mathematics

MDPI

Article

Communication-Efficient Distributed Learning for High-Dimensional Support Vector Machines

Xingcai Zhou * **and Hao Shen**

School of Statistics and Data Science, Nanjing Audit University, Nanjing 211085, China; haoshennau@163.com
* Correspondence: xczhou@nau.edu.cn

Abstract: Distributed learning has received increasing attention in recent years and is a special need for the era of big data. For a support vector machine (SVM), a powerful binary classification tool, we proposed a novel efficient distributed sparse learning algorithm, the communication-efficient surrogate likelihood support vector machine (CSLSVM), in high-dimensions with convex or nonconvex penalties, based on a communication-efficient surrogate likelihood (CSL) framework. We extended the CSL for distributed SVMs without the need to smooth the hinge loss or the gradient of the loss. For a CSLSVM with lasso penalty, we proved that its estimator could achieve a near-oracle property for l_1 penalized SVM estimators on whole datasets. For a CSLSVM with smoothly clipped absolute deviation penalty, we showed that its estimator enjoyed the oracle property, and that it used local linear approximation (LLA) to solve the optimization problem. Furthermore, we showed that the LLA was guaranteed to converge to the oracle estimator, even in our distributed framework and the ultrahigh-dimensional setting, if an appropriate initial estimator was available. The proposed approach is highly competitive with the centralized method within a few rounds of communications. Numerical experiments provided supportive evidence.

Keywords: distributed learning; support vector machine; communication efficiency; surrogate loss function; LLA algorithm; oracle property

MSC: 62H30; 62J07; 68W15

Citation: Zhou, X.; Shen, H. Communication-Efficient Distributed Learning for High-Dimensional Support Vector Machines. *Mathematics* **2022**, *10*, 1029. https://doi.org/10.3390/math10071029

Academic Editor: Chao Huang

Received: 21 February 2022
Accepted: 22 March 2022
Published: 23 March 2022

1. Introduction

The support vector machine (SVM), originally introduced by [1], has been a great success when applied to many classification problems. Owing to its high accuracy and flexibility it has provided solid mathematical foundations in machine learning. It is one of the most popular binary classification tools. The motivation of an SVM is to find a maximum-margin hyperplane by a regularized functional optimization problem. In statistical machine learning, the penalized functional is a sum of the hinge loss plus l_2-norm regularization. The statistical properties of an SVM have been studied in a lot of works. In this work, we focused on a distributed penalized linear SVM for datasets with large sample sizes and large dimensions.

With the development of modern technology, the size of data has become incredibly large, and, in some cases, cannot even be stored on a single machine. In real-world applications, many datasets are stored locally on individual servers and individual's devices, such as mobile phones and computers. It is difficult to collect these local data onto a single machine due to communication costs and privacy preservation. Thus, new methods and theories in distributed learning are called for. Distributed learning has attracted increasing attention in recent years, for example, see Refs. [2,3] for M-estimation, Refs. [4–6] for quantile regression, Refs. [7,8] for nonparametric regression, Refs. [3,9] for confidence intervals, and so on. These works focus on the simple setting of data parallelism under which the dataset is partitioned and distributed on m worker machines that can analyze data independently. Most methods suggest that in each round of communication,

each worker machine estimates the parameters of the model locally, and then communicates these local estimators to a master machine that averages these estimators to form a global estimator. Although this divide-and-conquer approach is communication-efficient, it has some restrictions: for achieving the minimax rate of convergence, the number of worker machines cannot be too large and samples in each worker machine should be large enough, these restrictions are highly restrictive. In addition, averaging can perform poorly if the estimator is nonlinear.

Not only can the size of data be exceedingly large, but also many successful models are heavily over-parameterized. These problems have been discussed widely. Zou [10] proposed an improved l_1 penalized SVM for simultaneous feature selection and classification, and showed that the hybrid SVM not only often improved the classification accuracy, but also enjoyed better feature-selection performance. Meinshausen and Buhlmann [11] studied the problem of variable selection in high-dimensional graphs, and explained that neighborhood selection with the lasso is a computationally attractive alternative to standard covariance selection for sparse high-dimensional graphs. Zhao and Yu [12] studied almost necessary and sufficient conditions for lasso to select the true model when $p < n$ or $p \gg n$, where p was the dimension of the model parameters and n was the sample size. Meinshausen and Yu [13] introduced sparse representations for high-dimensional data and proved that the estimator was still consistent even though the lasso could not recover the correct sparsity pattern. The adaptive lasso was proposed by Zou [14] and is a new version of the lasso that enjoys oracle properties. In the high-dimensional problems, the dimension p of the covariates was larger than the size of data, but there were only a few covariates that were relevant to the response. As a concrete example, in a microarray data set, which contains more than 10,000 genes, only several genes will make a difference to the result. Recently, the statistical inference for high-dimensional data has been investigated; readers can refer to [15,16] for details. In high-dimensional surroundings, a standard SVM can be easily affected by many redundant variables, so variable selection is important for high-dimensional SVMs. Fan and Fan [17] has shown that it was as poor as "tossing a coin" if all features were used in classification due to the accumulation of noise in high-dimensional analysis. Many works have been proposed to handle such a problem. Bradley and Mangasarian [18], Peng et al. [19], Zhu et al. [20] and Wegkamp and Yuan [21] studied the l_1 penalized SVM; Fan and Li [22] proposed an approach of variable selection and the estimation of model simultaneously by using a smoothly clipped absolute deviation penalty (SCAD) or minimax concave penalty (MCP), which are non-convex. Becker et al. [23], Park et al. [24] and Zhang et al. [25] considered the SCAD penalized SVM. Lian and Fan [26] gave the divide-and-conquer debiased estimator for an SVM, but such simple averaging might result in high computational costs for the high-dimensional problem, although it could become a debiased estimator by the lasso penalty. Jordan et al. [3] proposed the communication-efficient surrogate likelihood (CSL) framework for solving distributed statistical inference problems, which could work for high-dimensional penalized regression. As [27] stated, the CSL approach is different from distributed first-order optimization methods, which leverage both global first-order information and local higher-order information; yet, to the best of our knowledge, the distributed inference of variable selection for high-dimensional SVM has not been studied in the CSL framework.

In this paper, we propose communication-efficient distributed learning for support vector machines in high dimensions. Instead of using all the data to estimate the parameters, our method only needed to solve a regularized optimization problem on the first machine that was based on all gradients obtained from all worker machines. For the penalty function, we considered the convex l_1 and the non-convex SCAD penalty in a high-dimensional SVM, which could achieve variable selection and estimation simultaneously. In the distributed learning high-dimensional SVM, we did not need smooth assumptions to the loss or the gradient of the loss. We give some theoretical results in the paper.

The reminder of this paper is organized as follows. In Section 2, we give the problem formulation. Communication-efficient distributed estimation for an SVM is presented in

Section 3. In Section 4, we provide simulation studies and real data examples and demonstrate encouraging performances.

2. Problem Formulation

In this section, we set up our learning problem formally. We considered a strategy, which was empirical risk minimization, to obtain the optimal model. We considered a distributed learning framework with m worker machines, in which the 1st machine was regarded as the central machine. The 1st worker could aggregate information from another $m-1$ worker machines. In addition, every machine had n samples. So the size of total samples was $N = nm$. For a standard binary classification problem, we denoted $\mathcal{X}$ to be the input space and $\mathcal{Y} = \{-1, +1\}$ to be the output space. Random vectors $(\boldsymbol{X}, \boldsymbol{Y}) \in \mathcal{X} \times \mathcal{Y}$ were drawn from an unknown joint distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$. Let the parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^{\mathrm{T}}$ and the features $\mathbf{x}_i = (1, x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$. Suppose that training data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ are available from $\mathcal{D}$. Let $\ell(\boldsymbol{X}, \boldsymbol{\beta})$ be a loss function and

$$\boldsymbol{\beta}_0 = \arg\min_{\boldsymbol{\beta}} \mathrm{E}\ell(\boldsymbol{X}, \boldsymbol{\beta}),$$

where $\boldsymbol{\beta}_0 = (\beta_{00}, \beta_{01}, \ldots, \beta_{0p})^{\mathrm{T}}$ is the true parameter. Let the i.i.d (independent identically distributed) samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be stored on m machines and use $\mathcal{I}_k$ to denote the indices of samples on the kth machine with $|\mathcal{I}_k| = n = N/m$ for all $k \in [m]$ and $\mathcal{I}_j \cap \mathcal{I}_k = \emptyset$ for $j \neq k, j, k \in [m]$. The empirical risk of the kth machine is defined by

$$\widehat{L}_k(\boldsymbol{\beta}) = \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} \ell(\boldsymbol{\beta}; (\mathbf{x}_i, y_i)),$$

the empirical risk based on all N samples is

$$\widehat{L}(\boldsymbol{\beta}) = \frac{1}{m} \sum_{k=1}^{m} \widehat{L}_k(\boldsymbol{\beta}).$$

We used structural risk minimization strategy to learn $\boldsymbol{\beta}_0$, which is defined by

$$\widehat{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta}} \{\widehat{L}(\boldsymbol{\beta}) + g(\boldsymbol{\beta})\},$$

where $g(\boldsymbol{\beta})$ is the penalty term, such as l_1, l_2, SCAD and MCP.

In distributed statistical learning, Jordan et al. [3] proposed a distributed estimator with statistical guarantee and communication efficiency. Given an appropriate initial estimator $\widetilde{\boldsymbol{\beta}}$, we have

$$\widetilde{L}(\boldsymbol{\beta}) = \widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}} \Big(\nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}}) - \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}}) \Big).$$

By the above formula, we could introduce a communication-efficient distributed learning algorithm: the first worker machine broadcasts the initial $\widetilde{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}_1$ to the remaining $m-1$ machines and each machine computes the local gradient $\nabla \widehat{L}_k(\widetilde{\boldsymbol{\beta}})$. Then, these local gradients are sent back to the first machine where the first worker carries out an SVM by aggregating these local gradients. The communication-efficient estimator is given by

$$\check{\boldsymbol{\beta}} = \mathrm{argmin}_{\boldsymbol{\beta}} \{\widetilde{L}(\boldsymbol{\beta}) + g(\boldsymbol{\beta})\},$$

with approximate loss $\widetilde{L}(\boldsymbol{\beta})$. The architecture of the distributed learning reduced the total communication costs $O((m-1)np)$ to $O((m-1)p)$, where p is the dimension of $\boldsymbol{\beta}$.

3. Distributed Learning for an SVM

A standard non-separable SVM has the following form,

$$\widehat{L}(\boldsymbol{\beta}) + g(\boldsymbol{\beta}) = \frac{1}{N}\sum_{i=1}^{N}\Big(1 - y_i \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}\Big)_+ + \frac{\lambda}{2}\|\boldsymbol{\beta}^*\|_2^2,$$

where $(x)_+ = max(x, 0)$ is the hinge loss function that is piecewise linear and is differentiable except at point 0; $\boldsymbol{\beta}^* = (\beta_1, \ldots, \beta_p)^{\mathrm{T}}$ is the unknown p-dimensional parameter; λ is the regularization parameter, which determines the importance of penalty term. The true parameter $\boldsymbol{\beta}_0$ is the minimum of the following population loss function

$$L(\boldsymbol{\beta}) = \mathrm{E}\Big[1 - Y\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}\Big]_+.$$

We define

$$S(\boldsymbol{\beta}) = -\mathrm{E}\Big\{I\Big(1 - Y\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta} \geqslant 0\Big)Y\boldsymbol{X}\Big\},$$
$$H(\boldsymbol{\beta}) = \mathrm{E}\Big\{\delta\Big(1 - Y\boldsymbol{X}^{\mathrm{T}}\boldsymbol{\beta}\Big)\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\Big\},$$

where $I\{\cdot\}$ is the indicator function and $\delta\{\cdot\}$ is the Dirac delta function. The $S(\boldsymbol{\beta})$ and $H(\boldsymbol{\beta})$ could be viewed as the gradient vector and Hessian matrix of $L(\boldsymbol{\beta})$.

