

# QUADRATIC CONVERGENCE OF SMOOTHING NEWTON'S METHOD FOR 0/1 LOSS OPTIMIZATION\*

SHENGLONG ZHOU<sup>†</sup>, LILI PAN<sup>‡</sup>, NAIHUA XIU<sup>§</sup>, AND HOU-DUO QI<sup>¶</sup>

**Abstract.** It has been widely recognized that the 0/1-loss function is one of the most natural choices for modelling classification errors, and it has a wide range of applications including support vector machines and 1-bit compressed sensing. Due to the combinatorial nature of the 0/1-loss function, methods based on convex relaxations or smoothing approximations have dominated the existing research and are often able to provide approximate solutions of good quality. However, those methods are not optimizing the 0/1-loss function directly and hence no optimality has been established for the original problem. This paper aims to study the optimality conditions of the 0/1 function minimization, and for the first time to develop Newton's method that directly optimizes the 0/1 function with a local quadratic convergence under reasonable conditions. Extensive numerical experiments demonstrate its superior performance as one would expect from Newton-type methods.

**Key words.** 0/1-loss function, optimality conditions, Newton's method, locally quadratic convergence, superior numerical performance

**AMS subject classifications.** 49M05, 90C26, 90C30, 65K05

**1. Introduction.** This paper is concerned with the 0/1-loss optimization:

$$(1.1) \quad \min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) + \lambda \|(\mathbf{A}\mathbf{x} + \mathbf{b})_+\|_0,$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is twice continuously differentiable,  $\lambda > 0$  is a penalty parameter and  $A \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ . Moreover,  $\mathbf{z}_+ := ((z_1)_+, \dots, (z_m)_+)^{\top}$  with  $z_+ := \max\{z, 0\}$  and  $\|\mathbf{z}\|_0$  is the  $\ell_0$  norm of  $\mathbf{z}$ , counting the number of its non-zero entries. Hence,  $\|\mathbf{z}_+\|_0$  counts the number of positive entries of  $\mathbf{z}$ , i.e.,  $\|\mathbf{z}_+\|_0 = \sum_{i=1}^m \ell_{0/1}(z_i)$ , where

$$\ell_{0/1}(z) = \begin{cases} 1, & z > 0, \\ 0 & z \leq 0. \end{cases}$$

The function  $\ell_{0/1}(\cdot)$  is known as the Heaviside step function (or the unit step function) in [41, 13] or simply the 0/1-loss function in [17, 19, 6]. It plays an active role in many applications including support vector machines (SVM) [11], the one-bit compressed sensing [5], the maximum rank correlation [18], and the problem of area under curves [31]. However, optimization related to the 0/1-loss function is NP-hard, see [4, 16].

A vast body of work has developed algorithms for optimization involving the 0/1-loss function by making use of its continuous surrogates. A major concern on this part of research is that convergence analysis is often conducted on the surrogate problems rather than on the original ones that involve 0/1-loss functions. On the other hand, there also exists a large body of research that addresses the 0/1-loss optimization directly by taking advantage of the intrinsic appealing feature of the loss function, which captures the discrete nature of the binary classification. We will review two classes of such methods below.

The first class consists of mixed integer programming (MIP), which has become a leading approach to directly optimizing the 0/1-loss function, see [28, 1] for some earlier work. It is straightforward to relate the 0/1 loss to the misclassification minimization for discrimination problems [36, 7]. This

---

\*Received by the editors April 1, 2021; accepted for publication (in revised form) September 5, 2021; published electronically December 13, 2021.

<https://doi.org/10.1137/21M1409445>

**Funding:** This work was funded by the the National Science Foundation of China (11971052, 11801325, 11771255) and Young Innovation Teams of Shandong Province (2019KJJI013).

<sup>†</sup>Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, United Kingdom. ([shenglong.zhou@imperial.ac.uk](mailto:shenglong.zhou@imperial.ac.uk))

<sup>‡</sup>Department of Mathematics, Shandong University of Technology, Zibo 255049, People's Republic of China. ([panlili1979@163.com](mailto:panlili1979@163.com)).

<sup>§</sup>Department of Applied Mathematics, Beijing Jiaotong University, Beijing 10044, People's Republic of China. ([nhxiu@bjtu.edu.cn](mailto:nhxiu@bjtu.edu.cn)).

<sup>¶</sup>School of Mathematics, University of Southampton, Southampton SO17 1BJ, United Kingdom. ([h.qi@soton.ac.uk](mailto:h.qi@soton.ac.uk)).

approach is in general effective with a major issue of scalability for large-sized problems. Much progress has also been made in improving the scalability of MIP by employing various strategies of reducing the problem sizes. Those include, for instance, decomposition strategy [36], local search [32], and convex hull cuts in a branch-and-bound framework [6], to name a few. Some recent work includes [38, 39], where different integer programming reformulations for the 0/1-loss minimizations are built and then tackled via the modern commercial MIP solvers, such as Gurobi and CPLEX. Despite those progresses, the speed of computation is still a bottleneck for the MIP approach

The second class of methods comes from continuous optimization. Since the 0/1-loss function is non-convex, non-differentiable and has zero gradients whenever differentiable, coordinate descent directions are natural choices for decreasing the objective. There are a number of such methods including random coordinate descent algorithms [27], greedy coordinate descent algorithms [45], and stochastic coordinate descent heuristic [43]. Convergence for this class of algorithms is often established in the probabilistic sense. Other types include a column generation approach [8] and an alternating direction method of multipliers [40], which is devoted to the 0/1-loss regularized SVM problem.

This paper aims to extend the classical Newton method to (1.1) and to prove its local quadratic convergence. The investigation of Newton's method is motivated and supported by the following facts.

- (i) Newton's method has been recently developed by the authors in [46] for optimization problems with a sparse constraint  $\|\mathbf{x}\|_0 \leq s$ . Its performance is outstanding in comparison with a number of leading solvers that employ either hard- or soft-thresholding techniques. The essential difference of problem (1.1) from that in [46] is that our operator is a composite one that involves the operators  $(\cdot)_+$ ,  $\|\cdot\|_0$ , and the linear classification inequalities  $\mathbf{A}\mathbf{x} - \mathbf{b} \geq 0$ . Because of this, the framework developed in [46] cannot be used here. However, the success in [46] naturally leads us to investigate what form a Newton's method would take for (1.1) and whether it is computationally efficient.
- (ii) In some important applications, the objective function  $f(\mathbf{x})$  is separable in the following form:

$$(1.2) \quad f(\mathbf{x}) = \sum_{i=1}^M f_i(\mathbf{x}_{(i)}),$$

where each  $\mathbf{x}_{(i)}$  ( $i = 1, \dots, M$ ) is a subvector of  $\mathbf{x}$  and not overlapping with each other. Consequently, the Hessian of  $f(\mathbf{x})$  is block-diagonal. When each block is of small size, the inverse of Hessian (when exists) can be fast computed. In the particular applications of SVM and one-bit compressed sensing, the block-size is 1 and the Hessian matrix is hence diagonal. One can imagine that Newton's method would be extremely efficient for such applications.

- (iii) Although it is challenging to design a gradient-type method for the 0/1-loss function due to its zero gradient (when exists), we would like to emphasize that it is easy to compute the proximal operator of the 0/1-loss function. Proximal operators have long been known to be closely related to optimality conditions in constrained optimization. In particular, the proximal operator of the zero norm  $\|\cdot\|_0$  characterizes a class of stationary points for sparse optimization and many hard- and soft-thresholding algorithms actually converge to such stationary points, see Beck and Eldar [2] for an excellent illustration.

Our first step towards developing Newton's method is to establish a stationary equation of the type:

$$(1.3) \quad F(\mathbf{w}; T) := \begin{bmatrix} \nabla f(\mathbf{x}) + A_T^\top \mathbf{z}_T \\ A_T \mathbf{x} + \mathbf{b}_T \\ \mathbf{z}_{\bar{T}} \end{bmatrix} = 0,$$

where  $\mathbf{w}^\top := (\mathbf{x}; \mathbf{z})$  with  $\mathbf{z} := \mathbf{A}\mathbf{x} + \mathbf{b}$ ,  $T \subseteq \{1, \dots, m\}$  is an index set with  $\bar{T}$  being its complementary set, and  $A_T$  consists of the rows in  $A$  indexed by  $T$ . This equation is characterized by the proximal operator of the 0/1-loss function at a local minimum of (1.1). We call it the  $P$ -stationary (abbreviation for Proximal-stationary) equation. See Theorem 3.3 and Equation (4.2) for more details. The index set  $T$  depends on the optimal solution  $\mathbf{w}^*$  and hence is unknown.

The second step is to construct a scheme that defines  $T_k$  at a given point  $\mathbf{w}^k$  and approximate the true  $T$ . The Newton step is to solve the equation  $F(\mathbf{w}; T_k) = 0$  for the next iterate  $\mathbf{w}^{k+1}$ . Such a computational scheme for  $T_k$  is described in (4.1) and (4.5). However, the difficulty is that there

is no guarantee that this scheme will be able to identify the correct  $T$ . In other words, we may be encountered with different  $T_k$  each iteration no matter how close our iterate is to the optimal solution. This is where the convergence theory of classical Newton's method fails to go through. Now we introduce a practically important technique of smoothing motivated by Chen et al. [10]. Instead of solving the equation  $F(\mathbf{w}; T_k) = 0$ , we try to solve its perturbed version:

$$(1.4) \quad F_{\mu_k}(\mathbf{w}; T_k) := F(\mathbf{w}; T_k) + \begin{bmatrix} 0 \\ -\mu_k \mathbf{z}_{T_k} \\ 0 \end{bmatrix} = 0,$$

where the smoothing parameter  $\mu_k > 0$  will be properly chosen as in (4.8). Its Jacobian matrix has the following structure

$$\nabla F_{\mu_k}(\mathbf{w}; T_k) := \begin{bmatrix} \nabla^2 f(\mathbf{x}) & A_{T_k}^\top & 0 \\ A_{T_k} & -\mu_k I & 0 \\ 0 & 0 & I \end{bmatrix}$$

and it is nonsingular if and only if the matrix  $(\nabla^2 f(\mathbf{x}) + A_{T_k}^\top A_{T_k} / \mu_k)$  is nonsingular (i.e., the Schur complement of  $(-\mu_k I)$  in the top  $2 \times 2$  block is nonsingular). We note that nonsingularity may still hold even if  $f(\cdot)$  is not convex provided that  $A_{T_k}$  has full row-rank. Due to its connection to [10], we call our method a smoothing Newton method. We also like to point out another interesting connection. When  $\mu_k$  is fixed, our algorithmic framework is analogous to the primal-dual active-set algorithms extensively studied in [20, 24, 14], whose main targets are quadratic objective functions.

Our last step is to establish the bounds

$$\|F(\mathbf{w}^{k+1}; T_{k+1})\| = C_1 \|F(\mathbf{w}^k; T_k)\|^2, \quad \|\mathbf{w}^{k+1} - \mathbf{w}^*\| = C_2 \|\mathbf{w}^k - \mathbf{w}^*\|^2,$$

where the constants  $C_1$  and  $C_2$  only depend on the optimal solution  $\mathbf{w}^*$ . Those bounds imply the quadratic convergence of our smoothing Newton method provided that the initial point is close to  $\mathbf{w}^*$  and complete our theoretical investigation, see Theorem 4.8. The efficiency of the Newton method is confirmed through extensive numerical experiments including 40 SVM problems from real data (16 of them for  $m \leq n$  and 24 for  $m > n$ ) and simulated 1-bit compressing data against some existing solvers. As far as we know, this is the first Newton-type method for the 0/1-loss optimization (1.1).

This paper is organized as follows. In the next section, we analyze the 0/1-loss function, calculating its subdifferentials and the proximal operator. In Section 3, we establish the first-order necessary and sufficient optimality conditions of the problem (1.1) through the proximal operator of 0/1-loss function, leading to the well-defined  $P$ -stationary points. In Section 4, we reformulate the  $P$ -stationary condition as a system of nonlinear equations and develop Newton's method with the promised quadratic convergence. In Section 5, we conduct extensive numerical experiments to demonstrate the outstanding performance of Newton's method against a few leading solvers for the problems of SVM and 1-bit compressed sensing. We conclude the paper in Section 6.

**2. Preliminaries.** We first list some notation that is frequently used throughout the paper. Given a subset  $T \subseteq \mathbb{N}_m := \{1, 2, \dots, m\}$ , its cardinality and complementary set are  $|T|$  and  $\bar{T}$ . The neighbourhood of  $\mathbf{x} \in \mathbb{R}^n$  with a radius  $\delta > 0$  is denoted by  $N(\mathbf{x}, \delta) = \{\mathbf{v} \in \mathbb{R}^n : \|\mathbf{v} - \mathbf{x}\| < \delta\}$ . Moreover,  $\mathbf{x}_T$  (resp.  $A_T$ ) represents the sub-vector (resp. sub-matrix) contains elements (resp. rows) of  $\mathbf{x}$  indexed on  $T$ . Particularly,  $A_i$  is the  $i$ th row of  $A$ . We combine two vectors as  $(\mathbf{x}; \mathbf{y}) := (\mathbf{x}^\top \ \mathbf{y}^\top)^\top$ . The  $i$ th largest singular value of  $H \in \mathbb{R}^{n \times n}$  is written  $\sigma_i(H)$ , namely  $\sigma_1(H) \geq \sigma_2(H) \geq \dots \geq \sigma_n(H)$ . Particularly, we write  $\|H\| := \sigma_1(H)$  and  $\sigma_{\min}(H) := \sigma_n(H)$ . Finally, let  $I$  be the identity matrix and  $\mathbf{1}$  be the vector with all entries being ones.

Next we describe the formula for computing the subdifferential of  $\|\mathbf{z}_+\|_0$  and its proximal operator. We note that  $\|\cdot\|_0$  is lower semi-continuous (lsc) and  $\mathbf{z}_+$  is obviously continuous, the composition  $\|\mathbf{z}_+\|_0$  is also lsc. For a proper and lsc function  $g : \mathbb{R}^m \mapsto \mathbb{R}$ , its subdifferential  $\partial g(\cdot)$  is well defined as in [35, Definition 8.3]. The following results are easy to prove after simple calculations.

LEMMA 2.1. (i) *We have*

$$(2.1) \quad \partial \|\mathbf{y}_+\|_0 = \left\{ \mathbf{v} \in \mathbb{R}^m : v_i \begin{cases} \geq 0, & y_i = 0, \\ = 0, & y_i \neq 0, \end{cases} \quad i \in \mathbb{N}_m \right\}.$$

(ii) Let  $g(\mathbf{x}) := \|(h(\mathbf{x}))_+\|_0$ , where  $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is differentiable, and  $\Gamma := \{i \in \mathbb{N}_m : h_i(\mathbf{x}) = 0\}$  for a given a point  $\mathbf{x} \in \mathbb{R}^n$ . If

$$(2.2) \quad \forall \mathbf{z} \in \mathbb{R}^{|\Gamma|}, \quad \mathbf{z} \geq 0, \quad (\nabla h(\mathbf{x}))_\Gamma^\top \mathbf{z} = 0 \quad \implies \quad \mathbf{z} = 0,$$

then the subdifferential of  $g(\mathbf{x})$  at  $\mathbf{x}$  is

$$(2.3) \quad \partial g(\mathbf{x}) = \nabla h(\mathbf{x})^\top \partial \|(h(\mathbf{x}))_+\|_0.$$

*Proof.* i) It is straight to check that the regular subdifferential  $\widehat{\partial}\|\mathbf{y}_+\|_0$  (see [35, Definition 8.3]) of  $\|\mathbf{y}_+\|_0$  at  $\mathbf{y}$  takes the following form,

$$(2.4) \quad \widehat{\partial}\|\mathbf{y}_+\|_0 = \left\{ \mathbf{v} \in \mathbb{R}^m : v_i \begin{cases} \geq 0, & y_i = 0, \\ = 0, & y_i \neq 0, \end{cases} i \in \mathbb{N}_m \right\} =: \Omega(\mathbf{y}).$$

We next verify  $\partial\|\mathbf{y}_+\|_0 = \Omega(\mathbf{y})$ . By letting  $\varphi(\cdot) := \|(\cdot)_+\|_0$ , we have

$$(2.5) \quad \begin{aligned} \partial\|\mathbf{y}_+\|_0 &= \limsup_{\mathbf{z} \xrightarrow{\varphi} \mathbf{y}} \widehat{\partial}\|\mathbf{z}_+\|_0 \quad (\text{by [35, Equation 8(5)]}) \\ &= \limsup_{\mathbf{z} \xrightarrow{\varphi} \mathbf{y}} \Omega(\mathbf{z}) \quad (\text{by (2.4)}) \\ &= \{ \mathbf{v} \in \mathbb{R}^m : \exists \mathbf{z}^k \xrightarrow{\varphi} \mathbf{y}, \mathbf{v}^k \rightarrow \mathbf{v} \text{ with } \mathbf{v}^k \in \Omega(\mathbf{z}^k) \} =: \Theta, \end{aligned}$$

where  $\mathbf{z} \xrightarrow{\varphi} \mathbf{y}$  represents  $\mathbf{z} \rightarrow \mathbf{y}, \varphi(\mathbf{z}) \rightarrow \varphi(\mathbf{y})$ . Clearly,  $\Omega(\mathbf{y}) \subseteq \Theta$ . On the other hand,  $\Theta \subseteq \Omega(\mathbf{y})$  follows from that  $\Omega(\mathbf{z}^k) \subseteq \Omega(\mathbf{y})$  for any  $\mathbf{z}^k \xrightarrow{\varphi} \mathbf{y}$  and  $\Omega(\cdot)$  is closed.

