Density-Convoluted Support Vector Machines for High-Dimensional Classification

Boxiang Wang, Le Zhou, Yuwen Gu, and Hui Zou

***Abstract*— The support vector machine (SVM) is a popular classification method which enjoys good performance in many real applications. The SVM can be viewed as a penalized minimization problem in which the objective function is the expectation of hinge loss function with respect to the standard non-smooth empirical measure corresponding to the true underlying measure. We further extend this viewpoint and propose a smoothed SVM by substituting a kernel density estimator for the measure in the expectation calculation. The resulting method is called density convoluted support vector machine (DCSVM). We argue that the DCSVM is particularly more interesting than the standard SVM in the context of high-dimensional classification. We systematically study the rate of convergence of the elastic-net penalized DCSVM under general random design setting. We further develop novel efficient algorithm for computing elastic-net penalized DCSVM. Simulation studies and ten benchmark datasets are used to demonstrate the superior classification performance of elastic-net DCSVM over other competitors, and it is demonstrated in these numerical studies that the computation of DCSVM can be more than 100 times faster than that of the SVM.**

***Index Terms*— Classification, ultra-high dimension, DCSVM, support vector machines, kernel density smoother.**

I. Introduction

DUE to the advanced technology for data collection over the past decades, there has been a surge of data complexity in many research fields such as genomics, genetics, and finance, among others. Consequently, it is very common for the number of predictors in the dataset to be far larger than the number of observations [6]. For example, in genomics it is crucial to build a classifier for the purpose of disease diagnosis, with thousands of candidate genes at hand but only tens of instances available for study. Such high dimensionality in data makes traditional classification methods infeasible and poses new challenges from both theoretical and computational perspectives.

One method for performing high dimensional classification is the penalized large margin classifier. The standard support vector machine (SVM), initially proposed and investigated in [4] and [29], has an objective equal to hinge loss plus an ℓ_2 penalty. It is also referred to as ℓ_2-norm SVM. When the dimension greatly exceeds the sample size and there are many noisy features in the predictor set, it has been shown that it is more beneficial to use a sparse penalty such as the ℓ_1-norm penalty (a.k.a. the lasso) to replace the ℓ_2-norm penalty in order to perform classification and variable selection simultaneously in high dimensional setting [33], [37]. Suppose the training data consists of n observations $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}} \in \mathbb{R}^p$ are predictors and $y_i \in \{-1, 1\}$ is the class label for the ith subject. Consider the ℓ_1 norm SVM for example. It can be written as

$$\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{n} \sum_{i=1}^{n} L\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \beta_0)\big) + \lambda \|\boldsymbol{\beta}\|_1, \tag{I.1}$$

where $L(u) = (1-u)_+$ is the hinge loss. Just like in lasso regression, the ℓ_1 penalty induces sparsity in the solution and is thus capable of removing irrelevant features. More recently, [23] investigated the rate of convergence of the ℓ_1-norm SVM and an error bound.

The sparse penalized SVM can be computationally intensive especially when the number of predictors is huge in the dataset, owing to the non-differentiable loss function part. It is known that penalized problem in high dimensions with a smooth loss function can be efficiently computed by cyclical coordinate descent algorithm [9]. Nevertheless, the SVM is based on the non-differentiable hinge loss, which means that there is no convergence guarantee if one uses cyclical coordinate descent to solve the SVM. In principle, coordinate descent may not give the right solution due to the non-differentiability of the objective function [20], [27]. A similar problem under regression context is the quantile regression, in which the check loss is not differentiable [7]. The typical method of solving quantile regression is the interior point algorithm. Since ℓ_1-norm SVM can be transformed into linear programming, one may also consider interior point algorithm for solving it. However, interior point algorithm may not scale well with high dimensional input and thus is not suitable for solving SVM in high dimensions.

Manuscript received 15 December 2021; revised 19 August 2022; accepted 13 November 2022. Date of publication 17 November 2022; date of current version 17 March 2023. The work of Hui Zou was supported in part by NSF under Grant 1915842 and Grant 2015120. *(Corresponding author: Hui Zou.)*

Boxiang Wang is with the Department of Statistics and Actuarial Science, The University of Iowa, Iowa City, IA 52242 USA.

Le Zhou is with the Department of Mathematics, Hong Kong Baptist University, Hong Kong, SAR.

Yuwen Gu is with the Department of Statistics, University of Connecticut, Storrs, CT 06269 USA.

Hui Zou is with the School of Statistics, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: zouxx019@umn.edu).

Communicated by S. Jalali, Associate Editor for Machine Learning and Statistics.

Digital Object Identifier 10.1109/TIT.2022.3222767

Recently, [8] studied an interesting smoothing technique for solving quantile regression with statistical guarantees. Later, [25] further studied the smoothing quantile regression under high dimensional settings and showed that the statistical property of quantile regression is maintained after smoothing. Motivated by their work, we develop a smooth version of SVM from statistical perspective, as opposed to trying to solve it exactly. Consider the first term in (I.1)

$$\frac{1}{n}\sum_{i=1}^{n} L\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)\big), \tag{I.2}$$

which is non-smooth. If we could replace it by some smooth loss such that the resulting estimator has nice theoretical properties, then we should focus on solving the smooth problem instead of the original problem. In fact, one may view (I.2) as the expectation of the hinge loss function with respect to the empirical measure assigning $\frac{1}{n}$ probability mass to each $y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0), i=1,\dots,n$. The empirical measure is viewed as an estimator for the true distribution of the random variable $y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)$. Clearly, if we estimate the true distribution by using a smoothed kernel density estimator, see [e.g., Chapter 6 of 12], then we can take the expectation of the hinge loss function with respect to the distribution determined by such smoothed kernel density estimator. This leads us to a new objective function which we can use to replace the original objective function in (I.2). The resulting estimator is named as *density convoluted support vector machine (DCSVM)*, since the kernel density estimator has a convolution interpretation. We further study the following general form of penalized DCSVM in high dimensions using the elastic-net penalty [38]. The resulting estimator is called elastic-net DCSVM, which involves both ℓ_1-DCSVM and ℓ_2-DCSVM as special cases. By its convexity and smoothness, elastic-net DCSVM can be efficiently solved by using the generalized coordinate descent algorithm [35].

In this paper, we first study the theoretical properties of the elastic-net DCSVM. We give the convergence rate of the elastic-net DCSVM under the general random design setting. Furthermore, we developed a novel efficient algorithm for computing elastic-net DCSVM. We used simulation studies and ten benchmark datasets to demonstrate the superior classification performance of elastic-net DCSVM over its competitors, and the computation speed of DCSVM can be two orders of magnitude faster than that of SVM.

II. Density-Convoluted SVM

A. Notation and Definitions

We first introduce some notation that is used throughout the paper. For an arbitrary index set $\mathbf{A}\subset\{1,\dots,p\}$, any vector $\mathbf{c}=(c_1,\dots,c_p)$ and any $n\times p$ matrix $\mathbf{U}$, let $\mathbf{c}_{\mathbf{A}}=(c_i, i\in\mathbf{A})$, and let $\mathbf{U}_{\mathbf{A}}$ be the submatrix with columns of $\mathbf{U}$ whose indices are in $\mathbf{A}$. The complement of an index set $\mathbf{A}$ is denoted as $\mathbf{A}^{\mathrm{c}}=\{1,\dots,p\}\setminus\mathbf{A}$. For any finite set $\mathbf{B}$, let $|\mathbf{B}|$ be the number of elements in $\mathbf{B}$. For a vector $\mathbf{c}\in\mathbb{R}^p$ and $q\in[1,\infty)$, let $\|\mathbf{c}\|_q=(\sum_{j=1}^p|c_j|^q)^{\frac{1}{q}}$ be its ℓ_q norm, let $\|\mathbf{c}\|_\infty$ (or $\|\mathbf{c}\|_{\max}$) $=\max_j|c_j|$ be its ℓ_∞ norm, and let $\|\mathbf{c}\|_{\min}=\min_j|c_j|$ be its minimum absolute value. For a matrix $\mathbf{M}$, let $\lambda_{\min}(\mathbf{M})$ and $\lambda_{\max}(\mathbf{M})$ be its eigenvalue with smallest absolute value and largest absolute value, respectively. This is the common notation for eigenvalues of a matrix, and $\lambda_{\min}$, $\lambda_{\max}$ should not be confused with the penalization parameter used in a penalty function. For any matrix $\mathbf{G}$, let $\|\mathbf{G}\|=\sqrt{\lambda_{\max}(\mathbf{G}^{\mathrm{T}}\mathbf{G})}$ be its spectral norm. In particular, for a vector $\mathbf{c}$, $\|\mathbf{c}\|=\|\mathbf{c}\|_2$. For $a,b\in\mathbb{R}$, let $a\wedge b=\min\{a,b\}$ and $a\vee b=\max\{a,b\}$. For a sequence $\{a_n\}$ and another nonnegative sequence $\{b_n\}$, we write $a_n=O(b_n)$ if there exists a constant $c>0$ such that $|a_n|\le cb_n$ for all $n\ge1$. Also, we use $a_n=o(b_n)$, or $a_n\ll b_n$, to represent $\lim_{n\to\infty}\frac{a_n}{b_n}=0$. We write $b_n\gg a_n$ if $a_n\ll b_n$. Let $(\Omega,\mathcal{G},\mathrm{P})$ be a probability space on which all the random variables that appear in this paper are defined. Let $\mathbb{E}[\cdot]$ be the expectation corresponding to the probability measure P. Let $\psi:[0,\infty)\to[0,\infty]$ be a nondecreasing, convex function with $\psi(0)=0$, then we denote $\|Z\|_\psi=\inf\{t>0:\mathbb{E}[\psi(\frac{|Z|}{t})]\le1\}$ as the ψ-Orlicz norm for any random variable Z. In particular, if $p\ge1$, let $\psi_p(x):=\mathrm{e}^{x^p}-1$ which is a nondecreasing convex function with $\psi_p(0)=0$, then we denote its corresponding Orlicz norm as $\|Z\|_{\psi_p}=\inf\{t>0:\mathbb{E}[\mathrm{e}^{\frac{|Z|^p}{t^p}}]\le2\}$ where Z is any random variable. For a sequence of random variables $\{Z_n\}_{n\ge1}$, we write $Z_n=O_p(1)$ if $\lim_{M\to\infty}\limsup_{n\to\infty}\mathrm{P}(|Z_n|>M)=0$, and we write $Z_n=o_p(1)$ if $\lim_{n\to\infty}\mathrm{P}(|Z_n|>\epsilon)=0,\forall\epsilon>0$. For two sequences of random variables Z_n and Z_n', we write $Z_n=O_p(Z_n')$ if $\frac{Z_n}{Z_n'}=O_p(1)$, and we write $Z_n=o_p(Z_n')$ if $\frac{Z_n}{Z_n'}=o_p(1)$.

B. Density-Convoluted SVM

We use $\mathbf{X}=(\mathbf{X}_1,\dots,\mathbf{X}_p)$ to denote the design matrix, where $\mathbf{X}_j=(x_{1j},\dots,x_{nj})^{\mathrm{T}}$ contains observations for the jth variable, and use $\mathbf{y}=(y_1,\dots,y_n)^{\mathrm{T}}$ to represent the response vector. We focus on the general case where the observed data $\{(y_i,\mathbf{x}_i)\}_{i=1}^n$ are i.i.d. samples from the distribution of a random vector $(y,\mathbf{x})$. Let the jth component of the random vector $\mathbf{x}$ be denoted as x_j. Meanwhile, let $\tilde{\mathbf{x}}=(1,\mathbf{x}^{\mathrm{T}})^{\mathrm{T}}$ and $\tilde{\mathbf{x}}_i=(1,\mathbf{x}_i^{\mathrm{T}})^{\mathrm{T}}$, $i=1,\dots,n$. To perform the classification task, the support vector machine [SVM, 29] seeks a separating hyperplane $\{\mathbf{x}:\beta_0+\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}=0\}$ where

$$\begin{aligned}&\min_{\beta_0,\boldsymbol{\beta},\xi_i}\ \frac{1}{2}\|\boldsymbol{\beta}\|_2^2\\ &\text{subject to}\quad y_i\left(\beta_0+\mathbf{x}_i^\top\boldsymbol{\beta}\right)\ge1-\xi_i,\xi_i\ge0,\sum_{i=1}^n\xi_i\le c.\end{aligned} \tag{II.1}$$

It is well known that the above problem can be equivalently formulated as a penalized empirical risk minimization problem:

$$\min_{\beta_0,\boldsymbol{\beta}}\frac{1}{n}\sum_{i=1}^n L\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)\big)+\lambda_0\|\boldsymbol{\beta}\|_2^2, \tag{II.2}$$

where $L(u)=(1-u)_+=\max\{1-u,0\}$ is known as the SVM hinge loss and $\lambda_0>0$ is a tuning parameter that is one-to-one correspondent to the constant c in problem (II.1); a reference can be seen in Chapter 12 of [12] for example.

Some recent developments of the SVM include [3], [13], and [17] to name a few.