The empirical loss function of the kth worker machine is $\widehat{L}_k(\boldsymbol{\beta}) = \frac{1}{|\mathcal{I}_k|}\sum_{i\in\mathcal{I}_k}\big(1 - y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}\big)_+$. Given $\widetilde{\boldsymbol{\beta}}$ is an initial estimator of $\boldsymbol{\beta}_0$, we use $\widetilde{L}(\boldsymbol{\beta}) := \widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}}\Big(\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}})\Big)$ to replace $\widehat{L}(\boldsymbol{\beta}) = \frac{1}{N}\sum_{i=1}^{N}\big(1 - y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}\big)_+$, and then obtain the distributed estimator

$$\check{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}}\widetilde{L}(\boldsymbol{\beta}) + g(\boldsymbol{\beta}). \tag{1}$$

In this paper, we considered the l_1 penalty and SCAD penalty, respectively. The l_1 penalty is

$$g(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|_1,$$

and the SCAD penalty is

$$g(\boldsymbol{\beta}) = \sum_{j=1}^{p} p_\lambda(|\beta_j|),$$

where

$$\begin{aligned} p_\lambda(t) &= \lambda|t|I(0 \leqslant |t| < \lambda) + \frac{a\lambda|t| - \big(t^2 + \lambda^2\big)/2}{a-1}I(\lambda \leqslant |t| \leqslant a\lambda) \\ &\quad + \frac{(a+1)\lambda^2}{2}I(|t| > a\lambda) \end{aligned}$$

for some $a > 2$. Note that the SCAD penalty has the following properties:

Property 1: $p_\lambda(t)$ is symmetric and for $t \in [0, \infty)$ is non-decreasing and concave, with $p_\lambda(0) = 0$.

Property 2: The derivative of $p_\lambda(t)$ is continuous on $(0, \infty)$: for some $a > 1$, $\lim_{t\to 0+} p'_\lambda(t) = \lambda$, $p'_\lambda(t) \geqslant \lambda - t/a$ for $0 < t < a$, and $p'_\lambda(t) = 0$ for $t \geqslant a\lambda$.

In practice, we adopted the following CSL distributed learning for an SVM, which is summarized in Algorithm 1. However, our theories were based on the distributed estimator $\check{\boldsymbol{\beta}}$ in (1).

Algorithm 1: CSL distributed support vector machine (CSLSVM).

Input: Compute the initial estimator $\boldsymbol{\beta}^{(0)} = \widetilde{\boldsymbol{\beta}}$ on the first machine using l_1 penalized SVM.

for $t = 0, 1, \cdots, T-1$ **do**

The 1st machine: broadcast the current iterate $\boldsymbol{\beta}^{(t)}$ to other worker machines.

for all $k \in \{1, 2, \cdots, m\}$ **parallel do**

Worker machine k:
evaluate local gradient $\nabla \widehat{L}_k(\boldsymbol{\beta}^{(t)})$
send $\nabla \widehat{L}_k(\boldsymbol{\beta}^{(t)})$ to the 1st machine.

end

The 1st machine: aggregate gradients

$$\nabla \widehat{L}(\boldsymbol{\beta}^{(t)}) = \frac{1}{m} \sum_{k=1}^{m} \widehat{L}_k(\nabla \boldsymbol{\beta}^{(t)}),$$

for SVM with l_1 penalty (Call it L1SVM algorithm), compute

$$\boldsymbol{\beta}^{(t+1)} = \arg\min_{\boldsymbol{\beta}} \widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}} \Big(\nabla \widehat{L}_1(\boldsymbol{\beta}^{(t)}) - \nabla \widehat{L}(\boldsymbol{\beta}^{(t)}) \Big) + \lambda \|\boldsymbol{\beta}\|_1;$$

for SVM with SCAD penalty (Call it SCADSVM algorithm), computes

$$\boldsymbol{\beta}^{(t+1)} = \arg\min_{\boldsymbol{\beta}} \widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}} \Big(\nabla \widehat{L}_1(\boldsymbol{\beta}^{(t)}) - \nabla \widehat{L}(\boldsymbol{\beta}^{(t)}) \Big) + SCAD.$$

end

Output: $\boldsymbol{\beta}^{(T)}$.

3.1. A Communication-Efficient Distributed SVM with Lasso Penalty

In this section, we establish the theoretical properties of the proposed estimator. Despite the generality and elegance of [3]'s method, the approach uses only at least second derivatives for smooth loss functions, such as cross-entropy loss, which can not directly apply to the hinge loss of an SVM. Recall that surrogate loss $\widetilde{L}(\boldsymbol{\beta}) := \widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}} \Big(\nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}}) - \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}}) \Big)$. We only used first-order information, so if the gradient existed almost everywhere, it was usable for the aforementioned method. For an SVM, the hinge loss was differentiable except at point 0. We could use a subgradient function

$$\nabla \widehat{L}(\boldsymbol{\beta}) = -\frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} I\Big(1 - y_i \mathbf{x}_i^T \boldsymbol{\beta} \geq 0\Big) y_i \mathbf{x}_i$$

in place of the gradient and, thus, the surrogate loss was directly usable.

Our main results were established under the following assumptions.

(**A1**) The conditional densities of $\boldsymbol{X}^T \boldsymbol{\beta_0}$ given $\boldsymbol{Y} = 1$ and $\boldsymbol{Y} = -1$ are denoted as f and g, respectively. It is assumed that f is uniformly bounded away from 0 and ∞ in a neighborhood of 1, and g is uniformly bounded away from 0 and ∞ in a neighborhood of -1.

(**A2**) $\boldsymbol{\beta}_0$ is a sparse and nonzero vector, and S denotes the support of $\boldsymbol{\beta}_0$.

(**A3**) $\mathbf{x}$ is a sub-Gaussian random vector. That is, for any $\boldsymbol{\eta} \in \mathbb{R}^p$,

$$E\Big[\exp\Big\{\boldsymbol{\eta}^{\mathrm{T}} \mathbf{x}\Big\}\Big] \leq \exp\Big\{C\|\boldsymbol{\eta}\|^2\Big\};$$

it is assumed that each component x_{ij} of feather x_i is a random variable with a mean of zero and variance 1.

Denote $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)^T$ to be the feature design matrix and define restricted eigenvalues as follows,

$$\lambda_{\max} = \max_{\delta \in \mathbb{R}^{p+1}: \|\delta\|_0 \le Cq} \frac{\delta^T \mathcal{X}^T \mathcal{X} \delta}{n\|\delta\|_2^2},$$

and

$$\lambda_{\min}(H(\boldsymbol{\beta}^*); q) = \min_{\delta \in \Delta} \frac{\delta^T H(\boldsymbol{\beta}^*) \delta}{\|\delta\|_2^2},$$

where Δ is a restricted cone in $\mathbb{R}^{p+1}$,

$$\Delta = \left\{ \gamma \in \mathbb{R}^{p+1} : \|\gamma_{S_+}\|_1 \le 3 \left\| \gamma_{S_+^c} \right\|_1 \right\},$$

$S_+ = S \cup \{0\}$, $S \subset \{1, 2, \ldots, p\}$ and $|S| \le q$.

(**A4**) $\lambda_{\max}$ and $\lambda_{\min}$ are bounded away from zero.

(**A5**) The initial estimator $\widetilde{\boldsymbol{\beta}}$ is sparse and $\left\| \widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 \right\|_1 \le Cq\sqrt{\frac{\log p}{n}}$.

Remark 1. *Under Assumption (**A1**), the Hessian matrix $H(\boldsymbol{\beta})$ was well-defined and continuous in $\boldsymbol{\beta}$. (**A1**) ensured that we could obtain sufficient information around the non-differentiable point of the hinge loss; see more details in [24,28]. (**A2**) is a common assumption in high-dimensional problems and we knew $|S| \le s$ for some $s \le min\{p, n\}$. In this paper, we used q as the number of nonzero entries in $\boldsymbol{\beta}_0$. Using the sub-Gaussianity assumption (**A3**), we easily obtained $\max_i \|\mathbf{x}_i\|_\infty \le C\sqrt{\log p}$ with probability $1 - p^{-C}$. Assumption (**A4**) is similar to Lemma 2 in [19]; we used these to control the bounds of the empirical loss function of the SVM and it's expectation. (**A5**) is an assumption of the initial estimator for the iterative algorithm; Ref. [19] proved that the L_1-norm SVM coefficients satisfied such assumptions.*

Our results were as follows.

Theorem 1. *Assume that (**A1**)–(**A5**) above and that $\lambda \ge 2\left\|\nabla \widetilde{L}(\boldsymbol{\beta}_0)\right\|_\infty$, we have with probability at least $1 - n^{-C}$,*

$$\left\| \check{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 \right\| \le C\left[\lambda\sqrt{q} + \frac{q^{3/2}(\log p)^{5/2}}{n} + \frac{(q \log p)^{1/2}}{n^{1/2}} + \frac{q^{3/2} \log p}{n^{3/4}} \right].$$

Remark 2. *Although N did not appear in the formula, in fact, the condition $\lambda \ge 2\left\|\nabla \widetilde{L}(\boldsymbol{\beta}_0)\right\|_\infty$ implied that the convergence rate was dependent on N. If m was not too big, that is, n was not too small, we chose $\lambda \simeq \sqrt{\log p / N}$. If $q \gtrsim n^{1/4}/(\log p)^{1/2}$ and $q \log p / n^{3/4} \lesssim \sqrt{\log p / N}$, the convergence rate would be dominated by the first term $\lambda\sqrt{q}$. That is,*

$$\left\| \check{\boldsymbol{\beta}} - \boldsymbol{\beta}_0 \right\| \le C\sqrt{\frac{q \log p}{N}}.$$

This was a near-oracle property for the l_1 penalized SVM estimator based on the entirety of the datasets [27].

Proof of Theorem 1. We prove the result by the following three steps in line with the proof of [6].

