Density-Convoluted Support Vector Machines for High-Dimensional Classification

LE ZHOU, BOXIANG WANG, 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 and prove it has order $O_p(\sqrt{\frac{s\log p}{n}})$ under general random design setting. We further develop novel efficient algorithm for computing elastic-net penalized DCSVM. Simulation studies and 8 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 (Donoho et al., 2000). 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

Le Zhou is with the School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA (email: zhou0819@umn.edu).

Boxiang Wang is with the Department of Statistics and Actuarial Science, University of Iowa, Iowa City, IA 52242, USA (email: boxiang-wang@uiowa.edu).

Yuwen Gu is with the Department of Statistics, University of Connecticut, Storrs, CT 06269, USA (email: yuwen.gu@uconn.edu).

Hui Zou is with the School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA (email: zouxx019@umn.edu). Zou is supported in part by NSF grants DMS 1915842 and 2015120.

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 Boser et al. (1992) and Vapnik (1995), 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 (Zhu et al., 2003). 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, Peng et al. (2016) investigated the rate of convergence of the ℓ_1-norm SVM and an error bound of $O(\sqrt{\frac{s\log p}{n}})$ was established in their paper.

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 (Friedman et al., 2010). 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 (Luo and Tseng, 1992; Tseng, 2001). A similar problem under regression context is the quantile regression, in which the check loss is not differentiable (Fan et al., 2020). 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.

Recently, Fernandes et al. (2021) studied an interesting smoothing technique for solving quantile regression with statistical guarantees. Later, Tan et al. (2021) 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 $y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0), i=1,\ldots,n$. 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, then we can take the expectation of the hinge loss function with respect to the distribution determined by the smoothed kernel density estimator. This leads us to a new objective function

$$\frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)\big), \tag{I.3}$$

which we use to replace (I.2). Here h is the bandwidth of kernel density estimator and is used to index the new classifier. The resulting estimator is named as *density convoluted support vector machine (DCSVM)*, since the kernel density estimator has a convolution interpretation. We study the following general form of penalized DCSVM in high dimensions,

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

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 (Yang and Zou, 2013).

In this paper, we first study the theoretical properties of the elastic-net DCSVM. We show that the convergence rate of the elastic-net DCSVM is $O_p(\sqrt{\frac{s\log p}{n}})$ under the general random design setting. Furthermore, we develop novel efficient algorithm for computing elastic-net DCSVM. We use simulation studies and 8 benchmark datasets to demonstrate that elastic-net DCSVM delivers superior classification performance over its competitors, and the computational 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 $\mathbf{A}\subset\{1,\ldots,p\}$, any vector $\mathbf{c}=(c_1,\ldots,c_p)$ 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 $\mathbf{A}^{\mathrm{c}}=\{1,\ldots,p\}\setminus\mathbf{A}$. 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 $\|Z\|_\psi = \inf\{t > 0 : \mathbb{E}[\psi(\frac{|Z|}{t})] \leq 1\}$ as th ψ-Orlicz norm for any random variable Z. In particular, if $p \geq 1$, 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 $\|Z\|_{\psi_p} = \inf\{t > 0 : \mathbb{E}[\mathrm{e}^{\frac{|Z|^p}{t^p}}] \leq 2\}$ where Z is any random variable. For a sequence of random variables $\{Z_n\}_{n\geq 1}$, 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

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. We use $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_p)$ to denote the design matrix, where $\mathbf{X}_j = (x_{1j}, \ldots, x_{nj})^\mathrm{T}$ contains observations for the jth variable, and use $\mathbf{y} = (y_1, \ldots, y_n)^\mathrm{T}$ 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 $\tilde{\mathbf{x}}_i = (1, \mathbf{x}_i^\mathrm{T})^\mathrm{T}$, $i = 1, \ldots, n$. To perform the classification task, the support vector machine (SVM, Vapnik, 1995) seeks a separating hyperplane $\{\mathbf{x} : \beta_0 + \mathbf{x}^\mathrm{T}\boldsymbol{\beta} = 0\}$ where

$$\begin{aligned} \min_{\beta_0, \boldsymbol{\beta}, \xi_i} \quad & \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 \\ \text{subject to} \quad & y_i\left(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta}\right) \geq 1 - \xi_i, \xi_i \geq 0, \sum_{i=1}^n \xi_i \leq c. \end{aligned} \tag{II.1}$$

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).

Let us consider the population version of risk appearing in (II.2), $\mathbb{E}[L\big(y(\mathbf{x}^\mathrm{T}\boldsymbol{\beta} + \beta_0)\big)]$. 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 $\int_{-\infty}^{\infty} L(t)\mathrm{d}F(t; \boldsymbol{\beta}, \beta_0)$. 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 $\hat{F}(t; \boldsymbol{\beta}, \beta_0) = \frac{1}{n}\sum_{i=1}^n 1_{\{y_i(\mathbf{x}_i^\mathrm{T}\boldsymbol{\beta} + \beta_0) \leq t\}}$ is the empirical cdf based on i.i.d. realizations of U. 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 "$*$" stands for convolution.

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

$$\min_{\beta_0,\boldsymbol{\beta}}\sum_{i=1}^{n}L_h(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0))+\lambda_0\|\boldsymbol{\beta}\|_2^2. \tag{II.3}$$

We treat the classifier arisen from the above problem as a new classifier and coin it the density-convoluted SVM (DCSVM).

