Journal of Data Science 19 (1), 56–74
January 2021
DOI: 10.6339/20-JDS1000
Statistical Data Science

Sparse Learning with Non-convex Penalty in Multi-classification

Nan Li[1] and Hao Helen Zhang[2,*]

[1]*Department of Epidemiology and Cancer Control, St. Jude Children's Research Hospital, Memphis, Tennessee, U.S.A.*
[2]*Department of Mathematics, University of Arizona, Tucson, Arizona, U.S.A.*

Abstract

Multi-classification is commonly encountered in data science practice, and it has broad applications in many areas such as biology, medicine, and engineering. Variable selection in multiclass problems is much more challenging than in binary classification or regression problems. In addition to estimating multiple discriminant functions for separating different classes, we need to decide which variables are important for each individual discriminant function as well as for the whole set of functions. In this paper, we address the multi-classification variable selection problem by proposing a new form of penalty, supSCAD, which first groups all the coefficients of the same variable associated with all the discriminant functions altogether and then imposes the SCAD penalty on the supnorm of each group. We apply the new penalty to both soft and hard classification and develop two new procedures: the supSCAD multinomial logistic regression and the supSCAD multi-category support vector machine. Our theoretical results show that, with a proper choice of the tuning parameter, the supSCAD multinomial logistic regression can identify the underlying sparse model consistently and enjoys oracle properties even when the dimension of predictors goes to infinity. Based on the local linear and quadratic approximation to the non-concave SCAD and nonlinear multinomial log-likelihood function, we show that the new procedures can be implemented efficiently by solving a series of linear or quadratic programming problems. Performance of the new methods is illustrated by simulation studies and real data analysis of the Small Round Blue Cell Tumors and the Semeion Handwritten Digit data sets.

Keywords *logistic regression; SCAD; supnorm; SVM; variable selection*

1 Introduction

Multiclass classification is an important topic in statistical machine learning and has broad applications in practice such as handwritten zip code digit recognition and cancer classification based on DNA microarray data (Hastie et al., 2009). In practice, a large number of variables are usually collected but some of them are uninformative in prediction. For example, in biological or medical data sets, the overwhelming number of variables far exceeds the sample size, but the underlying model is sparse and depends only on a small subset of predictors. Including all the predictors in a classifier may lead to poor generalization performance and interpretability (Fan and Li, 2001). Therefore, it is essential to identify important variables in order to increase both classification accuracy and model interpretability.

This paper is motivated by precision medicine in cancer research where one goal is to extract important information from omics data, such as genomics, transcriptomics, metabolomics,

*Corresponding author. Email: hzhang@math.arizona.edu.

Received November, 2020; Accepted December, 2020

and proteomics and classify tumors into different cancer subtypes in order to provide optimal treatment. Microarray and RNA-Seq data are typically analyzed to identify genes with differential expressions across different subtypes of cancer. Since the number of genes is usually much larger than the sample size, it is critical to select "signature" genes which can characterize cancer subtypes and have strong prediction power. This work is motivated by the classification of small round blue cell tumors in childhood (Khan et al., 2001) using a small set of important genes. We propose and study a new class of learning methods for joint multiclass classification and variable selection. The new tools are computationally efficient and scalable, with high discrimination power and interpretability, and they have broad applications to cancer classification and many areas such as biology, medicine, and engineering. In the following, we first review the problem of multi-classification, with a focus on variable selection in classification, and then introduce the proposed methods.

In a K-class classification problem, we are given a training data set containing n pairs of observations $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{id})^T \in R^d$ is an input vector and the output $y_i \in \mathcal{A} = \{1, 2, \ldots, K\}$ indicates its class label. Our task is to learn a discriminant rule $\phi : R^d \to \mathcal{A}$ which can assign a class label for any new input $\mathbf{x}$. Assume that $(\mathbf{x}_i, y_i)$'s are independent and identically distributed $(i.i.d.)$ samples from an unknown distribution $P(\mathbf{x}, y)$, with the conditional probabilities $p_k(\mathbf{x}) = P(y = k|\mathbf{x}), k = 1, \ldots, K$. The prediction performance of ϕ is measured by its misclassification error rate $\mathrm{GE}(\phi) = E_{(\mathbf{x},y)}[I(y \neq \phi(\mathbf{x}))] = P(y \neq \phi(\mathbf{x}))$. The optimal classifier minimizing the misclassification error rate is the *Bayes classifier*, i.e.

