 1.24e-13 |
| (38,7129)        | 1e-04  | (1280, 370)   | 1.58 | 0.64    | 3.76e-08 | 2.60e-13 |
|                  | 1e-02  | (82, 42)      | 1.07 | 0.17    | 2.53e-14 | 1.50e-13 |
| duke             | 1e-03  | (337, 84)     | 1.55 | 0.32    | 9.66e-14 | 4.75e-13 |
| (44,7129)        | 1e-04  | (1196, 339)   | 1.85 | 0.73    | 3.48e-08 | 2.93e-12 |
|                  | 1e-03  | (255, 100)    | 0.47 | 0.31    | 1.27e-13 | 6.40e-12 |
| colon-cancer     | 1e-04  | (716, 288)    | 0.77 | 0.63    | 1.09e-08 | 2.64e-11 |
| (62,2000)        |        |               |      |          |
|                  | 1e-02  | (26, 19)      | 1.61 | 0.24    | 1.36e-12 | 3.15e-10 |
| mpg7             | 1e-03  | (69, 51)      | 1.64 | 0.29    | 2.13e-12 | 9.44e-10 |
| (392,3432)       | 1e-04  | (384, 192)    | 2.91 | 0.96    | 4.88e-12 | 1.16e-10 |
| pyrim5           | 1e-03  | (314, 113)    | 41.74| 5.57    | 9.68e-09 | 1.25e-12 |
| (74,201376)      | 1e-04  | (2161, 433)   | 42.30| 7.95    | 5.80e-08 | 4.95e-12 |
| (74,201376)      | 1e-05  | (12191, 2304) | 47.28| 19.12   | 4.90e-08 | 2.25e-08 |
| bodyfat7         | 1e-03  | (11, 4)       | 33.68| 5.06    | 4.17e-14 | 3.00e-12 |
| (252,116280)     | 1e-04  | (199, 14)     | 54.26| 7.61    | 5.60e-09 | 1.63e-12 |
| (252,116280)     | 1e-05  | (6901, 647)   | 60.62| 15.01   | 6.78e-08 | 7.48e-08 |
| triazines4       | 1e-01  | (269, 269)    | 745.92| 106.51  | 6.43e-12 | 1.28e-09 |
| (186,635376)     | 1e-02  | (439, 409)    | 532.64| 94.63   | 6.60e-09 | 5.66e-10 |
| (186,635376)     | 1e-03  | (1038, 699)   | 487.09| 93.09   | 1.48e-09 | 1.72e-09 |
| problem name | $\lambda$ | $\text{nnz}$ | $\text{time}$ | $\text{error}$ |
|--------------|-------------|-------------|--------------|--------------|
| Leukaemia (38,7129) | 1e+02 | (2411, 1221) | 2.79 | 1.40 | 2.25e-09 | 4.96e-09 |
| | 1e+01 | (2185, 358) | 2.73 | 0.87 | 1.18e-08 | 2.05e-13 |
| | 1e+00 | (1280, 94) | 2.96 | 0.55 | 1.73e-08 | 1.24e-13 |
| | 1e-01 | (370, 44) | 3.53 | 0.46 | 3.23e-13 | 3.29e-13 |
| | 1e-02 | (97, 39) | 6.91 | 0.79 | 7.93e-09 | 2.67e-12 |
| duke (44,7129) | 1e+02 | (2151, 1131) | 3.28 | 1.72 | 7.41e-09 | 3.59e-13 |
| | 1e+01 | (1972, 326) | 3.09 | 1.11 | 2.56e-08 | 2.30e-13 |
| | 1e+00 | (1196, 82) | 3.61 | 0.68 | 7.70e-09 | 2.47e-13 |
| | 1e-01 | (339, 42) | 3.72 | 0.54 | 3.81e-09 | 7.48e-12 |
| | 1e-02 | (85, 39) | 5.85 | 0.83 | 1.19e-10 | 9.12e-12 |
| colon-cancer (62,2000) | 1e+02 | (962, 640) | 1.12 | 1.02 | 1.58e-08 | 3.00e-13 |
| | 1e+01 | (920, 254) | 1.00 | 0.78 | 1.57e-08 | 8.66e-13 |
| | 1e+00 | (716, 92) | 0.89 | 0.49 | 1.08e-13 | 2.62e-12 |
| | 1e-01 | (288, 59) | 1.02 | 0.46 | 2.20e-13 | 9.61e-12 |
| | 1e-02 | (102, 55) | 1.97 | 0.85 | 2.84e-13 | 7.32e-11 |
| mpg7 (392,3432) | 1e+02 | (1574, 118) | 5.43 | 1.95 | 4.64e-12 | 5.70e-12 |
| | 1e+01 | (914, 44) | 4.04 | 0.88 | 4.03e-12 | 1.58e-11 |
| | 1e+00 | (384, 24) | 4.64 | 0.83 | 3.62e-12 | 3.54e-10 |
| | 1e-01 | (192, 18) | 5.93 | 1.06 | 3.34e-12 | 6.39e-10 |
| | 1e-02 | (138, 18) | 6.95 | 1.23 | 3.28e-12 | 9.39e-09 |
| pyrim5 (74,201376) | 1e+02 | (51298, 9319) | 78.46 | 52.21 | 8.13e-08 | 2.62e-10 |
| | 1e+01 | (38448, 1613) | 69.98 | 22.74 | 5.30e-08 | 9.78e-09 |
| | 1e+00 | (12191, 314) | 76.34 | 15.20 | 3.26e-08 | 8.37e-10 |
| | 1e-01 | (2304, 113) | 104.40 | 14.92 | 1.41e-08 | 4.69e-12 |
| | 1e-02 | (432, 71) | 328.09 | 22.35 | 2.81e-08 | 1.49e-11 |
| bodyfat7 (252,116280) | 1e+02 | (32516, 1663) | 79.46 | 30.06 | 2.56e-07 | 1.44e-08 |
| | 1e+01 | (21871, 40) | 72.36 | 16.28 | 5.42e-08 | 4.87e-09 |
| | 1e+00 | (6901, 11) | 55.50 | 10.44 | 2.40e-09 | 6.37e-11 |
| | 1e-01 | (647, 4) | 57.23 | 9.67 | 1.51e-09 | 1.81e-11 |
| | 1e-02 | (79, 4) | 92.08 | 12.05 | 5.20e-14 | 8.07e-10 |
| triazines4 (186,635376) | 1e+02 | (10693, 365) | 953.57 | 134.71 | 9.17e-09 | 4.84e-11 |
| | 1e+01 | (2820, 295) | 764.55 | 90.65 | 5.90e-10 | 3.60e-10 |
| | 1e+00 | (1038, 269) | 930.33 | 111.89 | 5.33e-09 | 2.85e-09 |
| | 1e-01 | (699, 269) | 1399.66 | 170.56 | 1.22e-07 | 3.19e-08 |
| | 1e-02 | (619, 269) | 6197.73 | 442.13 | 3.46e-01 | 2.58e-07 |
5 Conclusion