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* (Friedrichs, 1944) and has also been studied in the optimization community, for example, Bertsekas (1973) and Rubinstein (1983). The method was recently adopted to smooth the quantile regression by He et al. (2021), Fernandes et al. (2021) and Tan et al. (2021).

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 $K(u)=\frac{3}{4}(1-u^2)I(-1\le u\le 1)$, where $I(\cdot)$ is the indicator function,

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

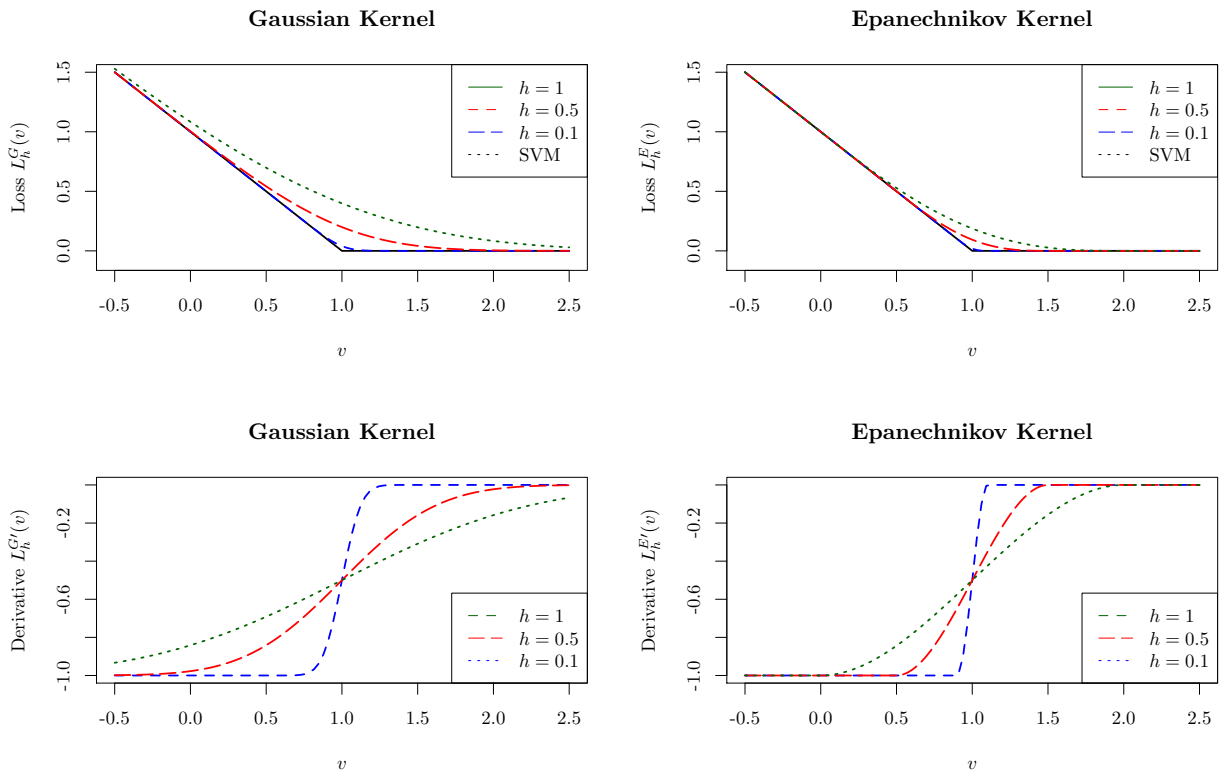

Fig. 1. *Top row: plots of $L_h^G(v)$ and $L_h^E(v)$, the density-convoluted SVM loss functions with Gaussian kernel (left) and Epanechnikov kernels (right). Bottom row: plots of the first-order derivatives, $L_h^{G\prime}(v)$ and $L_h^{E\prime}(v)$.*

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

Intuitively, h should be small such that the DCSVM is very close to the SVM. 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 $(0.25, 3)$. The actual value of C in practice can be determined by cross-validation.

C. Sparse density-convoluted SVM

Define $(\beta_0^*, \boldsymbol{\beta}^*) = \operatorname{argmin}_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^p} \mathbb{E}\big[L_h\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)\big)\big]$. In high dimensions, we consider designing the estimator under a sparsity assumption that $\boldsymbol{\beta}^*$ has many zero components. Let $\mathbb{A} = \{j : \beta_j^* \neq 0, 1 \leq j \leq p\}$ 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 $s_n \geq 1$ 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 additional ℓ_1-penalty term

$$(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) := \operatorname*{argmin}_{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} L_h(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)) + \lambda_0\|\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{II.4}$$

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:

$$(\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) := \underset{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} L_h(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)) + \lambda\|\boldsymbol{\beta}\|_1. \tag{II.5}$$

Borrowing the commonly used terminologies for different penalties in high dimensional literature, we refer to the estimator in (II.4) as elastic-net DCSVM, and refer to the estimator in (II.5) as lasso DCSVM. Note that the lasso DCSVM is a special case of elastic-net DCSVM with $\lambda_0 = 0$. In the statistics literature it is well-known than the elastic-net often yields much improved prediction than the lasso owing to the additional ℓ_2 regularization that can well handle the correlated covariates which occurs often in high-dimensional data. For example, in the case of the SVM Wang et al. (2006) showed that the elastic-net regularized SVM is more accurate than the ℓ_1-norm SVM. Therefore, we focus on the elastic-net penalized DCSVM in theory and in applications.