Step 1. Let $\delta = \check{\boldsymbol{\beta}} - \boldsymbol{\beta}_0$. Since $\widetilde{L}(\boldsymbol{\beta})$ is convex in $\boldsymbol{\beta}$, we have

$$\widetilde{L}(\boldsymbol{\beta}) - \widetilde{L}(\boldsymbol{\beta}_0) \ge \nabla \widetilde{L}(\boldsymbol{\beta}_0)(\boldsymbol{\beta} - \boldsymbol{\beta}_0)$$

for all $\boldsymbol{\beta}$. In terms of $\widetilde{L}(\check{\boldsymbol{\beta}}) + \lambda\|\check{\boldsymbol{\beta}}\|_1 \leq \widetilde{L}(\boldsymbol{\beta}_0) + \lambda\|\boldsymbol{\beta}_0\|_1$ and Hölder's inequality, we obtain

$$-\left\|\nabla\widetilde{L}(\boldsymbol{\beta}_0)\right\|_\infty\|\boldsymbol{\delta}\|_1 \leq \widetilde{L}(\check{\boldsymbol{\beta}}) - \widetilde{L}(\boldsymbol{\beta}_0) \leq \lambda\|\boldsymbol{\beta}_0\|_1 - \lambda\|\boldsymbol{\beta}_0 + \boldsymbol{\delta}\|_1.$$

Using $\lambda \geq 2\left\|\nabla\widetilde{L}(\boldsymbol{\beta}_0)\right\|_\infty$, we obtain

$$-\frac{\lambda}{2}\|\boldsymbol{\delta}\|_1 \leq \lambda\|\boldsymbol{\beta}_0\|_1 - \lambda\|\boldsymbol{\beta}_0 + \boldsymbol{\delta}\|_1.$$

Writing $\|\boldsymbol{\delta}\|_1 = \|\boldsymbol{\delta}_S\|_1 + \|\boldsymbol{\delta}_{S^c}\|_1$, $\|\boldsymbol{\beta}_0\|_1 = \|\boldsymbol{\beta}_{0S}\|_1$ and $\|\boldsymbol{\beta}_0 + \boldsymbol{\delta}\|_1 = \|\boldsymbol{\beta}_{0S} + \boldsymbol{\delta}_S\|_1 + \|\boldsymbol{\delta}_{S^c}\|_1$, we obtain

$$-\frac{\lambda}{2}\|\boldsymbol{\delta}_S\|_1 - \frac{\lambda}{2}\|\boldsymbol{\delta}_{S^c}\|_1 \leq \lambda\|\boldsymbol{\delta}_S\|_1 - \lambda\|\boldsymbol{\delta}_{S^c}\|_1.$$

After rearranging, we have

$$\|\boldsymbol{\delta}_{S^c}\|_1 \leq 3\|\boldsymbol{\delta}_S\|_1.$$

Step 2. We observe that

$$\begin{aligned}
&\widetilde{L}(\boldsymbol{\beta}_0 + \boldsymbol{\delta}) - \widetilde{L}(\boldsymbol{\beta}_0) - \boldsymbol{\delta}^{\mathrm{T}}\nabla\widetilde{L}(\boldsymbol{\beta}_0) = \widehat{L}_1(\boldsymbol{\beta}_0 + \boldsymbol{\delta}) - \widehat{L}_1(\boldsymbol{\beta}_0) - \boldsymbol{\delta}^{\mathrm{T}}\nabla\widehat{L}_1(\boldsymbol{\beta}_0)\\
= \;& \frac{1}{n}\sum_{i=1}^{n}\Big(1 - y_i\boldsymbol{x}_i^T(\boldsymbol{\beta}_0 + \boldsymbol{\delta})\Big)I\Big\{y_i\boldsymbol{x}_i^T(\beta_0 + \boldsymbol{\delta}) \leq 1\Big\}\\
& -\frac{1}{n}\sum_{i=1}^{n}\Big(1 - y_i\boldsymbol{x}_i^T(\boldsymbol{\beta}_0)\Big)I\Big\{y_i\boldsymbol{x}_i^T(\beta_0) \leq 1\Big\} - \boldsymbol{\delta}^{\mathrm{T}}\nabla\widetilde{L}(\boldsymbol{\beta}_0)\\
= \;& \frac{1}{n}\sum_{i=1}^{n}\Big(I\Big\{y_i\boldsymbol{x}_i^T(\beta_0 + \boldsymbol{\delta}) \leq 1\Big\} - I\Big\{y_i\boldsymbol{x}_i^T(\beta_0) \leq 1\Big\}\Big)\\
& +\frac{1}{n}\sum_{i=1}^{n}y_i\boldsymbol{x}_i^T\Big[(\boldsymbol{\beta}_0 + \boldsymbol{\delta})I\Big\{y_i\boldsymbol{x}_i^T(\beta_0 + \boldsymbol{\delta}) \leq 1\Big\} - \boldsymbol{\beta}_0 I\Big\{y_i\boldsymbol{x}_i^T\beta_0 \leq 1\Big\}\Big]\\
& -\boldsymbol{\delta}^{\mathrm{T}}\nabla\widehat{L}_1(\boldsymbol{\beta}_0)\\
= \;& Q_{1n} + Q_{2n} + Q_{3n}. \qquad (2)
\end{aligned}$$

With arguments basically the same as the proof of Proposition 3, we could prove that for any δ

$$\begin{aligned}
\sup_{\|\boldsymbol{\delta}\|\leq t}|Q_{1n} - E(Q_{1n})| &= \sup_{\|\boldsymbol{\delta}\|\leq t} O\left(\|x\|_\infty^{1/2}q^{1/4}\|\boldsymbol{\delta}\|^{1/2}\sqrt{\frac{q\log p}{n}} + \|\boldsymbol{x}\|_\infty\frac{q\log p}{n}\right)\\
&= O_p\left(\frac{(q\log p)^{3/4}t^{1/2}}{\sqrt{n}} + \frac{q(\log p)^{3/2}}{n}\right), \qquad (3)
\end{aligned}$$

$$\begin{aligned}
\sup_{\|\boldsymbol{\delta}\|\leq t}|Q_{2n} - E(Q_{2n})| &= \sup_{\|\boldsymbol{\delta}\|\leq t} O\left(\|x\|_\infty^{3/2}q^{1/4}\|\boldsymbol{\beta}_0\|_1\|\boldsymbol{\delta}\|^{1/2}\sqrt{\frac{q\log p}{n}} + \|\boldsymbol{x}\|_\infty\frac{q\log p}{n}\right)\\
&= O_p\left(\frac{q^{3/4}(\log p)^{5/4}t^{1/2}}{\sqrt{n}} + \frac{q(\log p)^{3/2}}{n}\right). \qquad (4)
\end{aligned}$$

By following Proposition 1, we have

$$\sup_{\|\boldsymbol{\delta}\|\leq t}|Q_{3n}| = O\Big(\left\|\nabla\widehat{L}_1(\boldsymbol{\beta}_0)\right\|_\infty\|\boldsymbol{\delta}\|_1\Big) = O_p\left(\left[\sqrt{\frac{\log p}{n}} + \frac{q\log p}{n^{3/4}} + \frac{q(\log p)^{3/2}}{n}\right]\sqrt{q}t\right). \qquad (5)$$

Based on (2)–(5), we have with probability $1 - n^{-C}$

$$\sup_{\|\delta\| \le t} | \widehat{L}_1(\boldsymbol{\beta}_0 + \boldsymbol{\delta}) - \widehat{L}_1(\boldsymbol{\beta}_0) - \boldsymbol{\delta}^{\mathrm{T}} \nabla \widehat{L}_1(\boldsymbol{\beta}_0) - E\widehat{L}_1(\boldsymbol{\beta}_0 + \boldsymbol{\delta}) + E\widehat{L}_1(\boldsymbol{\beta}_0) | \le C \left\{ \frac{q^{3/4} (\log p)^{5/4} t^{1/2}}{\sqrt{n}} + \left[\sqrt{\frac{\log p}{n}} + \frac{q \log p}{n^{3/4}} + \frac{q (\log p)^{3/2}}{n} \right] \sqrt{q} t \right\}.$$

Step 3. Assume that $\|\check{\boldsymbol{\beta}} - \boldsymbol{\beta}\| > t$ for some $t > 0$. By step 1, this implies

$$\inf_{\substack{\|\delta\| \ge t \\ \|\delta_{S^c}\|_1 \le 3 \|\delta_S\|_1}} \widetilde{L}(\boldsymbol{\beta}_0 + \boldsymbol{\delta}) - \widetilde{L}(\boldsymbol{\beta}_0) + \lambda \|\boldsymbol{\beta}_0 + \boldsymbol{\delta}\|_1 - \lambda \|\boldsymbol{\beta}_0\|_1 \le 0.$$

By the triangle inequality, we have $\|\boldsymbol{\beta}_0 + \boldsymbol{\delta}\|_1 - \|\boldsymbol{\beta}_0\|_1 \ge -\|\boldsymbol{\delta}_S\|_1 \ge -\sqrt{q}\|\boldsymbol{\delta}_S\| \ge -\sqrt{q} t$. Using the result from Step 2 and the lower bound for $E\left[\widehat{L}_1(\boldsymbol{\beta}_0 + \boldsymbol{\delta})\right] - E\left[\widehat{L}_1(\boldsymbol{\beta}_0)\right]$, similar to Lemma 4 of [29], we have

$$\begin{aligned}
&\widetilde{L}(\boldsymbol{\beta}_0 + \boldsymbol{\delta}) - \widetilde{L}(\boldsymbol{\beta}_0) \\
&\quad \ge E\left[\widehat{L}_1(\boldsymbol{\beta}_0 + \boldsymbol{\delta})\right] - E\left[\widehat{L}_1(\boldsymbol{\beta}_0)\right] - \|\boldsymbol{\delta}\|_1 \left\| \nabla \widetilde{L}(\boldsymbol{\beta}_0) \right\|_\infty \\
&\qquad - C \left\{ \frac{q^{3/4} (\log p)^{5/4} t^{1/2}}{\sqrt{n}} + \left[\sqrt{\frac{\log p}{n}} + \frac{q \log p}{n^{3/4}} + \frac{q (\log p)^{3/2}}{n} \right] \sqrt{q} t \right\} \\
&\quad \ge -C \left\{ \frac{q^{3/4} (\log p)^{5/4} t^{1/2}}{\sqrt{n}} + \left[\sqrt{\frac{\log p}{n}} + \frac{q \log p}{n^{3/4}} + \frac{q (\log p)^{3/2}}{n} \right] \sqrt{q} t \right\} \\
&\qquad + C(t^2 \wedge t) - C\lambda \sqrt{q} t.
\end{aligned}$$

Thus, we have

$$C(t^2 \wedge t) - C\lambda \sqrt{q} t - C \left\{ \frac{q^{3/4} (\log p)^{5/4} t^{1/2}}{\sqrt{n}} + \left[\sqrt{\frac{\log p}{n}} + \frac{q \log p}{n^{3/4}} + \frac{q (\log p)^{3/2}}{n} \right] \sqrt{q} t \right\} \le 0.$$

Some algebra shows that

$$t \le C \left[\lambda \sqrt{q} + \frac{q^{3/2} (\log p)^{5/2}}{n} + \sqrt{\frac{q \log p}{n}} + \frac{q^{3/2} \log p}{n^{3/4}} \right].$$

□

Proposition 1. *Under the same assumptions as Theorem 1 with probability at least* $1 - p^{-C}$

$$\left\| \nabla \widetilde{L}(\boldsymbol{\beta}_0) \right\|_\infty \le C \left(\sqrt{\frac{\log p}{N}} + \frac{q \log p}{n^{3/4}} + \frac{q (\log p)^{3/2}}{n} \right).$$

