Support Vector Machine Classifier via $L_{0/1}$ Soft-Margin Loss

Huajun Wang, Yuanhai Shao, Shenglong Zhou, Ce Zhang and Naihua Xiu*

Abstract—Support vector machine (SVM) has attracted great attentions for the last two decades due to its extensive applications, and thus numerous optimization models have been proposed. To distinguish all of them, in this paper, we introduce a new model equipped with an $L_{0/1}$ soft-margin loss (dubbed as $L_{0/1}$-SVM) which well captures the nature of the binary classification. Many of the existing convex/non-convex soft-margin losses can be viewed as a surrogate of the $L_{0/1}$ soft-margin loss. Despite the discrete nature of $L_{0/1}$, we manage to establish the existence of global minimizer of the new model as well as revealing the relationship among its minimizers and KKT/P-stationary points. These theoretical properties allow us to take advantage of the alternating direction method of multipliers. In addition, the $L_{0/1}$-support vector operator is introduced as a filter to prevent outliers from being support vectors during the training process. Hence, the method is expected to be relatively robust. Finally, numerical experiments demonstrate that our proposed method generates better performance in terms of much shorter computational time with much fewer number of support vectors when against with some other leading methods in areas of SVM. When the data size gets bigger, its advantage becomes more evident.

Index Terms—SVM, $L_{0/1}$ soft-margin loss, $L_{0/1}$-proximal operator, minimizers and KKT/P-stationary points, $\mathtt{L}_{0/1}\mathtt{ADMM}$.

1 Introduction

Support vector machine (SVM) was first introduced by Vapnik and Cortes [1] and then has been widely applied into machine learning, statistic, pattern recognition and so forth. The basic idea is to find a hyperplane in the input space that separates the training data set. In the paper, we consider a binary classification problem that can be described as follows. Suppose we are given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, where $\mathbf{x}_i \in \mathbb{R}^n$ are the input vectors and $y_i \in \{-1, 1\}$ are the output labels. The purpose of SVM is to train a hyperplane $\langle \mathbf{w}, \mathbf{x} \rangle + b = w_1x_1 + \cdots + w_nx_n + b = 0$ with $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$ by given training set. For any new input vector $\mathbf{x}'$, we can predict the corresponding label y' as $y' = 1$ for $\langle \mathbf{w}, \mathbf{x}' \rangle + b > 0$ and $y' = -1$ otherwise. In order to find optimal hyperplane, there are two possible cases: linearly separable and inseparable training data. If the training data is able to be linearly separated in the input space, then the unique optimal hyperplane can be obtained by solving a convex quadratic programming (QP) problem:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, i \in \mathbb{N}_m, \end{aligned} \tag{1}$$

where $\mathbb{N}_m := \{1, 2, \cdots, m\}$. Here, the $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)$ provides the distance between the ith sample and the hyperplane. The above model is termed as hard-margin SVM because it requires correct classifications of all samples. When it comes to the training data that are linearly inseparable in the input space, the popular approach is to allow violations in the satisfaction of the constraints in problem (1) and penalize such violations in the objective function, namely,

$$\min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^m \ell(1 - y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)), \tag{2}$$

where $C > 0$ is a penalty parameter and ℓ is one of some loss functions that aim at penalizing some incorrectly classified samples and leaving the other ones. Therefore, the above model allows misclassified samples, and thus is known as soft-margin SVM. Clearly, different soft-margin loss functions yield different soft-margin SVM models. Generally speaking, soft-margin loss functions can be summarized as two categories based on the convexity of ℓ.

1.1 Convex Soft-Margin Losses

Since there are large numbers of convex soft-margin loss function that have been proposed to deal with the soft-margin SVM problems, we only review some popular ones.

- *Hinge loss function:* $\ell_{\rm hinge}(t) = \max\{0, t\} (\forall t \in \mathbb{R})$. It is non-differentiable at $t = 0$ and unbounded. SVM with hinge loss ($\ell_{\rm hinge}$-SVM) was first proposed by Vapnik and Cortes [1], aiming at only penalizing the samples with $t \geq 0$.
- *Pinball loss function:* $\ell^\tau_{\rm pinball}(t) = \max\{t, -\tau t\}$, with $0 \leq \tau \leq 1$, which is still non-differentiable at $t = 0$ and unbounded. SVM with this loss function ($\ell_{\rm pinball}$-SVM) was proposed in [2], [3] to pay penalty for all samples. There is a quadratic programming solver embedded in Matlab to solve the SVM with pinball loss function [3].

- *H.J. Wang, C. Zhang and N.H. Xiu are with the Department of Applied Mathematics, Beijing Jiaotong University, Beijing, China. Email:huajunwang@bjtu.edu.cn, czhang@bjtu.edu.cn, nhxiu@bjtu.edu.cn.*
- *Y.H. Shao is with the School of Management, Hainan University, Haikou, China. Email: shaoyuanhai@hainanu.edu.cn.*
- *S.L. Zhou is with the School of Mathematical Sciences, University of Southampton, Southampton, UK. Email: shenglong.zhou@soton.ac.uk.*
- ** Corresponding author*

Manuscript received xx, xx; revised xx, xx.

- *Hybrid Huber loss function:* $\ell_{\rm HH}^{\tau}(t) = \max\{0, t-\tau\} - (\max\{0, \tau/2 - t^2/2\tau\} - t^2/2)$ with $\tau > 0$. It is differentiable everywhere but still unbounded. This function was first introduced in [4], while SVM with such loss ($\ell_{\rm HH}$-SVM) was first proposed in [5] which can be solved by proximal gradient method [6].
- *Square loss function:* $\ell_{\rm square}(t) = t^2$, a differentiable but unbounded function. SVM with square loss ($\ell_{\rm square}$-SVM) can be found in [7], [8].
- *Some other convex loss functions:* the insensitive zone pinball loss [3], the exponential loss function [9] and log loss function [10].

Since those functions are convex, their corresponding SVM models are not difficult to be dealt with. However, the convexity often induces the unboundedness, which removes robustness of those loss functions to outliers from the training data. In order to overcome such drawback, authors in [11], [12] set an upper bound and enforce the loss to stop increasing after a certain extent. Doing so, the original convex loss functions become non-convex.

1.2 Non-Convex Soft-Margin Losses

Again since there are large numbers of non-convex soft-margin losses that have been studied, which is beyond our scope of review, we only present some of them.

- *Ramp loss function:* $\ell_{\rm ramp}^{\mu}(t) = \max\{0, t\} - \max\{0, t-\mu\}$ with $\mu \geq 0$, which is non-differentiable at $t = \mu$ and $t = 0$ but bounded between 0 and μ. It does not penalize the case when $t < 0$, while pays linear penalty when $0 \leq t \leq \mu$ and a fixed penalty μ when $t > \mu$. This makes this function robust to outliers. Authors in [13] investigated SVM with ramp loss ($\ell_{\rm ramp}$-SVM).
- *Truncated pinball loss function* (truncate left side of pinball loss function): $\ell_{\rm Tpin}^{\tau,\kappa}(t) = \max\{0, (1+\tau)t\} - (\max\{0, \tau(t+\kappa)\} - \tau\kappa)$, with $0 \leq \tau \leq 1$ and $\kappa \geq 0$. It is non-differentiable at $t = -\kappa$ and $t = 0$ and unbounded. The penalty is fixed at κ for $t < -\kappa$ and is linear otherwise. SVM with such loss ($\ell_{\rm Tpin}$-SVM) can be referred in [14].
- *Asymmetrical truncated pinball loss function* (truncate two side of pinball loss function): $\ell_{\rm ATpin}^{\tau,\kappa,\mu}(t) = \max\{0, (1+\tau)t\} - (\max\{0, \tau(t+\kappa)\} + \max\{0, t-\mu\} - \tau\kappa\}$ with $0 \leq \tau \leq 1$ and $\mu, \kappa \geq 0$. This function is non-differentiable at $t = \mu, -\kappa, 0$ but bounded between 0 and $\max(\kappa, \mu)$. The penalty is fixed at κ for $t < -\kappa$ and at μ for $t > \mu$ but is linear otherwise. SVM with such loss ($\ell_{\rm ATpin}$-SVM) was from [15].
- *Sigmoid loss function*: $\ell_{\rm sigmoid}(t) = 1/(1+\exp(-t))$, a differentiable and bounded (between 0 and 1) function. It penalizes all samples. SVM with this loss (sigmoid-SVM) can be seen in [16].
- *Some other non-convex loss function*: normalized sigmoid cost loss function [17].

Compared with convex soft-margin loss, most of non-convex loss functions are less sensitive to feature noises or outliers due to their boundedness. Apparently, non-convexity would lead to difficulties to computations in terms of solving corresponding SVM models. In summary, the basic principles to choose a soft-margin loss are three aspects[18],[19]: (i) It is able to capture the discrete nature of the binary classification. (ii) It is suggested to be bounded to be robust to feature noises or outliers. (iii) It makes itself based SVM model easy to be computed.

1.3 $\ell_{0/1}$ Soft-Margin Loss

Taking above principles into consideration, we now introduce the 0-1 ($\ell_{0/1}$ for short) soft-margin loss defined as

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

The $\ell_{0/1}$ soft-margin loss function is the most nature loss function for binary classification[20],[21]. Its properties are summarized as below.

(i) It is discontinuous at $t = 0$, which captures the discrete nature of the binary classification (correctness or incorrectness) [22].

(ii) It is lower semi-continuous and nonconvex by the definition in [23]. Since it is either 0 or 1, sparsity and robustness will be guaranteed. In fact, it does not count the number of samples with $t < 0$, which leads to sparsity, while returns 1 otherwise, which ensures robustness to outliers.

(iii) It is differentiable everywhere but at $t = 0$. However, it has subdifferential

$$\partial \ell_{0/1}(0) = \mathbb{R}_+ := \{t \in \mathbb{R} : t \geq 0\}$$

and zero gradients elsewhere, see Lemma 2.1, which makes the computation tractable.

1.4 $L_{0/1}$-SVM

For the sake of easing the reading, we present some notations here. Let $\|\mathbf{x}\|$ and $\|\mathbf{x}\|_0$ be the Euclidean norm and the zero norm of $\mathbf{x}$ that counts the number of non-zero elements of $\mathbf{x}$. Denote $A := {\rm Diag}(\mathbf{y})X^\top$ with $X = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_m] \in \mathbb{R}^{n\times m}$ and $\mathbf{y} = (y_1, y_2 \cdots, y_m)^\top \in \mathbb{R}^m$, where ${\rm Diag}(\mathbf{y})$ is a diagonal matrix with diagonal elements being elements in $\mathbf{y}$. For a positive integer m and a vector $\mathbf{u} \in \mathbb{R}^m$, denote

$$\begin{aligned} \mathbb{N}_m &:= \{1, 2, \cdots, m\}, \\ \mathbf{1} &:= (1, 1, \cdots, 1)^\top \in \mathbb{R}^m, \\ \mathbb{R}^m_+ &:= \{\mathbf{u} \in \mathbb{R}^m : u_i \geq 0, i \in \mathbb{N}_m\}, \\ |\mathbf{u}| &:= (|u_1|, \cdots, |u_m|)^\top, \\ \mathbf{u}_+ &:= (\max\{u_1, 0\}, \cdots, \max\{u_m, 0\})^\top. \end{aligned}$$

These notations indicate

$$L_{0/1}(\mathbf{u}) := \|\mathbf{u}_+\|_0 = \sum_{i=1}^m l_{0/1}(u_i), \tag{3}$$

which returns the number of all positive elements in $\mathbf{u}$. We call (3) the $L_{0/1}$ soft-margin loss. Now, replacing ℓ by $\ell_{0/1}$ in (2) and using above notations allow us to rewrite model (2) in a matrix form,

$$\min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} f(\mathbf{w}; b) := \frac{1}{2}\|\mathbf{w}\|^2 + C\|(\mathbf{1} - (A\mathbf{w} + b\mathbf{y}))_+\|_0. \tag{4}$$

We call this model $L_{0/1}$-SVM. The objective function f is lower semicontinuous, non-differentiable and non-convex.

It is difficult to be solved directly by most existing optimization algorithms. Despite that the discrete nature of zero norm makes above model NP-hard to be solved, the $L_{0/1}$-SVM model is an ideal SVM model because it guarantees as few misclassified as possible for binary classification. Therefore, we carry out this paper along with this model.

1.5 Contributions

In this paper, we start to study the theoretical properties of the $L_{0/1}$-SVM model and then design a new efficient and robust algorithm to solve the model. The main contributions of the paper can be summarized as follows.

(i) We prove the existence of a global minimizer of $L_{0/1}$-SVM, which has not been thoroughly studied in prior works. Based on the explicit expressions of subdifferential and proximal operator of the $L_{0/1}$ loss (3), we introduce two types of optimality conditions of the problem: KKT and P-stationary points. We then unravel the relationships among a global/local minimizer and the above two points. This result is essential to our algorithmic design later on.

(ii) We adopt the famous alternating direction method of multipliers (ADMM) to solve the $L_{0/1}$-SVM problem, and thus the method is dubbed as $\mathtt{L_{0/1}ADMM}$. We show that if the sequence generated by the proposed method converges, then it must converge to a P-stationary points. To the best of our knowledge, it is the first time that a method being created aims at solving (4) directly rather than its surrogate model (2). The novelty of the method is using the $L_{0/1}$-support vector operator as a filter to prevent the outliers from being support vectors during training process.

(iii) We compare $\mathtt{L_{0/1}ADMM}$ with other four existing leading methods on solving SVM problems with synthetic and real data sets. Extensive numerical experiments demonstrate that our proposed method achieves better performance in terms of providing higher prediction accuracy, using a small number of support vectors and consuming shorter computational time.

This paper is organized as follows. In Section 2, we will give the explicit expressions of three subdifferentials of $L_{0/1}$ soft-margin loss and derive its proximal operator. Section 3 presents the main theoretical contributions. We will show the existence of a global minimizer to problem (4) as well as investigating the relationships among a global/local minimizer and the KKT/P-stationary points of $L_{0/1}$-SVM problem. In Section 4, we will introduce the $L_{0/1}$-support vector operator and design the algorithm based on the optimality conditions established in previous section. Numerical experiments including comparison with other solvers and concluding remarks are given in the last two sections.