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. *$\{(y_i, \mathbf{x}_i)\}_{i=1}^n$, $(y, \mathbf{x})$ are independent and identically distributed on $\mathbb{R}\times\mathbb{R}^p$. $\mathbf{x}$ 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*

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

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

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

For any index set $\mathbf{A} \subset \{1,\dots,p\}$, consider the cone $\mathcal{S}_{\mathbf{A}} := \{(\delta,\mathbf{u}) \in \mathbb{R}\times\mathbb{R}^p : \|\mathbf{u}_{\mathbf{A}^{\mathrm{c}}}\|_1 \le 3\|\mathbf{u}_{\mathbf{A}}\|_1 + |\delta|\}$. 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 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 (Peng et al., 2016). Assumption 2, which is a restricted eigenvalue 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 (Bühlmann and Van De Geer, 2011; Fan et al., 2020).

Theorem 1. *Assume assumptions 1-2 hold, and $s\log p = o(n)$. Choose the tuning parameters such that $8\lambda_0\|\beta^*\|_{\max} < \lambda$. 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 DCSVM estimator can achieve the same rate of convergence as the ℓ_1-SVM (Peng et al., 2016). Meanwhile, the sparse DCSVM has better computational efficiency than penalized SVM due to the smoothness of its loss function, as shown in the next section.

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:

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

$$L_h^{E\prime}(v) = \begin{cases} -1, & v \le 1-h, \\ -\dfrac{(1-v+h)^2(2h-(1-v))}{4h^3}, & 1-h < v \le 1+h, \\ 0, & v \ge 1+h. \end{cases}$$

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:

$$L_h(v_1) \le L_h(v_2) + L_h'(v_2)(v_1 - v_2) + \frac{c_h}{2}(v_1 - v_2)^2, \tag{IV.3}$$

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 (Yang and Zou, 2013) 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

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

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

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

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

In the appendix we provide theoretical analysis of the convergence of the generalized coordinate descent algorithm which is not in Yang and Zou (2013). In our implementation, we further apply the strong rule (Tibshirani et al., 2010), warm start, and active set strategy (Friedman et al., 2010) to further accelerate the algorithm.

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

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 (Wang et al., 2006) and elastic-net logistic regression that is fitted using the R package `gcdnet` (Yang and Zou, 2013). 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

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

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.

B. Benchmark data applications

In this section, we demonstrate the performance of DCSVM using several 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 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.

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

$$L_h(t) = \int_{-\infty}^{1} \frac{1-u}{h}K(\frac{t-u}{h})\mathrm{d}u,$$

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	**32.24**	(1.46)	53.26	37.09	(1.59)	8912.87	35.82	(1.65)	219.30
breast	42	22283	**25.90**	(1.64)	51.33	30.38	(2.05)	1946.98	30.76	(2.14)	227.88
colon	62	2000	**18.13**	(1.03)	10.22	18.90	(1.55)	722.48	23.87	(1.51)	27.33
leuk	72	7128	**3.50**	(0.47)	22.98	3.89	(0.51)	1863.23	4.33	(0.61)	115.00
LSVT	126	309	16.01	(0.73)	6.25	16.20	(0.68)	74.20	**15.87**	(0.68)	9.05
malaria	71	22283	**5.37**	(0.68)	85.52	7.60	(1.21)	12046.09	6.80	(0.98)	483.20
ovarian	253	15154	**0.63**	(0.12)	189.22	4.87	(1.23)	14442.87	0.87	(0.14)	964.16
prostate	102	6033	9.25	(0.67)	29.34	**8.98**	(0.50)	2421.20	10.24	(0.61)	116.50

$$L_h'(t) = -\int_{-\infty}^{\frac{1-t}{h}} K(u)\mathrm{d}u,$$
$$L_h''(t) = \frac{1}{h}K(\frac{1-t}{h}),\ \forall t \in \mathbb{R}. \tag{A.0.1}$$

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

Proof of Theorem 1. By definition of the ℓ_1-penalized CRR estimator and triangle inequality, we have

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \hat{\beta}_0)\big) - \frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big) \\
&+ \lambda_0(\|\hat{\boldsymbol{\beta}}\|_2^2 - \|\boldsymbol{\beta}^*\|_2^2) \\
&\leq \lambda(\|\boldsymbol{\beta}^*\|_1 - \|\hat{\boldsymbol{\beta}}\|_1) \\
&\leq \lambda(\|\boldsymbol{\beta}^*_{\mathbb{A}} - \hat{\boldsymbol{\beta}}_{\mathbb{A}}\|_1 + \|\hat{\boldsymbol{\beta}}_{\mathbb{A}}\|_1 - \|\hat{\boldsymbol{\beta}}_{\mathbb{A}}\|_1 - \|\hat{\boldsymbol{\beta}}_{\mathbb{A}^{\mathrm{c}}} - \boldsymbol{\beta}^*_{\mathbb{A}^{\mathrm{c}}}\|_1) \\
&= \lambda(\|\mathbf{u}_{\mathbb{A}}\|_1 - \|\mathbf{u}_{\mathbb{A}^{\mathrm{c}}}\|_1),
\end{aligned} \tag{A.0.2}$$