Let us consider the population version of risk appearing in (II.2). If we define new random variable $U = y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)$ and let $F(u;\boldsymbol{\beta},\beta_0)$ be its cumulative distribution function (cdf), then the population risk is written as

$$\mathbb{E}[L\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)\big)] = \int_{-\infty}^{\infty} L(t)\mathrm{d}F(t;\boldsymbol{\beta},\beta_0).$$

The unpenalized objective function in (II.2) can be further viewed as $\int_{-\infty}^{\infty} L(t)\mathrm{d}\hat{F}(t;\boldsymbol{\beta},\beta_0)$, where $\hat{F}(t;\boldsymbol{\beta},\beta_0) = \frac{1}{n}\sum_{i=1}^{n} I_{\{y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)\le t\}}$ is the empirical cdf based on i.i.d. realizations of U, where I is the indicator function. The usage of the discontinuous empirical cdf here makes the objective in (II.2) to have the same degree of smoothness as the hinge loss $L(\cdot)$, i,e. continuous but nondifferentiable. This has motivated us to consider an alternative estimator for the cdf. If we use an estimator $\tilde{F}(\cdot\,;\boldsymbol{\beta},\beta_0)$ that is smooth enough, the $\int_{-\infty}^{\infty} L(t)\mathrm{d}\tilde{F}(t;\boldsymbol{\beta},\beta_0)$ shall lead us towards a new objective which is differentiable to certain degrees.

In particular, we consider the cdf from the kernel density estimator

$$\tilde{F}(t;\boldsymbol{\beta},\beta_0) = \int_{-\infty}^{t} \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{u - y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)}{h}\Big)\mathrm{d}u,$$

where $K:\mathbb{R}\to[0,\infty)$ is a smooth kernel function satisfying $K(-u) = K(u), \forall u \in \mathbb{R}$, $\int_{-\infty}^{\infty} K(t)\mathrm{d}t = 1$ and $\int_{-\infty}^{\infty} |t|K(t)\mathrm{d}t < \infty$, and $h > 0$ is the bandwidth parameter to be tuned. Replacing $\hat{F}$ by $\tilde{F}$ gives the new objective function,

$$\begin{aligned}&\int_{-\infty}^{\infty} L(t)\mathrm{d}\tilde{F}(t;\boldsymbol{\beta},\beta_0)\\ &= \frac{1}{n}\sum_{i=1}^{n}\int_{-\infty}^{\infty} L(t)\frac{1}{h}K\Big(\frac{t - y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)}{h}\Big)\mathrm{d}t\\ &\triangleq \frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)\big)\end{aligned}$$

where $L_h(t) = \int_{-\infty}^{\infty}(1-u)_+\frac{1}{h}K\left(\frac{u-t}{h}\right)du$. Note that $L_h(\cdot)$ is a convex function that is at least second order differentiable. Also, it satisfies the relation $L_h = L * K_h$ where $K_h(u) = \frac{1}{h}K(\frac{u}{h})$ and the operation "$*$" stands for convolution.

As such, with the penalty term $\lambda_0\|\boldsymbol{\beta}\|_2^2$, we obtain

$$\min_{\beta_0,\boldsymbol{\beta}} \sum_{i=1}^{n} L_h(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)) + \lambda_0\|\boldsymbol{\beta}\|_2^2,$$

We treat the classifier arisen from the above problem as a new classifier and coin it the density-convoluted SVM (DCSVM). The bandwidth h is used to index the new classifier.

As discussed above, DCSVM originates from a statistical view of the SVM, while it shows merit from the computational perspective as it overcomes the non-differentiability of the original SVM problem. Smoothing a non-differentiable problem through convolution can be traced back to the idea of *mollification* [10] and has also been studied in the optimization community, for example, [1] and [24]. The method was recently adopted to smooth the quantile regression by [14], [8], and [25].

In this work, we focus on two most popular kernel functions, Gaussian kernel and Epanechnikov kernel in DCSVM, and we denote the corresponding convoluted loss function by $L_h^G(v)$ and $L_h^E(v)$, respectively.

For the Gaussian kernel $K(u) = \frac{1}{\sqrt{2\pi}}\exp\{-u^2/2\}$, one can show that

$$L_h^G(v) = (1-v)\Phi\left(\frac{1-v}{h}\right) + \frac{h}{\sqrt{2\pi}}\exp\left\{-\frac{(1-v)^2}{2h^2}\right\},$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

For the Epanechnikov kernel, namely $K(u) = \frac{3}{4}(1-u^2)I_{\{-1\le u\le 1\}}$,

$$L_h^E(v) = \begin{cases} 1-v, & v \le 1-h,\\ \dfrac{(1-v+h)^3(3h-(1-v))}{16h^3}, & 1-h < v \le 1+h,\\ 0, & v \ge 1+h.\end{cases}$$

The top row of Figure 1 depicts the DCSVM losses with Gaussian kernel and Epanechnikov kernel.

Intuitively, h should be small such that the density convoluted support vector machine is very close to the support vector machine. According to density estimator, the optimal rate for h is $O(n^{-1/5})$. So, we adopt $h = Cn^{-1/5}$ in our implementation, where C is some numerical constant within the range $(0.25, 3)$.

C. Sparse Density-Convoluted SVM

Let $(\beta_0^*,\boldsymbol{\beta}^*) = \mathrm{argmin}_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^p}\,\mathbb{E}\big[L_h\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)\big)\big]$. In high dimensions, we consider designing the estimator under a sparsity assumption that $\boldsymbol{\beta}^*$ has many zero components. Let $\mathbb{A} = \{j : \beta_j^* \ne 0, 1 \le j \le p\}$ be the support set of $\boldsymbol{\beta}^*$, i.e., the set of indices of the important covariates. Let $s = |\mathbb{A}|$. Throughout this paper, we allow $p = p_n$ and $s = s_n$ to diverge with n, and we assume $s_n \ge 1$ and p_n goes to infinity as n goes to infinity. For convenience, we still use p and s to represent these quantities since no confusion is caused. In ultra-high dimensions, the dimension p is allowed to increase exponentially with the sample size n. We also assume that s is relatively of smaller order compared to n, which is necessary for the existence of a consistent estimator.

To perform the classification for high-dimensional data, we present sparse DCSVM with an elastic-net penalty,

$$\begin{aligned}(\hat{\beta}_0,\hat{\boldsymbol{\beta}}) := \underset{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^p}{\mathrm{argmin}}\ &\frac{1}{n}\sum_{i=1}^{n} L_h(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0))\\ &+ \lambda_0\|\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1. \qquad \text{(II.3)}\end{aligned}$$

The ℓ_1-penalty is used to induce sparsity in the estimator. We also consider the following version of sparse DCSVM with only an ℓ_1-penalty term:

$$(\tilde{\beta}_0,\tilde{\boldsymbol{\beta}}) := \underset{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^p}{\mathrm{argmin}}\ \frac{1}{n}\sum_{i=1}^{n} L_h(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)) + \lambda\|\boldsymbol{\beta}\|_1. \qquad \text{(II.4)}$$

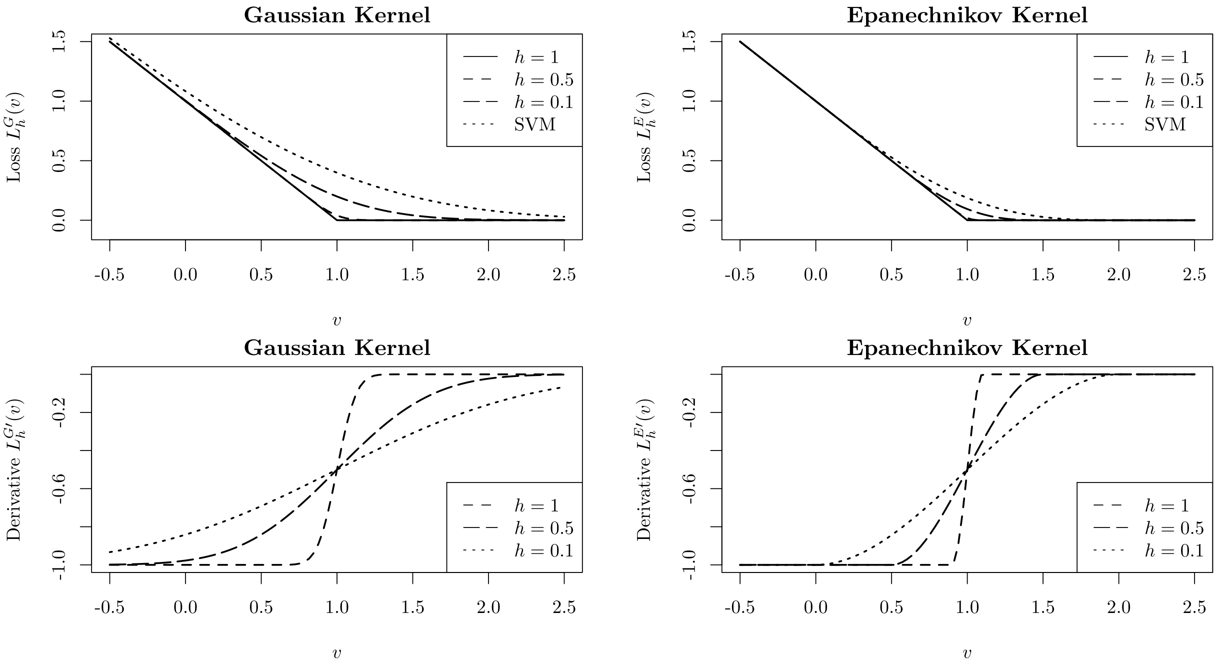

Fig. 1. Top row: plots of $L_h^G(v)$ and $L_h^E(v)$, the density-convoluted SVM loss functions with Gaussian kernel (left) and Epanechnikov kernels (right). Bottom row: plots of the first-order derivatives, $L_h^{G\prime}(v)$ and $L_h^{E\prime}(v)$.

Borrowing the commonly used terminologies for different penalties in high dimensional literature, we refer to the estimator in (II.3) as elastic-net DCSVM, and refer to the estimator in (II.4) as lasso DCSVM. Note that the lasso DCSVM is a special case of elastic-net DCSVM with $\lambda_0 = 0$.

III. Theoretical Studies

We now state the assumptions needed to establish our theoretical results. We first impose the following conditions on the random design.

Assumption 1: The data $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$ and $(y, \mathbf{x})$ are independent and identically distributed on $\mathbb{R} \times \mathbb{R}^p$, where $\mathbf{x}$ is a zero-mean sub-exponential random vector, i.e. $\mathbb{E}[\mathbf{x}] = \mathbf{0}$, and there exists a constant $m_0 > 0$ such that

$$\sup_{\mathbf{a}\in\mathbb{R}^p:\|\mathbf{a}\|_2\le 1} \|\mathbf{a}^{\mathrm{T}}\mathbf{x}\|_{\psi_1} \le m_0.$$

By definition of Orlicz norm and Markov's inequality, this further implies

$$\sup_{\mathbf{a}\in\mathbb{R}^p:\|\mathbf{a}\|_2\le 1} \mathrm{P}(|\mathbf{a}^{\mathrm{T}}\mathbf{x}| > t) \le 2\mathrm{e}^{-\frac{t}{m_0}}, \forall t \ge 0.$$

For any index set $\mathbf{A} \subset \{1, \ldots, p\}$, consider the cone $\mathcal{S}_{\mathbf{A}} := \{(\delta, \mathbf{u}) \in \mathbb{R} \times \mathbb{R}^p : \|\mathbf{u}_{\mathbf{A}^c}\|_1 \le 3\|\mathbf{u}_{\mathbf{A}}\|_1 + |\delta|\}$. Such type of cone has been widely considered in literature on high dimensional statistics. Meanwhile, let $I(\beta_0, \boldsymbol{\beta}) := \mathbb{E}[L_h''\big(y(\beta_0 + \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})\big)\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathrm{T}}]$ be the Hessian matrix of the population loss, or information matrix. We impose the following condition on the information.

Assumption 2: There exists a constant $\rho > 0$ such that

$$\min_{(\delta,\mathbf{u})\in\mathcal{S}_{\mathbb{A}}:\delta^2+\|\mathbf{u}\|_2^2=O(\frac{s\log p}{n})} \lambda_{\min}\big(I(\beta_0^* + \delta, \boldsymbol{\beta}^* + \mathbf{u})\big) \ge \rho$$

for large enough n.

Assumption 1 is a general setting in the random design, which relaxes the classical condition that the components of $\mathbf{x}$ are bounded random variables [23]. Assumption 2, which is a restricted eigenvalue (RE) type of condition, is needed to establish ℓ_2-type error bound for ℓ_1-penalized type of estimator. Similar conditions have been widely adopted in the literature [5], [7]. Besides restricted minimum eigenvalue, restricted maximum eigenvalue is also imposed for the random sample covariance matrix in [23] to establish theory for ℓ_1-norm SVM. Such condition is avoided in our theory, which is achieved by a more careful analysis and a different strategy in our proof.

Theorem 1: Assume assumptions 1-2 hold, and $s\log p = o(n)$. Choose the tuning parameters such that $8\lambda_0\|\boldsymbol{\beta}^*\|_{\max} < \lambda$. Then there exists a large enough constant $c_0 > 0$ such that with the choice $\lambda = c_0\sqrt{\frac{\log p}{n}}$, the elastic-net DCSVM estimator $(\hat{\beta}_0, \hat{\boldsymbol{\beta}})$ satisfies

$$|\hat{\beta}_0 - \beta_0^*|^2 + \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2 = O_p\Big(\frac{s\log p}{n}\Big).$$

Theorem 1 shows that the sparse density convoluted SVM estimator shares the same optimal rate of convergence as the ℓ_1-SVM [23]. Meanwhile, the sparse DCSVM has better computational efficiency than penalized SVM due to the smoothness of its loss function.