$$\phi_B(\mathbf{x}) = \arg\min_{k=1,\ldots,K} \big[1 - p_k(\mathbf{x})\big] = \arg\max_{k=1,\ldots,K} p_k(\mathbf{x}).$$

Since $p_k(\mathbf{x})$'s are generally unknown, we need to construct a decision function vector $\hat{\mathbf{f}} = \{\hat{f}_1, \ldots, \hat{f}_K\}$ with each $\hat{f}_k$ representing strength of evidence that a data point $\mathbf{x}$ belongs to class k. The classifier induced by $\hat{\mathbf{f}}$ assigns the label to $\mathbf{x}$ using the largest $\hat{f}_k(\mathbf{x})$, i.e. $\hat{\phi}(\mathbf{x}) = \arg\max_{1 \leqslant k \leqslant K} \hat{f}_k(\mathbf{x})$.

In literature, there are two main types of classifiers: soft classifiers and hard classifiers, for constructing decision rules for classification problems. Soft classification rules approximate the Bayes rule by first estimating $\hat{p}_k(\mathbf{x})$ and then predicting the class label based on the maximum probability. Among them are linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and logistic regression (McCullagh and Nelder, 1989). In particular, logistic regression assumes that the form of posterior log-odds is linear in $\mathbf{x}$, i.e. $\log \frac{p_k(\mathbf{x})}{p_K(\mathbf{x})} = \beta_{k0} + \boldsymbol{\beta}_k^T \mathbf{x}$, $k = 1, \ldots, K-1$, leading to

$$\begin{aligned} p_k(\mathbf{x}) &= \frac{\exp(\beta_{k0} + \boldsymbol{\beta}_k^T \mathbf{x})}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \boldsymbol{\beta}_l^T \mathbf{x})}, \quad k = 1, \ldots, K-1, \\ p_K(\mathbf{x}) &= \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \boldsymbol{\beta}_l^T \mathbf{x})}. \end{aligned} \tag{1}$$

Logistic regression models can be fitted by maximizing the conditional log-likelihood of y given $\mathbf{x}$ using the training dataset, such as maximizing the binomial or multinomial log-likelihood.

On the other hand, hard classification rules directly target on the argmax function without estimating the conditional class probabilities, such as support vector machine (SVM, Vapnik (1995)). Various multiclass SVMs have been proposed in the literature, including Vapnik (1998), Weston et al. (1999), Crammer and Singer (2001), Lee et al. (2004), Liu and Shen (2006), and Liu and Yuan (2011). For example, Lee et al. (2004) proposed the multi-category SVM (MSVM) which solves

$$\begin{aligned} &\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)\big[\beta_{k0} + \beta_k^T \mathbf{x}_i + 1\big]_+ + \lambda \sum_{k=1}^{K}\sum_{j=1}^{d} \beta_{kj}^2 \\ &\text{subject to } \sum_{k=1}^{K} \beta_{k0} = 0, \text{ and } \sum_{k=1}^{K} \beta_{kj} = 0,\ j = 1, \ldots, d. \end{aligned} \tag{2}$$

where $(x)_+ = \max(0, x)$. The sum-to-zero constraint in (2) is enforced to eliminate redundancy of the parameters and guarantee the solution identifiability. Besides the hinge loss, the squared error loss can also be incorporated with SVM, such as the least squares support vector machine (LSSVM; Suykens and Vandewalle (1999)) and the proximal support vector machine (PSVM; Mangasarian and Wild (2001)). Tang and Zhang (2006) extended PSVM to the multiclass PSVM (MPSVM) by solving

$$\begin{aligned} &\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)\big[\beta_{k0} + \boldsymbol{\beta}_k^T \mathbf{x}_i + 1\big]^2 + \lambda \sum_{k=1}^{K}\sum_{j=0}^{d} \beta_{kj}^2 \\ &\text{subject to } \sum_{k=1}^{K} \beta_{k0} = 0, \text{ and } \sum_{k=1}^{K} \beta_{kj} = 0,\ j = 1, \ldots, d. \end{aligned} \tag{3}$$

In addition to producing the classification decision rules, MPSVM can also estimate the conditional class probabilities.