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

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \hat{\beta}_0)\big) - \frac{1}{n}\sum_{i=1}^{n} L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big) \\
&+ \lambda_0(\|\hat{\boldsymbol{\beta}}\|_2^2 - \|\boldsymbol{\beta}^*\|_2^2) \\
&\geq \frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big) y_i(\hat{\beta}_0 - \beta_0^*) \\
&+ \Big(2\lambda_0\boldsymbol{\beta}^{*\mathrm{T}} + \frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big) y_i\mathbf{x}_i^{\mathrm{T}}\Big)(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) \\
&\geq -\Big|\frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big) y_i\Big| \cdot |\delta| \\
&- \Big\|2\lambda_0\boldsymbol{\beta}^* + \frac{1}{n}\sum_{i=1}^{n} L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big) y_i\mathbf{x}_i\Big\|_\infty (\|\mathbf{u}_{\mathbb{A}}\|_1 + \|\mathbf{u}_{\mathbb{A}^{\mathrm{c}}}\|_1),
\end{aligned} \tag{A.0.3}$$

where $\delta \coloneqq \hat{\beta}_0 - \beta_0^*$. Define event $\mathcal{E}_1 \coloneqq \{|\frac{1}{n}\sum_{i=1}^n L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i| \le \frac{\lambda}{2}\}$ and $\mathcal{E}_2 \coloneqq \{\|2\lambda_0\boldsymbol{\beta}^* + \frac{1}{n}\sum_{i=1}^n L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\mathbf{x}_i\|_\infty \le \frac{\lambda}{2}\}$. Note that $\mathbb{E}\big[L_h'\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y\big] = 0$, and $\big|L_h'\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y\big| \le 1$. So by Hoeffding's inequality,

$$\begin{aligned}\mathrm{P}(\mathcal{E}_1^{\mathrm{c}}) &= \mathrm{P}\Big(\Big|\frac{1}{n}\sum_{i=1}^n L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\Big| > \frac{\lambda}{2}\Big)\\ &\le 2\exp\Big\{-\frac{n\lambda^2}{8}\Big\}. \end{aligned} \tag{A.0.4}$$

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

$$\begin{aligned}\mathrm{P}(\mathcal{E}_2^{\mathrm{c}}) &= \mathrm{P}\Big(\big\|2\lambda_0\boldsymbol{\beta}^* + \frac{1}{n}\sum_{i=1}^n L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\mathbf{x}_i\big\|_\infty > \frac{\lambda}{2}\Big)\\ &\le \mathrm{P}\Big(\big\|\frac{1}{n}\sum_{i=1}^n L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_i\mathbf{x}_i\big\|_\infty > \frac{\lambda}{4}\Big)\\ &\le \sum_{j=1}^p \mathrm{P}\Big(\big|\frac{1}{n}\sum_{i=1}^n L_h'\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)y_ix_{ij}\big| > \frac{\lambda}{4}\Big). \end{aligned} \tag{A.0.5}$$

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 $\|L_h'(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*))y_ix_{ij}\|_{\psi_1} \le m_0$, $\forall i \in \{1,\dots,n\}, \forall j \in \{1,\dots,p\}$. By Theorem 1.4 in Götze et al. (2021), there exists an absolute constant $\eta_0 > 0$ such that

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

So following (A.0.5) 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.0.2) and (A.0.3) 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 Van Der Vaart and Wellner (1996)) and contraction inequality (see for instance, Theorem 4.12 in Ledoux and Talagrand (1991)), $|L_h'(\cdot)| \le 1$ and Cauchy-Schwarz inequality, we have

$$\begin{aligned}&\mathbb{E}[H(r)]\\ &\le 2\mathbb{E}\Big[\sup_{(\beta_0,\boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)} \Big|\frac{1}{n}\sum_{i=1}^n \sigma_i\Big\{L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} + \beta_0)\big)\\ &\qquad\qquad - L_h\big(y_i(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^* + \beta_0^*)\big)\Big\}\Big|\Big]\\ &\le 4\mathbb{E}\Big[\sup_{(\beta_0,\boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)} \Big|\frac{1}{n}\sum_{i=1}^n \sigma_i y_i\big(\mathbf{x}_i^{\mathrm{T}}(\boldsymbol{\beta} - \boldsymbol{\beta}^*) + \beta_0 - \beta_0^*\big)\Big|\Big]\end{aligned}$$

$$\leq \frac{4}{n}\,\mathbb{E}\Big[\Big\|\sum_{i=1}^{n}\sigma_i y_i(1,\mathbf{x}_i^{\mathrm{T}})^{\mathrm{T}}\Big\|_{\infty}\Big]\Big(4\sqrt{s}\cdot r\sqrt{\frac{s\log p}{n}}+2r\sqrt{\frac{s\log p}{n}}\Big). \tag{A.0.6}$$