2 Subdifferential and Proximal Operator

To well analyze the properties of the $L_{0/1}$ soft-margin loss, we need introduce the necessary background of the subdifferential and the proximal operator of the $\|\mathbf{u}_+\|_0$.

2.1 $L_{0/1}$ Subdifferential

From [24, Definition 8.3], for a proper and lower semicontinuous function $f:\mathbb{R}^m\to\mathbb{R}$, the regular, limiting and horizon subdiffential are defined respectively as

$$\begin{aligned}
\widehat{\partial} f(\mathbf{u}) = & \left\{\mathbf{v}\in\mathbb{R}^m:\ \liminf_{\substack{\mathbf{z}\to\mathbf{u}\\ \mathbf{z}\neq\mathbf{u}}}\frac{f(\mathbf{z})-f(\mathbf{u})-\langle\mathbf{v},\mathbf{z}-\mathbf{u}\rangle}{\|\mathbf{z}-\mathbf{u}\|}\geq 0\right\},\\
\partial f(\mathbf{u}) = & \ \limsup_{\mathbf{z}\xrightarrow{f}\mathbf{u}}\widehat{\partial} f(\mathbf{z})\\
= & \left\{\mathbf{v}\in\mathbb{R}^m:\ \begin{array}{r}\exists\, \mathbf{z}_j\xrightarrow{f}\mathbf{u},\ \mathbf{v}_j\in\widehat{\partial} f(\mathbf{z}_j)\\ \text{with } \mathbf{v}_j\to\mathbf{v}\end{array}\right\},\\
\partial^\infty f(\mathbf{u}) = & \ \limsup_{\sigma\downarrow 0,\ \mathbf{z}\xrightarrow{f}\mathbf{u}}\sigma\widehat{\partial} f(\mathbf{z})\\
= & \left\{\mathbf{v}\in\mathbb{R}^m:\ \begin{array}{r}\exists\, \mathbf{z}_j\xrightarrow{f}\mathbf{u},\ \mathbf{v}_j\in\widehat{\partial} f(\mathbf{z}_j)\\ \text{with } \sigma_j\mathbf{v}_j\to\mathbf{v}\end{array}\right\},
\end{aligned}$$

where $\sigma\downarrow 0$ means $\sigma>0$ and $\sigma\to 0$, and $\mathbf{z}\xrightarrow{f}\mathbf{u}$ means both $\mathbf{z}\to\mathbf{u}$ and $f(\mathbf{z})\to f(\mathbf{u})$. If the function f is convex, then the limiting subdifferential is also known to the subgradient.

Lemma 2.1. The regular, limiting and horizon subdifferentials of $\|\mathbf{u}_+\|_0$ at $\mathbf{u}$ enjoy following property,

$$\begin{aligned}
\Omega(\mathbf{u}) \ := & \ \widehat{\partial}\|\mathbf{u}_+\|_0=\partial\|\mathbf{u}_+\|_0=\partial^\infty\|\mathbf{u}_+\|_0 \\
= & \left\{\mathbf{v}\in\mathbb{R}^m: v_i\left\{\begin{array}{ll}\geq 0, & u_i=0,\\ =0, & u_i\neq 0,\end{array}\right. i\in\mathbb{N}_m\right\}.
\end{aligned}\tag{5}$$

We use a simple example to illustrate the three subdifferentials of $\|\mathbf{u}_+\|_0$. Consider one dimensional case $m=1$. As shown in Figure 1, the red lines denote some elements in $\partial\|0_+\|_0=\partial\ell_{0/1}(0)$. In fact, all right slashes crossing the origin comprise of the subdifferential $\partial\|0_+\|_0$.

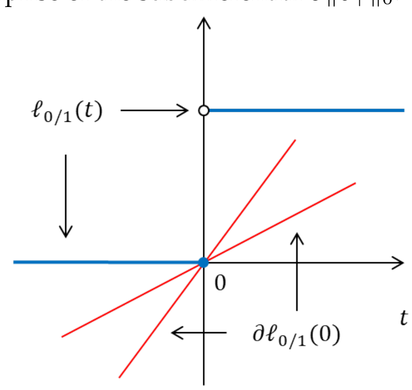

Fig. 1: The $\ell_{0/1}$ soft-margin loss function. The blue line (including the blue original) is the function value and the red lines are two of subdifferentials in $\partial\ell_{0/1}(0)$.

Our next result is about $L_{0/1}$ proximal operator, which will be very useful in designing the algorithm in Section 4.

2.2 $L_{0/1}$ Proximal Operator

By [25, Definition 12.23], the proximal operator of $f:\mathbb{R}\to\mathbb{R}$, associated with a parameter $\alpha>0$, at point $s\in\mathbb{R}$, is defined by

$$\mathrm{Prox}_{\alpha f}(s)=\arg\min_{u\in\mathbb{R}}\ \alpha f(u)+\frac{1}{2}(u-s)^2. \tag{6}$$

The following lemma states that the proximal operator admits a closed form solution when $f=\ell_{0/1}$.

Lemma 2.2 (One-dimensional case). For an $\alpha > 0$, the proximal operator of $\ell_{0/1}(\cdot)$ at s is given by

$$\text{Prox}_{\alpha\ell_{0/1}}(s) := \begin{cases} 0, & 0 \le s < \sqrt{2\alpha}, \\ 0 \text{ or } s, & s = \sqrt{2\alpha}, \\ s, & s > \sqrt{2\alpha} \text{ or } s < 0. \end{cases} \tag{7}$$

It is worth mentioning that the proximal operator may not be unique if $s = \sqrt{2\alpha}$ in (7). However, to guarantee the uniqueness, hereafter, we always choose the proximal operator to be zero if it is not unique. Because of this, the proximal operator of $\ell_{0/1}$ is rewritten as

$$\text{Prox}_{\alpha\ell_{0/1}}(s) = \begin{cases} 0, & 0 \le s \le \sqrt{2\alpha}, \\ s, & \text{otherwise}. \end{cases} \tag{8}$$

The proximal operator of $\ell_{0/1}$ is shown in Figure 2, where the red line denotes the proximal operator.

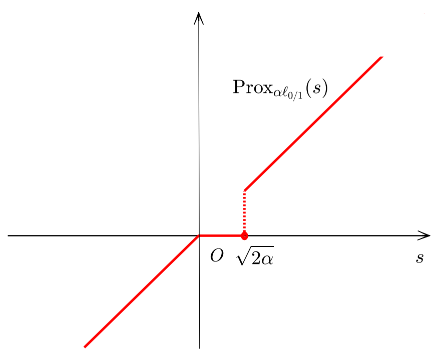

Fig. 2: Demonstration of $\text{Prox}_{\alpha\ell_{0/1}}(s)$.

Based on the one dimensional case, we could derive the proximal operator of $L_{0/1}(\cdot) = \|(\cdot)_+\|_0$. The proof is similar to that of Lemma 2.2 and thus is omitted.

Lemma 2.3 (Multi-dimensional case). For an $\alpha > 0$, the proximal operator of $L_{0/1}$ at $\mathbf{s} \in \mathbb{R}^m$ is given by

$$\text{Prox}_{\alpha L_{0/1}}(\mathbf{s}) = \begin{bmatrix} \text{Prox}_{\alpha\ell_{0/1}}(s_1) \\ \vdots \\ \text{Prox}_{\alpha\ell_{0/1}}(s_m) \end{bmatrix}. \tag{9}$$

To proceed further, we consider the following problem

$$\min_{\mathbf{u}\in\mathbb{R}^m} \ f_C(\mathbf{u}) := h(\mathbf{u}) + C\|\mathbf{u}_+\|_0, \tag{10}$$

where $h : \mathbf{u} \to \mathbb{R}$ is a smooth convex function and gradient Lipschitz continuous with a Lipschitz constant $\tau_h > 0$ and $C > 0$ is given. To see the global solution of above problem, same as [26], we introduce an auxiliary problem,

$$\begin{aligned} \min_{\mathbf{u}\in\mathbb{R}^m} \ f_\gamma(\mathbf{u},\mathbf{z}) \ &:= \ C\|\mathbf{u}_+\|_0 + h(\mathbf{z}) \qquad (11) \\ &+ \ \langle \nabla h(\mathbf{z}), \mathbf{u}-\mathbf{z}\rangle + \frac{1}{2\gamma}\|\mathbf{u}-\mathbf{z}\|^2, \end{aligned}$$

for some $\gamma > 0$ and fixed $\mathbf{z} \in \mathbb{R}^m$, where ∇h is the gradient of h. This problem allows us to acquire the result related to the proximal operator of $L_{0/1}$.

Lemma 2.4. For any given $C > 0$, we have following results.

(i) If $\mathbf{u}^*$ is the global optimal solution to (11) for any fixed $\gamma > 0$ and $\mathbf{z} \in \mathbb{R}^m$, then it holds

$$\mathbf{u}^* = \text{prox}_{\gamma C L_{0/1}}(\mathbf{z} - \gamma\nabla h(\mathbf{z})).$$

(ii) If $\mathbf{u}^*$ is a global optimal solution to (10), then it is also a global optimal solution to (11) with $\mathbf{z} = \mathbf{u}^*$ and $0 < \gamma \le 1/\tau_h$, namely,

$$f_C(\mathbf{u}^*) = f_\gamma(\mathbf{u}^*, \mathbf{u}^*) \le f_\gamma(\mathbf{u}, \mathbf{u}^*), \ \ \forall \mathbf{u} \in \mathbb{R}^m.$$

This lemma suffices to show that a global optimal solution $\mathbf{u}^*$ to (10) must satisfy a fixed point equation, which is well established by following theorem whose proof is easy and is omitted here.

Theorem 2.1. If $\mathbf{u}^*$ is a global optimal solution to (10), then for any given $0 < \gamma \le 1/\tau_h$ it satisfies

$$\mathbf{u}^* \ = \text{prox}_{\gamma C L_{0/1}}(\mathbf{u}^* - \gamma\nabla h(\mathbf{u}^*)). \tag{12}$$

3 Optimality Conditions of $L_{0/1}$-SVM

This section provides the existence of optimal solutions of $L_{0/1}$-SVM and establishes two types of first-order optimality conditions: KKT points and P-stationary points.

3.1 Existence of $L_{0/1}$-SVM Minimizer

Theorem 3.1. Assume b is finite-valued. Then the solution set of (4) is bounded and its global minimizer exists.

We observe that $(\mathbf{w}; b) = (\mathbf{0}; b)$ may be an optimal solution (trivial solution) to (4), which possibly incorrectly predict the corresponding label y' for some new input vector $\mathbf{x}'$ because $\langle \mathbf{w}, \mathbf{x}'\rangle + b = b$. However, for any $b \in \mathbb{R}$, it follows from $y_i \in \{1, -1\}$ that

$$f(\mathbf{0}; b) \ \ = C\|(\mathbf{1} - b\mathbf{y})_+\|_0 = C\min\{m_+, m_-\},$$

where m_+ and m_- denote the number of the positive and the negative labels in $\mathbf{y}$. Based on above equation, this means that any optimal solution $(\mathbf{w}; b)$ satisfying

$$f(\mathbf{w}; b) < C\min\{m_+, m_-\}$$

is a non-trivial optimal solution to (4).

3.2 First-Order Optimality Condition

In this subsection, we discuss the first-order optimality conditions for the problem (4). To proceed this, we introduce a variable $\mathbf{u} \in \mathbb{R}^m$ to equivalently reformulate (4) as

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^m, b\in\mathbb{R}, \mathbf{u}\in\mathbb{R}^m} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\|\mathbf{u}_+\|_0 \\ \text{s.t.} \quad & \mathbf{u} + A\mathbf{w} + b\mathbf{y} = \mathbf{1}. \end{aligned} \tag{13}$$

The Lagrangian function of above problem is

$$\begin{aligned} & L(\mathbf{w}, b, \mathbf{u}, \boldsymbol{\lambda}) \qquad (14) \\ = \ & \frac{1}{2}\|\mathbf{w}\|^2 + C\|\mathbf{u}_+\|_0 + \langle \boldsymbol{\lambda}, \mathbf{u} + A\mathbf{w} + b\mathbf{y} - \mathbf{1}\rangle, \end{aligned}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the Lagrange multiplier, based on which we introduce the well known Karush-Kuhn-Tucker (KKT) point of problem (13).

Definition 3.1 (KKT point of (13)). For a given $C > 0$, we say that $(\mathbf{w}^*; b^*; \mathbf{u}^*)$ is a KKT point of problem (13) if there is a multiplier vector $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ such that

$$\left\{\begin{array}{rcl} \mathbf{w}^* + A^\top \boldsymbol{\lambda}^* &=& \mathbf{0}, \\ \langle \mathbf{y}, \boldsymbol{\lambda}^* \rangle &=& \mathbf{0}, \\ \mathbf{u}^* + A\mathbf{w}^* + b^*\mathbf{y} &=& \mathbf{1}, \\ C\partial \|\mathbf{u}^*_+\|_0 + \boldsymbol{\lambda}^* &\ni& \mathbf{0}. \end{array}\right. \tag{15}$$

The following result reveals the relationship between a local minimizer and a KKT point of (13).

Theorem 3.2. For a given $C > 0$, then $(\mathbf{w}^*; b^*; \mathbf{u}^*)$ is a local minimizer of (13) if and only if it is a KKT point.

Now let us define some notation

$$B := [A \ \mathbf{y}] \in \mathbb{R}^{m\times(n+1)}, \quad H := \begin{bmatrix} I_{n\times n} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix} B^+, \tag{16}$$

where B^+ is the generalized inverse of B. These notations could equivalently rewrite (13) as

$$\min_{\mathbf{u}\in\mathbb{R}^m} \ \frac{1}{2}\|H(\mathbf{u}-\mathbf{1})\|^2 + C\|\mathbf{u}_+\|_0, \tag{17}$$

which is an unconstrained non-convex optimization problem. Based on (17), we will derive the proximal stationary point of (13), and this point is useful as a stop criteria of our algorithm proposed later.

