A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems[*]

Xudong Li[†] Defeng Sun[‡] and Kim-Chuan Toh[§]

October 6, 2016

Abstract

We develop a fast and robust algorithm for solving large scale convex composite optimization models with an emphasis on the ℓ_1-regularized least squares regression (Lasso) problems. Despite the fact that there exist a large number of solvers in the literature for the Lasso problems, we found that no solver can efficiently handle difficult large scale regression problems with real data. By leveraging on available error bound results to realize the asymptotic superlinear convergence property of the augmented Lagrangian algorithm, and by exploiting the second order sparsity of the problem through the semismooth Newton method, we are able to propose an algorithm, called Ssnal, to efficiently solve the aforementioned difficult problems. Under very mild conditions, which hold automatically for Lasso problems, both the primal and the dual iteration sequences generated by Ssnal possess a remarkably fast linear convergence rate, which can even be made to be superlinear asymptotically. Numerical comparisons between our approach and a number of state-of-the-art solvers, on real data sets, are presented to demonstrate the high efficiency and robustness of our proposed algorithm in solving difficult large scale Lasso problems.

Keywords: Lasso, sparse optimization, augmented Lagrangian, metric subregularity, semismoothness, Newton's method

AMS subject classifications: 65F10, 90C06, 90C25, 90C31

1 Introduction

In this paper, we aim to design a highly efficient and robust algorithm for solving convex composite optimization problems including the following ℓ_1-regularized least squares (LS) problem

$$\min\left\{\frac{1}{2}\|\mathcal{A}x-b\|^2+\lambda\|x\|_1\right\}, \tag{1}$$

[*] An earlier version of this paper was made available in arXiv as arXiv:1607.05428. This research is supported in part by the Ministry of Education, Singapore, Academic Research Fund under Grant R-146-000-194-112.

[†] Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore (`matlixu@nus.edu.sg`).

[‡] Department of Mathematics and Risk Management Institute, National University of Singapore, 10 Lower Kent Ridge Road, Singapore (`matsundf@nus.edu.sg`).

[§] Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore (`mattohkc@nus.edu.sg`).

where $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is a linear map whose adjoint is denoted as $\mathcal{A}^*$, $b \in \mathcal{Y}$ and $\lambda > 0$ are given data, and $\mathcal{X}$, $\mathcal{Y}$ are two real finite dimensional Euclidean spaces each equipped with an inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\| \cdot \|$.

With the advent of convenient automated data collection technologies, the Big Data era brings new challenges in analyzing massive data due to the inherent sizes – large samples and high dimensionality [14]. In order to respond to these challenges, researchers have developed many new statistical tools to analyze such data. Among these, the ℓ_1-regularized models are arguably the most intensively studied. They are used in many applications, such as in compressive sensing, high-dimensional variable selection and image reconstruction, etc. Most notably, the model (1), named as Lasso, was proposed in [46] and has been used heavily in high-dimensional statistics and machine learning. The model (1) has also been studied in the signal processing context under the name of basis pursuit denoising [9]. In addition to its own importance in statistics and machine learning, Lasso problem (1) also appears as an inner subproblem of many important algorithms. For example, in a recent paper [1], a level set method was proposed to solve a computationally more challenging reformulation of the Lasso problem, i.e.,

$$\min \left\{ \|x\|_1 \mid \frac{1}{2}\|\mathcal{A}x - b\|^2 \leq \sigma \right\}.$$

The level set method relies critically on the assumption that the following optimization problem, the same type as (1),

$$\min \left\{ \frac{1}{2}\|\mathcal{A}x - b\|^2 + \delta_{B_\lambda}(x) \right\}, \quad B_\lambda = \{x \mid \| \cdot \|_1 \leq \lambda\},$$

can be efficiently solved to high accuracy. The Lasso-type optimization problems also appear as subproblems in various proximal Newton methods for solving convex composite optimization problems [7, 28, 51]. Notably, in a broader sense, all these proximal Newton methods belong to the class of algorithms studied in [16].

The above mentioned importance together with a wide range of applications of (1) has inspired many researchers to develop various algorithms for solving this problem and its equivalent reformulations. These algorithms can roughly be divided into two categories. The first category consists of algorithms that use only the gradient information, for example, accelerated proximal gradient (APG) method [36, 2], GPSR [15], SPGL1 [4], SpaRSA [48], FPC_AS [47], and NESTA [3], to name only a few. Meanwhile, algorithms in the second category, including but not limited to mfIPM [17], SNF [35], BAS [6], SQA [7], OBA [25], FBS-Newton [49], utilize the second order information of the underlying problem in the algorithmic design to accelerate the convergence. For nondegenerate problems, the solvers in the second category work quite well and usually outperform the algorithms in the first category when high accuracy solutions are sought. One exception is the inexact interior-point-algorithm based solver mfIPM, which does not rely on the nondegeneracy assumption but requires the availability of appropriate cheap pre-conditioners for solving the linear systems in the subproblems. In this paper, we also aim to solve the Lasso problems by making use of the second order information. The novelty of our approach is that we do not need any nondegeneracy assumption in our theory or computations. The core idea is to analyze the fundamental nonsmooth structures in the problems to formulate and solve specific semismooth equations with well conditioned symmetric and positive definite generalized Jacobian matrices, which consequently play a critical role in our algorithmic design. When applied to solve difficult large scale sparse optimization problems, even for degenerate ones, our approach can outperform the first order algorithms by a huge margin regardless of whether low or high accuracy solutions are sought. This is in a sharp contrast to most of the other second order based solvers mentioned above, where their competitive advantages over first-order methods only become apparent when high accuracy solutions are sought.

Our proposed algorithm is a semismooth Newton augmented Lagrangian method, named as Ssnal, for solving the dual of problem (1) where the sparsity property of the second order generalized Hessian is wisely exploited. This algorithmic framework is adapted from those appeared in [52, 24, 50, 30] for solving semidefinite programming problems where impressive numerical results have been reported. Specialized to the vector case, our Ssnal possesses unique features that are not available in the semidefinite programming case. It is these combined unique features that allow our algorithm to converge at a very fast speed with very low computational costs at each step. Indeed, for large scale sparse Lasso problems, our numerical experiments show that the proposed algorithm needs at most a few dozens of outer iterations to reach solutions with the desired accuracy while all the inner semismooth Newton subproblems can be solved very cheaply. One reason for this impressive performance is that the piecewise linear-quadratic structure of the Lasso problem (1) guarantees the asymptotic superlinear convergence of the augmented Lagrangian algorithm. Beyond the piecewise linear-quadratic case, we also study more general functions to guarantee this fast convergence rate. More importantly, since there are several desirable properties including the strong convexity of the objective function in the inner subproblems, in each outer iteration we only need to execute a few (usually one to four) semismooth Newton steps to solve the underlying subproblem. As will be shown later, for Lasso problems with sparse optimal solutions, the computational costs of performing these semismooth Newton steps can be made to be extremely cheap compared to other costs. This seems to be counter-intuitive as normally one would expect a second order method to be computationally much more expensive than the first order methods at each step. Here, we make this remarkable achievement possible by carefully exploiting the second order sparsity in the augmented Lagrangian functions. Notably, our algorithmic framework not only works for models such as Lasso, adaptive Lasso [54] and elastic net [55], but can also be applied to more general convex composite optimization problems. The high performance of our algorithm also serves to show that the second order information, more specifically the nonsmooth second order information, can be and should be incorporated intelligently into the algorithmic design for large scale optimization problems.

The remaining parts of this paper are organized as follows. In the next section, we introduce some definitions and present preliminary results on the metric subregularity of multivalued mappings. We should emphasize here that these stability results play a pivotal role in the analysis of the convergence rate of our algorithm. In Section 3, we propose an augmented Lagrangian algorithm to solve the general convex composite optimization model and analyze its asymptotic superlinear convergence. The semismooth Newton algorithm for solving the inner subproblems and the efficient implementation of the algorithm are also presented in this section. In Section 4, we conduct extensive numerical experiments to evaluate the performance of Ssnal in solving various Lasso problems. We conclude our paper in the final section.

2 Preliminary

We discuss in this section some stability properties of the convex composite optimization problems. It will become apparent later that these stability properties are the key ingredients for establishing the fast convergence of our augmented Lagrangian method.

Recall that $\mathcal{X}$ and $\mathcal{Y}$ are two real finite dimensional Euclidean spaces. For a given closed proper convex function $p : \mathcal{X} \to (-\infty, +\infty]$, the proximal mapping $\text{Prox}_p(\cdot)$ associated with p is defined by

$$\text{Prox}_p(x) := \arg\min_{u \in \mathcal{X}} \left\{ p(x) + \frac{1}{2}\|u - x\|^2 \right\}, \quad \forall x \in \mathcal{X}.$$

We will often make use of the following Moreau identity $\text{Prox}_{tp}(x) + t\text{Prox}_{p^*/t}(x/t) = x$, where $t > 0$ is a given parameter. Denote $\text{dist}(x, C) = \inf_{x' \in C} \|x - x'\|$ for any $x \in \mathcal{X}$ and any set $C \subset \mathcal{X}$.

Let $F : \mathcal{X} \rightrightarrows \mathcal{Y}$ be a multivalued mapping. We define the graph of F to be the set

$$\text{gph}F := \{(x, y) \in \mathcal{X} \times \mathcal{Y} \mid y \in F(x)\}.$$

F^{-1}, the inverse of F, is the multivalued mapping from $\mathcal{Y}$ to $\mathcal{X}$ whose graph is $\{(y, x) \mid (x, y) \in \text{gph}F\}$.

Definition 1 (Error bound)**.** *Let $F : \mathcal{X} \rightrightarrows \mathcal{Y}$ be a multivalued mapping and $y \in \mathcal{Y}$ satisfy $F^{-1}(y) \neq \emptyset$. F is said to satisfy the error bound condition for the point y with modulus $\kappa \geq 0$, if there exists $\varepsilon > 0$ such that if $x \in \mathcal{X}$ with $\text{dist}(y, F(x)) \leq \varepsilon$ then*

$$\text{dist}(x, F^{-1}(y)) \leq \kappa \, \text{dist}(y, F(x)). \tag{2}$$

The above error bound condition was called the growth condition in [33] and was used to analyze the local linear convergence properties of the proximal point algorithm. Recall that $F : \mathcal{X} \rightrightarrows \mathcal{Y}$ is called a polyhedral multifunction if its graph is the union of finitely many polyhedral convex sets. In [38], Robinson established the following celebrated proposition on the error bound result for polyhedral multifunctions.

Proposition 1. *Let F be a polyhedral multifunction from $\mathcal{X}$ to $\mathcal{Y}$. Then, F satisfies the error bound condition* (2) *for any point $y \in \mathcal{Y}$ satisfying $F^{-1}(y) \neq \emptyset$ with a common modulus $\kappa \geq 0$.*

For later uses, we present the following definition of metric subregularity from Chapter 3 in [12].

Definition 2 (Metric subregularity)**.** *Let $F : \mathcal{X} \rightrightarrows \mathcal{Y}$ be a multivalued mapping and $(\bar{x}, \bar{y}) \in \text{gph}F$. F is said to be metrically subregular at $\bar{x}$ for $\bar{y}$ with modulus $\kappa \geq 0$, if there exist neighborhoods U of $\bar{x}$ and V of $\bar{y}$ such that*

$$\text{dist}(x, F^{-1}(\bar{y})) \leq \kappa \, \text{dist}(\bar{y}, F(x) \cap V), \quad \forall x \in U.$$

From the above definition, we see that if $F : \mathcal{X} \rightrightarrows \mathcal{Y}$ satisfies the error bound condition (2) for $\bar{y}$ with the modulus κ, then it is metrically subregular at $\bar{x}$ for $\bar{y}$ with the same modulus κ for any $\bar{x} \in F^{-1}(\bar{y})$.

