A sparse semismooth Newton based augmented Lagrangian method for large-scale support vector machines

Dunbiao Niu* Chengjing Wang† Peipei Tang‡ Qingsong Wang§ and Enbin Song¶

September 30, 2019

Abstract

Support vector machines (SVMs) are successful modeling and prediction tools with a variety of applications. Previous work has demonstrated the superiority of the SVMs in dealing with the high dimensional, low sample size problems. However, the numerical difficulties of the SVMs will become severe with the increase of the sample size. Although there exist many solvers for the SVMs, only few of them are designed by exploiting the special structures of the SVMs. In this paper, we propose a highly efficient sparse semismooth Newton based augmented Lagrangian method for solving a large-scale convex quadratic programming problem with a linear equality constraint and a simple box constraint, which is generated from the dual problems of the SVMs. By leveraging the primal-dual error bound result, the fast local convergence rate of the augmented Lagrangian method can be guaranteed. Furthermore, by exploiting the second-order sparsity of the problem when using the semismooth Newton method, the algorithm can efficiently solve the aforementioned difficult problems. Finally, numerical comparisons demonstrate that the proposed algorithm outperforms the current state-of-the-art solvers for the large-scale SVMs.

Keywords: support vector machines, semismooth Newton method, augmented Lagrangian method

*College of Mathematics, Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu 610065, China. (dunbiaoniu_sc@163.com).

†School of Mathematics, Southwest Jiaotong University, No.999, Xian Road, West Park, High-tech Zone, Chengdu 611756, China. (renascencewang@hotmail.com).

‡School of Computing Science, Zhejiang University City College, Hangzhou 310015, China. (tangpp@zucc.edu.cn). The research work of Peipei Tang is supported by the Natural Science Foundation of Zhejiang Province of China under Grant LY19A010028 and the Science & Technology Development Project of Hangzhou, China under Grant 20170533B22.

§School of Mathematics and System Sciences, Beihang University, No.37 Xueyuan Road, Haidian District, Beijing 100191, China; School of Mathematics, Southwest Jiaotong University, No.999, Xian Road, West Park, High-tech Zone, Chengdu 611756, China. (nothing2wang@hotmail.com).

¶College of Mathematics, Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu 610065, China. (e.b.song@163.com). The research work of Enbin Song is supported by the Sichuan Science and Technology Program under Grant 2019YJ0115 and the National Natural Science Foundation of China under Grants 61473197 and 11871357.

1 Introduction

Support Vector Machines (SVMs), introduced by [1], are originally formulated for binary classification problems that aim to separate two data sets with the widest margin. Nowadays, the SVMs have been extended to solve a variety of pattern recognition and data mining problems such as feature selection [2], text categorization [3], hand-written character recognition [4], image classification [5], and so on. Among the different applications of the SVMs, we first introduce three specific examples.

- The C-Support Vector Classification (C-SVC) [6, 7]:

  Given a training set of instance-label pairs $(\tilde{\mathbf{x}}_i, y_i)$, $i = 1, ..., n$, where $\tilde{\mathbf{x}}_i \in \mathcal{R}^p$ and $y_i \in \{+1, -1\}$, the C-SVC aims to find a hyperplane in a given reproducing kernel Hilbert space $\mathcal{H}$ [8] to separate the data set into two classes with the widest margin. In this model, we need to solve the following optimization problem

$$\begin{aligned} \min_{w,b,\xi} \ & \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle_{\mathcal{H}} + C\sum_{i=1}^{n}\xi_i \\ s.t. \ & y_i\left(\langle \mathbf{w}, \phi(\tilde{\mathbf{x}}_i)\rangle_{\mathcal{H}} + b\right) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \ i = 1, \ldots, n, \end{aligned} \tag{1}$$

  where $C > 0$ is a regularization parameter, $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product in a reproducing kernel Hilbert space $\mathcal{H}$ and $\phi : \mathcal{R}^p \to \mathcal{H}$ is a feature map such that the function $K(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) := \langle \phi(\tilde{\mathbf{x}}_i), \phi(\tilde{\mathbf{x}}_j)\rangle_{\mathcal{H}}$ is a reproducing kernel of $\mathcal{H}$ for any data $\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j \in \mathcal{R}^p$. For example, $K(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) = \tilde{\mathbf{x}}_i^T\tilde{\mathbf{x}}_j$ is a linear kernel and $K(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) = \exp^{-\|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|^2/2\alpha}$ is a radial basis function (RBF) kernel, where $\alpha > 0$ is a fixed parameter called the width. As it has been shown in [6], the dual of the problem (1) is the following quadratic programming problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{R}^n} \ & \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \mathbf{e}^T\mathbf{x} \\ s.t. \ & \mathbf{y}^T\mathbf{x} = 0, \\ & 0 \leq x_i \leq C, \ i = 1, \ldots, n, \end{aligned} \tag{2}$$

  where $\mathbf{e} = [1, \ldots, 1]^T \in \mathcal{R}^n$ is a vector of all ones, $\mathbf{y} = [y_1, \ldots, y_n]^T \in \mathcal{R}^n$ is a label vector and $Q \in \mathcal{S}^n$ (the space of $n \times n$ symmetric matrices) is a positive semidefinite matrix with $Q_{ij} = y_i y_j K(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$, $i, j = 1, \ldots, n$.

- The ϵ-Support Vector Regression (ϵ-SVR) [9]:

  For a set of training points $(\tilde{\mathbf{x}}_i, y_i)$, $i = 1, ..., n$, the regression problem is to predict a continuous output $y_i \in \mathcal{R}$ given an input $\tilde{\mathbf{x}}_i \in \mathcal{R}^n$. For given parameters $C > 0$ and $\epsilon > 0$, the standard form of support vector regression is

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\xi}^*} \ & \frac{1}{2}\langle \mathbf{w}, \mathbf{w}\rangle_{\mathcal{H}} + C\sum_{i=1}^{n}\xi_i + C\sum_{i=1}^{n}\xi_i^* \\ s.t. \ & \langle \mathbf{w}, \phi(\tilde{\mathbf{x}}_i)\rangle_{\mathcal{H}} + b - y_i \geq \epsilon + \xi_i, \\ & y_i - \langle \mathbf{w}, \phi(\tilde{\mathbf{x}}_i)\rangle_{\mathcal{H}} - b \geq \epsilon - \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, \ i = 1, \ldots, n. \end{aligned} \tag{3}$$

Furthermore, the dual of the problem (3) is

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}\in\mathcal{R}^n} \ & \frac{1}{2}[\mathbf{x}^T, \mathbf{z}^T] \begin{pmatrix} Q & -Q \\ -Q & Q \end{pmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} + \sum_{i=1}^{n} (\varepsilon + y_i)x_i + \sum_{i=1}^{n} (\varepsilon - y_i)z_i \\ s.t. \ & [\mathbf{e}^T, -\mathbf{e}^T] \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} = 0, \\ & 0 \le x_i, z_i \le C, \ i = 1, \ldots, n. \end{aligned} \tag{4}$$

- The one-class SVM [10]:

  The one-class SVM is an unsupervised machine learning model that learns a decision function for novelty detection, which is used to detect whether new data is similar to the elements of the training set. The corresponding optimization problem takes the following form

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\xi},\rho} \ & \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle_{\mathcal{H}} - \rho + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i \\ s.t. \ & \langle \mathbf{w}, \phi(\tilde{\mathbf{x}}_i) \rangle_{\mathcal{H}} \ge \rho - \xi_i, \\ & \xi_i \ge 0, \ i = 1, \ldots, n, \end{aligned} \tag{5}$$

  where $\nu \in (0, 1]$ is a parameter. The dual of the problem (5) is

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{R}^n} \ & \frac{1}{2}\mathbf{x}^T Q \mathbf{x} \\ s.t. \ & \mathbf{e}^T \mathbf{x} = 1, \\ & 0 \le x_i \le \frac{1}{\nu n}, \ i = 1, \ldots, n. \end{aligned} \tag{6}$$

The above three problems (1), (3) and (5) show that there are various formulations for the SVMs in different scenarios. However, their dual problems (2), (4) and (6) can be summarized as the following unified form

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{R}^n} \ & \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ s.t. \ & \mathbf{a}^T \mathbf{x} = d, \\ & \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \end{aligned} \tag{7}$$

where $Q \in \mathcal{S}^n$ is a positive semidefinite matrix and $\mathbf{c}, \mathbf{a}, \mathbf{l}, \mathbf{u} \in \mathcal{R}^n$ and $d \in \mathcal{R}$ are given vectors and scalar, respectively. Moreover, we assume that $\mathbf{l} < \mathbf{u}$, i.e., $l_i < u_i$ for all $i \in \{1, \ldots, n\}$, where l_i and u_i are the ith elements of $\mathbf{l}$ and $\mathbf{u}$. Currently most of the work on computational aspects of the SVMs concentrate on solving the dual problem (7) since it is a more general framework to handle the SVMs. But there still exist some algorithms that solve the primal problem. For example, Yin and Li [11] proposed a semismooth Newton method to solve the primal problems of L2-loss SVC model and the ϵ-L2-loss SVR model with linear kernel recently.

For the convex problem (7), many existing optimization algorithms including the Newton method, the accelerated proximal gradient (APG) method, the alternating direction method of multipliers (ADMM) and the interior-point method (IPM) et al., can be applied to solve it efficiently

when the problem scale is small or moderate. However, when facing the large-scale problems, the numerical difficulties become severe. Specifically, when the dimension n is very large, the full storage of the $n \times n$ dense matrix Q in (7) is very difficult and even impossible for a typical computer. Therefore we can not apply the standard quadratic programming solvers which require the full storage of Q directly. An alternative approach is to compute the elements of Q from the original data when it is required. However, this becomes prohibitively time consuming since the elements of Q are required at each iteration.

Currently, one approach to avoid using the whole elements of the matrix Q is to adopt the decomposition strategy [12–14]. In this kind of approach, only a small subset of variables in each iteration are needed to be updated so that only a few rows of the matrix Q are involved in, which significantly reduces the computational cost in each iteration. The Sequential Minimal Optimization (SMO) method [14] is one of the well-known decomposition methods, in which only two variables are considered in each iteration. The popular SVMs solver LIBSVM [15] also implements an SMO-type method [16] to solve the problem (7).

Another widely used algorithm for the problem (7) is the gradient projection (GP) method. Thanks to the low-cost algorithm of the projection onto the feasible set of (7) and the identification properties of the GP method [17], we only need to consider a reduced subproblem which is related to the current active set of variables. Furthermore, inspired by the algorithm for the bound constrained quadratic programming problem [18], the proportionality-based 2-phase gradient projection (P2GP) method [19] is derived to solve the problem (7). In addition, the fast APG (FAPG) method [20] is another commonly used algorithm to solve various classification models, including the C-SVC, L2-loss SVC, and v-SVM et al..

The advantage of the decomposition and PG methods is that only a small subset of variables are involved in each subproblem, i.e., only a small part of the elements of the matrix Q are required in each iteration. However, as the numerical experiments show in Section 4, both of these types of algorithms may encounter the problem of slow convergence. If Q is positive definite and the nondegeneracy assumption holds, the SMO-type decomposition methods are shown to be linear convergent to the solution of (7) in [21]. It is well-known that the PG-type algorithms can exhibit (linear) sublinear convergence when the objective function is (strongly) convex. These existing theories partly explain why the convergences of these algorithms are not ideal.

In this paper, we aim to solve the problem (7) by applying the augmented Lagrangian (AL) method [22] to the dual problem of (7). Meanwhile, a semismooth Newton (SsN) method is used to solve the inner subproblems of the AL method. It is well known that to fully fulfill the potential of the AL method, the inner subproblems should be solved accurately and efficiently. Although the SsN method is an ideal approach to solve the inner subproblems, it can not be applied directly because the costs of the SsN method may be very high at the beginning few iterations due to the lack of a sparse structure of the generalized Hessian when the scale of the problem is large. To overcome this difficulty, we may use the gradient method to produce a good initial point, and then transfer to the SsN method. The generalized Hessian for the inner subproblem at the initial point generated by the gradient method may probably has some kind of sparsity structure. Wisely exploiting this nice structure may largely reduce the computational cost and the memory consumption in each SsN step. Hence, our proposed algorithm not only has the same advantage as the decomposition method in [13] with memory requirements being linear as the number of training examples and support vectors, but also has the fast local convergent rate in both inner and outer iterations. Since our algorithm fully takes advantage of the sparsity structure, we call it a Sparse SsN based

AL (SSsNAL) method. Besides, there are three main reasons why the SSsNAL method can be implemented efficiently to solve the problem (7):

(I) The piecewise linear-quadratic structure of the problem (7) guarantees the fast local convergence of the AL method [23, 24].

(II) There exist many efficient algorithms [20, 25–28] to compute the value of the Euclidean projection of any given point onto the feasible set (7) due to its special structure. Furthermore, the explicit formula of the generalized Jacobian, which is named HS-Jacobian [29], can be easily derived.

(III) It is generally true for many SVMs that the number of the support vectors are much less than that of the training examples, and many multiplier variables of the support vectors are at the upper bound of the box constraint [13]. That is, most of the elements of the optimal solution lie in the interior of the box constraint.

As will be shown later, the above (I) and (II) guarantee the inner subproblem can be solved by the SsN method with a very low memory consumption in each iteration. And the above (I), (II) and (III) together provide an insight into the compelling advantages of applying the SSsNAL method to the problem (7). Indeed, for the large-scale problem (7), the numerical experiments in Section 4 will show that the SSsNAL method only needs at most a few dozens of outer iterations to reach the desired solutions while all the inner subproblems can be solved without too much effort.

The rest of the paper is organized as follows. In Section 2, some preliminaries about the restricted dual formulation and some necessary error bound results are provided. Section 3 is dedicated to present the SSsNAL method for the restricted dual problem in details. Numerical experiments are presented on real data in Section 4, which verify the performance of our SSsNAL method against other solvers. Finally, Section 5 concludes this paper.