Definition 3.2 (P-stationary point of (13)). For a given $C > 0$, we say $(\mathbf{w}^*; b^*; \mathbf{u}^*)$ is a proximal stationary (P-stationary) point of problem (13) if there is a multiplier vector $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ and constant $\gamma > 0$ such that

$$\left\{\begin{array}{rcl} \mathbf{w}^* + A^T \boldsymbol{\lambda}^* &=& \mathbf{0}, \\ \langle \mathbf{y}, \boldsymbol{\lambda}^* \rangle &=& \mathbf{0}, \\ \mathbf{u}^* + A\mathbf{w}^* + b^*\mathbf{y} &=& \mathbf{1}, \\ \mathrm{prox}_{\gamma C L_{0/1}}(\mathbf{u}^* - \gamma\boldsymbol{\lambda}^*) &=& \mathbf{u}^*. \end{array}\right. \tag{18}$$

We now reveal the relationship between a global minimizer and a P-stationary point of (13). Before which, let

$$\gamma_H := 1/\lambda_{\max}(H^\top H),$$

where $\lambda_{\max}(H^\top H)$ denotes maximum eigenvalue of $H^\top H$.

Theorem 3.3. Assume B has a full column rank. For a given $C > 0$, if $(\mathbf{w}^*; b^*; \mathbf{u}^*)$ is a global minimizer of (13) then it is a P-stationary point with $0 < \gamma \leq \gamma_H$.

Note that B having a full column rank means m $\geq$ n. However, numerical experiments will demonstrate that our proposed algorithm also works for the cases of m<n in terms of finding a P-stationary point. To end this section, we also unravel the relationship between a P-stationary point and a KKT point of (13).

Theorem 3.4. For a given $C > 0$, if $(\mathbf{w}^*; b^*; \mathbf{u}^*)$ is a P-stationary point with $0 < \gamma \leq \gamma_H$ of (13), then it is also KKT point.

The above two theorems state that a global minimizer of (13) is a P-stationary point which is also a KKT point. Most importantly, we could use the P-stationary point as a termination rule in terms of guaranteeing the local optimality of a point generated by the algorithm proposed in next section.

4 Algorithmic Design

In this section, we introduce the concept of $L_{0/1}$-support vector operator and describe how ADMM can be applied into solving the $L_{0/1}$-SVM problem (13).

4.1 $L_{0/1}$-Support Vector Operator

In SVMs, the optimal hyperplane is actually only determined by a small portion of training samples. These samples are called support vectors. It is well known that soft-margin loss functions at non-support vectors have zero subdifferentials [13], [14], [28], [29]. In other words, to select support vectors, one could find samples at which the loss function has nonzero subdifferentials. However, this approach is not suitable for $L_{0/1}$ soft-margin loss, since $\partial\ell_{0/1}(0) = \mathbb{R}_+$ and $\partial\ell_{0/1}(t) = \{0\}$ elsewhere. This indicates samples with $u_i = 1 - y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \neq 0$ always have zero subdifferentials and samples with $u_i = 0$ also have zero subdifferentials due to $0 \in \mathbb{R}_+$, which probably leads to empty set of support vectors. To overcome such drawback, we introduce a novel selection scheme, $L_{0/1}$-support vectors operator, to choose samples to be support vectors.

Definition 4.1 ($L_{0/1}$-support vector operator). For a given $\alpha > 0$, the $L_{0/1}$-support vector operator is defined by

$$T^\alpha(\mathbf{z}) \ := \ \left\{ i \in \mathbb{N}_m : \ \left[\mathrm{prox}_{\alpha L_{0/1}}(\mathbf{z})\right]_i = 0 \right\}. \tag{19}$$

Hereafter, we let $\mathbf{z}_T$ (resp. A_T) be the sub-vector (resp. submatrix) contains elements of $\mathbf{z}$ (resp. rows of A) indexed on T. Let $T := T^\alpha(\mathbf{z})$ and its complementarity set be $\overline{T} := \mathbb{N}_m \setminus T$. It follows from Definition 4.1 and (8) that

$$\begin{bmatrix} (\mathrm{prox}_{\alpha L_{0/1}}(\mathbf{z}))_T \\ (\mathrm{prox}_{\alpha L_{0/1}}(\mathbf{z}))_{\overline{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{z}_{\overline{T}} \end{bmatrix}.$$

This leads to

$$\mathbf{u} = \mathrm{prox}_{\alpha L_{0/1}}(\mathbf{z}) \iff \begin{bmatrix} \mathbf{u}_T \\ \mathbf{u}_{\overline{T}} - \mathbf{z}_{\overline{T}} \end{bmatrix} = \mathbf{0}. \tag{20}$$

The above equivalence will help us to design the algorithm that we are ready to outline as below.

4.2 Framework of ADMM

The augmented Lagrangian function associated with the model (13) can be written as

$$L_\sigma(\mathbf{w}, b, \mathbf{u}, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\|\mathbf{u}_+\|_0 + \langle \boldsymbol{\lambda}, \boldsymbol{\varpi}\rangle + \frac{\sigma}{2}\|\boldsymbol{\varpi}\|^2, \tag{21}$$

where $\boldsymbol{\lambda}$ is Lagrangian multiplier, $\sigma > 0$ is a given parameter and

$$\boldsymbol{\varpi} := \mathbf{u} + A\mathbf{w} + b\mathbf{y} - \mathbf{1}.$$

We take advantage of the ADMM to solve the augmented Lagrangian function. Given the kth iteration $(\mathbf{w}^k, b^k, \mathbf{u}^k, \boldsymbol{\lambda}^k)$, its framework takes the following form

$$\begin{array}{rcl} \mathbf{u}^{k+1} &=& \underset{\mathbf{u}\in\mathbb{R}^m}{\mathrm{argmin}} \ L_\sigma(\mathbf{w}^k, b^k, \mathbf{u}, \boldsymbol{\lambda}^k), \\ \mathbf{w}^{k+1} &=& \underset{\mathbf{w}\in\mathbb{R}^n}{\mathrm{argmin}} \ L_\sigma(\mathbf{w}, b^k, \mathbf{u}^{k+1}, \boldsymbol{\lambda}^k) + \frac{\sigma}{2}\|\mathbf{w}-\mathbf{w}^k\|^2_{D_k}, \\ b^{k+1} &=& \underset{b\in\mathbb{R}}{\mathrm{argmin}} \ L_\sigma(\mathbf{w}^{k+1}, b, \mathbf{u}^{k+1}, \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+1} &=& \boldsymbol{\lambda}^k + \eta\sigma\boldsymbol{\varpi}^{k+1}, \end{array} \tag{22}$$

where $\eta > 0$ is referred as the dual step size and $\boldsymbol{\varpi}^{k+1} := \mathbf{u}^{k+1} + A\mathbf{w}^{k+1} + b^{k+1}\mathbf{y} - \mathbf{1}$. Here,

$$\|\mathbf{w} - \mathbf{w}^k\|_{D_k}^2 = \langle \mathbf{w} - \mathbf{w}^k, D_k(\mathbf{w} - \mathbf{w}^k)\rangle$$

is the so-called proximal term and $D_k \in \mathbb{R}^{n\times n}$ is symmetric. Note that if D_k is positive semidefinite, then the above framework is the standard semi-proximal ADMM [30]. However, authors in papers [31]–[33] have also investigated ADMM with the indefinite proximal terms, namely D_k is indefinite. The basic principle of choosing D_k is to guarantee the convexity of $\mathbf{w}$-subproblem of (22). Since $L_\sigma(\mathbf{w}, b^k, \mathbf{u}^{k+1}, \boldsymbol{\lambda}^k)$ here is strongly convex with respect to $\mathbf{w}$, D_k is able to be chosen as a negative semidefinite matrix. The flexibility of selecting D_k allows us to design a very efficient algorithm when support vectors are used.

4.3 $\mathtt{L}_{0/1}\mathtt{ADMM}$

We mainly describe how each subproblem of (22) can be addressed efficiently as well as how the support vectors can be applied into reducing the computational cost.

(i) Updating $\mathbf{u}^{k+1}$. By (19), we denote

$$\mathbf{z}^k := \mathbf{1} - A\mathbf{w}^k - b^k\mathbf{y} - \boldsymbol{\lambda}^k/\sigma, \quad T_k := T^{C/\sigma}(\mathbf{z}^k). \tag{23}$$

Then the $\mathbf{u}$-subproblem of (22) is reformulated as

$$\begin{aligned} \mathbf{u}^{k+1} &= \underset{\mathbf{u}\in\mathbb{R}^m}{\operatorname{argmin}}\ C\|\mathbf{u}_+\|_0 + \frac{\sigma}{2}\|\mathbf{u} - \mathbf{z}^k\|^2. \\ &= \mathrm{Prox}_{\frac{C}{\sigma}L_{0/1}}(\mathbf{z}^k), \end{aligned}$$

which combining (20) results in

$$\mathbf{u}_{T_k}^{k+1} = \mathbf{0}, \quad \mathbf{u}_{\overline{T}_k}^{k+1} = \mathbf{z}_{\overline{T}_k}^k. \tag{24}$$

(ii) Updating $\mathbf{w}^{k+1}$. We always choose

$$D_k = -A_{\overline{T}_k}^\top A_{\overline{T}_k}, \tag{25}$$

which enables us to derive the $\mathbf{w}$-subproblem of (22) as

$$\begin{aligned} \mathbf{w}^{k+1} &= \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}}\ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\sigma}{2}\|A\mathbf{w} - \mathbf{v}^k\|^2 + \\ &\qquad\qquad \frac{\sigma}{2}\|\mathbf{w} - \mathbf{w}^k\|^2_{-A_{\overline{T}_k}^\top A_{\overline{T}_k}} \\ &= \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}}\ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\sigma}{2}\|A\mathbf{w} - \mathbf{v}^k\|^2 - \\ &\qquad\qquad \frac{\sigma}{2}\|A_{\overline{T}_k}\mathbf{w} - A_{\overline{T}_k}\mathbf{w}^k\|^2, \end{aligned} \tag{26}$$

where $\mathbf{v}^k := -(\mathbf{u}^{k+1} + b^k\mathbf{y} - \mathbf{1} + \boldsymbol{\lambda}^k/\sigma)$. Moreover,

$$\begin{aligned} \mathbf{v}_{\overline{T}_k}^k &= -(\mathbf{u}_{\overline{T}_k}^{k+1} + b^k\mathbf{y}_{\overline{T}_k} - \mathbf{1} + \boldsymbol{\lambda}_{\overline{T}_k}^k/\sigma) \\ &= -(\mathbf{z}_{\overline{T}_k}^k + b^k\mathbf{y}_{\overline{T}_k} - \mathbf{1} + \boldsymbol{\lambda}_{\overline{T}_k}^k/\sigma) \\ &= A_{\overline{T}_k}\mathbf{w}^k, \end{aligned}$$

where the second and third equation are from (24) and (23). Now we rewrite (26) as

$$\begin{aligned} \mathbf{w}^{k+1} &= \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}}\ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\sigma}{2}\|A\mathbf{w} - \mathbf{v}^k\|^2 - \\ &\qquad\qquad \frac{\sigma}{2}\|A_{\overline{T}_k}\mathbf{w} - \mathbf{v}_{\overline{T}_k}^k\|^2 \\ &= \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}}\ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\sigma}{2}\|A_{T_k}\mathbf{w} - \mathbf{v}_{T_k}^k\|^2. \end{aligned} \tag{27}$$

To solve (27), we need find the solution to the equation

$$\underbrace{(I + \sigma A_{T_k}^\top A_{T_k})}_{=:P_k}\mathbf{w} = \sigma A_{T_k}^\top \mathbf{v}_{T_k}^k. \tag{28}$$

Note that $A_{T_k} \in \mathbb{R}^{|T_k|\times n}$, where $|T_k|$ is the cardinality of T_k. Then (28) can be addressed efficiently by following rules:

- If $n \le |T_k|$, one could just solve (28) through
$$\mathbf{w}^{k+1} = \sigma P_k^{-1} A_{T_k}^\top \mathbf{v}_{T_k}^k. \tag{29}$$
- If $n > |T_k|$, the matrix inverse lemma enables us to calculate the inverse as
$$P_k^{-1} = I - \sigma A_{T_k}^\top \underbrace{(I + \sigma A_{T_k} A_{T_k}^\top)}_{=:Q_k}{}^{-1} A_{T_k}. \tag{30}$$

 Then we update $\mathbf{w}^{k+1}$ as

$$\begin{aligned} \mathbf{w}^{k+1} &= \sigma A_{T_k}^\top \mathbf{v}_{T_k}^k - \sigma A_{T_k}^\top Q_k^{-1}\sigma A_{T_k} A_{T_k}^\top \mathbf{v}_{T_k}^k \\ &= \sigma A_{T_k}^\top \mathbf{v}_{T_k}^k - \sigma A_{T_k}^\top Q_k^{-1}(Q_k - I)\mathbf{v}_{T_k}^k \\ &= \sigma A_{T_k}^\top Q_k^{-1}\mathbf{v}_{T_k}^k. \end{aligned} \tag{31}$$

(iii) Updating $\mathbf{b}^{k+1}$. By letting $\mathbf{r}^k := -(A\mathbf{w}^{k+1} - \mathbf{1} + \mathbf{u}^{k+1} + \boldsymbol{\lambda}^k/\sigma)$, it follows from b-subproblem in (22) that

$$\begin{aligned} b^{k+1} &= \underset{b\in\mathbb{R}}{\operatorname{argmin}}\ \frac{\sigma}{2}\|\mathbf{u}^{k+1} - \mathbf{1} + A\mathbf{w}^{k+1} + b\mathbf{y}\|^2 + \langle \lambda^k, b\mathbf{y}\rangle \\ &= \underset{b\in\mathbb{R}}{\operatorname{argmin}}\ \frac{\sigma}{2}\|b\mathbf{y} - \mathbf{r}^k\|^2, \\ &= \langle \mathbf{y}, \mathbf{r}^k\rangle/\|\mathbf{y}\|^2 = \langle \mathbf{y}, \mathbf{r}^k\rangle/m. \end{aligned} \tag{32}$$

(iv) Updating λ^{k+1}. According to (15) and Lemma 2.1, $\boldsymbol{\lambda}$ and $\mathbf{u}$ have the relation $-\boldsymbol{\lambda} \in C\partial\|\mathbf{u}_+\|_0$, namely $\lambda_i = 0$ if $u_i \neq 0$. Based on this, we update the Lagrangian multiplier $\boldsymbol{\lambda}^{k+1}$ in the following way:

$$\left\{ \begin{aligned} \boldsymbol{\lambda}_{T_k}^{k+1} &= \boldsymbol{\lambda}_{T_k}^k + \eta\sigma\boldsymbol{\varpi}_{T_k}^{k+1}, \\ \boldsymbol{\lambda}_{\overline{T}_k}^{k+1} &= \mathbf{0}. \end{aligned} \right. \tag{33}$$

We now summarize the framework of the algorithm in Algorithm 1. We call the method $\mathtt{L}_{0/1}\mathtt{ADMM}$, an abbreviation for $L_{0/1}$-SVM solved by ADMM.