IV. Computation

In this section, we develop an efficient algorithm for computing the solution path of DCSVM.

At the outset, we present the first-order derivative of the density-convoluted SVM loss and show they are Lipschitz continuous in Lemma 1:

$$L_h^{G\prime}(v) = -\Phi\left(\frac{1-v}{h}\right),$$

$$L_h^{E\prime}(v) =$$

$$\begin{cases} -1, & v \leq 1-h, \\ -\dfrac{(1-v+h)^2(2h-(1-v))}{4h^3}, & 1-h < v \leq 1+h, \\ 0, & v \geq 1+h. \end{cases}$$

Lemma 1: Let $L_h^G(v)$ and $L_h^E(v)$ be the DCSVM loss using Gaussian kernel and Epanechnikov kernel, respectively. For $v_1 < v_2$,

$$|L_h^{G\prime}(v_1) - L_h^{G\prime}(v_2)| < c_h^G|v_1 - v_2|, \tag{IV.1}$$

$$|L_h^{E\prime}(v_1) - L_h^{E\prime}(v_2)| < c_h^E|v_1 - v_2|, \tag{IV.2}$$

where the Lipschitz constants are given as $c_h^G = \frac{1}{\sqrt{2\pi}h}$ and $c_h^E = \frac{3}{4h}$.

The bottom row of Figure 1 depicts $L_h^{G\prime}(v)$ and $L_h^{E\prime}(v)$.

Lemma 1 gives rise to the following quadratic majorization condition for the DCSVM:

$$L_h(v_1) \leq L_h(v_2) + L_h^\prime(v_2)(v_1 - v_2) + \frac{c_h}{2}(v_1 - v_2)^2,$$

where L_h is exemplified by L_h^G and L_h^E and c_h is the corresponding Lipschitz constant.

Based on the Lipschitz condition, we develop a generalized coordinate descent (GCD) algorithm [35] to solve those sparse penalized DCSVMs. We first consider the adaptive lasso penalty. The algorithm can be easily adjusted to handle lasso and elastic net.

Without loss of generality, we assume each $\mathbf{X}_j$ has zero mean and unit length. In a coordinate-wise manner, suppose the coordinate $\beta_1, \beta_2, \ldots, \beta_{j-1}$ have been updated and we now update β_j. Denote by $\tilde{\beta}_0$ and $\tilde{\boldsymbol{\beta}}$ by the current solution and let $v_i = y_i(\tilde{\beta}_0 + \mathbf{x}_i^{\mathrm{T}}\tilde{\boldsymbol{\beta}})$. To update β_j, instead of solving the coordinate-wise update function,

$$F(\beta_j) = \frac{1}{n}\sum_{i=1}^n L_h\left(v_i + y_i x_{ij}\left(\beta_j - \tilde{\beta}_j\right)\right) + \lambda w_j|\beta_j|,$$

we solve its majorization function

$$Q\left(\beta_j\right) = \frac{1}{n}\sum_{i=1}^n L_h\left(v_i\right) + \frac{1}{n}\sum_{i=1}^n L_h^\prime\left(v_i\right) y_i x_{ij}\left(\beta_j - \tilde{\beta}_j\right) + \frac{c_h}{2}\left(\beta_j - \tilde{\beta}_j\right)^2 + \lambda w_j|\beta_j|$$

that is obtained through the quadratic majorization condition. The minimizer of $Q\left(\beta_j\right)$ is c_1c_2, where

$$c_1 = \tilde{\beta}_j - \frac{1}{c_h n}\sum_{i=1}^n L_h^\prime(v_i)y_i x_{ij},$$

$$c_2 = \left(1 - \frac{\lambda w_j}{\left|c_h\tilde{\beta}_j - \frac{1}{n}\sum_{i=1}^n L_h^\prime(v_i)y_i x_{ij}\right|}\right)_+.$$

Likewise, β_0 is updated to be $\tilde{\beta}_0 - \frac{1}{c_h n}\sum_{i=1}^n L_h^\prime(v_i)y_i$.

In our implementation, we further apply the strong rule [26], warm start, and active set strategy [9] to further accelerate the algorithm.

V. Numerical Studies

A. Simulation

In this section, we use several simulation examples to demonstrate the performance of DCSVM.

The response variables of all the simulated data are binary and the two classes are balanced, i.e., $P(Y=1) = P(Y=-1) = 0.5$. In each example, define the p-dimensional mean vectors $\boldsymbol{\mu}_+ = (0.7, 0.7, 0.7, 0.7, 0.7, 0, 0, \ldots, 0)$ and $\boldsymbol{\mu}_- = -\boldsymbol{\mu}_+$, where $p = 500$ or 5000 in our experiments. Each observation from the positive class is drawn from $\mathrm{N}(\boldsymbol{\mu}_+, \boldsymbol{\Sigma})$ and each observation from the negative class is drawn from $\mathrm{N}(\boldsymbol{\mu}_-, \boldsymbol{\Sigma})$. We consider three different choices of $\boldsymbol{\Sigma}$. In example 1, $\boldsymbol{\Sigma} = \mathbf{I}_{p\times p}$ so the variables are independent. In both examples 2 and 3,

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}^\star_{5\times 5} & \mathbf{0}_{5\times(p-5)} \\ \mathbf{0}_{(p-5)\times 5} & \mathbf{I}_{(p-5)\times(p-5)} \end{pmatrix}$$

where $\boldsymbol{\Sigma}^\star_{5\times 5}$ have all diagonal elements of 1 and off-diagonal elements of ρ in example 2, and $(\boldsymbol{\Sigma}^\star_{5\times 5})_{i,j} = \rho^{|i-j|}$ in example 3. We use $\rho = 0.2, 0.7$, and 0.9.

We first compared elastic-net DCSVM with Gaussian kernel and Epanechnikov kernel with elastic-net SVM [33] and elastic-net logistic regression that is fitted using the R package `gcdnet` [35]. For each example, the training size is 200 and we use five-fold cross-validation to select the best tuple of (h, λ_0, λ) where h is chosen from $0.1, 0.25, 0.5$, and 1, λ_0 is selected from $0.5 * (10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 5)$, and λ is searched along the solution path; for the SVM and logistic regression, we select λ_0 and λ in the same manner.

We record the prediction error and run time in Table I. The run time include all the time spent on tuning and training the models. We observe the DCSVM with Epanechnikov kernel has slightly better performance than DCSVM with Gaussian kernel, and both of them have better prediction accuracy than the other two methods. DCSVM with Epanechnikov kernel is the fastest while the elastic-net SVM is the slowest.

All the methods exhibited in Table I use elastic-net penalty. We now study the performance when using other sparse penalities. Due to the overall best performance, we stay with DCSVM with Epanechnikov kernel and we compare the prediction accuracy and variable selection when using lasso and elastic-net penalties. We present the results in Table II. In general, we find the elastic-net has the best performance in both prediction error and variable selection.

To visualize the algorithmic convergence, we generate a convergence plot when using GCD algorithm to computing a DCSVM with certain tuning parameters in Example 1 with $n = 500$. As displayed in Figure 2, we see the objective value strictly decreases.

B. Benchmark Data Applications

In this section, we demonstrate the performance of DCSVM using ten benchmark data, which are available from UCI machine learning repository. We randomly split each data set into a training set and a test set with a 1:1 ratio. On the training set, we fit elastic-net DCSVM, elastic-net logistic regression, and elastic-net SVM, and tune each method using five-fold

TABLE I

COMPARISON OF PREDICTION ERROR (IN PERCENTAGE) AND RUN TIME (IN SECOND) OF ELASTIC-NET DENSITY-CONVOLUTED SVM WITH GAUSSIAN AND EPANECHNIKOV KERNELS, ELASTIC-NET SVM, AND ELASTIC-NET LOGISTC REGRESSION. UNDER EACH SIMULATION SETTING, THE METHOD WITH THE LOWEST PREDICTION ERROR IS MARKED BY A BLACK BOX. ALL THE QUANTITIES ARE AVERAGED OVER 50 INDEPENDENT RUNS AND THE STANDARD ERRORS OF THE PREDICTION ERROR ARE GIVEN IN PARENTHESES

		DCSVM-Gaussian			DCSVM-Epanechnikov			SVM			logistic		
p	ρ	err (%)		time	err (%)		time	err (%)		time	err (%)		time
Example 1													
500		6.83	(0.14)	267.89	**6.75**	(0.14)	29.67	9.76	(1.51)	1362.44	6.98	(0.15)	49.78
5000		**7.11**	(0.13)	771.87	7.29	(0.16)	139.07	7.90	(0.87)	28323.47	7.33	(0.17)	417.54
Example 2													
500	0.2	13.52	(0.19)	305.95	**13.48**	(0.17)	33.42	16.02	(1.26)	1687.62	13.88	(0.22)	52.44
	0.7	22.65	(0.25)	385.08	**22.50**	(0.27)	41.39	25.75	(1.21)	1585.23	22.88	(0.28)	59.99
	0.9	24.76	(0.24)	467.40	**24.57**	(0.24)	48.78	27.42	(1.16)	1510.98	24.82	(0.31)	69.52
5000	0.2	13.78	(0.18)	806.36	**13.72**	(0.21)	142.09	16.32	(1.25)	30170.44	14.12	(0.26)	420.09
	0.7	**22.66**	(0.21)	890.84	23.00	(0.24)	150.44	24.15	(0.79)	31865.01	23.03	(0.23)	435.63
	0.9	**24.70**	(0.25)	975.34	24.76	(0.24)	154.73	26.88	(1.00)	32132.55	25.03	(0.24)	450.30
Example 3													
500	0.2	10.30	(0.15)	290.41	**10.13**	(0.16)	31.53	12.04	(1.14)	1476.20	10.69	(0.24)	51.16
	0.7	19.48	(0.18)	368.74	**19.40**	(0.18)	39.71	22.90	(1.34)	1726.07	19.80	(0.25)	60.53
	0.9	**23.50**	(0.22)	435.55	23.54	(0.22)	44.92	26.55	(1.19)	1625.15	23.93	(0.28)	66.23
5000	0.2	10.51	(0.20)	793.67	**10.46**	(0.18)	141.23	13.02	(1.35)	34555.70	10.74	(0.21)	418.58
	0.7	**19.70**	(0.21)	877.54	19.89	(0.22)	146.99	22.54	(1.18)	34574.72	20.09	(0.25)	433.84
	0.9	23.85	(0.23)	944.63	**23.81**	(0.24)	152.78	26.55	(1.11)	36732.99	23.90	(0.24)	445.60

TABLE II

COMPARISON OF PREDICTION ERROR (IN PERCENTAGE) AND VARIABLE SELECTION OF DENSITY-CONVOLUTED SVM WITH EPANECHNIKOV KERNELS USING LASSO AND ELASTIC-NET (ENET) PENALTIES. DENOTE BY C AND IC THE NUMBER OF CORRECTLY AND INCORRECTLY SELECTED VARIABLES, RESPECTIVELY. UNDER EACH SIMULATION SETTING, THE METHOD WITH THE LOWEST PREDICTION ERROR IS MARKED BY A BLACK BOX. ALL THE QUANTITIES ARE AVERAGED OVER 50 INDEPENDENT RUNS AND THE STANDARD ERRORS OF THE PREDICTION ERROR ARE GIVEN IN PARENTHESES

		lasso-DCSVM				enet-DCSVM			
p	ρ	err (%)		C	IC	err (%)		C	IC
Example 1									
500		6.88	(0.14)	5	0	**6.77**	(0.14)	5	0
5000		7.31	(0.19)	5	0	**7.29**	(0.16)	5	0
Example 2									
500	0.2	13.89	(0.23)	5	0	**13.47**	(0.17)	5	0
	0.7	22.86	(0.20)	3	0	**22.51**	(0.27)	5	0
	0.9	24.53	(0.19)	2	0	**24.51**	(0.23)	4	0
5000	0.2	14.55	(0.25)	5	0	**13.72**	(0.21)	5	0
	0.7	23.41	(0.23)	3	0	**23.05**	(0.25)	4	0
	0.9	25.36	(0.35)	2	0	**24.76**	(0.26)	3	0
Example 3									
500	0.2	10.47	(0.22)	5	0	**10.09**	(0.15)	5	0
	0.7	19.90	(0.22)	3	0	**19.44**	(0.19)	4	0
	0.9	23.74	(0.20)	3	0	**23.49**	(0.22)	4	0
5000	0.2	10.78	(0.23)	5	0	**10.48**	(0.18)	5	0
	0.7	20.12	(0.22)	3	0	**19.89**	(0.22)	4	0
	0.9	24.34	(0.31)	2	0	**23.81**	(0.24)	3	0

cross-validation. The prediction accuracy is computed based on the test set.