ii) Direct verifications yield the following chain of equations,

$$\begin{aligned} \partial^\infty\|\mathbf{y}_+\|_0 &= \limsup_{\sigma \downarrow 0, \mathbf{z} \xrightarrow{\varphi} \mathbf{y}} \sigma \widehat{\partial}\|\mathbf{z}_+\|_0 \quad (\text{by [35, Equation 8(5)]}) \\ &= \limsup_{\mathbf{z} \xrightarrow{\varphi} \mathbf{y}} \widehat{\partial}\|\mathbf{z}_+\|_0 \quad (\text{by (2.4)}) \\ &= \partial\|\mathbf{y}_+\|_0, \quad (\text{by (2.5)}) \end{aligned}$$

where  $\partial^\infty\|\mathbf{y}_+\|_0$  is the horizon subdifferential of  $\|\mathbf{y}_+\|_0$ . Therefore, we derive that  $\widehat{\partial}\varphi = \partial\varphi = \partial^\infty\varphi$ . One can easily prove that the horizon cone  $\widehat{\partial}\varphi(\mathbf{y})^\infty$  (see [35, Definition 3.3]) of  $\widehat{\partial}\varphi(\mathbf{y})$  satisfies  $\widehat{\partial}\varphi(\mathbf{y})^\infty = \partial^\infty\varphi(\mathbf{y})$ . These conditions indicate that the function  $\varphi$  is regular by [35, Corollary 8.11], which together with (2.2),  $g(\mathbf{x}) = \varphi(h(\mathbf{x}))$  and [35, Theorem 10.6] derives (2.3) immediately.  $\square$

The assumption in (2.2) can be regarded as a constraint qualification for the chain rule in (2.3) to hold. We finish this section with a formula to compute the proximal operator of  $\|(\cdot)_+\|_0$ . Let  $\alpha > 0$ , the proximal operator of  $\alpha\|(\cdot)_+\|_0$  at  $\boldsymbol{\nu}$  is defined by

$$\text{Prox}_{\alpha\|(\cdot)_+\|_0}(\boldsymbol{\nu}) = \underset{\mathbf{y} \in \mathbb{R}^m}{\text{argmin}} \frac{1}{2} \|\mathbf{y} - \boldsymbol{\nu}\|^2 + \alpha\|\mathbf{y}_+\|_0.$$

As shown in [40, Lemma 2.2], the proximal operator admits a closed form as

$$(2.6) \quad \left[ \text{Prox}_{\alpha\|(\cdot)_+\|_0}(\boldsymbol{\nu}) \right]_i = \begin{cases} 0, & \nu_i \in (0, \sqrt{2\alpha}), \\ 0 \text{ or } \nu_i, & \nu_i \in \{0, \sqrt{2\alpha}\}, \\ \nu_i, & \nu_i \in (-\infty, 0) \cup (\sqrt{2\alpha}, \infty). \end{cases}$$

**3. Optimality Conditions.** In this section, we study the optimality conditions of (1.1) and characterize the conditions in terms of Proximal-stationarity (i.e., P-stationarity) using the proximal operator. Those results will lay down the foundation for Newton's method in the next section. For a given point  $\mathbf{x}^* \in \mathbb{R}^n$ , we denote

$$(3.1) \quad \Gamma_* := \{i \in \mathbb{N}_m : A_i \mathbf{x}^* + b_i = 0\}.$$

Our first result is to characterize a local minimizer of (1.1).

LEMMA 3.1. *The following relationships hold for the problem (1.1).*

i) *A local minimizer  $\mathbf{x}^*$  satisfies the following condition if  $A_{\Gamma_*}$  is full row rank,*

$$(3.2) \quad -\nabla f(\mathbf{x}^*) \in A^\top \partial \|(A\mathbf{x}^* + \mathbf{b})_+\|_0.$$

ii) *A point  $\mathbf{x}^*$  satisfying (3.2) is a local minimizer if the function  $f$  is locally convex around  $\mathbf{x}^*$ .*

*Proof.* i) It follows from [35, Theorem 10.1] that a local minimizer of (1.1) must satisfy  $-\nabla f(\mathbf{x}^*) \in \lambda \partial g(\mathbf{x}^*)$ , where  $g(\mathbf{x}) := \|(A\mathbf{x} + \mathbf{b})_+\|_0$ . This together with Lemma 2.1 and  $\lambda \partial \|( \cdot )_+\|_0 = \partial \|( \cdot )_+\|_0$  by (2.1) derives the result immediately.

ii) Since the problem (1.1) is equivalent to the following problem,

$$(3.3) \quad \begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^m} & f(\mathbf{x}) + \lambda \|\mathbf{y}_+\|_0, \\ \text{s.t.} & \quad A\mathbf{x} + \mathbf{b} - \mathbf{y} = 0, \end{aligned}$$

it suffices to show that  $(\mathbf{x}^*; \mathbf{y}^*)$  is a local minimizer of the problem (3.3), where  $\mathbf{x}^*$  satisfies (3.2) and  $\mathbf{y}^* = A\mathbf{x}^* + \mathbf{b}$ , namely, there is a  $\mathbf{z}^*$  such that

$$(3.4) \quad \nabla f(\mathbf{x}^*) + A^\top \mathbf{z}^* = 0, \quad A\mathbf{x}^* + \mathbf{b} - \mathbf{y}^* = 0, \quad \partial \|\mathbf{y}_+\|_0 \ni \mathbf{z}^*.$$

It follows from  $\mathbf{y}^* = A\mathbf{x}^* + \mathbf{b}$  and (3.1) that  $\Gamma_* = \{i \in \mathbb{N}_m : y_i^* = 0\}$ . This together with  $\partial \|\mathbf{y}_+\|_0 \ni \mathbf{z}^*$  and the expression of the  $\partial \|\mathbf{y}_+\|_0$  in (2.1) indicates

$$(3.5) \quad \mathbf{y}_{\Gamma_*}^* = 0, \quad \mathbf{z}_{\Gamma_*}^* \geq 0, \quad \mathbf{y}_{\Gamma_*^c}^* \neq 0, \quad \mathbf{z}_{\Gamma_*^c}^* = 0.$$

Define a radius  $\delta := \min\{\delta_1, \delta_2\}$ , where

$$(3.6) \quad \delta_1 := \begin{cases} +\infty, & A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^* = 0 \\ \frac{\lambda}{\|A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^*\|}, & \text{otherwise,} \end{cases} \quad \delta_2 := \begin{cases} +\infty, & \mathbf{y}^* \leq \mathbf{0}, \\ \min_i \{y_i^* : y_i^* > 0\}, & \text{otherwise,} \end{cases}$$

and consider a local region of  $\mathbf{w}^* := (\mathbf{x}^*; \mathbf{y}^*)$  by

$$(3.7) \quad N(\mathbf{w}^*, \delta) = \{(\mathbf{x}; \mathbf{y}) \in \mathbb{R}^{n+m} : A\mathbf{x} + \mathbf{b} - \mathbf{y} = 0, \|\mathbf{w} - \mathbf{w}^*\| < \delta\}.$$

Indeed,  $N(\mathbf{w}^*, \delta)$  is a neighbourhood of  $\mathbf{w}^*$  since  $A\mathbf{x}^* + \mathbf{b} - \mathbf{y}^* = 0$  from (3.4). Next we show that, for any  $\mathbf{w} \in N(\mathbf{w}^*, \delta)$ ,

$$(3.8) \quad \|\mathbf{y}_+\|_0 \leq \|\mathbf{y}_+\|_0.$$

Obviously, it is true if  $\mathbf{y}^* \leq \mathbf{0}$  as  $\|\mathbf{y}_+\|_0 = 0$ . For  $\mathbf{y}^* \not\leq \mathbf{0}$ , to guarantee (3.8), it suffices to show that for any  $i, y_i^* > 0 \implies y_i > 0$ . Suppose there is a  $j \in \mathbb{N}_m$  such that  $y_j^* > 0$  but  $y_j \leq 0$ . This incurs the following contradiction

$$\begin{aligned} \delta_2 &\geq \delta > \|\mathbf{w} - \mathbf{w}^*\| && \text{(by (3.7))} \\ &\geq |y_j - y_j^*| = y_j^* - y_j \geq y_j^* \geq \delta_2. && \text{(by (3.6))} \end{aligned}$$

Again, for any  $\mathbf{w} \in N(\mathbf{w}^*, \delta)$ , we have  $A\mathbf{x} + \mathbf{b} - \mathbf{y} = 0$ , which and (3.4) generate

$$(3.9) \quad \mathbf{y} - \mathbf{y}^* = A(\mathbf{x} - \mathbf{x}^*).$$

Next, the convexity of  $f$  gives rise to

$$(3.10) \quad \begin{aligned} f(\mathbf{x}) - f(\mathbf{x}^*) &\geq \langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle = -\langle A^\top \mathbf{z}^*, \mathbf{x} - \mathbf{x}^* \rangle && \text{(by (3.4))} \\ &= -\langle A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^*, \mathbf{x} - \mathbf{x}^* \rangle = -\langle A_{\Gamma_*}(\mathbf{x} - \mathbf{x}^*), \mathbf{z}_{\Gamma_*}^* \rangle =: \phi && \text{(by (3.5))} \end{aligned}$$

Now we make the conclusion by two cases. If  $\|\mathbf{y}_+\|_0 = \|\mathbf{y}_+\|_0$ , then  $\mathbf{y}_{\Gamma_*} \leq \mathbf{0}$  due to  $\mathbf{y}_{\Gamma_*}^* = \mathbf{0}$ . This and (3.9) yield that

$$(3.11) \quad 0 \geq \mathbf{y}_{\Gamma_*} = \mathbf{y}_{\Gamma_*} - \mathbf{y}_{\Gamma_*}^* = A_{\Gamma_*}(\mathbf{x} - \mathbf{x}^*),$$

which together with  $\mathbf{z}_{\Gamma_*}^* \geq 0$  from (3.5) indicates  $\phi \geq 0$ , namely  $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ . So

$$f(\mathbf{x}) + \lambda \|\mathbf{y}_+\|_0 \geq f(\mathbf{x}^*) + \lambda \|\mathbf{y}_+^*\|_0.$$

If  $\|\mathbf{y}_+^*\|_0 \neq \|\mathbf{y}_+\|_0$ , we must have  $\|\mathbf{y}_+\|_0 \geq 1 + \|\mathbf{y}_+^*\|_0$  by (3.8). If  $A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^* = 0$ , then  $\phi = 0 > -\lambda$ . Otherwise, it follows

$$\begin{aligned} \phi &\geq -\|A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^*\| \|\mathbf{x} - \mathbf{x}^*\| \geq -\|A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^*\| \|\mathbf{w} - \mathbf{w}^*\| \\ &\geq -\|A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^*\| \delta \geq -\|A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^*\| \delta_1 = -\lambda. \quad (\text{by (3.6)}) \end{aligned}$$

Both lead to  $\phi \geq -\lambda$ , which results in

$$\begin{aligned} f(\mathbf{x}) + \lambda \|\mathbf{y}_+\|_0 &\geq f(\mathbf{x}^*) + \phi + \lambda \|\mathbf{y}_+\|_0 \quad (\text{by (3.10)}) \\ &\geq f(\mathbf{x}^*) - \lambda + \lambda \|\mathbf{y}_+\|_0 \quad (\text{by } \phi \geq -\lambda) \\ &\geq f(\mathbf{x}^*) + \lambda \|\mathbf{y}_+^*\|_0. \quad (\text{by } \|\mathbf{y}_+\|_0 \geq 1 + \|\mathbf{y}_+^*\|_0) \end{aligned}$$

Overall, the two cases show that  $(\mathbf{x}^*; \mathbf{y}^*)$  is a local minimizer to (3.3). Namely,  $\mathbf{x}^*$  is a local minimizer to (1.1).  $\square$

The characterization (3.2) is nice and it is in the classic form of differential inclusion. However, the challenge is that it is difficult to extract second-order information which is essential to Newton's method. To this purpose, we continue to characterize it in terms of P-stationarity.

**DEFINITION 3.2.** *A point  $\mathbf{x}^*$  is called a P-stationary point of the problem (1.1) if there exist a constant  $\tau > 0$  and a point  $\mathbf{z}^* \in \mathbb{R}^m$  such that*

$$(3.12) \quad \begin{cases} \nabla f(\mathbf{x}^*) + A^\top \mathbf{z}^* = 0 \\ \text{Prox}_{\tau\lambda\|(\cdot)_+\|_0}(A\mathbf{x}^* + \mathbf{b} + \tau\mathbf{z}^*) \ni A\mathbf{x}^* + \mathbf{b}. \end{cases}$$

We also say a point  $(\mathbf{x}^*; \mathbf{z}^*)$  is a P-stationary point of the problem (1.1) if it satisfies the conditions in (3.12). For a point  $\mathbf{x}^*$ , we denote two constants by

$$(3.13) \quad \tau_1 := \begin{cases} +\infty, & \mathbf{y}^* \leq 0, \\ \min \left\{ \frac{(y_i^*)^2}{2\lambda} : y_i^* > 0 \right\}, & \text{otherwise,} \end{cases} \quad \tau_2 := \begin{cases} +\infty, & \Gamma_* = \emptyset, \\ \frac{2\lambda}{\max_i |p_i^*|^2}, & \text{otherwise,} \end{cases}$$

where  $\mathbf{y}^* := A\mathbf{x}^* + \mathbf{b}$  and  $\mathbf{p}^* := -(A_{\Gamma_*} A_{\Gamma_*}^\top)^{-1} A_{\Gamma_*}^\top \nabla f(\mathbf{x}^*)$ . Clearly, both  $\tau_1 > 0$  and  $\tau_2 > 0$ . Based on these notation, we have the following main result of this section.

**THEOREM 3.3.** *The following relationships hold for the problem (1.1).*

- i) *A local minimizer  $\mathbf{x}^*$  is a P-stationary point for any  $0 < \tau < \tau_* := \min\{\tau_1, \tau_2\}$  if  $A_{\Gamma_*}$  is full row rank.*
- ii) *A P-stationary point with  $\tau > 0$  is a local minimizer if the function  $f$  is locally convex around  $\mathbf{x}^*$ .*
- iii) *A P-stationary point with  $\tau \geq \|A\|^2/c_f$  is a global minimizer if the function  $f$  is strongly convex with a constant  $c_f > 0$ .*

*Proof.* i) As  $\mathbf{x}^*$  is a local minimizer of (1.1), condition (3.2) is valid by Lemma 3.1 if  $A_{\Gamma_*}$  is full row rank. In other words, there is a  $\mathbf{z}^*$  such that

$$(3.14) \quad \nabla f(\mathbf{x}^*) + A^\top \mathbf{z}^* = 0, \quad A\mathbf{x}^* + \mathbf{b} - \mathbf{y}^* = 0, \quad \mathbf{z}^* \in \partial \|\mathbf{y}_+^*\|_0.$$

Therefore, to show (3.12), we only need to verify that, for any  $0 < \tau < \tau^*$ ,

$$\mathbf{z}^* \in \partial \|\mathbf{y}_+^*\|_0 \Rightarrow \mathbf{y}^* \in \mathbb{P} := \text{Prox}_{\tau\lambda\|(\cdot)_+\|_0}(\mathbf{y}^* + \tau\mathbf{z}^*).$$

Recall the definition of  $\Gamma_*$  in (3.1) and the second condition in (3.14), we have  $\mathbf{y}_{\Gamma_*} = (A\mathbf{x}^* + \mathbf{b})_{\Gamma_*} = 0$ . Same reasoning also allows for obtaining (3.5) due to  $\mathbf{z}^* \in \partial \|\mathbf{y}_+^*\|_0$ . As  $A_{\Gamma_*}$  is full row rank, the first condition in (3.14) and  $A^\top \mathbf{z}^* = A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^*$  derive that

$$\mathbf{z}_{\Gamma_*}^* = \mathbf{p}^*.$$

Now,  $0 < \tau < \tau_* = \min\{\tau_1, \tau_2\}$  in (3.13) results in

$$(3.15) \quad \begin{aligned} y_i^* &\geq \min_{i: y_i^* > 0} y_i^* = \sqrt{2\tau_1\lambda} \geq \sqrt{2\tau_*\lambda} > \sqrt{2\tau\lambda} \quad \text{if } y_i^* > 0, \\ z_i^* &\leq \max_i |p_i^*| = \sqrt{2\lambda/\tau_2} \leq \sqrt{2\lambda/\tau_*} < \sqrt{2\lambda/\tau} \quad \text{if } z_i^* > 0. \end{aligned}$$