Notations: Let $\mathcal{X}$ and $\mathcal{Y}$ be two real finite dimensional Euclidean spaces. For any convex function $p : S \subset \mathcal{X} \to (-\infty, \infty]$, its conjugate function is denoted by p^*, i.e., $p^*(x) = \sup_y\{\langle x, y\rangle - p(y)\}$, and its subdifferential at x is denoted by $\partial p(x)$, i.e., $\partial p(x) := \{y \,|\, p(z) \geq p(x) + \langle y, z - x\rangle, \forall\, z \in \mathrm{dom}(p)\}$. For a given closed convex set Ω and a vector x, we denote the distance from x to Ω by $\mathrm{dist}(x, \Omega) := \inf_{y\in\Omega} \|x - y\|$ and the Euclidean projection of x onto Ω by $\Pi_\Omega(x) := \mathrm{argmin}_{y\in\Omega} \|x - y\|$. For any set-valued mapping $F : \mathcal{X} \rightrightarrows \mathcal{Y}$, we use gph F to denote the graph of F, i.e., gph $F := \{(x, y) \in \mathcal{X} \times \mathcal{Y} \mid y \in F(x)\}$. We use I_n to denote the $n \times n$ identity matrix in $\mathcal{R}^n$ and $A^\dagger$ to denote the Moore-Penrose pseudo-inverse of a given matrix $A \in \mathcal{R}^{n\times n}$.

2 Preliminaries

In this section, we present some necessary error bound results, which will be used in the convergence rate analysis of the AL method in Section 3.1.

We denote the single linear constraint and the box constraint in the problem (7) by

$$L := \{\mathbf{x} \in \mathcal{R}^n \mid \mathbf{a}^T\mathbf{x} = d\} \text{ and } K := \{\mathbf{x} \in \mathcal{R}^n \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}, \tag{8}$$

respectively. Then the problem (7) can be equivalently rewritten as

$$(\mathbf{P}) \quad \min_{\mathbf{x}\in\mathcal{R}^n} \left\{ f(\mathbf{x}) := \tfrac{1}{2}\langle \mathbf{x}, Q\mathbf{x}\rangle + \langle \mathbf{c}, \mathbf{x}\rangle + \delta_{K\cap L}(\mathbf{x}) \right\},$$

where $\delta_{K\cap L}$ is the indicator function for the polyhedral convex set $K \cap L$, i.e., $\delta_{K\cap L}(\mathbf{x}) = 0$ if $\mathbf{x} \in K\cap L$, otherwise $\delta_{K\cap L}(\mathbf{x}) = +\infty$. Note that the problem (**P**) is already the dual formulation of the SVMs in the introduction, but we still regard it as the primal problem according to our custom. The dual of the problem (**P**) is

$$(\mathbf{D}) \quad \min_{\mathbf{w}\in\mathcal{R}^n,\ \mathbf{z}\in\mathcal{R}^n} \left\{ \tfrac{1}{2}\langle \mathbf{w}, Q\mathbf{w}\rangle + \delta^*_{K\cap L}(\mathbf{z}) \,|\, Q\mathbf{w} + \mathbf{z} + \mathbf{c} = 0,\ \mathbf{w} \in \mathrm{Ran}(Q) \right\},$$

where $\delta^*_{K\cap L}$ is the conjugate of the indicator function $\delta_{K\cap L}$, and $\mathrm{Ran}(Q)$ denotes the range space of Q. Note that the additional constraint $\mathbf{w} \in \mathrm{Ran}(Q)$ is reasonable because, for any $\mathbf{w}^0 \in \mathrm{Ran}^{\perp}(Q)$, $\mathbf{w}$ and $\mathbf{w} + \mathbf{w}^0$ have the same objective function values and both satisfy the linear constraint in (**D**). As will be shown in the next section, the constraint $\mathbf{w} \in \mathrm{Ran}(Q)$ in fact plays an important role to guarantee that the subproblem has a unique solution and our designed algorithm is efficient. Since we restrict $\mathbf{w}$ in the range space artificially, we may also call (**D**) a restricted dual problem. Correspondingly, the Karush-Kuhn-Tucker (KKT) condition associated with the problem (**P**) is given by

$$\mathbf{x} - \mathrm{Prox}_{\delta_{K\cap L}}(\mathbf{x} + \mathbf{z}) = 0,\ Q\mathbf{w} - Q\mathbf{x} = 0,\ Q\mathbf{w} + \mathbf{z} + \mathbf{c} = 0, \tag{9}$$

where the proximal mapping for a given closed proper convex function $p : \mathcal{R}^n \to (-\infty, +\infty]$ is defined by

$$\mathrm{Prox}_p(\mathbf{u}) := \operatorname*{argmin}_{\mathbf{x}} \left\{ p(\mathbf{x}) + \frac{1}{2}\|\mathbf{u} - \mathbf{x}\|^2 \right\}, \forall\, \mathbf{u} \in \mathcal{R}^n.$$

Moreover, for any given parameter $\lambda > 0$, we introduce the following Moreau identity, which will be used frequently.

$$\mathrm{Prox}_{\lambda p}(\mathbf{u}) + \lambda \mathrm{Prox}_{p^*/\lambda}(\mathbf{u}/\lambda) = \mathbf{u}. \tag{10}$$

Let l be the ordinary Lagrangian function for the problem (**D**)

$$l(\mathbf{w}, \mathbf{z}, \mathbf{x}) = \begin{cases} \frac{1}{2}\langle \mathbf{w}, Q\mathbf{w}\rangle + \delta^*_{K\cap L}(\mathbf{z}) - \langle \mathbf{x}, Q\mathbf{w} + \mathbf{z} + \mathbf{c}\rangle, & \mathbf{w} \in \mathrm{Ran}(Q), \\ +\infty, & \text{otherwise.} \end{cases} \tag{11}$$

Then, we define the maximal monotone operators $\mathcal{T}_f$ and $\mathcal{T}_l$ [22] by

$$\mathcal{T}_f(\mathbf{x}) := \partial f(\mathbf{x}) = \{\mathbf{u}_x \in \mathcal{R}^n |\ \mathbf{u}_x \in Q\mathbf{x} + \mathbf{c} + \partial\delta_{K\cap L}(\mathbf{x})\},\ \forall\, \mathbf{x} \in \mathcal{R}^n$$

and

$$\mathcal{T}_l(\mathbf{w}, \mathbf{z}, \mathbf{x}) := \{(\mathbf{u}_w, \mathbf{u}_z, \mathbf{u}_x) \in \mathcal{R}^{3n} \mid (\mathbf{u}_w, \mathbf{u}_z, -\mathbf{u}_x) \in \partial l(\mathbf{w}, \mathbf{z}, \mathbf{x})\},$$

respectively. Correspondingly, the inverses of $\mathcal{T}_f$ and $\mathcal{T}_l$ are

$$\mathcal{T}_f^{-1}(\mathbf{u}_x) := \{\mathbf{x} \in \mathcal{R}^n \mid \mathbf{u}_x \in \partial f(\mathbf{x})\},$$

and

$$\mathcal{T}_l^{-1}(\mathbf{u}_w, \mathbf{u}_z, \mathbf{u}_x) := \{(\mathbf{w}, \mathbf{z}, \mathbf{x}) \in \mathcal{R}^{3n} \mid (\mathbf{u}_w, \mathbf{u}_z, \mathbf{u}_x) \in \partial l(\mathbf{w}, \mathbf{z}, \mathbf{x})\},$$

,respectively. Recall that a closed proper convex function $g : \mathcal{X} \to (-\infty, +\infty]$ is said to be *piecewise linear-quadratic* if dom g is the union of finitely many polyhedral sets and on each of these polyhedral sets, g is either an affine or a quadratic function [30, Definition 10.20]. Hence the objective function f in (**P**) is piecewise linear-quadratic. Meanwhile, by [30, Theorem 11.14] the support function $\delta^*_{K\cap L}$ is piecewise linear-quadratic, which implies that l is also piecewise linear-quadratic.

In addition, $F : \mathcal{X} \rightrightarrows \mathcal{Y}$ is called *piecewise polyhedral* if its graph is the union of finitely many polyhedral convex sets. Therefore, according to the following proposition established in [24], both $\mathcal{T}_f(\mathbf{x})$ and $\mathcal{T}_l(\mathbf{w}, \mathbf{z}, \mathbf{x})$ are piecewise polyhedral multivalued mappings.

Proposition 1 *[24] Let $\mathcal{X}$ be a finite-dimensional real Euclidean space and $\theta : \mathcal{X} \to (-\infty, +\infty]$ be a closed proper convex function. Then θ is piecewise linear-quadratic if and only if the graph of $\partial\theta$ is piecewise polyhedral. Moreover, θ is piecewise linear-quadratic if and only if its conjugate θ^* is piecewise linear-quadratic.*

In [23], Robinson established the following fundamental property to describe the locally upper Lipschitz continuity of a piecewise polyhedral multivalued mapping.

Proposition 2 *[23] If the multivalued mapping $F : \mathcal{X} \rightrightarrows \mathcal{Y}$ is piecewise polyhedral, then F is locally upper Lipschitz continuous at any $\mathbf{x}^0 \in \mathcal{X}$ with modulus κ_0 independent of the choice of $\mathbf{x}^0$. i.e., there exists a neighborhood V of $\mathbf{x}^0$ such that $F(\mathbf{x}) \subseteq F(\mathbf{x}^0) + \kappa_0\|\mathbf{x} - \mathbf{x}^0\|\boldsymbol{B}_{\mathcal{Y}}, \ \ \forall\, \mathbf{x} \in V$.*

Therefore, the above Propositions 1 and 2 imply that $\mathcal{T}_f(\mathbf{x})$ and $\mathcal{T}_l(\mathbf{w}, \mathbf{z}, \mathbf{x})$ are both locally upper Lipschitz continuous. Combining [31, Theorem 3H.3], we have the following result.

Proposition 3 *Assume that the KKT system* (9) *has at least one solution. Let $(\bar{\mathbf{w}}, \bar{\mathbf{z}}, \bar{\mathbf{x}})$ be a solution of the KKT system* (9)*. Then $\mathcal{T}_f^{-1}$ is metrically subregular at $\bar{\mathbf{x}}$ for the origin and $\mathcal{T}_l^{-1}$ is also metrically subregular at $(\bar{\mathbf{w}}, \bar{\mathbf{z}}, \bar{\mathbf{x}})$ for the origin, i.e., there exist a neighborhood of origin $\mathcal{V}$ and constants $\kappa_l > 0, \ \kappa_f > 0$ along with neighborhoods $\boldsymbol{B}_{\delta_l}(\bar{\mathbf{w}}, \bar{\mathbf{z}}, \bar{\mathbf{x}})$ and $\boldsymbol{B}_{\delta_f}(\bar{\mathbf{x}})$ such that*

$$dist\big(\mathbf{x}, \mathcal{T}_f^{-1}(0)\big) \le \kappa_f dist\big(0, \mathcal{T}_f(\mathbf{x}) \cap \mathcal{V}\big),$$

holds for any $\mathbf{x} \in \boldsymbol{B}_{\delta_f}(\bar{\mathbf{x}})$ and

$$dist\big((\mathbf{w}, \mathbf{z}, \mathbf{x}), \mathcal{T}_l^{-1}(0)\big) \le \kappa_l dist\big(0, \mathcal{T}_l(\mathbf{w}, \mathbf{z}, \mathbf{x}) \cap \mathcal{V}\big),$$

holds for any $(\mathbf{w}, \mathbf{z}, \mathbf{x}) \in \boldsymbol{B}_{\delta_l}(\bar{\mathbf{w}}, \bar{\mathbf{z}}, \bar{\mathbf{x}})$.

Besides, we may go one step further to present the error bound condition in some semilocal sense. Although Zhang et al. [32] presented a similar result without a proof, for the sake of completeness, we still present the detailed results and the proof.

Proposition 4 *For any $r > 0$ and $(\bar{\mathbf{w}}, \bar{\mathbf{z}}, \bar{\mathbf{x}}) \in \mathcal{T}_l^{-1}(0)$, there exist $\kappa_f(r) > 0$ and $\kappa_l(r) > 0$ such that*

$$dist(\mathbf{x}, \mathcal{T}_f^{-1}(0)) \le \kappa_f(r)\ dist(0, \mathcal{T}_f(\mathbf{x})), \tag{12a}$$

$$dist\big((\mathbf{w}, \mathbf{z}, \mathbf{x}), \mathcal{T}_l^{-1}(0)\big) \le \kappa_l(r) dist\big(0, \mathcal{T}_l(\mathbf{w}, \mathbf{z}, \mathbf{x})\big). \tag{12b}$$

hold for any $\mathbf{x} \in \mathcal{R}^n$ with dist$(\mathbf{x}, \mathcal{T}_f^{-1}(0)) \le r$ and $\|(\mathbf{w}, \mathbf{z}, \mathbf{x}) - (\bar{\mathbf{w}}, \bar{\mathbf{z}}, \bar{\mathbf{x}})\| \le r$.

Proof For the sake of contradiction, we assume that the first assertion about inequality (12a) is false. Then for some $\tilde{r} > 0$ and any $\kappa_f(\tilde{r}) = k > 0$, there exists $\mathbf{x}^k \in \mathcal{R}^n$ with $\text{dist}(\mathbf{x}^k, \mathcal{T}_f^{-1}(0)) \leq \tilde{r}$ such that

$$\text{dist}(\mathbf{x}^k, \mathcal{T}_f^{-1}(0)) > k\text{dist}(0, \mathcal{T}_f(\mathbf{x}^k)). \tag{13}$$

Next, by using the fact that $\mathcal{T}_f^{-1}(\mathbf{0})$ is compact, we know that $\{\mathbf{x}^k\}$ is a bounded sequence. Therefore, there exists a subsequence $\{\mathbf{x}^{k_j}\}$ such that $\mathbf{x}^{k_j} \to \mathbf{x}^*$ as $k_j \to +\infty$. Then, together with (13), we have

$$0 \leq \text{dist}\big(\mathbf{0}, \mathcal{T}_f(\mathbf{x}^*)\big) = \lim_{k_j \to +\infty} \text{dist}(\mathbf{0}, \mathcal{T}_f(\mathbf{x}^{k_j})) \leq \lim_{k_j \to +\infty} \frac{\text{dist}(\mathbf{x}^{k_j}, \mathcal{T}_f^{-1}(\mathbf{0}))}{k_j} \leq \lim_{k_j \to +\infty} \frac{\tilde{r}}{k_j} = 0,$$

which implies that $\mathbf{x}^* \in \mathcal{T}_f^{-1}(\mathbf{0})$. Moreover, there exist $\bar{k} > \kappa_f$ such that $\|\mathbf{x}^{k_j} - \mathbf{x}^*\| \leq \delta_f$ holds for all $k_j > \bar{k}$, where the parameters κ_f and δ_f have been defined in Proposition 3. Thus, for some $\varepsilon > 0$ and all $k_j > \max\{\bar{k}, \frac{\tilde{r}}{\varepsilon}\}$, we have

$$k_j\text{dist}(\mathbf{0}, \mathcal{T}_f(\mathbf{x}^{k_j})) \leq \text{dist}(\mathbf{x}^{k_j}, \mathcal{T}_f^{-1}(\mathbf{0})) \leq \kappa_f\text{dist}(\mathbf{0}, \mathcal{T}_f(\mathbf{x}^{k_j}) \cap \mathcal{B}_\varepsilon(\mathbf{0})) = \kappa_f\text{dist}(\mathbf{0}, \mathcal{T}_f(\mathbf{x}^{k_j})),$$

which, together with $k_j > \bar{k} > \kappa_f$, is a contradiction. Hence, the first assertion is true.