Algorithm 1 : $\mathtt{L}_{0/1}\mathtt{ADMM}$ for solving problem (4)

Initialize $(\mathbf{w}^0, b^0, \mathbf{u}^0, \boldsymbol{\lambda}^0)$. Choose parameters $C, \sigma, K > 0$ and set $k = 0$.
while The halting condition does not hold and $k \le K$ **do**
  Update $T_k := T^{C/\sigma}(\mathbf{z}^k)$ as in (23).
  Update $\mathbf{u}^{k+1}$ by (24).
  Update $\mathbf{w}^{k+1}$ by (29) if $n \le |T_k|$ and by (31) otherwise.
  Update b^{k+1} by (32).
  Update $\boldsymbol{\lambda}^{k+1}$ by (33).
  Set $k = k + 1$.
end while
return the final solution $(\mathbf{w}^k, b^k)$ to (4) .

Remark 4.1. We have some comments on Algorithm 1 regarding to the computational complexity. Note that in each step, updating $\mathbf{w}^{k+1}$ dominates the whole computation, which needs solve a linear equation system (28) through (29) or (31). If $n \le |T_k|$, then the computational complexities of calculating $A_{T_k}^\top A_{T_k}$ and P_k^{-1} are

$\mathcal{O}(n^2|T_k|)$ and $\mathcal{O}(n^\kappa)$ with $\kappa \in (2,3)$, respectively. If $n > |T_k|$, then the computational complexities of calculating $A_{T_k}A_{T_k}^\top$ and Q_k^{-1} are $\mathcal{O}(n|T_k|^2)$ and $\mathcal{O}(|T_k|^\kappa)$ with $\kappa \in (2,3)$, respectively. Overall the whole complexity in each step is $\mathcal{O}(\min\{n^2, |T_k|^2\}\max\{n, |T_k|\})$. Therefore, if there are few number of $L_{0/1}$ support vectors, namely $|T_k|$ is very small, then the complexity is very low, which allows us to do large scale computations.

The following theorem shows that if the sequence generated by $\mathrm{L_{0/1}ADMM}$ converges, then it must converge to a P-stationary point of (13).

Theorem 4.1. Let $(\mathbf{w}^*, b^*, \mathbf{u}^*, \boldsymbol{\lambda}^*)$ be the limit point of the sequence $\{(\mathbf{w}^k, b^k, \mathbf{u}^k, \boldsymbol{\lambda}^k)\}$ generated by $\mathrm{L_{0/1}ADMM}$. Then $(\mathbf{w}^*; b^*; \mathbf{u}^*)$ is a P-stationary point of problem (13) where $\gamma = 1/\sigma$.

Remark 4.2. The convergence result in above theorem is best result that we expect since (13) is non-convex and discrete. Establishment of convergence property of ADMM to address such kind of problem is a very challenging topic for recent decades. There are a few publications that aim at studying ADMM to solve non-convex optimization problems while the established convergence results always rely on heavy assumptions.

Importantly, Theorem 4.1 allows us to take advantage of the P-stationary point as a stopping criteria. In fact, we will terminate the algorithm if the point $(\mathbf{w}^k, b^k, \mathbf{u}^k, \boldsymbol{\lambda}^k)$ closely satisfies the conditions in (18), namely,

$$\max\{\theta_1^k, \theta_2^k, \theta_3^k, \theta_4^k\} < \mathtt{tol},$$

where $\mathtt{tol}$ is the tolerance level and

$$\begin{aligned}
\theta_1^k &:= \frac{\|\mathbf{w}^k + A_{T_k}^\top \boldsymbol{\lambda}_{T_k}^k\|}{1+\|\mathbf{w}^k\|}, \quad \theta_2^k := \frac{|\langle \mathbf{y}_{T_k}, \boldsymbol{\lambda}_{T_k}^k\rangle|}{1+|T_k|},\\
\theta_3^k &:= \frac{\|\mathbf{u}^k - \mathbf{1} + A\mathbf{w}^k + b^k\mathbf{y}\|}{\sqrt{m}},\\
\theta_4^k &:= \frac{\|\mathbf{u}^k - \mathrm{prox}_{C/\sigma L_{0/1}}(\mathbf{u}^k - \boldsymbol{\lambda}^k/\sigma)\|}{1+\|\mathbf{u}^k\|}.
\end{aligned}$$

5 Numerical Experiments

In this part, we will conduct extensive numerical experiments to show sparsity, robustness and effectiveness of our algorithm $\mathrm{L_{0/1}ADMM}$ by using MATLAB (2017a) on a laptop of 32GB of memory and Inter Core i7 2.7Ghz CPU, against four leading methods both on synthetic data and real data.

(a) Parameters setting. In our algorithm, the parameters C and σ control the number of support vectors, see (23), so choosing a good value of these two parameters is crucial. The standard 10-fold cross validation is employed in training set to choose optimal parameters, where the parameters C and σ are both selected from the candidate values $\{a^{-7}, a^{-6}, \cdots, a^7\}$ with $a = \sqrt{2}$. The parameters with highest cross validation accuracy are picked out. In addition, we set $\eta = 1.618$. For the initial points, $\mathbf{w}^0 = 0.01\times\mathbf{1}, b^0 = 0$ and $\mathbf{u}^0 = \boldsymbol{\lambda}^0 = \mathbf{0}$. Finally, the maximum iteration number is $K = 10^4$ and the tolerance level is set as $\mathtt{tol}= 10^{-5}$ on synthetic data and $\mathtt{tol}= 10^{-3}$ on real data.

(b) Benchmark methods. Four leading methods are introduced to make comparisons. All their parameters are optimized to maximize the accuracy by 10-fold cross validation in each training set.

- `HSVM` SVM with hinge soft-margin loss is implemented by LibSVM [34], where the parameter C is selected from the set $\{2^{-7}, 2^{-6}, \cdots, 2^7\} =: \Omega$.
- `SSVM` SVM with square soft-margin loss [7] is implemented by LibLSSVM [35], where the parameter C is picked from the range Ω.
- `PSVM` SVM with pinball soft-margin loss can be achieved by using the traversal algorithm [36], where C and τ are turned from the candidate values $\{0.1, 0.5, 1, 5, 10\}\bigcup\Omega$ and $\{-1, -0.99, \cdots, 0.99\}$, respectively [36]. In order to improve computational efficiency of the traversal algorithm, authors in [36] suggested $\tau = 0$ (i.e., `HSVM`) when the number of training data is large.
- `RSVM` SVM with ramp soft-margin loss can be achieved by employing the `CCCP` [37], where the parameters C and μ are selected from Ω and $\{0.1, 0.2, \cdots, 1\}$.

(c) Evaluation criterions. For the evaluation of classification performances, we report three evaluation criterions of five methods, that is, accuracy (`ACC`), number of support vectors (`NSV`) and `CPU` time (`CPU`). Let $\{\mathbf{x}_j^{\rm test}, y_j^{\rm test}\}_{j=1}^{m_t}$ be m_t test samples data. The testing accuracy is defined by

$$\mathtt{ACC} := 1 - \frac{1}{2m_t}\sum_{j=1}^{m_t}\left|\mathrm{sign}(\langle \mathbf{w}, \mathbf{x}_j^{\rm test}\rangle + b) - y_j^{\rm test}\right|,$$

where $\mathrm{sign}(\overline{a}) = 1$ if $\overline{a} > 0$ and $\mathrm{sign}(\overline{a}) = -1$ otherwise, $(\mathbf{w}, b)$ is obtained by each method. The accuracy measures the ability of a model/method to correctly predict the class labels of any new input vectors. The higher the value of `ACC` is, the better the model/method is. The `NSV` and `CPU` are two comprehensive measures for classification models. The smaller their values are, the better the model is.

5.1 Comparisons with Synthetic Data

In this subsection, we first show that $\mathrm{L_{0/1}ADMM}$ has the ability of support vector selection. For visualization, we consider a two-dimensional example where the features come from Gaussian distributions used in [3], [36].

Example 5.1 (Synthetic data in $\mathbb{R}^2$ without outliers). In this example, samples $\mathbf{x}_i$ with positive labels $y_i = +1$ are drawn from $N(\boldsymbol{\mu}_1, \Sigma_1)$ and samples $\mathbf{x}_i$ with negative labels $y_i = -1$ are drawn from $N(\boldsymbol{\mu}_2, \Sigma_2)$, where $\boldsymbol{\mu}_1 = [0.5, -3]^\top$, $\boldsymbol{\mu}_2 = [-0.5, 3]^\top$ and $\Sigma_1 = \Sigma_2 = \mathrm{Diag}(0.2, 3)$. We generate m samples with two classes having equal numbers, and then evenly split all samples into a training set and a testing set.

Data generated in this way has centralized features of each class. For this experiment, the corresponding Bayes classifier is $x_2 = 2.5x_1$. We display Bayes classifier and 100 training data for each class in Figure 3 (a), where samples are able to be linearly separated and no extra noises contaminate the samples. We then add outliers on data generated in Example 5.1 as follows.

Example 5.2 (Synthetic data in $\mathbb{R}^2$ with outliers). Firstly, m samples with two classes having equal numbers are generated as in Example 5.1. Then in each class, we randomly flip r percentage of labels. For instance, in $m/2$ samples with positive labels $+1$, we change $mr/2$ labels of them to -1. This means r percentage of m samples are flipped their labels, namely rm outliers are generated. Finally, we again evenly split those samples into a training set and a testing set. In Figure 3 (b), one training set with r=10% outliers are produced.

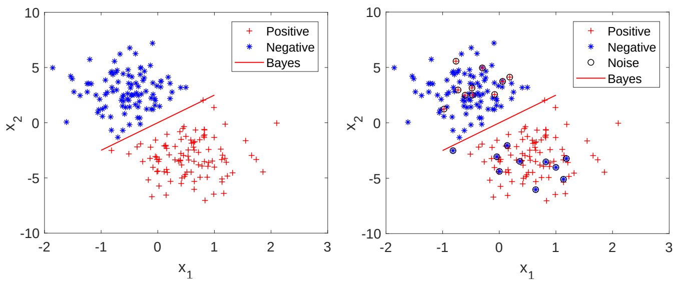

Fig. 3: Blue stars: sampling points in class -1. Red crosses: sampling points in class $+1$. Red dashed lines: the Bayes classifier. (a) A two dimensions training set with $m = 200$ samples. (b) Data in (a) but with r=10% outliers.

To solve these two examples, five methods are applied to calculate the classification boundary $x_2 = w_1x_1 + b$. Since data are generated randomly, we repeat above process 10 times to avoid randomness and report average results of ACC, NSV and CPU.

(d) Synthetic data without outliers. We first compare five methods for solving Example 5.1, where $m \in \{4000, 8000, \cdots, 20000\}$. Average results are reported in Table 1. It can be clearly seen that all methods achieved desirable ACC and $\mathrm{L}_{0/1}$ADMM got slightly better ones. When it comes to NSV, the picture is significant different. $\mathrm{L}_{0/1}$ADMM used a very small portion of samples as the support vectors, while SSVM and PSVM used all samples. Therefore, the phenomenon manifests that our constructed $L_{0/1}$ support vector operator is very effective to choose informative samples as the support vectors. As we mentioned in Remark 4.1, a small portion of samples used will greatly speed up the computation. This is testified by very short CPU time taken by $\mathrm{L}_{0/1}$ADMM. Apparently, PSVM and RSVM consumed much longer time, which indicates these two methods would suffer from computational slowness in large scale date settings.

(e) Synthetic data with outliers. In the following experiment, we test five methods for solving Example 5.2, with fixing $m = 10000, n = 2, r \in \{0, 0.05, 0.1, 0.15, 0.2\}$. Average results are presented in Table 2. Again, there is no big difference of ACC generated by five methods. When more outliers were added, ACC became smaller. In addition, $\mathrm{L}_{0/1}$ADMM got slightly better ACC, which means it is more robust to outliers than other methods. As for NSV, SSVM and PSVM again took all samples. Compared with solving Example 5.1, HSVM this time used more support vectors and NSV increased when more outliers added, which means it is sensitive to the outliers. By contrast $\mathrm{L}_{0/1}$ADMM and RSVM seem to be more robust to the outliers since NSVs did not vary greatly with r altering. Interestingly, being different with HSVM, these two methods needed fewer support vectors when more outliers added. Finally, $\mathrm{L}_{0/1}$ADMM always ran the fastest, with only

TABLE 1: Comparisons of five methods for solving Ex. 5.1.