These and (3.5) yield the following condition,

$$y_i^* \begin{cases} = 0, & z_i^* = 0 \text{ or } 0 < z_i^* < \sqrt{2\lambda/\tau}, \\ < 0 \text{ or } > \sqrt{2\tau\lambda}, & z_i^* = 0. \end{cases}$$

It is easy to see that the above condition satisfies that

$$(3.16) \quad y_i^* \in \mathbb{P}_i = \begin{cases} 0, & y_i^* + \tau z_i^* \in (0, \sqrt{2\tau\lambda}), \\ 0 \text{ or } y_i^* + \tau z_i^*, & y_i^* + \tau z_i^* \in \{0, \sqrt{2\tau\lambda}\}, \\ y_i^* + \tau z_i^*, & y_i^* + \tau z_i^* \in (-\infty, 0) \cup (\sqrt{2\tau\lambda}, \infty). \end{cases}$$

ii) Note that the second condition in (3.12) means  $\mathbf{y}^* \in \mathbb{P}$ , which by (3.16) implies  $z_i^* = 0$  if  $y_i^* \neq 0$  and  $z_i^* \geq 0$  if  $y_i^* = 0$ , resulting in  $\mathbf{z}^* \in \partial\|\mathbf{y}_+\|_0$  by (2.1). Consequently, we obtain (3.14) and (3.2). The claim follows from Lemma 3.1 ii).

iii) Let  $(\mathbf{x}^*; \mathbf{z}^*)$  be a P-stationary point with  $\tau \geq \|A\|^2/c_f$ . Then the second condition in (3.12) indicates that

$$\begin{aligned} &(1/2\tau)\|A\mathbf{x}^* + \mathbf{b} - (A\mathbf{x}^* + \mathbf{b} + \tau\mathbf{z}^*)\|^2 + \lambda\|(A\mathbf{x}^* + \mathbf{b})_+\|_0 \\ &\leq (1/2\tau)\|A\mathbf{x} + \mathbf{b} - (A\mathbf{x}^* + \mathbf{b} + \tau\mathbf{z}^*)\|^2 + \lambda\|(A\mathbf{x} + \mathbf{b})_+\|_0 \end{aligned}$$

for any  $\mathbf{x} \in \mathbb{R}^n$ , which after simplifying leads to

$$(3.17) \quad \begin{aligned} &\lambda\|(A\mathbf{x}^* + \mathbf{b})_+\|_0 - \|A(\mathbf{x} - \mathbf{x}^*)\|^2/(2\tau) \\ &\leq \lambda\|(A\mathbf{x} + \mathbf{b})_+\|_0 - \langle \mathbf{z}^*, A(\mathbf{x} - \mathbf{x}^*) \rangle \\ &= \lambda\|(A\mathbf{x} + \mathbf{b})_+\|_0 - \langle A^\top \mathbf{z}^*, \mathbf{x} - \mathbf{x}^* \rangle \\ &= \lambda\|(A\mathbf{x} + \mathbf{b})_+\|_0 + \langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle. \quad (\text{by (3.12)}) \end{aligned}$$

The strong convexity of  $f$  implies

$$\begin{aligned} &f(\mathbf{x}) + \lambda\|(A\mathbf{x} + \mathbf{b})_+\|_0 - f(\mathbf{x}^*) - \lambda\|(A\mathbf{x}^* + \mathbf{b})_+\|_0 \\ &\geq (c_f/2)\|\mathbf{x} - \mathbf{x}^*\|^2 + \langle \nabla f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle + \lambda\|(A\mathbf{x} + \mathbf{b})_+\|_0 - \lambda\|(A\mathbf{x}^* + \mathbf{b})_+\|_0 \\ &\geq (c_f/2)\|\mathbf{x} - \mathbf{x}^*\|^2 - 1/(2\tau)\|A(\mathbf{x} - \mathbf{x}^*)\|^2 \quad (\text{by (3.17)}) \\ &\geq (c_f/2 - \|A\|^2/(2\tau))\|\mathbf{x} - \mathbf{x}^*\|^2 \geq 0. \quad (\text{by } \tau \geq \|A\|^2/c_f) \end{aligned}$$

This shows the global optimality of  $\mathbf{x}^*$  to the problem (1.1).  $\square$

**4. Smoothing Newton's Method.** The main purpose of this section is to formulate Newton's method and establish its quadratic convergence. We first state two assumptions for this purpose.

ASSUMPTION 4.1. *Suppose  $f$  is twice continuously differentiable,  $\nabla^2 f(\mathbf{x}^*)$  is positive definite and  $A_{\Gamma_*}$  is full row rank, where  $\Gamma_*$  is given by (3.1).*

ASSUMPTION 4.2. *Suppose  $\nabla^2 f$  is locally Lipschitz continuous around  $\mathbf{x}^*$  with a constant  $L_* > 0$ , namely*

$$\|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{x}')\| \leq L_* \|\mathbf{x} - \mathbf{x}'\|$$

for any  $\mathbf{x}$  and  $\mathbf{x}'$  in the neighbourhood of  $\mathbf{x}^*$ .



**4.1. Stationary equations.** For a point  $\mathbf{w} := (\mathbf{x}; \mathbf{z})$ , we define the sets

$$(4.1) \quad \begin{aligned} \mathcal{S} &:= \left\{ i \in \mathbb{N}_m : A_i \mathbf{x} + b_i + \tau z_i \in (0, \sqrt{2\tau\lambda}) \right\}, \\ \mathcal{E} &:= \left\{ i \in \mathbb{N}_m : A_i \mathbf{x} + b_i + \tau z_i \in \{0, \sqrt{2\tau\lambda}\} \right\}, \\ \mathcal{O} &:= \left\{ i \in \mathbb{N}_m : A_i \mathbf{x} + b_i + \tau z_i \in (-\infty, 0) \cup (\sqrt{2\tau\lambda}, \infty) \right\}, \\ \mathcal{E}^o &:= \left\{ i \in \mathbb{N}_m : A_i \mathbf{x} + b_i = 0, \tau z_i \in \{0, \sqrt{2\tau\lambda}\} \right\}, \end{aligned}$$

for a given  $\tau > 0$ . Obviously,  $\mathcal{E}^o \subseteq \mathcal{E}$ . It is worth mentioning that all sets depend on  $\mathbf{w}$ . For simplicity, we drop their dependence whenever there is no confusion to be caused. Same rules are also applied into  $\mathcal{S}_*, \mathcal{E}_*, \mathcal{O}_*$  and  $\mathcal{E}_*^o$  for  $\mathbf{w}^* := (\mathbf{x}^*; \mathbf{z}^*)$ . A key step towards the Newton method is the construction of the following system of equations. For a given subset  $\Gamma \subseteq \mathbb{N}_m$  and a scalar  $\mu \geq 0$ , it follows the definitions in (1.3) and (1.4) that

$$(4.2) \quad F(\mathbf{w}; \Gamma) = \begin{bmatrix} \nabla f(\mathbf{x}) + A_\Gamma^\top \mathbf{z}_\Gamma \\ A_\Gamma \mathbf{x} + \mathbf{b}_\Gamma \\ \mathbf{z}_\Gamma \end{bmatrix} = 0, \quad \nabla F_\mu(\mathbf{w}; \Gamma) = \begin{bmatrix} \nabla^2 f(\mathbf{x}) & A_\Gamma^\top & 0 \\ A_\Gamma & -\mu I & 0 \\ 0 & 0 & I \end{bmatrix}.$$

We note that the matrix  $\nabla F_\mu$  is a slight perturbation of the Jacobian matrix of  $F$  and  $\nabla F(\mathbf{w}; \Gamma) = \nabla F_0(\mathbf{w}; \Gamma)$ . The following result relates a P-stationary point to a system of equations.

**THEOREM 4.3.** *A point  $\mathbf{w}^* = (\mathbf{x}^*; \mathbf{z}^*)$  is a P-stationary point with  $\tau > 0$  of the problem (1.1) if and only if  $F(\mathbf{w}^*; \Gamma_*) = 0$  and  $\Gamma_* = (\mathcal{S}_* \cup \mathcal{E}_*^o)$ , where  $\Gamma_*$  is defined by (3.1). The Jacobian  $\nabla F(\mathbf{w}^*; \Gamma_*)$  is non-singular if Assumption 4.1 holds.*

*Proof.* The second claim is obvious due to Assumption 4.1. We only need to prove the first claim. We start with the sufficiency. The definitions in (4.1) show the following relationships,

$$(4.3) \quad \Gamma_* = \mathcal{S}_* \cup \mathcal{E}_*^o, \quad \bar{\Gamma}_* = \mathcal{O}_* \cup (\mathcal{E}_* \setminus \mathcal{E}_*^o).$$

We recall  $\mathbf{y}^* = A\mathbf{x}^* + \mathbf{b}$  and  $\mathbb{P} := \text{Prox}_{\tau\lambda\|\cdot\|_+}(\mathbf{y}^* + \tau\mathbf{z}^*)$ . It follows that

$$\mathbb{P}_i = \begin{cases} 0, & i \in \mathcal{S}, \\ 0 \text{ or } y_i^* + \tau z_i^*, & i \in \mathcal{E}_*, \\ y_i^* + \tau z_i^*, & i \in \mathcal{O}_*, \end{cases} = \begin{cases} 0, & i \in \mathcal{S}_* \subseteq \Gamma_*, \\ 0 \text{ or } \tau z_i^*, & i \in \mathcal{E}_*^o \subseteq \Gamma_*, \\ 0 \text{ or } y_i^*, & i \in (\mathcal{E}_* \setminus \mathcal{E}_*^o) \subseteq \bar{\Gamma}_*, \\ y_i^*, & i \in \mathcal{O}_* \subseteq \bar{\Gamma}_*, \end{cases}$$

where the first equation is by (2.6) and second one is by (4.2) and (4.3), which indicates  $y_i^* \in \mathbb{P}_i$  due to  $\mathbf{y}_{\Gamma_*}^* = 0$  from (4.2). Moreover, the first and third equations in (4.2) suffice to  $\nabla f(\mathbf{x}^*) + A^\top \mathbf{z}^* = 0$ , showing (3.12). Namely,  $\mathbf{w}^*$  is a P-stationary point.

Necessity. Let  $\mathbf{w}^*$  be a P-stationary point satisfying (3.12) and  $T_* := \mathcal{S}_* \cup \mathcal{E}_*^o$ . Then  $\bar{T}_* = \mathcal{O}_* \cup (\mathcal{E}_* \setminus \mathcal{E}_*^o)$ . It follows from  $y_i^* \in \mathbb{P}_i$  and (2.6) that

$$y_i^* \in \begin{cases} 0, & i \in \mathcal{S}_*, \\ 0 \text{ or } y_i^* + \tau z_i^*, & i \in \mathcal{E}_*, \\ y_i^* + \tau z_i^*, & i \in \mathcal{O}_*, \end{cases} = \begin{cases} 0, & i \in \mathcal{S}_*, \\ 0 \text{ or } \tau z_i^*, & i \in \mathcal{E}_*^o, \\ 0 \text{ or } y_i^* + \tau z_i^*, & i \in \mathcal{E}_* \setminus \mathcal{E}_*^o, \\ y_i^* + \tau z_i^*, & i \in \mathcal{O}_*, \end{cases}$$

where the equality is by the definition of  $\mathcal{E}_*^o$ , which together with  $y_i^* = 0, i \in \mathcal{E}_*^o$  and  $y_i^* \neq 0, i \in \mathcal{E}_* \setminus \mathcal{E}_*^o$  suffices to

$$\begin{cases} y_i^* = 0, & i \in \mathcal{S}_* \cup \mathcal{E}_*^o = T_*, \\ y_i^* = y_i^* + \tau z_i^*, & i \in \mathcal{E}_* \setminus \mathcal{E}_*^o, \\ y_i^* = y_i^* + \tau z_i^*, & i \in \mathcal{O}_*, \end{cases} \iff \begin{cases} y_i^* = 0, & i \in T_*, \\ z_i^* = 0, & i \in \bar{T}_*. \end{cases}$$

This gives rise to the last two conditions  $\mathbf{y}_{T_*}^* = 0, \mathbf{z}_{\bar{T}_*}^* = 0$  in (4.2). Furthermore, the first condition in (3.12) and  $\mathbf{z}_{\bar{T}_*}^* = 0$  derive the first condition of (4.2). Overall, we have  $F(\mathbf{w}^*; T_*) = 0$ . Now we



show  $T_* = \Gamma_*$ . By (3.1) that  $\Gamma_* = \{i \in \mathbb{N}_m : y_i^* = 0\}$ , it follows  $T_* \subseteq \Gamma_*$ . Suppose, there is a  $j \in \Gamma_*$  but  $j \notin T_*$ , then we have  $y_j^* = z_j^* = 0$  and thus  $j \in \mathcal{E}_*^o \subseteq T_*$  by (4.1), a contradiction. Therefore,  $T_* = \Gamma_*$ , finishing the proof.  $\square$

*Remark 4.4.* It is interesting to note that [Theorem 4.3](#) suggests a threshold value for  $\lambda$  to exclude the zero solutions when  $b_i \neq 0, \forall i \in \mathbb{N}_m$ . Suppose  $\mathbf{x}^* = 0$  is a P-stationary point. The second equation  $A_{\Gamma_*} \mathbf{x} + \mathbf{b}_{\Gamma_*} = 0$  in  $F(\mathbf{w}^*; \Gamma_*) = 0$  indicates  $\Gamma_* = \emptyset$  and thus  $\mathbf{z}^* = 0$ . These and (4.1) give rise to  $b_i \in (-\infty, 0) \cup [\sqrt{2\tau\lambda}, \infty), \forall i \in \mathbb{N}_m$ . In real applications (e.g., SVM and 1-bit CS), there is at least one  $i \in \mathbb{N}_m$  such that  $b_i > 0$ , which results in  $\lambda \leq \min_i \{b_i^2 : b_i > 0\} / (2\tau)$ . Hence, to exclude the zero solutions for some real applications, we choose

$$(4.4) \quad \lambda > \min_i \{b_i^2 : b_i > 0\} / (2\tau).$$

**4.2. Algorithmic design.** [Theorem 4.3](#) lays the foundation for developing Newton's method, which is to solve the stationarity equation in (4.2). Let  $\mathbf{w}^k := (\mathbf{x}^k; \mathbf{z}^k)$  be the current iterate. We define  $\mathcal{S}_k$  and  $\mathcal{E}_k^o$  by (4.1) with  $\mathbf{w}$  being replaced by  $\mathbf{w}^k$  and let

$$(4.5) \quad T_k := \mathcal{S}_k \cup \mathcal{E}_k^o.$$

Let  $\mathbf{d}^k = (\mathbf{u}^k; \mathbf{v}^k)$  with  $\mathbf{u}^k \in \mathbb{R}^n$  and  $\mathbf{v}^k \in \mathbb{R}^m$ . For such a defined  $T_k$ , a Newton direction  $\mathbf{d}^k$  for the equation (4.2) solves the following linear equations:

$$\nabla F(\mathbf{w}^k; T_k) \mathbf{d} = -F(\mathbf{w}^k; T_k).$$

To improve the nonsingularity of the Jacobian matrix  $\nabla F(\mathbf{w}^k; T_k)$ , we replace it with  $\nabla F_{\mu_k}(\mathbf{w}^k; T_k)$ . That is, at  $\mathbf{w}^k$ , we solve the equation:

$$(4.6) \quad \nabla F_{\mu_k}(\mathbf{w}^k; T_k) \mathbf{d} = -F(\mathbf{w}^k; T_k),$$

where  $\nabla F_{\mu_k}(\mathbf{w}^k; T_k)$  is defined in (4.2). The Newton direction  $\mathbf{d}^k$  satisfies

$$(4.7) \quad \begin{bmatrix} \nabla^2 f(\mathbf{x}^k) & A_{T_k}^\top & 0 \\ A_{T_k} & -\mu_k I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{u}^k \\ \mathbf{v}_{T_k}^k \\ \mathbf{v}_{\bar{T}_k}^k \end{bmatrix} = - \begin{bmatrix} \nabla f(\mathbf{x}^k) + A_{T_k}^\top \mathbf{z}_{T_k}^k \\ A_{T_k} \mathbf{x}^k + \mathbf{b}_{T_k} \\ \mathbf{z}_{\bar{T}_k}^k \end{bmatrix}.$$

Here, the rule to update  $\mu_k$  is as follows:

$$(4.8) \quad \mu_k = \min\{\alpha\mu_{k-1}, \rho\|F(\mathbf{w}^k; T_k)\|\},$$

where  $\alpha \in (0, 1)$  and  $\rho > 0$ . Now we summarize the proposed method in [Algorithm 4.1](#).