Proof of Proposition 1. By the definition of $\widetilde{L}$, we have $\nabla \widetilde{L}(\boldsymbol{\beta}_0) = \nabla \widehat{L}_1(\boldsymbol{\beta}_0) - \nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}}) + \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}})$ and thus

$$\left\| \nabla \widetilde{L}(\boldsymbol{\beta}_0) \right\|_\infty \le \left\| \nabla \widehat{L}_1(\boldsymbol{\beta}_0) - \nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}}) - \nabla \widehat{L}(\boldsymbol{\beta}_0) + \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}}) \right\|_\infty + \left\| \nabla \widehat{L}(\boldsymbol{\beta}_0) \right\|_\infty.$$

The last term above is the same as that dealt with in Lemma 1 in [19], which shows that with probability at least $1 - p^{-C}$,

$$\left\| \nabla \widehat{L}(\boldsymbol{\beta}_0) \right\|_\infty \le C \sqrt{\log p / N}.$$

In Proposition 2, we show that with probability $1 - p^{-C}$

$$\left\| \frac{1}{n} \sum_i y_i \mathbf{x}_i \left(I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \widetilde{\boldsymbol{\beta}} \le 1 \right\} - I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) - E y \mathbf{x} \left(I\left\{ y \mathbf{x}^{\mathrm{T}} \widetilde{\boldsymbol{\beta}} \le 1 \right\} - I\left\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) \right\|_\infty$$
$$\le C \left(\frac{q \log p}{n^{3/4}} + \frac{q (\log p)^{3/2}}{n} \right).$$

Similarly,

$$\left\| \frac{1}{N} \sum_i y_i \mathbf{x}_i \left(I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \widetilde{\boldsymbol{\beta}} \le 1 \right\} - I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) - E y \mathbf{x} \left(I\left\{ y \mathbf{x}^{\mathrm{T}} \widetilde{\boldsymbol{\beta}} \le 1 \right\} - I\left\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) \right\|_\infty$$
$$\le C \left(\frac{q \log p}{N^{3/4}} + \frac{q (\log p)^{3/2}}{N} \right) \le C \left(\frac{q \log p}{n^{3/4}} + \frac{q (\log p)^{3/2}}{n} \right).$$

Thus, we have

$$\left\| \nabla \widehat{L}_1(\boldsymbol{\beta}_0) - \nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}}) - \nabla \widehat{L}(\boldsymbol{\beta}_0) + \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}}) \right\|_\infty \le C \left(\frac{q \log p}{n^{3/4}} + \frac{q (\log p)^{3/2}}{n} \right).$$

Then we can obtain

$$\left\| \nabla \widetilde{L}(\boldsymbol{\beta}_0) \right\|_\infty \le C \left(\sqrt{\frac{\log p}{N}} + \frac{q \log p}{n^{3/4}} + \frac{q (\log p)^{3/2}}{n} \right).$$

□

Proposition 2. *Under the same assumptions as Theorem 1, with probability at least* $1 - p^{-C}$, *we have*

$$\left\| \frac{1}{N} \sum_i y_i \mathbf{x}_i \left(I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \widetilde{\boldsymbol{\beta}} \le 1 \right\} - I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) - E y \mathbf{x} \left(I\left\{ y \mathbf{x}^{\mathrm{T}} \widetilde{\boldsymbol{\beta}} \le 1 \right\} - I\left\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) \right\|_\infty$$
$$\le C \left(\frac{q \log p}{N^{3/4}} + \frac{q (\log p)^{3/2}}{N} \right).$$

Proof of Proposition 2. We take $\Omega = \{ \boldsymbol{\beta} \in \mathbb{R}^p : \|\boldsymbol{\beta}\|_0 \le q, \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1 \le C q \sqrt{\log p / N} \}$. Define the class of functions

$$\mathcal{G}_j = \left\{ y x_j \left(I\left\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} \le 1 \right\} - I\left\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) : \boldsymbol{\beta} \in \Omega \right\}$$

with squared integrable envelope function $F(\mathbf{x}, y) = |x_j|$. We decompose Ω as $\Omega = \cup_{T \subset \{1,\ldots,p\}, |T| \le K} \Omega(T)$ with $\Omega(T) = \{ \boldsymbol{\beta} : \text{support of } \boldsymbol{\beta} \subset T \} \cap \Omega$. We also define $\mathcal{G}_j(T) = \{ y x_j (I\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} \le 1 \} - I\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \}) : \boldsymbol{\beta} \in \Omega(T) \}$. By Lemma 2.6.15, Lemma 2.6.18 (vi) and (viii) in [30], for each fixed $T \subset \{1, \ldots, p\}$ with $|T| \le Cq$, $\mathcal{G}_j(T)$ is a VC-subgraph with index bounded by Cq and by Theorem 2.6.7 of [30], we have

$$N\left(\epsilon, \mathcal{G}_j(T), L_2(P_n)\right) \le \left(\frac{C \|F\|_{L_2(P_n)}}{\epsilon} \right)^{Cq} \le \left(\frac{C}{\epsilon} \right)^{Cq}.$$

Since there are at most $\begin{pmatrix} p \\ Cq \end{pmatrix} \leq (ep/Cq)^{Cq}$ different such T, we have

$$N(\epsilon, \mathcal{G}_j, L_2(P_n)) \leq \left(\frac{C}{\epsilon}\right)^{Cq} \left(\frac{ep}{Cq}\right)^{Cq} \leq \left(\frac{Cp}{\epsilon}\right)^{Cq}$$

and thus

$$N\left(\epsilon, \cup_{j=1}^{p} \mathcal{G}_j, L_2(P_n)\right) \leq p\left(\frac{Cp}{\epsilon}\right)^{Cq}.$$

Let $\sigma^2 = \sup_{f \in \cup_j \mathcal{G}_j} Pf^2$. Then by Theorem 3.12 of [31], we have

$$E\|R_n\|_{\cup_j \mathcal{G}_j} \leq C\left(\sigma\sqrt{\frac{q\log p}{N}} + \frac{q\sqrt{\log p}\log p}{N}\right),$$

where $\|R_n\|_{\cup_j \mathcal{G}_j} = \sup_{f \in \cup_j \mathcal{G}_j} N^{-1} \sum_{i=1}^{N} \varepsilon_i f(\mathbf{x}_i, y_i)$ with ε_i being i.i.d. Rademacher random variables. Using the symmetrization inequality, which states that $E\|P_n - P\|_{\cup_j \mathcal{G}_j} \leq 2E\|R_n\|_{\cup_j \mathcal{G}_j}$, where $\|P_n - P\|_{\cup_j \mathcal{G}_j} = \sup_{f \in \cup_j \mathcal{G}_j} N^{-1} \sum_i f(\mathbf{x}_i, y_i) - Ef(\mathbf{x}, y)$, Talagrand's inequality (page 24 of [31]) gives

$$P\left(\|P_n - P\|_{\cup_j \mathcal{G}_j} \geq C\left(\sigma\sqrt{\frac{q\log p}{N}} + \frac{q\sqrt{\log p}\log p}{N} + \sqrt{\frac{\sigma^2 t}{N}} + \frac{\sqrt{\log p}t}{N}\right)\right) \leq e^{-t},$$

that is, with probability at least $1 - p^{-C}$,

$$\left\|\frac{1}{N}\sum_i y_i\mathbf{x}_i\left(I\left\{y_i\mathbf{x}_i^{\mathrm{T}}\widetilde{\boldsymbol{\beta}} \leq 1\right\} - I\left\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \leq 1\right\}\right) - Ey\mathbf{x}\left(I\left\{y\mathbf{x}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}} \leq 1\right\} - I\left\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \leq 1\right\}\right)\right\|_{\infty}$$
$$\leq C\left(\sigma\sqrt{\frac{q\log p}{N}} + \sqrt{\log p}\frac{q\log p}{N}\right).$$

Finally, we need to decide the size of σ^2. For $\boldsymbol{\beta} \in \Omega$, we have that

$$\begin{aligned}
&E\left[\left(I\left\{\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} \leq 1\right\} - I\left\{\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \leq 1\right\}\right)^2 \mid y = 1\right] \\
&\leq P\left(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} \leq 1, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \geq 1 \mid y = 1\right) + P\left(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} \geq 1, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \leq 1 \mid y = 1\right) \\
&\leq C|\mathbf{x}^{\mathrm{T}}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)| \\
&\leq C\sqrt{\log p}q\sqrt{\log p/N}.
\end{aligned}$$

Thus, with probability at least $1 - p^{-C}$,

$$\left\|\frac{1}{N}\sum_i y_i\mathbf{x}_i\left(I\left\{y_i\mathbf{x}_i^{\mathrm{T}}\widetilde{\boldsymbol{\beta}} \leq 1\right\} - I\left\{y_i\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}_0 \leq 1\right\}\right) - Ey\mathbf{x}\left(I\left\{y\mathbf{x}^{\mathrm{T}}\widetilde{\boldsymbol{\beta}} \leq 1\right\} - I\left\{y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}_0 \leq 1\right\}\right)\right\|_{\infty}$$
$$\leq C\left(\frac{q\log p}{N^{3/4}} + \frac{q(\log p)^{3/2}}{N}\right).$$

□

Proposition 3. *Under the same assumptions as Theorem 1, with probability at least* $1 - p^{-C}$, *we have*

$$\left\| \frac{1}{N}\sum_i y_i \mathbf{x}_i \left(\boldsymbol{\beta} I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta} \le 1 \right\} - \boldsymbol{\beta}_0 I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) - E y \mathbf{x} \left(\boldsymbol{\beta} I\left\{ y \mathbf{x}^{\mathrm{T}} \widetilde{\boldsymbol{\beta}} \le 1 \right\} - \boldsymbol{\beta}_0 I\left\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) \right\|_\infty$$
$$\le C \left(\frac{q^{5/4} (\log p)^{3/2}}{n^{3/4}} + \frac{q^2 (\log p)^{3/2}}{n} \right).$$