We present the result in Table III. We observe the elastic-net DCSVM has the best performance in general. We further

TABLE III

COMPARISON OF PREDICTION ERROR (IN PERCENTAGE) AND RUN TIME (IN SECOND) OF ELASTIC-NET DENSITY-CONVOLUTED SVM WITH EPANECHNIKOV KERNEL, ELASTIC-NET SVM, AND ELASTIC-NET LOGISTC REGRESSION. FOR EACH BENCHMARK DATA, THE METHOD WITH THE LOWEST PREDICTION ERROR IS MARKED BY A BLACK BOX. ALL THE QUANTITIES ARE AVERAGED OVER 50 INDEPENDENT RUNS AND THE STANDARD ERRORS OF THE PREDICTION ERROR ARE GIVEN IN PARENTHESES

			enet-DCSVM			enet-SVM			enet-logistic		
data	n	p	err (%)		time	err (%)		time	err (%)		time
arcene	100	9920	$\boxed{\mathbf{31.82}}$	(1.42)	48.05	36.06	(1.63)	6121.27	34.24	(1.64)	229.13
breast	42	22283	$\boxed{\mathbf{25.07}}$	(2.14)	43.29	29.27	(2.80)	1051.75	31.37	(3.08)	219.00
colon	62	2000	$\boxed{\mathbf{17.81}}$	(1.01)	9.68	18.39	(1.54)	534.62	23.55	(1.51)	33.12
DLBCL	47	1909	$\boxed{\mathbf{13.04}}$	(1.03)	7.08	14.52	(1.09)	172.65	14.61	(1.40)	20.67
leuk	72	7128	$\boxed{\mathbf{3.56}}$	(0.47)	21.14	3.83	(0.51)	1137.33	4.39	(0.61)	117.30
LSVT	126	309	16.10	(0.70)	7.53	16.25	(0.68)	74.53	$\boxed{\mathbf{15.71}}$	(0.64)	13.22
lung	181	12533	$\boxed{\mathbf{1.00}}$	(0.12)	83.15	1.76	(0.38)	14633.12	1.33	(0.20)	624.62
malaria	71	22283	$\boxed{\mathbf{5.86}}$	(0.86)	75.18	9.00	(1.72)	7280.00	8.00	(1.29)	481.80
ovarian	253	15154	$\boxed{\mathbf{0.63}}$	(0.12)	139.08	4.87	(1.23)	11186.04	0.87	(0.14)	913.36
prostate	102	6033	9.22	(0.66)	25.35	$\boxed{\mathbf{8.78}}$	(0.51)	1503.13	10.27	(0.62)	116.58

TABLE IV

COMPARISON OF PREDICTION ERROR (IN PERCENTAGE) OF ELASTIC-NET DENSITY-CONVOLUTED SVM WITH ℓ_1 LDA, RANDOM FOREST, AND NEURAL NETS. FOR EACH BENCHMARK DATA, THE METHOD WITH THE LOWEST PREDICTION ERROR IS MARKED BY A BLACK BOX. ALL THE QUANTITIES ARE AVERAGED OVER 50 INDEPENDENT RUNS AND THE STANDARD ERRORS OF THE PREDICTION ERROR ARE GIVEN IN PARENTHESES

data	n	p	enet-DCSVM		ℓ_1 LDA		RF		neural nets	
arcene	100	9920	31.82	(1.42)	41.94	(1.13)	$\boxed{\mathbf{29.27}}$	(1.43)	44.12	(0.72)
breast	42	22283	$\boxed{\mathbf{25.07}}$	(2.14)	29.13	(1.81)	26.33	(1.51)	41.32	(1.20)
colon	62	2000	$\boxed{\mathbf{17.81}}$	(1.01)	18.52	(1.13)	21.61	(1.24)	36.77	(0.93)
DLBCL	47	1909	$\boxed{\mathbf{13.04}}$	(1.03)	14.52	(1.07)	13.39	(1.35)	42.70	(0.69)
leuk	72	7128	3.56	(0.47)	$\boxed{\mathbf{2.78}}$	(0.43)	4.78	(0.64)	33.28	(0.69)
LSVT	126	309	$\boxed{\mathbf{16.10}}$	(0.70)	23.02	(1.17)	17.56	(0.66)	33.11	(0.65)
lung	181	12533	$\boxed{\mathbf{1.00}}$	(0.12)	1.80	(0.35)	$\boxed{\mathbf{1.00}}$	(0.18)	16.80	(0.37)
malaria	71	22283	$\boxed{\mathbf{5.86}}$	(0.86)	7.21	(0.66)	8.79	(1.15)	33.00	(0.85)
ovarian	253	15154	$\boxed{\mathbf{0.63}}$	(0.12)	10.37	(0.64)	2.05	(0.13)	35.44	(0.46)
prostate	102	6033	$\boxed{\mathbf{9.22}}$	(0.66)	34.86	(1.75)	14.00	(0.94)	45.69	(0.54)

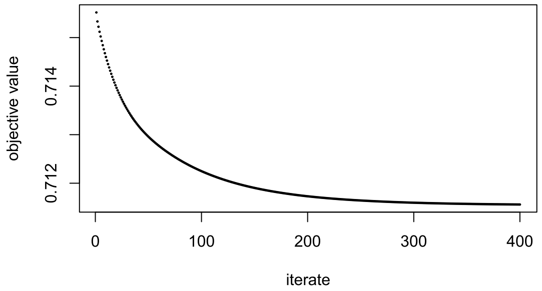

Fig. 2. Convergence plot for solving a DCSVM with cross-validated tuning parameters using GCD algorithm under the simulation setting in Example 1 with $n = 500$.

compare the elastic-net DCSVM with ℓ_1 LDA [34], random forest (implemented in the R package `randomForest` [19]), and neural nets (implemented in the R package `nnet` [30]). As shown in Table IV, DCSVM in general outperforms the other three methods.

VI. DISCUSSION

In this article we have proposed a new classification method called DCSVM, which is motivated from smoothing the SVM by the density convolution. We have imposed the elastic-net penalty on the DCSVM to perform high-dimensional classification. We have rigorously shown elastic-net DCSVM retains the nice statistical property of the sparse SVM and share the same convergence rate $O_p(\sqrt{s \log p/n})$. It is also worth noting that the theoretical conditions imposed in this paper is more general than that imposed for the sparse SVM. By showing elastic-net DCSVM satisfies the Lipschitz condition, we developed an efficient algorithm for solving elastic-net DCSVM. We further rigorously justified that the algorithm converges at least linearly. With extensive numerical studies, we have demonstrated the superior performance of DCSVM over the original SVM, as well as many popular classifiers such as logistic regression, LDA, random forest, and neural nets. The R package for DCSVM is available from URL: https://z.umn.edu/dcsvm.

In this work, we focus on binary classification. In the future research, it will be interesting to generalize the proposal in this paper to multi-class classification. A potential approach is the so-called margin vector that has been developed in [39] and [32] to extend the binary large-margin classifiers to the multi-class situation.

Appendix

A. Proof of Theorem 1

We first give some general formula regarding the loss function L_h and its derivatives. Recall $L_h(u) = \int_{-\infty}^{\infty}(1-u+v)_+\frac{1}{h}K(\frac{v}{h})\mathrm{d}v, u \in \mathbb{R}$. A direct calculation gives

$$\begin{aligned}
L_h(t) &= \int_{-\infty}^{1} \frac{1-u}{h}K(\frac{t-u}{h})\mathrm{d}u, \\
L_h'(t) &= -\int_{-\infty}^{\frac{1-t}{h}} K(u)\mathrm{d}u, \\
L_h''(t) &= \frac{1}{h}K(\frac{1-t}{h}),\ \forall t \in \mathbb{R}.
\end{aligned}$$

It is important to note that $|L_h'(\cdot)| \le 1$, since $K(t) \ge 0, \forall t$ and $\int_{-\infty}^{\infty} K(u)\mathrm{d}u = 1$.

Proof of Theorem 1: By definition of the elastic-net-penalized DCSVM in (II.3), we see

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \hat{\beta}_0)\big) + \lambda\|\hat{\boldsymbol{\beta}}\|_1 + \lambda_0\|\hat{\boldsymbol{\beta}}\|_2^2 \\
&\le \frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big) + \lambda\|\boldsymbol{\beta}^*\|_1 + \lambda_0\|\boldsymbol{\beta}^*\|_2^2.
\end{aligned}$$

By triangle inequality, we further see

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \hat{\beta}_0)\big) - \frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big) \\
&+ \lambda_0(\|\hat{\boldsymbol{\beta}}\|_2^2 - \|\boldsymbol{\beta}^*\|_2^2) \\
&\le \lambda(\|\boldsymbol{\beta}^*\|_1 - \|\hat{\boldsymbol{\beta}}\|_1) \\
&\le \lambda(\|\boldsymbol{\beta}^*_{\mathbb{A}} - \hat{\boldsymbol{\beta}}_{\mathbb{A}}\|_1 + \|\hat{\boldsymbol{\beta}}_{\mathbb{A}}\|_1 - \|\hat{\boldsymbol{\beta}}_{\mathbb{A}}\|_1 - \|\hat{\boldsymbol{\beta}}_{\mathbb{A}^{\mathrm{c}}} - \boldsymbol{\beta}^*_{\mathbb{A}^{\mathrm{c}}}\|_1) \\
&= \lambda\|\mathbf{u}_{\mathbb{A}}\|_1 - \|\mathbf{u}_{\mathbb{A}^{\mathrm{c}}}\|_1),
\end{aligned}$$

where we denote $\mathbf{u} := \hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$. On the other hand, by convexity of $L_h(\cdot)$, we have

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \hat{\beta}_0)\big) - \frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big) \\
&+ \lambda_0(\|\hat{\boldsymbol{\beta}}\|_2^2 - \|\boldsymbol{\beta}^*\|_2^2) \\
&\ge \frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i(\hat{\beta}_0 - \beta_0^*) \\
&+ \Big(2\lambda_0\boldsymbol{\beta}^{*\mathrm{T}} + \frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\mathbf{x}_i^{\mathrm{T}}\Big)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \\
&\ge -\Big|\frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\Big| \cdot |\delta| \\
&- \Big\|2\lambda_0\boldsymbol{\beta}^* + \frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\mathbf{x}_i\Big\|_\infty \\
&(\|\mathbf{u}_{\mathbb{A}}\|_1 + \|\mathbf{u}_{\mathbb{A}^{\mathrm{c}}}\|_1),
\end{aligned}$$

where $\delta := \hat{\beta}_0 - \beta_0^*$. Define events

$$\begin{aligned}
&\mathcal{E}_1 := \left\{\left|\frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\right| \le \frac{\lambda}{2}\right\}, \\
&\mathcal{E}_2 := \\
&\left\{\left\|2\lambda_0\boldsymbol{\beta}^* + \frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\mathbf{x}_i\right\|_\infty \le \frac{\lambda}{2}\right\}.
\end{aligned}$$

Note that $\mathbb{E}\big[L_h'\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y\big] = 0$, and $\big|L_h'\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y\big| \le 1$. Hence by Hoeffding's inequality,

$$\begin{aligned}
&\mathrm{P}(\mathcal{E}_1^{\mathrm{c}}) \\
&= \mathrm{P}\Big(\Big|\frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\Big| > \frac{\lambda}{2}\Big) \\
&\le 2\exp\Big\{-\frac{n\lambda^2}{8}\Big\}.
\end{aligned}$$

Meanwhile, we have $\mathbb{E}\big[L_h'\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y\mathbf{x}\big] = \mathbf{0}$ by the definition of $\boldsymbol{\beta}^*$ and optimality condition. By the choice of tuning parameters we have

$$\begin{aligned}
&\mathrm{P}(\mathcal{E}_2^{\mathrm{c}}) \\
&= \mathrm{P}\Big(\big\|2\lambda_0\boldsymbol{\beta}^* + \frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\mathbf{x}_i\big\|_\infty > \frac{\lambda}{2}\Big) \\
&\le \mathrm{P}\Big(\big\|\frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\mathbf{x}_i\big\|_\infty > \frac{\lambda}{4}\Big) \\
&\le \sum_{j=1}^{p}\mathrm{P}\Big(\big|\frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_ix_{ij}\big| > \frac{\lambda}{4}\Big).
\end{aligned}$$

Notice that by assumption 1 and $|L_h'(\cdot)| \le 1$,

$$\mathbb{E}\Big[\mathrm{e}^{|L_h'(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*))y_ix_{ij}|/m_0}\Big] \le \mathbb{E}\Big[\mathrm{e}^{\frac{|x_{ij}|}{m_0}}\Big] \le 2.$$

This implies that

$$\begin{aligned}
&\|L_h'(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*))y_ix_{ij}\|_{\psi_1} \le m_0, \\
&\forall i \in \{1, \ldots, n\}, \forall j \in \{1, \ldots, p\}.
\end{aligned}$$

By Theorem 1.4 in [11], there exists an absolute constant $\eta_0 > 0$ such that

$$\begin{aligned}
&\mathrm{P}\Big(\big|\frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_ix_{ij}\big| > \frac{\lambda}{4}\Big) \qquad \text{(A.1.1)} \\
&\le 2\mathrm{e}^{-\frac{1}{\eta_0}(\frac{\lambda^2}{16m_0^2} \wedge \frac{\lambda}{4m_0})n}.
\end{aligned}$$

So following (A.1.1) we have

$$\mathrm{P}(\mathcal{E}_2^{\mathrm{c}}) \le 2p\mathrm{e}^{-\frac{1}{\eta_0}(\frac{\lambda^2}{16m_0^2} \wedge \frac{\lambda}{4m_0})n}.$$

Now, under $\mathcal{E}_1 \cap \mathcal{E}_2$, combining (A.1.1) and (A.1.1) we have

$$-\frac{\lambda}{2}(|\delta| + \|\mathbf{u}_{\mathbb{A}}\|_1 + \|\mathbf{u}_{\mathbb{A}^{\mathrm{c}}}\|_1) \le \lambda(\|\mathbf{u}_{\mathbb{A}}\|_1 - \|\mathbf{u}_{\mathbb{A}^{\mathrm{c}}}\|_1),$$

which implies $\|\mathbf{u}_{\mathbb{A}^{\mathrm{c}}}\|_1 \le 3\|\mathbf{u}_{\mathbb{A}}\|_1 + |\delta|$, or $(\delta, \mathbf{u}) \in \mathcal{S}_{\mathbb{A}}$.