---

**Algorithm 4.1** NM01: Newton's method for 0/1-loss optimization

---

- 1: Initialize  $\mathbf{w}^0 = (\mathbf{x}^0; \mathbf{z}^0)$  and  $\mu_{-1} > 0$ . Set the parameter  $\tau, \lambda, \rho > 0, \alpha \in (0, 1)$ .  
Compute  $T_0$  by (4.5) and set  $k := 0$ .
  - 2: **if**  $\|F(\mathbf{w}^k; T_k)\| > 0$  **then**
  - 3: Update  $\mu_k$  by (4.8).
  - 4: Update  $\mathbf{d}^k$  by solving (4.7).
  - 5: Update  $\mathbf{w}^{k+1} = \mathbf{w}^k + \mathbf{d}^k$ .
  - 6: Update  $T_{k+1}$  by (4.5) and set  $k := k + 1$ .
  - 7: **end if**
  - 8: **return**  $\mathbf{w}^k$ .
- 

*Remark 4.5.* In general, the computational complexity for solving the equation (4.7) is approximately  $O(n^2 \max\{n, |T_k|\})$ . This is fine for small-sized problems. When  $n$  is large, the computational cost is too high and existing first-order algorithms would be faster. Fortunately, for many real applications, such as SVM and 1-bit CS, their functions are separable and have block structures such as in (1.2). This implies that the Hessian matrix  $\nabla^2 f(\mathbf{x}^k)$  is of diagonal blocks and is invertible. The worst-case computational complexity can be reduced to  $O(|T_k|^2 \max\{n, |T_k|\})$ . For SVM or 1-bit CS problems,  $T_k$  coincides with the indices of incorrectly classified samples that take a relatively small portion of the total samples. Hence,  $|T_k|$  can be on a small scale and computation of the Newton direction can be very cheap.

**4.3. Quadratic convergence.** Let us first explain why it is a challenging task to establish the quadratic convergence of the proposed Newton method. Suppose  $\mathbf{w}^*$  satisfies the stationarity equation  $F(\mathbf{w}^*; \Gamma_*) = 0$  (see [Theorem 4.3](#)). If we know  $\Gamma_*$  beforehand, then by fixing  $T_k = \Gamma_*$ , our proposed method reduces to the standard Newton's method that solves equations with smooth functions. The quadratic convergence follows under [Assumption 4.1](#) and [Assumption 4.2](#). However, the difficulty we are facing is that the set  $T_k$  may change from iteration to iteration. A different  $T_k$  leads to a different system of equations  $F(\mathbf{w}; T_k) = 0$ . Hence, in each step, the algorithm finds a Newton direction for a different system of equations instead of a fixed system. This is where the standard proof for quadratic convergence breaks down. As we will see below, it takes a great deal of effort in establishing quadratic convergence.

The first technical result is about extending the stationarity equation to some indices that are given in a neighborhood of  $\mathbf{w}^*$ . In the proof, we recall  $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$  and  $\mathbf{y}^* = \mathbf{A}\mathbf{x}^* + \mathbf{b}$ .

LEMMA 4.6. *Let  $\mathbf{w}^*$  be a P-stationary point with  $0 < \tau < \tau_* := \min\{\tau_1, \tau_2\}$  of the problem (1.1),  $\tau_1, \tau_2$  and  $\Gamma_*$  be given by (3.13) and (3.1). Then there is a  $\delta_1^* > 0$  such that, for any  $\mathbf{w} \in N(\mathbf{w}^*, \delta_1^*)$  with its associated indices  $\mathcal{S}$  and  $\mathcal{E}^o$ , it holds*

$$(4.9) \quad F(\mathbf{w}^*; T) = 0 \quad \text{and} \quad T := (\mathcal{S} \cup \mathcal{E}^o) \subseteq \Gamma_*.$$

*Proof.* i) [Theorem 4.3](#) states that the P-stationary point  $\mathbf{w}^*$  of (1.1) satisfies

$$(4.10) \quad \nabla f(\mathbf{x}^*) + A_{\Gamma_*} \mathbf{z}_{\Gamma_*}^* = 0, \quad \mathbf{y}_{\Gamma_*}^* = 0, \quad \mathbf{z}_{\Gamma_*}^* = 0$$

for  $0 < \tau < \tau_*$ , where  $\Gamma_* = \mathcal{S}_* \cup \mathcal{E}_*^o$ . Note that  $\mathcal{E}_* \setminus \mathcal{E}_*^o \subseteq \bar{\Gamma}_*$  which by (4.10) leads to

$$(4.11) \quad \mathbf{z}_{\mathcal{E}_* \setminus \mathcal{E}_*^o}^* = 0.$$

Using the same reasoning for proving (3.15), we can prove for  $0 < \tau < \tau_* = \min\{\tau_1, \tau_2\}$  in (3.13) that

$$(4.12) \quad y_i^* > \sqrt{2\tau\lambda} \quad \text{if} \quad y_i^* > 0, \quad \tau z_i^* < \sqrt{2\tau\lambda} \quad \text{if} \quad z_i^* > 0.$$

Therefore, we have the following facts

$$\begin{aligned} \mathcal{E}_*^o &= \{i \in \mathbb{N}_m : y_i^* = 0, \tau z_i^* \in \{0, \sqrt{2\tau\lambda}\}\} && \text{(by (4.1))} \\ &= \{i \in \mathbb{N}_m : y_i^* = 0, z_i^* = 0\}, && \text{(by (4.12))} \\ \mathcal{E}_* \setminus \mathcal{E}_*^o &= \{i \in \mathbb{N}_m : y_i^* \neq 0, y_i^* + \tau z_i^* = \sqrt{2\tau\lambda}\} && \text{(by (4.1))} \\ &= \{i \in \mathbb{N}_m : y_i^* = \sqrt{2\tau\lambda}\} && \text{(by (4.11))} \\ &= \emptyset, && \text{(by (4.12))} \end{aligned}$$

These facts lead to

$$(4.13) \quad \mathcal{E}_* = \mathcal{E}_*^o \cup (\mathcal{E}_* \setminus \mathcal{E}_*^o) = \mathcal{E}_*^o = \left\{i \in \mathbb{N}_m : y_i^* = z_i^* = 0\right\},$$

which yields the following relations

$$(4.14) \quad \Gamma_* = \mathcal{S}_* \cup \mathcal{E}_*^o = \mathcal{S}_* \cup \mathcal{E}_*, \quad \bar{\Gamma}_* = \mathcal{O}_*.$$

For a sufficiently small  $\delta_1^*$ , any  $\mathbf{w} \in N(\mathbf{w}^*, \delta_1^*)$  satisfies,

$$(4.15) \quad |y_i + \tau z_i - y_i^* - \tau z_i^*| \leq c\delta_1^*, \quad \forall i \in \mathbb{N}_m,$$

where  $c > 0$  is a constant relied on  $A$  and  $\tau$ . The definitions of  $\mathcal{S}$  and  $\mathcal{S}_*$  in (4.1) mean that, for any  $i \in \mathcal{S}$  or  $i \in \mathcal{S}_*$ ,

$$(4.16) \quad \begin{aligned} y_i + \tau z_i \in (0, \sqrt{2\tau\lambda}) &\iff |y_i + \tau z_i - \sqrt{\tau\lambda/2}| < \sqrt{\tau\lambda/2}, \\ y_i^* + \tau z_i^* \in (0, \sqrt{2\tau\lambda}) &\iff |y_i^* + \tau z_i^* - \sqrt{\tau\lambda/2}| < \sqrt{\tau\lambda/2}. \end{aligned}$$

Using this fact, if  $\mathcal{S}_* \not\subseteq \mathcal{S}$ , then there is an  $i \in \mathcal{S}_*$  but  $i \notin \mathcal{S}$  such that

$$\begin{aligned} |y_i + \tau z_i - y_i^* - \tau z_i^*| &= |y_i + \tau z_i - \sqrt{\tau\lambda/2} - y_i^* - \tau z_i^* + \sqrt{\tau\lambda/2}| \\ &\geq \sqrt{\tau\lambda/2} - |y_i^* + \tau z_i^* - \sqrt{\tau\lambda/2}| \quad (\text{by } i \notin \mathcal{S} \text{ and (4.16)}) \\ &\geq \sqrt{\tau\lambda/2} - \max_{i \in \mathcal{S}_*} |y_i^* + \tau z_i^* - \sqrt{\tau\lambda/2}| =: \delta_s \\ &> 0, \quad (\text{by (4.16)}) \end{aligned}$$

Since  $c\delta_1^*$  can be smaller than  $\delta_s$ , the above fact contradicts with (4.15). Hence, it holds  $\mathcal{S}_* \subseteq \mathcal{S}$ . Similar reasoning also derives  $\mathcal{O}_* \subseteq \mathcal{O}$ . These allow us to obtain

$$\mathcal{E} = \mathbb{N}_m \setminus (\mathcal{S} \cup \mathcal{O}) \subseteq \mathbb{N}_m \setminus (\mathcal{S}_* \cup \mathcal{O}_*) = \mathcal{E}_*.$$

Overall, for any  $\mathbf{w} \in N(\mathbf{w}^*, \delta_1^*)$ , it holds

$$(4.17) \quad \mathcal{S}_* \subseteq \mathcal{S}, \quad \mathcal{O}_* \subseteq \mathcal{O}, \quad \mathcal{E} \subseteq \mathcal{E}_*.$$

The above relations enable us to claim that

$$(4.18) \quad (\mathcal{S} \setminus \mathcal{S}_*) \subseteq \mathcal{E}_*, \quad (\mathcal{O} \setminus \mathcal{O}_*) \subseteq \mathcal{E}_*.$$

Now, we can show

$$\begin{aligned} \mathbf{y}_{\mathcal{S}_*}^* &= 0, \quad (\text{by } \mathcal{S}_* \subseteq \Gamma_* \text{ and (4.10)}) \\ \mathbf{y}_{\mathcal{S} \setminus \mathcal{S}_*}^* &= 0, \quad (\text{by } \mathcal{S} \setminus \mathcal{S}_* \subseteq \mathcal{E}_* \text{ from (4.18) and (4.13)}) \\ \mathbf{y}_{\mathcal{E}^o}^* &= 0, \quad (\text{by } \mathcal{E}^o \subseteq \mathcal{E} \subseteq \mathcal{E}_* \text{ from (4.17) and (4.13)}) \\ \mathbf{z}_{\mathcal{O}_*}^* &= 0, \quad (\text{by } \mathcal{O}_* \subseteq \bar{\Gamma}_* \text{ and (4.10)}) \\ \mathbf{z}_{\mathcal{O} \setminus \mathcal{O}_*}^* &= 0, \quad (\text{by } \mathcal{O} \setminus \mathcal{O}_* \subseteq \mathcal{E}_* \text{ from (4.18) and (4.13)}) \\ \mathbf{z}_{\mathcal{E} \setminus \mathcal{E}^o}^* &= 0. \quad (\text{by } \mathcal{E} \setminus \mathcal{E}^o \subseteq \mathcal{E} \subseteq \mathcal{E}_* \text{ from (4.17) and (4.13)}) \end{aligned}$$

These conditions combining with

$$\begin{aligned} T &= \mathcal{S} \cup \mathcal{E}^o = \mathcal{S}_* \cup (\mathcal{S} \setminus \mathcal{S}_*) \cup \mathcal{E}^o, \\ \bar{T} &= \mathcal{O} \cup (\mathcal{E} \setminus \mathcal{E}^o) = \mathcal{O}_* \cup (\mathcal{O} \setminus \mathcal{O}_*) \cup (\mathcal{E} \setminus \mathcal{E}^o), \end{aligned}$$

imply  $\mathbf{y}_T^* = 0$  and  $\mathbf{z}_{\bar{T}}^* = 0$ . As a consequence of this and  $\mathbf{z}_{\Gamma_*}^* = 0$  from (4.10),

$$0 = f(\mathbf{x}^*) + A_{\Gamma_*}^\top \mathbf{z}_{\Gamma_*}^* = f(\mathbf{x}^*) + A^\top \mathbf{z}^* = f(\mathbf{x}^*) + A_T^\top \mathbf{z}_T^*.$$

Overall, we verify  $F(\mathbf{w}^*; T) = 0$ , as desired. Finally, we observe that

$$\begin{aligned} T &= \mathcal{S} \cup \mathcal{E}^o = \mathcal{S}_* \cup (\mathcal{S} \setminus \mathcal{S}_*) \cup \mathcal{E}^o \\ &\subseteq \mathcal{S}_* \cup \mathcal{E}_* \cup \mathcal{E}^o \quad (\text{by } \mathcal{S} \setminus \mathcal{S}_* \subseteq \mathcal{E}_* \text{ from (4.18)}) \\ &\subseteq \mathcal{S}_* \cup \mathcal{E}_* \cup \mathcal{E}_* \quad (\text{by } \mathcal{E}^o \subseteq \mathcal{E} \subseteq \mathcal{E}_* \text{ from (4.17)}) \\ &= \Gamma_*. \quad (\text{by (4.14)}) \end{aligned}$$

The whole proof is completed.  $\square$

The second technical result is about the uniform nonsingularity of the perturbed Jacobian matrix  $\nabla F_\mu(\mathbf{w}; T)$  over a neighborhood of  $\mathbf{w}^*$ .

LEMMA 4.7. *Let  $\mathbf{w}^*$  be a  $P$ -stationary point with  $0 < \tau < \tau_*$  of the problem (1.1) and  $\tau_*$  be given by (3.13). Assume Assumption 4.1 and Assumption 4.2. It holds*

$$(4.19) \quad C_* \geq \|\nabla F_\mu(\mathbf{w}; T)\| \geq \sigma_{\min}(\nabla F_\mu(\mathbf{w}; T)) \geq c_* > 0,$$

for any  $\mathbf{w} \in N(\mathbf{w}^*, \delta_2^*)$  and any  $0 \leq \mu \leq c_*/2$ , where  $T = \mathcal{S} \cup \mathcal{E}^o$  and

$$(4.20) \quad C_* := 2 \max\{1, \|H(\Gamma_*)\|\}, \quad c_* := 0.5 \min\{1, \min_{\Gamma \subseteq \Gamma_*} \sigma_{\min}(H(\Gamma))\},$$

$$(4.21) \quad \delta_2^* := \min\left\{\delta_1^*, \frac{c_*}{2L_*}\right\}, \quad H(\Gamma) := \begin{bmatrix} \nabla^2 f(\mathbf{x}^*) & A_\Gamma^\top \\ A_\Gamma & 0 \end{bmatrix}.$$

*Proof.* Since  $\nabla^2 f(\mathbf{x}^*)$  is positive definite and  $A_{\Gamma_*}$  is full row rank by [Assumption 4.1](#),  $A_\Gamma$  is full row rank for any  $\Gamma \subseteq \Gamma_*$  and thus  $H(\Gamma)$  is non-singular. We have  $\sigma_{\min}(H(\Gamma)) > 0$  for any  $\Gamma \subseteq \Gamma_*$  and  $c_* > 0$ . Now we build the bounds of  $\nabla F_\mu(\mathbf{w}; T)$ . For any given two matrices  $D'$  and  $D$ , we have the first fact

$$(4.22) \quad \begin{aligned} \|D' - D\| &\geq \max_i |\sigma_i(D') - \sigma_i(D)| \geq |\sigma_{i_0}(D') - \sigma_{i_0}(D)| \\ &\geq \sigma_{i_0}(D') - \sigma_{\min}(D) \geq \sigma_{\min}(D') - \sigma_{\min}(D), \end{aligned}$$

where the first inequality is from [\[30, Reminder \(2\), on Page 76\]](#) and  $i_0$  satisfies that  $\sigma_{i_0}(D) = \sigma_{\min}(D)$ . Recall that

$$H(\Gamma) \stackrel{(4.21)}{=} \begin{bmatrix} \nabla^2 f(\mathbf{x}^*) & A_\Gamma^\top \\ A_\Gamma & 0 \end{bmatrix}, \quad \nabla F(\mathbf{w}^*; T) = \begin{bmatrix} H(T) & 0 \\ 0 & I \end{bmatrix}.$$

For any  $\mathbf{w} \in N(\mathbf{w}^*, \delta_2^*)$  and  $\delta_2^* \leq \delta_1^*$ , [Lemma 4.6](#) contributes to  $T \subseteq \Gamma_*$ . So  $H(T)$  is a submatrix of  $H(\Gamma_*)$ . Hence,

$$\|H(T)\| \leq \|H(\Gamma_*)\|, \quad \sigma_{\min}(H(T)) \geq \min_{\Gamma \subseteq \Gamma_*} \sigma_{\min}(H(\Gamma)),$$

where the latter is by  $T \subseteq \Gamma_*$ , which gives us the second fact

$$(4.23) \quad \begin{aligned} \sigma_{\min}(\nabla F(\mathbf{w}^*; T)) &= \min\{1, \sigma_{\min}(H(T))\} \\ &\geq \min\{1, \min_{\Gamma \subseteq \Gamma_*} \sigma_{\min}(H(\Gamma))\} = 2c_*, \end{aligned}$$

$$(4.24) \quad \|\nabla F(\mathbf{w}^*; T)\| = \max\{1, \|H(T)\|\} \leq \max\{1, \|H(\Gamma_*)\|\} = C_*/2.$$