Proof of Proposition 3. The proof is similar to the proof of Proposition 2. We take $\Omega = \{ \boldsymbol{\beta} \in \mathbb{R}^p : \|\boldsymbol{\beta}\|_0 \le q, \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_1 \le Cq\sqrt{\log p / N} \}$. Define the class of functions

$$\mathcal{G}_j = \left\{ y x_j \left(\boldsymbol{\beta} I\left\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} \le 1 \right\} - \boldsymbol{\beta}_0 I\left\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) : \boldsymbol{\beta} \in \Omega \right\}$$

with squared integrable envelope function $F(\mathbf{x}, y) = C|x_j|$. With probability at least $1 - p^{-C}$, we have

$$\left\| \frac{1}{N}\sum_i y_i \mathbf{x}_i \left(\boldsymbol{\beta} I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \widehat{\boldsymbol{\beta}} \le 1 \right\} - \boldsymbol{\beta}_0 I\left\{ y_i \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) - E y \mathbf{x} \left(\boldsymbol{\beta} I\left\{ y \mathbf{x}^{\mathrm{T}} \widehat{\boldsymbol{\beta}} \le 1 \right\} - \boldsymbol{\beta}_0 I\left\{ y \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) \right\|_\infty$$
$$\le C \left(\sigma \sqrt{\frac{q \log p}{N}} + \sqrt{\log p} \frac{q \log p}{N} \right),$$

where $\sigma^2 = \sup_{f \in \mathcal{G}_j} Pf^2$. Next, we need to decide the order of σ^2. For $\boldsymbol{\beta} \in \Omega$, using basic inequality $2ab \le a^2 + b^2$, we have that

$$\begin{aligned}
&E\left[\left(y \mathbf{x}^T \left(\boldsymbol{\beta} I\left\{ \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} \le 1 \right\} - \boldsymbol{\beta}_0 I\left\{ \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) \right)^2 \mid y = 1, \mathbf{x} \right] \\
\le & E\left[\left(\mathbf{x}^T \boldsymbol{\beta}_0 \left(I\left\{ \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} \le 1 \right\} - I\left\{ \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right) + \left(\mathbf{x}^T (\boldsymbol{\beta} - \boldsymbol{\beta}_0) I\left\{ \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} \le 1 \right\} \right) \right)^2 \mid \mathbf{x} \right] \\
\le & 2E\left[(\mathbf{x}^T \boldsymbol{\beta}_0)^2 \left(I\left\{ \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} \le 1 \right\} - I\left\{ \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta}_0 \le 1 \right\} \right)^2 | \mid \mathbf{x} \right] + 2E\left[\left(\left[\mathbf{x}^T (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right]^2 I\left\{ \mathbf{x}^{\mathrm{T}} \boldsymbol{\beta} \le 1 \right\} \right) \mid \mathbf{x} \right] \\
\le & 2\|\mathbf{x}\|_\infty^2 \|\boldsymbol{\beta}_0\|_1^2 \left| \mathbf{x}^T (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right| + 2 \left| \mathbf{x}^T (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \right|^2 \\
\le & 2\sqrt{q} \|\boldsymbol{\beta}_0\|_1^2 \|\mathbf{x}\|_\infty^3 \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\| + 2q \|\mathbf{x}\|_\infty^2 \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|^2 \\
= & O_p \left(\frac{q^{3/2} (\log p)^2}{\sqrt{n}} + \frac{q^3 (\log p)^2}{n} \right).
\end{aligned}$$

Therefore, $\sigma^2 = O\left(\frac{q^{3/2} (\log p)^2}{\sqrt{n}} + \frac{q^3 (\log p)^2}{n} \right)$. Thus, we complete the proof of Proposition 3.

□

3.2. A Communication-Efficient SVM with SCAD Penalty

In this section, we further discuss the advantage of a distributed non-convex penalized SVM in ultra-high dimension. Similarly, the oracle property of distributed non-convex penalized SVM coefficients are be investigated.

Our main results were established under the following assumptions.

(**C1**) The densities of $\boldsymbol{X}$ given $Y = 1$ and $Y = -1$ are continuous and have common support in R^q.

(**C2**) The densities of $\boldsymbol{X}$ given $Y = 1$ and $Y = -1$ have finite second moments.

(**C3**) The true model dimension $q = O(N^{c_1})$ for some $0 \leqslant c_1 < \frac{1}{2}$.

(**C4**) $\lambda_{\max}\left(N^{-1} \boldsymbol{X}_A^{\mathrm{T}} \boldsymbol{X}_A \right) \leqslant M_1$ for a constant $M_1 > 0$, where $\lambda_{\max}$ denotes the largest eigenvalue and $\boldsymbol{X}_A$ is the first $q_N + 1$ columns of the design matrix.

(**C5**) $\lambda_{\min}\{H(\beta_{01})\} \geqslant M_2$ for some constant $M_2 > 0$, where $\lambda_{\min}$ denotes the smallest eigenvalue.

(**C6**) There exist constants $M_3 > 0$ and $2c_1 < c_2 \leqslant 1$ such that $N^{(1-c_2)/2} \min_{1\leqslant j\leqslant q_N} |\beta_{0j}| \geqslant M_3$.

(**C7**) f is uniformly bounded away from 0 and ∞ in a neighborhood of 1, and g is uniformly bounded away from 0 and ∞ in a neighborhood of -1, where f and g are the conditional densities of $\boldsymbol{X}^T\boldsymbol{\beta_{01}}$ given $Y = 1$ and $Y = -1$, respectively.

Remark 3. *Assumptions (**C1**)–(**C2**) and (**C4**)–(**C5**) are similar to the assumptions in Section 3.1, which have been used by [25]. Assumption (**C3**) controled the divergence rate of the number of nonzero coefficients, which could not be faster than $\sqrt{N}$. In addition, see Remark 2. Assumption (**C6**) simply required that the signals could not decay too quickly, which implied that the relevant signals were not too small so that it could be identified, which is common in the literature of high-dimensional problems. Assumption (**C7**) was trivially held by the unbounded support of the conditional distribution of $\boldsymbol{X}_A$ given Y. See Remark 1 in [25].*

First, we introduced the oracle estimator $\dot{\boldsymbol{\beta}} = \left(\hat{\boldsymbol{\beta}}_1, \mathbf{0}\right)$, where $\hat{\boldsymbol{\beta}}_1$ was estimated by covariates associated with the true model, and $\left\|\hat{\boldsymbol{\beta}}_1 - \boldsymbol{\beta}_{01}\right\| = O_p\{\sqrt{q_N/N}\}$ when $N \to \infty$ (based on all dataset) in [25].

In the non-convex penalty, there might be multiple local minimums. We used $B_N(\lambda)$ to denote the set of local minimums. T he non-convex problem could be written as the difference of two convex functions, and then presented as a sufficient local optimal condition.

Let

$$f(\boldsymbol{\beta}) = \widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}}\left(\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}})\right) + \sum_{j=1}^{p} p_\lambda(|\beta_j|).$$

Although $f(\boldsymbol{\beta})$ is non-convex, we can write it as

$$f(\boldsymbol{\beta}) = g(\boldsymbol{\beta}) - h(\boldsymbol{\beta}),$$

where

$$g(\boldsymbol{\beta}) = n^{-1}\sum_{i=1}^{n}\left(1 - Y_i\mathbf{X}_i^{\mathrm{T}}\boldsymbol{\beta}\right)_+ + \lambda\sum_{j=1}^{p}|\beta_j| - \boldsymbol{\beta}^{\mathrm{T}}\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}})$$

and

$$h(\boldsymbol{\beta}) = \lambda\sum_{j=1}^{p}|\beta_j| - \sum_{j=1}^{p} p_\lambda(|\beta_j|) - \boldsymbol{\beta}^{\mathrm{T}}\nabla\widehat{L}_N(\widetilde{\boldsymbol{\beta}}).$$

Obviously, $h(\boldsymbol{\beta})$ and $g(\boldsymbol{\beta})$ are convex.

To present our main results, we need a sufficient local optimal condition based on subgradient estimation as described below.

Lemma 1. *(Sufficient local optimal condition) if there is a neighbourhood U around the point $\mathbf{x}^*$ such that $\partial h(\mathbf{x}) \cap \partial g(\mathbf{x}^*) \neq \emptyset, \forall \mathbf{x} \in U \cap \mathrm{dom}(g)$, then $\mathbf{x}^*$ is a local minimum of $g(\mathbf{x}) - h(\mathbf{x})$.*

Lemma 1 has been stated as Corollary 1 in [32]. The main results are summarized in the following theorem.

Theorem 2. *Assume that assumptions (C1)–(C7) hold, the oracle estimator satisfies*

$$P\{\dot{\boldsymbol{\beta}} \in B_N(\lambda)\} \to 1,$$

when $N \to \infty$, $\lambda = o\left(N^{-(1-c_2)/2}\right)$ and $q\log p\log(N)N^{-1/2} = o(\lambda)$.

Remark 4. *From Theorem 2, we could see that the oracle estimator held when taking* $\lambda = N^{-1/2+\delta}$ *for some* $c_1 < \delta < c_2/2$ *even for* $p = o\left(\exp(N^{(\delta-c_1)/2})\right)$. *So, the local oracle property held for a non-convex distributed penalized SVM even when the number of features,* p, *grew exponentially with the sample size,* N, *of the whole dataset.*

Proof of Theorem 2. We sketch our proof as follows:

Step 1. From Theorem 1 in [25], we obtain some properties about $s_j(\dot{\boldsymbol{\beta}})$ and $\hat{\beta}_j$, with probability approaching 1,

$$s_j(\dot{\boldsymbol{\beta}}) = 0, \quad j = 0, 1, \ldots, q,$$

$$|\hat{\beta}_j| \geqslant \left(a + \frac{1}{2}\right)\lambda, \quad j = 1, \ldots, q,$$

$$|s_j(\dot{\boldsymbol{\beta}})| \leqslant \lambda, |\hat{\beta}_j| = 0, \quad j = q+1, \ldots, p.$$

Step 2. By Proposition 1, we have with probability approaching $1 - p^{-c}$,

$$\left\|\nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}}) - \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}})\right\|_\infty \leq C\left(\sqrt{\frac{\log p}{N}} + \sqrt{\frac{\log p}{n}} + \frac{q \log p}{n^{3/4}} + \frac{q(\log p)^{3/2}}{n}\right),$$

so that when $n \to \infty$, we obtain $P\left\{\left\|\nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}}) - \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}})\right\|_\infty < \delta\right\} \to 1$ for some $\delta > 0$.

Step 3. Let

$$\mathcal{G} = \{\boldsymbol{\xi} = (\xi_0, \ldots, \xi_p)\},$$

where

$$\xi_0 = \nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}})_0,$$

$$\xi_j = \lambda \operatorname{sgn}(\dot{\boldsymbol{\beta}})_j + \nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}})_j, \qquad j = 1, \ldots, q,$$

$$\xi_j = s_j(\boldsymbol{\beta}) + \nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}})_j + \lambda l_j, \qquad j = q+1, \ldots, p,$$

$$l_j \in [-1, 1], \qquad j = q+1, \ldots, p.$$

By Step 1, we obtain $P\{\mathcal{G} \subseteq \partial g(\dot{\boldsymbol{\beta}})\} \to 1$. Then we show that there exist $\xi^* \in \mathcal{G}$ such that $P\left\{\xi_j^* = \partial h(\boldsymbol{\beta})/\partial \beta_j\right\} \to 1$ as $n \to \infty$ for any $\boldsymbol{\beta}$ in $\mathbf{R}^{p+1}$ with center $\dot{\boldsymbol{\beta}}$ and radius $\lambda/2$.

Since $\partial h(\boldsymbol{\beta})/\partial \beta_0 = \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}})_0$, by Step 2 we have $\xi_0^* = \partial h(\boldsymbol{\beta})/\partial \beta_0$.

For $j = 1, \ldots, q$, we have $\min_{1 \leqslant j \leqslant q}|\beta_j| \geqslant \min_{1 \leqslant j \leqslant q}|\hat{\beta}_j| - \max_{1 \leqslant j \leqslant q}|\hat{\beta}_j - \beta_j| \geqslant \left(a + \frac{1}{2}\right)\lambda - \lambda/2 = a\lambda$ with probability 1 by Step 1. Therefore, by Property 2 of the class of penalties $P\left\{\partial h(\beta)/\partial \beta_j = \xi_j = \lambda \operatorname{sgn}(\beta_j) + \nabla \widehat{L}_N(\widetilde{\boldsymbol{\beta}})_j\right\} \to 1$ for $j = 1, \ldots, q$. For sufficiently large n, $\operatorname{sgn}(\beta_j) = \operatorname{sgn}(\dot{\beta}_j)$, $\nabla \widehat{L}_N(\widetilde{\boldsymbol{\beta}})_j = \nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}})_j$. Thus we have $\mathrm{P}\left\{\xi_j^* = \partial h(\boldsymbol{\beta})/\partial \beta_j\right\} \to 1$ as $n \to \infty$ for $j = 1, \ldots, q$.

