

Structured sparse support vector machine with ordered features

Kuangnan Fang, Peng Wang, Xiaochen Zhang & Qingzhao Zhang

To cite this article: Kuangnan Fang, Peng Wang, Xiaochen Zhang & Qingzhao Zhang (2020): Structured sparse support vector machine with ordered features, Journal of Applied Statistics, DOI: [10.1080/02664763.2020.1849053](https://doi.org/10.1080/02664763.2020.1849053)

To link to this article: <https://doi.org/10.1080/02664763.2020.1849053>

 View supplementary material 

 Published online: 18 Nov 2020.

 Submit your article to this journal 

 Article views: 67

 View related articles 

 View Crossmark data 



Structured sparse support vector machine with ordered features

Kuangnan Fang^{a,b}, Peng Wang^a, Xiaochen Zhang^a and Qingzhao Zhang^{a,b,c}

^aDepartment of Statistics, School of Economics, Xiamen University, Xiamen, Fujian, China; ^bKey Laboratory of Econometrics, Ministry of Education, Xiamen University, Xiamen, Fujian, China; ^cThe Wang Yanan Institute for Studies in Economics, Xiamen University, Xiamen, Fujian, China

ABSTRACT

In the application of high-dimensional data classification, several attempts have been made to achieve variable selection by replacing the ℓ_2 -penalty with other penalties for the support vector machine (SVM). However, these high-dimensional SVM methods usually do not take into account the special structure among covariates (features). In this article, we consider a classification problem, where the covariates are ordered in some meaningful way, and the number of covariates p can be much larger than the sample size n . We propose a structured sparse SVM to tackle this type of problems, which combines the non-convex penalty and cubic spline estimation procedure (i.e. penalizing second-order derivatives of the coefficients) to the SVM. From a theoretical point of view, the proposed method satisfies the local oracle property. Simulations show that the method works effectively both in feature selection and classification accuracy. A real application is conducted to illustrate the benefits of the method.

ARTICLE HISTORY

Received 27 November 2019
Accepted 6 November 2020

KEYWORDS

Structured sparse; support vector machine; variable selection; local oracle property

1. Introduction

Classification is one of the most important research fields in statistics and machine learning, and it is also a common practical problem. The support vector machine (SVM)[19] is a powerful classification tool with high accuracy and great flexibility. In this article, we will focus on a classification with n cases having class labels $\{y_i \in \{1, -1\}; i = 1, \dots, n\}$ and features $\{x_{ij}; i = 1, 2, \dots, n, j = 1, 2, \dots, p\}$. The SVM has an equivalent formulation as the ℓ_2 penalized hinge loss [11]:

$$\min_{(\beta_0, \boldsymbol{\beta})} \frac{1}{n} \sum_{i=1}^n \ell\{y_i(\beta_0 + \mathbf{x}_i^\top \boldsymbol{\beta})\} + \frac{\lambda}{2} \|\boldsymbol{\beta}\|^2, \quad (1)$$

where the loss $\ell(t) = [1 - t]_+$ is called hinge loss, and $\|\cdot\|$ is the ℓ_2 -norm. λ is the tuning parameter, which is used to control the tradeoff between loss and penalty.

In the application of high-dimensional data classification, several attempts have been made to achieve variable selection by replacing the ℓ_2 -penalty with other penalties for

CONTACT Qingzhao Zhang  zhangqingzhao@amss.ac.cn

 Supplemental data for this article can be accessed here. <https://doi.org/10.1080/02664763.2020.1849053>

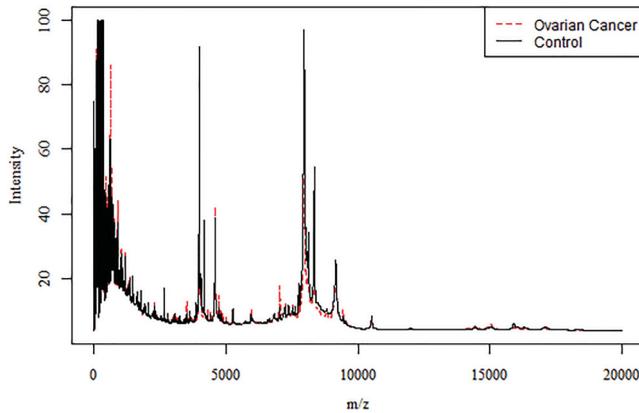


Figure 1. Protein mass spectroscopy data: average profiles from control (—) and ovarian cancer patients (---).

the SVM, such as ℓ_1 -SVM [1,28], ℓ_0 -SVM and ℓ_∞ -SVM [10], ℓ_p -SVM [4], SCAD-SVM [25,27], Hybrid Huberized SVM [20,21], and MCP-SVM [27]. Since the hinge loss does not have the first derivative, it causes some difficulties in the calculation. [6,15] considered square hinge loss in the SVM; [17,20–22] suggested using the Huberized hinge loss in the SVM. [17] point out that the Huberized regularized model paths are both less affected by the outlier than the non-Huberized squared loss.

This paper concerns a class of structured sparse classification problems with ordering features, i.e. x_j that can be ordered as x_1, x_2, \dots, x_p in some sense. A motivating example comes from protein mass spectroscopy. For each blood serum sample i , we observed the intensity x_{ij} for many time-of-flight values t_j . Time of flight is related to the mass over charge ratio m/z of the constituent proteins in the blood. Figure 1 shows an example that protein mass spectroscopy taken from [16]. We plot intensity x_{ij} on the vertical Y -axis against m/z on the horizontal x -axis. The features are ordered in a meaningful way, i.e. x_{ij} are ordered by m/z , which may lead to a high correlation among closely located variables. The SVM methods mentioned above for processing high-dimensional data classification, do not consider the structure in which the variables are arranged in order. Our goal is to predict the label from the ordered features, especially for $p \gg n$.

Besides the above example, there are also many data like this, such as the gene expression data in microarray studies, single nucleotide polymorphisms (SNPs) data in genome-wide association studies (GWAS), graph and image data[9]. Those special structures among variables may lead to successive coefficients vary slowly. Fused lasso [18] encourages flatness of the coefficients by penalizing the ℓ_1 -norm of coefficients' successive differences. However, it may not perform well when the features vary smoothly, rather than being like a step function. To capture the smooth features in a group, smooth-lasso [12] replaces the ℓ_1 -penalty of the difference of the adjacent coefficients in the fused lasso by the ℓ_2 -penalty. Recently, [9] proposed the spline-lasso and spline-MCP, in which they imposed an ℓ_2 -penalty on the discrete version of the second derivatives of coefficients. However, these methods mentioned above are mainly used in the regression.

To the best of our knowledge, the present article is the first to develop theory and methodology for SVM to incorporate the ordered structure among predictors. This study

may advance from the existing ones along the following aspects. First, the structured sparse SVM can achieve variable selection as well as capture the ordered structure of features. The subsequent numerical analysis proves that ignoring this data structure has an impact on the accuracy of the classification and the variable selection. Second, we can theoretically prove that our method has local oracle properties. Even when the number of covariates grows exponentially with the sample size, the local oracle property still holds for the structured sparse SVM. Finally, the algorithm and asymptotic properties are established based on the general form, and we can advocate many kinds of loss functions in the formulation of structured sparse SVM, such as Huberized hinge loss and squared hinge loss.

The rest of the article is organized as follows. In Section 2, we describe the model, an efficient algorithm, and local oracle properties for structured sparse SVM. Simulation results and an application of the proposed method to a protein mass spectroscopy dataset are presented in Sections 3 and 4. Discussions of the proposed method and results are given in Section 5. Proofs for the oracle properties of structured sparse SVM are provided in the Appendix.