By assumption 1 and definition of Orlicz norm, we know $\|\sigma_i y_i x_{ij}\|_{\psi_1}=\|x_{ij}\|_{\psi_1}\leq m_0$, $\forall i\in\{1,\ldots,n\},\forall j\in\{1,\ldots,p\}$. Also, it is straightforward to see $\|\sigma_i y_i\|_{\psi_1}=\frac{1}{\log 2}$. By Proposition 2.7.1 in Vershynin (2018), there exists a constant $c_1>0$ such that $\mathbb{E}[\mathrm{e}^{t\sigma_i y_i x_{ij}}]\leq \mathrm{e}^{c_1^2t^2}$ and $\mathbb{E}[\mathrm{e}^{t\sigma_i y_i}]\leq \mathrm{e}^{c_1^2t^2}$ for all $|t|<\frac{1}{c_1}$, $\forall i\in\{1,\ldots,n\},\forall j\in\{1,\ldots,p\}$. By Jensen's inequality, we have for any $0<t<\frac{1}{c_1}$,

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

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

$$\mathbb{E}\Big[\Big\|\sum_{i=1}^{n}\sigma_i y_i(1,\mathbf{x}_i^{\mathrm{T}})^{\mathrm{T}}\Big\|_{\infty}\Big]\leq\frac{\log p+\log 4}{t}+c_1^2tn. \tag{A.0.7}$$

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.0.7) we obtain

$$\mathbb{E}\Big[\Big\|\sum_{i=1}^{n}\sigma_i y_i(1,\mathbf{x}_i^{\mathrm{T}})^{\mathrm{T}}\Big\|_{\infty}\Big]\leq 2c_1\sqrt{(\log p+\log 4)n} \tag{A.0.8}$$

for large enough n. Thus, combining (A.0.6) and (A.0.8) we get

$$\begin{aligned}
&\mathbb{E}[H(r)]\\
&\leq\frac{4}{n}\cdot 2c_1\sqrt{(\log p+\log 4)n}\cdot\Big(4\sqrt{s}\cdot r\sqrt{\frac{s\log p}{n}}+2r\sqrt{\frac{s\log p}{n}}\Big)\\
&\leq\frac{96c_1rs\log p}{n}.
\end{aligned}$$

This implies that $H(r)=O_p(\frac{rs\log p}{n})$. Define event $\mathcal{G}_T:=\{H(r)\leq\frac{Trs\log p}{n}\}$ for any $T>0$, 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

$$\begin{aligned}
&\mathbb{E}[G(\beta_0,\boldsymbol{\beta})]=\mathbb{E}\big[L_h\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}+\beta_0)\big)\big]-\mathbb{E}\big[L_h\big(y(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}^*+\beta_0^*)\big)\big]\\
&=\frac{1}{2}(\beta_0-\beta_0^*,(\boldsymbol{\beta}-\boldsymbol{\beta}^*)^{\mathrm{T}})I\big(\beta_0^*+a(\beta_0-\beta_0^*),\boldsymbol{\beta}^*+a(\boldsymbol{\beta}-\boldsymbol{\beta}^*)\big)\\
&\qquad(\beta_0-\beta_0^*,(\boldsymbol{\beta}-\boldsymbol{\beta}^*)^{\mathrm{T}})^{\mathrm{T}}\\
&\geq\frac{1}{2}\rho\big((\beta_0-\beta_0^*)^2+\|\boldsymbol{\beta}-\boldsymbol{\beta}^*\|_2^2\big)\\
&\geq\frac{1}{2}\rho r^2\frac{s\log p}{n}.
\end{aligned} \tag{A.0.9}$$

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

$$\begin{aligned}
&\lambda\big|\|\boldsymbol{\beta}\|_1 - \|\boldsymbol{\beta}^*\|_1\big| \\
&\le \lambda\|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}}\|_1 + \lambda\|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}^{\mathrm{c}}}\|_1 \\
&\le 4\lambda\|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}}\|_1 + \lambda|\beta_0 - \beta_0^*| \\
&\le 4\lambda\sqrt{s}\|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}}\|_2 + \lambda r\sqrt{\frac{s\log p}{n}} \\
&\le 4\lambda\sqrt{s}r\sqrt{\frac{s\log p}{n}} + \lambda r\sqrt{\frac{s\log p}{n}} \\
&\le 5c_0 sr\frac{\log p}{n},
\end{aligned} \tag{A.0.10}$$

and we also have, by convexity of ℓ_2 norm,

$$\begin{aligned}
&\lambda_0(\|\boldsymbol{\beta}\|_2^2 - \|\boldsymbol{\beta}^*\|_2^2) \ge 2\lambda_0\boldsymbol{\beta}^{*\mathrm{T}}(\boldsymbol{\beta} - \boldsymbol{\beta}^*) \ge -2\lambda_0\|\boldsymbol{\beta}^*\|_{\max}\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\|_1 \\
&\ge -\frac{\lambda}{4}(4\|(\boldsymbol{\beta} - \boldsymbol{\beta}^*)_{\mathbb{A}}\|_1 + |\beta_0 - \beta_0^*|) \\
&\ge -\lambda\sqrt{s}r\sqrt{\frac{s\log p}{n}} - \frac{\lambda}{4}r\sqrt{\frac{s\log p}{n}} \\
&\ge -\frac{2c_0 sr\log p}{n}.
\end{aligned} \tag{A.0.11}$$

Thus, combining (A.0.9), (A.0.10) and (A.0.11), under $\mathcal{G}_T$, we have for any $(\beta_0, \boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)$,