For $j = q+1, \ldots, p$, we have $P\{|\beta_j| \leqslant |\hat{\beta}_j| + |\beta_j - \hat{\beta}_j| \leqslant \lambda\} \to 1$ by Step 1 . Therefore $P\{\partial h(\boldsymbol{\beta})/\partial \beta_j = 0\} \to 1$ for SCAD. By Property 2, $P\{|\partial h(\boldsymbol{\beta})/\partial \beta_j| \leqslant \lambda\} \to 1$ for the class of penalties. By lemma 1, we have $P\{|s_j(\hat{\beta}_j)| \leqslant \lambda\} \to 1$ for $j = q+1, \ldots, p$. We can always find $l_j \in [-1, 1]$ such that $P\left\{\xi_j^* = s_j(\dot{\beta}) + \lambda l_j + \nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}})_j = \partial h(\boldsymbol{\beta})\partial \beta_j\right\}$ $\to 1$ for $j = 1, \ldots, q$. This completes the proof. □

In this paper, we did not need to assume that the solution of the minimum problem was unique. By some numerical algorithms, which solve the non-convex penalized SCADSVM, we could identify the oracle estimator. Ref. [33] introduced the LLA algorithm to obtain the sparse estimator in non-convex penalized likelihood models. We applied the LLA algorithm to our SCADSVM approach. Now, we flesh out the problem.

Let $\widetilde{\boldsymbol{\beta}}^{(0)} = \left(\widetilde{\beta}_0^{(0)}, \ldots, \widetilde{\beta}_p^{(0)}\right)^{\mathrm{T}}$. We update $\widetilde{\boldsymbol{\beta}}^{(t)}$ by solving

$$\min_{\boldsymbol{\beta}} \left\{ \widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}} \left(\nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}}^{(t-1)}) - \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}}^{(t-1)}) \right) + \sum_{j=1}^{p} p'_\lambda \left(\left| \widetilde{\beta}_j^{(t-1)} \right| \right) |\beta_j| \right\}.$$

Consider the following events:

(a) $F_{n1} = \left\{ \left| \widetilde{\beta}_j^{(0)} - \beta_{0j} \right| > \lambda, for\ 1 \leqslant j \leqslant p \right\}$;
(b) $F_{n2} = \left\{ \left| \beta_{0j} \right| < (a+1)\lambda, for\ 1 \leqslant j \leqslant q \right\}$;
(c) $F_{n3} = \{$for all subgradients $s(\dot{\boldsymbol{\beta}})$, $\left| s_j(\dot{\boldsymbol{\beta}}) \right| > (1 - 1/a)\lambda$ for some $q+1 \leqslant j \leqslant p$ or $\left| s_j(\dot{\boldsymbol{\beta}}) \right| \neq 0$ for some $0 \leqslant j \leqslant q\}$;
(d) $F_{n4} = \left\{ \left| \hat{\beta}_j \right| < a\lambda, for\ 1 \leqslant j \leqslant q \right\}$.
(e) $F_{n5} = \left\{ \left| \nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}})_j - \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}})_j \right| \geqslant \delta, for\ 1 \leqslant j \leqslant q \right\}$.

The first four events are similar to [25]. Denote $P_{ni} = \mathrm{P}(F_{ni})$, then we have the following Theorem 3.

Theorem 3. *Using LLA algorithm initiated by $\widetilde{\boldsymbol{\beta}}^{(0)}$, we can obtain the oracle estimator after two iterations with probability at least $1 - P_{n1} - P_{n2} - P_{n3} - P_{n4} - P_{n5}$.*

Remark 5. *Theorem 3 gave a non-asymptotic lower bounded probability, which implied the oracle estimator could be obtained by the LAA algorithm. That is, the LAA algorithm could identify the oracle estimator in two iterations.*

Proof of Theorem 3. Assume that none of the events F_{ni} is true, for $i = 1, \ldots, 5$. The probability that none of these event is true is at least $1 - P_{n1} - P_{n2} - P_{n3} - P_{n4} - P_{n5}$. Then we have

$$\begin{aligned} \left| \widetilde{\beta}_j^{(0)} \right| = \left| \widetilde{\beta}_j^{(0)} - \beta_{0j} \right| \leqslant \lambda, \qquad & q+1 \leqslant j \leqslant p, \\ \left| \widetilde{\beta}_j^{(0)} \right| \geqslant \left| \beta_{0j} \right| - \left| \widetilde{\beta}_j^{(0)} - \beta_{0j} \right| \geqslant a\lambda, \qquad & 1 \leqslant j \leqslant q. \end{aligned}$$

□

By properties of the class of non-convex penalties, we have $p'_\lambda \left(\left| \widetilde{\beta}_j^{(0)} \right| \right) = 0$ for $1 \leqslant j \leqslant q$. Therefore, the solution of the next iteration of $\widetilde{\boldsymbol{\beta}}^{(1)}$ is the solution to the convex optimization

$$\widetilde{\boldsymbol{\beta}}^{(1)} = \arg\min_{\boldsymbol{\beta}} \widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}} \left(\nabla \widehat{L}_1(\widetilde{\boldsymbol{\beta}}^{(0)}) - \nabla \widehat{L}(\widetilde{\boldsymbol{\beta}}^{(0)}) \right) + \sum_{j=q+1}^{p} p'_\lambda \left(\left| \widetilde{\beta}_j^{(0)} \right| \right) |\beta_j|.$$

By the fact that F_{n3} is not true, there are some subgradients of oracle estimator $s(\dot{\boldsymbol{\beta}})$ such that $s_j(\dot{\boldsymbol{\beta}}) = 0$ for $0 \leqslant j \leqslant q$ and $\left| s_j(\dot{\boldsymbol{\beta}}) \right| < (1 - 1/a)\lambda$ for $q+1 \leqslant j \leqslant p$. By the definition of subgradient, we have

$$\begin{aligned} \widehat{L}_1(\boldsymbol{\beta}) &\geq \widehat{L}_1(\dot{\boldsymbol{\beta}}) + \sum_{0 \leqslant j \leq p} s_j(\dot{\boldsymbol{\beta}})(\beta_j - \hat{\beta}_j) \\ &= \widehat{L}_1(\dot{\boldsymbol{\beta}}) + \sum_{q+1 \leqslant j \leqslant p} s_j(\hat{\boldsymbol{\beta}})(\beta_j - \hat{\beta}_j). \end{aligned}$$

Then we have for any $\boldsymbol{\beta}$

$$\begin{aligned}
&\left\{\widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}}\left(\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}}^{(0)}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}}^{(0)})\right) + \sum_{j=q+1}^{p} p'_\lambda\left(\left|\widetilde{\beta}_j^{(0)}\right|\right)|\beta_j|\right\} \\
&- \left\{\widehat{L}_1(\dot{\boldsymbol{\beta}}) - \dot{\boldsymbol{\beta}}^{\mathrm{T}}\left(\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}}^{(0)}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}}^{(0)})\right) + \sum_{j=q+1}^{p} p'_\lambda\left(\left|\widetilde{\beta}_j^{(0)}\right|\right)|\hat{\beta}_j|\right\} \\
&\geq \sum_{q+1\leqslant j\leqslant p}\left\{p'_\lambda\left(\left|\widetilde{\beta}_j^{(0)}\right|\right) - s_j(\hat{\boldsymbol{\beta}})\operatorname{sgn}(\beta_j)\right\}|\beta_j| - (\boldsymbol{\beta}-\dot{\boldsymbol{\beta}})^{\mathrm{T}}\left(\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}}^{(0)}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}}^{(0)})\right) \\
&\geq \sum_{q+1\leqslant j\leqslant p}\left\{(1-1/a)\lambda - s_j(\hat{\boldsymbol{\beta}})\operatorname{sgn}(\beta_j)\right\}|\beta_j| - (\boldsymbol{\beta}-\dot{\boldsymbol{\beta}})^{\mathrm{T}}\left(\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}}^{(0)}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}}^{(0)})\right) \\
&\geq 0\,.
\end{aligned}$$

So we can obtain $\widetilde{\boldsymbol{\beta}}^{(1)} = \dot{\boldsymbol{\beta}}$. This proves that the LLA algorithm finds the oracle estimator after one iteration.

If F_{n2} is not true, one obtains $|\hat{\beta}_j| > a\lambda$ for all $1 \leqslant j \leqslant q$. So we have $p'_\lambda(|\hat{\beta}_j|) = 0$ for all $1 \leqslant j \leqslant q$ and $p'_\lambda(|\hat{\beta}_j|) = p'_\lambda(0) = \lambda$ for all $q+1 \leqslant j \leqslant p$ by Property 2 of the class of penalties. At iteration 1, when the LLA algorithm has found $\hat{\boldsymbol{\beta}}$, the solution to the next LLA iteration $\tilde{\boldsymbol{\beta}}^{(2)}$ is the minimum of the convex optimization problem

$$\tilde{\boldsymbol{\beta}}^{(2)} = \arg\min_{\boldsymbol{\beta}} \widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}}\left(\nabla\widehat{L}_1(\tilde{\boldsymbol{\beta}}^{(1)}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}}^{(1)})\right) + \sum_{q+1\leqslant j\leqslant p}\lambda|\beta_j|.$$

Then we have for any $\boldsymbol{\beta}$

$$\begin{aligned}
&\left\{\widehat{L}_1(\boldsymbol{\beta}) - \boldsymbol{\beta}^{\mathrm{T}}\left(\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}}^{(1)}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}}^{(1)})\right) + \sum_{q+1\leqslant j\leqslant p}\lambda|\beta_j|\right\} \\
&- \left\{\widehat{L}_1(\dot{\boldsymbol{\beta}}) - \dot{\boldsymbol{\beta}}^{\mathrm{T}}\left(\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}}^{(1)}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}}^{(1)})\right) + \sum_{q+1\leqslant j\leqslant p}\lambda|\beta_j|\right\} \\
&\geq \sum_{q+1\leqslant j\leqslant p}\{\lambda - s_j(\dot{\boldsymbol{\beta}})\operatorname{sgn}(\beta_j)\}|\beta_j| - (\boldsymbol{\beta}-\dot{\boldsymbol{\beta}})^{\mathrm{T}}\left(\nabla\widehat{L}_1(\widetilde{\boldsymbol{\beta}}^{(0)}) - \nabla\widehat{L}(\widetilde{\boldsymbol{\beta}}^{(0)})\right) \\
&\geq 0.
\end{aligned}$$

Hence, the iteration 2 finds an oracle estimator $\widetilde{\boldsymbol{\beta}}^{(2)} = \dot{\boldsymbol{\beta}}$ again and the algorithm stops.

4. Numerical Experiments

4.1. Simulation Experiments

We considered four models to evaluate the finite sample performance of the distributed SVM. The first and the second models were similar to models 1 and 2 of [27], respectively. The first model was essentially a standard linear discriminant analysis which was used in [19,24,25]. The other three models were probit regression under a different setting. In numerical simulation experiments, we generated the data in R, and used CPLEX solver to solve the optimization problem in AMPL for model 1. In addition, we used Python for the other models.