Define $F(\beta_0, \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \beta_0)\big)$ for any $(\beta_0, \boldsymbol{\beta}) \in \mathbb{R} \times \mathbb{R}^p$. Also, define

$$\mathbb{C}(r) = \Big\{(w, \mathbf{w}) \in \mathcal{S}_{\mathbb{A}} : |w|^2 + \|\mathbf{w}\|_2^2 = r^2\frac{s\log p}{n}\Big\}$$

for any $r > 0$. Let $G(\beta_0, \boldsymbol{\beta}) = F(\beta_0, \boldsymbol{\beta}) - F(\beta_0^*, \boldsymbol{\beta}^*)$, and let

$$H(r) = \sup_{(\beta_0, \boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)} \big| G(\beta_0, \boldsymbol{\beta}) - \mathbb{E}[G(\beta_0, \boldsymbol{\beta})] \big|.$$

We give an upper bound for $\mathbb{E}[H(r)]$. Let $\sigma_1, \dots, \sigma_n$ be i.i.d. Rademacher random variables (i.e. $\mathrm{P}(\sigma_i = 1) = \mathrm{P}(\sigma_i = -1) = \frac{1}{2}$), which is independent from all the other random elements. By the symmetrization inequality (see for instance, Lemma 2.3.1 in [28]) and contraction inequality (see for instance, Theorem 4.12 in [18]), $|L_h'(\cdot)| \leq 1$ and Cauchy-Schwarz inequality, we have

$$\begin{aligned}
&\mathbb{E}[H(r)] \\
&\leq 2\mathbb{E}\Big[\sup_{(\beta_0, \boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)} \Big| \frac{1}{n} \sum_{i=1}^n \sigma_i \Big\{ L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta} + \beta_0)\big) \\
&\quad - L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}^* + \beta_0^*)\big) \Big\} \Big| \Big] \\
&\leq 4\mathbb{E}\Big[\sup_{(\beta_0, \boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)} \Big| \frac{1}{n} \sum_{i=1}^n \sigma_i y_i \big(\mathbf{x}_i^{\mathrm{T}} (\boldsymbol{\beta} - \boldsymbol{\beta}^*) \\
&\quad + \beta_0 - \beta_0^* \big) \Big| \Big] \\
&\leq \frac{4}{n}\, \mathbb{E}\Big[\Big\| \sum_{i=1}^n \sigma_i y_i (1, \mathbf{x}_i^{\mathrm{T}})^{\mathrm{T}} \Big\|_\infty \Big] \\
&\quad \Big(4\sqrt{s} \cdot r \sqrt{\frac{s \log p}{n}} + 2r \sqrt{\frac{s \log p}{n}} \Big).
\end{aligned}$$

By assumption 1 and definition of Orlicz norm, we know $\|\sigma_i y_i x_{ij}\|_{\psi_1} = \|x_{ij}\|_{\psi_1} \leq m_0$, $\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, p\}$. Also, it is straightforward to see $\|\sigma_i y_i\|_{\psi_1} = \frac{1}{\log 2}$. By Proposition 2.7.1 in [31], there exists a constant $c_1 > 0$ such that $\mathbb{E}[\mathrm{e}^{t\sigma_i y_i x_{ij}}] \leq \mathrm{e}^{c_1^2 t^2}$ and $\mathbb{E}[\mathrm{e}^{t\sigma_i y_i}] \leq \mathrm{e}^{c_1^2 t^2}$ for all $|t| < \frac{1}{c_1}$, $\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, p\}$. By Jensen's inequality, we have for any $0 < t < \frac{1}{c_1}$,

$$\begin{aligned}
&\mathrm{e}^{t\mathbb{E}[\max\{\max_{1 \leq j \leq p} |\sum_{i=1}^n \sigma_i y_i x_{ij}|, |\sum_{i=1}^n \sigma_i y_i|\}]} \\
&\leq \mathbb{E}[\mathrm{e}^{t \max\{\max_{1 \leq j \leq p} |\sum_{i=1}^n \sigma_i y_i x_{ij}|, |\sum_{i=1}^n \sigma_i y_i|\}}] \\
&\leq \mathbb{E}\Big[\max_{1 \leq j \leq p} (\mathrm{e}^{t \sum_{i=1}^n \sigma_i y_i x_{ij}} + \mathrm{e}^{-t \sum_{i=1}^n \sigma_i y_i x_{ij}}) \\
&\quad + \mathrm{e}^{t \sum_{i=1}^n \sigma_i y_i} + \mathrm{e}^{-t \sum_{i=1}^n \sigma_i y_i} \Big] \\
&\leq \sum_{j=1}^p (\prod_{i=1}^n \mathbb{E}[\mathrm{e}^{t\sigma_i y_i x_{ij}}] + \prod_{i=1}^n \mathbb{E}[\mathrm{e}^{-t\sigma_i y_i x_{ij}}]) \\
&\quad + \prod_{i=1}^n \mathbb{E}[\mathrm{e}^{t\sigma_i y_i}] + \prod_{i=1}^n \mathbb{E}[\mathrm{e}^{-t\sigma_i y_i}] \\
&\leq 2p\mathrm{e}^{c_1^2 t^2 n} + 2\mathrm{e}^{c_1^2 t^2 n} \leq 4p\mathrm{e}^{c_1^2 t^2 n}.
\end{aligned}$$

Consequently, for any $0 < t < \frac{1}{c_1}$,

$$\mathbb{E}\Big[\Big\| \sum_{i=1}^n \sigma_i y_i (1, \mathbf{x}_i^{\mathrm{T}})^{\mathrm{T}} \Big\|_\infty \Big] \leq \frac{\log p + \log 4}{t} + c_1^2 t n. \quad \text{(A.1.2)}$$

By the condition of Theorem 1, we know

$$\frac{\sqrt{\log p + \log 4}}{c_1 \sqrt{n}} = o(1),$$

so for large enough n,

$$\frac{\sqrt{\log p + \log 4}}{c_1 \sqrt{n}} < \frac{1}{c_1}.$$

Thus, choosing $t = \frac{\sqrt{\log p + \log 4}}{c_1 \sqrt{n}}$ in (A.1.2) we obtain

$$\mathbb{E}\Big[\Big\| \sum_{i=1}^n \sigma_i y_i (1, \mathbf{x}_i^{\mathrm{T}})^{\mathrm{T}} \Big\|_\infty \Big] \leq 2c_1 \sqrt{(\log p + \log 4)n} \quad \text{(A.1.3)}$$

for large enough n. Thus, combining (A.1.2) and (A.1.3) we get

$$\begin{aligned}
&\mathbb{E}[H(r)] \\
&\leq \frac{4}{n} \cdot 2c_1 \sqrt{(\log p + \log 4)n} \\
&\Big(4\sqrt{s} \cdot r \sqrt{\frac{s \log p}{n}} + 2r \sqrt{\frac{s \log p}{n}} \Big) \\
&\leq \frac{96 c_1 r s \log p}{n}.
\end{aligned}$$

This implies that $H(r) = O_p(\frac{rs \log p}{n})$. For any $T > 0$, define event

$$\mathcal{G}_T \coloneqq \{H(r) \leq \frac{Trs \log p}{n}\},$$

then we have $\lim_{T \to \infty} \limsup_{n \to \infty} P(\mathcal{G}_T^{\mathrm{c}}) = 0$.

Next, for any $(\beta_0, \boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)$, we derive a lower bound for $\mathbb{E}[G(\beta_0, \boldsymbol{\beta})]$. For large enough n, for any $(\beta_0, \boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)$, by Taylor's theorem and assumption 2, there exists $a \in [0, 1]$ such that

$$\begin{aligned}
&\mathbb{E}[G(\beta_0, \boldsymbol{\beta})] \\
&= \mathbb{E}\big[L_h\big(y(\mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} + \beta_0)\big)\big] - \mathbb{E}\big[L_h\big(y(\mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}^* + \beta_0^*)\big)\big] \\
&= \frac{1}{2}(\beta_0 - \beta_0^*, (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^{\mathrm{T}}) I_{(\beta_0^* + a(\beta_0 - \beta_0^*), \boldsymbol{\beta}^* + a(\boldsymbol{\beta} - \boldsymbol{\beta}^*))} \\
&(\beta_0 - \beta_0^*, (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^{\mathrm{T}})^{\mathrm{T}} \\
&\geq \frac{1}{2}\rho\big((\beta_0 - \beta_0^*)^2 + \|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_2^2\big) \\
&\geq \frac{1}{2}\rho r^2 \frac{s \log p}{n}.
\end{aligned}$$

On the other hand, by our choice for tuning parameters, for any $(\beta_0, \boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)$ we have

$$\begin{aligned}
&\lambda \big| \|\boldsymbol{\beta}\|_1 - \|\boldsymbol{\beta}^*\|_1 \big| \\
&\leq \lambda \|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}}\|_1 + \lambda \|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}^{\mathrm{c}}}\|_1 \\
&\leq 4\lambda \|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}}\|_1 + \lambda |\beta_0 - \beta_0^*| \\
&\leq 4\lambda \sqrt{s} \|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}}\|_2 + \lambda r \sqrt{\frac{s \log p}{n}} \\
&\leq 4\lambda \sqrt{s} r \sqrt{\frac{s \log p}{n}} + \lambda r \sqrt{\frac{s \log p}{n}} \\
&\leq 5 c_0 s r \frac{\log p}{n},
\end{aligned}$$

and we also have, by convexity of ℓ_2 norm,

$$\begin{aligned}&\lambda_0(\|\boldsymbol{\beta}\|_2^2 - \|\boldsymbol{\beta}^*\|_2^2)\\ &\geq 2\lambda_0\boldsymbol{\beta}^{*\mathrm{T}}(\boldsymbol{\beta} - \boldsymbol{\beta}^*)\\ &\geq -2\lambda_0\|\boldsymbol{\beta}^*\|_{\max}\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_1\\ &\geq -\frac{\lambda}{4}(4\|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}}\|_1 + |\beta_0 - \beta_0^*|)\\ &\geq -\lambda\sqrt{s}r\sqrt{\frac{s\log p}{n}} - \frac{\lambda}{4}r\sqrt{\frac{s\log p}{n}}\\ &\geq -\frac{2c_0 sr\log p}{n}.\end{aligned}$$

Thus, combining (A.1.4), (A.1.4) and (A.1.4), under $\mathcal{G}_T$, we have for any $(\beta_0, \boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)$,

$$\begin{aligned}&F(\beta_0, \boldsymbol{\beta}) + \lambda_0\|\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\\ &- F(\beta_0^*, \boldsymbol{\beta}^*) - \lambda_0\|\boldsymbol{\beta}^*\|_2^2 - \lambda\|\boldsymbol{\beta}^*\|_1\\ &\geq G(\beta_0, \boldsymbol{\beta}) - \frac{7c_0 sr\log p}{n}\\ &\geq \mathbb{E}[G(\beta_0, \boldsymbol{\beta})] - H(r) - \frac{7c_0 sr\log p}{n}\\ &\geq \mathbb{E}[G(\beta_0, \boldsymbol{\beta})] - \frac{Trs\log p}{n} - 7c_0 sr\frac{\log p}{n}\\ &\geq \Big(\frac{1}{2}\rho r - T - 7c_0\Big)\frac{rs\log p}{n}.\end{aligned}$$

Now, choose $r = \frac{4T+28c_0}{\rho}$, we have that under $\mathcal{G}_T$,

$$\begin{aligned}&\inf_{(\beta_0,\boldsymbol{\beta})\in(\beta_0^*,\boldsymbol{\beta}^*)+\mathbb{C}(r)} F(\beta_0, \boldsymbol{\beta}) + \lambda_0\|\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\\ &> F(\beta_0^*, \boldsymbol{\beta}^*) + \lambda_0\|\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_1. \qquad \text{(A.1.4)}\end{aligned}$$

Recall that under $\mathcal{E}_1 \cap \mathcal{E}_2$,

$$(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) \in (\beta_0, \boldsymbol{\beta}^*) + \mathcal{S}_{\mathbb{A}}.$$

We next claim that under $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{G}_T$,

$$|\hat{\beta}_0 - \beta_0^*|^2 + \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2 \leq r^2\frac{s\log p}{n}.$$