The following definition on essential smoothness is taken from [39, Section 26].

Definition 3 (Essential smoothness)**.** *A proper convex function f is essentially smooth if f is differentiable on* $\operatorname{int}(\operatorname{dom} f) \neq \emptyset$ *and* $\lim_{k\to\infty} \|\nabla f(x^k)\| = +\infty$ *whenever* $\{x^k\}$ *is a sequence in* $\operatorname{int}(\operatorname{dom} f)$ *converging to a boundary point x of* $\operatorname{int}(\operatorname{dom} f)$*.*

Consider the following composite convex optimization model

$$\max -\{f(x) := h(\mathcal{A}x) + \langle c, x\rangle + p(x)\}, \tag{3}$$

where $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is a linear map, $h : \mathcal{Y} \to \Re$ and $p : \mathcal{X} \to (-\infty, +\infty]$ are two closed proper convex functions, $c \in \mathcal{X}$ is a given vector. The dual of (3) can be written as

$$\min \{h^*(y) + p^*(z) \mid \mathcal{A}^*y + z = c\}, \tag{4}$$

where g^* and p^* are the Fenchel conjugate functions of g and h, respectively. Throughout this section, we make the following blanket assumption on $h^*(\cdot)$ and $p^*(\cdot)$.

Assumption 1. *$h^*(\cdot)$ is essentially smooth and $p^*(\cdot)$ is either an indicator function $\delta_P(\cdot)$ or a support function $\delta^*_P(\cdot)$ for some nonempty polyhedral convex set $P \subseteq \mathcal{X}$. Note that ∇h^* is locally Lipschitz continuous and directionally differentiable on* $\operatorname{int}(\operatorname{dom} h^*)$*.*

Under Assumption 1, by [39, Theorem 26.1], we know that $\partial h^*(y) = \emptyset$ whenever $y \notin \operatorname{int}(\operatorname{dom} h^*)$. Denote by l the Lagrangian function for (4):

$$l(y, z; x) = h^*(y) + p^*(z) - \langle x, \mathcal{A}^*y + z - c\rangle, \quad \forall\, (y, z, x) \in \mathcal{Y} \times \mathcal{X} \times \mathcal{X}. \tag{5}$$

Corresponding to the closed proper convex function f in the objective of (3) and the convex-concave function l in (4), define the maximal monotone operators $\mathcal{T}_f$ and $\mathcal{T}_l$ [41], by

$$\mathcal{T}_f(x) := \partial f(x), \quad \mathcal{T}_l(y, z, x) := \{(y', z', x') \mid (y', z', -x') \in \partial l(y, z, x)\},$$

whose inverse are given, respectively, by

$$\mathcal{T}_f^{-1}(x') := \partial f^*(x'), \quad \mathcal{T}_l^{-1}(y', z', x') := \{(y, z, x) \mid (y', z', -x') \in \partial l(y, z, x)\}.$$

Different from the case for $\mathcal{T}_f$ [32, 45, 51], very limited stability results are known for $\mathcal{T}_l$. Next, as a tool for studying the convergence rate of Ssnal, we shall establish a theorem which reveals the metric subregularity of $\mathcal{T}_l$ under some mild assumptions.

The KKT system associated with problem (4) is given as follows:

$$0 \in \partial h^*(y) - \mathcal{A}x, \quad 0 \in -x + \partial p^*(z), \quad 0 = \mathcal{A}^*y + z - c, \quad (x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{X}.$$

Assume that the above KKT system has at least one solution. This existence assumption together with the essentially smoothness assumption on h^* implies that the above KKT system can be equivalently rewritten as

$$0 = \nabla h^*(y) - \mathcal{A}x, \quad 0 \in -x + \partial p^*(z), \quad 0 = \mathcal{A}^*y + z - c, \quad (x, y, z) \in \mathcal{X} \times \operatorname{int}(\operatorname{dom} h^*) \times \mathcal{X}. \tag{6}$$

Therefore, under the above assumptions, we only need to focus on $\operatorname{int}(\operatorname{dom} h^*) \times \mathcal{X}$ when solving problem (4). Let $(\bar{y}, \bar{z})$ be an optimal solution to problem (4). Then, we know that the set of the

Lagrangian multipliers associated with $(\bar{y}, \bar{z})$, denoted as $\mathcal{M}(\bar{y}, \bar{z})$, is nonempty. Define the critical cone associated with (4) at $(\bar{y}, \bar{z})$ as follows:

$$\mathcal{C}(\bar{y}, \bar{z}) := \left\{(d_1, d_2) \in \mathcal{Y} \times \mathcal{X} \mid \mathcal{A}^* d_1 + d_2 = 0, \langle \nabla h^*(\bar{y}),\, d_1\rangle + (p^*)'(\bar{z}; d_2) = 0, d_2 \in \mathcal{T}_{\mathrm{dom}(p^*)}(\bar{z})\right\}, \tag{7}$$

where $(p^*)'(\bar{z}; d_2)$ is the directional derivative of p^* at $\bar{z}$ with respect to d_2, $\mathcal{T}_{\mathrm{dom}(p^*)}(\bar{z})$ is the tangent cone of $\mathrm{dom}(p^*)$ at $\bar{z}$. When the conjugate function p^* is taken to be the indicator function of a nonempty polyhedral set P, the above definition reduces directly to the standard definition of the critical cone in the nonlinear programming setting.

Definition 4 (Second order sufficient condition)**.** *Let $(\bar{y}, \bar{z}) \in \mathcal{Y} \times \mathcal{X}$ be an optimal solution to problem* (4) *with $\mathcal{M}(\bar{y}, \bar{z}) \neq \emptyset$. We say that the second order sufficient condition for problem* (4) *holds at $(\bar{y}, \bar{z})$ if*

$$\langle d_1,\, (\nabla h^*)'(\bar{y}; d_1)\rangle > 0, \quad \forall\, 0 \neq (d_1, d_2) \in \mathcal{C}(\bar{y}, \bar{z}).$$

By building on the proof ideas from the literature on nonlinear programming problems [11, 26, 23], we are able to prove the following result on the metric subregularity of $\mathcal{T}_l$. This allows us to prove the linear and even the asymptotic superlinear convergence of the sequences generated by the Ssnal algorithm to be presented in the next section even when the objective in problem (3) does not possess the piecewise linear-quadratic structure as in the Lasso problem (1).

Theorem 1. *Assume that the KKT system* (6) *has at least one solution and denote it as $(\bar{x}, \bar{y}, \bar{z})$. Suppose that Assumption* 1 *holds and that the second order sufficient condition for problem* (4) *holds at $(\bar{y}, \bar{z})$. Then, $\mathcal{T}_l$ is metrically subregular at $(\bar{y}, \bar{z}, \bar{x})$ for the origin.*

Proof. First, we claim that there exists a neighborhood $\mathcal{U}$ of $(\bar{x}, \bar{y}, \bar{z})$ such that for any $w = (u_1, u_2, v) \in \mathcal{W} := \mathcal{Y} \times \mathcal{X} \times \mathcal{X}$ with $\|w\|$ sufficiently small, any solution $(x(w), y(w), z(w)) \in \mathcal{U}$ of the perturbed KKT system

$$0 = \nabla h^*(y) - \mathcal{A}x - u_1, \quad 0 \in -x - u_2 + \partial p^*(z), \quad 0 = \mathcal{A}^* y + z - c - v \tag{8}$$

satisfies the following estimate

$$\|(y(w), z(w)) - (\bar{y}, \bar{z})\| = O(\|w\|). \tag{9}$$

For the sake of contradiction, suppose that our claim is not true. Then, there exist some sequences $\{w^k := (u_1^k, u_2^k, v^k)\}$ and $\{(x^k, y^k, z^k)\}$ such that $w^k \to 0$, $(x^k, y^k, z^k) \to (\bar{x}, \bar{y}, \bar{z})$, for each k the point (x^k, y^k, z^k) is a solution of (8) for $w = w^k$, and

$$\|(y^k, z^k) - (\bar{y}, \bar{z})\| > \gamma_k \|w^k\|$$

with some $\gamma_k > 0$ such that $\gamma_k \to \infty$. Passing onto a subsequence if necessary, we assume that $\{\big((y^k, z^k) - (\bar{y}, \bar{z})\big)/\|(y^k, z^k) - (\bar{y}, \bar{z})\|\}$ converges to some $\xi = (\xi_1, \xi_2) \in \mathcal{Y} \times \mathcal{X}$, $\|\xi\| = 1$. Then, setting $t_k = \|(y^k, z^k) - (\bar{y}, \bar{z})\|$ and passing to a subsequence further if necessary, by the local Lipschitz continuity and the directional differentiability of $\nabla h^*(\cdot)$ at $\bar{y}$, we know that for all k sufficiently large

$$\begin{aligned}\nabla h^*(y^k) - \nabla h^*(\bar{y}) =& \nabla h^*(\bar{y} + t_k \xi_1) - \nabla h^*(\bar{y}) + \nabla h^*(y^k) - \nabla h^*(\bar{y} + t_k \xi_1) \\ =& t_k (\nabla h^*)'(\bar{y}; \xi_1) + o(t_k) + O(\|y^k - \bar{y} - t_k \xi_1\|) \\ =& t_k (\nabla h^*)'(\bar{y}; \xi_1) + o(t_k).\end{aligned}$$

Meanwhile, simple calculations show that for all k sufficiently large

$$0 = \mathcal{A}(y^k - \bar{y}) + (z^k - \bar{z}) - v^k = t_k(\mathcal{A}^*\xi_1 + \xi_2) + o(t_k) \tag{10}$$

and

$$0 = \nabla h^*(y^k) - \nabla h^*(\bar{y}) - \mathcal{A}(x^k - \bar{x}) + \mathcal{A}u_2^k - u_1^k = t_k(\nabla h^*)'(\bar{y}; \xi_1) + o(t_k) - \mathcal{A}(x^k - \bar{x}). \tag{11}$$

Dividing both sides of equation (10) by t_k and then taking limits, we obtain

$$\mathcal{A}^*\xi_1 + \xi_2 = 0, \tag{12}$$

which further implies that

$$\langle \nabla h^*(\bar{y}),\, \xi_1 \rangle = \langle \mathcal{A}\bar{x},\, \xi_1 \rangle = -\langle \bar{x},\, \xi_2 \rangle. \tag{13}$$

Denote for each k, $\hat{x}^k := x^k + u_2^k$. Since $\hat{x}^k \in \partial p^*(z^k)$, we know that for all k sufficiently large,

$$z^k = \bar{z} + t_k\xi_2 + o(t_k) \in \operatorname{dom} p^*.$$

That is, $\xi_2 \in \mathcal{T}_{\operatorname{dom} p^*}(\bar{z})$.

According to the structure of p^*, we separate our discussions into two cases.