The locally Lipschitz continuity of  $\nabla^2 f$  around  $\mathbf{x}^*$  with  $L_*$  yields the third fact,

$$(4.25) \quad \begin{aligned} \|\nabla F(\mathbf{w}^*; T) - \nabla F(\mathbf{w}; T)\| &= \|\nabla^2 f(\mathbf{x}) - \nabla^2 f(\mathbf{x}^*)\| \leq L_* \|\mathbf{x} - \mathbf{x}^*\| \\ &\leq L_* \|\mathbf{w} - \mathbf{w}^*\| \leq L_* \delta_2^* \leq c_*/2. \quad (\text{by (4.21)}) \end{aligned}$$

Now these three facts allow us to derive

$$(4.26) \quad \begin{aligned} &\sigma_{\min}(\nabla F_\mu(\mathbf{w}; T)) \\ &\geq \sigma_{\min}(\nabla F(\mathbf{w}; T)) - \|\nabla F(\mathbf{w}; T) - \nabla F_\mu(\mathbf{w}; T)\| \quad (\text{by (4.22)}) \\ &= \sigma_{\min}(\nabla F(\mathbf{w}; T)) - \mu \quad (\text{by (4.2)}) \\ &\geq \sigma_{\min}(\nabla F(\mathbf{w}^*; T)) - \|\nabla F(\mathbf{w}^*; T) - \nabla F(\mathbf{w}; T)\| - \mu \quad (\text{by (4.22)}) \\ &\geq \sigma_{\min}(\nabla F(\mathbf{w}^*; T)) - c_*/2 - \mu \quad (\text{by (4.25)}) \\ &\geq \sigma_{\min}(\nabla F(\mathbf{w}^*; T)) - c_* \quad (\text{by } \mu \leq c_*/2) \\ &\geq c_*. \quad (\text{by (4.23)}) \end{aligned}$$

Similarly, we also have

$$\begin{aligned}
& \|\nabla F_\mu(\mathbf{w}; T)\| \\
& \leq \|\nabla F(\mathbf{w}; T)\| + \|\nabla F(\mathbf{w}; T) - \nabla F_\mu(\mathbf{w}; T)\| \\
& = \|\nabla F(\mathbf{w}; T)\| + \mu && \text{(by (4.2))} \\
(4.27) \quad & \leq \|\nabla F(\mathbf{w}^*; T)\| + \|\nabla F(\mathbf{w}^*; T) - \nabla F(\mathbf{w}; T)\| + \mu && \text{(by (4.22))} \\
& \leq \|\nabla F(\mathbf{w}^*; T)\| + c_*/2 + \mu && \text{(by (4.25))} \\
& \leq \|\nabla F(\mathbf{w}^*; T)\| + c_* && \text{(by } \mu \leq c_*/2\text{)} \\
& \leq C_*/2 + c_* && \text{(by (4.24))} \\
& \leq C_*. && \text{(by } c_* \leq C_*/2\text{)}
\end{aligned}$$

The whole proof is completed.  $\square$

Now we are ready to claim the following local quadratic convergence.

**THEOREM 4.8.** *Let  $\mathbf{w}^*$  be any  $P$ -stationary point with  $0 < \tau < \tau_*$  of (1.1),  $\tau_*$  and  $\delta_2^*, c_*, C_*$  be given by (3.13) and (4.20). Assume [Assumption 4.1](#) and [Assumption 4.2](#). Let  $\{\mathbf{w}^k\}$  be the sequence generated by [Algorithm 4.1](#) and  $0 \leq \mu_{-1} \leq c_*/2$ . If the initial point satisfies  $\mathbf{w}^0 \in N(\mathbf{w}^*, \delta_*)$ , where*

$$(4.28) \quad \delta_* := \min \{ \delta_2^*, c_*/(2(L_* + 2\rho C_*)) \},$$

then the following results hold.

- a) The sequence  $\{\mathbf{d}^k\}_{k \geq 0}$  is well defined and  $\lim_{k \rightarrow \infty} \mathbf{d}^k = 0$ .
- b) The whole sequence  $\{\mathbf{w}^k\}$  converges to  $\mathbf{w}^*$  quadratically, namely,

$$\|\mathbf{w}^{k+1} - \mathbf{w}^*\| \leq ((L_* + 2\rho C_*)/c_*) \|\mathbf{w}^k - \mathbf{w}^*\|^2.$$

- c) The halting condition satisfies

$$\|F(\mathbf{w}^{k+1}; T_{k+1})\| \leq (L_* + 2\rho C_*)(C_*/c_*^3) \|F(\mathbf{w}^k; T_k)\|^2$$

and [Algorithm 4.1](#) reaches  $\|F(\mathbf{w}^k; T_k)\| < \epsilon$  for a given tolerance  $\epsilon > 0$  when

$$(4.29) \quad k \geq \left\lceil \log_2 \left( 2\sqrt{(L_* + 2\rho C_*)(C_*/c_*^3)} \|\mathbf{w}^0 - \mathbf{w}^*\| \right) - \log_2(\sqrt{\epsilon}) \right\rceil.$$

*Proof.* a) It is easily observed

$$(4.30) \quad 0 < \mu_k \leq \mu_{k-1}, \quad k = 1, 2, 3, \dots, \quad \text{and} \quad \lim_{k \rightarrow \infty} \mu_k = 0.$$

It follows from [Lemma 4.6](#) and the facts  $\delta_* \leq \delta_2^* \leq \delta_1^*$  and  $\mathbf{w}^0 \in N(\mathbf{w}^*, \delta_*)$  that

$$(4.31) \quad F(\mathbf{w}^*; T_0) = 0$$

with  $T_0 = \mathcal{S}_0 \cup \mathcal{E}_0^o$  and from [Lemma 4.7](#) that

$$(4.32) \quad C_* \geq \|\nabla F_{\mu_0}(\mathbf{w}^0; T_0)\| \geq \sigma_{\min}(\nabla F_{\mu_0}(\mathbf{w}^0; T_0)) \geq c_*$$

Here, we used the fact that  $\mu_0 \leq \mu_{-1} \leq c_*/2$  by (4.30). From (4.6), we have

$$(4.33) \quad \nabla F_{\mu_0}(\mathbf{w}^0; T_0) \mathbf{d}^0 = -F(\mathbf{w}^0; T_0).$$

[Lemma 4.7](#) states that  $\nabla F_{\mu_0}(\mathbf{w}^0; T_0)$  is non-singular and thus  $\mathbf{d}^0$  is well defined. Let

$$(4.34) \quad \mathbf{w}_\beta^0 = \mathbf{w}^* + \beta(\mathbf{w}^0 - \mathbf{w}^*) = (\mathbf{x}_\beta^0; \mathbf{z}_\beta^0)$$

where  $\beta \in [0, 1]$ . One can easily check that  $\mathbf{w}_\beta^0 \in N(\mathbf{w}^*, \delta_*)$  as

$$\|\mathbf{w}_\beta^0 - \mathbf{w}^*\| = \beta \|\mathbf{w}^0 - \mathbf{w}^*\| \leq \delta_*.$$

The definition in (4.2) enables us to obtain

$$\begin{aligned}
& \|\nabla F_{\mu_0}(\mathbf{w}^0; T_0) - \nabla F(\mathbf{w}_\beta^0; T_0)\| \\
& \leq \|\nabla^2 f(\mathbf{x}^0) - \nabla^2 f(\mathbf{x}_\beta^0)\| + \mu_0 && \text{(by (4.2))} \\
& \leq L_* \|\mathbf{x}^0 - \mathbf{x}_\beta^0\| + \rho \|F(\mathbf{w}^0; T_0)\| && \text{(by (4.8))} \\
& \leq L_* \|\mathbf{w}^0 - \mathbf{w}_\beta^0\| + \rho \|\nabla F_{\mu_0}(\mathbf{w}^0; T_0)\| \|\mathbf{d}^0\| && \text{(by (4.6))} \\
& \leq L_* \|\mathbf{w}^0 - \mathbf{w}_\beta^0\| + \rho C_* \|\mathbf{w}^1 - \mathbf{w}^0\| && \text{(by (4.32))} \\
& \leq L_*(1 - \beta) \|\mathbf{w}^0 - \mathbf{w}^*\| + \rho C_*(\|\mathbf{w}^1 - \mathbf{w}^*\| + \|\mathbf{w}^0 - \mathbf{w}^*\|) && \text{(by (4.34))} \\
& = (L_*(1 - \beta) + \rho C_*) \|\mathbf{w}^0 - \mathbf{w}^*\| + \rho C_* \|\mathbf{w}^1 - \mathbf{w}^*\|.
\end{aligned}$$

Denote  $\Theta_{\mu_0} := \nabla F_{\mu_0}(\mathbf{w}^0; T_0)$  and  $\Theta(\beta) := \nabla F(\mathbf{w}_\beta^0; T_0)$ . The above condition yields

$$(4.35) \quad \int_0^1 \|\Theta_{\mu_0} - \Theta(\beta)\| d\beta \leq (L_*/2 + \rho C_*) \|\mathbf{w}^0 - \mathbf{w}^*\| + \rho C_* \|\mathbf{w}^1 - \mathbf{w}^*\|.$$

For the fixed  $T_0$ , the function  $F(\cdot, T_0)$  is differentiable, which by (4.31) derives

$$(4.36) \quad F(\mathbf{w}^0; T_0) = F(\mathbf{w}^*; T_0) + \int_0^1 \Theta(\beta)(\mathbf{w}^0 - \mathbf{w}^*) d\beta = \int_0^1 \Theta(\beta)(\mathbf{w}^0 - \mathbf{w}^*) d\beta.$$

Now the following chain of inequalities holds.

$$\begin{aligned}
c_* \|\mathbf{w}^1 - \mathbf{w}^*\| &= c_* \|\mathbf{w}^0 + \mathbf{d}^0 - \mathbf{w}^*\| \\
&= c_* \|\mathbf{w}^0 - \mathbf{w}^* - \Theta_{\mu_0}^{-1} F(\mathbf{w}^0; T_0)\| && \text{(by (4.33))} \\
&\leq \|\Theta_{\mu_0}(\mathbf{w}^0 - \mathbf{w}^*) - F(\mathbf{w}^0; T_0)\| && \text{(by (4.32))} \\
&= \|\Theta_{\mu_0}(\mathbf{w}^0 - \mathbf{w}^*) - \int_0^1 \Theta(\beta)(\mathbf{w}^0 - \mathbf{w}^*) d\beta\| && \text{(by (4.36))} \\
&\leq \int_0^1 \|\Theta_{\mu_0} - \Theta(\beta)\| \|\mathbf{w}^0 - \mathbf{w}^*\| d\beta \\
&= (\theta_*/2) \|\mathbf{w}^0 - \mathbf{w}^*\|^2 + \rho C_* \|\mathbf{w}^1 - \mathbf{w}^*\| \|\mathbf{w}^0 - \mathbf{w}^*\| && \text{(by (4.35))} \\
&\leq (\theta_*/2) \|\mathbf{w}^0 - \mathbf{w}^*\|^2 + \rho C_* \delta_* \|\mathbf{w}^1 - \mathbf{w}^*\| \\
&\leq (\theta_*/2) \|\mathbf{w}^0 - \mathbf{w}^*\|^2 + (c_*/2) \|\mathbf{w}^1 - \mathbf{w}^*\|,
\end{aligned}$$

where  $\theta_* := L_* + 2\rho C_*$  and the last inequality is from  $\delta_* \leq c_*/(2\theta_*)$  by (4.28) and

$$(4.37) \quad \|\mathbf{w}^0 - \mathbf{w}^*\| < \delta_* \leq c_*/(2\theta_*) < c_*/(2\rho C_*).$$

The above chain of inequalities suffices to the following fact

$$(4.38) \quad \|\mathbf{w}^1 - \mathbf{w}^*\| \leq (\theta_*/c_*) \|\mathbf{w}^0 - \mathbf{w}^*\|^2.$$

This together with  $\|\mathbf{w}^0 - \mathbf{w}^*\| < \delta_*$  and (4.37) derives

$$\|\mathbf{w}^1 - \mathbf{w}^*\| \leq (\theta_*/c_*) \delta_* \|\mathbf{w}^0 - \mathbf{w}^*\| \leq (1/2) \|\mathbf{w}^0 - \mathbf{w}^*\| < \delta_*,$$

which means  $\mathbf{w}^1 \in N(\mathbf{w}^*, \delta_*)$ . In addition,  $\mu_1 \leq \mu_0 \leq c_*/2$  by (4.30). Hence, replacing  $T_0$  by  $T_1$ , the same reasoning allows us to show that  $\mathbf{d}^1$  is well defined and

$$\|\mathbf{w}^2 - \mathbf{w}^*\| \leq (\theta_*/c_*) \|\mathbf{w}^1 - \mathbf{w}^*\|^2.$$

By the induction, we can conclude that  $\mathbf{w}^k \in N(\mathbf{w}^*, \delta_*)$ ,  $\mathbf{d}^k$  is well defined and

$$(4.39) \quad \|\mathbf{w}^{k+1} - \mathbf{w}^*\| \leq (\theta_*/c_*) \|\mathbf{w}^k - \mathbf{w}^*\|^2,$$

$$(4.40) \quad \leq (\theta_*/c_*) \|\mathbf{w}^k - \mathbf{w}^*\| \delta_* \leq (1/2) \|\mathbf{w}^k - \mathbf{w}^*\|. \quad \text{(by (4.37))}$$

Therefore, (4.39) claims b). The conclusion of a) can be made by (4.40) that

$$\mathbf{w}^k \rightarrow \mathbf{w}^*, \quad \mathbf{d}^k = \mathbf{w}^{k+1} - \mathbf{w}^k = \mathbf{w}^{k+1} - \mathbf{w}^* + \mathbf{w}^* - \mathbf{w}^k \rightarrow 0.$$

c) The above proof shows  $\mathbf{w}^k \in N(\mathbf{w}^*, \delta_*)$  and hence (4.9) results in

$$(4.41) \quad F(\mathbf{w}^*; T_k) = 0,$$

where  $T_k = \mathcal{S}_k \cup \mathcal{E}_k^c$ . By letting  $\mathbf{w}_\beta^k = \mathbf{w}^* + \beta(\mathbf{w}^k - \mathbf{w}^*)$ , where  $\beta \in [0, 1]$ , we have  $\mathbf{w}_\beta^k \in N(\mathbf{w}^*, \delta_*)$ . To show (4.19) in Lemma 4.7, we verified the lower and upper bounds by (4.26) and (4.27). Similarly, we can prove these bounds hold for  $\nabla F(\mathbf{w}_\beta^k; T_k)$ . (In fact, since  $\mathbf{w}_\beta^k \in N(\mathbf{w}^*, \delta_*) \subseteq N(\mathbf{w}^*, \delta_2^*)$  by  $\delta_* \leq \delta_2^*$ , one just needs to set  $\mu = 0$  and  $\mathbf{w} = \mathbf{w}_\beta^k$  in (4.26) and (4.27). Therefore,

$$(4.42) \quad C_* \geq \|\nabla F(\mathbf{w}_\beta^k; T_k)\| \geq \sigma_{\min}(\nabla F(\mathbf{w}_\beta^k; T_k)) \geq c_*,$$

Again, the function  $F(\cdot, T_k)$  is differentiable for the fixed  $T_k$ , so the Mean-value theorem states that there is a  $\beta_0 \in (0, 1)$  satisfying

$$(4.43) \quad \begin{aligned} \|F(\mathbf{w}^k, T_k)\| &= \|F(\mathbf{w}^*; T_k) + \nabla F(\mathbf{w}_{\beta_0}^k; T_k)(\mathbf{w}^k - \mathbf{w}^*)\| \\ &= \|\nabla F(\mathbf{w}_{\beta_0}^k; T_k)(\mathbf{w}^k - \mathbf{w}^*)\| && \text{(by (4.41))} \\ &\in [c_* \|\mathbf{w}^k - \mathbf{w}^*\|, C_* \|\mathbf{w}^k - \mathbf{w}^*\|], && \text{(by (4.42))} \end{aligned}$$

This contributes to

$$\begin{aligned} \|F(\mathbf{w}^k; T_k)\| &\leq C_* \|\mathbf{w}^k - \mathbf{w}^*\| \leq (\theta_* C_* / c_*) \|\mathbf{w}^{k-1} - \mathbf{w}^*\|^2 && \text{(by (4.39))} \\ &\leq (\theta_* C_* / c_*^3) \|F(\mathbf{w}^{k-1}; T_{k-1})\|^2 && \text{(by (4.43))} \\ &\leq (\theta_* C_*^3 / c_*^3) \|\mathbf{w}^{k-1} - \mathbf{w}^*\|^2 && \text{(by (4.43))} \\ &\leq (\theta_* C_*^3 / c_*^3) 2^{-2} \|\mathbf{w}^{k-2} - \mathbf{w}^*\|^2 && \text{(by (4.40))} \\ &\vdots \\ &\leq (\theta_* C_*^3 / c_*^3) 2^{2-2k} \|\mathbf{w}^0 - \mathbf{w}^*\|^2, && \text{(by (4.40))} \end{aligned}$$

where the third inequality yields the first conclusion in c). This also enable to verify that  $\|F(\mathbf{w}^k; T_k)\| < \epsilon$  if  $k$  satisfies (4.29). The whole proof is completed.  $\square$

*Remark 4.9. Relationship to primal-dual active-set algorithms.* It is interesting to note that when  $\mu_k \equiv \mu$  (a constant) for all indices  $k$ , Algorithm 4.1 shares a similar framework to the primal-dual active-set algorithm in [14, Alg. 1], whose main target is the convex quadratic programming in compressed sensing with  $\ell_1$  regularization. In terms of convergence theory, both Theorem 4.8 and [14, Thm. 2] require the initial point to be close to the interested solution point. There are two key differences. (i) Theorem 4.8 is able to identify the quadratic convergence region  $N(\mathbf{w}^*, \delta_*)$  with  $\delta_*$  being given by (4.28), while [14, Thm. 2] does not have such a characterization and is only about its local convergence (not its convergence rate). (ii) However, [14, Thm. 2] can be globalized via a continuation technique, while it is challenging to globalize Algorithm 4.1 because we are dealing with 0/1-loss function and there are no merit functions available for globalization.