In fact, if

$$|\hat{\beta}_0 - \beta_0^*|^2 + \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2 > r^2\frac{s\log p}{n},$$

let

$$t_0 := \frac{r\sqrt{\frac{s\log p}{n}}}{\sqrt{|\hat{\beta}_0 - \beta_0^*|^2 + \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2}},$$

then $0 < t_0 < 1$. Further define

$$(\beta_0', \boldsymbol{\beta}') := t_0(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) + (1 - t_0)(\beta_0^*, \boldsymbol{\beta}^*),$$

then we have

$$|\beta_0' - \beta_0^*|^2 + \|\boldsymbol{\beta}' - \boldsymbol{\beta}^*\|_2^2 = r^2\frac{s\log p}{n}.$$

Moreover, since $(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) - (\beta_0, \boldsymbol{\beta}^*) \in \mathcal{S}_{\mathbb{A}}$ under $\mathcal{E}_1 \cap \mathcal{E}_2$ and $\mathcal{S}_{\mathbb{A}}$ is a cone, we know

$$(\beta_0', \boldsymbol{\beta}') - (\beta_0^*, \boldsymbol{\beta}^*) = t_0\big((\hat{\beta}_0, \hat{\boldsymbol{\beta}}) - (\beta_0^*, \boldsymbol{\beta}^*)\big) \in \mathcal{S}_{\mathbb{A}}.$$

This means that under $\mathcal{E}_1 \cap \mathcal{E}_2$,

$$(\beta_0', \boldsymbol{\beta}') \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r).$$

By convexity of $F(\cdot)$ and norm functions and by (A.1.4), we further have

$$\begin{aligned}&t_0\Big(F(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) + \lambda_0\|\hat{\boldsymbol{\beta}}\|_2^2 + \lambda\|\hat{\boldsymbol{\beta}}\|_1\Big)\\ &+ (1 - t_0)\Big(F(\beta_0^*, \boldsymbol{\beta}^*) + \lambda_0\|\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_1\Big)\\ &\geq F(\beta_0', \boldsymbol{\beta}') + \lambda_0\|\boldsymbol{\beta}'\|_2^2 + \lambda\|\boldsymbol{\beta}'\|_1\\ &\geq \inf_{(\beta_0,\boldsymbol{\beta})\in(\beta_0^*,\boldsymbol{\beta}^*)+\mathbb{C}(r)} F(\beta_0, \boldsymbol{\beta}) + \lambda_0\|\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\\ &> F(\beta_0^*, \boldsymbol{\beta}^*) + \lambda_0\|\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_1\end{aligned}$$

under $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{G}_T$. The above inequality implies

$$\begin{aligned}&F(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) + \lambda_0\|\hat{\boldsymbol{\beta}}\|_2^2 + \lambda\|\hat{\boldsymbol{\beta}}\|_1\\ &> F(\beta_0^*, \boldsymbol{\beta}^*) + \lambda_0\|\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_1,\end{aligned}$$

which is a contradiction with the definition of $(\hat{\beta}_0, \hat{\boldsymbol{\beta}})$. So the claim is proved. By union bound, previous results and choice of tuning parameters, we have

$$\begin{aligned}&\mathrm{P}\big((\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{G}_T)^{\mathrm{c}}\big)\\ &\leq \mathrm{P}(\mathcal{E}_1^{\mathrm{c}}) + \mathrm{P}(\mathcal{E}_2^{\mathrm{c}}) + \mathrm{P}(\mathcal{G}_T^{\mathrm{c}})\\ &\leq 2\exp\Big\{-\frac{n\lambda^2}{8}\Big\} + 2p\mathrm{e}^{-\frac{1}{\eta_0}(\frac{\lambda^2}{16m_0^2}\wedge\frac{\lambda}{4m_0})n} + \mathrm{P}(\mathcal{G}_T^{\mathrm{c}})\\ &\leq 2p^{-\frac{c_0^2}{8}} + 2p\mathrm{e}^{-\frac{1}{\eta_0}\frac{\lambda^2 n}{16m_0^2}} + 2p\mathrm{e}^{-\frac{1}{\eta_0}\frac{\lambda n}{4m_0}} + \mathrm{P}(\mathcal{G}_T^{\mathrm{c}})\\ &\leq 2p^{-\frac{c_0^2}{8}} + 2p^{-\left(\frac{1}{\eta_0}\frac{c_0^2}{16m_0^2}-1\right)}\\ &+ 2\mathrm{e}^{-\sqrt{n\log p}\left(\frac{1}{\eta_0}\frac{c_0}{4m_0}-\sqrt{\frac{\log p}{n}}\right)} + \mathrm{P}(\mathcal{G}_T^{\mathrm{c}}).\end{aligned}$$

Since $\frac{\log p}{n} = o(1)$, as long as c_0 is large enough (for instance $c_0 > 4\sqrt{2\eta_0}m_0$), we have

$$\lim_{T\to\infty}\limsup_{n\to\infty}\mathrm{P}\big((\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{G}_T)^{\mathrm{c}}\big) = 0.$$

Combining this result and the previous claim, the proof of Theorem 1 is finished.

□

B. Proof of Lemma 1

It is seen that $L_h^G(v)$ is twice differentiable with

$$L_h^{G\prime\prime}(v) = \frac{1}{\sqrt{2\pi}h}\exp\Big\{-\frac{(1-v)^2}{2h^2}\Big\} \leq \frac{1}{\sqrt{2\pi}h}. \qquad \text{(A.2.1)}$$

Thus inequality (IV.1) is obtained due to the mean value theorem.

We then prove inequality (IV.2). The inequality is trivial when $v_1 < v_2 \leq 1 - h$ or $v_2 > v_1 \geq 1 + h$. When $1 - h < v_1 < v_2 < 1 + h$, since L_h^E is twice differentiable between $1 - h$ and $1 + h$, we see

$$|L_h^{E\prime}(v_1) - L_h^{E\prime}(v_2)| < \sup_{v\in(1-h,1+h)}|L_h^{E\prime\prime}(v)||v_1 - v_2|,$$

and

$$\begin{aligned}&\sup_{v\in(1-h,1+h)}|L_h^{E\prime\prime}(v)|\\ &= \sup_{v\in(1-h,1+h)}\left|\frac{3(h^2 - (1-u)^2)}{4h^3}\right|\\ &< \frac{3}{4h}.\end{aligned}$$

When $v_1 \le 1-h$ and $v_2 \ge 1+h$,

$$|L_h^{E\prime}(v_1) - L_h^{E\prime}(v_2)| < 1 < \frac{3}{4h}(2h) \le \frac{3}{4h}|v_1 - v_2|.$$

When $v_1 \le 1-h$ and $1-h < v_2 < 1+h$,

$$\begin{aligned} |L_h^{E\prime}(v_1) - L_h^{E\prime}(v_2)| &= \left|1 - \frac{(1-v_2+h)^2(2h-1+v_2)}{4h^3}\right| \\ &< \frac{3}{4h}|1-h-v_2| \\ &\le \frac{3}{4h}|v_1 - v_2|, \end{aligned}$$

where the second to the last inequality is due to

$$\begin{aligned} &\sup_{v_2\in(1-h,1+h)} \frac{\left|1 - \frac{(1-v_2+h)^2(2h-1+v_2)}{4h^3}\right|}{|1-h-v_2|} \\ &\le \frac{9}{16h} < \frac{3}{4h}. \end{aligned}$$

When $1-h < v_1 < 1+h$ and $v_2 \ge 1+h$,

$$\begin{aligned} |L_h^{E\prime}(v_1) - L_h^{E\prime}(v_2)| &= \left|\frac{(1-v_1+h)^2(2h-1+v_1)}{4h^3}\right| \\ &< \frac{3}{4h}|v_1 - (1+h)| \\ &\le \frac{3}{4h}|v_1 - v_2|, \end{aligned}$$

where the second to the last inequality is due to

$$\begin{aligned} &\sup_{v_2\in(1-h,1+h)} \frac{\left|\frac{(1-v_1+h)^2(2h-1+v_1)}{4h^3}\right|}{|1-v_1+h|} \\ &\le \frac{9}{16h} < \frac{3}{4h}. \end{aligned}$$

□

C. Iteration Complexity Analysis of the GCD Algorithm

a) Notation: For a vector $\mathbf{v} = (v_1, \ldots, v_d)^{\mathrm{T}} \in \mathbb{R}^d$ and a univariate function $u(\cdot)$, we write $u(\mathbf{v}) = (u(v_1), \ldots, u(v_d))^{\mathrm{T}}$. Also, denote the subvector of $\mathbf{v}$ with its kth component removed by $\mathbf{v}_{-k} = (v_1, \ldots, v_{k-1}, v_{k+1}, \ldots, v_d)^{\mathrm{T}}$ and recover $\mathbf{v}$ from $\mathbf{v}_{-k}$ by $\mathbf{v} = [v_k, \mathbf{v}_{-k}]$. We also let ∂h be the sub-differential of a nonsmooth convex function h [see e.g., 2].

b) Iteration complexity analysis: Without loss of generality, we focus solely on the GCD algorithm for solving the weighted lasso penalized DCSVM

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \sum_{i=1}^n L_h(y_i \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \sum_{k=1}^p w_k|\beta_k|, \tag{A.3.1}$$

where $w_k \ge 0$ are the weights of the penalty. Indeed, this formulation covers all the sparsity patterns in Section II-C. Also, the intercept term β_0 can be absorbed into the formulation by setting $x_{i1} = 1$ for $i = 1, \ldots, n$ and $w_1 = 0$. For ease of exposition, let us rewrite (A.3.1) as the following unconstrained optimization problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) + \sum_{k=1}^p h_k(\beta_k), \tag{A.3.2}$$

where $g(\boldsymbol{\beta}) = \sum_{i=1}^n L_h(y_i \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})$ is smooth convex in $\boldsymbol{\beta} \in \mathbb{R}^p$, while $h_k(\beta_k) = w_k|\beta_k|$ is nonsmooth convex in β_k for each $k = 1, \ldots, p$. Let $h(\boldsymbol{\beta}) = \sum_{k=1}^p h_k(\beta_k)$. Note that $\nabla g(\boldsymbol{\beta}) = \sum_{i=1}^n y_i L_h'(y_i \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})\mathbf{x}_i$ with $\nabla_k g(\boldsymbol{\beta}) = \sum_{i=1}^n y_i L_h'(y_i \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) x_{ik}$ for $k = 1, \ldots, p$. Let $\rho_{\max} = \lambda_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X}) = \lambda_{\max}(\mathbf{X}\mathbf{X}^{\mathrm{T}})$ and $\boldsymbol{\ell}(\boldsymbol{\beta}) = (\ell_1(\boldsymbol{\beta}), \ldots, \ell_n(\boldsymbol{\beta}))^{\mathrm{T}}$ with $\ell_i(\boldsymbol{\beta}) = L_h'(y_i \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})$ for $i = 1, \ldots, n$. Denote by $\circ$ the Hadamard product. It follows that

$$\begin{aligned} &\|\nabla g(\boldsymbol{\beta}) - \nabla g(\boldsymbol{\beta}')\| \\ &= \|\mathbf{X}^{\mathrm{T}}[\mathbf{y} \circ (\boldsymbol{\ell}(\boldsymbol{\beta}) - \boldsymbol{\ell}(\boldsymbol{\beta}'))]\| \\ &\le \rho_{\max}^{1/2}\|\boldsymbol{\ell}(\boldsymbol{\beta}) - \boldsymbol{\ell}(\boldsymbol{\beta}')\| \\ &\le \rho_{\max}^{1/2} c_h \|\mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}')\| \\ &\le c_h \rho_{\max}\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|, \end{aligned}$$

which implies that the gradient of $g(\cdot)$ is uniformly Lipschitz continuous with Lipschitz constant $L = c_h\rho_{\max}$. When restricted to each coordinate, we have

$$\begin{aligned} &|\nabla_k g([\beta_k, \boldsymbol{\beta}_{-k}]) - \nabla_k g([\beta_k', \boldsymbol{\beta}_{-k}])| \\ &\le c_h\|\mathbf{X}_k\|^2|\beta_k - \beta_k'|,\ k = 1, \ldots, p, \end{aligned}$$

which implies that the gradient of $g(\cdot)$ is coordinate-wise uniformly Lipschitz continuous with Lipschitz constants $L_k = c_h\|\mathbf{X}_k\|^2$, $k = 1, \ldots, p$.