Case I: There exists a nonempty polyhedral convex set P such that $p^*(z) = \delta_P^*(z)$, $\forall z \in \mathcal{X}$. Then, for each k, it holds that

$$\hat{x}^k = \Pi_P(z^k + \hat{x}^k).$$

By [13, Theorem 4.1.1], we have

$$\hat{x}^k = \Pi_P(z^k + \hat{x}^k) = \Pi_P(\bar{z} + \bar{x}) + \Pi_{\mathcal{C}}(z^k - \bar{z} + \hat{x}^k - \bar{x}) = \bar{x} + \Pi_{\mathcal{C}}(z^k - \bar{z} + \hat{x}^k - \bar{x}), \tag{14}$$

where $\mathcal{C}$ is the critical cone of P at $\bar{z} + \bar{x}$, i.e.,

$$\mathcal{C} \equiv \mathcal{C}_P(\bar{z} + \bar{x}) := \mathcal{T}_P(\bar{x}) \cap \bar{z}^\perp.$$

Since $\mathcal{C}$ is a polyhedral cone, we know from [13, Proposition 4.1.4] that $\Pi_{\mathcal{C}}(\cdot)$ is a piecewise linear function, i.e., there exist a positive integer l and orthogonal projectors $B_1, \ldots, B_l$ such that for any $x \in \mathcal{X}$,

$$\Pi_{\mathcal{C}}(x) \in \{B_1 x, \ldots, B_l x\}.$$

By restricting to a subsequence if necessary, we may further assume that there exists $1 \le j' \le l$ such that for all $k \ge 1$,

$$\Pi_{\mathcal{C}}(z^k - \bar{z} + \hat{x}^k - \bar{x}) = B_{j'}(z^k - \bar{z} + \hat{x}^k - \bar{x}) = \Pi_{\mathrm{Range}(B_{j'})}(z^k - \bar{z} + \hat{x}^k - \bar{x}). \tag{15}$$

Denote $L := \mathrm{Range}(B_{j'})$. Combining (14) and (15), we get

$$\mathcal{C} \cap L \ni (\hat{x}^k - \bar{x}) \perp (z^k - \bar{z}) \in \mathcal{C}^\circ \cap L^\perp,$$

where $\mathcal{C}^\circ$ is the polar cone of $\mathcal{C}$. Since $\hat{x}^k - \bar{x} \in \mathcal{C}$, we have $\langle \hat{x}^k - \bar{x},\, \bar{z} \rangle = 0$, which, together with $\langle \hat{x}^k - \bar{x},\, z^k - \bar{z} \rangle = 0$, implies $\langle \hat{x}^k - \bar{x},\, z^k \rangle = 0$. Thus for all k sufficiently large,

$$\langle z^k,\, \hat{x}^k \rangle = \langle z^k,\, \bar{x} \rangle = \langle \bar{z} + t_k\xi_2 + o(t_k),\, \bar{x} \rangle,$$

and it follows that

$$\begin{aligned}(p^*)'(\bar{z};\xi_2) &= \lim_{k\to\infty}\frac{\delta_P^*(\bar{z}+t_k\xi_2)-\delta_P^*(\bar{z})}{t_k} = \lim_{k\to\infty}\frac{\delta_P^*(z^k)-\delta_P^*(\bar{z})}{t_k} \\ &= \lim_{k\to\infty}\frac{\langle z^k,\, \hat{x}^k\rangle - \langle \bar{z},\, \bar{x}\rangle}{t_k} = \langle \bar{x},\, \xi_2\rangle.\end{aligned} \tag{16}$$

By (12), (13) and (16), we know that $(\xi_1,\xi_2)\in\mathcal{C}(\bar{y},\bar{z})$. Since $\mathcal{C}\cap\mathcal{L}$ is a polyhedral convex cone, we know from [39, Theorem 19.3] that $\mathcal{A}(\mathcal{C}\cap\mathcal{L})$ is also a polyhedral convex cone, which, together with (11), implies

$$(\nabla h^*)'(\bar{y};\xi_1)\in\mathcal{A}(\mathcal{C}\cap L).$$

Therefore, there exists a vector $\eta\in\mathcal{C}\cap L$ such that

$$\langle \xi_1,\, (\nabla h^*)'(\bar{y};\xi_1)\rangle = \langle \xi_1,\, \mathcal{A}\eta\rangle = -\langle \xi_2,\, \eta\rangle = 0,$$

where the last equality follows from the fact that $\xi_2\in\mathcal{C}^\circ\cap L^\perp$. Note that the last inclusion holds since the polyhedral convex cone $\mathcal{C}^\circ\cap L^\perp$ is closed and

$$\xi_2 = \lim_{k\to\infty}\frac{z^k-\bar{z}}{t_k}\in\mathcal{C}^\circ\cap L^\perp.$$

As $0\neq\xi=(\xi_1,\xi_2)\in\mathcal{C}(\bar{y},\bar{z})$, but $\langle\xi_1,\, (\nabla h^*)'(\bar{y};\xi_1)\rangle = 0$, this contradicts the assumption that the second order sufficient condition holds for (4) at $(\bar{y},\bar{z})$. Thus we have proved our claim for Case I.

Case II: There exists a nonempty polyhedral convex set P such that $p^*(z)=\delta_P(z)$, $\forall z\in\mathcal{X}$. Then, we know that for each k,

$$z^k = \Pi_P(z^k+\hat{x}^k).$$

Since $(\delta_P)'(\bar{z};d_2)=0, \forall d_2\in\mathcal{T}_{\mathrm{dom}(p^*)}(\bar{z})$, the critical cone in (7) now takes the following form

$$\mathcal{C}(\bar{y},\bar{z}) = \left\{(d_1,d_2)\in\mathcal{Y}\times\mathcal{X}\mid \mathcal{A}^*d_1+d_2=0, \langle\nabla h^*(\bar{y}),\, d_1\rangle = 0, d_2\in\mathcal{T}_{\mathrm{dom}(p^*)}(\bar{z})\right\}.$$

Similar to Case I, without loss of generality, we can assume that there exists a subspace L such that for all $k\geq 1$,

$$\mathcal{C}\cap L\ni (z^k-\bar{z})\perp(\hat{x}^k-\bar{x})\in\mathcal{C}^\circ\cap L^\perp,$$

where

$$\mathcal{C}\equiv\mathcal{C}_P(\bar{z}+\bar{x}) := \mathcal{T}_P(\bar{z})\cap\bar{x}^\perp.$$

Since $\mathcal{C}\cap L$ is a polyhedral convex cone, we know

$$\xi_2 = \lim_{k\to\infty}\frac{z^k-\bar{z}}{t_k}\in\mathcal{C}\cap L,$$

and consequently $\langle\xi_2,\, \bar{x}\rangle = 0$, which, together with (12) and (13), implies $\xi=(\xi_1,\xi_2)\in\mathcal{C}(\bar{y},\bar{z})$. By (11) and the fact that $\mathcal{A}(\mathcal{C}^\circ\cap\mathcal{L}^\perp)$ is a polyhedral convex cone, we know that

$$(\nabla h^*)'(\bar{y};\xi_1)\in\mathcal{A}(\mathcal{C}^\circ\cap L^\perp).$$

Therefore, there exists a vector $\eta\in\mathcal{C}^\circ\cap L^\perp$ such that

$$\langle \xi_1,\, (\nabla h^*)'(\bar{y};\xi_1)\rangle = \langle \xi_1,\, \mathcal{A}\eta\rangle = -\langle \xi_2,\, \eta\rangle = 0.$$

Since $\xi = (\xi_1, \xi_2) \neq 0$, we arrive at a contradiction to the assumed second order sufficient condition. So our claim is also true for this case.

In summary, we have proven that there exists a neighborhood $\mathcal{U}$ of $(\bar{x}, \bar{y}, \bar{z})$ such that for any w close enough to the origin, equation (9) holds for any solution $(x(w), y(w), z(w)) \in \mathcal{U}$ to the perturbed KKT system (8). Next we show that $\mathcal{T}_l$ is metrically subregular at $(\bar{y}, \bar{z}, \bar{x})$ for the origin.

Define the mapping $\Theta_{KKT} : \mathcal{X} \times \mathcal{Y} \times \mathcal{X} \times \mathcal{W} \to \mathcal{Y} \times \mathcal{X} \times \mathcal{X}$ by

$$\Theta_{KKT}(x, y, z, w) := \begin{pmatrix} \nabla h^*(y) - \mathcal{A}x - u_1 \\ z - \mathrm{Prox}_{p^*}(z + x + u_2) \\ \mathcal{A}^*y + z - c - v \end{pmatrix}, \quad \forall (x, y, z, w) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{X} \times \mathcal{W}$$

and define the mapping $\theta : \mathcal{X} \to \mathcal{Y} \times \mathcal{X} \times \mathcal{X}$ as follows:

$$\theta(x) := \Theta_{\mathrm{KKT}}(x, \bar{y}, \bar{z}, 0), \quad \forall x \in \mathcal{X}.$$

Then, we have $x \in \mathcal{M}(\bar{y}, \bar{z})$ if and only if $\theta(x) = 0$. Since $\mathrm{Prox}_{p^*}(\cdot)$ is a piecewise affine function, $\theta(\cdot)$ is a piecewise affine function and thus a polyhedral multifunction. By using Theorem 1 and shrinking the neighborhood $\mathcal{U}$ if necessary, for any w close enough to the origin and any solution $(x(w), y(w), z(w)) \in \mathcal{U}$ of the perturbed KKT system (8), we have

$$\begin{aligned} \mathrm{dist}(x(w), \mathcal{M}(\bar{y}, \bar{z})) &= O(\|\theta(x(w))\|) \\ &= O(\|\Theta_{\mathrm{KKT}}(x(w), \bar{y}, \bar{z}, 0) - \Theta_{\mathrm{KKT}}(x(w), y(w), z(w), w)\|) \\ &= O(\|w\| + \|(y(w), z(w)) - (\bar{y}, \bar{z})\|), \end{aligned}$$

which, together with (9), implies the existence of a constant $\kappa \geq 0$ such that

$$\|(y(w), z(w)) - (\bar{y}, \bar{z})\| + \mathrm{dist}(x(w), \mathcal{M}(\bar{y}, \bar{z})) \leq \kappa \|w\|. \tag{17}$$

Thus, by Definition 2, we have proven that $\mathcal{T}_l$ is metrically subregular at $(\bar{y}, \bar{z}, \bar{x})$ for the origin. The proof of the theorem is completed. □

Remark 1. *For convex piecewise linear-quadratic programming problems such as the ℓ_1 and elastic net regularized LS problem, we know from* [44] *and* [42, Proposition 12.30] *that the corresponding operators $\mathcal{T}_l$ and $\mathcal{T}_f$ are polyhedral multifunctions and thus, by Theorem 1, the error bound condition holds. Here we emphasize again that the error bound condition holds for the Lasso problem* (1)*. Moreover, $\mathcal{T}_f$, associated with the ℓ_1 or elastic net regularized logistic regression model, i.e., for a given vector $b \in \Re^m$, the loss function $h : \Re^m \to \Re$ in problem* (3) *takes the form*

$$h(y) = \sum_{i=1}^{m} \log(1 + e^{-b_i y_i}), \quad \forall y \in \Re^m,$$

also satisfies the error bound condition [32, 45]. *Meanwhile, from Theorem 1, we know that $\mathcal{T}_l$ corresponding to the ℓ_1 or the elastic net regularized logistic regression model is metrically subregular at any solutions to the KKT system* (6) *for the origin.*

3 An augmented Lagrangian method with asymptotic superlinear convergence

Recall the general convex composite model (3)

$$(\mathbf{P}) \quad \max -\{f(x) = h(\mathcal{A}x) + \langle c,\, x\rangle + p(x)\}$$

and its dual

$$(\mathbf{D}) \quad \min\{h^*(y) + p^*(z) \mid \mathcal{A}^*y + z = c\}.$$

In this section, we shall propose an asymptotically superlinearly convergent augmented Lagrangian method for solving (**P**) and (**D**). In this section, we make the following standing assumptions regarding the function h.

Assumption 2. (a) *$h : \mathcal{Y} \to \Re$ is a convex differentiable function whose gradient is $1/\alpha_h$-Lipschitz continuous, i.e.,*

$$\|\nabla h(y') - \nabla h(y)\| \le (1/\alpha_h)\|y' - y\|, \quad \forall y', y \in \mathcal{Y}.$$

(b) *h is essentially locally strongly convex* [20]*, i.e., for any compact and convex set $K \subset \operatorname{dom} \partial f$, there exists $\beta_K > 0$ such that*

$$(1-\lambda)h(y') + \lambda h(y) \ge h((1-\lambda)y' + \lambda y) + \frac{1}{2}\beta_K \lambda(1-\lambda)\|y' - y\|^2, \quad \forall y', y \in K,$$

for all $\lambda \in [0, 1]$.