	$m/2$	$\mathrm{L}_{0/1}$ADMM	HSVM	SSVM	PSVM	RSVM
ACC(%)	2000	97.05	97.05	97.00	97.05	97.05
	4000	97.35	97.25	97.30	97.30	97.33
	6000	97.33	97.28	97.33	97.24	97.33
	8000	96.96	96.91	96.89	96.91	96.96
	10000	97.20	97.18	97.16	97.19	97.20
NSV	2000	6	187	2000	2000	96
	4000	10	301	4000	4000	141
	6000	17	439	6000	6000	201
	8000	20	571	8000	8000	223
	10000	22	658	10000	10000	240
CPU (seconds)	2000	0.003	0.014	0.221	9.642	3.969
	4000	0.006	0.022	0.626	67.58	16.29
	6000	0.009	0.036	1.200	209.9	31.44
	8000	0.013	0.069	2.342	493.2	65.25
	10000	0.019	0.094	3.951	775.3	124.7

taking less than 0.01 seconds, followed by HSVM and SSVM. Same as solving such data without outliers, PSVM consumed quite long CPU time. This implies that it may suffer from severe computational slowness for data with large size.

TABLE 2: Comparisons of five methods for solving Ex. 5.2

	r	$\mathrm{L}_{0/1}$ADMM	HSVM	SSVM	PSVM	RSVM
ACC(%)	0.00	97.16	97.08	97.10	97.16	97.16
	0.05	92.65	92.46	92.50	92.60	92.65
	0.10	87.98	87.78	87.78	87.90	87.90
	0.15	83.06	82.86	82.80	82.98	83.06
	0.20	78.32	78.16	78.12	78.28	78.28
NSV	0.00	19	364	5000	5000	184
	0.05	19	947	5000	5000	175
	0.10	17	1385	5000	5000	170
	0.15	14	1795	5000	5000	161
	0.20	12	2160	5000	5000	137
CPU (seconds)	0.00	0.008	0.027	0.801	93.11	22.53
	0.05	0.007	0.075	0.823	101.3	20.99
	0.10	0.006	0.123	0.853	105.4	19.43
	0.15	0.006	0.172	0.885	108.3	18.96
	0.20	0.005	0.236	0.898	110.6	18.41

5.2 Comparisons with Real Data

We now focus on applying five methods into solving 13 real data sets. Table 3 presents the detailed information of them, where the last five ones have the training and testing data.

Example 5.3 (Real data without outliers). We perform 10-fold cross validation for the first six data sets, where each data is randomly split into ten parts, one of which is used for testing and the remaining nine parts is for training. We thus record average results to evaluate the performance. However, for the two large size samples: SUSY and HIGGS, the last 500,000 samples are used for testing, and the rest are for training. In our experiments, all features in each data set are scaled to $[-1, 1]$.

Example 5.4 (Real data with outliers). We still use these 13 real data sets in Example 5.3 but with adding outliers.

TABLE 3: Descriptions of 13 real data sets

	Training data	Testing data	Features
Datesets	m	m_t	n
Colon-cancer (col)	62	0	2000
Australian (aus)	690	0	14
Two-norm (two)	7400	0	20
Mushrooms (mus)	8124	0	112
Adult (adu)	17887	0	13
Covtype.binaty (cov)	581012	0	54
SUSY (sus)	5000000	0	18
HIGGS (hig)	11000000	0	28
Lekemia (lek)	38	34	7129
Splice (spl)	1000	2175	60
W6a (w6a)	17188	32561	300
W8a (w8a)	49749	14951	300
ijcnn1 (ijc)	49990	91701	22

For each data set, we randomly pick r percentage of training samples and then flip their labels. Same procedure is also applied into testing samples.

(f) Real data without outliers. Average results of five methods are recorded in Table 4. Note that some large size data sets make the other four methods run too much time, (e.g. over than one hour), so we do not report theirs results relating to those data sets. Clearly, $\text{L}_{0/1}$ADMM outperformed others in terms of biggest ACC, smallest NSV and shortest CPU for the most of data sets. More detailed, $\text{L}_{0/1}$ADMM and RSVM got better ACC than the other three methods. For instance, they predicted almost 90% samples correctly for col testing data whilst HSVM and PSVM only got less than 80% correct predictions. In terms of using support vectors, SSVM and PSVM again took all samples into consideration. By contrast, $\text{L}_{0/1}$ADMM made use of a few number of support vectors, e.g. 113 v.s. 1247 by RSVM for adu data. As what we expected, $\text{L}_{0/1}$ADMM ran much faster than other methods for large size data sets because of small number of support vectors being used. For instance, 0.573 seconds v.s. 36.95 seconds by HSVM for ijc data. In addition, it only took 14.26 seconds to get the solution for hig data with more than ten million samples. This demonstrated that $\text{L}_{0/1}$ADMM is capable of dealing with data in extremely large scales.

(g) Real data with outliers. Finally, we would like to see the performance of each method on solving the real date sets with outliers, namely Example 5.4. We choose different ratios r from $\{0.01, 0.02, \cdots, 0.1\}$. As reported in Table 4, the other four methods suffered from the computational slowness for data sets with large sizes, thus we only present results of six data sets with small sizes: col, aus, two, mus, lek and spl. In terms of the accuracy in Figure 4, ACC obtained by all methods dropped down with r ascending, namely, more outliers being added. Generally speaking, $\text{L}_{0/1}$ADMM got the highest ACC except for spl, followed by RSVM. As for NSV in Figure 5, SSVM and PSVM always took all samples. It can be seen that lines from $\text{L}_{0/1}$ADMM and RSVM did not go up when r rose, which means they were quite robust to r, namely robust to the outliers. By contrast, more support vectors were needed by HSVM due to the rising of NSV when r got increased. For each data set and each r, $\text{L}_{0/1}$ADMM always used the fewest support vectors, followed by RSVM and HSVM. When it comes to the CPU time in Figure 6, since col and lek have very small sizes, all methods got solutions quickly. While for other four data sets with moderate sizes, $\text{L}_{0/1}$ADMM ran fastest, and PSVM and RSVM came the last, such as, less than 0.1 second by $\text{L}_{0/1}$ADMM v.s. more than 100 seconds by PSVM and RSVM.

6 CONCLUSION

In this paper, we proposed a new soft-margin SVM model with the $L_{0/1}$ soft-margin loss function. It well captures the nature of the binary classification. The establishment of its optimality conditions made this NP-hard problem tractable. We then took advantage of the negative semidefinite proximal ADMM to solve this problem. The creation of $L_{0/1}$ support vectors greatly reduced the computational complexity. Extensive numerical experiments demonstrated that our proposed method enjoys high order of accuracy and super fast computational speed. What is more, since it only took very small number of support vectors into consideration, the proposed method turns out to be very robust to the outliers. The idea of using $L_{0/1}$ soft-margin loss function might be able to extend to deal with the different types of SVM models, such as SVM [38]–[41], which severely suffers from outliers. It is also interesting to see how similar method and techniques can be designed to solve the kernel SVM problems. We leave this topic as a future research.

APPENDIX A
PROOFS OF ALL THEOREMS

A.1 Proof of Lemma 2.1

Denote $\mathbb{I}(\mathbf{u}) := \{i \in \mathbb{N}_m : u_i = 0\}$ and $\Psi(\mathbf{u}) := \{\mathbf{z} \in \mathbb{R}^m : z_i \le 0,\ \forall\, i \in \mathbb{I}(\mathbf{u})\}$. We split the proof of the lemma into the following two case:

Case 1: $\mathbf{u} = 0$. For any $\mathbf{z} \in \mathbb{R}^m$, it holds that $\|\mathbf{z}_+\|_0 - \|\mathbf{u}_+\|_0 - \langle \mathbf{v}, \mathbf{z} - \mathbf{u}\rangle = -\langle \mathbf{v}, \mathbf{z} - \mathbf{u}\rangle \ge 0$ for any $\mathbf{z} \in \Psi(\mathbf{u})$ and $\mathbf{z}$ sufficiently closed to $\mathbf{u}$. From the definition of the regular, limiting and horizon subdifferentials, we have

$$\widehat{\partial}\|\mathbf{u}_+\|_0 = \partial\|\mathbf{u}_+\|_0 = \partial^{\infty}\|\mathbf{u}_+\|_0 = \mathbb{R}^m_+.$$

Case 2: $\mathbf{u} \ne 0$. Since $\|\mathbf{u}_+\|_0$ is lower-semicontinuous at $\mathbf{u}$, there is a neighborhood $\mathrm{U}(\mathbf{u}, \delta)$ of $\mathbf{u}$ such that $\|\mathbf{u}_+\|_0 \le \|\mathbf{z}_+\|_0$ for all $\mathbf{z} \in \mathrm{U}(\mathbf{u}, \delta)$ with $\delta > 0$. By the definition of the regular subdifferential of $\|\mathbf{u}_+\|_0$, we only need to consider some sequence $\mathbf{z}_j \in \mathrm{U}(\mathbf{u}, \delta) \cap \Psi(\mathbf{u})$ such that $\mathbf{z}_j \to \mathbf{u}$ and $\|(\mathbf{z}_j)_+\| = \|\mathbf{u}_+\|_0$. For all such sequence $\{\mathbf{z}_j\}$, we have

$$\|(\mathbf{z}_j)_+\|_0 - \|\mathbf{u}_+\|_0 - \langle \mathbf{v}, \mathbf{z}_j - \mathbf{u}\rangle = -\langle \mathbf{v}, \mathbf{z}_j - \mathbf{u}\rangle \ge 0$$

if and only if $\mathbf{v} \in \Omega(\mathbf{u})$. Hence, $\widehat{\partial}\|u_+\|_0 = \mathbf{v} \in \Omega(\mathbf{u})$. From the definition of the limiting subdifferential, letting $f := \|(\cdot)_+\|_0$, we have

$$\begin{aligned}\partial\|\mathbf{u}_+\|_0 &= \limsup_{\mathbf{z} \xrightarrow{f} \mathbf{u}} \widehat{\partial}\|\mathbf{z}_+\|_0 \\ &= \limsup_{\mathbf{z} \xrightarrow{f} \mathbf{u}} \{\mathbf{v} \in \mathbb{R}^m : \mathbf{v} \in \Omega(\mathbf{z})\} = \Omega(\mathbf{u}).\end{aligned}$$

TABLE 4: Comparisons of five methods for solving Ex. 5.3, where $L_{0/1}$ stands for $\mathtt{L}_{0/1}\mathtt{ADMM}$.

	ACC(%)					NSV					CPU(seconds)				
Name	$L_{0/1}$	HSVM	SSVM	PSVM	RSVM	$L_{0/1}$	HSVM	SSVM	PSVM	RSVM	$L_{0/1}$	HSVM	SSVM	PSVM	RSVM
col	90.23	64.52	85.48	77.69	89.68	34	46	54	54	38	0.021	0.009	0.001	0.010	0.003
aus	86.23	85.51	85.80	85.80	86.02	24	203	621	621	89	0.005	0.014	0.033	0.874	0.650
two	98.37	98.02	97.97	97.97	98.24	30	758	6600	6600	108	0.054	0.265	2.506	516.7	139.2
mus	100.0	100.0	100.0	100.0	100.0	135	550	7311	7311	506	0.074	0.997	3.419	769.5	153.4
adu	83.90	83.29	83.01	83.07	83.79	113	6379	16098	16098	1247	0.576	3.775	24.58	1633.4	1013.2
lek	82.35	58.82	79.41	58.82	76.47	26	31	38	38	29	0.072	0.057	0.004	0.010	0.008
spl	85.52	88.97	85.75	85.52	85.47	70	607	1000	1000	87	0.043	0.117	0.083	7.976	0.631
w6a	97.93	97.21	97.58	97.21	97.86	429	1128	17188	17188	946	0.226	1.532	170.9	5947.2	2747.4
w8a	98.54	98.27	-	-	-	867	2857	-	-	-	2.576	64.33	-	-	-
ijc	94.33	92.73	-	-	-	215	8508	-	-	-	0.573	36.95	-	-	-
cov	71.79	-	-	-	-	137	-	-	-	-	3.870	-	-	-	-
sus	67.58	-	-	-	-	730	-	-	-	-	10.38	-	-	-	-
hig	65.21	-	-	-	-	1338	-	-	-	-	14.26	-	-	-	-

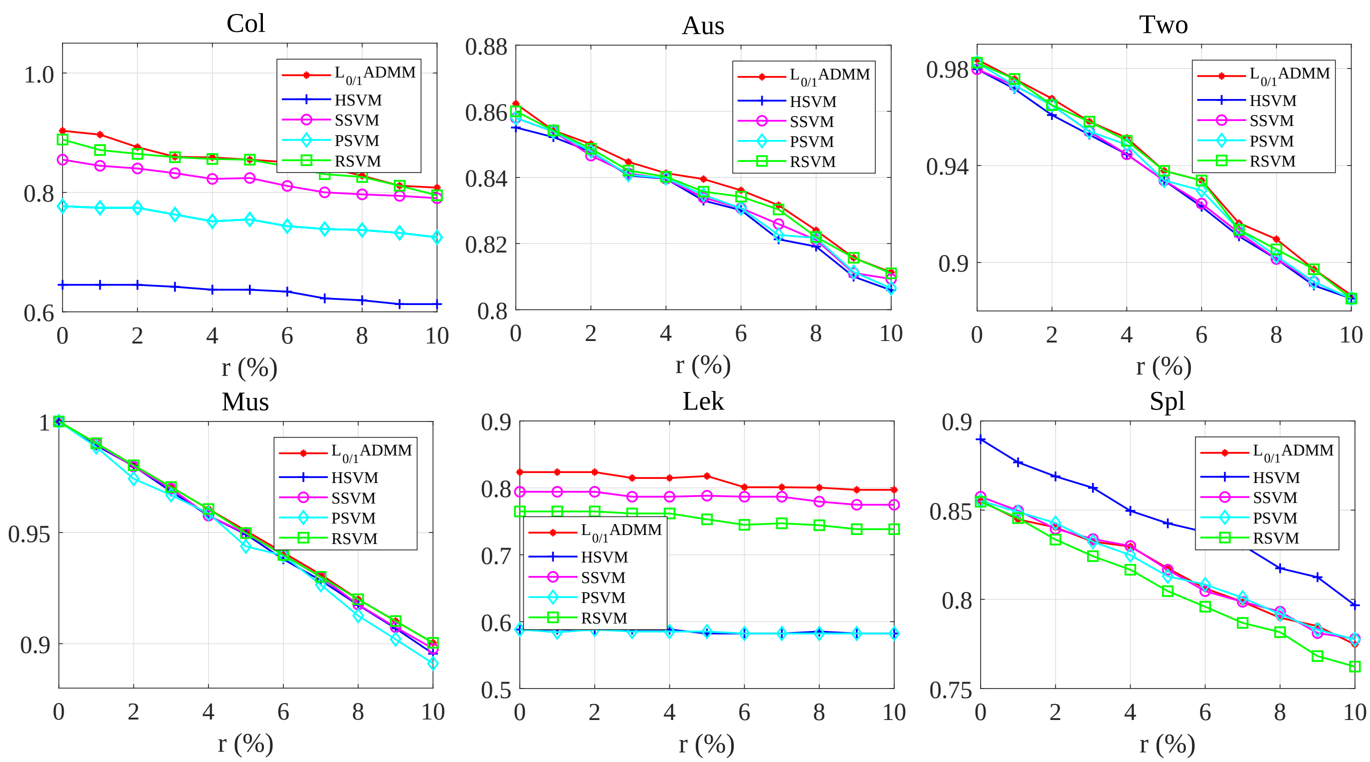

Fig. 4: ACC v.s. r of five methods for solving six data sets.