Similarly, for the sake of contradiction, we also assume that the second assertion about inequality (12b) is false. Then, there exist $\tilde{r} > 0$ and $(\tilde{\mathbf{w}}, \tilde{\mathbf{z}}, \tilde{\mathbf{x}}) \in \mathcal{T}_l^{-1}(\mathbf{0})$, for any $\kappa_l(\tilde{r}) = k > 0$, such that

$$\text{dist}\big((\mathbf{w}^k, \mathbf{z}^k, \mathbf{x}^k), \mathcal{T}_l^{-1}(\mathbf{0})\big) > k\text{dist}\big(\mathbf{0}, \mathcal{T}_l(\mathbf{w}^k, \mathbf{z}^k, \mathbf{x}^k)\big),\ \exists\, (\mathbf{w}^k, \mathbf{z}^k, \mathbf{x}^k) \in \mathcal{B}_{\tilde{r}}(\tilde{\mathbf{w}}, \tilde{\mathbf{z}}, \tilde{\mathbf{x}}). \tag{14}$$

Note that $\{(\mathbf{w}^k, \mathbf{z}^k, \mathbf{x}^k)\}$ is a bounded sequence. Hence, there exists a subsequence $\{(\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i})\}$ such that $(\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i}) \to (\tilde{\mathbf{w}}^*, \tilde{\mathbf{z}}^*, \tilde{\mathbf{x}}^*)$ as $k_i \to +\infty$. Then we have

$$\text{dist}\big(\mathbf{0}, \mathcal{T}_l(\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i})\big) < \frac{\text{dist}\big((\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i}), \mathcal{T}_l^{-1}(\mathbf{0})\big)}{k_i}. \tag{15}$$

Now taking the limits on both sides of the inequality (15), we have

$$0 \leq \text{dist}\big(\mathbf{0}, \mathcal{T}_l(\tilde{\mathbf{w}}^*, \tilde{\mathbf{z}}^*, \tilde{\mathbf{x}}^*)\big) \leq \lim_{k_i \to +\infty} \frac{\text{dist}\big((\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i}), \mathcal{T}_l^{-1}(\mathbf{0})\big)}{k_i} = 0,$$

which implies that $(\tilde{\mathbf{w}}^*, \tilde{\mathbf{z}}^*, \tilde{\mathbf{x}}^*) \in \mathcal{T}_l^{-1}(\mathbf{0})$. Note that $\mathcal{T}_l^{-1}$ is metrically subregular at $(\tilde{\mathbf{w}}^*, \tilde{\mathbf{z}}^*, \tilde{\mathbf{x}}^*)$ for the origin. Then, for all k_i sufficiently large and some $\varepsilon > 0$, we have

$$\text{dist}\big((\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i}), \mathcal{T}_l^{-1}(\mathbf{0})\big) \leq \kappa_l\text{dist}\big(\mathbf{0}, \mathcal{T}_l(\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i}) \cap \mathcal{B}_\varepsilon(\mathbf{0})\big) = \kappa_l\text{dist}\big(\mathbf{0}, \mathcal{T}_l(\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i})\big).$$

On the other hand, (14) implies that

$$\text{dist}\big((\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i}), \mathcal{T}_l^{-1}(\mathbf{0})\big) > k_i\text{dist}\big(\mathbf{0}, \mathcal{T}_l(\mathbf{w}^{k_i}, \mathbf{z}^{k_i}, \mathbf{x}^{k_i})\big).$$

Thus, by taking $k_i > \max(\varepsilon, \kappa_l)$, we obtain a contradiction. So (12b) is true. ■

3 The SSsNAL method for the SVM problems

In this section, we detailedly discuss how to apply the SSsNAL method to solve the problem (**D**) and establish its convergence theories.

3.1 The SSsNAL method for the problem (D)

Firstly, we provide the framework of the SSsNAL method. Given $\sigma > 0$, the AL function associated with the problem (**D**) is given as follows

$$\mathcal{L}_\sigma(\mathbf{w}, \mathbf{z}; \mathbf{x}) \;\; = \;\; \frac{1}{2}\langle \mathbf{w}, Q\mathbf{w}\rangle + \delta^*_{K\bigcap L}(\mathbf{z}) + \frac{\sigma}{2}\|Q\mathbf{w} + \mathbf{z} + \mathbf{c} - \frac{1}{\sigma}\mathbf{x}\|^2 - \frac{1}{2\sigma}\|\mathbf{x}\|^2,$$

where $(\mathbf{w}, \mathbf{z}, \mathbf{x}) \in \mathrm{Ran}(Q) \times \mathcal{R}^n \times \mathcal{R}^n$. The SSsNAL method for solving the problem (**D**) can be sketched as below.

Algorithm 1 : the SSsNAL method for the problem (**D**)

Let $\sigma_0 > 0$ be a given parameter. Choose $\mathbf{x}^0 \in \mathcal{R}^n$. For $k = 0, 1, 2, \ldots$, generate $(\mathbf{w}^{k+1}, \mathbf{z}^{k+1})$ and $\mathbf{x}^{k+1}$ by executing the following iterations:

Step 1. Apply the SsN method to compute

$$(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}) \approx \underset{\mathbf{w}\in \mathrm{Ran}(Q), \mathbf{z}\in\mathcal{R}^n}{\mathrm{argmin}} \big\{\Psi^k(\mathbf{w}, \mathbf{z}) := \mathcal{L}_{\sigma_k}(\mathbf{w}, \mathbf{z}; \mathbf{x}^k)\big\}. \tag{16}$$

Step 2. Compute

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \sigma_k(Q\mathbf{w}^{k+1} + \mathbf{z}^{k+1} + \mathbf{c}),$$

and update σ_{k+1}.

Notably, the inner subproblem (16) has no closed-form solution in general. So we consider how to solve it approximately with the following stopping criteria introduced in [22, 33]:

$$\text{(A)}\;\; \Psi^k(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}) - \inf_{\mathbf{w}\in\mathrm{Ran}(\mathcal{Q}),\mathbf{z}} \Psi^k(\mathbf{w}, \mathbf{z}) \le \epsilon_k^2/(2\sigma_k),\; \sum_{k=1}^{+\infty} \epsilon_k < +\infty,$$

$$\text{(B)}\;\; \Psi^k(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}) - \inf_{\mathbf{w}\in\mathrm{Ran}(\mathcal{Q}),\mathbf{z}} \Psi^k(\mathbf{w}, \mathbf{z}) \le \delta_k^2/(2\sigma_k)\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2,\; \sum_{k=1}^{+\infty} \delta_k < +\infty.$$

Since $\inf_{\mathbf{w}\in\mathrm{Ran}(\mathcal{Q}),\mathbf{z}} \Psi^k(\mathbf{w}, \mathbf{z})$ is unknown, in order to apply the stopping criteria (A) and (B) in Algorithm 1, we need to further analyze the following optimization problem.

$$\min\big\{\Psi^k(\mathbf{w}, \mathbf{z}) | (\mathbf{w}, \mathbf{z}) \in \mathrm{Ran}(Q) \times \mathcal{R}^n\big\}.$$

Obviously, it is easily seen from the definition of $\Psi^k(\mathbf{w}, \mathbf{z})$ in (16) that the above problem has a

unique optimal solution in $\mathrm{Ran}(\mathcal{Q}) \times \mathcal{R}^n$. For any $\mathbf{w} \in \mathrm{Ran}(Q)$, we define

$$\begin{aligned}\psi^k(\mathbf{w}) :=& \inf_{\mathbf{z}\in\mathcal{R}^n} \Psi^k(\mathbf{w},\mathbf{z})\\ =&\frac{1}{2}\langle \mathbf{w}, Q\mathbf{w}\rangle + \delta^*_{K\cap L}\left(\mathrm{Prox}_{\frac{1}{\sigma_k}\delta^*_{K\cap L}}(\mathbf{u}(\mathbf{w})/\sigma_k)\right)\\ &+\frac{\sigma_k}{2}\|\mathrm{Prox}_{\frac{1}{\sigma_k}\delta^*_{K\cap L}}(\mathbf{u}(\mathbf{w})/\sigma_k) - \mathbf{u}(\mathbf{w})/\sigma_k\|^2 - \frac{1}{2\sigma_k}\|\mathbf{x}^k\|^2,\\ =&\frac{1}{2}\langle \mathbf{w}, Q\mathbf{w}\rangle + \frac{1}{2\sigma_k}\left(\|\mathbf{u}(\mathbf{w})\|^2 - \|\mathbf{u}(\mathbf{w}) - \Pi_{K\cap L}(\mathbf{u}(\mathbf{w}))\|^2\right) - \frac{1}{2\sigma_k}\|\mathbf{x}^k\|^2,\end{aligned} \tag{17}$$

where $\mathbf{u}(\mathbf{w}) := \mathbf{x}^k - \sigma_k(Q\mathbf{w}+\mathbf{c})$ and the last equality directly follows from (2.2) in [34]. Then $(\mathbf{w}^{k+1}, \mathbf{z}^{k+1})$ in Step 1 of Algorithm 1 can be obtained in the following manner

$$\mathbf{w}^{k+1} \approx \mathrm{argmin}\{\psi^k(\mathbf{w}) | \mathbf{w} \in \mathrm{Ran}(Q)\}, \tag{18a}$$

$$\mathbf{z}^{k+1} = \mathrm{Prox}_{\frac{1}{\sigma_k}\delta^*_{K\cap L}}(\mathbf{u}(\mathbf{w}^{k+1})/\sigma_k) = \sigma_k^{-1}\big(\mathbf{u}(\mathbf{w}^{k+1}) - \Pi_{K\cap L}(\mathbf{u}(\mathbf{w}^{k+1}))\big), \tag{18b}$$

where the last equality in (18b) is due to the Moreau identity (10). Moreover, in combination with (18b), the update in Step 2 of Algorithm 1 can be simplified as

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \sigma_k\left(Q\mathbf{w}^{k+1} + \sigma_k^{-1}\left(\mathbf{u}(\mathbf{w}^{k+1}) - \Pi_{K\cap L}(\mathbf{u}(\mathbf{w}^{k+1}))\right) + \mathbf{c}\right) = \Pi_{K\cap L}(\mathbf{u}(\mathbf{w}^{k+1})).$$

Finally, note that $\psi^k(\mathbf{w})$ is continuously differentiable and strongly convex with modulus $\tilde{\lambda}_{\min}(Q)$ in $\mathrm{Ran}(Q)$, where $\tilde{\lambda}_{\min}(Q)$ is the minimum nonzero eigenvalue of Q. Then we have

$$\begin{aligned}&\Psi^k(\mathbf{w}^{k+1},\mathbf{z}^{k+1}) - \inf_{\mathbf{w}\in\mathrm{Ran}(\mathcal{Q}),\mathbf{z}} \Psi^k(\mathbf{w},\mathbf{z})\\ &= \psi^k(\mathbf{w}^{k+1}) - \inf_{\mathbf{w}\in\mathrm{Ran}(\mathcal{Q})}\psi^k(\mathbf{w}) \le \frac{1}{2\tilde{\lambda}_{\min}(Q)}\|\nabla\psi^k(\mathbf{w}^{k+1})\|^2,\end{aligned} \tag{19}$$

where the last inequality is due to Theorem $2.1.10$ in [35] and the fact that $\nabla\psi^k(\mathbf{w}^*) = \mathbf{0}$ with $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathrm{Ran}(\mathcal{Q})}\psi^k(\mathbf{w})$. Therefore, we replace the above stopping criteria (A) and (B) by the following easy-to-check criteria

$$(\mathrm{A}')\ \ \|\nabla\psi^k(\mathbf{w}^{k+1})\| \le \sqrt{\tilde{\lambda}_{\min}(Q)}\epsilon_k/\sqrt{\sigma_k},\ \epsilon_k \ge 0,\ \sum_{k=1}^{+\infty}\epsilon_k < +\infty,$$

$$(\mathrm{B}')\ \ \|\nabla\psi^k(\mathbf{w}^{k+1})\| \le \sqrt{\tilde{\lambda}_{\min}(Q)}\delta_k/\sqrt{\sigma_k}\|\mathbf{x}^{k+1}-\mathbf{x}^k\|,\ \delta_k \ge 0,\ \sum_{k=1}^{+\infty}\delta_k < +\infty.$$

Next, we shall adapt the results developed in [22, 33, 36] to establish the convergence theory of the AL method for the problem (**D**). Take a positive scalar r such that $\sum_{k=1}^{+\infty}\epsilon_k < r$, then we can state the convergence theory as below.