For high dimensional data, a variety of penalized methods have been proposed to predict outcomes and select variables at the same time. Modern penalized methods include LASSO (Tibshirani, 1996), adaptive LASSO (Zou, 2006), SCAD (Fan and Li, 2001), and MCP (Zhang et al., 2010). These methods can shrink small coefficients to exactly 0 and hence achieve sparsity in the solution. Some of these methods have been applied to binary classification, such as L_1 logistic regression (Hastie et al., 2009), L_1 SVM (Bradley and Mangasarian, 1998), and SCAD SVM (Zhang et al., 2006). In addition to selecting individual variables, various group selection methods have also been developed, including the group LASSO (Yuan and Lin, 2006), F_∞ SVM (Zou and Yuan, 2008), and the Composite Absolute Penalties (CAP) family (Zhao et al., 2009). Huang et al. (2012) provided a detailed review on group selection in regression settings, such as the group LASSO, the group SCAD, and the group MCP. The extension of L_1-type methods from binary classification to multiclass problems include the L_1 multinomial logistic regression, the L_1 MSVM (Wang and Shen, 2007), the grouped L_1 in multinomial logistic regression (Tutz et al., 2015), and the supnorm MSVM (Zhang et al., 2008).

In this paper, we propose a new form of penalty called the supSCAD penalty, based on the composite of supnorm and SCAD function, to achieve group sparsity for multiclass problems. What makes the supSCAD penalty attractive is its ability to remove noise covariates, i.e., those covariates not contributing to discriminating different classes. One motivating example is the multi-type cancer classification using genes. Typically, only a small subset of "important" genes are needed to classify cancer into different subtypes, and the rest of genes are either redundant or non-informative. The supSCAD penalty is designed to enforce group-wise parsimony in all the coefficients associated with one variable without directly penalizing individual coefficients, and therefore the estimated coefficients are less biased than other methods such as the group LASSO. The new penalty demonstrates competitive performance for both multinomial logistics regression and MSVM/MPSVM, and enjoys nice theoretical properties, even if the data dimension diverges. An efficient algorithm is developed by combining the difference convex algorithm (DCA; Wu and Liu (2009)) and the local linear approximation (Zou and Li, 2008).

The rest of the article is organized as follows. Section 2 introduces the supSCAD penalty and establishes theoretical properties of the supSCAD logistic regression. Sections 3 and 4 present the computational algorithms for the supSCAD logistic regression and MSVM/MPSVM. Sections 5 and 6 evaluate performance of the new methods via simulations and two real examples. Final discussions are in Section 7. Major technical derivations and proofs are delegated to the Appendix.

2 Methodology

2.1 The supSCAD Penalty

Consider a K-class problem with the input vector $\mathbf{x} \in R^d$ and the output $y \in \{1, \ldots, K\}$. For linear classification rules, there are $(d+1)$ coefficients associated with each decision function, including the intercept term. Altogether, all the coefficients associated with the K decision functions can be expressed as a $K \times (d+1)$ coefficient matrix. The jth column of the matrix, expressed as $\boldsymbol{\beta}_{(j)} = (\beta_{1j}, \ldots, \beta_{Kj})^T$, consists of K coefficients associated with x_j, where $j = 0, 1, \ldots, d$ and x_0 is the intercept. The kth row $\boldsymbol{\beta}_k = (\beta_{k0}, \beta_{k1}, \ldots, \beta_{kd})$ consists of $(d+1)$ coefficients characterizing the decision function f_k, where $k = 1, \ldots, K$. For the variable selection purpose, we treat the elements in $\boldsymbol{\beta}_{(j)}$ as a group. Define the supnorm of $\boldsymbol{\beta}_{(j)}$ as

$$\|\boldsymbol{\beta}_{(j)}\|_\infty = \max_{k=1,\ldots,K} |\beta_{kj}|, \tag{4}$$