2. Structured sparse support vector machine

2.1. Methodology

In this paper, we allow the number of covariates p to increase with the sample size n . It is even possible that p is much larger than n . We assume that the true parameter is sparse, and the features are ordered in some meaningful way. Thus, we need to get an estimator that enjoys the structured sparse property.

The structured sparse SVM is formulated in terms of a loss function that is regularized by penalty terms. Our proposed minimization objective function is

$$\min_{(\beta_0, \beta)} \frac{1}{n} \sum_{i=1}^n \ell\{y_i(\beta_0 + \mathbf{x}_i^\top \beta)\} + \sum_{j=1}^p p_{\lambda_1}(|\beta_j|) + \lambda_2 \sum_{j=2}^{p-1} (\Delta_j^{(2)} \beta)^2. \quad (2)$$

In (2), the first part is a convex loss function. We can advocate many kinds of loss functions. Huberized hinge loss,

$$\ell_\delta(t) = \begin{cases} 0, & t > 1, \\ (1 - t)^2 / (2\delta), & 1 - \delta < t \leq 1, \\ 1 - t - \delta/2, & t \leq 1 - \delta, \end{cases}$$

is adopted in this paper. We fix the pre-specified constant $\delta = 2$ following by [21]. The results with other losses are provided in the supplemental materials.

The second part is used to achieve variable selection. We consider the penalized SVM with a general class of non-convex penalties, such as the smoothly clipped absolute deviation (SCAD) penalty [7] and the minimax concave penalty (MCP) [24]. The SCAD penalty is defined by $p_\lambda(x) = \lambda \int_0^{|x|} \min\{1, (a - t/\lambda)_+ / (a - 1)\} dt$ for some $a > 2$. The MCP is defined by $\lambda \int_0^{|x|} \{1 - t/(a\lambda)\}_+ dt$ for some $a > 1$. The experiments with different a values are presented in supplemental materials. We find our results to be insensitive to these choices, and for brevity, we fixed $a = 3.7$ for SCAD penalty and $a = 3$ for MCP as suggested in the literature [3,7,27,29].

The third part mimics the cubic spline to encourage the smoothness of coefficients. As the coefficient of the variables might vary smoothly, we encourage the smoothness of coefficients by penalizing the ℓ_2 norm of the discrete version of the second-order derivatives of the coefficients. Denote the first- and second-order difference (or discrete versions of derivatives) of the coefficients by $\Delta_j \boldsymbol{\beta} =: (\beta_{j+1} - \beta_j)$ and $\Delta_j^{(2)} \boldsymbol{\beta} = \Delta_j \boldsymbol{\beta} - \Delta_{j-1} \boldsymbol{\beta} = \beta_{j+1} - 2\beta_j + \beta_{j-1}$. Then the ℓ_2 norm of the discrete version of the second-order derivatives of the coefficients is $\sum_{j=2}^{p-1} (\Delta_j^{(2)} \boldsymbol{\beta})^2$. The estimator by minimizing (2) enjoys structured sparse property.

Remark 2.1: The idea of the third part in (2) is similar to spline-lasso [9], which is used in the regression. The computation of spline-lasso is converted to lasso through a certain transformation. However, the conversion is no longer applicable when we solve the SVM problem. The computation is more complicated for the SVM problem than regression. The details of the algorithm are shown in Section 2.2.

2.2. Algorithm

Then we give the algorithm to solve this problem. Without loss of generality, we assume that the input data are standardized: $\sum_{i=1}^n x_{ij}/n = 0$, $\sum_{i=1}^n x_{ij}^2/n = 1$, $j = 1, 2, \dots, p$. We use the generalized coordinate descent (GCD) algorithm [22] to calculate the structured sparse SVM problem.

When using the GCD algorithm, the loss function $\ell(\cdot)$ should satisfy the following condition

$$\ell(t + a) \leq \ell(a) + \ell'(a)t + \frac{M}{2}t^2 \quad \forall a, t, \quad (3)$$

where M is a constant greater than 0. The corresponding M value is $2/\delta$ for Huberized hinge loss. It can be proved that the common loss functions such as Huberized hinge loss, logistic loss, and square hinge loss satisfy the above condition. Although the hinge loss, the loss for the standard SVM, does not satisfy (3), [21] showed that Huberized hinge function with the parameter $\delta = 0.01$ is nearly identical to the hinge loss.

Let D be a $(p-2) \times p$ matrix with $D_{ii} = D_{i,i+2} = 1$, $D_{i,i+1} = -2$, and $D_{ij} = 0$ otherwise. Given current estimate $\{\tilde{\beta}_0, \tilde{\boldsymbol{\beta}}\}$. Define the current margin $r_i = y_i(\tilde{\beta}_0 + \mathbf{x}_i^\top \tilde{\boldsymbol{\beta}})$. The coordinate descent algorithm cyclically minimizes

$$\begin{aligned} F(\beta_j | \tilde{\beta}_0, \tilde{\boldsymbol{\beta}}) &= \frac{1}{n} \sum_{i=1}^n \ell\{r_i + y_i x_{ij}(\beta_j - \tilde{\beta}_j)\} + p_{\lambda_1}(|\beta_j|) \\ &\quad + \lambda_2 (D^\top D)_{jj} \beta_j^2 + 2\lambda_2 \sum_{l=1, l \neq j}^n (D^\top D)_{lj} \tilde{\beta}_l \beta_j \end{aligned} \quad (4)$$

with respect to β_j . According to local linear approximation (LLA) [29], we have $p_{\lambda_1}(|\beta_j|) \approx p_{\lambda_1}(|\tilde{\beta}_j|) + p'_{\lambda_1}(|\tilde{\beta}_j|)(|\beta_j| - |\tilde{\beta}_j|)$, for $\beta_j \approx \tilde{\beta}_j$. As pointed out by a referee, CCCP (constrained concave-convex procedure) algorithm is also an efficient algorithm for solving this problem, which is worth investigating as future work. When $\ell(\cdot)$ satisfies (3), we can

get $F(\beta_j|\tilde{\beta}_0, \tilde{\beta}) \leq \hat{F}(\beta_j|\tilde{\beta}_0, \tilde{\beta})$, where

$$\begin{aligned} \hat{F}(\beta_j|\tilde{\beta}_0, \tilde{\beta}) &= \frac{1}{n} \sum_{i=1}^n \ell(r_i) + \frac{1}{n} \sum_{i=1}^n \ell'(r_i) y_i x_{ij} (\beta_j - \tilde{\beta}_j) + \frac{M}{2} (\beta_j - \tilde{\beta}_j)^2 \\ &\quad + p'_{\lambda_1}(|\tilde{\beta}_j|) |\beta_j| + \lambda_2 (D^\top D)_{jj} \beta_j^2 + 2\lambda_2 \sum_{l=1, l \neq j}^p (D^\top D)_{lj} \tilde{\beta}_l \beta_j. \end{aligned} \quad (5)$$

Since \hat{F} is a quadratic majorization function of F , we can get the new update by minimizing \hat{F} :

$$\hat{\beta}_j^{new} = \arg \min_{\beta_j} \hat{F}(\beta_j|\tilde{\beta}_0, \tilde{\beta}) = \frac{S(z, p'_{\lambda_1}(|\tilde{\beta}_j|))}{M + 2\lambda_2 (D^\top D)_{jj}}, \quad (6)$$

where $S(z, t) = (|z| - t)_+ \text{sign}(z)$, $z = M\tilde{\beta}_j - \frac{1}{n} \sum_{i=1}^n \ell'(r_i) y_i x_{ij} - 2\lambda_2 \sum_{l=1, l \neq j}^p (D^\top D)_{lj} \tilde{\beta}_l$.