Theorem 1 *Let $\{(\mathbf{w}^k, \mathbf{z}^k, \mathbf{x}^k)\}$ be any infinite sequence generated by Algorithm 1 with stopping criterion (A') and (B') for solving subproblem* (18a)*. Let Ω_P be the solution set of (**P**) and $(\mathbf{w}^*, \mathbf{z}^*)$ be the unique optimal solution of (**D**). Then the sequence $\{\mathbf{x}^k\}$ converges to $\mathbf{x}^* \in \Omega_P$ and the*

sequence $(\mathbf{w}^k, \mathbf{z}^k)$ *converges to the unique optimal solution* $(\mathbf{w}^*, \mathbf{z}^*)$. *Moreover, if dist*$(\mathbf{x}^0, \Omega_P) \leq r - \sum_{k=1}^{+\infty} \epsilon_k$, *then for all* $k > 0$

$$\textit{dist}(\mathbf{x}^{k+1}, \Omega_P) \leq \mu_k \textit{dist}(\mathbf{x}^k, \Omega_P), \tag{20a}$$

$$\|(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}) - (\mathbf{w}^*, \mathbf{z}^*)\| \leq \mu_k' \|\mathbf{x}^{k+1} - \mathbf{x}^k\|, \tag{20b}$$

where $\mu_k := \left[\delta_k + (1+\delta_k)\kappa_f(r)/\sqrt{\kappa_f^2(r) + \sigma_k^2}\right]/(1-\delta_k) \to \mu_\infty := \kappa_f(r)/\sqrt{\kappa_f^2(r) + \sigma_\infty^2}$, $\mu_k' := \kappa_l(r)\sqrt{1/\sigma_k^2 + \tilde{\lambda}_{\min}(Q)\delta_k^2/\sigma_k} \to \mu_\infty' := \kappa_l(r)/\sigma_\infty$, $\kappa_f(r)$, $\kappa_l(r) > 0$ *are constant and the parameter* r *is determined by Proposition 4. Moreover,* μ_k *and* μ_k' *go to* 0 *as* $\sigma_k \uparrow \sigma_\infty = +\infty$.

Proof The statements on the global convergence directly follow from [22]. The proof for the first inequality (20a) follows from the similar idea of the proof in [32, Lemma 4.1], so we omit the details. Next, to prove the second inequality (20b), for the given parameter r, we have

$$\|(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}, \mathbf{x}^{k+1}) - (\mathbf{w}^*, \mathbf{z}^*, \mathbf{x}^*)\| \leq r, \ \forall\, k \geq 0,$$

which follows from the fact that $(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}, \mathbf{x}^{k+1}) \to (\mathbf{w}^*, \mathbf{z}^*, \mathbf{x}^*)$. Therefore, according to Proposition 4, we have

$$\|(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}) - (\mathbf{w}^*, \mathbf{z}^*)\| + \text{dist}(\mathbf{x}^{k+1}, \Omega_P) \leq \kappa_l(r)\text{dist}\big(\mathbf{0}, \mathcal{T}_l(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}, \mathbf{x}^{k+1})\big), \ \forall\, k \geq 0,$$

which, together with the estimate (4.21) in [22], implies

$$\|(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}) - (\mathbf{w}^*, \mathbf{z}^*)\| \leq \kappa_l(r)\left[\text{dist}^2(\mathbf{0}, \partial\Psi^k(\mathbf{w}^{k+1}, \mathbf{z}^{k+1})) + \sigma_k^{-2}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2\right]^{1/2}. \tag{21}$$

Then, according to the update rule of $(\mathbf{w}^{k+1}, \mathbf{z}^{k+1})$ in (18) and the Danskin-type theorem [37], one has

$$\text{dist}(\mathbf{0}, \partial\Psi^k(\mathbf{w}^{k+1}, \mathbf{z}^{k+1})) = \|\nabla\psi^k(\mathbf{w}^{k+1})\|. \tag{22}$$

In combination of (21), (22) and the stopping criterion (B'), we obtain that for all $k \geq 0$,

$$\|(\mathbf{w}^{k+1}, \mathbf{z}^{k+1}) - (\mathbf{w}^*, \mathbf{z}^*)\| \leq \mu_k' \|\mathbf{x}^{k+1} - \mathbf{x}^k\|,$$

where $\mu_k' := \kappa_l(r)\sqrt{1/\sigma_k^2 + \tilde{\lambda}_{\min}(Q)\delta_k^2/\sigma_k}$. This completes the proof of the theorem. ■

3.2 The SsN method for the inner subproblem (16)

In this subsection, we propose the SsN method to solve the inner subproblem (16). Additionally, we need to carefully study the structure of the projection operator $\Pi_{K \cap L}$ and its associated generalized Jacobian, which plays a fundamental role in the design of the SsN method.

3.2.1 The computation of the projection operator $\Pi_{K\cap L}$

In this part, we focus on how to compute the projection operator $\Pi_{K\cap L}(\mathbf{v})$ efficiently, where $\mathbf{v} \in \mathcal{R}^n$ is a given vector. Firstly, it follows from the definition of K and L in (8) that $\Pi_{K\cap L}(\mathbf{v})$ is the solution of the following optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{R}^n} \ & \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2 \\ s.t. \ & \mathbf{a}^T\mathbf{x} = d, \\ & \mathbf{l} \le \mathbf{x} \le \mathbf{u}. \end{aligned} \tag{23}$$

Furthermore, by introducing a slack variable y, the problem (23) can be reformulated as

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{R}^n,\mathbf{y}\in\mathcal{R}^n} \ & \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2 + \delta_K(\mathbf{y}) \\ s.t. \ & \mathbf{a}^T\mathbf{x} = d, \\ & \mathbf{x}-\mathbf{y} = 0, \end{aligned} \tag{24}$$

where δ_K is an indicator function for the polyhedral convex set K. Correspondingly, the dual problem of (24) is

$$\min_{\lambda\in\mathcal{R},\mathbf{z}\in\mathcal{R}^n} \ \frac{1}{2}\|\mathbf{v}-\lambda\mathbf{a}-\mathbf{z}\|^2 + \delta_K^*(\mathbf{z}) - \frac{1}{2}\|\mathbf{v}\|^2 + \lambda d, \tag{25}$$

and the associated KKT condition is

$$\begin{aligned} \mathbf{x} = \Pi_K(\mathbf{v}-\lambda\mathbf{a}), \ \ \mathbf{z} = \mathbf{v}-\lambda\mathbf{a}-\mathbf{x}, \\ \mathbf{a}^T\mathbf{x} = d, \ \ \mathbf{x} = \mathbf{y}, \ \ (\mathbf{x},\mathbf{y},\lambda,\mathbf{z}) \in \mathcal{R}^n \times K \times \mathcal{R} \times \mathcal{R}^n. \end{aligned} \tag{26}$$

Hence, combining (24), (25) and (26), it is sufficient for us to obtain $\Pi_{K\cap L}(\mathbf{v})$ by solving the problem (25). Let

$$\varphi(\lambda) := \inf_{\mathbf{z}} \big\{ \frac{1}{2}\|\mathbf{v}-\lambda\mathbf{a}-\mathbf{z}\|^2 + \delta_K^*(\mathbf{z}) + \lambda d \big\},$$

which implies that the optimal solution $(\hat{\lambda}, \hat{\mathbf{z}})$ of problem (25) is given by

$$\hat{\lambda} = \operatorname{argmin}\{\varphi(\lambda) | \lambda \in \mathcal{R}\}, \tag{27}$$

and

$$\hat{\mathbf{z}} = \mathbf{v} - \hat{\lambda}\mathbf{a} - \Pi_K(\mathbf{v}-\mathbf{a}\hat{\lambda}),$$

respectively. Since the function φ is convex and continuously differentiable with its gradient given by

$$\nabla\varphi(\lambda) = -\mathbf{a}^T\Pi_K(\mathbf{v}-\lambda\mathbf{a}) + d,$$

the optimal solution $\hat{\lambda}$ of (27) is the zero point of $\nabla\varphi$, i.e.,

$$\nabla\varphi(\hat{\lambda}) = 0. \tag{28}$$

It follows by [25] that $\nabla\varphi(\lambda)$ is a continuous non-decreasing piecewise linear function which has break points in the following set

$$T := \left\{ \frac{v_i - u_i}{a_i}, \frac{v_i - l_i}{a_i} \mid i = 1, \ldots, n \right\}.$$

Moreover, the range of $\nabla\varphi(\lambda)$ is a closed and bounded interval

$$\left[d + \sum_{k \in I_{a_-}} l_k |a_k| - \sum_{j \in I_{a_+}} u_j |a_j|, d + \sum_{k \in I_{a_-}} u_k |a_k| - \sum_{j \in I_{a_+}} l_j |a_j| \right],$$

where $I_{a_+} := \{i \,|\, a_i > 0, i = 1, \ldots, n\}$ and $I_{a_-} := \{i \,|\, a_i < 0, i = 1, \ldots, n\}$.

Next, we introduce the algorithm in [25] to obtain the zero point of the equation (28). The procedure consists of a binary search among the $2n$ break points until bracketing $\hat{\lambda}$ between two breakpoints. The algorithm can be summarized as follows

---

Algorithm 2 : The breakpoint search algorithm for solving (28) [25]

Initialization: Sort all break points given in (3.2.1) with an ascending order

$$t^{(1)} \leq t^{(2)} \leq \ldots \leq t^{(2n)}, \ \left(t^{(i)} \in T, \ \forall i = 1, \cdots, 2n \right)$$

Given $I_l = 1$, $I_u = 2n$, $f_l = \nabla\varphi(t^{(I_l)})$, and $f_u = \nabla\varphi(t^{(I_u)})$.
While $I_u - I_l > 1$ and $t^{(I_u)} > t^{(I_l)}$
  $I_m = [\frac{I_l + I_u}{2}]$, $f_m = \nabla\varphi(t^{(I_m)})$.
  if $f_m \geq 0$
    $I_u = I_m$, $f_u = f_m$.
  else
    $I_l = I_m$, $f_l = f_m$.
  end
end
if $f_u = f_l = 0$
  $\hat{\lambda} = t^{(I_l)}$.
else
  $\hat{\lambda} = t^{(I_l)} - \frac{f_l}{f_u - f_l}(t^{(I_u)} - t^{(I_l)})$.
end

Remark 1 *Compared with the bisection method used in [20], the breakpoint search algorithm in our paper can avoid the complicated operations on some index sets. Hence, it is often faster than the bisection method in the practical implementation.*

Finally, suppose that the optimal solution $\hat{\lambda}$ of (27) is obtained by Algorithm 2. Then, in combination with the KKT condition (26), we have

$$\begin{aligned} \Pi_{K \bigcap L}(\mathbf{v}) &= \Pi_K(\mathbf{v} - \hat{\lambda}\mathbf{a}) \\ &= \left(\Pi_{[l_1, u_1]}(v_1 - \hat{\lambda}a_1), \ldots, \Pi_{[l_n, u_n]}(v_n - \hat{\lambda}a_n) \right)^T, \end{aligned}$$

where

$$\Pi_{[l_i,u_i]}(v_i-\hat{\lambda}a_i)=\begin{cases}u_i & \text{if } v_i-\hat{\lambda}a_i\geq u_i,\\ v_i-\hat{\lambda}a_i & \text{if } l_i<v_i-\hat{\lambda}a_i<u_i,\\ l_i & \text{if } v_i-\hat{\lambda}a_i\leq l_i,\end{cases}\quad (i=1,\ldots,n).$$

3.2.2 The computation of the HS-Jacobian of $\Pi_{K\cap L}$

In the following, we proceed to find the HS-Jacobian of $\Pi_{K\cap L}$ at a given point $\mathbf{v}\in\mathcal{R}^n$. For the sake of clarity, we rewrite the box constraint in (23) as a general linear inequality constraint so that problem (23) becomes

$$\begin{array}{rl}\min\limits_{\mathbf{x}\in\mathcal{R}^n} & \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2\\ s.t. & \mathbf{a}^T\mathbf{x}=d,\\ & A\mathbf{x}\geq\mathbf{g},\end{array}\tag{29}$$

where $A:=[I_n^T,-I_n^T]^T\in\mathcal{R}^{2n\times n}$ and $\mathbf{g}:=[\mathbf{l}^T,-\mathbf{u}^T]^T\in\mathcal{R}^{2n}$. Denote

$$\mathcal{I}(\mathbf{v}):=\{i|\ A_i\Pi_{K\cap L}(\mathbf{v})=g_i, i=1,\ldots,2n\},\tag{30}$$

and $r=|\mathcal{I}(\mathbf{v})|$, the cardinality of $\mathcal{I}(\mathbf{v})$, where A_i is the ith row of the matrix A. Then, with the notation introduced above, we may compute the HS-Jacobian following the results below.

Theorem 2 *For any $\mathbf{v}\in\mathcal{R}^n$, let $\mathcal{I}(\mathbf{v})$ be given in* (30) *and*

$$\Sigma:=I_n-\mathit{Diag}(\boldsymbol{\theta})\in\mathcal{R}^{n\times n},\tag{31}$$

where $\boldsymbol{\theta}\in\mathcal{R}^n$ is defined as

$$\theta_i=\begin{cases}1 & i\in\mathcal{I}(\mathbf{v}),\\ 0 & \mathit{otherwise}.\end{cases}\quad i=1,\ldots,n.$$

Then

$$P=\begin{cases}\Sigma(I_n-\frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}\mathbf{a}\mathbf{a}^T)\Sigma & \mathit{if}\ \mathbf{a}^T\Sigma\mathbf{a}\neq 0,\\ \Sigma & \mathit{otherwise},\end{cases}\tag{32}$$

is the HS-Jacobian of $\Pi_{K\cap L}$ at $\mathbf{v}$.

Proof Firstly, it follows from Theorem 1 in [38] that the following matrix

$$P_0=I_n-\left[\begin{array}{cc}A_{\mathcal{I}(\mathbf{v})}^T & \mathbf{a}\end{array}\right]\left(\left[\begin{array}{c}A_{\mathcal{I}(\mathbf{v})}\\ \mathbf{a}^T\end{array}\right]\left[\begin{array}{cc}A_{\mathcal{I}(\mathbf{v})}^T & \mathbf{a}\end{array}\right]\right)^{\dagger}\left[\begin{array}{c}A_{\mathcal{I}(\mathbf{v})}\\ \mathbf{a}^T\end{array}\right]\tag{33}$$

is the HS-Jacobian of $\Pi_{K\cap L}$ at $\mathbf{v}$, where $A_{\mathcal{I}(\mathbf{v})}$ is the matrix consisting of the rows of A indexed by $\mathcal{I}(\mathbf{v})$. Moreover, the definitions of $\mathcal{I}(\mathbf{v})$ and $A_{\mathcal{I}(\mathbf{v})}$ imply that $A_{\mathcal{I}(\mathbf{v})}A_{\mathcal{I}(\mathbf{v})}^T=I_r$ and $A_{\mathcal{I}(\mathbf{v})}^TA_{\mathcal{I}(\mathbf{v})}=I_n-\Sigma$.