Model 1: $\Pr(Y=1) = \Pr(Y=-1) = 0.5, \boldsymbol{X}^*|(Y=1) \sim \mathrm{MN}(\mu,\Sigma), \boldsymbol{X}^*|(Y=-1) \sim \mathrm{MN}(-\boldsymbol{\mu},\boldsymbol{\Sigma}), q = 5, \boldsymbol{\mu} = (0.1, 0.2, 0.3, 0.4, 0.5, 0, \ldots, 0)^{\mathrm{T}} \in \mathbf{R}^p, \boldsymbol{\Sigma} = (\sigma_{ij})$ with non-zero elements $\sigma_{ii} = 1$ for $i = 1, 2, \ldots, p$ and $\sigma_{ij} = \rho = -0.2$ for $1 \leqslant i \neq j \leqslant q$. The Bayes rule is $\operatorname{sgn}(2.67X_1 + 2.83X_2 + 3X_3 + 3.17X_4 + 3.33X_5)$ with Bayes error 6.3%.

Model 2: $X^* \sim \mathrm{MN}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma} = (\sigma_{ij})$ and $\sigma_{ij} = 0.4^{|i-j|}$ for $1 \leqslant i \neq j \leqslant p, \sigma_{ij} = 1$ for $i = j$, $\Pr(Y = 1 \mid \mathbf{X}^*) = \Phi\left\{(\mathbf{X}^*)^{\mathrm{T}}\boldsymbol{\beta}^*\right\}$, where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution. The Bayes rule is $\mathrm{sgn}(1X_1 + 1X_2 + 1X_3 + 1X_4)$ with Bayes error 10.4%.

Model 3: $X_{ij} \sim \mathrm{MN}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma} = (\sigma_{ij})$ and $\sigma_{ij} = 0.5^{|i-j|}$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$, $\Pr(Y = 1 \mid \mathbf{X}^*) = \Phi\left\{(\mathbf{X}^*)^{\mathrm{T}}\boldsymbol{\beta}^*\right\}$ where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution. The true parameter $\boldsymbol{\beta}_0$ is set to be sparse and it's first q entries are uniformly distributed i.i.d. random variables from $[0, 1]$.

Model 4: $X_{ij} \sim \mathrm{MN}(\mathbf{0}, \mathbf{\Sigma}), \mathbf{\Sigma} = (\sigma_{ij})$ and $\sigma_{ij} = 0.5^{|i-j|/5}$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m$, $\Pr(Y = 1 \mid \mathbf{X}^*) = \Phi\left\{(\mathbf{X}^*)^{\mathrm{T}}\boldsymbol{\beta}^*\right\}$, where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution. The true parameter $\boldsymbol{\beta}_0$ is set to be sparse and it's first q entries are uniformly distributed i.i.d. random variables from $[0, 1]$.This is an ill-conditioned case for model 3.

For model 1, we used the dimension $p = 500$ and $p = 1000$, local sample size $n = 200$ and 500 , number of machines $m = 5, 10, 15, 20$. For models 2–4, the dimension $p = 1000$, number of machines $m = 5, 10, 20$, and total sample size $N = nm = 10{,}000$.

We compared the finite sample performances of the following four estimators:

- L1SVM algorithm: the proposed communication-efficient estimator $\check{\boldsymbol{\beta}}^{L1}$;
- SCADSVM algorithm: the proposed communication-efficient estimator $\check{\boldsymbol{\beta}}^{SCAD}$;
- Cen algorithm: the central estimator $\widehat{\boldsymbol{\beta}}^{\mathrm{Cen}}$, which computes the l_1-regularized estimator using all of the dataset;
- Sub algorithm: the sub-data estimator $\widehat{\boldsymbol{\beta}}^{\mathrm{Sub}}$, which computes the l_1-regularized estimator using data only on the first machine.

We use the first model to compare the performance of variable selection among the above four algorithms, which are listed in Table 1; and use the other models to evaluate the estimation errors of parameters of algorithms via MSE, which are presented Figures 1–3.

The numbers in Table 1 are the number of zero coefficients incorrectly estimated to be nonzero. The number of nonzero coefficients incorrectly estimated is zero, which meant all of the four algorithms could find the relevant variables; hence, they are not listed. From Table 1, we observed that:

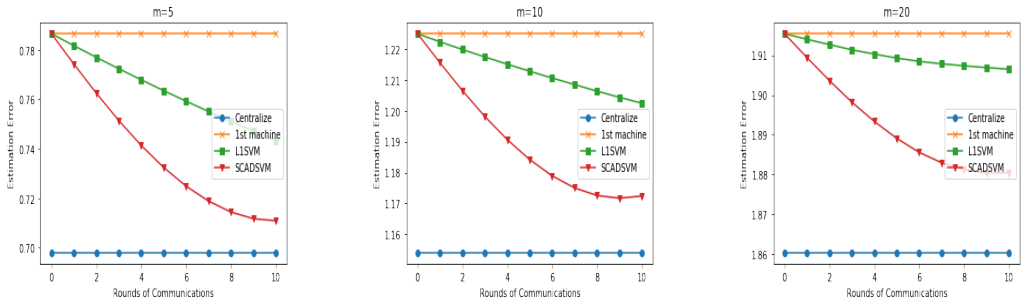
(i) The centralized algorithm was the best among these algorithms because it used the information of the whole dataset.
(ii) The sub algorithm was bad because it only used the information of the data on the first machine.
(iii) Our proposed L1SVM and SCADSVM could both select relevant variables, and the SCADSVM had a better performance than L1SVEM. This implied the non-convex SCADSVM algorithm was more robust than convex L1SVEM, especially for the complex models and massive datasets.
(iv) When $N = mn$ was large, our two proposed distributed SVM algorithms were as good as the centralized algorithm.

We gave the prediction error analysis for models 2–4, see Figures 1–3. From Figures 1–3, we have the following observations:

(i) The central algorithm was still the best classifier, but had the highest communication cost and risk of privacy leakage.There was a big gap between the sub estimator and the centralized estimator.
(ii) Our two proposed communication-efficient estimators could match the central estimator with a few rounds of communication. The prediction errors of SCADSVM were lower than that of L1SVM, and it was more robust than L1SVM.

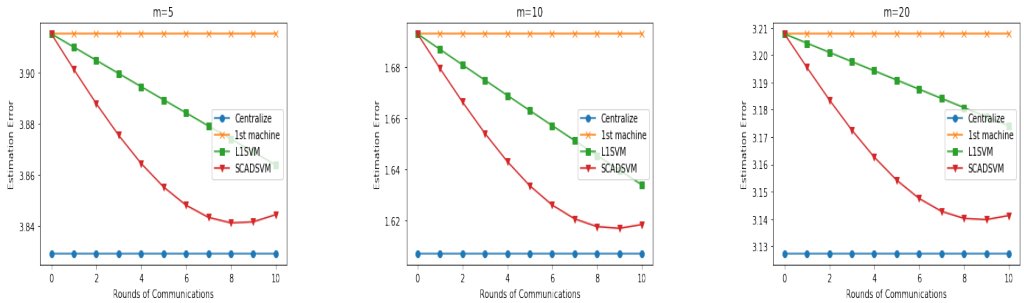
Table 1. Variable selection results for Model 1.

$n = 200, p = 500$				
m	Sub	L1SVM	SCADSVM	Cen
5	21	2	1	0
10	22	3	4	0
15	28	0	1	0
20	24	0	0	0
$n = 200, p = 1000$				
m	Sub	L1SVM	SCADSVM	Cen
5	49	8	7	0
10	39	2	0	0
15	42	1	0	0
20	42	1	1	0
$n = 400, p = 500$				
m	Sub	L1SVM	SCADSVM	Cen
5	4	4	0	0
10	3	0	0	0
15	5	0	0	0
20	1	0	0	0
$n = 400, p = 1000$				
m	Sub	L1SVM	SCADSVM	Cen
5	6	0	0	0
10	7	0	0	0
15	4	0	0	0
20	4	0	0	0



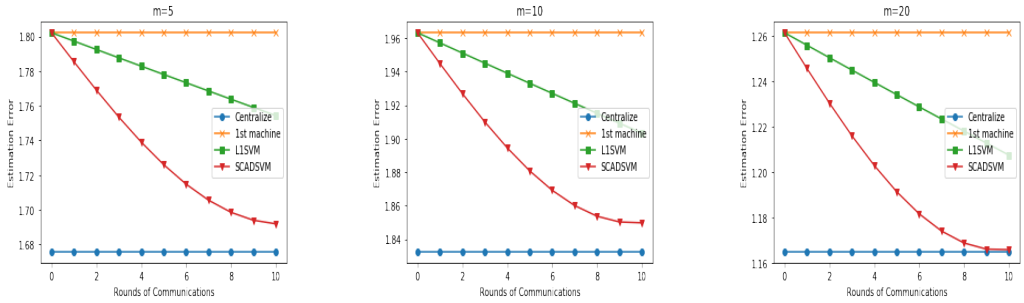
$$p = 1000, q = 5, \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \sigma_{ij} = 0.4^{|i-j|}$$

Figure 1. Prediction error analysis vs. rounds of communication when $m = 5$, 10, and 20 for model 2.



$$p = 1000, q = 5, \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \sigma_{ij} = 0.5^{|i-j|}$$

Figure 2. Prediction error analysis vs. rounds of communication when $m = 5$, 10, and 20 for model 3.



$$p = 1000, q = 5, \mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \sigma_{ij} = 0.5^{|i-j|/5}$$

Figure 3. Prediction error analysis vs. rounds of communication when $m = 5$, 10, and 20 for model 4.

4.2. Real Data

In this subsection, we verify the performance of the CSLSVM algorithm (L1SVM and SCADSVM) using three real datasets. We use 'a9a', 'w8a', and 'phishing' datasets from the LIBSVM website (https://www.csie.ntu.edu.tw/∼cjlin/libsvm/ accessed on 12 February 2022). These real datasets are listed Table 2. The 'a9a' dataset was an adult dataset from the 1994 Census database. The prediction task was to determine whether a person makes over 50K a year or not. The 'w8a' dataset was also based on the Census database, but it had more features than 'a9a'. The phishing dataset aimed to predict phishing websites. Phishing is the process by which a fraudster impersonates a legitimate person by simulating the same or similar web pages or websites to steal personal or private information for illegal political and economic gain. As phishing is becoming more and more serious, phishing web detection is gaining attention as an anti-phishing measure and technique.

Approximately 80% of the data was used to train the model and the remaining was applied to test the model. In distributed learning, we used the number of worker machines m = 5, 10, and 20, respectively. The results of classification errors for the three datasets are provided in Figures 4–6. From Figures 4–6, we found the following:

(i) Since these datasets had no well-specified model, the curves behaved quite differently on these datasets. However, overall there was a large gap between the sub algorithm and centralized solution.
(ii) In most of the cases, the distributed L1SVM algorithm still converged quite slowly.
(iii) The proposed distributed SCADSVM could obtain a solution that was highly competitive with the centralized model within a few rounds of communications, and was more robust than the distributed L1SVM.

The experimental results on simulated and real datasets verified that the proposed distributed SCADSVM/L1SVM algorithms were two effective procedures for distributed sparse learning on classification via the SVM technique, which maintained efficiency in both communication and computation.

Table 2. Real data used in the experiments.