Likewise, we can update the intercept by minimizing

$$\hat{F}(\beta_0|\tilde{\beta}_0, \tilde{\beta}) = \frac{1}{n} \sum_{i=1}^n \ell(r_i) + \frac{1}{n} \sum_{i=1}^n \ell'(r_i) y_i (\beta_0 - \tilde{\beta}_0) + \frac{M}{2} (\beta_0 - \tilde{\beta}_0)^2. \quad (7)$$

Then the intercept is updated by

$$\hat{\beta}_0^{new} = \arg \min_{\beta_0} Q(\beta_0|\tilde{\beta}_0, \tilde{\beta}) = \tilde{\beta}_0 - \frac{\sum_{i=1}^n \ell'(r_i) y_i}{Mn}. \quad (8)$$

Then we can iterate (4)–(8) until convergence.

Remark 2.2: In this paper, we specify the initial value by L_1 penalized SVM following by [27], and it leads to a satisfactory result.

Remark 2.3: The above algorithm satisfies the majorization–minimization (MM) principle [5,13,14], and the MM principle ensures the descent property of the GCD algorithm. The proof is similar to [21,22] and we omit here.

2.3. Asymptotic properties

In this subsection, we establish the theory of the local oracle property for the structured sparse SVM, namely the oracle estimator is one of the local minimizers of (2).

Since β_0 does not affect variable selection, we make $\beta_0 = 0$ for the convenience of expression without loss of generality in this section. Let $\beta^* = (\beta_1^*, \beta_2^*, \dots, \beta_p^*)^\top$ denote the true parameter value, which is defined as the minimizer of the population loss: $\beta^* = \arg \min_{\beta} L(\beta) = \arg \min_{\beta} \mathbb{E}\{\ell(yx^\top \beta)\}$.

We use p_n in this section to denote the number of features. Let $\mathcal{A} = \{j : \beta_j^* \neq 0, 1 \leq j \leq p_n\}$ be the index set of the non-zero coefficients, and $\mathcal{A}^c = \{j : \beta_j^* = 0, 1 \leq j \leq p_n\}$ be the index set of the zero coefficients. $q_n = |\mathcal{A}|$ is the cardinality of set \mathcal{A} . $D_{\mathcal{A}}$ is a submatrix formed by D removing the column corresponding to the

element in the \mathcal{A}^c . $\boldsymbol{\beta}_{\mathcal{A}^c} = (\beta_j)_{j \in \mathcal{A}^c}$ is a vector composed of the components of $\boldsymbol{\beta}$ corresponding to the elements in \mathcal{A}^c . Then the oracle estimate $\hat{\boldsymbol{\beta}}$ is defined as $\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}_{\mathcal{A}^c} = 0} \{ \frac{1}{n} \sum_{i=1}^n \ell(y_i \mathbf{x}_{i,\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}) + \lambda_{2n} \boldsymbol{\beta}_{\mathcal{A}}^\top D_{\mathcal{A}}^\top D_{\mathcal{A}} \boldsymbol{\beta}_{\mathcal{A}} \}$.

Theorem 2.1: *Assume that the conditions 1–5 listed in the Appendix hold. Let $B_n(\lambda_{1n}, \lambda_{2n})$ be the set of local minimizers of the objective function*

$$Q_n(\boldsymbol{\beta}) = L_n(\boldsymbol{\beta}) + \sum_{j=1}^p p_{\lambda_{1n}}(|\beta_j|) + \lambda_{2n} \boldsymbol{\beta}^\top D^\top D \boldsymbol{\beta}$$

with regularization parameter $\lambda_{1n}, \lambda_{2n}$. The oracle estimator $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}_{\mathcal{A}}^\top, \mathbf{0}^\top)^\top$ satisfies

$$\Pr \{ \hat{\boldsymbol{\beta}} \in B_n(\lambda_{1n}, \lambda_{2n}) \} \rightarrow 1$$

as $n \rightarrow \infty$, if $q_n n^{-1/2} \log p_n \log n = o(\lambda_{1n})$, $\lambda_{2n} q_n^{1/2} n^{-1/2} = o(\lambda_{1n})$, and $\lambda_{1n} = o(n^{-(1-c_3)/2})$.

From Theorem 2.1, we can see that if we take $\lambda_{1n} = n^{-1/2+\tau}$ for some $c_1 < \tau < c_3/2$, then the oracle property holds even for $p = o\{\exp(n^{(\tau-c_1)/2})\}$. Thus, even when the number of covariates grows exponentially with the sample size, the local oracle property still holds for the structured sparse SVM.

3. Simulations

In this section, numerical experiments are conducted to study the performance of our proposed method. We use Spline-penalty-HSVM, where penalty includes SCAD and MCP, to represent our proposed method (i.e. Spline-SCAD-HSVM and Spline-MCP-HSVM). To investigate the performance, we compared performances of the proposed method with other alternatives without considering structured sparsity: SCAD-HSVM and MCP-HSVM.

Three data generation processes are considered in this paper. We set $n = 100$ and $p = 1000$. In Example 3.1, the non-zero coefficients of the variables are completely smooth by position. Partial non-zero coefficients are smooth in Example 3.2. For Example 3.3, the non-zero coefficients are not smooth. Within each example, our simulated data consist of a training set and a testing set. Models are fitted on training data only, and the testing set with sample size 500 is used to show the predictions of each method. The optimal regularization parameters λ_1 and λ_2 are selected on a 15-by-20 meshgrid through a 5-fold cross validation. Possible values of λ_2 are from $[0.1, 0.2, \dots, 1.9, 2]$. For each fixed λ_2 , we compute the solutions for a fine grid of λ_1 s. Following [21], we start with $\lambda_{1(\max)}$ which is the smallest λ_1 to set all β_j to be zero, and set $\lambda_{1(\min)} = 0.01\lambda_{1(\max)}$. Between $\lambda_{1(\min)}$ and $\lambda_{1(\max)}$, 15 points are placed uniformly in the log-scale. Then we select the optimal regularization parameters that achieve the maximum of the classification accuracy rate. Here are the details of the three scenarios.

Example 3.1: Consider $\mathbf{x} \sim N(0, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = (0.5^{|i-j|})_{p \times p}$, $\beta_j = j/40$ for $j = 1, 2, \dots, 20$; $\beta_j = 1 - j/40$ for $j = 21, \dots, 40$; $\beta_j = -\sin(\pi j/40)$ for $j = 81, \dots, 120$; $\beta_j = 0.5$

$(1 - \cos(\pi j/20))$, for $j = 161, \dots, 200$; and $\beta_j = 0$ for the otherwise. $\Pr(y = -1) = 1/(1 + \exp(-\mathbf{x}^\top \boldsymbol{\beta}))$, and $\Pr(y = 1) = 1/(1 + \exp(\mathbf{x}^\top \boldsymbol{\beta}))$. The Bayes rule is $\text{sgn}(\mathbf{x}^\top \boldsymbol{\beta})$ with Bayes error 5.2%.

Example 3.2: The setting of \mathbf{x} is the same as in Example 3.1. β_j takes the same value as in Example 1 for $j = 1, 2, \dots, 120$; $\beta_j \sim \text{Uniform}(-0.5, 0.5)$, for $j = 160, \dots, 200$; and $\beta_j = 0$ for the otherwise. $\Pr(y = 1|\mathbf{x}) = \Phi(\mathbf{x}^\top \boldsymbol{\beta})$, where $\Phi(\cdot)$ is the distribution function of standard normal distribution. The Bayes rule is $\text{sgn}(\mathbf{x}^\top \boldsymbol{\beta})$ with Bayes error 9.1%.