*Remark 4.10. On the choice of the smoothing parameter  $\mu_k$ .* An interesting question raised by one referee is whether the particular choice of  $\mu_k$  in (4.8) may play a role in globalization of Algorithm 4.1. From the smoothing perspective, there exists a number of good strategies to update  $\mu$  as long as it drives  $\mu_k \rightarrow 0$ . For example, we may update  $\mu_k$  by solving the equation  $e^\mu - 1 = 0$  via Newton's method as done in [34], see also [23] for other options. To incorporate such a strategy in a globalization scheme, we must find a merit function to work with. As commented in Remark 4.9, it is not easy to construct a merit function because the composition of the operator  $\|(\cdot)_+\|_0$  with the inequality constraint  $\mathbf{A}\mathbf{x} \leq \mathbf{b}$  leads to the scenario where the sparsity is not over a symmetric set any more. We refer to [3, 29] for detailed discussion on algorithmic advantages of sparsity being over symmetric sets.



**5. Numerical Experiments.** In this part, we will conduct extensive numerical experiments of NM01 in [Algorithm 4.1](#) by using MATLAB (R2019a) on a laptop with 32GB memory and Inter(R) Core(TM) i9-9880H 2.3Ghz CPU, against a few leading solvers for solving SVM and 1-bit CS problems.

**5.1. Experiments for SVM.** There exists a large body of SVM literature. We only focus on the binary classification, which has a training dataset  $\{(\mathbf{a}_i^0, c_i) : i \in \mathbb{N}_m\}$ , with  $\mathbf{a}_i^0 \in \mathbb{R}^{n-1}$  being samples and  $c_i \in \{-1, 1\}$  being the two classes. It is widely recognized that the data are often linearly inseparable and (1.1) is an ideal model to deal with this case with the following setup

$$f(\mathbf{x}) = \|D\mathbf{x}\|^2, \quad A = -[c_1\mathbf{a}_1, \dots, c_m\mathbf{a}_m]^\top, \quad \mathbf{b} = \mathbf{1},$$

where  $D$  is a diagonal matrix with  $D_{ii} = 1, i \in \mathbb{N}_{n-1}$  and  $D_{nn} \geq 0$  (e.g.,  $D_{nn} = 10^{-4}$ ), and  $\mathbf{a}_i = (\mathbf{a}_i^0; 1) \in \mathbb{R}^n, i \in \mathbb{N}_m$ . We will consider two types of datasets: synthetic data and real data described below.

**EXAMPLE 5.1** (Synthetic data in  $\mathbb{R}^2$ ). *Give four samples  $(0, 0), (0, 1), (1, 0), (1, a)$  with labels  $+1, +1, -1, -1$ , where the last point can be treated as an outlier when  $a > 1$ .*

**EXAMPLE 5.2** (Real data in higher dimensions). *We select 40 datasets from three libraries: libsvm, uci and kaggle. All datasets are feature-wisely scaled to  $[-1, 1]$  and all the classes not being 1 are treated as  $-1$ . Their details are presented in [Table 1](#). There are 16 datasets with  $m \leq n$  and 24 datasets with  $m > n$ .*

There are large numbers of methods that have been proposed for SVMs, each with its advantages/disadvantages. It is more reasonable to compare NM01 with those methods that aim at optimizing the 0/1-loss function directly, such as MIP-based methods. However, it is known that MIP-based methods prefer the datasets on small scales (see the numerical experiments reported in [\[32, 38, 39\]](#)) and behave very slowly for the datasets on mediate/large scales, such as most datasets in [Table 1](#). Therefore, we will not include them in the following numerical comparisons.

On the other hand, there is very limited work on developing methods that directly optimize 0/1-loss from the perspective of continuous optimization. Because of this, we are unable to find an available Matlab implementation for such kinds of methods. Hence, we only select five leading solvers, with available Matlab implementations from the machine learning community. These methods solve the surrogate/relaxations of 0/1-loss involved SVMs. They are HSVM from the library `libsvm`<sup>1</sup>[\[9\]](#), SSVM [\[37\]](#) implemented by `liblssvm`<sup>2</sup>[\[33\]](#), RSVM [\[42\]](#), LSVM from the library `liblinear`<sup>3</sup>[\[15\]](#), and FSVM (a MATLAB built-in function `fitclinear`<sup>4</sup>). All involved parameters are set as their default values. To demonstrate the performance of one method, let  $\mathbf{x}$  be its obtained solution and  $A_0 := [\mathbf{a}_1, \dots, \mathbf{a}_m]^\top$ . We will report the CPU Time and the classification accuracy  $\text{Acc}$  defined by  $\text{Acc} := 1 - \|\text{sgn}(A_0\mathbf{x}) - \mathbf{c}\|_0/m$ .

**5.1.1. Implementation of [Algorithm 4.1](#).** We terminate our algorithm if one of the conditions is satisfied:  $k \geq 1000$  or  $\|F(\mathbf{w}^k; T_k)\| < 10^{-4}$ . We initialize  $\mathbf{x}^0 = \mathbf{0}$  and  $\mathbf{z}^0 = \mathbf{1}$ , and set  $\mu_{-1} = 0.05$  if  $m < n$  and  $\mu_{-1} = 5$  otherwise. Moreover, we update  $\mu_k$  by (4.8) with  $\rho = 1$  and  $\alpha = 0.5$  if  $k$  is a multiple of 5. The rest of this part is about setting the parameters  $\tau$  and  $\lambda$ . We try to suggest general principles, but bearing in mind that the best strategy of setting  $\tau$  and  $\lambda$  is problem dependent. For the validation purpose, we conducted the performance comparison of [Algorithm 4.1](#) on four test problems `arce`, `colc`, `dbw1` and `fabc` where we vary one parameter while the other is being fixed. [Fig. 1](#) is for fixed  $\lambda$  and [Fig. 2](#) is for fixed  $\tau$ .

- (i) **For SVM problems.** It follows from (4.4) in [Remark 4.4](#) that if  $2\lambda\tau \leq \min_i\{b_i^2 : b_i > 0\}$ , then  $\mathbf{x}^* = \mathbf{0}$  and  $\mathbf{z}^* = \mathbf{0}$  is a P-stationary point. For SVM, this condition turns into  $2\lambda\tau \leq 1$  since  $\mathbf{b} = \mathbf{1}$ . This phenomenon can be observed in our numerical experiments. For example, zero solutions were obtained by [Algorithm 4.1](#) when  $\tau \leq 1/(2\lambda)$  for fixed  $\lambda = 15$  in [Fig. 1](#) and when  $\lambda \leq 1/(2\tau)$  for fixed  $\tau = 5$  in [Fig. 2](#). Hence, it is recommended to set  $\tau$  and  $\lambda$  to satisfy  $2\lambda\tau > 1$  for SVM problems.

<sup>1</sup><https://www.csie.ntu.edu.tw/~cjlin/libsvm/>

<sup>2</sup><https://www.esat.kuleuven.be/sista/lssvmlab/>

<sup>3</sup><https://www.csie.ntu.edu.tw/~cjlin/liblinear/>

<sup>4</sup><https://mathworks.com/help/stats/fitclinear.html>

Table 1: Descriptions of real datasets.

Data	Datasets	Source	$n$	$m$	Sparse
$m \leq n$					
arce	Arcene	uci	10000	100	No
colc	Colon-cancer	libsvm	2000	62	No
dbw1	Dbworld e-mails	uci	4702	64	Yes
dbw2		uci	3721	64	Yes
dbw3		uci	242	64	Yes
dbw4		uci	229	64	Yes
dext		Dexter	uci	19999	300
dmea	Detect malicious executable	uci	531	373	Yes
doro	Dorothea	uci	100000	800	Yes
dubc	Duke breast-cancer	libsvm	7129	38	No
fabc	Farm ads binary classification	kaggle	54877	4143	Yes
leuk	Leukemia	libsvm	7129	38	No
lsvt	Lsvt voice rehabilitation	uci	310	126	No
newb	News20.binary	libsvm	1355191	19996	Yes
rcvb	Rcv1.binary	libsvm	47236	20242	Yes
scad	Scadi	uci	205	70	No
$m > n$					
aips	Airline passenger satisfaction	kaggle	22	103904	No
ccfd	Credit card fraud detection	kaggle	28	284807	No
covt	Covtype.binary	libsvm	54	581012	Yes
dccc	Default of credit card clients	kaggle	23	30000	No
escd	Email spam classification dataset	kaggle	3000	5172	Yes
gise	Gisette	libsvm	5000	6000	Yes
hepm	Hepmass	uci	28	7000000	No
hfxf	Hedge fund x: financial mod. chal.	kaggle	88	10000	No
higg	Higgs	uci	28	11000000	No
hmeq	Hmeq_data	kaggle	10	5960	No
htru	Htru2	uci	8	17898	No
idac	Ida2016challenge	uci	170	60000	Yes
ijcn	Ijcn1	libsvm	22	49990	Yes
mrpe	Malware analysis datasets: raw pe	kaggle	1024	51959	No
mtpe	Malware analysis datasets: top-1000	kaggle	1000	47580	Yes
ospi	Online shoppers purchasing intention	uci	17	12330	No
pssr	Parkinson speech dataset	uci	26	1039	No
qsot	Qsar oral toxicity	uci	1024	8992	Yes
reas	Real-sim	libsvm	20958	72309	Yes
retb	Real time bidding	kaggle	88	1000000	No
sctp	Santander customer transaction	kaggle	200	200000	No
skin	Skin_nonskin	libsvm	3	245056	No
spli	Splice	libsvm	60	1000	No
susy	Susy	uci	18	5000000	No

- (ii) **On the choice of  $\tau$ .** Despite that a sufficient condition  $0 < \tau < \tau_*$  is provided in [Theorem 4.8](#), it is still difficult to set a proper  $\tau$  as  $\tau_*$  is not known. However, as the condition is sufficient, it is unnecessary to choose it from  $(0, \tau_*)$  strictly. To see its effect, we tested it with varying  $\tau \in [10^{-3}, 10]$ , fixed  $\lambda = 15$  and report the results in [Fig. 1](#). It can be clearly seen that bigger values of  $\tau$  (e.g.,  $\tau \geq 1$ ) lead to better accuracy ACC. An underlying heuristic explanation is as follows: [Algorithm 4.1](#) solves the system [\(4.6\)](#) with index set  $T_k = \mathcal{S}_k \cup \mathcal{E}_k^o$  being decided by the parameter  $\tau$ , see [\(4.1\)](#). We observed that setting  $\tau$  too small often led to infrequent change of  $T_k$  and this often forced the algorithm fell into (possibly undesirable) local regions too quickly. By contrast, setting  $\tau$  slightly bigger enabled altering  $T_k$  frequently enough to make [Algorithm 4.1](#) escape from undesirable local regions so as to achieve better solutions. Since the theoretical convergence is in favour of small values of  $\tau < \tau_*$ , it is not suggested to set the values of  $\tau$  too large.
- (iii) **On the choice of  $\lambda$ .** For the parameter  $\lambda$ , we varied values  $\lambda \in [10^{-2}, 10^2]$  and report its effect in [Fig. 2](#). As expected, zero solutions were achieved when  $\lambda \leq 1/(2\tau)$ . From the left sub-figure, ACCs are in favour of bigger values of  $\lambda > 1/(2\tau)$  which is reasonable since it penalizes the 0/1 loss in [\(1.1\)](#). This choice of  $\lambda$  is consistent with what we have observed in

- (i) above.
- (iv) **Estimating  $\tau_*$ .** Although  $\tau_*$  is unknown, we may be able to numerically estimate it by using the obtained solution and (3.13) provided that some information was available a priori. Note that if  $\mathbf{x}^* = 0$  and  $\mathbf{z}^* = 0$  then  $\tau_* = \min\{\tau_1, \tau_2\} = 1/(2\lambda)$  by (3.13), which can be seen in Fig. 1 and Fig. 2. On the other hand, if a solution satisfies that  $A\mathbf{x}^* + \mathbf{b} < 0$ , then  $\tau_1 = \tau_2 = \infty$  (3.13) and hence  $\tau_* = \infty$ . This phenomena can be observed for datasets `arce` and `colc` since they are linearly separable, see the results for  $\tau > 1/(2\lambda)$  in Fig. 1 and for  $\lambda > 1/(2\tau)$  in Fig. 2.

Therefore, in the following experiments, we set  $\tau = 5$  and  $\lambda = 15$  for simplicity.

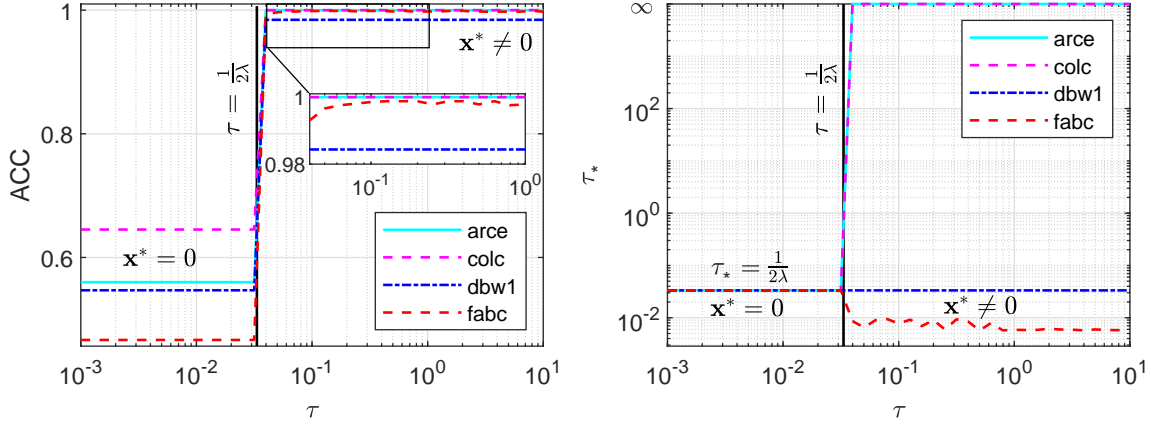


Fig. 1: Effect of  $\tau$  with fixed  $\lambda = 15$  for SVM.

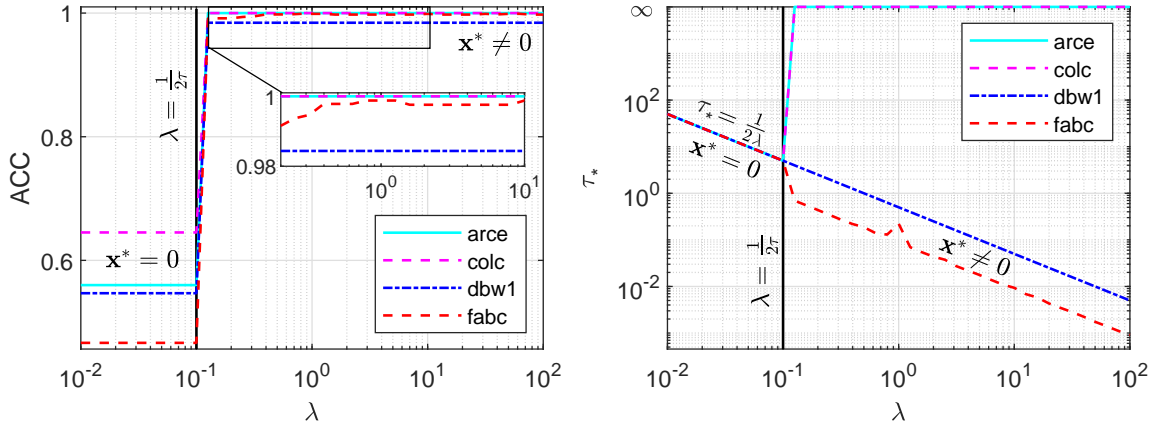


Fig. 2: Effect of  $\lambda$  with fixed  $\tau = 5$  for SVM.