where the importance of x_j is directly controlled by its largest absolute element. If $\|\boldsymbol{\beta}_{(j)}\|_\infty = 0$, then all the K coefficients associated with x_j are set to zero. Otherwise, if x_j is important with a positive supnorm, then no penalty is imposed on the remaining elements. The SCAD penalty (Fan and Li, 2001) is a symmetric and non-convex function to produce sparse solutions, with a continuous first-order derivative $J'_\lambda(\theta) = \lambda\{I(\theta \leqslant \lambda) + \frac{(a\lambda-\theta)_+}{(a-1)\lambda} I(\theta > \lambda)\}$ except at the origin. Our new penalty, called supSCAD, combines the SCAD and supnorm in the following form

$$J_\lambda(\|\boldsymbol{\zeta}\|_\infty) = \begin{cases} \lambda\|\boldsymbol{\zeta}\|_\infty & \text{if } \|\boldsymbol{\zeta}\|_\infty \leqslant \lambda, \\ -\dfrac{(\|\boldsymbol{\zeta}\|_\infty^2 - 2a\lambda\|\boldsymbol{\zeta}\|_\infty + \lambda^2)}{2(a-1)} & \text{if } \lambda < \|\boldsymbol{\zeta}\|_\infty \leqslant a\lambda, \\ \dfrac{(a+1)\lambda^2}{2} & \text{if } \|\boldsymbol{\zeta}\|_\infty > a\lambda. \end{cases} \tag{5}$$

where $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_K)^T$, $\|\boldsymbol{\zeta}\|_\infty = \max_{i=1,\ldots,K} |\zeta_i|$, $a > 2$, and $\lambda > 0$ is the tuning parameter. In contrast to some bi-level (group and individual level) variable selection methods such as the composite MCP, the proposed supSCAD penalty focuses on group selection solely. There is only one tuning parameter in the supSCAD penalty, which makes its implementation much easier with low computational cost even for high dimensional problems.

2.2 supSCAD Logistic Regression & Oracle Properties

We propose the supSCAD multinomial logistic regression by solving the penalized log-likelihood

$$\begin{aligned} &\min_{\boldsymbol{\beta}} -\left\{\sum_{i=1}^n \left[\sum_{k=1}^K I(y_i = k) \cdot \boldsymbol{\beta}_k^T \mathbf{x}_i - \log\left(\sum_{k=1}^K \exp\left(\boldsymbol{\beta}_l^T \mathbf{x}_i\right)\right)\right]\right\} + n \sum_{j=1}^d J_\lambda(\|\boldsymbol{\beta}_{(j)}\|_\infty) \\ &\text{subject to } \sum_{k=1}^K \beta_{kj} = 0, \quad j = 0, 1, \ldots, d, \end{aligned} \tag{6}$$

where the sum-to-zero constraint is required to assure the identifiability of the solution.

In the following, we study the large-sample properties of the supSCAD logistic regression estimator. In particular, we will establish its oracle properties even when the number of predictors d diverges to infinity with the sample size n. Theorem 1 shows that the supSCAD estimator is root-(n/d_n) consistent, where the subscript in d_n is used to emphasize its dependence on n. Theorem 2 includes two parts: the model selection consistency, and the asymptotic normality of the estimators with the variance as if the true model were known.

Assume that the training dataset consists of n $i.i.d.$ observations $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$ from the multinomial logistic model with the true parameters $\boldsymbol{\beta}^\star$, which consists of $s_n + 1$ non-zero columns and $d_n - s_n$ zero columns. Without losing generality, we arrange non-zero columns before zero columns in $\boldsymbol{\beta}^\star$ and denote them by $\boldsymbol{\beta}^\star_+$ and $\boldsymbol{\beta}^\star_z$.

$$\boldsymbol{\beta}^\star = \begin{pmatrix} \beta^\star_{10} & \beta^\star_{11} & \cdots & \beta^\star_{1d_n} \\ \vdots & \vdots & \ddots & \vdots \\ \beta^\star_{K,0} & \beta^\star_{K,1} & \cdots & \beta^\star_{K,d_n} \end{pmatrix} = \begin{pmatrix} \beta^\star_{10} & \beta^\star_{11} & \cdots & \beta^\star_{1,s_n} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \beta^\star_{K,0} & \beta^\star_{K,1} & \cdots & \beta^\star_{K,s_n} & 0 & \cdots & 0 \end{pmatrix} = (\boldsymbol{\beta}^\star_+, \boldsymbol{\beta}^\star_z).$$