Example 3.3: The generations of \mathbf{x} and \mathbf{y} are the same as in Example 3.1. $\beta_j \sim \text{Uniform}(0, 1)$, for $j = 1, 2, \dots, 40$; and $\beta_j = 0$ for the otherwise. The Bayes rule is $\text{sgn}(\mathbf{x}^\top \boldsymbol{\beta})$ with Bayes error 8.3%.

The performance of different methods will be examined in two aspects: classification prediction and feature selection. In the evaluation of classification prediction, the classification accuracy rate (ACC), area under curve (AUC), true positive rate (TPR) and false positive rate (FPR) are adopted. As for feature selection, we compare TPR and FPR for different methods.

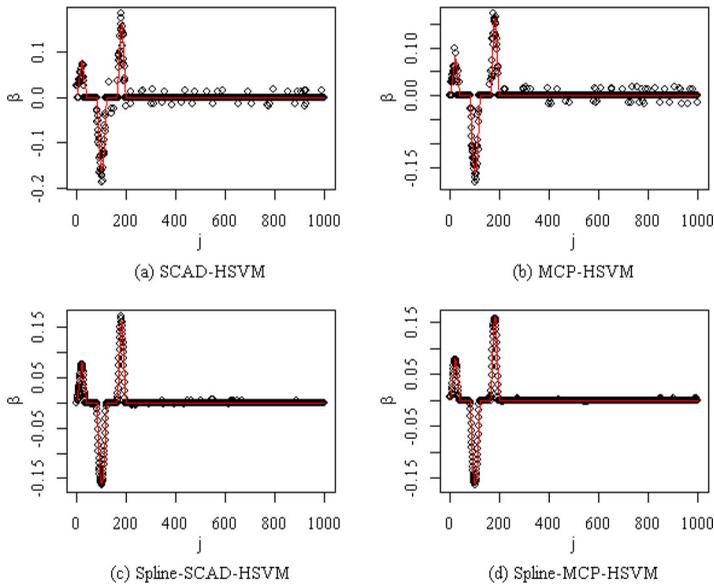
The results of the simulations are shown in Table 1. When the features vary smoothly, the SVM model with the spline penalty is significantly better in classification prediction and variable selection than the SVM model without the spline penalty (Example 3.1 and 3.2). The proposed method performs better, especially when non-zero coefficients of the variables are completely smooth by position in terms of classification prediction. In Figure 2, we present the estimation results for the correlated features in Example 3.1, by four different methods. From the figure, we can conclude that both the Spline-SCAD-HSVM and Spline-MCP-HSVM give a good estimate of the coefficients, while SCAD-HSVM and MCP-HSVM do not clean out the noisy signals very well. The improvement is not surprising since Spline-SCAD-HSVM and Spline-MCP-HSVM can capture the smoothing changes in coefficients. When the non-zero coefficients are not smooth, which means structural information described in this paper does not exist, the performances of both methods are similar (Example 3.3). This observation indicates that the proposed models are also applicable even when the features do not vary smoothly.

4. Real data analysis

In this section, we apply our methods to a dataset of Ovarian Dataset 8-7-02. The dataset is provided by the US Food and Drug Administration (FDA) and the National Cancer Institute (NCI), which can be downloaded and accessed at <http://home.ccr.cancer.gov/>. The data were collected as serum samples from normal and cancer patients, and the mass spectrometry technique was combined with the WCX2 protein chip and SELDI-TOF. The sample set included 91 controls and 162 ovarian cancers, which were not randomized. Each mass spectrometer sample contains a 15154-dimensional mass-to-charge ratio (m/z)/intensity characteristic. As mentioned in Section 1, the features are ordered in a meaningful way. Following the original researchers, we ignored m/z -sites below 100, where chemical artifacts can occur [16].

Table 1. The simulation results obtained from 100 Monte Carlo repetitions (with standard errors in parentheses).

Method	Classification prediction				Variable selection	
	ACC	AUC	TPR	FPR	TPR	FPR
Example 3.1						
SCAD-HSVM	0.782 (0.021)	0.874 (0.018)	0.769 (0.046)	0.205 (0.040)	0.732 (0.038)	0.309 (0.029)
MCP-HSVM	0.775 (0.025)	0.867 (0.024)	0.789 (0.031)	0.237 (0.047)	0.761 (0.028)	0.304 (0.036)
Spline-SCAD-HSVM	0.922 (0.022)	0.983 (0.011)	0.911 (0.034)	0.067 (0.030)	0.874 (0.019)	0.254 (0.021)
Spline-MCP-HSVM	0.932 (0.010)	0.987 (0.010)	0.920 (0.033)	0.055 (0.028)	0.868 (0.024)	0.143 (0.019)
Example 3.2						
SCAD-HSVM	0.756 (0.011)	0.845 (0.008)	0.723 (0.010)	0.258 (0.008)	0.611 (0.019)	0.253 (0.008)
MCP-HSVM	0.738 (0.015)	0.834 (0.007)	0.715 (0.009)	0.263 (0.008)	0.606 (0.020)	0.265 (0.009)
Spline-SCAD-HSVM	0.798 (0.013)	0.853 (0.011)	0.729 (0.009)	0.259 (0.008)	0.799 (0.019)	0.223 (0.010)
Spline-MCP-HSVM	0.804 (0.012)	0.854 (0.012)	0.731 (0.015)	0.257 (0.008)	0.799 (0.020)	0.225 (0.011)
Example 3.3						
SCAD-HSVM	0.788 (0.009)	0.880 (0.012)	0.799 (0.010)	0.244 (0.002)	0.782 (0.018)	0.215 (0.005)
MCP-HSVM	0.788 (0.009)	0.880 (0.012)	0.799 (0.010)	0.244 (0.002)	0.781 (0.020)	0.215 (0.005)
Spline-SCAD-HSVM	0.801 (0.006)	0.878 (0.011)	0.791 (0.010)	0.290 (0.003)	0.790 (0.019)	0.229 (0.011)
Spline-MCP-HSVM	0.808 (0.006)	0.878 (0.011)	0.795 (0.010)	0.281 (0.003)	0.790 (0.015)	0.229 (0.012)

**Figure 2.** The average estimation results for the correlated features in Example 1 of 100 Monte Carlo repetitions, by four different methods: SCAD-HSVM, MCP-HSVM, Spline-SCAD-HSVM and Spline-MCP-HSVM. The solid line curve is the true β , and the scatter dot represents the estimation for each method.

We randomly choose 173 samples from data as the training set, and the remaining 80 samples are used as the testing set. Four methods with Huberized hinge loss, i.e. SCAD-HSVM, MCP-HSVM, Spline-SCAD-HSVM, and Spline-MCP-HSVM, are fitted using the training set. Additional results with other losses are provided in the supplemental materials. Tuning parameters are chosen by 5-fold cross validation base on the training set. We select the optimal regularization parameters that achieve the maximum of the classification accuracy rate among grid points using two-dimensional grid search. We run the sample-splitting method 100 times, and the results are summarized in Table 2. We can see

Table 2. Results of 100 random splits of the ovarian cancer dataset (with standard errors in parentheses).