Similarly, the horizon subdifferential of $\|\mathbf{u}_+\|_0$ is given as the following,

$$\begin{aligned}\partial^\infty\|\mathbf{u}_+\|_0 &= \limsup_{\sigma\downarrow 0,\ \mathbf{z}\xrightarrow{f}\mathbf{u}} \ \sigma\widehat{\partial}\|\mathbf{z}_+\|_0\\ &= \limsup_{\sigma\downarrow 0,\ \mathbf{z}\xrightarrow{f}\mathbf{u}} \ \{\sigma\mathbf{v}\in\mathbb{R}^m : \mathbf{v}\in\Omega(\mathbf{z})\}\\ &= \limsup_{\sigma\downarrow 0,\ \mathbf{z}\xrightarrow{f}\mathbf{u}} \ \{\sigma\mathbf{v}\in\mathbb{R}^m : \sigma\mathbf{v}\in\Omega(\mathbf{z})\}\\ &= \limsup_{\mathbf{z}\xrightarrow{f}\mathbf{u}} \ \{\mathbf{v}\in\mathbb{R}^m : \mathbf{v}\in\Omega(\mathbf{z})\} = \Omega(\mathbf{u}),\end{aligned}$$

where the third equation is due to $\mathbf{v}\in\Omega(\mathbf{z})$ being equivalent to $\sigma\mathbf{v}\in\Omega(\mathbf{z})$ for any $\sigma>0$. $\square$

A.2 Proof of Lemma 2.2

It follows from (6) that

$$\mathrm{Prox}_{\alpha\ell_{0/1}}(s) = \arg\min_{u\in\mathbb{R}} \alpha\ell_{0/1}(u) + (u-s)^2/2.$$

Let $\phi(u) := \alpha\ell_{0/1}(u) + (u-s)^2/2$. Since $\phi_1(u) := \alpha + (u-s)^2/2$ for $u>0$ and $\phi_2(u) := (u-s)^2/2$ for $u<0$ are strongly convex and twice continuously differentiable, the unique minimal values of $\phi_1(u)$ and $\phi_2(u)$ are both attained at $u=s$. Moreover, $\phi_3(u) := (u-s)^2/2$ for $u=0$, we have $\phi_3(0) = s^2/2$. The rest part is to compare the three values $\phi_1(s)$ with $s>0$, $\phi_2(s)$ with $s<0$ and $\phi_3(0)$: (i) Since $s>\sqrt{2\alpha}\Leftrightarrow\phi_3(0)>\phi_1(s)$ and $\phi_2(s)>\phi_1(s)$, we can observe that the minimal value of the $\phi(u)$ is achieved at $u=s$. (ii) Since $0\le s<\sqrt{2\alpha}\Leftrightarrow\phi_1(s)>\phi_3(0)$ and

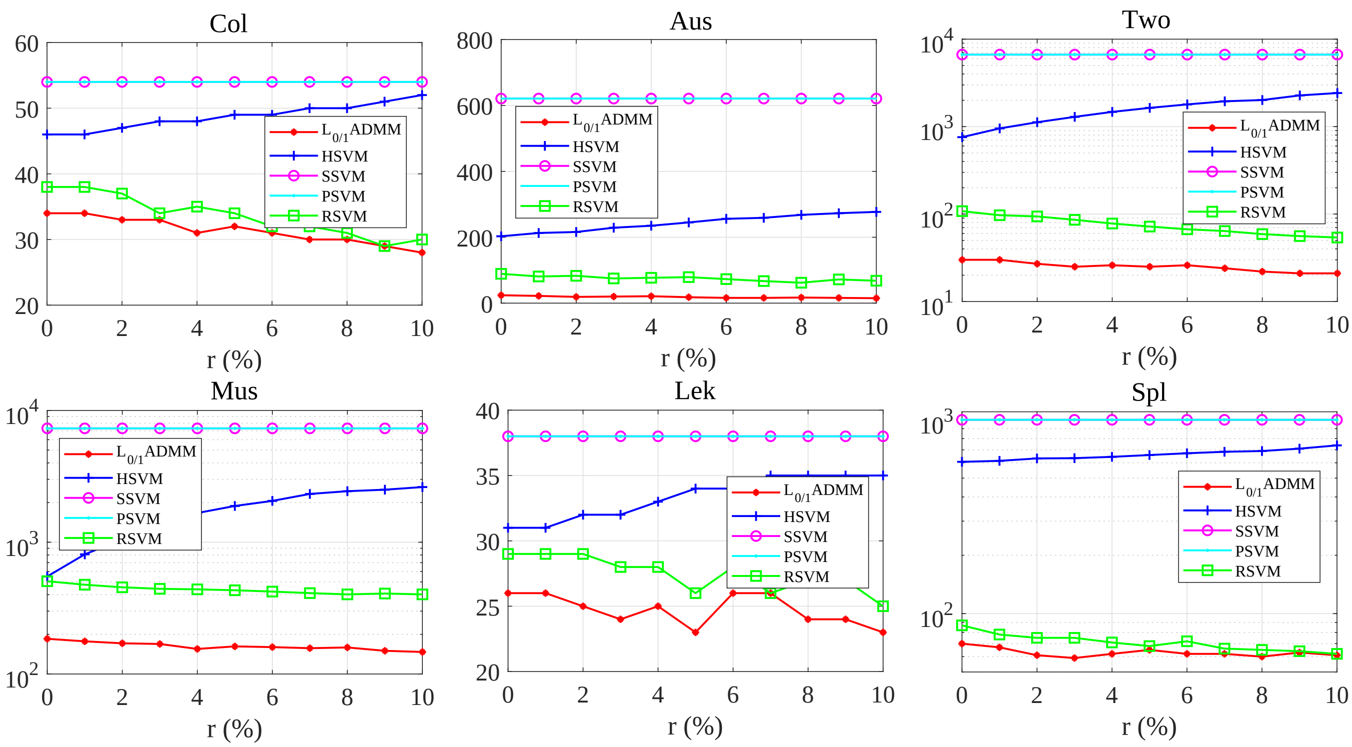

Fig. 5: NSV v.s. r of five methods for solving six data sets.

$\phi_2(s) > \phi_3(0)$, similarly, we have $u = 0$. (iii)Since $s < 0 \Leftrightarrow \phi_1(s) > \phi_2(s)$ and $\phi_3(0) > \phi_2(s)$, it is easy to see that $u = s$. (iv) Since $s = \sqrt{2\alpha} \Leftrightarrow \phi_2(s) > \phi_1(s) = \phi_3(0)$, then $u = 0$ or s. Thus, we have (7), which completes the proof. $\square$

A.3 Proof of Lemma 2.4

(i) It follows from (11) that

$$\begin{aligned} f_\gamma(\mathbf{u}, \mathbf{z}) &= C\|\mathbf{u}_+\|_0 + h(\mathbf{z}) + \langle \nabla h(\mathbf{z}), \mathbf{u} - \mathbf{z}\rangle + \frac{1}{2\gamma}\|\mathbf{u} - \mathbf{z}\|^2 \\ &= C\|\mathbf{u}_+\|_0 + h(\mathbf{z}) - \frac{1}{2\gamma}\|\nabla h(\mathbf{z})\|^2 \\ &+ \frac{1}{2\gamma}(\|\mathbf{u} - \mathbf{z}\|^2 + 2\gamma\langle \nabla h(\mathbf{z}), \mathbf{u} - \mathbf{z}\rangle + \|\nabla h(\mathbf{z})\|^2) \\ &= C\|\mathbf{u}_+\|_0 + \frac{1}{2\gamma}\|\mathbf{u} - (\mathbf{z} - \gamma\nabla h(\mathbf{z}))\|^2 \\ &+ (\text{constant term independent of } \mathbf{u}). \end{aligned}$$

Hence, the global solution of problem (11) for any fixed $\gamma, C > 0$ and $\mathbf{z} \in \mathbb{R}^m$ is equivalent to

$$\begin{aligned} \mathbf{u}^* &= \arg\min_{\mathbf{u}\in\mathbb{R}^m} C\|\mathbf{u}_+\|_0 + \frac{1}{2\gamma}\|\mathbf{u} - (\mathbf{z} - \gamma\nabla h(\mathbf{z}))\|^2 \\ &= \mathrm{prox}_{\gamma C L_{0/1}}(\mathbf{z} - \gamma\nabla h(\mathbf{z})). \end{aligned}$$

(ii) Since h is gradient Lipschitz continuous with a Lipschitz constant $\tau_h > 0$, then for any $0 < \gamma \le 1/\tau_h$, we have

$$h(\mathbf{u}) \le h(\mathbf{u}^*) + \langle \nabla h(\mathbf{u}^*), \mathbf{u} - \mathbf{u}^*\rangle + \frac{\tau_h}{2}\|\mathbf{u} - \mathbf{u}^*\|$$

from [27, Lemma 2.3]. This together with (11) yields that

$$\begin{aligned} & f_\gamma(\mathbf{u}, \mathbf{u}^*) \\ & = C\|\mathbf{u}_+\|_0 + h(\mathbf{u}^*) + \langle \nabla h(\mathbf{u}^*), \mathbf{u} - \mathbf{u}^*\rangle + \frac{1}{2\gamma}\|\mathbf{u} - \mathbf{u}^*\|^2 \\ & \ge C\|\mathbf{u}_+\|_0 + h(\mathbf{u}^*) + \langle \nabla h(\mathbf{u}^*), \mathbf{u} - \mathbf{u}^*\rangle + \frac{\tau_h}{2}\|\mathbf{u} - \mathbf{u}^*\|^2 \\ & \ge C\|\mathbf{u}_+\|_0 + h(\mathbf{u}) \ge C\|\mathbf{u}^*_+\|_0 + h(\mathbf{u}^*) = f_\gamma(\mathbf{u}^*, \mathbf{u}^*), \end{aligned}$$

the last inequality is from the global optimality of $\mathbf{u}^*$, which completes the proof. $\square$

A.4 Proof of Theorem 3.1

From (4), one can easily check that

$$\min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}} f(\mathbf{w}; b) \le f(\mathbf{1}; b) < n^2 + Cm < +\infty.$$

Next we proof the level set $S := \{(\mathbf{w}; b) \in \mathbb{R}^{n+1} : f(\mathbf{w}; b) < n^2 + Cm\}$ is non-empty and bounded. Clearly, $S \neq \emptyset$ due to $(\mathbf{1}; b) \in S$. Since b is finite-valued, we can obtain that b is bounded. Moreover, $Cm + n^2 > f(\mathbf{w}; b) \ge \|\mathbf{w}\|^2/2$, which indicates $\mathbf{w}$ is bounded. Hence, the level set S of f is non-empty and bounded and a global minimizer exists. $\square$

A.5 Proof of Theorem 3.2

(Necessity) Suppose that $\boldsymbol{\phi}^* := (\mathbf{w}^*; b^*; \mathbf{u}^*) \in \Theta$ is a local minimizer of problem (13), where Θ is the feasible set in

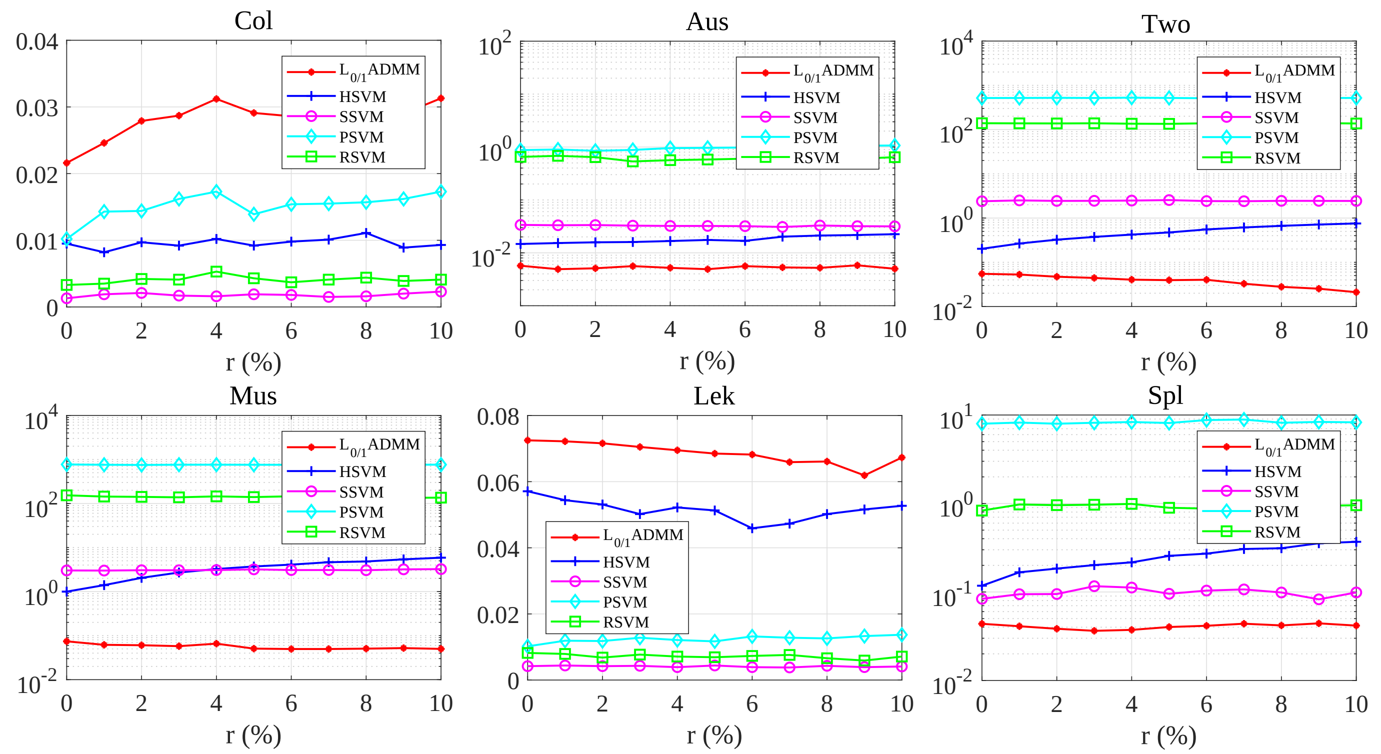

Fig. 6: CPU v.s. r of five methods for solving six data sets.