Secondly, we focus on the calculation of the Moore-Penrose pseudo-inverse in (33). Above all, it is easy to verify that

$$\det\left(\begin{bmatrix} A_{\mathcal{I}(\mathbf{v})} \\ \mathbf{a}^T \end{bmatrix}\begin{bmatrix} A_{\mathcal{I}(\mathbf{v})}^T & \mathbf{a} \end{bmatrix}\right) = \mathbf{a}^T\Sigma\mathbf{a}.$$

Then, on one hand, if $\mathbf{a}^T\Sigma\mathbf{a} \neq 0$, we have that

$$\begin{aligned} P_0 =& I_n - \begin{bmatrix} A_{\mathcal{I}(\mathbf{v})}^T & \mathbf{a} \end{bmatrix}\left(\begin{bmatrix} A_{\mathcal{I}(\mathbf{v})} \\ \mathbf{a}^T \end{bmatrix}\begin{bmatrix} A_{\mathcal{I}(\mathbf{v})}^T & \mathbf{a} \end{bmatrix}\right)^{-1}\begin{bmatrix} A_{\mathcal{I}(\mathbf{v})} \\ \mathbf{a}^T \end{bmatrix}, \\ =& I_n - \begin{bmatrix} A_{\mathcal{I}(\mathbf{v})}^T & \mathbf{a} \end{bmatrix}\begin{pmatrix} I_r + \frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}A_{\mathcal{I}(\mathbf{v})}\mathbf{a}\mathbf{a}^T A_{\mathcal{I}(\mathbf{v})}^T & -\frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}A_{\mathcal{I}(\mathbf{v})}\mathbf{a} \\ -\frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}\mathbf{a}^T A_{\mathcal{I}(\mathbf{v})}^T & \frac{1}{\mathbf{a}^T\Sigma\mathbf{a}} \end{pmatrix}\begin{bmatrix} A_{\mathcal{I}(\mathbf{v})} \\ \mathbf{a}^T \end{bmatrix}, \\ =& \Sigma(I_n - \frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}\mathbf{a}\mathbf{a}^T)\Sigma, \end{aligned}$$

where the last equality holds true because $\Sigma = I_n - A_{\mathcal{I}(\mathbf{v})}^T A_{\mathcal{I}(\mathbf{v})}$ and $\Sigma = \Sigma^2$.

On the other hand, if $\mathbf{a}^T\Sigma\mathbf{a} = (\Sigma\mathbf{a})^T\Sigma\mathbf{a} = 0$, i.e., $\Sigma\mathbf{a} = \mathbf{0}$, then

$$\begin{aligned} P_0 =& I_n - \begin{bmatrix} A_{\mathcal{I}(\mathbf{v})}^T & \mathbf{a} \end{bmatrix}\begin{pmatrix} I_r - \frac{\mathbf{a}^T\mathbf{a}+2}{(1+\mathbf{a}^T\mathbf{a})^2}A_{\mathcal{I}(\mathbf{v})}\mathbf{a}\mathbf{a}^T A_{\mathcal{I}(\mathbf{v})}^T & \frac{1}{(1+\mathbf{a}^T\mathbf{a})^2}A_{\mathcal{I}(\mathbf{v})}\mathbf{a} \\ \frac{1}{(1+\mathbf{a}^T\mathbf{a})^2}\mathbf{a}^T A_{\mathcal{I}(\mathbf{v})}^T & \frac{\mathbf{a}^T\mathbf{a}}{(1+\mathbf{a}^T\mathbf{a})^2} \end{pmatrix}\begin{bmatrix} A_{\mathcal{I}(\mathbf{v})} \\ \mathbf{a}^T \end{bmatrix} \\ =& \Sigma, \end{aligned} \tag{34}$$

where the first equality is due to the definition of the Moore-Penrose pseudo-inverse and (33), and the last equality is owing to $A_{\mathcal{I}(\mathbf{v})}^T A_{\mathcal{I}(\mathbf{v})}\mathbf{a} = \mathbf{a}$. Thus, the proof is completed. ■

3.2.3 The SsN method for the inner subproblem (16)

In this part, we formally present the SsN method for the subproblem (16). Recall that we need to solve the convex subproblem in each iteration of Algorithm 1, i.e.,

$$\min_{\mathbf{w}\in\mathrm{Ran}(Q)}\left\{\psi^k(\mathbf{w}) = \frac{1}{2}\langle\mathbf{w}, Q\mathbf{w}\rangle + \frac{1}{2\sigma_k}\left(\|\mathbf{u}(\mathbf{w})\|^2 - \|\mathbf{u}(\mathbf{w}) - \Pi_{K\cap L}(\mathbf{u}(\mathbf{w}))\|^2\right) - \frac{1}{2\sigma_k}\|\mathbf{x}^k\|^2\right\}.$$

Furthermore, note that

$$\nabla\psi^k(\mathbf{w}) = Q\mathbf{w} - Q\Pi_{K\cap L}(\mathbf{u}(\mathbf{w})), \tag{35}$$

which implies that the optimal solution $\bar{\mathbf{w}}$ can be obtained through solving the following nonsmooth piecewise affine equation

$$\nabla\psi^k(\mathbf{w}) = 0, \quad \mathbf{w} \in \mathrm{Ran}(Q). \tag{36}$$

Let $\mathbf{w} \in \mathrm{Ran}(Q)$ be any given point. We define the following operator

$$\hat{\partial}^2\psi^k(\mathbf{w}) := Q + \sigma_k Q\mathscr{P}(\mathbf{u}(\mathbf{w}))Q,$$

where the multivalued mapping $\mathscr{P} : \mathcal{R}^n \rightrightarrows \mathcal{R}^{n\times n}$ is the HS-Jacobian of $\Pi_{K\cap L}$ [29].

Now we are ready to state the SsN method as below.

Algorithm 3 : the SsN method for the problem (16)

Given $\mu \in (0, \frac{1}{2}), \delta \in (0,1), \tau \in (0,1]$ and $\eta \in (0,1)$. Adopt a gradient method (see Algorthim 4 in Section 4.2) to generate an initial point $\mathbf{w}^0 \in \mathcal{R}^n$. Iterate the following steps: For $j = 0, 1, 2, \ldots,$

Step 1. Let $\mathcal{M}_j := Q + \sigma Q \mathcal{P}_j Q$, where $\mathcal{P}_j \in \mathscr{P}(\mathbf{u}(\mathbf{w}^j))$. Apply the conjugate gradient (CG) method to find an approximate solution d_{w^j} to the following linear system

$$\mathcal{M}_j \mathbf{d}_{\mathbf{w}^j} + \nabla \psi^k(\mathbf{w}^j) = 0, \quad \mathbf{d}_{\mathbf{w}^j} \in \mathrm{Ran}(Q) \tag{37}$$

such that

$$\|\mathcal{M}_j \mathbf{d}_{\mathbf{w}^j} + \nabla \psi^k(\mathbf{w}^j)\| \le \min(\eta, \|\nabla \psi^k(\mathbf{w}^j)\|^{1+\tau}).$$

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

$$\psi^k(\mathbf{w}^j + \delta^m \mathbf{d}_{\mathbf{w}^j}) \le \psi^k(\mathbf{w}^j) + \mu \delta^m \langle \nabla \psi^k(\mathbf{w}^j), \mathbf{d}_{\mathbf{w}^j} \rangle.$$

Step 3. Set $\mathbf{w}^{j+1} = \mathbf{w}^j + \alpha_j \mathbf{d}_{\mathbf{w}^j}$.

Remark 2
(i) Since the matrix $\mathcal{M}_j$ is positive definite in Ran(Q) and $\nabla \psi^k(\mathbf{w}^j) \in$ Ran(Q), the linear system (37) *has a unique solution in Ran(Q).*
(ii) In Algorithm 3, we adopt a gradient method to generate an initial point. Actually, this process is very essential because if we choose an arbitrary initial point, we can hardly get the sparse structure in the early Newton iterations and the cost of the Newton step may be very high. We will present some necessary details about the above gradient method in Section 4.2.

Before analyzing the local convergence rate of the SsN method, we first consider the strong semismoothness of $\nabla \psi^k$. For the definitions of semismoothness and γ-order semismoothness, one may see the below.

Definition 3 *(semismoothness) [39, 40] Let $\mathcal{O} \subseteq \mathcal{R}^n$ be an open set, $\mathcal{K} : \mathcal{O} \subseteq \mathcal{R}^n \rightrightarrows \mathcal{R}^{n \times m}$ be a nonempty and compact valued, upper-semicontinuous set-valued mapping, and $F : \mathcal{O} \to \mathcal{R}^n$ be a locally Lipschitz continuous function. F is said to be semismooth at $x \in O$ with respect to the multifunction $\mathcal{K}$ if F is directionally differentiable at x and for any $V \in \mathcal{K}(x + \Delta x)$ with $\Delta x \to 0$,*

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

Let γ be a positive constant. F is said to be γ-order (strongly, if $\gamma = 1$) semismooth at X with respect to $\mathcal{K}$ if F is directionally differentiable at x and for any $V \in \mathcal{K}(x + \Delta x)$ with $\Delta x \to 0$,

$$F(x + \Delta x) - F(x) - V \Delta x = O(\|\Delta x\|^{1+\gamma}).$$

According to [29, Lemma 2.1] and [41, Theorem 7.5.17], we can immediately obtain that $\nabla \psi^k$ is strongly semismooth at $\mathbf{w}$ with respect to $\hat{\partial}^2 \psi^k$. The local convergence rate for Algorithm 3 is stated in the next theorem without proof.

Theorem 4 *Let $\{\mathbf{w}^j\}$ be the infinite sequence generated by Algorithm 3. Then $\{\mathbf{w}^j\}$ converges to the unique optimal solution $\tilde{\mathbf{w}} \in$ Ran(Q) to problem* (36) *and*

$$\|\mathbf{w}^{j+1} - \tilde{\mathbf{w}}\| = O(\|\mathbf{w}^{j+1} - \tilde{\mathbf{w}}\|^{1+\tau}).$$

3.2.4 An efficient implementation for solving the linear system (37)

In this part, we discuss how to solve the linear system of equations (37) efficiently. In Algorithm 3, the most time-consuming step is solving the linear system (37), so it is essential to make full use of the the sparse structure of the coefficient matrix to design an efficient algorithm to obtain the descent direction $\mathbf{d}_{\mathbf{w}^j}$. For the sake of convenience, we ignore the subscript and rewrite the linear system (37) as follows

$$(Q + \sigma Q\mathcal{P}Q)\,\mathbf{d} = -\nabla\psi^k(\mathbf{w}), \quad \mathbf{d} \in \mathrm{Ran}(Q), \tag{38}$$

where $\mathcal{P} \in \mathscr{P}(\mathbf{u}(\mathbf{w}))$ and $\mathbf{u}(\mathbf{w}) = \mathbf{x}^k - \sigma_k(Q\mathbf{w} + \mathbf{c})$. Although Q may be singular, the subspace constraint and the fact that $\nabla\psi^k(\mathbf{w}) \in \mathrm{Ran}(Q)$ together imply that there exists a unique solution to the linear system (38). It is extraordinarily time-consuming to solve (38) directly when the dimension of Q is large. However, we only need to update $Q\mathbf{d}$ and $\mathbf{d}^TQ\mathbf{d}$ in every iteration of Algorithm 1. Hence, as shown in the next proposition, instead of computing the solution $\hat{\mathbf{d}}$ of (38) directly, we may choose to solve a relatively much smaller linear system to obtain $Q\hat{\mathbf{d}}$ and $\hat{\mathbf{d}}^TQ\hat{\mathbf{d}}$ by deliberately employing the sparse structure of the HS-Jacobian.

Proposition 5 *Let the index set $\mathcal{J} = \{1, \cdots, n\}/\mathcal{I}(\mathbf{u}(\mathbf{w}))$, where $\mathcal{I}(\mathbf{u}(\mathbf{w}))$ is defined by* (30) *and $q = |\mathcal{J}|$ denotes the cardinality of $\mathcal{J}$. Moreover, Let $\Sigma \in \mathcal{R}^{n\times n}$ be a diagonal matrix whose i-th diagonal element Σ_{ii} is equal to 1 if $i \in \mathcal{J}$, otherwise Σ_{ii} is equal to 0. Assume that $\hat{\mathbf{d}} \in Ran(Q)$ is the solution to the linear system* (38)*.*

(a) *If $\mathbf{a}^T\Sigma\mathbf{a} \neq 0$, then we have*

$$Q\hat{\mathbf{d}} = Q_{\mathcal{J}}^T\mathbf{v}_{\mathcal{J}}(\mathbf{w}) - \nabla\psi^k(\mathbf{w}) + \sigma\frac{\mathbf{a}_{\mathcal{J}}^TQ_{\mathcal{J}\mathcal{J}}\mathbf{v}_{\mathcal{J}}(\mathbf{w}) - \mathbf{a}_{\mathcal{J}}^T\nabla\psi_{\mathcal{J}}^k(\mathbf{w})}{\mathbf{a}_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}_{\mathcal{J}}^TQ_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}}Q_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}} \tag{39}$$

and

$$\begin{aligned}\hat{\mathbf{d}}^TQ\hat{\mathbf{d}} =& \mathbf{v}_{\mathcal{J}}^T(\mathbf{w})Q_{\mathcal{J}\mathcal{J}}\mathbf{v}_{\mathcal{J}}(\mathbf{w}) - 2\mathbf{v}_{\mathcal{J}}^T(\mathbf{w})\nabla\psi_{\mathcal{J}}^k(\mathbf{w}) + (\mathbf{s}(\mathbf{w}))^TQ\mathbf{s}(\mathbf{w}) \\ &+ \frac{2\sigma\mathbf{a}_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}} - \sigma^2\mathbf{a}_{\mathcal{J}}^TQ_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}}{\left(\mathbf{a}_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}_{\mathcal{J}}^TQ_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}\right)^2}(\mathbf{a}_{\mathcal{J}}^TQ_{\mathcal{J}\mathcal{J}}\mathbf{v}_{\mathcal{J}}(\mathbf{w}) - \mathbf{a}_{\mathcal{J}}^T\nabla\psi_{\mathcal{J}}^k(\mathbf{w}))^2,\end{aligned} \tag{40}$$

where $\mathbf{s}(\mathbf{w}) = \mathbf{w} - \Pi_{K\cap L}(\mathbf{u}(\mathbf{w}))$ and $\mathbf{v}_{\mathcal{J}}(\mathbf{w}) \in \mathcal{R}^q$ is the solution to the following linear system

$$\left(\frac{1}{\sigma}I_q + Q_{\mathcal{J}\mathcal{J}} + \frac{\sigma Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}^TQ_{\mathcal{J}\mathcal{J}}}{\mathbf{a}_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}_{\mathcal{J}}^TQ_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right)\mathbf{v}_{\mathcal{J}}(\mathbf{w}) = \left(I_q + \frac{\sigma Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}^T}{\mathbf{a}_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}_{\mathcal{J}}^TQ_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right)\nabla\psi_{\mathcal{J}}^k(\mathbf{w}), \tag{41}$$

and $Q_{\mathcal{J}\mathcal{J}} \in \mathcal{R}^{q\times q}$ is the submatrix of Q with those rows and columns in $\mathcal{J}$. Moreover, $\nabla\psi_{\mathcal{J}}^k(\mathbf{w}) \in \mathcal{R}^q$, $\mathbf{a}_{\mathcal{J}} \in \mathcal{R}^q$ and $Q_{\mathcal{J}} \in \mathcal{R}^{q\times n}$ are matrices consisting of the rows of $\nabla\psi^k(\mathbf{w})$, $\mathbf{a}$ and Q indexed by $\mathcal{J}$, respectively.