Method	ACC	AUC	TPR	FPR
SCAD-HSVM	0.919 (0.025)	0.968 (0.010)	0.930 (0.021)	0.061 (0.012)
MCP-HSVM	0.921 (0.026)	0.971 (0.008)	0.932 (0.022)	0.060 (0.015)
Spline-SCAD-HSVM	0.947 (0.018)	0.989 (0.004)	0.972 (0.009)	0.043 (0.011)
Spline-MCP-HSVM	0.947 (0.019)	0.992 (0.003)	0.975 (0.009)	0.041 (0.010)

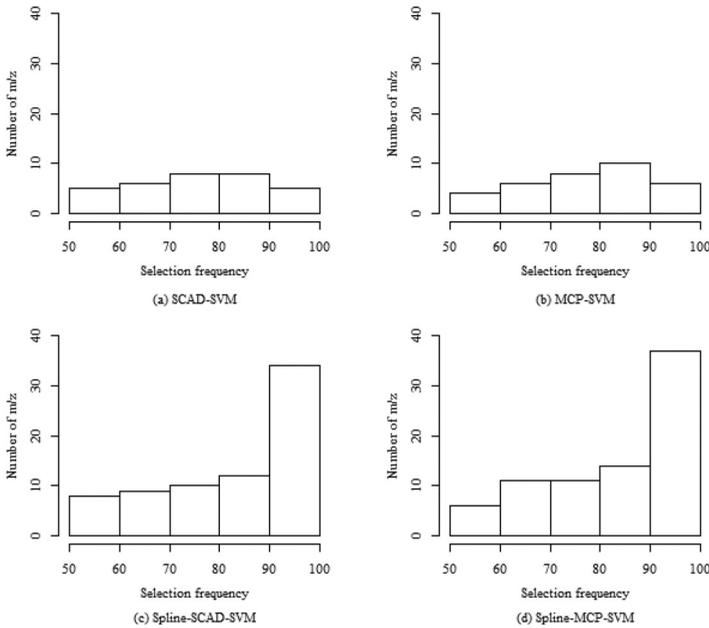


Figure 3. The figure shows the number of selected proteins versus the selection frequency of four different methods: (a) SCAD-HSVM, (b) MCP-HSVM, (c) Spline-SCAD-HSVM, and (d) Spline-MCP-HSVM.

that the performance of the Spline-SCAD-HSVM and Spline-MCP-HSVM is slightly better than the SCAD-HSVM and MCP-HSVM. The ACC, AUC and TPR are slightly higher when we consider the model that explicitly incorporates the special structures among the features.

To complement the estimation and identification analysis, we also evaluate the stability of analysis by computing the observed occurrence index (OOI). For each feature identified using the training data, we compute its probability of being identified out of the 100 resamplings; this probability has been referred to as the OOI. The median OOI values of SCAD-HSVM, MCP-HSVM, Spline-SCAD-HSVM, and Spline-MCP-HSVM are 0.736, 0.739, 0.857, and 0.862, respectively. Figure 3 shows the number of selected proteins versus the selection frequency of four different methods out of the 100 random splits. We can conclude that the OOI value of the model with the spline item is significantly higher, which indicates that Spline-SCAD-HSVM and Spline-MCP-HSVM are more stable than SCAD-HSVM and MCP-HSVM.

5. Discussion

In this article, we consider a high-dimensional data classification problem, where the features are ordered in some meaningful way. When the coefficients are sparse and change smoothly, we propose a structured sparse SVM, which combines the non-convex penalty and cubic spline estimation procedure (i.e. penalizing second-order derivatives of the coefficients) to the SVM, and proved that it satisfies the local oracle property under some conditions. The simulation and empirical results show that the proposed method has a higher accuracy of classification and prediction compared with the existing methods.

In the future, we want to work on more complex data. For example, when the high-dimensional data variables have group structure information and the intra-group features are ordered in some meaningful way. Moreover, our approach could also be extended to the framework of semisupervised learning and multi-class classification.

Acknowledgments

We would like to thank the editor, associate editor and two reviewers for their constructive comments that have led to a significant improvement of the manuscript.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This study was supported by the National Natural Science Foundation of China [grant number 11971404], [grant number 71471152], Humanity and Social Science Youth Foundation of Ministry of Education of China [grant number 19YJC910010], [grant number 20YJC910004], the 111 Project (B13028) and Fundamental Research Funds for the Central Universities [grant number 20720181003].

References

- [1] P.S. Bradley and O.L. Mangasarian, *Feature selection via concave minimization and support vector machines*, ICML 98 (1998), pp. 82–90.
- [2] P. Bühlmann and S. Van De Geer, *Statistics for High-dimensional Data: Methods, Theory and Applications*, Springer Science, New York, 2011.
- [3] W.H. Chan, M.S. Mohamad, S. Deris, J.M. Corchado, and S. Kasim, *An improved gSVM-SCADL2 with firefly algorithm for identification of informative genes and pathways*, Int. J. Bioinf. Res. Appl. 12 (2016), pp. 72–93.
- [4] W.J. Chen and Y.J. Tian, *ℓ_p -norm proximal support vector machine and its applications*, Procedia Computer Sci. 1 (2010), pp. 2417–2423.
- [5] J. De Leeuw and W.J. Heiser, *Convergence of Correction Matrix Algorithms for Multidimensional Scaling*, in *Geometric Representations of Relational Data*, Vol. 36. Mathesis Press, Ann Arbor, 1977, pp. 735–752.
- [6] J. Fan and Y. Fan, *High dimensional classification using features annealed independence rules*, Ann. Stat. 36 (2008), pp. 2605–2637.
- [7] J. Fan and R. Li, *Variable selection via nonconcave penalized likelihood and its oracle properties*, J. Am. Stat. Assoc. 96 (2001), pp. 1348–1360.

- [8] J. Fan and H. Peng, *Nonconcave penalized likelihood with a diverging number of parameters*, *Ann. Stat.* 32 (2004), pp. 928–961.
- [9] J. Guo, J. Hu, B.Y. Jing, and Z. Zhang, *Spline-lasso in high-dimensional linear regression*, *J. Am. Stat. Assoc.* 111 (2016), pp. 288–297.
- [10] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, *Gene selection for cancer classification using support vector machines*, *Mach. Learn.* 46 (2002), pp. 389–422.
- [11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, 2001.
- [12] M. Hebiri and S.V.D. Geer, *The smooth-lasso and other $\ell_1 + \ell_2$ -penalized methods*, *Electron. J. Stat.* 5 (2011), pp. 1184–1226.
- [13] D.R. Hunter and K. Lange, *A tutorial on MM algorithms*, *Am. Stat.* 58 (2004), pp. 30–37.
- [14] K. Lange, D.R. Hunter, and I. Yang, *Optimization transfer using surrogate objective functions*, *J. Comput. Graph. Stat.* 9 (2000), pp. 1–20.
- [15] O.L. Mangasarian, *A finite newton method for classification*, *Optim. Methods. Softw.* 17 (2002), pp. 913–929.
- [16] E.F. Petricoin III, A.M. Ardekani, B.A. Hitt, P.J. Levine, V.A. Fusaro, S.M. Steinberg, G.B. Mills, C. Simone, D.A. Fishman, E.C. Kohn, and L.A. Liotta, *Use of proteomic patterns in serum to identify ovarian cancer*, *The Lancet.* 359 (2002), pp. 572–577.
- [17] S. Rosset and J. Zhu, *Piecewise linear regularized solution paths*, *Ann. Stat.* 35 (2007), pp. 1012–1030.
- [18] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, *Sparsity and smoothness via the fused lasso*, *J. R. Stat. Soc. Ser. B(Methodological)* 67 (2005), pp. 91–108.
- [19] V. Vapnik, *The Nature of Statistical Learning Theory*, New York, Springer, 1995.
- [20] L. Wang, J. Zhu, and H. Zou, *Hybrid huberized support vector machines for microarray classification and gene selection*, *Bioinformatics.* 24 (2008), pp. 412–419.
- [21] Y. Yang and H. Zou, *An efficient algorithm for computing the HHSVM and its generalizations*, *J. Comput. Graph. Stat.* 22 (2013), pp. 396–415.
- [22] Y. Yang and H. Zou, *A fast unified algorithm for solving group-lasso penalize learning problems*, *Stat. Comput.* 25 (2015), pp. 1129–1141.
- [23] M. Yuan, *High dimensional inverse covariance matrix estimation via linear programming*, *J. Mach. Learn. Res.* 11 (2010), pp. 2261–2286.
- [24] C.H. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, *Ann. Stat.* 38 (2010), pp. 894–942.
- [25] H.H. Zhang, J. Ahn, X. Lin, and C. Park, *Gene selection using support vector machines with non-convex penalty*, *Bioinformatics.* 22 (2005), pp. 88–95.
- [26] C.H. Zhang and J. Huang, *The sparsity and bias of the lasso selection in high-dimensional linear regression*, *Ann. Stat.* 36 (2008), pp. 1567–1594.
- [27] X. Zhang, Y. Wu, L. Wang, and R. Li, *Variable selection for support vector machines in moderately high dimensions*, *J. R. Stat. Soc. Ser. B (Methodological)* 78 (2016), pp. 53–76.
- [28] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, *1-norm support vector machines*, *Adv. Neural. Inf. Process. Syst.* 16 (2004), pp. 49–56.
- [29] H. Zou and R. Li, *One-step sparse estimates in nonconcave penalized likelihood models*, *Ann. Stat.* 36 (2008), pp. 1509–1533.