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

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

$$\begin{aligned}
&\inf_{(\beta_0,\boldsymbol{\beta}) \in (\beta_0^*, \boldsymbol{\beta}^*) + \mathbb{C}(r)} F(\beta_0, \boldsymbol{\beta}) + \lambda_0\|\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \\
&> F(\beta_0^*, \boldsymbol{\beta}^*) + \lambda_0\|\boldsymbol{\beta}^*\|_2^2 + \lambda\|\boldsymbol{\beta}^*\|_1.
\end{aligned} \tag{A.0.12}$$

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 $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{G}_T$, $|\hat{\beta}_0 - \beta_0^*|^2 + \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2 \le r^2\frac{s\log p}{n}$. In fact, if $|\hat{\beta}_0 - \beta_0^*|^2 + \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2 > r^2\frac{s\log p}{n}$, let $t_0 := \frac{r\sqrt{s\log p/n}}{\sqrt{|\hat{\beta}_0 - \beta_0^*|^2 + \|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2}}$, 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.0.12), we further have

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

$$> F(\beta_0^*, \boldsymbol{\beta}^*) + \lambda_0 \|\boldsymbol{\beta}^*\|_2^2 + \lambda \|\boldsymbol{\beta}^*\|_1$$

under $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{G}_T$. The above inequality implies $F(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) + \lambda_0 \|\hat{\boldsymbol{\beta}}\|_2^2 + \lambda \|\hat{\boldsymbol{\beta}}\|_1 > F(\beta_0^*, \boldsymbol{\beta}^*) + \lambda_0 \|\boldsymbol{\beta}^*\|_2^2 + \lambda \|\boldsymbol{\beta}^*\|_1$, 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

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

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. □

APPENDIX B
PROOF OF LEMMA 1

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

$$L_h^{G\prime\prime}(v) = \frac{1}{\sqrt{2\pi}h} \exp\left\{-\frac{(1-v)^2}{2h^2}\right\} \le \frac{1}{\sqrt{2\pi}h}. \tag{B.0.1}$$

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

We then prove inequality (IV.2). The inequality is trivial when $v_1 < v_2 \le 1-h$ or $v_2 > v_1 \ge 1+h$. 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

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

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

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

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

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

where the second to the last inequality is due to

$$\sup_{v_2\in(1-h,1+h)}\frac{\left|\frac{(1-v_1+h)^2(2h-1+v_1)}{4h^3}\right|}{|1-v_1+h|}\leq\frac{9}{16h}<\frac{3}{4h}.\quad\square$$

APPENDIX 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., Bertsekas, 1999).

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

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\sum_{i=1}^n L_h(y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})+\sum_{k=1}^p w_k|\beta_k|,\tag{C.0.1}$$

where $w_k\geq 0$ 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 (C.0.1) as the following unconstrained optimization problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta})=g(\boldsymbol{\beta})+\sum_{k=1}^p h_k(\beta_k),\tag{C.0.2}$$

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

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

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

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

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, $r\geq 0$. For ease of notation, denote

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

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 (Parikh and Boyd, 2013) 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

$$(\nabla u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) + \zeta_k^{r+1})(\beta_k - \beta_k^{r+1}) \geq 0, \ \forall \beta_k. \tag{C.0.3}$$

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 Luo and Tseng (1992), Luo and Tseng (1993), Zhang et al. (2013) and Hong et al. (2013).

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 (C.0.3)

$$\begin{aligned} & f(\mathbf{b}_k^{r+1}) - f(\mathbf{b}_{k+1}^{r+1}) = f([\beta_k^r, \mathbf{b}_{-k}^{r+1}]) - f([\beta_k^{r+1}, \mathbf{b}_{-k}^{r+1}]) \\ & \geq u_k(\beta_k^r; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) - u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) + h_k(\beta_k^r) - h_k(\beta_k^{r+1}) \\ & = \nabla_k u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}])(\beta_k^r - \beta_k^{r+1}) + h_k(\beta_k^r) - h_k(\beta_k^{r+1}) \\ & + \frac{L_k}{2}(\beta_k^r - \beta_k^{r+1})^2 \\ & \geq (\nabla_k u_k(\beta_k^{r+1}; [\beta_k^r, \mathbf{b}_{-k}^{r+1}]) + \zeta_k^{r+1})(\beta_k^r - \beta_k^{r+1}) \\ & + \frac{L_k}{2}(\beta_k^r - \beta_k^{r+1})^2 \\ & \geq \frac{L_k}{2}(\beta_k^r - \beta_k^{r+1})^2. \end{aligned} \tag{C.0.4}$$

It follows that

$$f(\boldsymbol{\beta}^r) - f(\boldsymbol{\beta}^{r+1}) = \sum_{k=1}^p \left[f(\mathbf{b}_k^{r+1}) - f(\mathbf{b}_{k+1}^{r+1})\right] \geq \frac{\underline{L}}{2}\|\boldsymbol{\beta}^r - \boldsymbol{\beta}^{r+1}\|^2, \tag{C.0.5}$$

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 (C.0.2). 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

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

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

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

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

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

It follows that

$$
f(\boldsymbol{\beta}^{r+1}) - f(\bar{\boldsymbol{\beta}}^{r}) \le (5L\sqrt{p} + \bar{L})\big[\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^{r}\|^2 + \mathrm{d}^2_{\mathscr{X}^*}(\boldsymbol{\beta}^{r})\big], \tag{C.0.6}
$$