In the GCD (cyclic coordinate descent) algorithm, let $\boldsymbol{\beta}^r$ be the update of $\boldsymbol{\beta}$ after the rth cycle, $r \ge 0$. For ease of notation, denote

$$\begin{aligned} \mathbf{b}_k^{r+1} &= (\beta_1^{r+1}, \ldots, \beta_{k-1}^{r+1}, \beta_k^r, \beta_{k+1}^r, \ldots, \beta_p^r)^{\mathrm{T}}, \\ \mathbf{b}_{-k}^{r+1} &= (\beta_1^{r+1}, \ldots, \beta_{k-1}^{r+1}, \beta_{k+1}^r, \ldots, \beta_p^r)^{\mathrm{T}}, \end{aligned}$$

for $k = 1, \ldots, p$. Clearly, we have $\mathbf{b}_1^{r+1} = \boldsymbol{\beta}^r$ and $\mathbf{b}_{p+1}^{r+1} = \boldsymbol{\beta}^{r+1}$. Note that in the proximal gradient update,

$$\beta_k^{r+1} := \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^r - L_k^{-1}\nabla_k g([\beta_k^r, \mathbf{b}_{-k}^{r+1}]))$$

is equivalent to

$$\beta_k^{r+1} := \operatorname*{argmin}_{\beta_k} u_k(\beta_k; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) + h_k(\beta_k),$$

where the proximity operator $\mathbf{prox}$ does the soft-thresholding [22] and

$$\begin{aligned} &u_k(\beta_k; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) \\ &= g([\beta_k^r, \mathbf{b}_{-k}^{r+1}]) + \nabla_k g([\beta_k^r, \mathbf{b}_{-k}^{r+1}])(\beta_k - \beta_k^r) \\ &+ \frac{L_k}{2}(\beta_k - \beta_k^r)^2 \end{aligned}$$

is a quadratic majorization function of $\hat{g}(\beta_k; \mathbf{b}_{-k}^{r+1}) := g([\beta_k, \mathbf{b}_{-k}^{r+1}])$ at β_k^r. It is easy to see that $u_k(\beta_k; [\beta_k^r, \mathbf{b}_{-k}^{r+1}])$ is strongly convex in β_k. By the optimality of β_k^{r+1}, there exists $\zeta_k^{r+1} \in \partial h_k(\beta_k^{r+1})$ such that

$$(\nabla u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) + \zeta_k^{r+1})(\beta_k - \beta_k^{r+1}) \ge 0,\ \forall \beta_k. \tag{A.3.3}$$

Our analysis will be divided into three parts: the sufficient descent step, the cost-to-go estimate step, and the local error bound step. Similar techniques can be found in [20], [21], [36], and [15].

c) Sufficient descent: Consider the proximal gradient method applied to solving the following problem

$$\min_{\beta_k \in \mathbb{R}} f([\beta_k, \mathbf{b}_{-k}^{r+1}]) = g([\beta_k, \mathbf{b}_{-k}^{r+1}]) + h_k(\beta_k),$$

we have by (A.3.3)

$$\begin{aligned}
&f(\mathbf{b}_k^{r+1}) - f(\mathbf{b}_{k+1}^{r+1})\\
&= f([\beta_k^r, \mathbf{b}_{-k}^{r+1}]) - f([\beta_k^{r+1}, \mathbf{b}_{-k}^{r+1}])\\
&\geq u_k(\beta_k^r; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) - u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}])\\
&+ h_k(\beta_k^r) - h_k(\beta_k^{r+1})\\
&= \nabla_k u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}])(\beta_k^r - \beta_k^{r+1})\\
&+ h_k(\beta_k^r) - h_k(\beta_k^{r+1}) + \frac{L_k}{2}(\beta_k^r - \beta_k^{r+1})^2\\
&\geq (\nabla_k u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) + \zeta_k^{r+1})(\beta_k^r - \beta_k^{r+1})\\
&+ \frac{L_k}{2}(\beta_k^r - \beta_k^{r+1})^2\\
&\geq \frac{L_k}{2}(\beta_k^r - \beta_k^{r+1})^2. \qquad \text{(A.3.4)}
\end{aligned}$$

It follows that

$$\begin{aligned}
&f(\boldsymbol{\beta}^r) - f(\boldsymbol{\beta}^{r+1})\\
&= \sum_{k=1}^{p} \left[f(\mathbf{b}_k^{r+1}) - f(\mathbf{b}_{k+1}^{r+1})\right]\\
&\geq \frac{\underline{L}}{2}\|\boldsymbol{\beta}^r - \boldsymbol{\beta}^{r+1}\|^2, \qquad \text{(A.3.5)}
\end{aligned}$$

where $\underline{L} = \min_{1 \leq k \leq p} L_k = c_h \min_{1 \leq k \leq p} \|\mathbf{x}_k\|^2$.

d) Cost-to-go estimate: Let $\mathscr{X}^* := \{\boldsymbol{\beta}^* | f(\boldsymbol{\beta}^*) = \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})\}$ be the optimal solution set of problem (A.3.2). Let $\bar{\boldsymbol{\beta}}^r \in \mathscr{X}^*$ be the point in $\mathscr{X}^*$ such that $\mathrm{d}_{\mathscr{X}^*}(\boldsymbol{\beta}^r) := \min_{\boldsymbol{\beta} \in \mathscr{X}^*} \|\boldsymbol{\beta} - \boldsymbol{\beta}^r\| = \|\bar{\boldsymbol{\beta}}^r - \boldsymbol{\beta}^r\|$. By optimality of

$$\beta_k^{r+1} = \operatorname*{argmin}_{\beta_k \in \mathbb{R}} u_k(\beta_k; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) + h_k(\beta_k),$$

one has

$$\begin{aligned}
&h(\beta_k^{r+1}) - h(\bar{\beta}_k^r) + \nabla_k g([\beta_k^r, \mathbf{b}_{-k}^{r+1}])(\beta_k^{r+1} - \bar{\beta}_k^r)\\
&\leq \frac{L_k}{2}(\bar{\beta}_k^r - \beta_k^r)^2.
\end{aligned}$$

By the mean value theorem, there exists $\lambda \in [0,1]$ and $\boldsymbol{\xi}^r = \lambda \boldsymbol{\beta}^{r+1} + (1-\lambda)\bar{\boldsymbol{\beta}}^r$ such that

$$g(\boldsymbol{\beta}^{r+1}) - g(\bar{\boldsymbol{\beta}}^r) = \langle \nabla g(\boldsymbol{\xi}^r), \boldsymbol{\beta}^{r+1} - \bar{\boldsymbol{\beta}}^r \rangle.$$

It follows that

$$\begin{aligned}
&f(\boldsymbol{\beta}^{r+1}) - f(\bar{\boldsymbol{\beta}}^r)\\
&= g(\boldsymbol{\beta}^{r+1}) - g(\bar{\boldsymbol{\beta}}^r) + \sum_{k=1}^{p}\left[h_k(\beta_k^{r+1}) - h_k(\bar{\beta}_k^r)\right]\\
&= \sum_{k=1}^{p}\left[\nabla_k g(\boldsymbol{\xi}^r)(\beta_k^{r+1} - \bar{\beta}_k^r) + h_k(\beta_k^{r+1}) - h_k(\bar{\beta}_k^r)\right]\\
&= \sum_{k=1}^{p}\left[\nabla_k g([\beta_k^r, \mathbf{b}_{-k}^{r+1}])(\beta_k^{r+1} - \bar{\beta}_k^r) + h_k(\beta_k^{r+1}) - h_k(\bar{\beta}_k^r)\right.\\
&\quad \left. + \big(\nabla_k g(\boldsymbol{\xi}^r) - \nabla_k g([\beta_k^r, \mathbf{b}_{-k}^{r+1}])\big)(\beta_k^{r+1} - \bar{\beta}_k^r)\right]\\
&\leq \sum_{k=1}^{p}\left[\frac{L_k}{2}(\bar{\beta}_k^r - \beta_k^r)^2\right.\\
&\left. + \big(\nabla_k g(\boldsymbol{\xi}^r) - \nabla_k g([\beta_k^r, \mathbf{b}_{-k}^{r+1}])\big)(\beta_k^{r+1} - \bar{\beta}_k^r)\right].
\end{aligned}$$

By the fact that $\nabla g(\cdot)$ is Lipschitz continuous, it is implied that

$$\begin{aligned}
&\left(\sum_{k=1}^{p}\big(\nabla_k g(\boldsymbol{\xi}^r) - \nabla_k g([\beta_k^r, \mathbf{b}_{-k}^{r+1}])\big)(\beta_k^{r+1} - \bar{\beta}_k^r)\right)^2\\
&\leq \left(\sum_{k=1}^{p}\|\nabla g(\boldsymbol{\xi}^r) - \nabla g([\beta_k^r, \mathbf{b}_{-k}^{r+1}])\|^2\right)\left(\sum_{k=1}^{p}(\beta_k^{r+1} - \bar{\beta}_k^r)^2\right)\\
&\leq \left(\sum_{k=1}^{p} L^2\|\boldsymbol{\xi}^r - [\beta_k^r, \mathbf{b}_{-k}^{r+1}]\|^2\right)\|\boldsymbol{\beta}^{r+1} - \bar{\boldsymbol{\beta}}^r\|^2\\
&= \left(\sum_{k=1}^{p} L^2\|\lambda(\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r) + (1-\lambda)(\bar{\boldsymbol{\beta}}^r - \boldsymbol{\beta}^r) + \boldsymbol{\beta}^r\right.\\
&\left. - [\beta_k^r, \mathbf{b}_{-k}^{r+1}]\|^2\right) \cdot 2(\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 + \|\boldsymbol{\beta}^r - \bar{\boldsymbol{\beta}}^r\|^2)\\
&\leq 12(p+1)L^2\left[\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 + \|\boldsymbol{\beta}^r - \bar{\boldsymbol{\beta}}^r\|^2\right]^2\\
&\leq 25pL^2\left[\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 + \mathrm{d}_{\mathscr{X}^*}^2(\boldsymbol{\beta}^r)\right]^2.
\end{aligned}$$

It follows that

$$\begin{aligned}
&f(\boldsymbol{\beta}^{r+1}) - f(\bar{\boldsymbol{\beta}}^r)\\
&\leq (5L\sqrt{p} + \bar{L})\left[\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 + \mathrm{d}_{\mathscr{X}^*}^2(\boldsymbol{\beta}^r)\right], \qquad \text{(A.3.6)}
\end{aligned}$$

where $\bar{L} = \max_{1 \leq k \leq p} L_k = c_h \max_{1 \leq k \leq p} \|\mathbf{x}_k\|^2$.

e) Local error bound: Let $\mathbf{d}_{\mathscr{X}^*}(\boldsymbol{\beta}) \equiv \min_{\boldsymbol{\beta}^* \in \mathscr{X}^*} \|\boldsymbol{\beta}^* - \boldsymbol{\beta}\|$. Here we handle the Gaussian and Epanechnikov kernels separately. For the Gaussian kernel, that is, when $L_h(\cdot) = L_h^G(\cdot)$, according to (A.3.4) and (A.3.5), the GCD algorithm is descending along its iterations. We can thus restrict the domain of $\boldsymbol{\beta}$ to the sublevel set $\mathcal{L}_0 = \{\boldsymbol{\beta} : f(\boldsymbol{\beta}) \leq f(\mathbf{0})\}$. Let $\eta_i = \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}$ for $i = 1, \ldots, n$. It follows that the set $\mathcal{C}_0 = \{\boldsymbol{\eta} = (\eta_i, 1 \leq i \leq n)^{\mathrm{T}} : \boldsymbol{\beta} \in \mathcal{L}_0\}$ is convex compact. Therefore, for all $\boldsymbol{\beta} \in \mathcal{L}_0$, η_i is bounded by $\eta_{\max}$, where $\eta_{\max} = \max_{1 \leq i \leq n} \sup_{\boldsymbol{\beta} \in \mathcal{L}_0} |\eta_i| < \infty$. Note that the function $p(\mathbf{z}) = \sum_{i=1}^{n} L_h^G(y_i z_i)$ is strongly convex in $\mathbf{z} \in \mathcal{C}_0$ by (A.2.1). We can see that $g(\boldsymbol{\beta}) = p(\mathbf{X}\boldsymbol{\beta})$. It follows from [36] that for any $\xi \geq \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$, there exist $\kappa, \varepsilon > 0$ such that

$$\mathrm{d}_{\mathscr{X}^*}(\boldsymbol{\beta}) \leq \kappa\|\boldsymbol{\beta} - \mathbf{prox}_h(\boldsymbol{\beta} - \nabla g(\boldsymbol{\beta}))\|, \qquad \text{(A.3.7)}$$

for all $\boldsymbol{\beta}$ such that $\|\boldsymbol{\beta} - \mathbf{prox}_h(\boldsymbol{\beta} - \nabla g(\boldsymbol{\beta}))\| \leq \varepsilon$ and $f(\boldsymbol{\beta}) \leq \xi$.

For the Epanechnikov kernel, that is, when $L_h(\cdot) = L_h^E(\cdot)$, one needs to add an additional ridge penalty $\mu\|\boldsymbol{\beta}\|^2$ for some small $\mu > 0$ in order to achieve strong optimality. Thus, when the Epanechnikov kernel is used, we instead consider the following problem

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \sum_{i=1}^{n} L_h^E(y_i \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \sum_{k=1}^{p} w_k|\beta_k| + \mu\|\boldsymbol{\beta}\|^2$$

and solve it using the GCD algorithm.

As a summary, we show in the following theorem that the GCD algorithm converges at least linearly.

Theorem 2: The GCD algorithm converges at least linearly to a solution in $\mathscr{X}^*$.