In this paper, we have proposed a dual based semismooth Newton (DSSN) method for solving the sparse Tikhonov regularization. By taking advantage the Tikhonov regularization, the DSSN method can not only avoid increasing the scale of data matrix, but also can overcome the drawbacks of the primal semismooth Newton method (PSSN). In other words, the DSSN method is globally convergent and can achieve at least R-superlinearly convergence rate without reconstructing the data matrix. The numerical efficiency and stability have also been demonstrated by comparing with the PSSN method and S-NAL on high-dimensional the UCI data sets. Finally, it is worth noting that this paper focuses on a class of simple Tikhonov regularization, i.e., the Tikhonov matrix is taken as $\alpha I$ ($\alpha > 0$). We leave the research topic that how to generalize DSSN to general Tikhonov matrix as our further work.

References

[1] Z.-Z. Bai, A. Buccini, K. Hayami, L. Reichel, J.-F. Yin, and N. Zheng. Modulus-based iterative methods for constrained Tikhonov regularization. *Journal of Computational and Applied Mathematics*. 2020, vol. 378, pp. 127-142.
Mathematics, 319:1–13, 2017.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[3] X. Chen, A. Y. Aravkin, and R. D. Martin. Generalized linear model for gamma distributed variables via elastic net regularization. *arXiv preprint arXiv:1804.07780*, 2018.