(b) *If $\mathbf{a}^T\Sigma\mathbf{a} = 0$, then we have*

$$Q\hat{\mathbf{d}} = Q_{\mathcal{J}}^T\mathbf{v}_{\mathcal{J}}(\mathbf{w}) - \nabla\psi^k(\mathbf{w}) \tag{42}$$

and

$$\hat{\mathbf{d}}^TQ\hat{\mathbf{d}} = \mathbf{v}_{\mathcal{J}}^T(\mathbf{w})Q_{\mathcal{J}\mathcal{J}}\mathbf{v}_{\mathcal{J}}(\mathbf{w}) - 2\mathbf{v}_{\mathcal{J}}^T(\mathbf{w})\nabla\psi_{\mathcal{J}}^k(\mathbf{w}) + (\mathbf{s}(\mathbf{w}))^TQ\mathbf{s}(\mathbf{w}), \tag{43}$$

where $Q_{\mathcal{J}\mathcal{J}}$, $Q_{\mathcal{J}}$, $\nabla\psi^k_{\mathcal{J}}(\mathbf{w})$ and $\mathbf{a}_{\mathcal{J}}$ are same as those in (a) and $\mathbf{v}_{\mathcal{J}}(\mathbf{w}) \in \mathcal{R}^q$ is the solution to the following linear system

$$\left(\frac{1}{\sigma} I_q + Q_{\mathcal{J}\mathcal{J}}\right) \mathbf{v}_{\mathcal{J}}(\mathbf{w}) = \nabla\psi^k_{\mathcal{J}}(\mathbf{w}). \tag{44}$$

Proof To prove part (a), combining with Theorem 2 and the condition $\mathbf{a}^T\Sigma\mathbf{a} \neq 0$, we know that $\mathcal{P} = \Sigma(I_n - \frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}\mathbf{a}\mathbf{a}^T)\Sigma$ is an element in the HS-Jacobian of $\Pi_{K\cap L}$ at $\mathbf{u}(\mathbf{w})$. Then according to (35), we have $\nabla\psi^k(\mathbf{w}) = Q\mathbf{s}(\mathbf{w})$. Thus, without loss of generality, we assume that Q can be decomposed as $Q = LL^T$ where $L \in \mathcal{R}^{n\times r}$ is a full collum rank matrix and $r = \text{Rank}(Q)$. Then, substituting $Q = LL^T$ into (38), we obtain that

$$L\left(I_r + \sigma L^T\mathcal{P}L\right)L^T\mathbf{d} = -LL^T\mathbf{s}(\mathbf{w}), \quad \mathbf{d} \in \text{Ran}(Q). \tag{45}$$

Since L has a full collum rank and $\mathcal{P} = \Sigma(I_n - \frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}\mathbf{a}\mathbf{a}^T)\Sigma$, (45) is further equivalent to

$$\left(I_r + \sigma L^T\Sigma(I_n - \frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}\mathbf{a}\mathbf{a}^T)\Sigma L\right)L^T\mathbf{d} = -L^T\mathbf{s}(\mathbf{w}), \quad \mathbf{d} \in \text{Ran}(Q). \tag{46}$$

Since $\hat{\mathbf{d}} \in \text{Ran}(Q)$ is the solution to the linear system (38), we have

$$\begin{aligned}
L^T\hat{\mathbf{d}} =& -\left(I_r + \sigma L^T\Sigma(I_n - \frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}\mathbf{a}\mathbf{a}^T)\Sigma L\right)^{-1} L^T\mathbf{s}(\mathbf{w}) \\
=& -\left(I_r - \frac{\sigma}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}} L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}} + \sigma L^T_{\mathcal{J}}L_{\mathcal{J}}\right)^{-1} L^T\mathbf{s}(\mathbf{w}) \\
=& -\left(I_r + \frac{\sigma L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}}}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}}L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right) L^T\mathbf{s}(\mathbf{w}) + \left(I_r + \frac{\sigma L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}}}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}}L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right) L^T_{\mathcal{J}} \\
& \left(\frac{1}{\sigma} I_q + L_{\mathcal{J}}L^T_{\mathcal{J}} + \frac{\sigma L_{\mathcal{J}}L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}}L^T_{\mathcal{J}}}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}}L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right)^{-1} L_{\mathcal{J}}\left(I_r + \frac{\sigma L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}}}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}}L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right) L^T\mathbf{s}(\mathbf{w}) \\
=& -L^T\mathbf{s}(\mathbf{w}) - \frac{\sigma L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}\nabla\psi^k_{\mathcal{J}}(\mathbf{w})}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}} + L^T_{\mathcal{J}}\left(I_q + \frac{\sigma\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}Q_{\mathcal{J}\mathcal{J}}}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right) \\
& \left(\frac{1}{\sigma} I_q + Q_{\mathcal{J}\mathcal{J}} + \frac{\sigma Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}Q_{\mathcal{J}\mathcal{J}}}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right)^{-1}\left(I_q + \frac{\sigma Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right)\nabla\psi^k_{\mathcal{J}}(\mathbf{w}) \\
=& -L^T\mathbf{s}(\mathbf{w}) - \frac{\sigma L^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}\nabla\psi^k_{\mathcal{J}}(\mathbf{w})}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}} + L^T_{\mathcal{J}}\left(I_q + \frac{\sigma\mathbf{a}_{\mathcal{J}}\mathbf{a}^T_{\mathcal{J}}Q_{\mathcal{J}\mathcal{J}}}{\mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}^T_{\mathcal{J}}Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}}}\right)\mathbf{v}_{\mathcal{J}}(\mathbf{w})
\end{aligned} \tag{47}$$

where the first equality is thanks to (45) and the nonsingularity of $\left(I_r + \sigma L^T\Sigma(I_n - \frac{1}{\mathbf{a}^T\Sigma\mathbf{a}}\mathbf{a}\mathbf{a}^T)\Sigma L\right)$; the second equality follows from the special 0-1 structure of the diagonal matrix Σ and the fact that $L^T\Sigma\Sigma L = L^T_{\mathcal{J}}L_{\mathcal{J}}$, $\mathbf{a}^T\Sigma\mathbf{a} = \mathbf{a}^T_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}$ and $\mathbf{a}^T\Sigma L = \mathbf{a}^T_{\mathcal{J}}L_{\mathcal{J}}$, where $L_{\mathcal{J}} \in \mathcal{R}^{q\times r}$ is a matrix consisting of the rows of L indexed by $\mathcal{J}$; the third equality is obtained by using the Sherman-Morrison-Woodbury formula [42] twice; the fourth equality is due to $Q_{\mathcal{J}\mathcal{J}} = L_{\mathcal{J}}L^T_{\mathcal{J}}$ and $\nabla\psi^k_{\mathcal{J}}(\mathbf{w}) = L_{\mathcal{J}}L^T\mathbf{s}(\mathbf{w})$. Finally, it is immediately follows from $Q\hat{\mathbf{d}} = LL^T\hat{\mathbf{d}}$, $\hat{\mathbf{d}}^TQ\hat{\mathbf{d}} = (L^T\hat{\mathbf{d}})^TL^T\hat{\mathbf{d}}$ and (47) that (39) and (40) hold true, thus concluding the proof of part (a).

As for part (b), since $\mathbf{a}^T\Sigma\mathbf{a} = 0$, we have $\mathcal{P} = \Sigma \in \mathscr{P}(\mathbf{u}(\mathbf{w}))$ according to Theorem 2. Similar to the proof for part (a), we can obtain the desired results (42)-(44). ■

Remark 3 *In the proof of Proposition 5, we always assume that* $\mathbf{a}_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}_{\mathcal{J}}^T Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}} \neq 0$*, i.e., the matrix* $I_r - \frac{\sigma}{\mathbf{a}_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}}} L_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}}\mathbf{a}_{\mathcal{J}}^T L_{\mathcal{J}}$ *in* (47) *is invertible. Actually this assumption is reasonable because* $\mathbf{a}_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}} = \mathbf{a}^T\Sigma\mathbf{a} \neq 0$ *and* σ *can be adjusted in an appropriate way in the implementation of the algorithm to avoid* $\mathbf{a}_{\mathcal{J}}^T\mathbf{a}_{\mathcal{J}} - \sigma\mathbf{a}_{\mathcal{J}}^T Q_{\mathcal{J}\mathcal{J}}\mathbf{a}_{\mathcal{J}} = 0$.

From the above discussion, we can see that the computational costs for solving the Newton linear system (37) are reduced significantly from $O(n^3)$ to $O(q^3)$. Here the number q is equal to the number of support vectors in the SVMs, which is usually much smaller than the number of samples n [43]. So we can always solve the linear system (41) or (44) at very low costs.

4 Numerical experiments

In this section, we demonstrate the performances of SSsNAL method by solving various SVMs problems (7) on the benchmark datasets from the LIBSVM data [15]. As a comparison, three state-of-the-art SVMs solvers, P2GP [19], LIBSVM [15] and FAPG [20] are used to solve the same problems. Note that P2GP is a two-phase gradient-based method designed for problem (7) and its MATLAB code can be downloaded from *https://github.com/diserafi/P2GP*. Meanwhile, LIBSVM implements the sequential minimal optimization method [14] to solve problem (7) which is available on *https://www.csie.ntu.edu.tw/ cjlin/libsvm/index.html*, whereas FAPG is a general optimization algorithm based on an accelerated proximal gradient method and its code is programmed by ourselves in MATLAB.

All experiments are implemented in MATLAB R2018b on a PC with the Intel Core i7-6700HQ processors (2.60GHz) and 8 GB of physical memory.

4.1 The stopping criterion and the parameters setting

We use the following relative KKT residual as a stopping criterion for all the algorithms which only involves variable $\mathbf{x}$ in the primal problem (7).

$$\mathrm{R_{KKT}}(\mathbf{x}) = \frac{\|\mathbf{x} - \Pi_{K\cap L}(\mathbf{x} - Q\mathbf{x} - \mathbf{c})\|}{1 + \|\mathbf{x}\|} < \mathrm{Tol},$$

where Tol[1] $= 10^{-3}$. We also set the maximum number of iterations for the SSsNAL, FAPG, P2GP and LINBSVM to be 200, 20000, 20000 and 20000, respectively. Except for the stopping criterion and the maximum number of iteration, the remaining parameters for the FAPG, P2GP and LINBSVM solvers are set as default. Moreover, the initial values of variables $\mathbf{x}^0$ and $\mathbf{w}^0$ are set to be zero vectors in our SSsNAL algorithm.

[1]Note that our SSsNAL algorithm can generate a high accuracy solution with Tol $= 10^{-6}$. But we found that the error rate of classification with the high accuracy solution did not improve, so here we set the Tol $= 10^{-3}$.

4.2 The gradient method to generate an initial point for the SsN method

In this subsection, we briefly describe how to implement the gradient method on the subproblem (18a) to obtain an initial point for the SsN method. The template of the gradient method is given as below.

Algorithm 4 : the gradient method to generate an initial point for the problem (18a)

Given $\mu \in (0, \frac{1}{2}), \delta \in (0,1), \varepsilon_G > 0$ and $n_{\max} = \min\{n, \lfloor 3.6 \times 10^7/n \rfloor\}$. Set the initial point $\mathbf{w}^0 = \mathbf{0} \in \mathcal{R}^n$ and $j = 0$. Then, iterate the following steps:

Step 1. Set $\mathbf{d}_{\mathbf{w}^j} = -\nabla\psi^k(\mathbf{w}^j)$

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

$$\psi^k(\mathbf{w}^j + \delta^m \mathbf{d}_{\mathbf{w}^j}) \leq \psi^k(\mathbf{w}^j) + \mu\delta^m \langle \nabla\psi^k(\mathbf{w}^j), \mathbf{d}_{\mathbf{w}^j} \rangle.$$

Step 3. Update $\mathbf{w}^{j+1} = \mathbf{w}^j + \alpha_j \mathbf{d}_{\mathbf{w}^j}$.

If $n - |\mathcal{I}(\mathbf{u}(\mathbf{w}^{j+1}))| < n_{\max}$ or $\|\nabla\psi^k(\mathbf{w}^{j+1})\| \leq \varepsilon_G$, terminate the algorithm; otherwise, return to Step 1 with $j = j + 1$.

Note that the sequences $\{\mathbf{w}^j\}$ generated by the gradient method are feasible solutions of (18a), i.e., $\mathbf{w}^j \in \text{Ran}(Q)$ $(j = 1, 2, \ldots)$, because of $\mathbf{w}^0, \mathbf{d}_{\mathbf{w}^j} \in \text{Ran}(Q)$. Moreover, the gradient method will also be stopped when the maximum number of 50 iterations is reached.