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

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 (C.0.4) and (C.0.5), the GCD algorithm is descending along its iterations. We can thus restrict the domain of $\boldsymbol{\beta}$ to the sublevel set $\mathcal{L}_0 = \{\boldsymbol{\beta} : f(\boldsymbol{\beta}) \le f(\mathbf{0})\}$. Let $\eta_i = \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}$ for $i = 1, \ldots, n$. It follows that the set $\mathcal{C}_0 = \{\boldsymbol{\eta} = (\eta_i, 1 \le i \le n)^{\mathrm{T}} \colon \boldsymbol{\beta} \in \mathcal{L}_0\}$ is convex compact. Therefore, for all $\boldsymbol{\beta} \in \mathcal{L}_0$, η_i is bounded by $\eta_{\max}$, where $\eta_{\max} = \max_{1\le i\le n}\sup_{\boldsymbol{\beta}\in\mathcal{L}_0}|\eta_i| < \infty$. 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 (B.0.1). We can see that $g(\boldsymbol{\beta}) = p(\mathbf{X}\boldsymbol{\beta})$. It follows from Zhang et al. (2013) that for any $\xi \ge \min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$, there exist $\kappa, \varepsilon > 0$ such that

$$
\mathrm{d}_{\mathscr{X}^*}(\boldsymbol{\beta}) \le \kappa\|\boldsymbol{\beta} - \mathbf{prox}_h(\boldsymbol{\beta} - \nabla g(\boldsymbol{\beta}))\|, \tag{C.0.7}
$$

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

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

$$\|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r - \nabla g(\boldsymbol{\beta}^r))\| \le \sigma \|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\|, \ \forall r \ge 1. \tag{C.0.8}$$

For any $r \ge 1$ and any $1 \le k \le p$, 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 Kadkhodaie et al. (2014) that

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

It follows that

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

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 (C.0.8). Note that the sufficient descent property (C.0.5) implies that $\|\boldsymbol{\beta}^{r+1} - \boldsymbol{\beta}^r\| \to 0$ as $r \to \infty$. It follows from (C.0.8) that $\|\boldsymbol{\beta}^r - \mathbf{prox}_h(\boldsymbol{\beta}^r - \nabla g(\boldsymbol{\beta}^r))\| \to 0$ as $r \to \infty$. Thus, by (C.0.7) we have $\mathrm{d}_{\mathscr{X}^*}(\boldsymbol{\beta}^r) \to 0$ as $r \to \infty$. Consequently, from (C.0.6) 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 (C.0.7) and the cost-to-go estimate (C.0.6), we obtain

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

which implies that

$$\Delta^{r+1} \le \frac{c_3}{1 + c_3} \Delta^r, \tag{C.0.9}$$

where $c_3 = (c_2 \kappa^2 \sigma^2 + c_2) c_1^{-1}$. We can see from (C.0.9) that $f(\boldsymbol{\beta}^r)$ approaches f^* with at least linear rate of convergence. From (C.0.5) again, this further implies that the sequence $\{\boldsymbol{\beta}^r\}$ converges at least linearly. □

APPENDIX D
ADDITIONAL NUMERIC RESULTS WITH GAUSSIAN KERNEL

Under the same settings introduced in our simulation section, we compared the performance of lasso DCSVM and elastic-net DCSVM, using Gaussian kernel. The result is shown in Table S.1. Again, we can see that the elastic-net DCSVM outperforms lasso DCSVM. We also conducted elastic-net DCSVM with Gaussian kernel on the same real datasets that we introduced in our real data section, and compared its performance with the performance of elastic-net SVM and elastic-net logistic regression. The result is displayed in Table S.2. Overall, DCSVM still achieves the best performance.

TABLE S.1. *Comparison of prediction error (in percentage) and variable selection of density-convoluted SVM with Gaussian 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									
Example 1									
500		6.92	(0.14)	5	0	**6.84**	(0.14)	5	0
5000		7.22	(0.19)	5	0	**7.11**	(0.13)	5	0
Example 2									
500	0.2	13.96	(0.21)	5	0	**13.52**	(0.19)	5	1
	0.7	23.18	(0.26)	3	0	**22.65**	(0.25)	4	0
	0.9	24.83	(0.24)	2	0	**24.75**	(0.23)	4	0
5000	0.2	14.46	(0.23)	5	0	**13.78**	(0.18)	5	0
	0.7	23.57	(0.26)	3	0	**22.66**	(0.21)	4	0
	0.9	25.25	(0.25)	2	0	**24.70**	(0.25)	3	0
Example 3									
500	0.2	10.58	(0.21)	5	0	**10.27**	(0.15)	5	1
	0.7	19.78	(0.21)	4	0	**19.48**	(0.18)	4	0
	0.9	23.97	(0.22)	2	0	**23.49**	(0.21)	4	0
5000	0.2	10.70	(0.20)	5	0	**10.51**	(0.20)	5	0
	0.7	20.13	(0.24)	3	0	**19.70**	(0.21)	4	0
	0.9	24.34	(0.30)	2	0	**23.85**	(0.23)	4	0