Appendices

Appendix 1. Regularity conditions

To facilitate our technical proofs, we impose the following regularity conditions.

Condition A.1: The loss function $\ell(\cdot)$ is convex and it has a first order continuous derivative. There exist constants M_1 and M_2 , such that $|\ell'(t)| \leq M_1(|t| + 1)$, $|\partial(\ell'(t))| \leq M_2$, $\forall t$, where $\partial(\cdot)$ represents the subgradient.

Condition A.2: $q_n = O(n^{c_1})$, $0 \leq c_1 < 1/2$; $\lambda_{2n} \|D\boldsymbol{\beta}^*\| = O(n^{-c_2})$, $(1 - c_1)/2 < c_2 \leq 1/2$.

Condition A.3: The Hessian matrix $H(\boldsymbol{\beta}_{\mathcal{A}}) = \mathbb{E}[\nabla^2 \ell(\mathbf{y}\mathbf{x}_{\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}})]$ satisfies conditions

$$0 < M_3 < \lambda_{\min}\{H(\boldsymbol{\beta}_{\mathcal{A}}^*)\} \leq \lambda_{\max}\{H(\boldsymbol{\beta}_{\mathcal{A}}^*)\} < M_4 < \infty,$$

where $\mathbf{x}_{\mathcal{A}}$ is the matrix formed by \mathbf{x} removing the column corresponding to the element in the \mathcal{A}^c . where λ_{\min} and λ_{\max} denote the smallest and largest eigenvalue, respectively.

Condition A.4: There is a constant $M_5 > 0$ such that $\lambda_{\max}(n^{-1}\mathbf{x}_{\mathcal{A}}^\top \mathbf{x}_{\mathcal{A}}) \leq M_5$. It is further assumed that x_{ij} are sub-Gaussian random variables for $1 \leq i \leq n, j \in \mathcal{A}^c$.

Condition A.5 (Condition on the true model dimension): There exist positive constants c_3 and M_6 such that $1 - c_1 \leq c_3 \leq 1$ and $n^{(1-c_3)/2} \min_{j \in \mathcal{A}} |\beta_j^*| \geq M_6$.

Remark A.1: Condition 1 requires that the loss function be smooth and that the change is gentle, which is satisfactory for some common SVM loss functions, such as Huberized hinge loss function and square hinge loss function. Condition 2 states that the divergence rate of the number of non-zero coefficients cannot be faster than $n^{1/2}$, and the coefficient of the variable is slowly changing in position, which supports our introduction of spline penalty. Under Conditions 3, the Hessian matrix of the loss function is assumed to be positive definite, and its eigenvalues are uniformly bounded. The condition on the largest eigenvalues of the design matrix, which is assumed in Condition 4, is similar to that of [23,26,27]. Condition 5 simply states that the signals cannot decay too quickly.

Appendix 2. Some lemmas

The proof of Theorem 2.1 relies on the following lemmas.

Lemma A.1: Assume that Conditions 1-5 are satisfied. Then the oracle estimator satisfies $\|\hat{\boldsymbol{\beta}}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| = O_p(\sqrt{q_n/n})$.

Proof: Let $\alpha_n = \sqrt{q_n/n}$, and $Q_n(\boldsymbol{\beta}_{\mathcal{A}}) = \frac{1}{n} \sum_{i=1}^n \ell(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}) + \lambda_{2n} \boldsymbol{\beta}_{\mathcal{A}}^\top D_{\mathcal{A}}^\top D_{\mathcal{A}} \boldsymbol{\beta}_{\mathcal{A}}$, we want to show that for any given $\varepsilon > 0$, there exists a constant $C > 0$ such that

$$\Pr \left\{ \inf_{\|\mathbf{u}\|=C} Q_n(\boldsymbol{\beta}_{\mathcal{A}}^* + \alpha_n \mathbf{u}) > Q_n(\boldsymbol{\beta}_{\mathcal{A}}^*) \right\} \geq 1 - \varepsilon. \quad (\text{A.1})$$

This implies that there exists a local minimum in the ball $\{\boldsymbol{\beta}_{\mathcal{A}}^* + \alpha_n \mathbf{u} : \|\mathbf{u}\| \leq C\}$ with probability at least $1 - \varepsilon$. Hence, there exists a local minimizer such that $\|\hat{\boldsymbol{\beta}}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| = O_p(\alpha_n)$.

Let

$$\begin{aligned}\Lambda_n(\mathbf{u}) &= Q_n(\boldsymbol{\beta}_A^* + \alpha_n \mathbf{u}) - Q_n(\boldsymbol{\beta}_A^*) \\ &= \frac{1}{n} \sum_{i=1}^n \left[\ell\{y_i \mathbf{x}_{iA}^\top (\boldsymbol{\beta}_A^* + \alpha_n \mathbf{u})\} - \ell(y_i \mathbf{x}_{iA}^\top \boldsymbol{\beta}_A^*) \right] \\ &\quad + \lambda_{2n} \{(\boldsymbol{\beta}_A^* + \alpha_n \mathbf{u})^\top D_A^\top D_A (\boldsymbol{\beta}_A^* + \alpha_n \mathbf{u}) - \boldsymbol{\beta}_A^{*\top} D_A^\top D_A \boldsymbol{\beta}_A^*\}.\end{aligned}$$

By applying Taylor series expansion around $\boldsymbol{\beta}^*$, we have

$$\begin{aligned}\Lambda_n(\mathbf{u}) &= \frac{1}{n} \sum_{i=1}^n \left[\alpha_n \nabla^\top \{\ell(y_i \mathbf{x}_{iA}^\top \boldsymbol{\beta}_A^*)\} \mathbf{u} + \frac{\alpha_n^2}{2} \mathbf{u}^\top \nabla^2 \{\ell(y_i \mathbf{x}_{iA}^\top \tilde{\boldsymbol{\beta}}_A)\} \mathbf{u} \right] \\ &\quad + 2\lambda_{2n} \alpha_n \boldsymbol{\beta}_A^{*\top} D_A^\top D_A \mathbf{u} + \lambda_{2n} \alpha_n^2 \mathbf{u}^\top D_A^\top D_A \mathbf{u} \\ &\geq \frac{\alpha_n}{n} \sum_{i=1}^n \nabla^\top \{\ell(y_i \mathbf{x}_{iA}^\top \boldsymbol{\beta}_A^*)\} \mathbf{u} + \frac{\alpha_n^2 \mathbf{u}^\top}{2n} \sum_{i=1}^n \nabla^2 \{\ell(y_i \mathbf{x}_{iA}^\top \tilde{\boldsymbol{\beta}}_A)\} \mathbf{u} \\ &\quad + 2\lambda_{2n} \alpha_n \boldsymbol{\beta}_A^{*\top} D_A^\top D_A \mathbf{u} \\ &\stackrel{\wedge}{=} I_1 + I_2 + I_3,\end{aligned}\tag{A.2}$$