(13). Then we have the following chain of relations

$$\begin{aligned}
\mathbf{0} &\overset{(a)}{\in} \partial(\|\mathbf{w}^*\|^2/2 + \delta_\Theta(\boldsymbol{\phi}^*) + C\|\mathbf{u}^*_+\|_0) \\
&\overset{(b)}{\subseteq} \widehat{\partial}\|\mathbf{w}^*\|^2/2 + \widehat{\partial}\delta_\Theta(\boldsymbol{\phi}^*) + \widehat{\partial}C\|\mathbf{u}^*_+\|_0 \\
&\overset{(c)}{=} \partial\|\mathbf{w}^*\|^2/2 + \partial\delta_\Theta(\boldsymbol{\phi}^*) + \partial C\|\mathbf{u}^*_+\|_0 \\
&\overset{(d)}{=} (\mathbf{w}^*;0;\mathbf{0}) + N_\Theta(\boldsymbol{\phi}^*) + C(\mathbf{0};0;\partial\|\mathbf{u}^*_+\|_0) \\
&\overset{(e)}{=} (\mathbf{w}^*;0;\mathbf{0}) + \mathrm{Im}(A^\top;\mathrm{Diag}(\mathbf{y});I) + C(\mathbf{0};0;\partial\|\mathbf{u}^*_+\|_0) \\
&= (\mathbf{w}^* + A^\top\boldsymbol{\lambda}^*; \mathbf{y}^\top\boldsymbol{\lambda}^*; C\partial\|\mathbf{u}^*_+\|_0 + \boldsymbol{\lambda}^*),
\end{aligned}$$

where (a), (b), (d) and (e) hold from [17, Theorem 10.1], [24, Corollary 10.9], [23, Example 2.32] and [23, Proposition 2.12] respectively, (c) is due to the convexity of $\|\mathbf{w}^*\|^2/2$ and $\delta_\Theta(\boldsymbol{\phi}^*)$ and Lemma 2.1. Here, $\delta_\Theta(\mathbf{z})$ is the indicator function, namely, $\delta_\Theta(\mathbf{z}) = 0$ if $\mathbf{z} \in \Theta$ and $\delta_\Theta(\mathbf{z}) = +\infty$ otherwise. $N_\Theta(\mathbf{z})$ is the normal cone of the convex set Θ at point $\mathbf{z}$, which is defined as $N_\Theta(\mathbf{z}) = \{\mathbf{v} : \langle \mathbf{v}, \mathbf{z} - \mathrm{x}\rangle \geq 0, \forall\ \mathrm{x} \in \Theta\}$. $\mathrm{Im}(B)$ is the image of matrix B, i.e., $\mathrm{Im}(B) := \{B\boldsymbol{\lambda} : \boldsymbol{\lambda} \in \mathbb{R}^m\}$. I is the identity matrix. Finally, $(\mathbf{w}^*; b^*; \mathbf{u}^*) \in \Theta$ implies $\mathbf{u}^* - \mathbf{1} + A\mathbf{w}^* + b^*\mathbf{y} = \mathbf{0}$.

(Sufficiency) Suppose $\boldsymbol{\phi}^* = (\mathbf{w}^*; b^*; \mathbf{u}^*)$ and $\boldsymbol{\lambda}^* \in \mathbb{R}^m$ satisfy (15). For a given $C > 0$, (15) suffices to

$$\mathbf{0} \in \begin{bmatrix} \mathbf{w}^* + A^\top\boldsymbol{\lambda}^* \\ \langle \mathbf{y}, \boldsymbol{\lambda}^*\rangle \\ C\partial\|\mathbf{u}^*_+\|_0 + \boldsymbol{\lambda}^* \end{bmatrix}. \tag{34}$$

Denote $\mathbb{I}(\mathbf{u}^*) := \{i \in \mathbb{N}_m : u^*_i = 0\}$ and consider a problem

$$\begin{aligned}
\min_{\mathbf{w}\in\mathbb{R}^n, b\in\mathbb{R}, \mathbf{u}\in\mathbb{R}^m} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\
\text{s.t.} \quad & \mathbf{u} = \mathbf{1} - (A\mathbf{w} + b\mathbf{y}), \\
& u_i \leq 0, \;\; i \in \mathbb{I}(\mathbf{u}^*),
\end{aligned} \tag{35}$$

which is convex and thus has a global optimal solution $(\overline{\mathbf{w}}; \overline{b}; \overline{\mathbf{u}})$. It satisfies that there exists $(\overline{\boldsymbol{\lambda}}, \overline{\mu})$ such that

$$\left\{\begin{array}{rcl}
\overline{\mathbf{w}} + A^\top\overline{\boldsymbol{\lambda}} &=& \mathbf{0}, \\
\langle \mathbf{y}, \overline{\boldsymbol{\lambda}}\rangle &=& \mathbf{0}, \\
A\overline{\mathbf{w}} + \overline{b}\mathbf{y} + \overline{\mathbf{u}} - \mathbf{1} &=& \mathbf{0}, \\
\overline{\lambda}_i + \overline{\mu}_i &=& 0, \;\; i \in \mathbb{I}(\mathbf{u}^*), \\
\overline{\lambda}_i &=& 0, \;\; i \notin \mathbb{I}(\mathbf{u}^*), \\
\overline{\mu}_i \geq 0, \overline{u}_i \leq 0, \overline{\mu}_i\overline{u}_i &=& 0, \;\; i \in \mathbb{I}(\mathbf{u}^*).
\end{array}\right. \tag{36}$$

By the expression of $\partial\|\mathbf{u}^*_+\|_0$ in (5), (34) and $A\mathbf{w}^* + b^*\mathbf{y} + \mathbf{u}^* - \mathbf{1} = \mathbf{0}$ means $(\boldsymbol{\phi}^*, \boldsymbol{\lambda}^*)$ satisfy (36), which indicates $\boldsymbol{\phi}^*$ is a global solution of problem (35). Therefore, we have

$$\frac{1}{2}\|\mathbf{w}^*\|^2 \leq \frac{1}{2}\|\mathbf{w}\|^2, \;\; \forall(\mathbf{w}; b; \mathbf{u}) \in \Theta_1, \tag{37}$$

where Θ_1 is the feasible set of (35).

The function $C\|\mathbf{u}_+\|_0$ is lower semi-continuous at $\boldsymbol{\phi}^* \in \Theta$, then by [23, Proposition 4.3], there is a neighborhood $U(\boldsymbol{\phi}^*, \delta_1)$ of $\boldsymbol{\phi}^*$ with $\delta_1 > 0$ such that

$$||\mathbf{u}_+||_0 > ||\mathbf{u}^*_+||_0 - \frac{1}{2}, \forall(\mathbf{w}; b; \mathbf{u}) \in \Theta \cap U(\boldsymbol{\phi}^*, \delta_1).$$

While $||u_+||_0$ can only take values from $\{0, 1, \cdots, m\}$. It allows us to conclude that

$$||\mathbf{u}_+||_0 \geq ||\mathbf{u}^*_+||_0, \forall(\mathbf{w}; b; \mathbf{u}) \in \Theta \cap U(\boldsymbol{\phi}^*, \delta_1). \tag{38}$$

Clearly, $\Theta_1 \subseteq \Theta$. If any $(\mathbf{w}; b; \mathbf{u}) \in \Theta_1 \cap U(\boldsymbol{\phi}^*, \delta_1)$, then (37) and (38) lead to

$$\frac{1}{2}\|\mathbf{w}^*\|^2 + C\|\mathbf{u}^*_+\|_0 \leq \frac{1}{2}\|\mathbf{w}\|^2 + C\|\mathbf{u}_+\|_0. \tag{39}$$

If any $(\mathbf{w};b;\mathbf{u}) \in ((\Theta \setminus \Theta_1)) \cap U(\boldsymbol{\phi}^*, \delta_1)$, then there exists $i_0 \in \mathbb{I}(\mathbf{u}^*)$ with $u^*_{i_0} = 0$ but $u_{i_0} > 0$, which implies $\|(u^*_{i_0})_+\|_0 = 0$ but $\|(u_{i_0})_+\|_0 = 1$. By (38), we have

$$\|\mathbf{u}_+\|_0 \geq \|\mathbf{u}^*_+\|_0 + 1. \tag{40}$$

Since $\|\mathbf{w}\|^2/2$ is locally lipschitz continuous in $\mathbb{R}^n$, there exists a neighborhood $U(\boldsymbol{\phi}^*, \delta_2)$ of $\boldsymbol{\phi}^*$ with $\delta_2 > 0$ such that

$$|\|\mathbf{w}\|^2 - \|\mathbf{w}^*\|^2| \leq 2C, \ \ \forall (\mathbf{w};b;\mathbf{u}) \in U(\boldsymbol{\phi}^*, \delta_2). \tag{41}$$

Taking $\delta = \min\{\delta_1, \delta_2\}$ and combining (40) and (41), we obtain for any $(\mathbf{w};b;\mathbf{u}) \in (\Theta \setminus \Theta_1) \cap U(\boldsymbol{\phi}^*, \delta)$,

$$\begin{aligned} \frac{1}{2}\|\mathbf{w}^*\|^2 + C\|\mathbf{u}^*_+\|_0 \ &\leq \ \frac{1}{2}\|\mathbf{w}^*\|^2 + C\|\mathbf{u}_+\|_0 - C \\ &\leq \ \frac{1}{2}\|\mathbf{w}\|^2 + C\|\mathbf{u}_+\|_0. \end{aligned} \tag{42}$$

Overall, we prove the global optimality of $\boldsymbol{\phi}^*$ in a local region $\Theta \cap U(\boldsymbol{\phi}^*, \delta)$. □

A.6 Proof of Theorem 3.3

Denote $g(\mathbf{u}) := \|H(\mathbf{u} - \mathbf{1})\|^2/2$ in (17) with gradient $\nabla g(\mathbf{u}) = H^\top H(\mathbf{u} - \mathbf{1})$. From Theorem 2.1, we have

$$\mathbf{u}^* \ = \mathrm{prox}_{\gamma C L_{0/1}}(\mathbf{u}^* - \gamma \nabla g(\mathbf{u}^*)), \tag{43}$$

for any $0 < \gamma \leq \gamma_H$. Because B has a full column rank, B^+ exists, namely, $B^+ = (B^\top B)^{-1} B^\top$. Now, let $\boldsymbol{\lambda}^* = \nabla g(\mathbf{u}^*)$. Then we have

$$-\boldsymbol{\lambda}^* = H^\top H(\mathbf{u}^* - \mathbf{1}) = H^\top E B^+(\mathbf{u}^* - \mathbf{1}) = H^\top E \begin{bmatrix} \mathbf{w}^* \\ b^* \end{bmatrix},$$

where $E := \begin{bmatrix} I_{n\times n} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$, which suffices to

$$-B^\top \boldsymbol{\lambda}^* = B^\top H^\top E \begin{bmatrix} \mathbf{w}^* \\ b^* \end{bmatrix} = B^\top (B^+)^\top E^\top E \begin{bmatrix} \mathbf{w}^* \\ b^* \end{bmatrix} = \begin{bmatrix} \mathbf{w}^* \\ 0 \end{bmatrix}.$$

By the definition of $B := [A \ \mathbf{y}]$, above equation yields

$$\left\{ \begin{array}{rcl} \mathbf{w}^* + A^\top \boldsymbol{\lambda}^* & = & \mathbf{0}, \\ \langle \mathbf{y}, \boldsymbol{\lambda}^* \rangle & = & 0. \end{array} \right.$$

Finally, the above conditions, the feasibility of $(\mathbf{w}^*; b^*; \mathbf{u}^*)$ and (43) lead to (18). □

A.7 Proof of Theorem 3.4

According to (15) and (18), we only need to show that if $(\mathbf{u}^*; \boldsymbol{\lambda}^*)$ satisfies $\mathbf{u}^* = \mathrm{prox}_{\gamma C L_{0/1}}(\mathbf{u}^* - \gamma \boldsymbol{\lambda}^*)$ in (18) with $C > 0$ and $0 < \gamma \leq \gamma_H$, then $\mathbf{0} \in C\partial \|\mathbf{u}^*_+\|_0 + \boldsymbol{\lambda}^*$ in (15). In fact, it follows from (8) and (9) that for any $i \in \mathbb{N}_m$,

$$u^*_i = \begin{cases} 0, & 0 \leq v_i \leq \sqrt{2\gamma C}, \\ v_i, & \text{otherwise}, \end{cases}$$

where $\mathbf{v} := \mathbf{u}^* - \gamma \boldsymbol{\lambda}^*$. This means for any i with $u^*_i = 0$, $-\sqrt{2\gamma C}/\gamma \leq \lambda^*_i \leq 0$ and for any i with $u^*_i = v_i = u^*_i - \gamma \lambda^*_i$, $\lambda^*_i = 0$. Finally, the expression of $\partial \|\mathbf{u}^*_+\|_0$ in Lemma 2.1 allows us to complete the proof immediately. □