TABLE S.2. *Comparison of prediction error (in percentage) and run time (in second) of elastic-net density-convoluted SVM with Gaussian 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	**32.00**	(1.42)	454.36	37.09	(1.59)	8912.87	35.82	(1.65)	219.30
breast	42	22283	**24.86**	(1.79)	243.13	30.38	(2.05)	1946.98	30.76	(2.14)	227.88
colon	62	2000	**18.71**	(1.11)	91.70	18.90	(1.55)	722.48	23.87	(1.51)	27.33
leuk	72	7128	3.94	(0.51)	215.95	**3.89**	(0.51)	1863.23	4.33	(0.61)	115.00
LSVT	126	309	**15.74**	(0.62)	73.04	16.20	(0.68)	74.20	15.87	(0.68)	9.05
malaria	71	22283	**5.49**	(0.63)	818.98	7.60	(1.21)	12046.09	6.80	(0.98)	483.20
ovarian	253	15154	**0.67**	(0.13)	1491.25	4.87	(1.23)	14442.87	0.87	(0.14)	964.16
prostate	102	6033	9.69	(0.68)	199.85	**8.98**	(0.50)	2421.20	10.24	(0.61)	116.50

REFERENCES

BERTSEKAS, D. P. (1973). Stochastic optimization problems with nondifferentiable cost functionals. *Journal of Optimization Theory and Applications* **12**, 218–231.

BERTSEKAS, D. P. (1999). *Nonlinear Programming*. Athena Scientific.

BOSER, B. E., GUYON, I. M. and VAPNIK, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*.

BÜHLMANN, P. and VAN DE GEER, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media.

DONOHO, D. L. et al. (2000). High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture* **1**, 32.

FAN, J., LI, R., ZHANG, C.-H. and ZOU, H. (2020). *Statistical Foundations of Data Science*. Chapman and Hall/CRC.

FERNANDES, M., GUERRE, E. and HORTA, E. (2021). Smoothing quantile regressions. *Journal of Business and Economic Statistics* **39**, 338–357.

FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1–22.

FRIEDRICHS, K. O. (1944). The identity of weak and strong extensions of differential operators. *Transactions of the American Mathematical Society* **55**, 132–151.

GÖTZE, F., SAMBALE, H. and SINULIS, A. (2021). Concentration inequalities for polynomials in α-sub-exponential random variables. *Electronic Journal of Probability* **26**, 1–22.

HE, X., PAN, X., TAN, K. M. and ZHOU, W.-X. (2021). Smoothed quantile regression with large-scale inference. *Journal of Econometrics* .

HONG, M., WANG, X., RAZAVIYAYN, M. and LUO, Z.-Q. (2013). Iteration complexity analysis of block coordinate descent methods. *arXiv preprint arXiv:1310.6957* .

KADKHODAIE, M., SANJABI, M. and LUO, Z.-Q. (2014). On the linear convergence of the approximate proximal splitting method for non-smooth convex optimization. *Journal of the Operations Research Society of China* **2**, 123–141.

LEDOUX, M. and TALAGRAND, M. (1991). *Probability in Banach Spaces : Isoperimetry and Processes*. Ergebnisse der Mathematik und ihrer Grenzgebiete, A Series of Modern Surveys in Mathematics ; 23. Berlin, Heidelberg: Springer Berlin Heidelberg.

LUO, Z.-Q. and TSENG, P. (1992). On the linear convergence of descent methods for convex essentially smooth minimization. *SIAM Journal on Control and Optimization* **30**, 408–425.

LUO, Z.-Q. and TSENG, P. (1993). Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research* **46**, 157–178.

PARIKH, N. and BOYD, S. (2013). Proximal algorithms. *Foundations and Trends in optimization* **1**, 123–231.

PENG, B., WANG, L. and WU, Y. (2016). An error bound for l1-norm support vector machine coefficients in ultra-high dimension. *The Journal of Machine Learning Research* **17**, 8279–8304.

RUBINSTEIN, R. Y. (1983). Smoothed functionals in stochastic optimization. *Mathematics of Operations Research* **8**, 26–33.

TAN, K. M., WANG, L. and ZHOU, W.-X. (2021). High-dimensional quantile regression: convolution smoothing and concave regularization. *arXiv preprint arXiv:2109.05640* .

TIBSHIRANI, R., BIEN, J., FRIEDMAN, J., HASTIE, T., SIMON, N., TAYLOR, J. and TIBSHIRANI, R. J. (2010). Strong rules for discarding predictors in lasso-type problems. *Journal of the Royal Statistical Society, Series B* **74**, 245–266.

TSENG, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications* **109**, 475–494.

VAN DER VAART, A. W. and WELLNER, J. (1996). *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Science & Business Media.

VAPNIK, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.

VERSHYNIN, R. (2018). *High-Dimensional Probability: An Introduction With Applications in Data Science*, vol. 47. Cambridge University Press.

WANG, L., ZHU, J. and ZOU, H. (2006). The doubly regularized support vector machine. *Statistica Sinica* **16**, 589–616.

YANG, Y. and ZOU, H. (2013). An efficient algorithm for computing the hhsvm and its generalizations. *Journal of Computational and Graphical Statistics* **22**, 396–415.

ZHANG, H., JIANG, J. and LUO, Z.-Q. (2013). On the linear convergence of a proximal gradient method for a class of nonsmooth convex minimization problems. *Journal of the Operations Research Society of China* **1**, 163–186.

ZHU, J., ROSSET, S., TIBSHIRANI, R. and HASTIE, T. J. (2003). 1-norm support vector machines. In *Advances in Neural Information Processing Systems*. Citeseer.