where $\tilde{\boldsymbol{\beta}}_A = \boldsymbol{\beta}_A^* + \alpha_n t \mathbf{u}$, $0 < t < 1$. By Conditions 1–3, we have

$$\begin{aligned}|I_1| &= \left| \frac{\alpha_n}{n} \sum_{i=1}^n \nabla^\top \{\ell(y_i \mathbf{x}_{iA}^\top \boldsymbol{\beta}_A^*)\} \mathbf{u} \right| \leq \alpha_n \left\| \frac{1}{n} \sum_{i=1}^n \nabla^\top \{\ell(y_i \mathbf{x}_{iA}^\top \boldsymbol{\beta}_A^*)\} \right\| \cdot \|\mathbf{u}\| \\ &= \alpha_n \cdot O_p(\sqrt{q_n/n}) \|\mathbf{u}\| = O_p(\alpha_n^2) \|\mathbf{u}\|.\end{aligned}$$

With Conditions 1 and 4, using Chebyshev inequality similarly as that in [8], we have when $q_n = O(n^{c_1})$, $0 \leq c_1 < 1/2$

$$\Pr \left\{ \left\| \frac{1}{n} \sum_{i=1}^n \nabla^2 \{\ell(y_i \mathbf{x}_{iA}^\top \boldsymbol{\beta}_A^*)\} - H(\boldsymbol{\beta}_A^*) \right\| \geq \varepsilon q_n^{-1} \right\} \leq \frac{q_n^2}{n \varepsilon^2} = o(1),$$

Thus

$$\left\| \frac{1}{n} \sum_{i=1}^n \nabla^2 \{\ell(y_i \mathbf{x}_{iA}^\top \boldsymbol{\beta}_A^*)\} - H(\boldsymbol{\beta}_A^*) \right\| = o_p(q_n^{-1}),$$

Then

$$I_2 = \frac{1}{2} \alpha_n^2 \mathbf{u}^\top H(\boldsymbol{\beta}_A^*) \mathbf{u} \{1 + o_p(1)\}.$$

By choosing a sufficiently large C , the second term I_2 dominates the first term I_1 uniformly in $\|\mathbf{u}\| = C$. By Cauchy–Schwarz inequality and Condition 2, we have

$$\begin{aligned}|I_3| &= |2\lambda_{2n} \alpha_n \boldsymbol{\beta}_A^{*\top} D_A^\top D_A \mathbf{u}| \leq 2\lambda_{2n} \alpha_n \|D_A \boldsymbol{\beta}_A^*\| \|D_A \mathbf{u}\| \\ &= 2\lambda_{2n} \alpha_n \|D \boldsymbol{\beta}^*\| \sqrt{\mathbf{u}^\top D_A^\top D_A \mathbf{u}} \\ &\leq 2\lambda_{2n} \alpha_n \|D \boldsymbol{\beta}^*\| \|\mathbf{u}\| \sqrt{\lambda_{\max}(D_A^\top D_A)} = o(\alpha_n^2) \|\mathbf{u}\|.\end{aligned}$$

This is also dominated by the second term of (A2). Hence, by choosing a sufficiently large C , (A1) holds. This completes the proof of the lemma. ■

Lemma A.2: Assume that Conditions 1–5 hold and that $q_n n^{-1/2} \log p_n \log n = o(\lambda_{1n})$. Then

$$\Pr \left\{ \max_{j \in \mathcal{A}^c} \left| \frac{1}{n} \sum_{i=1}^n y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*) \right| > \frac{\lambda_{1n}}{2} \right\} \rightarrow 0,$$

as $n \rightarrow \infty$.

Proof: Recall that $\mathbb{E}[n^{-1} \sum_{i=1}^n y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*)] = 0$, and the fact that $\max_i |x_{ij}| = O_p(\sqrt{\log n})$ for sub-Gaussian random variables. For some positive constants C , we have

$$|y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*)| \leq M_1 |x_{ij}| (|\mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*| + 1) \leq M_1 |x_{ij}| (\|\mathbf{x}_{i\mathcal{A}}\| \|\boldsymbol{\beta}_{\mathcal{A}}^*\| + 1) \leq C q_n \log n.$$

By Lemma 14.11 of [2], we have

$$\Pr \left\{ \left| \frac{1}{n} \sum_{i=1}^n y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*) \right| > \frac{\lambda_{1n}}{2} \right\} \leq 2 \exp \left\{ -\frac{n \lambda_{1n}^2}{8C^2 q_n^2 \log^2 n} \right\}.$$

Then

$$\begin{aligned} & \Pr \left\{ \max_{j \in \mathcal{A}^c} \left| \frac{1}{n} \sum_{i=1}^n y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*) \right| > \frac{\lambda_{1n}}{2} \right\} \\ &= \Pr \left[\bigcup_{j \in \mathcal{A}^c} \left\{ \left| \frac{1}{n} \sum_{i=1}^n y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*) \right| > \frac{\lambda_{1n}}{2} \right\} \right] \leq 2p_n \exp \left\{ -\frac{n \lambda_{1n}^2}{8C^2 q_n^2 \log^2 n} \right\} \\ &= 2 \exp \left[\log p_n \left\{ 1 - \frac{n \lambda_{1n}^2}{8C^2 q_n^2 \log p_n \log^2 n} \right\} \right] \rightarrow 0 \end{aligned}$$

as $n \rightarrow \infty$ by the fact that $q_n n^{-1/2} \log p_n \log n = o(\lambda_{1n})$. ■

Lemma A.3: Suppose that Conditions 1–5 hold, $q_n n^{-1/2} \log p_n \log n = o(\lambda_{1n})$, $\lambda_{2n} q_n^{1/2} n^{-1/2} = o(\lambda_{1n})$, and $\lambda_{1n} = o(n^{-(1-c_3)/2})$. For $j = 1, 2, \dots, p$, denote

$$s_j(\hat{\boldsymbol{\beta}}) = \frac{\partial [L_n(\hat{\boldsymbol{\beta}}) + \lambda_{2n} \boldsymbol{\beta}_{\mathcal{A}}^\top D_{\mathcal{A}}^\top D_{\mathcal{A}} \boldsymbol{\beta}_{\mathcal{A}}]}{\partial \beta_j}.$$

For the oracle estimator $\hat{\boldsymbol{\beta}}$ and $s_j(\hat{\boldsymbol{\beta}})$, with probability approaching 1, we have

$$s_j(\hat{\boldsymbol{\beta}}) = 0, |\hat{\beta}_j| \geq \left(a + \frac{1}{2}\right) \lambda_{1n}, \quad j \in \mathcal{A},$$

$$|s_j(\hat{\boldsymbol{\beta}})| \leq \lambda_{1n}, |\hat{\beta}_j| = 0, \quad j \in \mathcal{A}^c.$$

Proof: The objective function $L_n(\boldsymbol{\beta}) + \lambda_{2n} \boldsymbol{\beta}^\top D_{\mathcal{A}}^\top D_{\mathcal{A}} \boldsymbol{\beta}$ is convex derivative function. By the convex optimization theorem, we have $s_j(\hat{\boldsymbol{\beta}}) = 0, j \in \mathcal{A}$.