Define the penalized log-likelihood function

$$R\big(\{\boldsymbol{\beta}_k\}_1^K\big) = -\mathcal{L}\big(\{\boldsymbol{\beta}_k\}_1^K\big) + n\sum_{j=1}^{d_n} J_\lambda(\|\boldsymbol{\beta}_{(j)}\|_\infty),$$

where $\mathcal{L}(\{\boldsymbol{\beta}_k\}_1^K) = \log\prod_{i=1}^n\prod_{k=1}^K y_{ik}p_k(\mathbf{x}_i) = \sum_{i=1}^n(\sum_{k=1}^K y_{ik}(\mathbf{x}_i^T\boldsymbol{\beta}_k) - \log(\sum_{l=1}^K e^{\mathbf{x}_i^T\boldsymbol{\beta}_l}))$. Due to the sum-to-zero constraints, we have the relationship $\beta_{Kj} = -\sum_{k=1}^{K-1}\beta_{kj}$ for $j = 1, \ldots, d_n$. Therefore, all the coefficients can be completely determined by the $K-1$ dimensional vectors $\boldsymbol{\beta}_{(j),-K} = (\beta_{1j}, \ldots, \beta_{K-1,j})^T$, $j = 1, \ldots, d_n$. Consequently, the $K \times (d_n + 1)$ coefficient matrix $\boldsymbol{\beta}$ can be reduced to a $(K-1) \times (d_n + 1)$ matrix, and correspondingly, the objective function can be reparametrized as $\tilde{\mathcal{L}}(\{\boldsymbol{\beta}_k\}_1^{K-1})$ and

$$\begin{aligned}\tilde{R}\big(\{\boldsymbol{\beta}_k\}_1^{K-1}\big) &= -\tilde{\mathcal{L}}\big(\{\boldsymbol{\beta}_k\}_1^{K-1}\big) \\ &\quad + n\sum_{j=1}^{d_n} J_\lambda\left(\max\left\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{(K-1),j}|, \left|\sum_{l=1}^{K-1}\beta_{lj}\right|\right\}\right).\end{aligned} \tag{7}$$

Using the reparametrized penalized multinomial logit model (7), we show that the proposed supSCAD estimator possesses oracle properties with a proper choice of tuning parameter.

With a slight abuse of notation, we reformulate the matrix $\boldsymbol{\beta}$ and $\boldsymbol{\beta}^\star$ as vectors

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_{(0),-K} \\ \boldsymbol{\beta}_{(1),-K} \\ \vdots \\ \boldsymbol{\beta}_{(d_n),-K} \end{pmatrix} = \begin{pmatrix} \beta_{10} \\ \vdots \\ \beta_{K-1,0} \\ \vdots \\ \beta_{1d_n} \\ \vdots \\ \beta_{K-1,d_n} \end{pmatrix}, \qquad \boldsymbol{\beta}^\star = \begin{pmatrix} \boldsymbol{\beta}^\star_{(0),-K} \\ \boldsymbol{\beta}^\star_{(1),-K} \\ \vdots \\ \boldsymbol{\beta}^\star_{(d_n),-K} \end{pmatrix} = \begin{pmatrix} \beta^\star_{10} \\ \vdots \\ \beta^\star_{K-1,0} \\ \vdots \\ \beta^\star_{1d_n} \\ \vdots \\ \beta^\star_{K-1,d_n} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta}^\star_+ \\ \boldsymbol{\beta}^\star_z \end{pmatrix}. \tag{8}$$

Denote $I(\boldsymbol{\beta}^\star)$ as the Fisher information matrix of $\tilde{\mathcal{L}}$ at $\boldsymbol{\beta}^\star$, and $I(\binom{\boldsymbol{\beta}^\star_+}{\mathbf{0}})$ the Fisher information matrix knowing $\boldsymbol{\beta}^\star_z = \mathbf{0}$. Let $\widehat{\boldsymbol{\beta}} = \binom{\widehat{\boldsymbol{\beta}}_+}{\widehat{\boldsymbol{\beta}}_z}$ be a local minimizer to (7), where $\widehat{\boldsymbol{\beta}}_+$ consists of the first $(K-1) \times (s_n+1)$ elements of $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\beta}}_z$ consists of the remaining last $(K-1) \times (d_n - s_n)$ elements.