4.3 Results on the LIBSVM datasets

In this subsection, the performances of the SSsNAL, FAPG, P2PG and LIBSVM on the LIBSVM datasets for the C-SVC problem (2) are presented. The penalty parameter C in (2) is set to be 10 and the parameter α in RBF kernel function is set to be 100. Moreover, we scaled the LIBSVM datasets, so that each sample $\tilde{\mathbf{x}}_i \in [0,1]^q, (i = 1, 2, \ldots, n)$. The details of the LIBSVM datasets (size of the problem, features and nonzeros in the data) are presented in Table 1.

We report the numerical results in Tables 2 and 3. For the numerical results, we report the data set name (Data), the number of samples (n) and features (q), the number of support vectors (suppvec), the relative KKT residual (R_{KKT}), the computing time (Time), the percentage of training error ($\text{Err}_{\text{train}}$) and the percentage of the testing error (Err_{test}). The computation time is in the format of "hours:minutes:seconds", and "00" in the time column means that the elapsed time is less than 0.5 second. For each LIBSVM data set, the C-SVC model was trained on the training set and the testing data were used to verify the classification efficiency. As for those datasets which do not have the corresponding testing data, we randomly and uniformly sampled 80% of the data from the dataset as a training set, and used the remained 20% as a testing set. In order to eliminate the effects of randomness, we repeated the above process 10 times and got an average result. Moreover, the reported support vectors are obtained from the SSsNAL method.

Firstly, in Table 2, we present the performances of the four algorithms on various of C-SVC problems (2) with a linear kernel function. Note that our SSsNAL method is not only achieved the desired tolerance in all cases, but also outperformed the other three solvers. For example, in the 'a4a' data set, only SSsNAL and P2PG achieve the predetermined tolerance, but SSsNAL just takes 1 seconds to reach the desired accuracy while P2PG needs about one and half minute

and other two solvers even needs much more times than P2PG. Moreover, in almost all cases, the number of support vectors is much smaller than that of samples, which means that the each subproblems in our SSsNAL method can be solved by semismooth Newton algorithm efficiently because the Newton direction can be obtained just by solving a much smaller linear system. The error rates of the classification on most of the training and testing sets are basically the same for different algorithms. But for the datasets 'madelon', 'w7a', and 'a4a', the error rates of FAPG are worse than those of the other three solvers, because its KKT residuals fail to achieve the given tolerance. Hence, to achieve a satisfactory accuracy of classification, the optimization problem should be solved in an appropriate precision.

Secondly, in Table 3, we present the performances of the four algorithms on the various C-SVC problems (2) with the RBF kernel function. In order to avoid save the $n \times n$ dense kernel matrix Q, which may cause an out of memory exception, we only save a portion columns of Q, denote by $Q_{\mathcal{P}}$, at first. Note that $Q_{\mathcal{P}} \in \mathcal{R}^{n\times p}$, where $p = \min\{n, \lfloor 3.6 \times 10^7/n \rfloor\}$. So the number of entries in $Q_{\mathcal{P}}$ is less than 6000×6000. Then, we compute the rest entries of Q on demand in the each iteration of FAGP and P2PG. We can see LIBSVM fails to produce a reasonably accurate solution for all the test instances and its error rates of the classification are worse than those of the other three solvers. Meanwhile, note that SSsNAL, FAGP and P2PG can solve all the test instances successfully. Nevertheless, in most cases, FAGP and P2PG require much more time than SSsNAL. One can also observe that for large-scale problems, SSsNAL outperforms FAGP and P2PG by a large margin (sometimes up to a factor of $10 \sim 100$). The superior numerical performance of SSsNAL indicates that it is a robust, high-performance solver for large-scale SVMs problems.

5 Conclusion

In this paper, we proposed a highly efficient SSsNAL method for solving the large-scale convex quadratic programming problem generated from the dual problem of SVMs. By leveraging the primal-dual error bound result, the fast local convergence property of the AL method can be guaranteed, and by exploiting the second-order sparsity of the Jacobian when using the SsN method, the algorithm can efficiently solve the problems. Finally, numerical experiments demonstrated the high efficiency and robustness of the SSsNAL method.

References

[1] V. N. Vapnik and A. Lerner, "Pattern recognition using generalized portrait method," *Autom. Remote Control*, vol. 24, pp. 774–780, 1963.

[2] H. Taira and M. Haruno, "Feature selection in SVM text categorization.," *Transactions of Information Processing Society of Japan*, vol. 41, pp. 480–486, 1999.

[3] M. Goudjil, M. Koudil, M. Bedda, and N. Ghoggali, "A novel active learning method using SVM for text classification," *International Journal of Automation and Computing*, no. 3, pp. 1–9, 2018.

[4] R. Azim, W. Rahman, and M. F. Karim, "Bangla hand written character recognition using support vector machine," *International Journal of Engineering Works*, vol. 3, no. 6, pp. 36–46, 2016.

[5] Y. Lin, F. Lv, S. Zhu, and M. Yang, "Large-scale image classification: Fast feature extraction and SVM training," *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 14, pp. 1689–1696, 2011.

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

[7] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *The Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, 1992.

[8] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[9] V. Vapnik, "Statistical learning theory," *Encyclopedia of the Sciences of Learning*, vol. 4, pp. 3185–3189, 1998.

[10] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[11] J. Yin and Q. N. Li, "A semismooth newton method for support vector classification and regression," *Computational Optimization and Applications*, vol. 73, no. 2, pp. 477–508, 2019.

[12] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Neural Networks for Signal Processing VII-Proceedings of the 1997 IEEE Workshop*, pp. 276–285, 1997.

[13] T. Joachims, "Making large-scale SVM learning practical," *Technical Reports*, vol. 8, no. 3, pp. 499–526, 1998.

[14] J. C. Platt, *Fast training of support vector machines using sequential minimal optimization*. MIT Press, 1999.

[15] 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. 1–27, 2011.

[16] R. E. Fan, P. H. Chen, and C. J. Lin, "Working set selection using second order information for training support vector machines," *Journal of Machine Learning Research*, vol. 6, no. 4, pp. 1889–1918, 2005.

[17] P. H. Calamai and J. J. Moré, "Projected gradient methods for linearly constrained problems," *Mathematical Programming*, vol. 39, no. 1, pp. 93–116, 1987.

[18] J. J. Moré and G. Toraldo, "On the solution of large quadratic programming problems with bound constraints," *SIAM Journal on Optimization*, vol. 1, no. 1, pp. 93–113, 1991.

[19] D. di Serafino, G. Toraldo, M. Viola, and J. Barlow, "A two-phase gradient method for quadratic programming problems with a single linear constraint and bounds on the variables," *SIAM Journal on Optimization*, vol. 28, no. 4, pp. 2809–2838, 2018.

[20] N. Ito, A. Takeda, and K. C. Toh, "A unified formulation and fast accelerated proximal gradient method for classification," *Journal of Machine Learning Research*, vol. 18, no. 1, pp. 510–558, 2017.

[21] P. H. Chen, R. E. Fan, and C. J. Lin, "A study on smo-type decomposition methods for support vector machines," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 893–908, 2006.

[22] R. T. Rockafellar, "Augmented Lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of Operations Research*, vol. 1, no. 2, pp. 97–116, 1976.

[23] S. M. Robinson, "Some continuity properties of polyhedral multifunctions," *Mathematical Programming at Oberwolfach*, vol. 14, no. 5, pp. 206–214, 1981.

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

[25] R. Helgason, J. Kennington, and H. Lall, "A polynomially bounded algorithm for a singly constrained quadratic program," *Mathematical Programming*, vol. 18, no. 1, pp. 338–343, 1980.

[26] P. Brucker, "An $O(n)$ algorithm for quadratic knapsack problems," *Operations Research Letters*, vol. 3, no. 3, pp. 163 – 166, 1984.

[27] S. Cosares and D. S. Hochbaum, "Strongly polynomial algorithms for the quadratic transportation problem with a fixed number of sources," *Mathematics of Operations Research*, vol. 19, no. 1, pp. 94–111, 1994.

[28] K. C. Kiwiel, "Variable fixing algorithms for the continuous quadratic knapsack problem," *Journal of Optimization Theory and Applications*, vol. 136, no. 3, pp. 445–458, 2008.

[29] J. Han and D. Sun, "Newton and Quasi-Newton methods for normal maps with polyhedral sets," *Journal of Optimization Theory and Applications*, vol. 94, no. 3, pp. 659–676, 1997.

[30] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis*. Springer, 1998.

[31] A. L. Dontchev and R. T. Rockafellar, *Implicit Functions and Solution Mappings*. World Book Inc, 2013.

[32] Y. J. Zhang, N. Zhang, D. F. Sun, and K. C. Toh, "An efficient Hessian based algorithm for solving large-scale sparse group lasso problems," *Mathematical Programming*, vol. 3, pp. 1–41, 2018.

[33] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[34] J.-B. Hiriart-Urruty, J.-J. Strodiot, and V. H. Nguyen, "Generalized Hessian matrix and second-order optimality conditions for problems with $C^{1,1}$ data," *Applied Mathematics and Optimization*, vol. 11, no. 1, pp. 43–56, 1984.

[35] Y. Nesterov, *Introductory lectures on convex programming: a basic course*. Kluwer Academic Publishers, 2004.

[36] F. J. Luque, "Asymptotic convergence analysis of the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 22, no. 2, pp. 277–293, 1984.

[37] J. M. Danskin, "The theory of max-min, with applications," *SIAM Journal on Applied Mathematics*, vol. 14, no. 4, pp. 641–664, 1966.

[38] X. D. Li, D. F. Sun, and K. C. Toh, "On the efficient computation of a generalized Jacobian of the projector over the Birkhoff polytope," *arXiv preprint arXiv:1702.05934*, 2017.

[39] B. Kummer, "Newton's method for non-differentiable functions," *Advances in Mathematical Optimization*, vol. 45, no. 1988, pp. 114–125, 1988.

[40] L. Q. Qi and J. Sun, "A nonsmooth version of Newton's method," *Mathematical Programming*, vol. 58, no. 1-3, pp. 353–367, 1993.

[41] F. Facchinei and J. S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer, 2003.

[42] G. H. Golub and C. F. Van Loan, *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, 1996.

[43] S. S. Keerthi, O. Chapelle, and D. Decoste, "Building support vector machines with reduced classifier complexity," *Journal of Machine Learning Research*, vol. 7, no. 2006, pp. 1493–1515, 2006.

Data	n_{train}(	n^+_{train},	n^-_{train})	n_{test}(	n^+_{test},	n^-_{test})	q	density
splice	1000(	517,	483)	2175(	1131,	1044)	60	1.000
madelon	2000(	1000,	1000)	600(	300,	300)	500	0.999
ijcnn1	35000(	3415,	31585)	91701(	8712,	82989)	22	0.591
svmguide1	3089(	1089,	2000)	4000(	2000,	2000)	4	0.997
svmguide3	1243(	947,	296)	41(	41,	0)	22	0.805
w1a	2477(	72,	2405)	47272(	1407,	45865)	300	0.039
w2a	3470(	107,	3363)	46279(	1372,	44907)	300	0.039
w3a	4912(	143,	4769)	44837(	1336,	43501)	300	0.039
w4a	7366(	216,	7150)	42383(	1263,	41120)	300	0.039
w5a	9888(	281,	9607)	39861(	1198,	38663)	300	0.039
w6a	17188(	525,	16663)	32551(	954,	31607)	300	0.039
w7a	5103(	740,	4363)	25057(	739,	24318)	300	0.039
australian	690(	307,	383)	*(	*,	*)	14	0.874
breast-cancer	683(	239,	444)	*(	*,	*)	10	1.000
mushrooms	8124(	3916,	4208)	*(	*,	*)	112	0.188
phishing	11055(	6157,	4898)	*(	*,	*)	68	0.441
diabetes	768(	500,	268)	*(	*,	*)	8	0.999
ionosphere	651(	225,	126)	*(	*,	*)	34	0.884
heart	270(	120,	150)	*(	*,	*)	13	0.962
fourclass	862(	307,	555)	*(	*,	*)	2	0.983
colon-cancer	62(	22,	40)	*(	*,	*)	2000	1.000
diabetes	768(	500,	268)	*(	*,	*)	8	0.999
a1a	1605(	395,	1210)	*(	*,	*)	119	0.116
a2a	2265(	572,	1693)	*(	*,	*)	119	0.115
a3a	3185(	773,	2412)	*(	*,	*)	122	0.114
a4a	4781(	1188,	3593)	*(	*,	*)	122	0.114
a6a	11220(	2692,	8528)	*(	*,	*)	122	0.114
a7a	16100(	3918,	12182)	*(	*,	*)	122	0.115

Table 1: The details of the LIBSVM Datasets. '$*$' means that it does not have the corresponding data or the features of the training data and testing data are not equal to each other.