Many commonly used loss functions in the machine learning literature satisfy the above mild assumptions. For example, h can be the loss function in the linear regression, logistic regression and Poisson regression models.

The aforementioned assumptions on h imply the following useful properties of h^*. Firstly, by [42, Proposition 12.60], we know that h^* is strongly convex with modulus α_h. Secondly, by [20, Corollary 4.4], we know that h^* is essentially smooth and ∇h^* is locally Lipschitz continuous on $\operatorname{int}(\operatorname{dom} h^*)$. If the solution set to the KKT system associated with (**P**) and (**D**) is further assumed to be nonempty, similar to what we have discussed in the last section, one only needs to focus on $\operatorname{int}(\operatorname{dom} h^*) \times \mathcal{X}$ when solving (**D**). Given $\sigma > 0$, the augmented Lagrangian function associated with (**D**) is given by

$$\mathcal{L}_\sigma(y, z; x) := l(y, z; x) + \frac{\sigma}{2}\|\mathcal{A}^*y + z - c\|^2, \quad \forall\, (y, z, x) \in \mathcal{Y} \times \mathcal{X} \times \mathcal{X},$$

where the Lagrangian function $l(\cdot)$ is defined in (5).

3.1 Ssnal: A semismooth Newton augmented Lagrangian algorithm for (D)

The detailed steps of our algorithm Ssnal for solving (**D**) are given as follows. Since a semismooth Newton method will be employed to solve the subproblems involved in this prototype of the inexact augmented Lagrangian method [41], it is natural for us to call our algorithm as Ssnal.

Algorithm Ssnal: **An inexact augmented Lagrangian method for (D).**

Let $\sigma_0 > 0$ be a given parameter. Choose $(y^0, z^0, x^0) \in \operatorname{int}(\operatorname{dom} h^*) \times \operatorname{dom} p^* \times \mathcal{X}$. For $k = 0, 1, \ldots$, perform the following steps in each iteration:

Step 1. Compute
$$(y^{k+1}, z^{k+1}) \approx \arg\min\{\Psi_k(y, z) := \mathcal{L}_{\sigma_k}(y, z; x^k)\}. \tag{18}$$

Step 2. Compute $x^{k+1} = x^k - \sigma_k(\mathcal{A}^* y^{k+1} + z^{k+1} - c)$ and update $\sigma_{k+1} \uparrow \sigma_\infty \leq \infty$.

Next, we shall adapt the results developed in [40, 41] and [33] to establish the global and local superlinear convergence of our algorithm.

Since $\Psi_k(\cdot)$ defined in (18) is strongly convex with modulus no smaller than $\min(\alpha_h, \sigma_k)$, the stopping criterion for (18) studied in [40, 41] can be simplified to:

$$\text{(A)} \quad \operatorname{dist}(0, \partial\Psi_k(y^{k+1}, z^{k+1})) \leq \varepsilon_k / \max(1, \sqrt{\sigma_k/\alpha_h}), \quad \sum_{k=0}^{\infty} \varepsilon_k < \infty.$$

Now, we can state the global convergence of Algorithm Ssnal from [40, 41] without much difficulty.

Theorem 2. *Suppose that Assumption* 2 *holds and that the solution set to* (**P**) *is nonempty. Let* $\{(y^k, z^k, x^k)\}$ *be the infinite sequence generated by Algorithm* Ssnal *with stopping criterion* (A)*. Then, the sequence* $\{x^k\}$ *is bounded and converges to an optimal solution of* (**P**)*. In addition,* $\{(y^k, z^k)\}$ *is also bounded and converges to the unique optimal solution* $(y^*, z^*) \in \operatorname{int}(\operatorname{dom} h^*) \times \operatorname{dom} p^*$ *of* (**D**).

Proof. Since the solution set to (**P**) is assumed to be nonempty, the optimal value of (**P**) is finite. From Assumption 2(a), we have that $\operatorname{dom} h = \mathcal{Y}$ and h^* is strongly convex [42, Proposition 12.60]. Then, by Fenchel's Duality Theorem [39, Corollary 31.2.1], we know that the solution set to (**D**) is nonempty and the optimal value of (**D**) is finite and equals to the optimal value of (**P**). That is, the solution set to the KKT system associated with (**P**) and (**D**) is nonempty. The uniqueness of the optimal solution $(y^*, z^*) \in \operatorname{int}(\operatorname{dom} h^*) \times \mathcal{X}$ of (**D**) then follows directly from the strong convexity of h^*. By combining this uniqueness with [41, Theorem 4], one can easily obtain the boundedness of $\{(y^k, z^k)\}$ and other desired results readily. □

We need the the following stopping criterion for the local convergence analysis:

$$\text{(B)} \quad \operatorname{dist}(0, \partial\Psi_k(y^{k+1}, z^{k+1})) \leq (\delta_k / \max(1, \sigma_k/\alpha_h)) \min\{1, \|x^{k+1} - x^k\|\}, \quad \sum_{k=0}^{\infty} \delta_k < +\infty,$$

where $x^{k+1} := x^k + \sigma_k(\mathcal{A}^* y^{k+1} + z^{k+1} - c)$.

Theorem 3. *Assume that Assumption* 2 *holds and that the solution set* Ω *to* (**P**) *is nonempty. Suppose that* $\mathcal{T}_f$ *satisfies the error bound condition* (2) *for the origin with modulus* a_f*. Let* $\{(y^k, z^k, x^k)\}$ *be any infinite sequence generated by Algorithm* Ssnal *with stopping criterion (B). Then, the sequence* $\{x^k\}$ *converges to* $x^* \in \Omega$ *and for all* k *sufficiently large,*

$$\operatorname{dist}(x^{k+1}, \Omega) \;\leq\; \theta_k \operatorname{dist}(x^k, \Omega),$$

where $\theta_k = \big(a_f(a_f^2+\sigma_k^2)^{-1/2}+2\delta_k\big)(1-\delta_k)^{-1} \to \theta_\infty = a_f(a_f^2+\sigma_\infty^2)^{-1/2} < 1$ *as* $k \to +\infty$*. Moreover, the sequence* $\{(y^k, z^k)\}$ *converges to the optimal unique solution* $(y^*, z^*) \in \operatorname{int}(\operatorname{dom} h^*) \times \operatorname{dom} p^*$ *to* $(\mathbf{D})$*.*

If in addition $\mathcal{T}_l$ *is metrically subregular at* (y^*, z^*, x^*) *for the origin with modulus* a_l*, then for all* k *sufficiently large,*

$$\|(y^{k+1}, z^{k+1}) - (y^*, z^*)\| \le \theta'_k \|x^{k+1} - x^k\|,$$

where $\theta'_k = a_l(1+\delta'_k)/\sigma_k$*, with* $\delta'_k = \delta_k / \max(\sigma_k^{-1}, \alpha_h^{-1})$ *and* $\lim_{k\to\infty} \theta'_k = a_l/\sigma_\infty$*.*

Proof. The first part of the theorem follows from [33, Theorem 2.1], [41, Proposition 7, Theorem 5] and Theorem 2. To prove the second part, we recall that if $\mathcal{T}_l$ is metrically subregular at (y^*, z^*, x^*) for the origin with the modulus a_l and $(y^k, z^k, x^k) \to (y^*, z^*, x^*)$, then for all k sufficiently large,

$$\|(y^{k+1}, z^{k+1}) - (y^*, z^*)\| + \operatorname{dist}(x^{k+1}, \Omega) \le a_l \operatorname{dist}(0, \mathcal{T}_l(y^{k+1}, z^{k+1}, x^{k+1})).$$

Therefore, by the estimate (4.21) in [41] and the stopping criterion (B), we obtain that for all k sufficiently large,

$$\|(y^{k+1}, z^{k+1}) - (y^*, z^*)\| \le a_l(1+\delta'_k)/\sigma_k \|x^{k+1} - x^k\|,$$

where $\delta'_k = \delta_k / \max(\sigma_k^{-1}, \alpha_h^{-1})$. This completes the proof for Theorem 3. □

We should emphasize here that by combining Remark 1 and Theorem 3, our Algorithm Ssnal is guaranteed to produce an asymptotically superlinearly convergent sequence when used to solve $(\mathbf{D})$ for many commonly used regularizers and loss functions. In particular, the Ssnal algorithm is asymptotically superlinearly convergent when applied to the dual of (1).

3.2 Solving the augmented Lagrangian subproblems

Here we shall propose an efficient semismooth Newton algorithm to solve the inner subproblems in the augmented Lagrangian method (18). That is, for some fixed $\sigma > 0$ and $\tilde{x} \in \mathcal{X}$, we consider to solve

$$\min_{y,z} \ \Psi(y,z) := \mathcal{L}_\sigma(y, z; \tilde{x}). \tag{19}$$

Since $\Psi(\cdot,\cdot)$ is a strongly convex function, we have that, for any $\alpha \in \Re$, the level set $\mathcal{L}_\alpha := \{(y,z) \in \operatorname{dom} h^* \times \operatorname{dom} p^* \mid \Psi(y,z) \le \alpha\}$ is a closed and bounded convex set. Moreover, problem (19) admits a unique optimal solution denoted as $(\bar{y}, \bar{z}) \in \operatorname{int}(\operatorname{dom} h^*) \times \operatorname{dom} p^*$.

Denote, for any $y \in \mathcal{Y}$,

$$\begin{aligned}\psi(y) :=& \inf_z \Psi(y,z)\\ =& h^*(y) + p^*(\operatorname{Prox}_{p^*/\sigma}(\tilde{x}/\sigma - \mathcal{A}^*y + c)) + \frac{1}{2\sigma}\|\operatorname{Prox}_{\sigma p}(\tilde{x} - \sigma(\mathcal{A}^*y - c))\|^2 - \frac{1}{2\sigma}\|\tilde{x}\|^2.\end{aligned}$$

Therefore, if $(\bar{y}, \bar{z}) = \arg\min \Psi(y,z)$, then $(\bar{y}, \bar{z}) \in \operatorname{int}(\operatorname{dom} h^*) \times \operatorname{dom} p^*$ can be computed simultaneously by

$$\bar{y} = \arg\min \psi(y), \quad \bar{z} = \operatorname{Prox}_{p^*/\sigma}(\tilde{x}/\sigma - \mathcal{A}^*\bar{y} + c).$$

Note that $\psi(\cdot)$ is strongly convex and continuously differentiable on $\operatorname{int}(\operatorname{dom} h^*)$ with

$$\nabla\psi(y) = \nabla h^*(y) - \mathcal{A}\operatorname{Prox}_{\sigma p}(\tilde{x} - \sigma(\mathcal{A}^*y - c)), \quad \forall y \in \operatorname{int}(\operatorname{dom} h^*).$$

Thus, $\bar{y}$ can be obtained via solving the following nonsmooth equation

$$\nabla \psi(y) = 0, \quad y \in \operatorname{int}(\operatorname{dom} h^*). \tag{20}$$

Let $y \in \operatorname{int}(\operatorname{dom} h^*)$ be any given point. Since h^* is a convex function with a locally Lipschitz continuous gradient on $\operatorname{int}(\operatorname{dom} h^*)$, the following operator is well defined:

$$\hat{\partial}^2 \psi(y) := \partial(\nabla h^*)(y) + \sigma \mathcal{A} \partial \mathrm{Prox}_{\sigma p}(\tilde{x} - \sigma(\mathcal{A}^* y - c))\mathcal{A}^*,$$

where $\partial(\nabla h^*)(y)$ is the Clarke subdifferential of ∇h^* at y [10], and $\partial \mathrm{Prox}_{\sigma p}(\tilde{x} - \sigma(\mathcal{A}^* y - c))$ is the Clarke subdifferential of the Lipschitz continuous mapping $\mathrm{Prox}_{\sigma p}(\cdot)$ at $\tilde{x} - \sigma(\mathcal{A}^* y - c)$. Note that from [10, Proposition 2.3.3 and Theorem 2.6.6], we know that

$$\partial^2 \psi(y)\,(d) \subseteq \hat{\partial}^2 \psi(y)\,(d), \quad \forall\, d \in \mathcal{Y},$$

where $\partial^2 \psi(y)$ denotes the generalized Hessian of ψ at y. Define

$$V := H + \sigma \mathcal{A} U \mathcal{A}^* \tag{21}$$

with $H \in \partial^2 h^*(y)$ and $U \in \partial \mathrm{Prox}_{\sigma p}(\tilde{x} - \sigma(\mathcal{A}^* y - c))$. Then, we have $V \in \hat{\partial}^2 \psi(y)$. Since h^* is a strongly convex function, we know that H is symmetric positive definite on $\mathcal{Y}$ and thus V is also symmetric positive definite on $\mathcal{Y}$.