Data Name	Number of Data	Features	Task
a9a	48,842	123	Classification
w8a	64,700	301	Classification
phishing	11,055	68	Classification

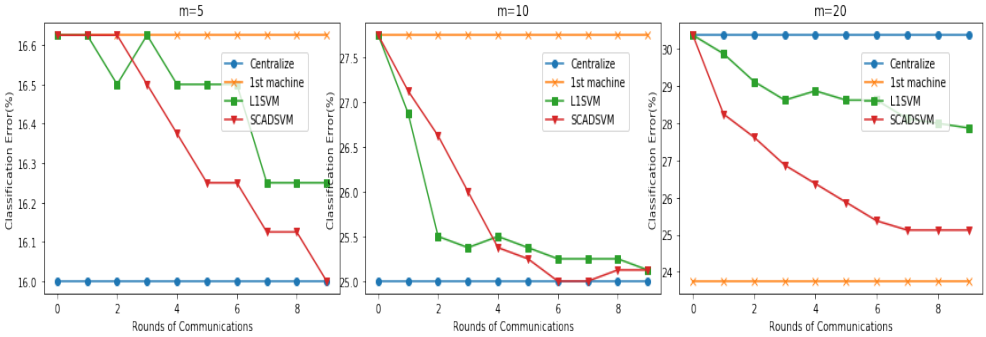


Figure 4. Classification error vs. rounds of communications for 'a9a' data.

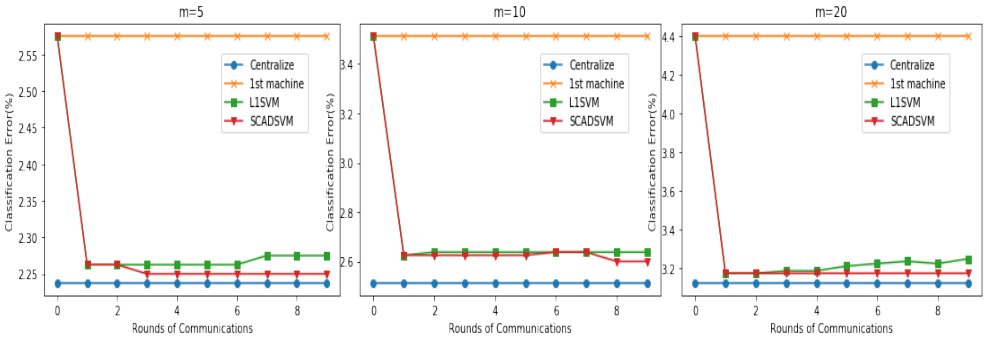


Figure 5. Classification error vs. rounds of communications for 'w8a' data.

In the computational effort of the four methods, by numerical experiments, we also observed the following. For the central algorithm, the whole dataset was used to train the model, so it was the most accurate estimation, but had the highest computational cost and had a risk of privacy leakage. The sub-data estimation algorithm had the least computational cost because of the small amount of data, and no communication was required; however, it had the largest estimation error. Our proposed L1SVM and SCADSVM algorithms were communication efficient and computational efficient using the CSL framework, and could match the central estimator.

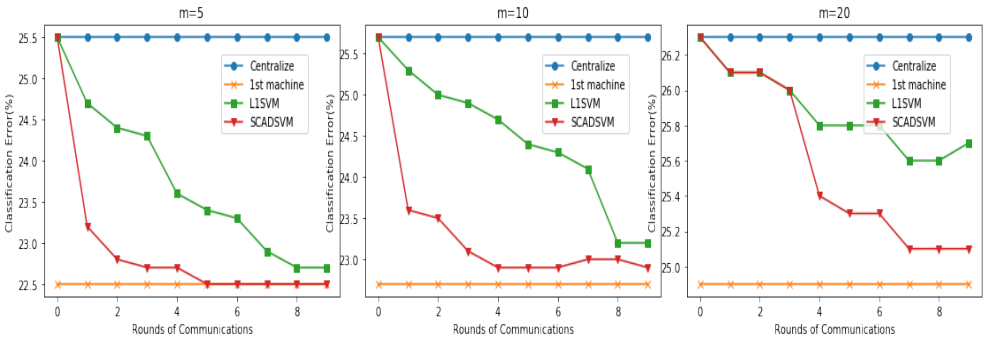


Figure 6. Classification error vs rounds of communications for 'phishing' data.

5. Conclusions

In the paper, we proposed a novel distributed CSLSVM learning algorithm with convex (l_1)/nonconvex (SCAD) penalties based on a communication-efficient surrogate likelihood (CSL) framework, which wads efficient in both communication and computation. For CSLSVM with l_1 penalty, we proved that the estimator of L1SVM could achieve a near-oracle property for an l_1 penalized SVM estimator based on the whole datasets. For CSLSVM with SCAD penalty, we showed that the estimator of SCADSVM enjoyed the oracle property, i.e., one of the local minimums of the distributed non-convex penalized

SVM behaved similarly to the oracle estimator based on the whole dataset, as if the true sparsity was known in advance and only the relevant features were found to form the decision boundary. We also showed that, as long as the initial estimator was appropriate, the oracle estimator could be identified with a probability tending to 1. Extensive experiments on both simulated and real data illustrated that the proposed SCLSVM algorithm improved the performance of the work on the first worker machine and matched the centralized method. In addition, the proposed distributed SCADSVM could obtain a solution that was highly competitive with the centralized model within a few rounds of communication, and was more robust than the distributed L1SVM.

Author Contributions: Conceptualization, X.Z. and H.S.; methodology, X.Z.; software, H.S.; validation, H.S.; investigation, X.Z.; writing—original draft preparation, H.S.; writing—review and editing, X.Z.; supervision, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Chinese National Social Science Fund (Grant No. 19BTJ034), National Natural Science Foundation of China (Grant No. 12171242, 11971235) and Postgraduate Research & Practice Innovation Program of Jiangsu Province (Grant No. KYCX20_1676).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1995.
2. Banerjee, M.; Durot, C.; Sen, B. Divide and conquer in nonstandard problems and the super-efficiency phenomenon. *Ann. Stat.* **2019**, *47*, 720–757. [CrossRef]
3. Jordan, M.I.; Lee, J.D.; Yang, Y. Communication-Efficient Distributed Statistical Inference. *J. Am. Stat. Assoc.* **2019**, *114*, 668–681. [CrossRef]
4. Volgushev, S.; Chao, S.K.; Cheng, G. Distributed inference for quantile regression processes. *arXiv* **2017**, arXiv:1701.06088.
5. Chen, X.; Liu, W.; Mao, X.; Yang, Z. Distributed High-dimensional Regression Under a Quantile Loss Function. *arXiv* **2019**, arXiv:1906.05741.
6. Wang, L.; Lian, H. Communication-efficient estimation of high-dimensional quantile regression. *Anal. Appl.* **2020**, *18*, 1057–1075. [CrossRef]
7. Zhang, Y.; Duchi, J.; Wainwright, M. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *J. Mach. Learn. Res.* **2015**, *16*, 3299–3340.
8. Han, Y.; Mukherjee, P.; Ozgur, A.; Weissman, T. Distributed statistical estimation of high-dimensional and nonparametric distributions. In Proceedings of the 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, 17–22 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 506–510.
9. Wang, X.; Yang, Z.; Chen, X.; Liu, W. Distributed inference for linear support vector machine. *J. Mach. Learn. Res.* **2019**, *20*, 1–41.
10. Zou, H. An improved 1-norm svm for simultaneous classification and variable selection. In Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, PMLR, San Juan, Puerto Rico, 21–24 March 2007; pp. 675–681.
11. Meinshausen, N.; Bühlmann, P. High-dimensional graphs and variable selection with the lasso. *Ann. Stat.* **2006**, *34*, 1436–1462. [CrossRef]
12. Zhao, P.; Yu, B. On model selection consistency of Lasso. *J. Mach. Learn. Res.* **2006**, *7*, 2541–2563.
13. Meinshausen, N.; Yu, B. Lasso-type recovery of sparse representations for high-dimensional data. *Ann. Stat.* **2009**, *37*, 246–270. [CrossRef]
14. Zou, H. The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.* **2006**, *101*, 1418–1429. [CrossRef]
15. Bühlmann, P.; Van De Geer, S. *Statistics for High-Dimensional Data: Methods, Theory and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
16. Giraud, C. Estimator Selection. In *Introduction to High-Dimensional Statistics*; Chapman and Hall: London, UK; CRC: London, UK, 2014; pp. 117–136.
17. Fan, J.; Fan, Y. High dimensional classification using features annealed independence rules. *Ann. Stat.* **2008**, *36*, 2605. [CrossRef]
18. Bradley, P.S.; Mangasarian, O.L. Feature selection via concave minimization and support vector machines. *ICML Citeseer* **1998**, *98*, 82–90.

19. Peng, B.; Wang, L.; Wu, Y. An error bound for l1-norm support vector machine coefficients in ultra-high dimension. *J. Mach. Learn. Res.* **2016**, *17*, 8279–8304.
20. Zhu, J.; Rosset, S.; Tibshirani, R.; Hastie, T.J. 1-norm support vector machines. Advances in neural information processing systems. *Citeseer* **2003**, *16*, 49–56.
21. Wegkamp, M.; Yuan, M. Support vector machines with a reject option. *Bernoulli* **2011**, *17*, 1368–1385. [CrossRef]
22. Fan, J.; Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.* **2001**, *96*, 1348–1360. [CrossRef]
23. Becker, N.; Toedt, G.; Lichter, P.; Benner, A. Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. *BMC Bioinform.* **2011**, *12*, 1–13. [CrossRef]
24. Park, C.; Kim, K.R.; Myung, R.; Koo, J.Y. Oracle properties of scad-penalized support vector machine. *J. Stat. Plan. Inference* **2012**, *142*, 2257–2270. [CrossRef]
25. Zhang, X.; Wu, Y.; Wang, L.; Li, R. Variable selection for support vector machines in moderately high dimensions. *J. R. Stat. Soc. Ser. Stat. Methodol.* **2016**, *78*, 53. [CrossRef]
26. Lian, H.; Fan, Z. Divide-and-conquer for debiased l1-norm support vector machine in ultra-high dimensions. *J. Mach. Learn. Res.* **2017**, *18*, 6691–6716.
27. Wang, J.; Kolar, M.; Srebro, N.; Zhang, T. Efficient Distributed Learning with Sparsity. *arXiv* **2016**, arXiv:1605.07991.
28. Koo, J.Y.; Lee, Y.; Kim, Y.; Park, C. A bahadur representation of the linear support vector machine. *J. Mach. Learn. Res.* **2008**, *9*, 1343–1368.
29. Belloni, A.; Chernozhukov, V. l1-penalized quantile regression in high-dimensional sparse models. *Ann. Stat.* **2011**, *39*, 82–130. [CrossRef]
30. Van Der Vaart, A.W.; van der Vaart, A.W.; van der Vaart, A.; Wellner, J. *Weak Convergence and Empirical Processes: With Applications to Statistics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1996.
31. Koltchinskii, V. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011; Volume 2033.
32. Tao, P.; An, I. Convex analysis approach to D.C. programming theory, algorithms and applications. *Acta Math. Vietnam.* **1997**, *22*, 289–355.
33. Zou, H.; Li, R. One-step sparse estimates in nonconcave penalized likelihood models. *Ann. Stat.* **2008**, *36*, 1509.