Data	n:q	suppvec	R_{KKT} a|b|c|d	Time a|b|c|d	Err_{train} a|b|c|d	Err_{test} a|b|c|d
splice	1000:60	90	8.6-4 | 2.4-3 | 1.9-3 | 1.53-4	01 | 56 | 35 | 01	15.6 | 15.8 | 16.0 | 16.0	15.4 | 15.4 | 15.6 | 15.2
madelon	2000:500	540	8.0-4 | 2.3-1 | 1.4-4 | 1.9-4	05 | 4:43 | 50 | 05	23.0 | 40.6 | 23.3 | 23.3	42.3 | 49.0 | 42.0 | 42.0
ijcnn1	35000:22	808	9.7-4 | 2.4-2 | 1.2-4 | 1.8-4	03 | 2:3:20 | 07| 16	8.6 | 8.8 | 8.6 | 8.6	8.3 | 8.5 | 8.3 | 8.3
cod-rna	59535:8	914	8.3-4 | 5.9-2 | 1.6-2 | 1.6-4	03 | 4:26:1 | 1:43 | 29	6.1 | 6.2 | 6.3 | 6.1	4.8 | 5.4 | 5.0 | 4.8
svmguide1	3089:4	49	9.6-4 | 1.0-3 | 1.9-4 | 3.9-5	02 | 21 | 1:13 | 00	4.5 | 4.5 | 4.5 | 4.5	5.2 | 5.2 | 5.2 | 5.2
svmguide3	1243:22	87	8.5-4 | 9.7-4 | 1.1-4 | 5.3-5	00 | 1:15 | 03 | 3:29	17.2 | 17.1 | 17.2 | 17.2	0.0 | 0.0| 0.0 | 0.0
w1a	2477:300	368	9.6-4 | 9.9-4 | 2.0-4 | 4.7-4	00 | 07 | 02 | 13:20	0.7 | 0.7 | 0.7 | 0.7	2.5 | 2.5 | 2.5 | 2.5
w2a	3470:300	180	7.6-4 | 4.8-3 | 2.3-4 | 3.7-4	00 | 1:06 | 1:08 | 27:08	0.7 | 0.7 | 0.7 | 0.7	2.4 | 2.4 | 2.5 | 2.5
w3a	4912:300	216	8.9-4 | 9.8-4 | 2.4-4 | 9.1-4	01 | 1:34 | 2:30 | 50:28	0.8 | 0.8 | 0.8 | 0.8	2.0 | 2.0 | 2.0 | 2.0
w4a	7366:300	237	9.1-4 | 1.4-1 | 2.1-4 | 7.8-4	01 | 7:59 | 05 | 1:53:32	0.9 | 1.2 | 0.9 | 0.9	1.7 | 2.0 | 1.7 | 1.7
w5a	9888:300	295	8.8-4 | 1.1-2 | 2.2-4 | 1.9-2	01 | 17:41 | 08 | 3:10:10	1.0 | 1.0 | 1.0 | 1.0	1.7 | 1.7 | 1.7 | 1.7
w6a	17188:300	429	8.8-4 | 3.4-2 | 2.0-4 | 9.6-3	01 | 52:19 | 13 | 10:46:19	1.2 | 1.2 | 1.2 | 1.2	1.3 | 1.4 | 1.3 | 1.3
w7a	5103:300	650	9.9-4 | 4.5-1 | 2.5-4 | 9.4-3	00 | 4:59 | 1:17 | 55:55	3.9 | 11.0 | 3.9 | 3.9	3.2 | 10.3 | 3.2 | 3.3
australian	690:14	111	9.1-4 | 9.9-4 | 1.7-4 | 1.4-4	00 | 05 | 02 | 34	14.2 | 14.2 | 14.2 | 14.1	14.7 | 14.7 | 14.7 | 15.2
breast-cancer	683:10	13	8.0-4 | 8.7-4 | 2.2-4 | 1.7-4	00 | 00 | 00 | 21	2.7 | 2.7 | 2.7 | 2.8	3.6 | 3.6 | 3.6 | 2.6
mushrooms	8124:112	451	5.4-4 | 9.9-4 | 2.5-2 | 1.3-2	00 | 1:33 | 00 | 43:15	0.0 | 0.0 | 0.0 | 0.0	0.0 | 0.0 | 0.0 | 0.0
phishing	11055:68	194	8.6-4 | 4.2-1 | 2.2-4 | 2.1-2	01 | 23:35 | 13 | 2:46:08	5.9 | 12.2 | 5.9 | 6.0	6.2 | 12.6 | 6.3 | 6.2
diabetes	768:8	30	8.6-4 | 9.5-4 | 7.2-5 | 5.0-5	00 | 01 | 05 | 46	22.7 | 22.6 | 22.7 | 21.7	21.7 | 21.8 | 21.7 | 24.3
ionosphere	351:34	32	7.8-4 | 9.6-4 | 1.7-4 | 2.1-4	00 | 00 | 00 | 09	4.8 | 4.8 | 4.9 | 5.7	12.3 | 12.3 | 12.3 | 12.3
heart	270:13	18	8.0-4 | 9.3-4 | 8.6-5 | 1.3-4	00 | 00 | 00 | 06	13.8 | 14.0 | 13.7 | 14.1	16.5 | 16.9 | 16.5 | 16.7
fourclass	862:2	33	8.4-4 | 9.2-4 | 1.1-4 | 1.4-4	00 | 02 | 04 | 44	23.0 | 23.0 | 23.0 | 22.5	22.7 | 22.6 | 22.7 | 24.5
colon-cancer	62:2000	30	4.2-4 | 8.7-4 | 4.0-3 | 1.2-2	00 | 00 | 00 | 00	0.0 | 0.0 | 0.0 | 0.0	10.8 | 10.8 | 10.8 | 13.1
diabetes	786:8	30	8.8-4 | 9.3-4 | 8.0-5 | 5.0-5	00 | 01 | 04 | 44	22.5 | 22.5 | 22.4 | 22.1	22.6 | 22.9 | 22.6 | 23.2
a1a	1605:119	102	8.6-4 | 4.8-2 | 1.4-4 | 5.3-4	00 | 1:06 | 08 | 4:02	12.7 | 13.9 | 12.7 | 13.2	17.5 | 17.5 | 17.3 | 16.9
a2a	2265:119	131	8.7-4 | 6.8-2 | 1.3-4 | 1.3-3	00 | 2:37 | 17 | 7:57	15.3 | 16.6 | 15.3 | 15.3	18.0 | 18.9 | 18.0 | 17.4
a3a	3185:122	149	7.9-4 | 6.3-2 | 1.4-4 | 3.0-3	00 | 5:20 | 38 | 15:37	14.6 | 16.5 | 14.5 | 15.5	16.8 | 18.3 | 16.8 | 17.4
a4a	4781:122	193	8.8-4 | 2.5-1 | 1.3-4 | 1.0-2	01 | 12:55 | 1:39 | 35:56	14.7 | 23.6| 14.7 | 15.0	17.1 | 25.1 | 17.1 | 16.2
a6a	11220:122	354	9.7-4 | 1.4-1 | 1.4-4 | 2.9-1	01 | 32:10 | 06 | 47:59	15.1 | 17.1 | 15.2 | 37.8	15.4 | 17.3 | 15.5 | 38.2
a7a	16100:122	509	7.3-4 | 5.5-2 | 1.4-4 | 3.0-1	03 | 49:55 | 11 | 1:38:40	15.2 | 16.0 | 15.2 | 47.6	15.3 | 16.0 | 15.4 | 48.7

Table 2: Comparisons of the SSsNAL, FAPG, P2GP, and LINBSVM for the C-SVC problems (2) with a linear kernel function. In the table, "a"=SSsNAL, "b"=FAGP, "c"=P2PG, and "d"=LINBSVM.

Data	n:q	suppvec	R_{KKT} a|b|c|d	Time a|b|c|d	Err_{train} a|b|c|d	Err_{test} a|b|c|d
splice	1000:60	82	6.4-4 | 9.4-4 | 7.8-5 | 9.7-1	01 | 01 | 00 | 37	15.3 | 15.0 | 14.9 | 20.0	14.7 | 14.3 | 14.6 | 16.9
madelon	2000:500	345	5.6-4 | 9.2-4 | 8.3-5 | 9.4-1	05 | 02 | 01 | 3:04	25.6 | 25.2 | 25.3 | 35.9	41.2 | 41.5 | 41.3 | 40.5
liver-disorders	145:5	26	9.7-4 | 9.2-4 | 7.9-5 | 5.0-1	00 | 00 | 00 | 02	35.9 | 35.9 | 35.9 | 37.9	50.5 | 50.5 | 50.5 | 50.0
leu	38:7129	38	6.2-4 | 8.8-4 | 1.0-4 | 2.7-1	00 | 00 | 00 | 00	0.0 | 0.0 | 0.0 | 0.0	41.2 | 41.2 | 41.2 | 38.2
gisette	6000:5000	3175	6.1-5 | 7.5-4 | 2.4-3 | 2.6-0	09 | 27 | 13 | 21:59	0.0 | 0.0 | 0.0 | 12.2	5.6 | 5.6 | 5.5 | 48.5
svmguide1	3089:4	140	4.5-4 | 8.9-4 | 1.4-4 | 4.0-1	04 | 02 | 49 | 19:44	7.8 | 7.7 | 100.0 | 35.3	9.5 | 9.4 | 100.0 | 50.0
svmguide3	1243:22	240	6.8-4 | 8.8-4 | 6.1-5 | 9.4-1	01 | 02 | 00 | 1:35	23.1 | 23.1 | 23.2 | 25.2	34.1 | 34.1 | 31.7 | 58.5
w1a	2477:300	198	4.0-4 | 9.9-4 | 1.4-4 | 3.1-1	02 | 05 | 02 | 4:41	2.1 | 2.1 | 2.1 | 2.1	2.5 | 2.5 | 2.5 | 2.6
w2a	3470:300	206	3.6-4 | 9.6-4 | 1.5-4 | 2.8-1	03 | 08 | 02 | 9:06	1.8 | 1.9 | 1.9 | 13.1	2.1 | 2.1 | 2.1 | 13.7
w3a	4912:300	205	7.5-4 | 9.8-4 | 5.6-5 | 3.1-1	07 | 13 | 04 | 18:43	1.8 | 1.8 | 1.8 | 33.1	2.2 | 2.2 | 2.2 | 33.3
w4a	7366:300	323	8.1-4 | 1.4-1 | 4.2-4 | 3.4-1	57 | 2:14 | 1:28:49 | 10:38	1.6 | 1.6 | 1.6 | 41.4	2.0 | 2.0 | 2.0| 41.9
w5a	9888:300	363	9.4-4 | 9.9-4 | 1.2-4 | 4.2-1	2:43 | 8:44 | 4:27:52 | 18:59	1.6 | 1.6 | 1.6 | 53.5	1.9 | 1.9 | 1.9 | 53.5
w6a	17188:300	686	9.1-4 | 9.9-4 | 1.4-4 | 4.4-1	8:24 | 3:13:42 | 21:22:58 | 53:09	1.6 | 1.6 | 1.6 | 53.9	1.6 | 1.6 | 1.6 | 53.9
w7a	5103:300	681	2.5-4 | 9.9-4 | 1.5-4 | 5.6-1	12 | 26 | 05 | 02	5.4 | 5.4 | 5.4 | 47.7	2.6 | 2.6 | 2.6 | 65.3
australian	690:14	276	9.4-4 | 9.4-4 | 1.5-4 | 7.2-1	00 | 00 | 00 | 20	14.1 | 14.1 | 14.1 | 13.2	16.0 | 16.0 | 16.0 | 15.4
breast-cancer	683:10	17	8.0-4 | 8.0-4 | 1.1-4 | 7.0-1	00 | 00 | 00 | 12	3.0 | 3.0 | 3.0 | 3.8	3.4 | 3.4 | 3.4 | 4.3
mushrooms	8124:112	397	6.8-4 | 9.9-4 | 7.4-4 | 2.3-0	09 | 1:14 | 10:16 | 21:37	0.0 | 0.0 | 0.0 | 11.1	0.1 | 0.1 | 0.1 | 10.5
phishing	11055:68	250	6.8-4 | 9.5-4 | 1.9-4 | 1.5-0	48 | 3:12 | 28:52 | 1:22:48	5.2 | 5.2 | 5.2 | 8.0	5.4 | 5.4 | 5.4 | 8.0
diabetes	768:8	49	7.8-4 | 8.6-4 | 8.4-5 | 9.0-1	00 | 00 | 00 | 27	31.7 | 31.7 | 31.7 | 34.8	31.4 | 31.6 | 31.5 | 35.1
ionosphere	351:34	32	8.9-4 | 9.1-4 | 7.6-5 | 7.6-1	00 | 00 | 00 | 05	10.7 | 10.7 | 10.7 | 14.4	11.0 | 11.0 | 11.0 | 13.5
heart	270:13	14	8.2-4 | 8.1-4 | 5.4-5 | 6.8-1	00 | 00 | 00 | 03	14.7 | 14.7 | 14.7 | 16.4	17.6 | 17.6 | 18.0 | 18.3
fourclass	862:2	28	7.2-4 | 8.8-4 | 5.6-5 | 1.5-0	00 | 00 | 00 | 30	21.6 | 21.6 | 60.8 | 35.6	22.4 | 22.4 | 61.8 | 35.8
colon-cancer	62:2000	39	7.6-4 | 8.1-4 | 2.1-4 | 6.9-1	00 | 00 | 00 | 00	0.0 | 0.0 | 0.0 | 7.1	20.0 | 20.0 | 20.0 | 20.8
diabetes	768:8	47	8.0-4 | 8.9-4 | 7.7-5 | 9.0-1	00 | 00 | 00 | 2:09	31.3 | 31.2 | 45.1 | 34.8	32.1 | 32.2 | 45.9 | 35.0
a1a	1605:119	89	6.1-4 | 9.3-4 | 1.1-4 | 1.0-0	00 | 01 | 00 | 1:03	14.7 | 14.6 | 14.7 | 27.9	17.2 | 17.1 | 17.1 | 30.3
a2a	2265:119	109	5.7-4 | 9.1-4 | 1.0-4 | 1.2-0	03 | 02 | 01 | 2:04	16.2 | 16.2 | 16.2 | 28.7	17.6 | 17.6 | 17.5 | 28.5
a3a	3185:122	127	8.0-4 | 9.5-4 | 1.2-4 | 1.3-0	06 | 04 | 01 | 4:48	15.1 | 15.1 | 15.1 | 28.9	16.5 | 16.5 | 16.6 | 27.7
a4a	4781:122	182	8.4-4 | 9.4-4 | 1.1-4 | 1.4-0	15 | 11 | 02 | 9:53	15.0 | 15.0 | 15.0 | 28.6	16.4 | 16.4 | 16.4 | 29.3
a6a	11220:122	360	8.6-4 | 9.5-4 | 1.1-4 | 1.3-0	2:15 | 6:43 | 57:21 | 15:14	15.1 | 15.1 | 15.1 | 28.8	15.9 | 16.0 | 15.1 | 28.5
a7a	16100:122	506	8.9-4 | 9.7-4 | 1.1-4 | 1.3-0	14:56 | 48:30 | 3:27:15 | 30:30	15.2 | 15.2 | 15.1 | 27.8	15.1 | 15.2 | 15.2 | 28.3

Table 3: Comparisons of the SSsNAL, FAPG, P2GP, and LINBSVM for the standard C-SVM problem (2) with the RBF kernel function. In the table, "a"=SSsNAL, "b"=FAGP, "c"=P2PG, and "d"=LINBSVM.