Under the mild assumption that ∇h^* and $\mathrm{Prox}_{\sigma p}$ are strongly semismooth (whose definition is given next), we can design a superlinearly convergent semismooth Newton method to solve the nonsmooth equation (20).

Definition 5 (Semismoothness [34, 37, 43]). *Let $F : \mathcal{O} \subseteq \mathcal{X} \to \mathcal{Y}$ be a locally Lipschitz continuous function on the open set $\mathcal{O}$. F is said to be semismooth at $x \in \mathcal{O}$ if F is directionally differentiable at x and for any $V \in \partial F(x + \Delta x)$ with $\Delta x \to 0$,*

$$F(x + \Delta x) - F(x) - V \Delta x = o(\|\Delta x\|).$$

F is said to be strongly semismooth at x if F is semismooth at x and

$$F(x + \Delta x) - F(x) - V \Delta x = O(\|\Delta x\|^2).$$

F is said to be a semismooth (respectively, strongly semismooth) function on $\mathcal{O}$ if it is semismooth (respectively, strongly semismooth) everywhere in $\mathcal{O}$.

Note that it is widely known in the nonsmooth optimization/equation community that continuous piecewise affine functions and twice continuously differentiable functions are all strongly semismooth everywhere. In particular, $\mathrm{Prox}_{\|\cdot\|_1}$, as a Lipschitz continuous piecewise affine function, is strongly semismooth. See [13] for more semismooth and strongly semismooth functions.

Now, we can design a semismooth Newton (Ssn) method to solve (20) as follows and could expect to get a fast superlinear or even quadratic convergence.

Algorithm Ssn: **A semismooth Newton algorithm for solving (20)** (Ssn($y^0, \tilde{x}, \sigma$)).

Given $\mu \in (0, 1/2)$, $\bar{\eta} \in (0, 1)$, $\tau \in (0, 1]$, and $\delta \in (0, 1)$. Choose $y^0 \in \operatorname{int}(\operatorname{dom} h^*)$. Iterate the following steps for $j = 0, 1, \ldots$.

Step 1. Choose $H_j \in \partial(\nabla h^*)(y^j)$ and $U_j \in \partial \operatorname{Prox}_{\sigma p}(\tilde{x} - \sigma(\mathcal{A}^* y^j - c))$. Let $V_j := H_j + \sigma \mathcal{A} U_j \mathcal{A}^*$. Solve the following linear system

$$V_j d + \nabla \psi(y^j) = 0 \tag{22}$$

exactly or by the conjugate gradient (CG) algorithm to find d^j such that

$$\|V_j d^j + \nabla \psi(y^j)\| \leq \min(\bar{\eta}, \|\nabla \psi(y^j)\|^{1+\tau}).$$

Step 2. (Line search) Set $\alpha_j = \delta^{m_j}$, where m_j is the first nonnegative integer m for which

$$y^j + \delta^m d^j \in \operatorname{int}(\operatorname{dom} h^*) \quad \text{and} \quad \psi(y^j + \delta^m d^j) \leq \psi(y^j) + \mu \delta^m \langle \nabla \psi(y^j), d^j \rangle.$$

Step 3. Set $y^{j+1} = y^j + \alpha_j \, d^j$.

The convergence results for the above Ssn algorithm are stated in the next theorem.

Theorem 4. *Assume that $\nabla h^*(\cdot)$ and $\operatorname{Prox}_{\sigma p}(\cdot)$ are strongly semismooth on $\operatorname{int}(\operatorname{dom} h^*)$ and $\mathcal{X}$, respectively. Let the sequence $\{y^j\}$ be generated by Algorithm* Ssn. *Then $\{y^j\}$ converges to the unique optimal solution $\bar{y} \in \operatorname{int}(\operatorname{dom} h^*)$ of the problem in (20) and*

$$\|y^{j+1} - \bar{y}\| = O(\|y^j - \bar{y}\|^{1+\tau}).$$

Proof. Since, by [52, Proposition 3.3], d^j is a descent direction, Algorithm Ssn is well-defined. By (21), we know that for any $j \geq 0$, $V_j \in \hat{\partial}^2 \psi(y^j)$. Then, we can prove the conclusion of this theorem by following the proofs to [52, Theorems 3.4 and 3.5]. We omit the details here. □

We shall now discuss the implementations of stopping criteria (A) and (B) for Algorithm Ssn to solve the subproblem (18) in Algorithm Ssnal. Note that when Ssn is applied to minimize $\Psi_k(\cdot)$ to find

$$y^{k+1} = \text{Ssn}(y^k, x^k, \sigma_k) \quad \text{and} \quad z^{k+1} = \operatorname{Prox}_{p^*/\sigma_k}(x^k/\sigma_k - \mathcal{A}^* y^{k+1} + c),$$

we have $(\nabla \psi_k(y^{k+1}), 0) \in \partial \Psi_k(y^{k+1}, z^{k+1})$ where $\psi_k(y) := \inf_z \Psi_k(y, z)$ for all $y \in \mathcal{Y}$. Therefore, the stopping criteria (A) and (B) can be achieved by the following implementable criteria

$$(\mathrm{A}') \quad \|\nabla \psi_k(y^{k+1})\| \leq \varepsilon_k / \max(1, \sqrt{\sigma_k/\alpha_h}), \quad \textstyle\sum_{k=0}^{\infty} \varepsilon_k < \infty,$$

$$(\mathrm{B}') \quad \|\nabla \psi_k(y^{k+1})\| \leq (\delta_k / \max(1, \sigma_k/\alpha_h)) \min\{1, \sigma_k \|\mathcal{A}^* y^{k+1} + z^{k+1} - c\|\}, \quad \textstyle\sum_{k=0}^{\infty} \delta_k < +\infty.$$

That is, the stopping criteria (A) and (B) will be satisfied as long as $\|\nabla \psi_k(y^{k+1})\|$ is sufficiently small.

3.3 An efficient implementation of Ssn for solving subproblems (18)

When Algorithm Ssnal is applied to solve the general convex composite optimization model (**P**), the key part is to use Algorithm Ssn to solve the subproblems (18). In this subsection, we shall discuss an efficient implementation of Ssn for solving the aforementioned subproblems when the nonsmooth regularizer p is chosen to be $\lambda\|\cdot\|_1$ for some $\lambda > 0$. Clearly, in Algorithm Ssn, the most important step is the computation of the search direction d^j from the linear system (22). So we shall first discuss the solving of this linear system.

Let $(\tilde{x}, y) \in \Re^n \times \Re^m$ and $\sigma > 0$ be given. We consider the following Newton linear system

$$(H + \sigma A U A^T)d = -\nabla\psi(y), \tag{23}$$

where $H \in \partial(\nabla h^*)(y)$, A denotes the matrix representation of $\mathcal{A}$ with respect to the standard bases of $\Re^n$ and $\Re^m$, $U \in \partial\mathrm{Prox}_{\sigma\lambda\|\cdot\|_1}(x)$ with $x := \tilde{x} - \sigma(A^T y - c)$. Since H is a symmetric and positive definite matrix, equation (23) can be equivalently rewritten as

$$\big(I_m + \sigma(L^{-1}A)U(L^{-1}A)^T\big)(L^T d) = -L^{-1}\nabla\psi(y),$$

where L is a nonsingular matrix obtained from the (sparse) Cholesky decomposition of H such that $H = LL^T$. In many applications, H is usually a sparse matrix. Indeed, when the function h in the primal objective is taken to be the squared loss or the logistic loss functions, the resulting matrices H are in fact diagonal matrices. That is, the costs of computing L and its inverse are negligible in most situations. Therefore, without loss of generality, we can consider a simplified version of (23) as follows

$$(I_m + \sigma A U A^T)d = -\nabla\psi(y), \tag{24}$$

which is precisely the Newton system associated with the standard Lasso problem (1). Since $U \in \Re^{n\times n}$ is a diagonal matrix, at the first glance, the costs of computing AUA^T and the matrix-vector multiplication $AUA^T d$ for a given vector $d \in \Re^m$ are $\mathcal{O}(m^2 n)$ and $\mathcal{O}(mn)$, respectively. These computational costs are too expensive when the dimensions of A are large and can make the commonly employed approaches such as the Cholesky factorization and the conjugate gradient method inappropriate for solving (24). Fortunately, under the sparse optimization setting, if the sparsity of U is wisely taken into the consideration, one can substantially reduce these unfavorable computational costs to a level such that they are negligible or at least insignificant compared to other costs. Next, we shall show how this can be done by taking full advantage of the sparsity of U. This sparsity will be referred as the second order sparsity of the underlying problem.

For $x = \tilde{x} - \sigma(A^T y - c)$, in our computations, we can always choose $U \in \partial\mathrm{Prox}_{\sigma\lambda\|\cdot\|_1}(x)$ to be $U = \mathrm{Diag}(u)$, the diagonal matrix whose ith diagonal element is given by u_i with

$$u_i = \begin{cases} 0, & \text{if} \quad |x_i| \le \sigma\lambda, \\ 1, & \text{otherwise}, \end{cases} \qquad i = 1, \ldots, n.$$

Let $\mathcal{J} := \{j \mid |x_j| > \sigma\lambda,\ j = 1, \ldots, n\}$ and $r = |\mathcal{J}|$, the cardinality of $\mathcal{J}$. By taking the special 0-1 structure of U into consideration, we have that

$$AUA^T = (AU)(AU)^T = A_{\mathcal{J}}A_{\mathcal{J}}^T, \tag{25}$$

where $A_{\mathcal{J}} \in \Re^{m\times r}$ is the sub-matrix of A with those columns not in $\mathcal{J}$ being removed from A. Then, by using (25), we know that now the costs of computing AUA^T and $AUA^T d$ for a given vector

d are reduced to $\mathcal{O}(m^2r)$ and $\mathcal{O}(mr)$, respectively. Due to the sparsity promoting property of the regularizer p, the number r is usually much smaller than n. Thus, by exploring the aforementioned second order sparsity, we can greatly reduce the computational costs in solving the linear system (24) when using the Cholesky factorization. More specifically, the total computational costs of using the Cholesky factorization to solve the linear system are reduced from $\mathcal{O}(m^2(m+n))$ to $\mathcal{O}(m^2(m+r))$. See Figure 1 for an illustration on the reduction. This means that even if n happens to be extremely large (say, larger than 10^7), one can still solve the Newton linear system (24) efficiently via the Cholesky factorization as long as both m and r are moderate (say, less than 10^4). If, in addition, $r \ll m$, which is often the case when m is large and the optimal solutions to the underlying problem are sparse, instead of factorizing an $m \times m$ matrix, we can make use of the Sherman-Morrison-Woodbury formula [21] to get the inverse of $I_m + \sigma AUA^T$ by inverting a much smaller $r \times r$ matrix as follows:

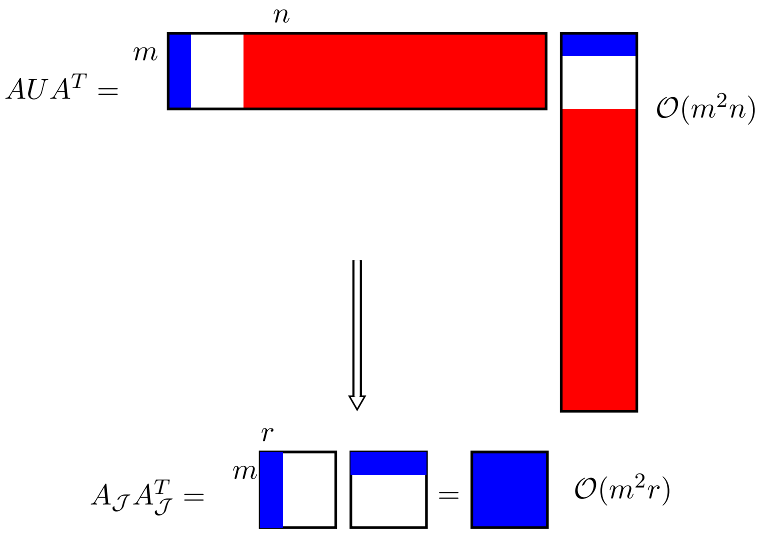

Figure 1: Reducing the computational costs from $\mathcal{O}(m^2n)$ to $\mathcal{O}(m^2r)$

$$(I_m + \sigma AUA^T)^{-1} = (I_m + \sigma A_{\mathcal{J}}A_{\mathcal{J}}^T)^{-1} = I_m - A_{\mathcal{J}}(\sigma^{-1}I_r + A_{\mathcal{J}}^TA_{\mathcal{J}})^{-1}A_{\mathcal{J}}^T.$$

See Figure 2 for an illustration on the computation of $A_{\mathcal{J}}^TA_{\mathcal{J}}$. In this case, the total computational costs for solving the Newton linear system (24) are reduced significantly further from $\mathcal{O}(m^2(m+r))$ to $\mathcal{O}(r^2(m+r))$. We should emphasize here that this dramatic reduction on the computational costs results from the wise combination of the careful examination of the existing second order sparsity in the Lasso-type problems and some "smart" numerical linear algebras.

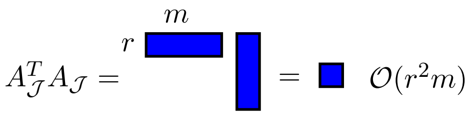

Figure 2: Further reducing the computational costs to $\mathcal{O}(r^2m)$

From the above arguments, we can see that as long as the number of the nonzero components of $\mathrm{Prox}_{\sigma\lambda\|\cdot\|_1}(x)$ is small, say less than $\sqrt{n}$ and $H_j \in \partial(\nabla h^*)(y^j)$ is a sparse matrix, e.g., a diagonal matrix, we can always solve the linear system (22) at very low costs. In particular, this is true for the Lasso problems admitting sparse solutions. Similar discussions on the reduction of the computational costs can also be conducted for the case when the conjugate gradient method is applied to solve the linear systems (22). Note that one may argue that even if the original problem has only sparse solutions, at certain stages, one may still encounter the situation that the number of the nonzero components of $\mathrm{Prox}_{\sigma\lambda\|\cdot\|_1}(x)$ is large. Our answer to this question is simple. Firstly, this phenomenon rarely occurs in practice since we always start with a sparse feasible point, e.g., the zero vector. Secondly, even at certain steps this phenomenon does occur, we just need to apply a small number of conjugate gradient iterations to the linear system (22) as in this case the parameter σ is normally small and the current point is far away from any sparse optimal solution. In summary, we have demonstrated how Algorithm Ssn can be implemented efficiently for solving sparse optimization problems of the form (18) with $p(\cdot)$ being chosen to be $\lambda\|\cdot\|_1$.

4 Numerical experiments for Lasso problems

In this section, we shall evaluate the performance of our algorithm Ssnal for solving large scale Lasso problems (1). We note that the relative performance of most of the existing algorithms mentioned in the introduction has recently been well documented in the two recent papers [17, 35], which appears to suggest that for some large scale sparse reconstruction problems, mfIPM[1] and FPC_AS[2] have mostly outperformed the other solvers. Hence, in this section we will compare our algorithm with these two state-of-the-art solvers. Note that mfIPM is a specialized interior-point based second-order method designed for the Lasso problem (1), whereas FPC_AS is a first-order method based on forward-backward operator splitting. Moreover, we also report the numerical performance of two most commonly used algorithms for solving Lasso problems: the accelerated proximal gradient (APG) algorithm as implemented by Liu et al. in SLEP[3] [31] and the alternating direction method of multipliers (ADMM) [18, 19]. For the purpose of comparisons, we also test the linearized ADMM (LADMM) [53]. We have implemented both ADMM and LADMM in Matlab with the step-length set to be 1.618. Although the existing solvers can perform impressively well on some easy-to-solve sparse reconstruction problems, as one will see later, they lack the ability to efficiently solve difficult problems such as the large scale regression problems when the data $\mathcal{A}$ is badly conditioned.

For the testing purpose, the regularization parameter λ in the Lasso problem (1) is chosen as

$$\lambda = \lambda_c \|\mathcal{A}^* b\|_\infty,$$

where $0 < \lambda_c < 1$. In our numerical experiments, we measure the accuracy of an approximate optimal solution $\tilde{x}$ for (1) by using the following relative KKT residual:

$$\eta = \frac{\|\tilde{x} - \mathrm{prox}_{\lambda\|\cdot\|_1}(\tilde{x} - (\mathcal{A}\tilde{x} - b))\|}{1 + \|\tilde{x}\| + \|\mathcal{A}\tilde{x} - b\|}.$$

[1] http://www.maths.ed.ac.uk/ERGO/mfipmcs/

[2] http://www.caam.rice.edu/~optimization/L1/FPC_AS/

[3] http://yelab.net/software/SLEP/

For a given tolerance $\epsilon > 0$, we will stop the tested algorithms when $\eta < \epsilon$. For all the tests in this section, we set $\epsilon = 10^{-6}$. The algorithms will also be stopped when they reach the maximum number of iterations (1000 iterations for our algorithm and mfIPM, and 20000 iterations for FPC_AS, APG, ADMM and LADMM) or the maximum computation time of 7 hours. All the parameters for mfIPM, FPC_AS and APG are set to the default values. All our computational results are obtained from a workstation running on 64-bit Windows Operating System having 16 cores with 32 Intel Xeon E5-2650 processors at 2.60GHz and 64 GB memory.

4.1 Numerical results for large scale regression problems

In this subsection, we test all the algorithms with the test instances $(\mathcal{A}, b)$ obtained from large scale regression problems in the LIBSVM datasets [8]. These data sets are collected from 10-K Corpus [27] and UCI data repository [29]. As suggested in [22], for the data sets **pyrim, triazines, abalone, bodyfat, housing, mpg, space_ga**, we expand their original features by using polynomial basis functions over those features. For example, the last digits in **pyrim5** indicates that an order 5 polynomial is used to generate the basis functions. This naming convention is also used in the rest of the expanded data sets. These test instances, shown in Table 1, can be quite difficult in terms of the problem dimensions and the largest eigenvalues of $\mathcal{A}\mathcal{A}^*$, which is denoted as $\lambda_{\max}(\mathcal{A}\mathcal{A}^*)$.

Table 1: Statistics of the UCI test instances.

probname	$m; n$	$\lambda_{\max}(\mathcal{A}\mathcal{A}^*)$
E2006.train	16087;150360	1.91e+05
log1p.E2006.train	16087;4272227	5.86e+07
E2006.test	3308;150358	4.79e+04
log1p.E2006.test	3308;4272226	1.46e+07
pyrim5	74;201376	1.22e+06
triazines4	186;635376	2.07e+07
abalone7	4177;6435	5.21e+05
bodyfat7	252;116280	5.29e+04
housing7	506;77520	3.28e+05
mpg7	392;3432	1.28e+04
space_ga9	3107;5005	4.01e+03

Table 2 reports the detailed numerical results for Ssnal, mfIPM, FPC_AS, APG, LADMM and ADMM in solving large scale regression problems. In the table, m denotes the number of samples, n denotes the number of features and "nnz" denotes the number of nonzeros in the solution x obtained by Ssnal using the following estimation

$$\text{nnz} := \min\Big\{k \mid \sum_{i=1}^{k} |\hat{x}_i| \geq 0.999\|x\|_1\Big\},$$

where $\hat{x}$ is obtained by sorting x such that $|\hat{x}_1| \geq \ldots \geq |\hat{x}_n|$. One can observe from Table 2 that all the tested first order algorithms except ADMM, i.e., FPC_AS, APG and LADMM fail to solve most of the test instances to the required accuracy after 20000 iterations or 7 hours. In particular, FPC_AS fails to produce a reasonably accurate solution for all the test instances. In fact, for 3 test instances, it breaks down due to some internal errors. This poor performance indicates that these first order methods cannot obtain reasonably accurate solutions when dealing with difficult large scale problems. While ADMM can solve most of the test instances, it needs much more time than Ssnal. For example, for the instance **housing7** with $\lambda_c = 10^{-3}$, we can see that Ssnal is at least 330 times faster than ADMM. In addition, Ssnal can solve the instance **pyrim5** in 9 seconds while ADMM reaches the maximum of 20000 iterations and consumes about 2 hours but only produce a rather inaccurate solution.

On the other hand, one can observe that the two second order information based methods Ssnal and mfIPM perform quite robustly despite the huge dimensions and the possibly badly conditioned data sets. More specifically, Ssnal is able to solve the instance **log1p.E2006.train** with approximately 4.3 million features in 20 seconds ($\lambda_c = 10^{-3}$). Among these two algorithms, clearly, Ssnal is far more efficient than the specialized interior-point method mfIPM for all the test instances, especially for large scale problems where the factor can be up to 300 times faster. While Ssnal can solve all the instances to the desired accuracy, as the problems get progressively more difficult (λ_c decreases from 10^{-3} to 10^{-4}), mfIPM fails on more test instances (2 out of 11 vs. 4 out of 11 instances). We also note that mfIPM can only reach a solution with the accuracy of 10^{-1} when it fails to compute the corresponding Newton directions. These facts indicate that the nonsmooth approach employed by Ssnal is far more superior compared to the interior point method in exploiting the sparsity in the generalized Hessian. The superior numerical performance of Ssnal indicates that it is a robust, high-performance solver for high-dimensional Lasso problems.

Table 2: The performance of Ssnal, mfIPM and FPC_AS on 11 selected regression problems (accuracy $\epsilon = 10^{-6}$). m is the sample size and n is the dimension of features. In the table, "a" = Ssnal, "b" = mfIPM, "c" = FPC_AS, "d" = APG, "e" = LADMM, and "f" = ADMM, respectively. "nnz" denotes the number of nonzeros in the solution obtained by Ssnal. "**Error**" indicates the algorithm breaks down due to some internal errors. The computation time is in the format of "hours:minutes:seconds". "00" in the time column means less than 0.5 seconds.