Note that $\min_{j \in \mathcal{A}} |\hat{\beta}_j| \geq \min_{j \in \mathcal{A}} |\beta_j^*| - \max_{j \in \mathcal{A}} |\hat{\beta}_j - \beta_j^*|$. Furthermore, we have $\min_{j \in \mathcal{A}} |\beta_j^*| \geq M_6 n^{-(1-c_3)/2}$ by Condition 5, and $\max_{j \in \mathcal{A}} |\hat{\beta}_j - \beta_j^*| \leq \|\hat{\boldsymbol{\beta}}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| = O_p(\sqrt{q_n/n}) = O_p(n^{-(1-c_1)/2}) = o_p(n^{-(1-c_3)/2})$. Then according to $\lambda_{1n} = o(n^{-(1-c_3)/2})$, we have

$$\Pr \left\{ |\hat{\beta}_j| \geq \left(a + \frac{1}{2}\right) \lambda_{1n} \right\} \rightarrow 1, \quad \text{for } j \in \mathcal{A}.$$

For $j \in \mathcal{A}^c$, we have

$$s_j(\hat{\boldsymbol{\beta}}) = \frac{1}{n} \sum_{i=1}^n y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}) + 2\lambda_{2n} \sum_{i=1}^{p_n} (D^\top D)_{ij} \hat{\beta}_i \quad (\text{A.3})$$

We observe that

$$\begin{aligned}
 & \Pr \left\{ \max_{j \in \mathcal{A}^c} \left| n^{-1} \sum_{i=1}^n y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}) \right| > \lambda_{1n} \right\} \\
 & \leq \Pr \left\{ \max_{j \in \mathcal{A}^c} \left| n^{-1} \sum_{i=1}^n y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*) \right| > \frac{\lambda_{1n}}{2} \right\} \\
 & \quad + \Pr \left\{ \max_{j \in \mathcal{A}^c} \left| n^{-1} \sum_{i=1}^n y_i x_{ij} [\ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}) - \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*)] \right| > \frac{\lambda_{1n}}{2} \right\}. \tag{A.4}
 \end{aligned}$$

By Lemma A.2, the first term of inequality (A.4) is $o_p(1)$. From Lemma A.1, the second term of inequality (A.4) is bounded by

$$\begin{aligned}
 & \Pr \left\{ \max_{j \in \mathcal{A}^c} \left| n^{-1} \sum_{i=1}^n y_i x_{ij} [\ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}) - \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*)] \right| > \frac{\lambda_{1n}}{2} \right\} \\
 & \leq \Pr \left\{ \max_{j \in \mathcal{A}^c} \sup_{\|\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| \leq C\sqrt{q_n/n}} \left| n^{-1} \sum_{i=1}^n y_i x_{ij} [\ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}) - \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*)] \right| > \frac{\lambda_{1n}}{2} \right\} \tag{A.5}
 \end{aligned}$$

Together with Conditions 1 and 4, we have

$$\begin{aligned}
 & \max_{j \in \mathcal{A}^c} \sup_{\|\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| \leq C\sqrt{q_n/n}} \left| n^{-1} \sum_{i=1}^n y_i x_{ij} [\ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}) - \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}^*)] \right| \\
 & \leq M_2 \sup_{\|\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| \leq C\sqrt{q_n/n}} \max_{ij} |x_{ij}| n^{-1} \sum_{i=1}^n \sqrt{(\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*)^\top \mathbf{x}_{i\mathcal{A}} \mathbf{x}_{i\mathcal{A}}^\top (\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*)} \\
 & \leq M_2 \sup_{\|\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| \leq C\sqrt{q_n/n}} \max_{ij} |x_{ij}| \sqrt{(\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*)^\top (n^{-1} \mathbf{x}_{\mathcal{A}}^\top \mathbf{x}_{\mathcal{A}}) (\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*)} \\
 & \leq M_2 \sup_{\|\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| \leq C\sqrt{q_n/n}} \max_{ij} |x_{ij}| \cdot \|\boldsymbol{\beta}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| \cdot \left[\sqrt{\lambda_{\max}(n^{-1} \mathbf{x}_{\mathcal{A}}^\top \mathbf{x}_{\mathcal{A}})} \right] \\
 & = O\{\sqrt{\log(p_n n)}\} \cdot \sqrt{q_n/n} = o(\lambda_{1n}), \tag{A.6}
 \end{aligned}$$

as $n \rightarrow \infty$ by the fact that $q_n n^{-1/2} \log p_n \log n = o(\lambda_{1n})$.

By (A.4)–(A.6), as $n \rightarrow \infty$, we have

$$\Pr \left\{ \max_{j \in \mathcal{A}^c} \left| n^{-1} \sum_{i=1}^n y_i x_{ij} \ell'(y_i \mathbf{x}_{i\mathcal{A}}^\top \hat{\boldsymbol{\beta}}_{\mathcal{A}}) \right| > \lambda_{1n} \right\} \rightarrow 0. \tag{A.7}$$

Then according to the nature of the matrix $D^\top D$,

$$\begin{aligned}
 2\lambda_{2n} \left| \sum_{i=1}^{p_n} (D^\top D)_{ij} \hat{\beta}_i \right| & \leq 2\lambda_{2n} \left| \sum_{i=1}^{p_n} (D^\top D)_{ij} (\hat{\beta}_i - \beta_i^*) \right| + 2\lambda_{2n} \left| \sum_{i=1}^{p_n} (D^\top D)_{ij} \beta_i^* \right| \\
 & \leq 2\sqrt{70}\lambda_{2n} \|\hat{\boldsymbol{\beta}}_{\mathcal{A}} - \boldsymbol{\beta}_{\mathcal{A}}^*\| + 2\sqrt{6}\lambda_{2n} \|D\boldsymbol{\beta}^*\| = o_p(\lambda_{1n}). \tag{A.8}
 \end{aligned}$$

By (A.3), (A.7) and (A.8), we have $|s_j(\hat{\boldsymbol{\beta}})| \leq \lambda_{1n}$ for $j \in \mathcal{A}^c$.

As the oracle estimate $\hat{\boldsymbol{\beta}}$ is defined as

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}_{\mathcal{A}^c} = 0} \left\{ \frac{1}{n} \sum_{i=1}^n \ell(y_i \mathbf{x}_{i,\mathcal{A}}^\top \boldsymbol{\beta}_{\mathcal{A}}) + \lambda_{2n} \boldsymbol{\beta}_{\mathcal{A}}^\top D_{\mathcal{A}}^\top D_{\mathcal{A}} \boldsymbol{\beta}_{\mathcal{A}} \right\},$$

$|\hat{\beta}_j| = 0$ for $j \in \mathcal{A}^c$ naturally. ■

Appendix 3. Proof of Theorem 2.1.

Proof: Let

$$Q_n(\boldsymbol{\beta}) = L_n(\boldsymbol{\beta}) + \sum_{j=1}^p p_{\lambda_{1n}}(|\beta_j|) + \lambda_{2n} \boldsymbol{\beta}^\top D^\top D \boldsymbol{\beta} \triangleq g(\boldsymbol{\beta}) - h(\boldsymbol{\beta}),$$

where

$$g(\boldsymbol{\beta}) = L_n(\boldsymbol{\beta}) + \lambda_{1n} \sum_{j=1}^p |\beta_j| + \lambda_{2n} \boldsymbol{\beta}^\top D^\top D \boldsymbol{\beta}, \quad h(\boldsymbol{\beta}) = \lambda_{1n} \sum_{j=1}^p |\beta_j| - \sum_{j=1}^p p_{\lambda_{1n}}(|\beta_j|).$$

By writing as $g(\boldsymbol{\beta}) - h(\boldsymbol{\beta})$, we need to show that $\hat{\boldsymbol{\beta}}$ is a local minimizer of $Q_n(\boldsymbol{\beta})$. Based on Lemma A3, the proof is similar to that of Theorem 3.2 in [27]. We omit the proof here. ■