Theorem 1 (Consistency). *Assume the regularity conditions in the Appendix hold. If* ${d_n}^4/n \to 0$ *as* $n \to \infty$, *then there exists a local minimizer* $\hat{\boldsymbol{\beta}}$ *such that* $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\| = O_p(\sqrt{d_n/n})$.

Lemma 1 (Sparsity). *Assume the regularity conditions in the Appendix hold. If* $\lambda_n \to 0$, $\sqrt{d_n/n}/\lambda_n \to 0$, *and* ${d_n}^4/n \to 0$ *as* $n \to \infty$, *then with probability tending to 1, for any given* $\boldsymbol{\beta}_+$ *satisfying* $||\boldsymbol{\beta}_+ - \boldsymbol{\beta}_+^\star|| = O_p(\sqrt{d_n/n})$ *and any constant* C,

$$\tilde{R}\left(\begin{pmatrix}\boldsymbol{\beta}_+\\ \mathbf{0}\end{pmatrix}\right) = \min_{||\boldsymbol{\beta}_z|| \leqslant C\sqrt{d_n/n}} \tilde{R}\left(\begin{pmatrix}\boldsymbol{\beta}_+\\ \boldsymbol{\beta}_z\end{pmatrix}\right).$$

Theorem 2 (Oracle). *Under the conditions of Theorem 1 and Lemma 1, with probability tending to 1, the* $\sqrt{d_n/n}$*-consistent local minimizer* $\begin{pmatrix}\hat{\boldsymbol{\beta}}_+\\ \hat{\boldsymbol{\beta}}_z\end{pmatrix}$ *in Theorem 1 satisfies:*

1. *(Sparsity):* $\hat{\boldsymbol{\beta}}_z = \mathbf{0}$.
2. *(Asymptotic normality):* $\sqrt{n}(\hat{\boldsymbol{\beta}}_+ - \boldsymbol{\beta}_+^\star) \to N(\mathbf{0}, I^{-1}\begin{pmatrix}\boldsymbol{\beta}_+^\star\\ \mathbf{0}\end{pmatrix})$.

All the regularity conditions and proofs are given in the Appendix.

3 Computational Algorithms

3.1 Quadratic Approximation for the Log-likelihood

We approximate the negative multinomial log-likelihood in (6) by its second-order Taylor expansion using Newton's method, which is essentially the iteratively reweighed least squares (IRLS). At each step, given the current parameter estimates $\tilde{\boldsymbol{\beta}}$, a quadratic approximation to the negative log-likelihood in its neighborhood is:

$$\begin{aligned}\mathcal{L}_n(\boldsymbol{\beta}) \approx Q(\boldsymbol{\beta}) =& \mathcal{L}_n(\tilde{\boldsymbol{\beta}}) + \dot{\mathcal{L}}_n(\tilde{\boldsymbol{\beta}})^T(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})\\ &+ \frac{1}{2}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^T\left\{\ddot{\mathcal{L}}_n(\tilde{\boldsymbol{\beta}})\right\}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})\end{aligned} \tag{9}$$

where $\dot{\mathcal{L}}_n(\cdot)$ and $\ddot{\mathcal{L}}_n(\cdot)$ are the first- and second-order derivatives of the negative log-likelihood function with respect to its parameters. We propose to solve the supSCAD logistic regression using two iterative loops. First, each outer loop computes the quadratic approximation Q at the current parameters $\tilde{\boldsymbol{\beta}}$. Second, each inner loop executes optimization iterations for the supSCAD penalized quadratic problem.

1. **OUTER LOOP:** Compute the quadratic approximation Q at the current estimates.
2. **INNER LOOP:** Solve the supSCAD problem based on the updated Q.

3.2 Optimization Algorithms for supSCAD Logistic Regression

Based on the ideas in the above section, we convert the supSCAD logistic regression optimization to a series of quadratic programming problems. To handle the non-convex and non-differentiable supSCAD penalty, we adopt the difference convex algorithm (DCA; Le Thi Hoai and Tao (1997)) and the local linear approximation (LLA; Zou and Li (2008)) techniques. These algorithms are computationally efficient and assure that the supSCAD can be solved within polynomial time.