	λ_c	nnz	η						time					
probname $m;n$			a	b	c	d	e	f	a	b	c	d	e	f
E2006.train	10^{-3}	1	1.3-7	3.9-7	9.0-4	1.1-8	5.1-14	9.1-7	01	11	1:34:40	01	03	11:00
16087;150360	10^{-4}	1	3.7-7	1.6-9	9.7-4	1.5-7	2.6-14	7.3-7	01	14	1:37:52	01	04	11:23
log1p.E2006.train	10^{-3}	5	5.6-7	4.7-8	4.7-1	1.1-5	6.4-4	9.9-7	20	41:01	7:00:20	2:00:33	2:22:21	39:36
16087;4272227	10^{-4}	599	4.4-7	7.9-1	1.6-1	1.5-4	6.3-3	9.8-7	55	3:20:44	7:00:00	2:04:54	2:24:23	36:09
E2006.test	10^{-3}	1	1.6-9	2.9-7	3.7-4	5.5-8	4.0-14	6.7-7	00	05	33:30	00	01	26
3308;150358	10^{-4}	1	2.1-10	2.9-7	4.3-4	3.7-7	2.9-10	6.3-7	00	05	32:36	00	01	27
log1p.E2006.test	10^{-3}	8	9.2-7	1.5-8	9.8-1	1.1-4	9.9-7	9.9-7	17	37:24	7:00:01	1:25:11	22:52	5:58
3308;4272226	10^{-4}	1081	2.2-7	8.7-1	8.7-1	3.9-4	9.9-7	9.9-7	34	7:00:24	7:00:01	1:23:07	1:00:10	4:00
pyrim5	10^{-3}	72	9.9-7	3.2-7	8.8-1	8.9-4	4.0-4	3.2-5	05	21:27	2:01:23	8:48	9:30	23:17
74;201376	10^{-4}	78	7.1-7	6.6-8	9.8-1	3.4-3	3.7-3	1.6-3	09	49:49	1:09:16	9:28	9:55	24:30
triazines4	10^{-3}	519	8.4-7	8.8-1	9.1-1	1.9-3	2.9-3	3.2-4	36	53:30	7:30:17	54:44	1:02:26	2:17:11
186;635376	10^{-4}	260	9.9-7	8.9-1	9.8-1	1.1-2	1.6-2	1.1-3	1:44	53:33	7:00:00	1:03:56	1:01:26	2:00:18
abalone7	10^{-3}	24	5.7-7	3.5-7	**Error**	5.3-5	4.9-6	9.9-7	02	49	**Error**	10:38	18:04	9:52
4177;6435	10^{-4}	59	3.7-7	4.7-7	**Error**	3.9-3	3.3-5	9.9-7	03	3:19	**Error**	10:43	13:58	9:36
bodyfat7	10^{-3}	2	3.8-8	4.0-9	3.6-1	8.8-7	9.0-7	9.9-7	02	1:29	1:12:02	4:00	3:08	1:49
252;116280	10^{-4}	3	4.6-8	3.4-7	1.9-1	3.3-5	9.8-7	9.9-7	03	2:41	1:13:08	12:16	4:19	4:05
housing7	10^{-3}	158	2.3-7	8.1-1	8.4-1	2.6-4	1.7-4	9.9-7	04	5:13:19	1:41:01	16:52	20:18	22:12
506;77520	10^{-4}	281	7.7-7	8.5-1	4.2-1	5.3-3	5.9-4	6.6-6	08	5:00:09	1:39:36	17:04	20:53	42:36
mpg7	10^{-3}	47	2.0-8	6.5-7	**Error**	1.9-6	9.9-7	9.8-7	00	04	**Error**	38	14	07
392;3432	10^{-4}	128	4.1-7	8.5-7	4.5-1	1.2-4	1.3-6	9.9-7	00	13	4:09	39	1:01	11

Table 2: The performance of Ssnal, mfIPM and FPC_AS on 11 selected regression problems (accuracy $\epsilon = 10^{-6}$). m is the sample size and n is the dimension of features. In the table, "a" = Ssnal, "b" = mfIPM, "c" = FPC_AS, "d" = APG, "e" = LADMM, and "f" = ADMM, respectively. "nnz" denotes the number of nonzeros in the solution obtained by Ssnal. "**Error**" indicates the algorithm breaks down due to some internal errors. The computation time is in the format of "hours:minutes:seconds". "00" in the time column means less than 0.5 seconds.

	λ_c	nnz	η						time					
probname $m;n$			a	b	c	d	e	f	a	b	c	d	e	f
space_ga9	10^{-3}	14	4.8-7	2.6-7	1.5-2	2.5-7	9.9-7	9.9-7	01	16	30:55	1:46	42	37
3107;5005	10^{-4}	38	3.7-7	3.0-7	4.0-2	1.8-5	9.9-7	9.9-7	01	41	30:26	6:46	2:20	56

4.2 Numerical results for Sparco collection

In this subsection, the test instances $(\mathcal{A}, b)$ are taken from 8 real valued sparse reconstruction problems in the Sparco collection [5]. For testing purpose, we introduce a 60dB noise to b (as in [17]) by using the Matlab command: `b = awgn(b,60,'measured').` For these test instances, the matrix representations of the linear maps $\mathcal{A}$ are not available. Hence, ADMM will not be tested in this subsection, as it will be extremely expensive, if not impossible, to compute and factorize $\mathcal{I} + \sigma\mathcal{A}\mathcal{A}^*$.

In Table 3, we report the detailed computational results obtained by Ssnal, mfIPM, FPC_AS, APG and LADMM in solving two large scale instances **srcsep1** and **srcsep2** in the Sparco collection. Here, we test five choices of λ_c, i.e., $\lambda_c = 10^{-0.8}, 10^{-1}, 10^{-1.2}, 10^{-1.5}, 10^{-2}$. As one can observe, as λ_c decreases, the number of nonzeros (nnz) in the computed solution increases. In the table, we list some statistics of the test instances, including the problem dimensions (m, n) and the largest eigenvalue of $\mathcal{A}\mathcal{A}^*$ ($\lambda_{\max}(\mathcal{A}\mathcal{A}^*)$). For all the tested algorithms, we present the iteration counts, the relative KKT residuals as well as the computation times (in the format of hours:minutes:seconds). One can observe from Table 3 that all the algorithms performs quite well for these easy-to-solve test instances. Our algorithm Ssnal is generally faster than mfIPM, APG, LADMM while comparable with the fastest algorithm FPC_AS.

Table 3: The performance of Ssnal, mfIPM, FPC_AS, APG and LADMM on srcsep1 and srcsep2 (accuracy $\epsilon = 10^{-6}$, noise 60dB). m is the sample size and n is the dimension of features. In the table, "a" = Ssnal, "b"= mfIPM, "c"= FPC_AS, "d" = APG and "e"= LADMM, respectively. "nnz" denotes the number of nonzeros in the solution obtained by Ssnal. The computation time is in the format of "hours:minutes:seconds".

	iteration					η					time				
λ_c; nnz	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
srcsep1, $m = 29166$, $n = 57344$, $\lambda_{\max}(\mathcal{A}\mathcal{A}^*) = 3.56$															
0.16 ; 380	14	28	42	401	89	6.4-7	9.0-7	2.7-8	5.3-7	9.5-7	07	09	05	15	07
0.10 ; 726	16	38	42	574	161	4.7-7	5.9-7	2.1-8	2.8-7	9.6-7	11	15	05	22	13
0.06 ; 1402	19	41	63	801	393	1.5-7	1.5-7	5.4-8	6.3-7	9.9-7	18	18	10	30	32
0.03 ; 2899	19	56	110	901	337	2.7-7	9.4-7	7.3-8	9.3-7	9.9-7	28	53	16	33	28
0.01 ; 6538	17	88	223	1401	542	7.1-7	5.7-7	1.1-7	9.9-7	9.9-7	1:21	2:15	34	53	45
srcsep2, $m = 29166$, $n = 86016$, $\lambda_{\max}(\mathcal{A}\mathcal{A}^*) = 4.95$															
0.16 ; 423	15	29	42	501	127	3.2-7	2.1-7	1.9-8	4.9-7	9.4-7	14	13	07	25	15

Table 3: The performance of SSNAL, mfIPM, FPC_AS, APG and LADMM on srcsep1 and srcsep2 (accuracy $\epsilon = 10^{-6}$, noise 60dB). m is the sample size and n is the dimension of features. In the table, "a" = SSNAL, "b"= mfIPM, "c"= FPC_AS, "d" = APG and "e"= LADMM, respectively. "nnz" denotes the number of nonzeros in the solution obtained by SSNAL. The computation time is in the format of "hours:minutes:seconds".

	iteration					η					time				
λ_c; nnz	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
0.10 ; 805	16	37	84	601	212	8.8-7	9.2-7	2.6-8	9.6-7	9.7-7	21	19	16	30	26
0.06 ; 1549	19	40	84	901	419	1.4-7	3.1-7	5.4-8	4.7-7	9.9-7	32	28	18	44	50
0.03 ; 3254	20	69	128	901	488	1.3-7	6.6-7	8.9-7	9.4-7	9.9-7	1:06	1:33	26	44	59
0.01 ; 7400	21	94	259	2201	837	8.8-7	4.0-7	9.9-7	8.8-7	9.3-7	1:42	5:33	59	2:05	1:43

In Table 4, we report the numerical results obtained by SSNAL, mfIPM, FPC_AS, APG and LADMM in solving various instances of the Lasso problem (1). For simplicity, we only test two cases with $\lambda_c = 10^{-3}$ and 10^{-4}. We can observe that FPC_AS performs very well when it succeeds in obtaining a solution with the desired accuracy. However it is not robust in that it fails to solve 4 out of 8 and 5 out of 8 problems when $\lambda_c = 10^{-3}$ and 10^{-4}, respectively. For a few cases, FPC_AS can only achieve a poor accuracy (10^{-1}). The same non-robustness also appears in the performance of APG. This non-robustness is in fact closely related to the value of $\lambda_{\max}(\mathcal{A}\mathcal{A}^*)$. For example, both FPC_AS and APG fail to solve a rather small problem **blknheavi** ($m = n = 1024$) whose corresponding $\lambda_{\max}(\mathcal{A}\mathcal{A}^*) = 709$. On the other hand, LADMM, SSNAL and mfIPM can solve all the test instances successfully. Nevertheless, in some cases, LADMM requires much more time than SSNAL. One can also observe that for large scale problems, SSNAL outperforms mfIPM by a large margin (sometimes up to a factor of 100). This also demonstrates the power of SSN based augmented Lagrangian methods over interior-point methods in solving large scale problems. Due to the high sensitivity of the first order algorithms to $\lambda_{\max}(\mathcal{A}\mathcal{A}^*)$, one can safely conclude that the first order algorithms can only be used to solve relatively easy problems. Moreover, in order to obtain efficient and robust algorithms for Lasso problems or more general convex composite optimization problems, it is necessary to carefully exploit the second order information in the algorithmic design.

Table 4: The performance of SSNAL, mfIPM, FPC_AS, APG and LADMM on 8 selected sparco problems (accuracy $\epsilon = 10^{-6}$, noise 60dB). m is the sample size and n is the dimension of features. In the table, "a" = SSNAL, "b" = mfIPM, "c" = FPC_AS, "d" = APG and "e" = LADMM, respectively. "nnz" denotes the number of nonzeros in the solution obtained by SSNAL. The computation time is in the format of "hours:minutes:seconds".