A.8 Proof of Theorem 4.1

Since $T_k \subseteq \mathbb{N}_m$ has finite many elements, for sufficient large k, there is a subset $J \subseteq \{1, 2, 3, \cdots\}$ such that

$$T_j \equiv: T, \ \ \forall\, j \in J. \tag{44}$$

For notational simplicity, denote $\boldsymbol{\phi}^k := (\mathbf{w}^k, b^k, \mathbf{u}^k, \boldsymbol{\lambda}^k)$ and $\boldsymbol{\phi}^* := (\mathbf{w}^*, b^*, \mathbf{u}^*, \boldsymbol{\lambda}^*)$. As $\{\boldsymbol{\phi}^k\} \to \boldsymbol{\phi}^*$, it follows $\{\boldsymbol{\phi}^j\}_{j\in J} \to \boldsymbol{\phi}^*$ and $\{\boldsymbol{\phi}^{j+1}\}_{j\in J} \to \boldsymbol{\phi}^*$. Taking the limit along with J of (33), namely, $k \in J, k \to \infty$, we have

$$\left\{ \begin{array}{rcl} \boldsymbol{\lambda}^*_T & = & \boldsymbol{\lambda}^*_T + \eta\sigma \boldsymbol{\varpi}^*_T, \\ \boldsymbol{\lambda}^*_{\overline{T}} & = & \mathbf{0}, \end{array} \right. \tag{45}$$

which derives $\boldsymbol{\varpi}^*_T = \mathbf{0}$. Taking the limit along with J of (23) and (24) respectively yields

$$\begin{aligned} \mathbf{z}^* \ &= \ \mathbf{1} - A\mathbf{w}^* - b^*\mathbf{y} - \boldsymbol{\lambda}^*/\sigma \\ &= \ \mathbf{1} - A\mathbf{w}^* - b^*\mathbf{y} - \mathbf{u}^* + \mathbf{u}^* - \boldsymbol{\lambda}^*/\sigma \\ &= \ -\boldsymbol{\varpi}^* + \mathbf{u}^* - \boldsymbol{\lambda}^*/\sigma \end{aligned} \tag{46}$$

and thus

$$\begin{aligned} \mathbf{u}^*_T = \mathbf{0}, \ \ \mathbf{u}^*_{\overline{T}} = \mathbf{z}^*_{\overline{T}} \ &\overset{(46)}{=} \ -\boldsymbol{\varpi}^*_{\overline{T}} + \mathbf{u}^*_{\overline{T}} - \boldsymbol{\lambda}^*_{\overline{T}}/\sigma \\ &\overset{(45)}{=} \ -\boldsymbol{\varpi}^*_{\overline{T}} + \mathbf{u}^*_{\overline{T}}. \end{aligned} \tag{47}$$

This proves $\boldsymbol{\varpi}^*_{\overline{T}} = \mathbf{0}$ and hence $\boldsymbol{\varpi}^* = \mathbf{0}$. Again by (46), we obtain $\mathbf{z}^* = \mathbf{u}^* - \boldsymbol{\lambda}^*/\sigma$, which together with (47) and the definition of proximal operator (9) indicates

$$\mathbf{u}^* = \mathrm{Prox}_{\frac{C}{\sigma} L_{0/1}}(\mathbf{z}^*) = \mathrm{Prox}_{\frac{C}{\sigma} L_{0/1}}(\mathbf{u}^* - \boldsymbol{\lambda}^*/\sigma). \tag{48}$$

Now taking the limit along with J of (28) results in

$$\begin{aligned} (I + \sigma A_T^\top A_T)\mathbf{w}^* \ &= \ \sigma A_T^\top \mathbf{v}^*_T \\ &= \ -\sigma A_T^\top (\mathbf{u}^*_T + b^*\mathbf{y}_T - \mathbf{1} + \boldsymbol{\lambda}^*_T/\sigma) \\ &= \ -\sigma A_T^\top (\boldsymbol{\varpi}^*_T - A_T\mathbf{w}^* + \boldsymbol{\lambda}^*_T/\sigma) \\ &= \ -\sigma A_T^\top (-A_T\mathbf{w}^* + \boldsymbol{\lambda}^*_T/\sigma), \end{aligned}$$

where $\mathbf{v}^* = -(\mathbf{u}^* + b^*\mathbf{y} - \mathbf{1} + \boldsymbol{\lambda}^*/\sigma)$ and the last two equations hold due to $\boldsymbol{\varpi}^* = \mathbf{u}^* + A\mathbf{w}^* + b^*\mathbf{y} - \mathbf{1} = \mathbf{0}$. The last equation suffices to that

$$\mathbf{w}^* \ = \ -A_T^\top \boldsymbol{\lambda}^*_T \overset{(45)}{=} -A^\top \boldsymbol{\lambda}^*.$$

Finally taking the limit along with J of (32) leads to

$$\begin{aligned} b^* = \langle \mathbf{y}, \mathbf{r}^* \rangle/m \ &= \ -\langle \mathbf{y}, A\mathbf{w}^* - \mathbf{1} + \mathbf{u}^* + \boldsymbol{\lambda}^*/\sigma \rangle/m \\ &= \ -\langle \mathbf{y}, \boldsymbol{\varpi}^* - b^*\mathbf{y} + \boldsymbol{\lambda}^*/\sigma \rangle/m \\ &= \ -\langle \mathbf{y}, -b^*\mathbf{y} + \boldsymbol{\lambda}^*/\sigma \rangle/m \\ &= \ b^* - \langle \mathbf{y}, \boldsymbol{\lambda}^* \rangle/(m\sigma), \end{aligned}$$

which contributes to $\langle \mathbf{y}, \boldsymbol{\lambda}^* \rangle = 0$. Overall, we have

$$\left\{ \begin{array}{rcl} \mathbf{w}^* + A^T \boldsymbol{\lambda}^* & = & \mathbf{0}, \\ \langle \mathbf{y}, \boldsymbol{\lambda}^* \rangle & = & \mathbf{0}, \\ \mathbf{u}^* + A\mathbf{w}^* + b^*\mathbf{y} & = & \mathbf{1}, \\ \mathrm{prox}_{\frac{C}{\sigma} L_{0/1}}(\mathbf{u}^* - \boldsymbol{\lambda}^*/\sigma) & = & \mathbf{u}^*. \end{array} \right.$$

Namely, $(\mathbf{w}^*; b^*; \mathbf{u}^*)$ is a P-stationary point of problem (13) where $\gamma = 1/\sigma$. □

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (11971052), the National Natural Science Foundation of China (61866010, 11871183), and the Natural Science Foundation of Hainan Province (118QN181).

REFERENCES

[1] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning,* vol. 20, no. 3, pp. 273-297, 1995.

[2] V. Jumutc, X. Huang, and A. K. Suykens, "Fixed-size pegasos for hinge and pinball loss SVM," *International Joint Conference on Neural Networks*, 2013.

[3] X. Huang, L. Shi, and A. K. Suykens, "Support vector machine classifier with pinball loss," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 36, no. 5, pp. 984-997, 2014.

[4] S. Rosset and J. Zhu, "Piecewise linear regularized solution paths," *The Annals of Statistics,* vol. 35, no. 3, pp. 1012-1030, 2007.

[5] L. Wang, J. Zhu, and H. Zou, "Hybrid huberized support vector machines for microarray classification," *Bioinformatics,* vol. 24, no. 3, pp. 412-419, 2008.

[6] Y. Xu, I. Akrotirianakis, and A. Chakraborty, "Proximal gradient method for huberized support vector machine," *Pattern Analysis and Applications,* vol. 19, no. 4, pp. 989-1005, 2016.

[7] A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters,* vol. 9, no. 3, pp. 293-300, 1999.

[8] X. Yang, L. Tan, and L. F. He, "A robust least squares support vector machine for regression and classification with noise," *Neurocomputing,* vol. 140, pp. 41-52, 2014.

[9] Y. Freund and R. E. Schapire, "A decision theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.

[10] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Annals of statistics,* vol. 28, no. 2, pp. 337-374, 2000.

[11] L. Mason, P. L. Bartlett, and J. Baxter, "Improved generalization through explicit optimization of margins," *Machine Learning,* vol. 38, no. 3, pp. 243-255, 2000.

[12] F. Perez-Cruz, A. Navia-Vazquez, A. R. Figueiras-Vidal, and A. Artes-Rodriguez, "Empirical risk minimization for support vector classifiers," *IEEE Transactions on Neural Networks,* vol. 14, no. 2, pp. 296-303, 2003.

[13] X. Huang, L. Shi, and J. A. K. Suykens, "Ramp loss linear programming support vector machine," *Journal of Machine Learning Research,* vol. 15, no. 1, pp. 2185-2211, 2014.

[14] X. Shen, L. F. Niu, Z. Qi, and Y. J. Tian, "Support vector machine classifier with truncated pinball loss," *Pattern Recognition,* vol. 68, pp. 199-210, 2017.

[15] L. M. Yang and H. G. Dong, "Support vector machine with truncated pinball loss and its application in pattern recognition," *Chemometrics and Intelligent Laboratory Systems,* vol. 177, pp. 89-99, 2018.

[16] F. Perez-Cruz, A. Navia-Vazquez, P. L. Alarcon-Diana, and A. Artes-Rodriguez, "Support vector classifier with hyperbolic tangent penalty function," *International Conference on Acoustics, Speech, and Signal Processing,* 2000.

[17] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," *International Conference on Neural Information Processing Systems,* 1999.

[18] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Large margin classifiers: convex loss, low noise, and convergence rates," *International Conference on Neural Information Processing Systems*, 2004.

[19] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association,* vol. 101, no. 473, pp. 138-156, 2006.

[20] J. H. Friedman, "On bias, variance, 0/1-loss, and the curse-of-dimensionality," *Data Mining and Knowledge Discovery,* vol. 1, no. 1, pp. 55-77, 1997.

[21] J. P. Brooks, "Support vector machines with the ramp loss and the hard margin loss," *Operations Research,* vol. 59, no. 2, pp. 467-479, 2011.

[22] L. Li and H. T. Lin, "Optimizing 0/1 loss for perceptrons by random coordinate descent," *International Joint Conference on Neural Networks,* 2007.

[23] B. Mordukhovich and N. Nam, "An easy path to convex analysis and applications," *Morgan and Claypool Pubulishers,* 2014.

[24] R. T. Rockafellar and R. J. B Wets, "Variational analysis,"*Springer Science and Business Media,* 1998.

[25] H. H. Bauschke and P. L. Combettes, "Convex analysis and monotone operator theory in hilbert space,"*New York: Springer,* 2011.

[26] Y. Q. Chen, N. H. Xiu, and D. T. Peng, "Global solutions of non-Lipschitz S_2-S_p minimization over the positive semidefinite cone," *Optimization Letters,* vol. 8, no. 7, pp. 2053-2064, 2014.

[27] A. Beck and Y. C. Eldar, "Sparsity constrained nonlinear optimization: optimality conditions and algorithms," *SIAM Journal on Optimization,* vol. 23, no. 3, pp. 1480-1509, 2013.

[28] I. Steinwart and N. Christianini, "Sparseness of support vector machines," *Journal of Machine Learning Research,* vol. 4, no. 6, pp. 1071-1105, 2004.

[29] S. Ertekin, L. Bottou, and C. L. Giles, "Nonconvex online support vector machines," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. 4, pp. 368-381, 2010.

[30] M. Fazel, T. K. Pong, D. F. Sun, and P. Tseng, "Hankel matrix rank minimization with applications to system identification and realization," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 946-977, 2013.

[31] M. Li, D. F. Sun and K. C. Toh, "A majorized ADMM with indefinite proximal terms for linearly constrained convex composite optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 922-950, 2016.

[32] B. S. He, F. Ma, and X. M. Yuan, "Optimal linearized alternating direction method of multipliers for convex programming," *Available on http://www.optimization-online.org/DB_FILE/2017/09/6228.pdf*, 2017.

[33] X. Chang, S. Liu, P. Zhao, and D. Song, "A generalization of linearized alternating direction method of multipliers for solving two-block separable convex programming," *Journal of Computational and Applied Mathematics*, vol. 357, no. 2, pp. 251-272, 2019.

[34] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology,* vol. 2, no. 3, pp. 27, 2011.

[35] K. Pelckmans, J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, L. Lukas, B. Hamers, B. D. Moor, and J. Vandewalle, "SSVM lab: a matlab/c toolbox for least squares support vector machines," *Tutorial. KULeuven-ESAT. Leuven, Belgium,* vol. 142, pp. 1-2, 2002.

[36] X. Huang, L. Shi, and J. A. K. Suykens, "Solution path for pin-SVM classifiers with positive and negative τ values," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 28, no. 7, pp. 1584-1593, 2016.

[37] Y. Wu and Y. Liu, "Robust truncated hinge loss support vector machines," *Journal of the American Statistical Association,* vol. 102, no. 479, pp. 974-983, 2006.

[38] R. Khemchandani and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 29, no. 5, pp. 905-910, 2007.

[39] Y. Xu, Z. Yang, and X. Pan, "A novel twin support vector machine with pinball loss," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 28, no. 2, pp. 359-370, 2017.

[40] B. Hong, W. Z. Zhang, W. Liu, J. P. Ye, D. Cai, X. f. He, and J. Wang, "Scaling up sparse support vector machines by simultaneous feature and sample reduction," *Journal of Machine Learning Research,* vol. 20, no. 121, pp. 1-39, 2019.

[41] C. N. Li, Y. H. Shao, H. J. Wang, Y. T. Zhao, L. W. Huang, N. H. Xiu, and N. Y. Deng, "Single Versus Union: Non-parallel Support Vector Machine Frameworks," *arXiv preprint arXiv:1910.09734,* 2019.