DCA Optimization Consider a non-convex minimization problem of

$$\min C(\theta) = C_{vex,1}(\theta) - C_{vex,2}(\theta)$$

where $C_{vex,1}(\theta)$ and $C_{vex,2}(\theta)$ are both convex functions. The DCA iteratively solves $\min C_{vex,1}(\theta) - C'_{vex,2}(\tilde{\theta})(\theta - \tilde{\theta})$ with a current solution $\tilde{\theta}$, which is shown to converge to a local minimum in finite steps. Following Wu and Liu (2009), we write the first-order derivative of the SCAD penalty function on $(0, +\infty)$ as the difference of two convex functions, $J_\lambda(\theta) = J_{\lambda,1}(\theta) - J_{\lambda,2}(\theta)$, where $J'_{\lambda,1}(\theta) = \lambda$ and $J'_{\lambda,2}(\theta) = \lambda(1 - \frac{(a\lambda-\theta)_+}{(a-1)\lambda})I(\theta > \lambda)$, and then apply the DCA.

LLA Optimization Zou and Li (2008) suggested LLA to the SCAD penalty and showed its ascent property for maximization and convergence (Lange et al., 2000). In particular, $J_\lambda(|\theta_j|) \approx J_\lambda(|\theta_j^{(0)}|) + J'_\lambda(|\theta_j^{(0)}|)(|\theta_j| - |\theta_j^{(0)}|)$ for $\theta_j \approx \theta_j^{(0)}$.

By applying DCA and LLA to the supSCAD penalty, we propose to iteratively solve the following optimization problem at each step.

$$\begin{aligned}
\boldsymbol{\beta}^{(t+1)} &= \arg\min_{\boldsymbol{\beta}} \left(Q(\boldsymbol{\beta}) + \sum_{j=1}^{d} \lambda \|\boldsymbol{\beta}_{(j)}\|_\infty - \sum_{j=1}^{d} J'_{\lambda,2}\big(\|\boldsymbol{\beta}_{(j)}^{(t)}\|_\infty\big)\big(\|\boldsymbol{\beta}\|_\infty - \|\boldsymbol{\beta}^{(t)}\|_\infty\big) \right) \\
&= \arg\min_{\boldsymbol{\beta}} \left(Q(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{d} \|\boldsymbol{\beta}_{(j)}\|_\infty - \sum_{j=1}^{d} J'_{\lambda,2}\big(\|\boldsymbol{\beta}_{(j)}^{(t)}\|_\infty\big) \cdot \|\boldsymbol{\beta}\|_\infty \right) \\
&= \arg\min_{\boldsymbol{\beta}} \left(Q(\boldsymbol{\beta}) + \sum_{j=1}^{d} J'_\lambda\big(\|\boldsymbol{\beta}_{(j)}^{(t)}\|_\infty\big) \cdot \|\boldsymbol{\beta}_{(j)}\|_\infty \right) \qquad (10)
\end{aligned}$$

3.3 Implementation Issues

Choice of Initial Points Since the supSCAD penalty is non-convex and both DCA and LLA are local algorithms, the proposed algorithm is not guaranteed to produce a global minimum in general. Therefore, choosing the initial point is critical to the quality of the solution and its performance. Denote the sample size of each class by n_k for $k = 1, \ldots, K$, and the input vector dimension by d. If $n_k \gg d$ and the standard maximum likelihood converges well, then the MLE would be appropriate for initialization. If $n_k \approx d$ or $n_k < d$, then we can use the origin or the ridge estimator as the initial point.

The Stopping Rule Similar to the traditional IRLS, our iterative quadratic approximation method is not guaranteed to converge for logistic regression. However, we have not encountered any divergent problems in our numerical experiments. Based on our empirical experience, the quadratic approximation converges very quickly, usually in fewer than 10 iterations. We set MaxIteration= 50 and use the stopping criterion based on the absolute difference between the current estimate $\boldsymbol{\beta}^{(t)}$ and the previous estimate $\boldsymbol{\beta}^{(t-1)}$. In particular, we define the stopping criterion as:

$$\sum_{k=1}^{K} \|\boldsymbol{\beta}_k^{(t)} - \boldsymbol{\beta}_k^{(t-1)}\|_2 < \tau.$$

In our algorithm, τ is set at 10^{-4}.