**5.1.2. Numerical comparisons.** We first employ five methods to solve Example 5.1 under different  $a = 1, 10, 100$  to test their robustness to the outliers. For such data, the classifier with a maximum margin is  $x_1^* = 1/2$ . The classifiers by each method are plotted in Fig. 3, where HSVM is omitted since it solves the dual problem and does not provide the solution  $\mathbf{x}$ . Obviously, NM01 finds the true classifiers for all scenarios, while the other methods are influenced significantly by  $a$ .

For Example 5.2, we have 40 datasets with sample size from a few to ten million (e.g., `higg` having 11,000,000 samples). Results of six methods are reported in Table 2, where “—” denotes the results are not obtained if a solver takes too much time or requires a large memory that is out of the capacity of our desktop. For example, HSVM consumes more than 10,000 seconds on the data `covt` and SSVM requires at least 32GB memory to solve `mtpe`. In general, NM01 renders the highest Acc for most datasets. For the computational time, FSVM and LSVM are very fast for datasets of moderate sizes. However, our method is more competitive especially when the data size is in million scale, such as `higg`, `retb`, `susy`, `hepm` with more than  $10^6$  samples, NM01 runs the fastest. For instance, FSVM

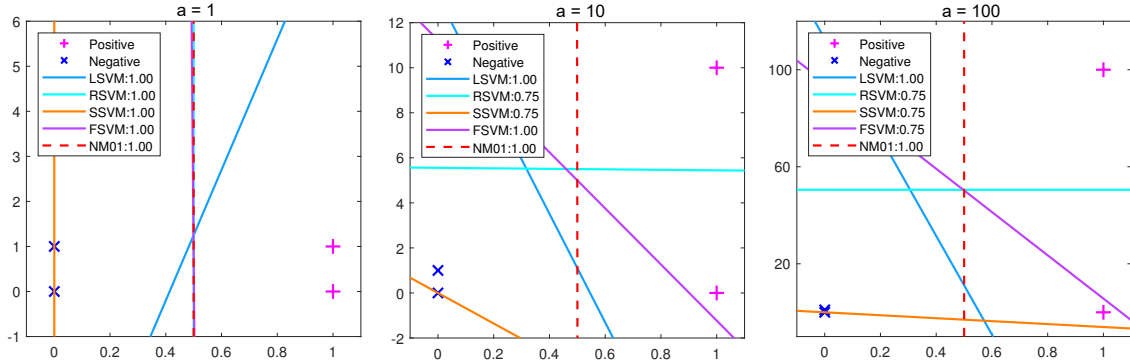


Fig. 3: Robustness to outliers.

and LSVM respectively took 80.69 seconds and 65.65 seconds for `higg`, which is solved by our method within 5.63 seconds.

Table 2: Results of six solvers for `Example 5.2`

data	Acc						Time (seconds)					
	FSVM	HSVM	LSVM	RSVM	SSVM	NM01	FSVM	HSVM	LSVM	RSVM	SSVM	NM01
arce	1.000	1.000	1.000	1.000	1.000	1.000	0.040	0.965	0.077	0.017	8.390	0.051
colc	0.952	1.000	1.000	1.000	1.000	1.000	0.207	0.018	0.015	0.493	0.864	0.008
dbw1	0.984	0.984	0.984	0.984	0.984	0.984	0.043	0.011	0.002	0.061	7.713	0.033
dbw2	0.984	0.984	0.984	0.984	0.984	0.984	0.038	0.010	0.002	0.069	4.787	0.023
dbw3	0.984	0.984	1.000	0.953	1.000	1.000	0.022	0.001	---	0.035	0.115	0.002
dbw4	0.984	0.984	1.000	0.938	1.000	1.000	0.094	0.001	0.001	0.027	0.124	0.005
dext	1.000	1.000	1.000	1.000	1.000	1.000	0.008	0.167	0.009	0.055	81.63	0.029
dmea	1.000	1.000	1.000	0.984	1.000	1.000	0.008	0.010	0.003	0.518	1.100	0.013
doro	1.000	1.000	1.000	1.000	---	1.000	0.026	8.451	0.078	0.564	---	0.163
dubc	1.000	1.000	1.000	1.000	1.000	1.000	0.010	0.035	0.019	0.007	5.969	0.006
fabc	0.996	0.999	0.999	0.994	---	0.999	0.040	8.434	0.194	96.08	---	0.275
leuk	1.000	1.000	1.000	1.000	1.000	1.000	0.010	0.044	0.022	0.006	5.991	0.004
lsvt	0.952	0.984	1.000	0.873	1.000	1.000	0.030	0.007	0.005	0.045	0.141	0.008
newb	0.995	---	0.999	---	---	0.999	0.481	---	2.031	---	---	1.251
rcvb	0.990	0.990	0.997	---	---	0.998	0.092	180.3	0.256	---	---	0.153
scad	0.986	1.000	1.000	0.971	1.000	1.000	0.011	0.001	---	0.015	0.097	0.004
aips	0.876	0.877	0.874	---	---	0.878	0.539	509.6	0.393	---	---	0.028
ccfd	0.999	0.999	0.999	---	---	0.999	7.155	117.8	1.451	---	---	0.209
covt	0.763	---	0.757	---	---	0.764	8.638	---	1.801	---	---	0.444
dccc	0.810	0.809	0.802	---	0.799	0.820	0.098	46.84	0.079	---	139.4	0.011
escd	0.993	0.992	0.996	0.856	0.971	0.996	0.077	8.442	0.113	459.2	13.09	0.228
gise	1.000	1.000	1.000	1.000	1.000	1.000	0.215	63.15	0.451	106.3	1238	0.298
hepm	0.837	---	0.836	---	---	0.840	16.72	---	37.37	---	---	2.789
hfxf	0.589	0.589	0.589	0.572	0.588	0.590	0.065	17.25	0.065	179.1	6.480	0.010
higg	0.641	---	0.641	---	---	0.651	80.69	---	65.65	---	---	5.631
hmeq	0.860	0.859	0.860	0.803	0.862	0.865	0.030	0.687	0.004	59.40	1.618	0.002
htru	0.977	0.977	0.977	---	0.971	0.979	0.038	0.626	0.016	---	25.81	0.006
idac	0.991	0.992	0.992	---	---	0.992	0.376	61.73	0.740	---	---	0.193
ijcn	0.924	0.924	0.923	---	---	0.931	0.223	39.26	0.097	---	---	0.025
mrpe	0.948	---	0.951	---	---	0.951	2.678	---	9.470	---	---	1.324
mtpe	0.968	0.984	0.981	---	---	0.984	0.396	275.5	13.11	---	---	2.185
ospi	0.884	0.884	0.879	---	0.873	0.893	0.066	3.916	0.021	---	10.35	0.006
pssr	0.641	0.643	0.654	0.626	0.647	0.663	0.216	0.063	0.006	1.105	0.089	0.001
qsot	0.946	0.969	0.967	0.849	0.945	0.971	0.058	23.91	0.126	1865	233.9	0.378
reas	0.989	0.989	0.994	---	---	0.994	0.320	741.7	0.505	---	---	0.682
retb	0.998	---	0.998	---	---	0.998	18.77	---	12.57	---	---	0.541
sctp	0.910	---	0.909	---	---	0.914	1.205	---	6.304	---	---	0.358
skin	0.929	0.929	0.924	---	---	0.943	0.172	232.7	0.109	---	---	0.029
spli	0.839	0.839	0.840	0.805	0.840	0.840	0.036	0.148	0.015	0.566	0.100	0.001
susy	0.788	---	0.787	---	---	0.790	18.38	---	22.16	---	---	1.916

**5.2. Simulations for 1-bit CS.** The aim of 1-bit CS is to recover a sparse signal  $\mathbf{x}$  from  $\mathbf{c} = \text{sgn}(A_0\mathbf{x})$ , where  $A_0 := [\mathbf{a}_1, \dots, \mathbf{a}_m]^\top \in \mathbb{R}^{m \times n}$  and  $c_i \in \{1, -1\}, i \in \mathbb{N}_m$ . The original optimization model for 1-bit CS [5] takes the following form:

$$\min \|\mathbf{x}\|_0, \quad \text{s.t.} \quad c_i \langle \mathbf{a}_i, \mathbf{x} \rangle \geq 0, \quad i \in \mathbb{N}_m.$$

Various relaxation methods have been proposed. Here we adopt the smoothing technique using the popular  $\ell_q$  norm ( $0 < q < 1$ ) to approximate the  $\ell_0$  norm [26] and use the 0/1-loss function to deal with the constraints. This leads to the model (1.1) with

$$f(\mathbf{x}) = \sum_{i=1}^n (x_i^2 + \varepsilon^2)^{q/2}, \quad A = -[c_1\mathbf{a}_1, \dots, c_m\mathbf{a}_m], \quad \mathbf{b} = \varepsilon\mathbf{1},$$

where  $\varepsilon > 0, \epsilon > 0$ . Here,  $\mathbf{b} = \epsilon\mathbf{1}$  is adopted from [12]. In our test, we set  $q = 0.5, \epsilon = 0.05$  but update  $\varepsilon$  by  $\varepsilon_0 = 0.5$  and  $\varepsilon_{k+1} = \varepsilon_k/2$ . The test problems are taken from [21] and are described as follows.

**EXAMPLE 5.3.** Rows of  $A_0$  are the independent and identically distributed (iid) samples of  $\mathcal{N}(0, \Sigma)$  with  $\Sigma_{ij} = v^{-|i-j|}, i, j \in \mathbb{N}_n$  and  $v \in (0, 1)$ . The nonzero entries of the ground truth  $s$ -sparse vector  $\mathbf{x}^* \in \mathbb{R}^n$ , namely,  $\|\mathbf{x}^*\|_0 \leq s$ , are generated from the i.i.d. samples of the standard Gaussian distribution  $\mathcal{N}(0, 1)$ , followed by a normalization of  $\mathbf{x}^*$  to be a unit vector. Let  $\mathbf{c}^* = \text{sgn}(A_0\mathbf{x}^*)$  and  $\tilde{\mathbf{c}} = \text{sgn}(A_0\mathbf{x}^* + \xi)$ , where entries of the noise  $\xi$  are the i.i.d. samples of  $\mathcal{N}(0, 0.1^2)$ . Finally, we randomly select  $\lceil rm \rceil$  entries in  $\tilde{\mathbf{c}}$  and flip their signs, and the flipped vector is denoted by  $\mathbf{c}$ , where  $r$  is the flipping ratio.

We report the CPU time, the signal-to-noise ratio  $\text{SNR} := -10\log_{10}(\|\mathbf{x} - \mathbf{x}^*\|^2)$ , the hamming error  $\text{HE} := \|\text{sgn}(A_0\mathbf{x}) - \mathbf{c}^*\|_0/m$ , and the hamming distance  $\text{HD} := \|\text{sgn}(A_0\mathbf{x}) - \mathbf{c}\|_0/m$ , where  $\mathbf{x}$  is the solution obtained by a method. We note that larger SNR (smaller HE, or smaller HD) corresponds to better recovery.

**5.2.1. Implementation and benchmark methods.** The stopping criteria and the rule for updating  $\mu_k$  for Algorithm 4.1 are the same as for the SVM case. We initialize  $\mathbf{x}^0 = 0$  and  $\mathbf{z}^0 = \mathbf{1}$ . We tested the algorithm under different choices of  $\tau$  and  $\lambda$ , and only report the numerical results with  $\tau = 1$  and  $\lambda = 1$ , which satisfy  $2\tau\lambda > \min_i\{b_i^2 : b_i > 0\} = \epsilon^2$  and could yield good overall performance. Moreover, it is observed that the generated solutions had many tiny values, see Fig. 4. Therefore, we apply a refinement step that keeps the  $s$  largest elements in the magnitude of the solution and sets the rest to zeros.

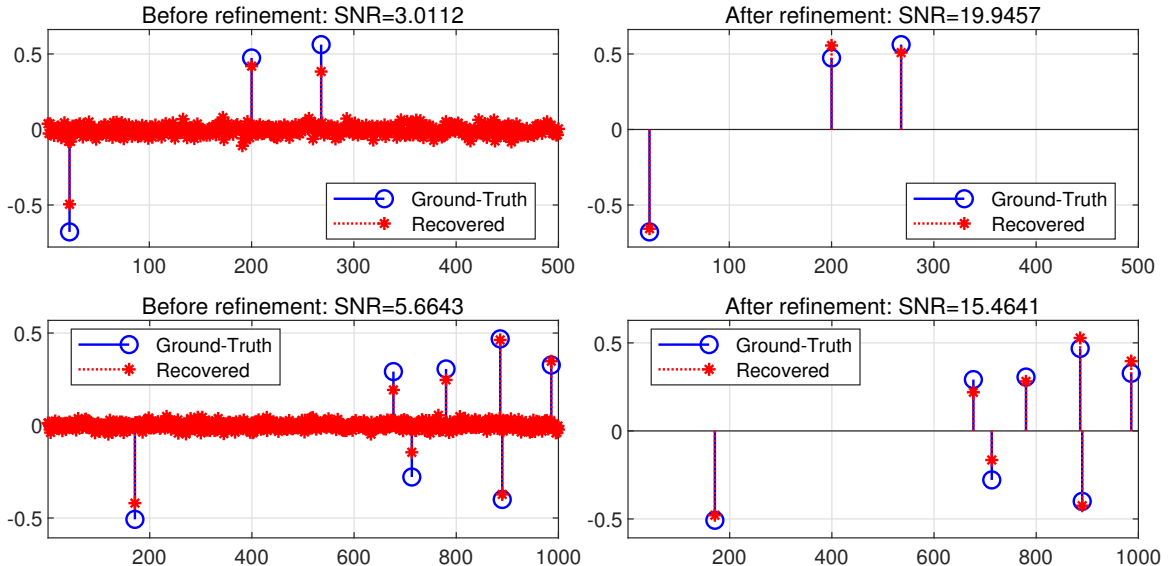
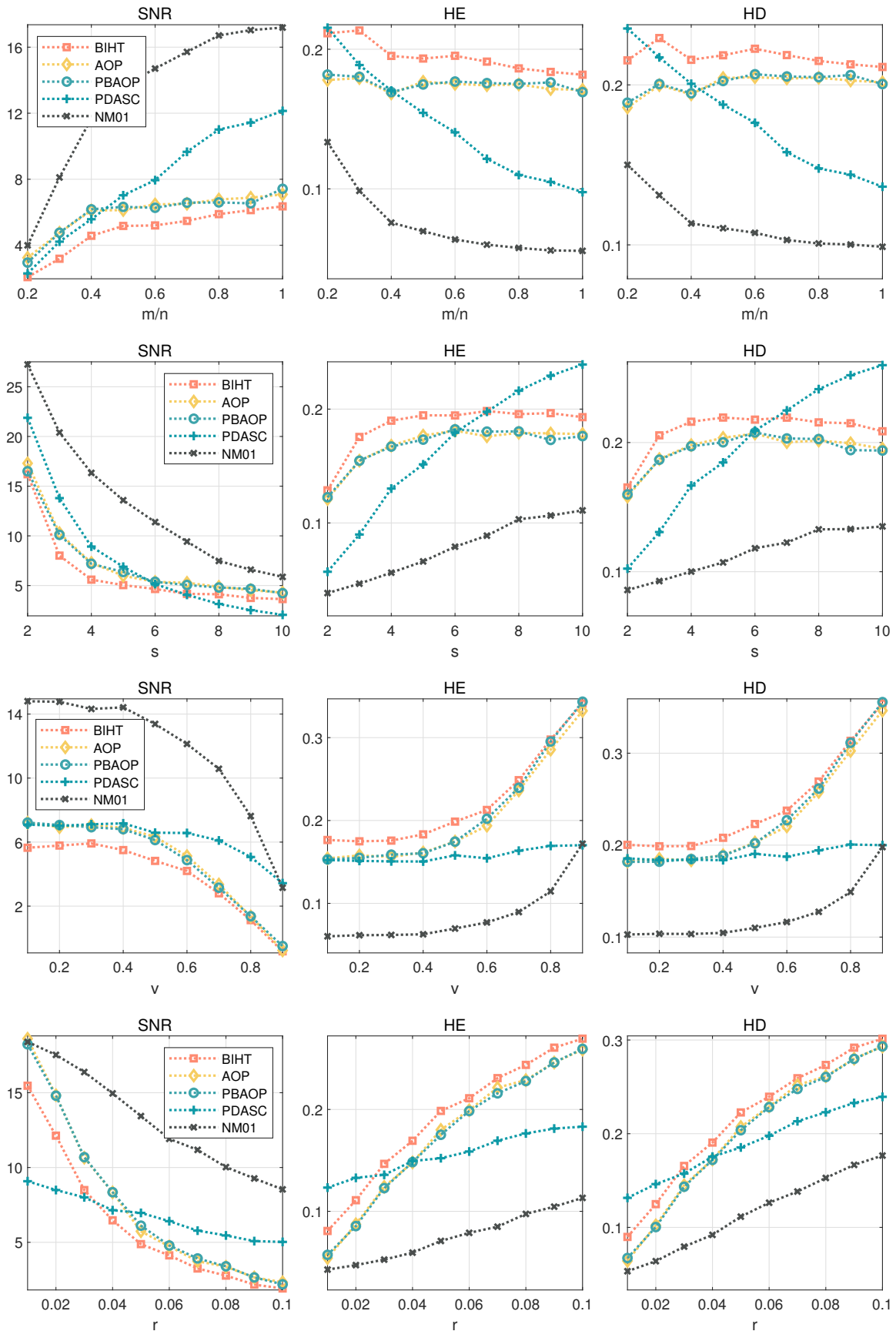


Fig. 4: Refinement of the solution.