[4] F. H. Clarke. *Optimization and Nonsmooth Analysis*. SIAM, 1990.

[5] A. Daducci, E. J. Canales-Rodríguez, H. Zhang, T. B. Dyrby, D. C. Alexander, and J.-P. Thiran. Accelerated microstructure imaging via convex optimization (AMICO) from diffusion MRI data. *NeuroImage*, 105:32–44, 2015.

[6] M. Demirer, F. X. Diebold, L. Liu, and K. Yilmaz. Estimating global bank network connectedness. *Journal of Applied Econometrics*, 33(1):1–15, 2018.

[7] F. Facchinei and J.-S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Science & Business Media, 2007.

[8] F. Facchinei and J.-S. Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[9] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

[10] Y. Gao and D. F. Sun. Calibrating least squares covariance matrix problems with equality and inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1432–1457, 2009.

[11] G. H. Golub and C. F. Van Loan. *Matrix Computations (3rd ed.*), volume 3. John Hopkins University Press, 1996.

[12] D. Han, D. F. Sun, and L. Zhang. Linear rate convergence of the alternating direction method of multipliers for convex composite programming. *Mathematics of Operations Research*, 43(2):622–637, 2017.

[13] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I: Fundamentals*, volume 305. Springer Science & Business Media, 2013.

[14] L. Huang, J. Jia, B. Yu, B.-G. Chun, P. Maniatis, and M. Naik. Predicting execution time of computer programs using sparse polynomial regression. In *Advances in Neural Information Processing Systems*, pages 883–891, 2010.

[15] X. Y. Lam, J. Marron, D. F. Sun, and K.-C. Toh. Fast algorithms for large-scale generalized distance weighted discrimination. *Journal of Computational and Graphical Statistics*, 27(2):368–379, 2018.
[16] C. Lemaréchal and C. Sagastizábal. Practical aspects of the Moreau–Yosida regularization: Theoretical preliminaries. *SIAM Journal on Optimization*, 7(2):367–385, 1997.

[17] X. Li, D. Sun, and K.-C. Toh. On efficiently solving the subproblems of a level-set method for fused lasso problems. *SIAM Journal on Optimization*, 28(2):1842–1866, 2018.

[18] X. Li, D. F. Sun, and K.-C. Toh. A highly efficient semismooth Newton augmented Lagrangian method for solving lasso problems. *SIAM Journal on Optimization*, 28(2):1842–1866, 2018.

[19] M. Lichman. UCI machine learning repository. *School of Information and Computer Sciences, University of California, Irvine*, 2013.

[20] J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93(2):273–299, 1965.

[21] H. Qi and D. F. Sun. A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, 28(2):360–385, 2006.

[22] L. Qi. Convergence analysis of some algorithms for solving nonsmooth equations. *Mathematics of Operations Research*, 18(1):227–244, 1993.

[23] L. Qi and J. Sun. A nonsmooth version of Newton’s method. *Mathematical Programming*, 58(1):353–367, 1993.

[24] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976.

[25] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 2015.

[26] D. F. Sun and L. Qi. Solving variational inequality problems via smoothing-nonsmooth reformulations. *Journal of Computational and Applied Mathematics*, 129(1-2):37–62, 2001.

[27] S. J. Teipel et al. Robust detection of impaired resting state functional connectivity networks in Alzheimer’s disease using elastic net regularized regression. *Frontiers in Aging Neuroscience*, 8:318, 2017.

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

[29] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):245–266, 2012.

[30] L. Wu and Y. Yang. Nonnegative elastic net and application in index tracking. *Applied Mathematics and Computation*, 227:541–552, 2014.

[31] K. Yosida. *Functional Analysis*. Springer, Berlin, 1964.
[32] Y.-L. Yu. On decomposing the proximal map. In Advances in Neural Information Processing Systems, pages 91–99, 2013.

[33] Y. Zhang, N. Zhang, D. F. Sun, and K.-C. Toh. An efficient Hessian based algorithm for solving large-scale sparse group Lasso problems. Mathematical Programming, 179(1-2):223–263, 2020.

[34] H.-J. Zhao and H. Yang. Semismooth Newton methods with domain decomposition for American options. Journal of Computational and Applied Mathematics, 337:37–50, 2018.

[35] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2):301–320, 2005.