	λ_c	nnz	η					time				
probname $m; n$			a	b	c	d	e	a	b	c	d	e
blknheavi	10^{-3}	12	5.7-7	9.2-7	1.3-1	2.0-6	9.9-7	01	01	55	08	06
1024;1024	10^{-4}	12	9.2-8	8.7-7	4.6-3	8.5-5	9.9-7	01	01	49	07	07
srcsep1	10^{-3}	14066	1.6-7	7.3-7	9.7-7	8.7-7	9.7-7	5:41	42:34	13:25	1:56	4:16
29166;57344	10^{-4}	19306	9.8-7	9.5-7	9.9-7	9.9-7	9.5-7	9:28	3:31:08	32:28	2:50	13:06
srcsep2	10^{-3}	16502	3.9-7	6.8-7	9.9-7	9.7-7	9.8-7	9:51	1:01:10	16:27	2:57	8:49
29166;86016	10^{-4}	22315	7.9-7	9.5-7	1.0-3	9.6-7	9.5-7	19:14	6:40:21	2:01:06	4:56	16:01
srcsep3	10^{-3}	27314	6.1-7	9.6-7	9.9-7	9.6-7	9.9-7	33	6:24	8:51	47	49
196608;196608	10^{-4}	83785	9.7-7	9.9-7	9.9-7	9.9-7	9.9-7	2:03	1:42:59	3:40	1:15	3:06
soccer1	10^{-3}	4	1.8-7	6.3-7	5.2-1	8.4-7	9.9-7	01	03	13:51	2:35	02
3200;4096	10^{-4}	8	8.7-7	4.3-7	5.2-1	3.3-6	9.6-7	01	02	13:23	3:07	02

Table 4: The performance of Ssnal, mfIPM, FPC_AS, APG and LADMM on 8 selected sparco problems (accuracy $\epsilon = 10^{-6}$, noise 60dB). m is the sample size and n is the dimension of features. In the table, "a" = Ssnal, "b" = mfIPM, "c" = FPC_AS, "d" = APG and "e" = LADMM, respectively. "nnz" denotes the number of nonzeros in the solution obtained by Ssnal. The computation time is in the format of "hours:minutes:seconds".

	λ_c	nnz	η	time
probname			a | b | c | d | e	a | b | c | d | e
$m;n$				
soccer2	10^{-3}	4	3.4-7 | 6.3-7 | 5.0-1 | 8.2-7 | 9.9-7	00 | 03 | 13:46 | 1:40 | 02
3200;4096	10^{-4}	8	2.1-7 | 1.4-7 | 6.8-1 | 1.8-6 | 9.1-7	01 | 03 | 13:27 | 3:07 | 02
blurrycam	10^{-3}	1694	1.9-7 | 6.5-7 | 3.6-8 | 4.1-7 | 9.4-7	03 | 09 | 03 | 02 | 07
65536;65536	10^{-4}	5630	1.0-7 | 9.7-7 | 1.3-7 | 9.7-7 | 9.9-7	05 | 1:35 | 08 | 03 | 29
blurspike	10^{-3}	1954	3.1-7 | 9.5-7 | 7.4-4 | 9.9-7 | 9.9-7	03 | 05 | 6:38 | 03 | 27
16384;16384	10^{-4}	11698	3.5-7 | 7.4-7 | 8.3-5 | 9.8-7 | 9.9-7	10 | 08 | 6:43 | 05 | 35

5 Conclusion

In this paper, we have proposed an inexact augmented Lagrangian method of an asymptotic superlinear convergence rate for solving the large scale convex composite optimization problems of the form (**P**). It is particularly well suited for solving ℓ_1-regularized least squares (LS) problems. With the intelligent incorporation of the semismooth Newton method, our algorithm Ssnal is able to fully exploit the second order sparsity of the problems. Numerical results have convincingly demonstrated the remarkable efficiency and robustness of our algorithm in solving large scale ℓ_1-regularized LS problems. Based on extensive numerical evidence, we firmly believe that our algorithmic framework can be adapted to design robust and efficient solvers for various large scale convex composite optimization problems.

Acknowledgments

The authors would like to thank Dr. Ying Cui at National University of Singapore and Dr. Chao Ding at Chinese Academy of Sciences for numerous discussions on the error bound conditions and the metric subregularity.

References

[1] A. Y. Aravkin, J. V. Burke, D. Drusvyatskiy, M. P. Friedlander, S. Roy, *Level-set methods for convex optimization*, arXiv:1602.01506, 2016.

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

[3] S. Becker, J. Bobin, and E. J. Candès, *NESTA: A fast and accurate first-order method for sparse recovery*, SIAM J. Imaging Sciences, 4 (2011), pp. 1–39.

[4] E. van den Berg and M. P. Friedlander, *Probing the Pareto frontier for basis pursuit solutions*, SIAM J. Scientific Computing, 31 (2008), pp. 890–912.

[5] E. van den Berg, M.P. Friedlander, G. Hennenfent, F.J. Herrman, R. Saab, and Ö. Yilmaz, *Sparco: A testing framework for sparse reconstruction*, ACM Trans. Math. Softw. 35 (2009), pp. 1–16.

[6] R. H. Byrd, G. M. Chin, J. Nocedal, and F. Oztoprak, *A family of second-order methods for convex ℓ_1-regularized optimization*, Mathematical Programming, 159 (2016), pp. 435–467.

[7] R. H. Byrd, J. Nocedal, and F. Oztoprak, *An inexact successive quadratic approximation method for L-1 regularized optimization*, Mathematical Programming, 157 (2016), pp. 375–396.

[8] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology, 2 (2011), pp. 27:1–27:27.

[9] S. S. Chen, D. L. Donoho, and, M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Scientific Computing, 20 (1998), pp. 33–61.

[10] F. Clarke, *Optimization and Nonsmooth Analysis*, John Wiley and Sons, New York, 1983.

[11] A. L. Dontchev and R. T. Rockafellar, *Characterizations of Lipschitzian stability in nonlinear programming*, Mathematical programming with data perturbations, Lecture notes in pure and applied mathematics, Dekker, 195 (1998), pp. 65–82.

[12] A. L. Dontchev and R. T. Rockafellar, *Implicit Functions and Solution Mappings*, Springer Monographs in Mathematics, Springer 2009.

[13] F. Facchinei and J.-S. Pang, *Finite-dimensional Variational Inequalities and Complementarity Problems*, Springer, New York, 2003.

[14] J. Fan, H. Fang, and L. Han, *Challenges of big data analysis*, National Science Review, 1 (2014), pp. 293–314.

[15] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE J. Selected Topics in Signal Processing, 1 (2007), pp. 586–597.

[16] A. Fischer, *Local behavior of an iterative framework for generalized equations with nonisolated solutions*, Mathematical Programming, 94 (2002), pp. 91–124.

[17] K. Fountoulakis, J. Gondzio, and P. Zhlobich, *Matrix-free interior point method for compressed sensing problems*, Mathematical Programming Computation, 6 (2014), pp. 1–31.

[18] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximations*, Comput. Math. Appl. 2 (1976), pp. 17–40.

[19] R. Glowinski and A. Marroco, *Sur lapproximation, par elements finis dordre un, et la resolution, par penalisation-dualite, dune classe de problemes de Dirichlet non lineares*, Revue Francaise dAutomatique, Informatique et Recherche Operationelle, 9 (R-2) (1975), pp. 41–76.

[20] R. Goebel and R. T. Rockafellar, *Local strong convexity and local Lipschitz continuity of the gradient of convex functions*, J. Convex Analysis, 15 (2008), pp. 263–270.

[21] G. Golub and C. F. Van Loan, *Matrix Computations*, 3nd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[22] 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, 2010, pp. 883–891.

[23] A. F. Izmailov, A. S. Kurennoy, M. V. Solodov, *A note on upper Lipschitz stability, error bounds, and critical multipliers for Lipschitz-continuous KKT systems*, Mathematical Programming, 142 (2013), pp. 591–604.

[24] K. Jiang, D. F. Sun, and K.-C. Toh, *A partial proximal point algorithm for nuclear norm regularized matrix least squares problems*, Mathematical Programming Computation, 6 (2014), pp. 281-325.

[25] N. Keskar, J. Nocedal, F. Oztoprak, and A. Wachter, *A second-order method for convex $\ell 1$-regularized optimization with active-set prediction*, Optimization Methods and Software, 31 (2016), pp. 605-621.

[26] D. Klatte, *Upper Lipschitz behavior of solutions to perturbed* $\mathrm{C}^{1,1}$ *optimization problems*, Mathematical Programming, 88 (2000), pp. 169–180.

[27] S. Kogan, D. Levin, B. R. Routledge, J. S. Sagi, and N. A. Smith, *Predicting risk from financial reports with regression*, NAACL-HLT 2009, Boulder, CO, May-June 2009.

[28] J. D. Lee, Y. Sun, and M. A. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM J. on Optimization, 24 (2014), pp. 1420–1443.

[29] M. Lichman, *UCI Machine Learning Repository*, `http://archive.ics.uci.edu/ml/datasets.html`.

[30] X. D. Li, D. F. Sun, and K.-C. Toh, *QSDPNAL: A two-phase proximal augmented Lagrangian method for convex quadratic semidefinite programming*, arXiv:1512.08872, 2015.

[31] J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning with Efficient Projections*, Arizona State University, 2009.

[32] Z.-Q. Luo and P. Tseng, *On the linear convergence of descent methods for convex essentially smooth minimization*, SIAM J. Control and Optimization, 30 (1992), pp. 408–425.

[33] F. J. Luque, *Asymptotic convergence analysis of the proximal point algorithm*, SIAM J. Control and Optimization, 22 (1984), pp. 277–293.

[34] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control and Optimization, 15 (1977), pp. 959–972.

[35] A. Milzarek and M. Ulbrich, *A semismooth Newton method with multidimensional filter globalization for ℓ_1-optimization*, SIAM J. Optimization, 24 (2014), pp. 298–333.

[36] Y. Nesterov, *A method of solving a convex programming problem with convergence rate $O(1/k^2)$*, Soviet Mathematics Doklady 27 (1983), pp. 372–376.

[37] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Mathematical Programming, 58 (1993), pp. 353–367.

[38] S. M. Robinson, *Some continuity properties of polyhedral multifunctions*, in Mathematical Programming at Oberwolfach, vol. 14 of Mathematical Programming Studies, Springer Berlin Heidelberg, 1981, pp. 206–214.

[39] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, N.J., 1970.

[40] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control and Optimization, 14 (1976), pp. 877–898.

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

[42] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, vol. 317 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer-Verlag, Berlin, 1998.

[43] D. F. Sun and J. Sun, *Semismooth matrix-valued functions*, Mathematics of Operations Research, 27 (2002), pp. 150–169.

[44] J. Sun, *On monotropic piecewise quadratic programming*, PhD thesis, Department of Mathematics, University of Washington, 1986.

[45] P. Tseng and S. Yun, *A coordinate gradient descent method for nonsmooth separable minimization*, Mathematical Programming, 125 (2010), pp. 387–423.

[46] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Royal Statistical Society: Series B, 58 (1996), pp. 267–288.

[47] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation*, SIAM J. Scientific Computing, 32 (2010), pp. 1832–1857.

[48] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Singal Process., 57 (2009), pp. 2479–2493.

[49] X. Xiao, Y. Li, Z. Wen, and L. Zhang, *Semi-smooth second-order type methods for composite convex programs*, arXiv:1603.07870, 2016.

[50] L. Yang, D. F. Sun, and K.-C. Toh, *SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints*, Mathematical Programming Computation, 7 (2015), pp. 331–366.

[51] M.-C. Yue, Z. Zhou, and A. M.-C. So, *Inexact regularized proximal Newton method: provable convergence guarantees for non-smooth convex minimization without strong convexity*, arXiv:1605.07522, 2016.

[52] X. Zhao, D. F. Sun, and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optimization, 20 (2010), pp. 1737–1765.

[53] X. Zhang, M. Burger and S. Osher, *A unified primal-dual algorithm framework based on Bregman iteration*, J. Scientific Computing, 49 (2011), pp. 20–46.

[54] H. Zhou, *The adaptive Lasso and its oracle properties*, J. American Statistical Association, 101 (2006), pp. 1418–1429.

[55] H. Zhou and T. Hastie, *Regularization and variable selection via the elastic net*, J. Royal Statistical Society, Series B, 67 (2005), pp. 301–320.