Fig. 5: Effects of  $m$ ,  $s$ ,  $v$  and  $r$  for Example 5.3.



Four leading solvers are selected for comparison. They are PDASC<sup>5</sup>[21], BIHT<sup>6</sup>[25], AOP<sup>7</sup>[44] and PBAOP<sup>7</sup>[22], where the last three require to specify the true sparsity level  $s$ , and the last two also need a flipping ratio  $L$ . As in [44], we choose  $L = \text{HD}$ , where HD is the hamming distance generated by BIHT. We also apply the refinement step to PDASC so that all five methods produce  $s$ -sparse solutions. Finally, all methods start with  $\mathbf{x}^0 = 0$  and their solutions are normalized to have a unit length.

**5.2.2. Comparison.** We now apply the five methods to solve [Example 5.3](#) under different scenarios. For each scenario, we report average results over 200 instances if  $n \leq 1000$  and 20 instances otherwise. For small scale instances, we set five parameters as  $(m, n, s, v, r) = (500, 250, 5, 0.5, 0.05)$ . To see the effect of each of these parameters, we tested one parameter while the others being fixed.

- Effect of  $m \in \{0.1, 0.2, \dots, 1\}n$ . We note that the bigger  $m$  enables the better performance, since more samples are available to recover the signal. It can be clearly seen from [Fig. 5](#) that NM01 gets the largest SNR and the smallest HD and HE, leading to a better performance than the others.
- Effect of  $s \in \{2, 3, \dots, 10\}$ . The three sub-figures in the second row of [Fig. 5](#) indicate that it is getting more difficult to recover the ground truth signal when  $s$  increases. In comparison with other methods, NM01 delivers the best recoveries as it achieves the highest SNR and the smallest HD and HE.
- Effect of  $v \in \{0.1, 0.2, \dots, 0.9\}$ . The third row sub-figures in [Fig. 5](#) demonstrate that the bigger values of  $v$  degrade the performance of each method, because each pair of two rows of  $A_0$  is more correlated with increasing  $v$ . It is observed that NM01 delivers the best results when  $v < 0.9$ .
- Effect of  $r \in \{0.02, 0.04, \dots, 0.2\}$ . As expected, the bigger  $r$  is (i.e., the more signs are flipped), the harder the recovery is. This can be seen in the sub-figure in the last row of [Fig. 5](#). NM01 outperforms the others.
- Effect of  $n \in \{2000, 4000, \dots, 10000\}$ . For the higher dimensional instances, we fix  $m = n/2, s = 5n/1000, v = 0.5$  and  $r = 0.05$ . We record the average results in [Table 3](#) where NM01 achieves the most desirable recovery accuracy. For the computational time, the other methods are naturally expected to run super-fast since they belong to the family of greedy methods that exploit the sparse structure of the solutions. Nevertheless, NM01 is relatively competitive in terms of the computational speed.

Table 3: Effect of the higher  $n$  for [Example 5.3](#).

$n$	BIHT	AOP	PBAOP	PDASC	NM01	BIHT	AOP	PBAOP	PDASC	NM01
	SNR					TIME				
2000	7.438	6.159	6.960	8.023	11.37	0.034	0.414	0.176	0.061	0.170
4000	6.509	6.791	6.843	5.545	10.96	0.276	1.832	0.773	0.255	0.880
6000	7.008	7.014	6.967	3.792	10.02	0.803	4.149	2.062	0.636	1.884
8000	7.357	7.436	7.225	3.346	10.01	1.466	7.279	3.778	1.186	3.715
10000	7.841	7.726	7.882	1.489	9.915	2.414	11.76	6.777	2.081	5.675
$n$	HE					HD				
	2000	0.201	0.204	0.206	0.180	0.129	0.170	0.176	0.175	0.145
4000	0.207	0.203	0.201	0.226	0.125	0.177	0.174	0.171	0.198	0.087
6000	0.203	0.204	0.206	0.271	0.134	0.173	0.174	0.176	0.247	0.097
8000	0.205	0.202	0.202	0.285	0.133	0.174	0.171	0.172	0.262	0.094
10000	0.200	0.201	0.197	0.330	0.135	0.168	0.169	0.165	0.312	0.097

*Remark 5.1. (On nonsingularity of the Jacobian matrix.)* We finish this section by discussing the important issue of nonsingularity of the (smoothing) Jacobian matrix  $\nabla F_{\mu_k}(\mathbf{w}^k, T_k)$  used in [Algorithm 4.1](#). As pointed out in Introduction, its nonsingularity is equivalent to the nonsingularity of  $M_k := \nabla^2 f(\mathbf{x}^k) + A_{T_k}^\top A_{T_k} / \mu_k$ . If  $\nabla^2 f(\mathbf{x}^k)$  is positive definite, then  $M_k$  is always nonsingular.

<sup>5</sup><http://jszy.whu.edu.cn/jiaoyuling/en/lwgc/1349484/content/54893.htm#lwgc>

<sup>6</sup><https://laurentjacques.gitlab.io/publication/>

<sup>7</sup><http://www.esat.kuleuven.be/stadius/ADB/huang/downloads/1bitCSLab.zip>



This is the case for the SVM problems tested. For the problem of 1-bit compressed sensing, we let

$$C_* := \nabla^2 f(\mathbf{x}^*) = \text{diag} \left\{ \frac{q [\varepsilon^2 - (1-q)(x_i^*)^2]}{(\varepsilon^2 + (x_i^*)^2)^{2-q/2}}, \quad i = 1, \dots, n \right\},$$

where  $\mathbf{x}^*$  is the limit of the sequence  $\{\mathbf{x}^k\}$ . Suppose  $C_*$  is nonsingular, then  $C_k := \nabla^2 f(\mathbf{x}^k)$  is also nonsingular when  $\mathbf{x}^k$  is close to  $\mathbf{x}^*$ . The Woodbury matrix identity implies that the nonsingularity of  $M_k$  is equivalent to that of the matrix

$$\widehat{M}_k := I + \underbrace{(1/\mu_k)A_{T_k}C_k^{-1}A_{T_k}^\top}_{=: \Delta_k}.$$

Since  $\mu_k$  converges to 0,  $\Delta_k$  cannot have  $(-1)$  as its eigenvalue when  $\mathbf{x}^k$  is close to  $\mathbf{x}^*$  and hence  $M_k$  is always nonsingular. To slightly generalize the above argument, as long as  $\Delta_k$  does not have  $(-1)$  among its eigenvalues,  $\widehat{M}_k$  (hence  $M_k$ ) is always nonsingular. And the chance for  $\Delta_k$  to have  $(-1)$  as its eigenvalue is extremely small in general. This is what we experienced in our test. The argument above does raise the question how to ensure the nonsingularity. It comes back to the globalization issue of [Algorithm 4.1](#). Our proposal is to use a gradient method whenever singularity becomes an issue. We leave this to future research.

**6. Conclusion.** Optimizing the 0/1-loss function has been a challenging task for several decades, and few optimality conditions or theoretical convergence guarantees have been established for most of the 0/1-loss function minimizations. This paper is the first to develop Newton's method with guaranteed quadratic convergence. This has come a long way by first proposing a  $P$ -stationarity condition that leads to stationarity equations, and then establishing the desired convergence results with help of very technical control over the growth of residue equations. The excellent numerical performance of the proposed method for solving the SVM and 1-bit CS problems indicate that it might work well for other related applications. We strongly feel that the techniques developed in this paper can be extended to a more general case, where  $A\mathbf{x} + \mathbf{b}$  in (1.1) is replaced by some non-linear functions.

**Acknowledgements.** We would like to thank both the referees for their detailed comments that have helped to improve the quality of the paper. In particular, we thank one referee for suggesting the current title and the other for pointing out the link of the proposed algorithm to the primal-dual active-set algorithms extensively studied among semi-smooth Newton methods.

#### REFERENCES

- [1] S. M. BAJGIER AND A. V. HILL, *An experimental comparison of statistical and linear programming approaches to the discriminant problem*, Decision Sciences, 13 (1982), pp. 604–618.
- [2] A. BECK AND Y. C. ELDAR, *Sparsity constrained nonlinear optimization: Optimality conditions and algorithms*, SIAM Journal on Optimization, 23 (2013), pp. 1480–1509.
- [3] A. BECK AND N. HALLAK, *On the minimization over sparse symmetric sets: projections, optimality conditions, and algorithms*, Mathematics of Operations Research, 41 (2016), pp. 196–223.
- [4] S. BEN-DAVID, N. EIRON, AND P. M. LONG, *On the difficulty of approximately maximizing agreements*, Journal of Computer and System Sciences, 66 (2003), pp. 496–514.
- [5] P. T. BOUFOUNOS AND R. G. BARANIUK, *1-bit compressive sensing*, in 2008 42nd Annual Conference on Information Sciences and Systems, IEEE, 2008, pp. 16–21.
- [6] J. P. BROOKS, *Support vector machines with the ramp loss and the hard margin loss*, Operations Research, 59 (2011), pp. 467–479.
- [7] J. P. BROOKS AND E. K. LEE, *Analysis of the consistency of a mixed integer programming-based multi-category constrained discriminant model*, Annals of Operations Research, 174 (2010), pp. 147–168.
- [8] E. CARRIZOSA, B. MARTIN-BARRAGAN, AND D. R. MORALES, *Binarized support vector machines*, INFORMS Journal on Computing, 22 (2010), pp. 154–167.
- [9] C. C. CHANG AND C. J. LIN, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology (TIST), 2 (2011), pp. 1–27.
- [10] X. CHEN, L. QI, AND D. SUN, *Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities*, Mathematics of computation, 67 (1998), pp. 519–540.
- [11] C. CORTES AND V. VAPNIK, *Support-vector networks*, Machine Learning, 20 (1995), pp. 273–297.
- [12] D. DAI, L. SHEN, Y. XU, AND N. ZHANG, *Noisy 1-bit compressive sensing: models and algorithms*, Applied and Computational Harmonic Analysis, 40 (2016), pp. 1–32.

- [13] T. EVGENIOU, M. PONTIL, AND T. POGGIO, *Regularization networks and support vector machines*, Advances in computational mathematics, 13 (2000), pp. 1–50.
- [14] Q. FAN, Y. JIAO, AND X. LU, *A primal dual active set algorithm with continuation for compressed sensing*, IEEE Transactions on Signal Processing, 62 (2014), pp. 6276–6285.
- [15] R. FAN, K. CHANG, C. HSIEH, X. WANG, AND C. LIN, *LIBLINEAR: A library for large linear classification*, Journal of Machine Learning Research, 9 (2008), pp. 1871–1874.
- [16] V. FELDMAN, V. GURUSWAMI, P. RAGHAVENDRA, AND Y. WU, *Agnostic learning of monomials by halfspaces is hard*, SIAM Journal on Computing, 41 (2012), pp. 1558–1590.
- [17] J. H. FRIEDMAN, *On bias, variance, 0/1 loss, and the curse-of-dimensionality*, Data Mining and Knowledge Discovery, 1 (1997), pp. 55–77.
- [18] A. K. HAN, *Non-parametric analysis of a generalized regression model: the maximum rank correlation estimator*, Journal of Econometrics, 35 (1987), pp. 303–316.
- [19] T. HASTIE, R. TIBSHIRANI, AND J. FRIEDMAN, *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, 2009.
- [20] M. HINTERMÜLLER, K. ITO, AND K. KUNISCH, *The primal-dual active set strategy as a semismooth Newton method*, SIAM Journal on Optimization, 13 (2002), pp. 865–888.
- [21] J. HUANG, Y. JIAO, X. LU, AND L. ZHU, *Robust decoding from 1-bit compressive sampling with ordinary and regularized least squares*, SIAM Journal on Scientific Computing, 40 (2018), pp. A2062–A2086.
- [22] X. HUANG, L. SHI, M. YAN, AND J. A. SUYKENS, *Pinball loss minimization for one-bit compressive sensing: Convex models and algorithms*, Neurocomputing, 314 (2018), pp. 275–283.
- [23] Z. HUANG, D. SUN, AND G. ZHAO, *A smoothing Newton-type algorithm of stronger convergence for the quadratically constrained convex quadratic programming*, Computational Optimization and Applications, 35 (2006), pp. 199–237.
- [24] K. ITO AND K. KUNISCH, *Semi-smooth Newton methods for state-constrained optimal control problems*, Systems & Control Letters, 50 (2003), pp. 221–228.
- [25] L. JACQUES, J. N. LASKA, P. T. BOUFOUNOS, AND R. G. BARANIUK, *Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors*, IEEE Transactions on Information Theory, 59 (2013), pp. 2082–2102.
- [26] M. LAI, Y. XU, AND W. YIN, *Improved iteratively reweighted least squares for unconstrained smoothed  $\ell_q$  minimization*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 927–957.
- [27] L. LI AND H.-T. LIN, *Optimizing 0/1 loss for perceptrons by random coordinate descent*, in 2007 International Joint Conference on Neural Networks, IEEE, 2007, pp. 749–754.
- [28] J. LIITTSCHWAGER AND C. WANG, *Integer programming solution of a classification problem*, Management Science, 24 (1978), pp. 1515–1525.
- [29] Z. LU, *Optimization over sparse symmetric sets via a nonmonotone projected gradient method*, arXiv preprint arXiv:1509.08581, (2015).
- [30] H. LÜTKEPOHL, *Handbook of matrices*, vol. 1, Wiley Chichester, 1996.
- [31] S. MA AND J. HUANG, *Regularized ROC method for disease classification and biomarker selection with microarray data*, Bioinformatics, 21 (2005), pp. 4356–4362.
- [32] T. NGUYEN AND S. SANNER, *Algorithms for direct 0-1 loss optimization in binary classification*, in International Conference on Machine Learning, 2013, pp. 1085–1093.
- [33] K. PELCKMANS, J. SUYKENS, T. GESTEL, J. BRABANTER, L. LUKAS, B. HAMERS, B. MOOR, AND J. VANDEWALLE, *A Matlab/c toolbox for least square support vector machines*, ESATSCD-SISTA Technical Report, (2002), pp. 02–145.
- [34] H.-D. QI AND L. LIAO, *A smoothing Newton method for general nonlinear complementarity problems*, Computational Optimization and Applications, 17 (2000), pp. 231–253.
- [35] R. T. ROCKAFELLAR AND R. J. WETS, *Variational analysis*, vol. 317, Springer Science & Business Media, 2009.
- [36] P. A. RUBIN, *Solving mixed integer classification problems by decomposition*, Annals of Operations Research, 74 (1997), pp. 51–64.
- [37] J. A. SUYKENS AND J. VANDEWALLE, *Least squares support vector machine classifiers*, Neural Processing Letters, 9 (1999), pp. 293–300.
- [38] Y. TANG, X. LI, Y. XU, S. LIU, AND S. OUYANG, *A mixed integer programming approach to maximum margin 0-1 loss classification*, in 2014 International Radar Conference, IEEE, 2014, pp. 1–6.
- [39] B. USTUN AND C. RUDIN, *Supersparse linear integer models for optimized medical scoring systems*, Machine Learning, 102 (2016), pp. 349–391.
- [40] H. WANG, Y. SHAO, S. ZHOU, C. ZHANG, AND N. XIU, *Support vector machine classifier via  $l_{0/1}$  soft-margin loss*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2021).
- [41] E. W. WEISSTEIN, *Heaviside step function*, <https://mathworld.wolfram.com/>, (2002).
- [42] Y. WU AND Y. LIU, *Robust truncated hinge loss support vector machines*, Journal of the American Statistical Association, 102 (2007), pp. 974–983.
- [43] M. XIE, Y. XUE, AND U. ROSHAN, *Stochastic coordinate descent for 01 loss and its sensitivity to adversarial attacks*, in 2019 18th IEEE International Conference On Machine Learning And Applications, IEEE, 2019, pp. 299–304.
- [44] M. YAN, Y. YANG, AND S. OSHER, *Robust 1-bit compressive sensing using adaptive outlier pursuit*, IEEE Transactions on Signal Processing, 60 (2012), pp. 3868–3875.
- [45] S. ZHAI, T. XIA, M. TAN, AND S. WANG, *Direct 0-1 loss minimization and margin maximization with boosting*, in Advances in Neural Information Processing Systems, 2013, pp. 872–880.
- [46] S. ZHOU, N. XIU, AND H.-D. QI, *Global and quadratic convergence of Newton hard-thresholding pursuit*, Journal of Machine Learning Research, 22 (2021), pp. 1–45.