Proof: We first show that there exists some $\sigma > 0$ such that

$$\|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r - \nabla g(\boldsymbol{\beta}^r))\| \leq \sigma \|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|, \; \forall r \geq 1. \tag{A.3.8}$$

For any $r \geq 1$ and any $1 \leq k \leq p$, by the optimality of

$$\beta_k^{r+1} = \operatorname*{argmin}_{\beta_k} u_k(\beta_k; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) + h_k(\beta_k),$$

we have

$$\beta_k^{r+1} = \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^{r+1} - L_k^{-1}\nabla u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}])).$$

Let $\hat{L}_k = \max(1, L_k)$ and $\tilde{L}_k = \max(1, L_k^{-1})$. It follows from Lemma 4.3 of [16] that

$$\begin{aligned}
&|\beta_k^r - \mathbf{prox}_{h_k}(\beta_k^r - \nabla_k g(\boldsymbol{\beta}^r))| \\
&\leq \hat{L}_k |\beta_k^r - \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^r - L_k^{-1}\nabla_k g(\boldsymbol{\beta}^r))| \\
&\leq \hat{L}_k \big[|\beta_k^{r+1} - \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^r - L_k^{-1}\nabla_k g(\boldsymbol{\beta}^r))| \\
&+ |\beta_k^{r+1} - \beta_k^r| \big] \\
&\leq \hat{L}_k \big[|\mathbf{prox}_{L_k^{-1}h_k}(\beta_k^{r+1} - L_k^{-1}\nabla u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}])) \\
&- \mathbf{prox}_{L_k^{-1}h_k}(\beta_k^r - L_k^{-1}\nabla_k g(\boldsymbol{\beta}^r))| + |\beta_k^{r+1} - \beta_k^r| \big] \\
&\leq 2\hat{L}_k |\beta_k^{r+1} - \beta_k^r| \\
&+ \hat{L}_k L_k^{-1} |\nabla u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) - \nabla_k g(\boldsymbol{\beta}^r)| \\
&\leq 3\hat{L}_k |\beta_k^{r+1} - \beta_k^r| + \tilde{L}_k \|\nabla g([\beta_k^r, \mathbf{b}_{-k}^{r+1}]) - \nabla g(\boldsymbol{\beta}^r)\| \\
&\leq (3\hat{L}_k + L\tilde{L}_k) \|\boldsymbol{\beta}_k^{r+1} - \boldsymbol{\beta}_k^r\|.
\end{aligned}$$

It follows that

$$\|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r - \nabla g(\boldsymbol{\beta}^r))\| \leq (3\hat{L} + L\tilde{L})\sqrt{p}\|\boldsymbol{\beta}_k^{r+1} - \boldsymbol{\beta}_k^r\|,$$

where $\hat{L} = \max(1, \bar{L})$ and $\tilde{L} = \max(1, \underline{L}^{-1})$. Therefore, when we take $\sigma = (3\hat{L} + L\tilde{L})\sqrt{p}$, we get the desired result in (A.3.8). Note that the sufficient descent property (A.3.5) implies that $\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\| \to 0$ as $r \to \infty$. It follows from (A.3.8) that $\|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r - \nabla g(\boldsymbol{\beta}^r))\| \to 0$ as $r \to \infty$. Thus, by (A.3.7) we have $\mathrm{d}_{\mathscr{X}^*}(\boldsymbol{\beta}^r) \to 0$ as $r \to \infty$. Consequently, from (A.3.6) it implies that $f(\boldsymbol{\beta}^r) \to f^* := \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$, which shows that the GCD algorithm converges to the global minimum.

Now let $c_1 = \underline{L}(2B)^{-1}$, $c_2 = 5L\sqrt{p} + \bar{L}$, and $\Delta^r = f(\boldsymbol{\beta}^r) - f^*$. By the local error bound (A.3.7) and the cost-to-go estimate (A.3.6), we obtain

$$\begin{aligned}
\Delta^{r+1} &\leq c_2 \big[\mathrm{d}^2_{\mathscr{X}^*}(\boldsymbol{\beta}^r) + \|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 \big] \\
&\leq c_2\kappa^2 \|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r - \nabla g(\boldsymbol{\beta}^r))\|^2 + c_2\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 \\
&\leq (c_2\kappa^2\sigma^2 + c_2)\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|^2 \\
&\leq (c_2\kappa^2\sigma^2 + c_2)c_1^{-1}[f(\boldsymbol{\beta}^r) - f(\boldsymbol{\beta}^{r+1})] \\
&= (c_2\kappa^2\sigma^2 + c_2)c_1^{-1}(\Delta^r - \Delta^{r+1}),
\end{aligned}$$

which implies that

$$\Delta^{r+1} \leq \frac{c_3}{1 + c_3}\Delta^r, \tag{A.3.9}$$

where $c_3 = (c_2\kappa^2\sigma^2 + c_2)c_1^{-1}$. We can see from (A.3.9) that $f(\boldsymbol{\beta}^r)$ approaches f^* with at least linear rate of convergence. From (A.3.5) again, this further implies that the sequence $\{\boldsymbol{\beta}^r\}$ converges at least linearly. □

Acknowledgment

The authors would like to thank the Associate Editor and the anonymous referees for their useful suggestions.

References

[1] D. P. Bertsekas, “Stochastic optimization problems with nondifferentiable cost functionals,” *J. Optim. Theory Appl.*, vol. 12, pp. 218–231, Aug. 1973.

[2] D. P. Bertsekas, *Nonlinear Programming*. Nashua, NH, USA: Athena Scientific, 1999.

[3] P. Borah and D. Gupta, “Affinity and transformed class probability-based fuzzy least squares support vector machines,” *Fuzzy Sets Syst.*, vol. 443, pp. 203–235, Aug. 2022.

[4] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proc. 5th Annu. Workshop Comput. Learn. Theory (COLT)*, 1992, pp. 144–152.

[5] P. Bühlmann and S. Van De Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Berlin, Germany: Springer, 2011.

[6] D. Donoho, “High-dimensional data analysis: The curses and blessings of dimensionality,” *AMS Math. Challenges Lect.*, vol. 1, pp. 1–33, Aug. 2000.

[7] J. Fan, R. Li, C.-H. Zhang, and H. Zou, *Statistical Foundations of Data Science*. London, U.K.: Chapman & Hall/CRC, 2020.

[8] M. Fernandes, E. Guerre, and E. Horta, “Smoothing quantile regressions,” *J. Bus. Econ. Statist.*, vol. 39, no. 1, pp. 338–357, Jan. 2021.

[9] J. Friedman, T. Hastie, and R. Tibshirani, “Regularization paths for generalized linear models via coordinate descent,” *J. Stat. Softw.*, vol. 33, pp. 1–22, Aug. 2010.

[10] K. O. Friedrichs, “The identity of weak and strong extensions of differential operators,” *Trans. Amer. Math. Soc.*, vol. 55, pp. 132–151, Jan. 1944.

[11] F. Götze, H. Sambale, and A. Sinulis, “Concentration inequalities for polynomials in α-sub-exponential random variables,” *Electron. J. Probab.*, vol. 26, pp. 1–22, Jan. 2021.

[12] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. New York, NY, USA: Springer-Verlag, 2009.

[13] B. B. Hazarika and D. Gupta, “Density weighted twin support vector machines for binary class imbalance learning,” *Neural Process. Lett.*, vol. 54, no. 2, pp. 1091–1130, Apr. 2022.

[14] X. He, X. Pan, K. M. Tan, and W.-X. Zhou, “Smoothed quantile regression with large-scale inference,” *J. Econometrics*, Aug. 2021, doi: 10.1016/j.jeconom.2021.07.010.

[15] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo, “Iteration complexity analysis of block coordinate descent methods,” *Math. Program.*, vol. 163, nos. 1–2, pp. 85–114, May 2017.

[16] M. Kadkhodaie, M. Sanjabi, and Z.-Q. Luo, “On the linear convergence of the approximate proximal splitting method for non-smooth convex optimization,” *J. Oper. Res. Soc. China*, vol. 2, no. 2, pp. 123–141, Jun. 2014.

[17] B. Kumar and D. Gupta, “Universum based Lagrangian twin bounded support vector machine to classify EEG signals,” *Comput. Methods Programs Biomed.*, vol. 208, Sep. 2021, Art. no. 106244.

[18] M. Ledoux and M. Talagrand, “Probability in Banach spaces: Isoperimetry and processes,” in *Ergebnisse der Mathematik und ihrer Grenzgebiete* (A Series of Modern Surveys in Mathematics), vol. 23. Berlin, Germany: Springer, 1991.

[19] A. Liaw and M. Wiener, “Classification and regression by randomforest,” *R News*, vol. 2, pp. 18–22, Dec. 2002.

[20] Z.-Q. Luo and P. Tseng, “On the linear convergence of descent methods for convex essentially smooth minimization,” *SIAM J. Control Optim.*, vol. 30, no. 2, pp. 408–425, Mar. 1992.

[21] Z.-Q. Luo and P. Tseng, “Error bounds and convergence analysis of feasible descent methods: A general approach,” *Ann. Operations Res.*, vols. 46–47, no. 1, pp. 157–178, Mar. 1993.

[22] N. Parikh and S. Boyd, “Proximal algorithms,” *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–231, 2013.

[23] B. Peng, L. Wang, and Y. Wu, "An error bound for L_1-norm support vector machine coefficients in ultra-high dimension," *J. Mach. Learn. Res.*, vol. 17, pp. 8279–8304, Dec. 2016.

[24] R. Y. Rubinstein, "Smoothed functionals in stochastic optimization," *Math. Oper. Res.*, vol. 8, no. 1, pp. 26–33, Feb. 1983.

[25] K. M. Tan, L. Wang, and W. Zhou, "High-dimensional quantile regression: Convolution smoothing and concave regularization," *J. Roy. Stat. Soc., B Stat. Methodol.*, vol. 84, no. 1, pp. 205–233, Feb. 2022.

[26] R. Tibshirani et al., "Strong rules for discarding predictors in lasso-type problems," *J. Roy. Stat. Soc., B Stat. Methodol.*, vol. 74, pp. 245–266, Nov. 2010.

[27] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

[28] A. W. Van Der Vaart and J. Wellner, *Weak Convergence and Empirical Processes: With Applications to Statistics*. Berlin, Germany: Springer, 1996.

[29] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.

[30] W. N. Venables and B. D. Ripley, *Modern Applied Statistics With S*, 4th ed. New York, NY, USA: Springer, 2002.

[31] R. Vershynin, *High-Dimensional Probability: An Introduction With Applications in Data Science*, vol. 47. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[32] B. Wang and H. Zou, "A multicategory kernel distance weighted discrimination method for multiclass classification," *Technometrics*, vol. 61, pp. 396–408, Sep. 2019.

[33] L. Wang, J. Zhu, and H. Zou, "The doubly regularized support vector machine," *Statistica Sinica*, vol. 16, pp. 589–616, Apr. 2006.

[34] D. M. Witten and R. Tibshirani, "Penalized classification using Fisher's linear discriminant," *J. Roy. Stat. Soc., B Stat. Methodol.*, vol. 73, no. 5, pp. 753–772, Nov. 2011.

[35] Y. Yang and H. Zou, "An efficient algorithm for computing the HHSVM and its generalizations," *J. Comput. Graph. Statist.*, vol. 22, no. 2, pp. 396–415, Apr. 2013.

[36] H. Zhang, J. Jiang, and Z.-Q. Luo, "On the linear convergence of a proximal gradient method for a class of nonsmooth convex minimization problems," *J. Oper. Res. Soc. China*, vol. 1, no. 2, pp. 163–186, Jun. 2013.

[37] J. Zhu, S. Rosset, R. Tibshirani, and T. Hastie, "1-norm support vector machines," in *Proc. Adv. Neural Inf. Process. Syst.* vol. 16, 2003, pp. 1–8.

[38] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc., B Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, Apr. 2005.

[39] H. Zou, J. Zhu, and T. Hastie, "New multicategory boosting algorithms based on multicategory Fisher-consistent losses," *Ann. Appl. Statist.*, vol. 2, no. 4, p. 1290, Dec. 2008.

Boxiang Wang received the B.S. degree in mathematics and statistics from Nankai University, China, the M.S. degree in statistics from Bowling Green State University, Bowling Green, OH, USA, and the Ph.D. degree in statistics from the University of Minnesota. He is currently an Assistant Professor of statistics at The University of Iowa. His research interests include machine learning, optimization, and data science.

Le Zhou received the B.S. degree in mathematics from the Tsinghua University, Beijing, China, in 2016, and the Ph.D. degree in statistics from the University of Minnesota in 2021. He is currently an Assistant Professor of mathematics at Hong Kong Baptist University. His research interests include high-dimensional data analysis, statistical learning theory, and high-dimensional probability.

Yuwen Gu received the B.S. degree in statistics from the University of Science and Technology of China in 2011 and the Ph.D. degree in statistics from the University of Minnesota in 2017. He is currently an Assistant Professor of statistics at the University of Connecticut. His research interests include high-dimensional statistics, nonparametric statistics, causal inference, and optimization.

Hui Zou received the B.S. and M.S. degrees in physics from the University of Science and Technology of China, Hefei, China, in 1997 and 1999, respectively, the M.S. degree in statistics from Iowa State University in 2001, and the Ph.D. degree in statistics from Stanford University in 2005. He is currently a Professor of statistics at the University of Minnesota. His research interests include high-dimensional statistics, machine learning, and computational statistics. He is a fellow of the American Statistical Association and the Institute of Mathematical Statistics.