4 Extensions to Multiclass Support Vector Machines

We propose two ways of extending the supSCAD penalty to multiclass support vector machines. The first method is in the context of the MSVM framework of Lee et al. (2004), and the proposed supSCAD MSVM solves

$$\begin{aligned} &\min_{\boldsymbol{\beta}_0,\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)\big[\beta_{k0}+\boldsymbol{\beta}_k^T\mathbf{x}_i+1\big]_+ + \sum_{j=1}^{d} J_\lambda(\|\boldsymbol{\beta}_{(j)}\|_\infty) \\ &\text{subject to } \sum_{k=1}^{K}\beta_{k0}=0, \text{ and } \sum_{k=1}^{K}\beta_{kj}=0, \quad j=1,\ldots,d. \end{aligned} \tag{11}$$

Next, we adopt the multiclass proximal SVMs of Tang and Zhang (2006) and propose the supSCAD MPSVM by solving

$$\begin{aligned} &\min_{\boldsymbol{\beta}_0,\boldsymbol{\beta}} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i \neq k)\big[\beta_{k0}+\boldsymbol{\beta}_k^T\mathbf{x}_i+1\big]^2 + \sum_{j=1}^{d} J_\lambda(\|\boldsymbol{\beta}_{(j)}\|_\infty) \\ &\text{subject to } \sum_{k=1}^{K}\beta_{k0}=0, \text{ and } \sum_{k=1}^{K}\beta_{kj}=0, j=1,\ldots,d. \end{aligned} \tag{12}$$

4.1 Computational Algorithms

We present the computational algorithms for solving the supSCAD MSVM. First, we introduce a set of slack variables

$$\begin{aligned} &\delta_{ik}=I(y_i\neq k) \text{ and } \xi_{ik}=\big[\beta_{k0}+\boldsymbol{\beta}_k^T\mathbf{x}_i+1\big]_+, \text{ for } i=1,\ldots,n,\ k=1,\ldots,K, \\ &\eta_j=\|\boldsymbol{\beta}_{(j)}\|_\infty=\max_{k=1,\ldots,K}|\beta_{kj}|, \text{ for } j=1,\ldots,d, \end{aligned}$$

and the new constraints $|\beta_{kj}|\leqslant\eta_j$, for $k=1,\ldots,K$ and $j=1,\ldots,d$. Then by applying DCA and LLA, we tackle the non-convex minimization supSCAD MSVM (11) by solving a sequence of linear programming (LP) problems,

$$\begin{aligned} \textbf{DCA}\ &\min_{\boldsymbol{\beta}_0,\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\eta}} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\delta_{ik}\xi_{ik}+\lambda\sum_{j=1}^{d}\eta_j-\sum_{j=1}^{d}J'_{\lambda,2}\big(\eta_j^{(t)}\big)\big(\eta_j-\eta_j^{(t)}\big) \\ \textbf{LLA}\ &\min_{\boldsymbol{\beta}_0,\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\eta}} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\delta_{ik}\xi_{ik}+\sum_{j=1}^{d}J'_{\lambda}\big(\eta_j^{(t)}\big)\cdot\eta_j \\ &\text{subject to } \sum_{k=1}^{K}\beta_{k0}=0, \quad \sum_{k=1}^{K}\beta_{kj}=0, \quad j=1,\ldots,d, \\ &\xi_{ik}\geqslant\beta_{k0}+\boldsymbol{\beta}_k^T\mathbf{x}_i+1, \xi_{ik}\geqslant 0, i=1,\ldots,n \text{ and } k=1,\ldots,K \\ &\boldsymbol{\beta}_{(j)}\leqslant\eta_j\mathbf{1}_K, -\boldsymbol{\beta}_{(j)}\leqslant\eta_j\mathbf{1}_K, j=1,\ldots,d. \end{aligned} \tag{13}$$

The complete algorithm for fitting the supSCAD MSVM is described in the Appendix, and the supSCAD MPSVM can be solved in a similar fashion.

5 Simulation Studies

We evaluate and illustrate finite sample performance of the supSCAD estimators in terms of their accuracy in variable selection, class prediction, and probability estimation, under three simulation experiment settings: (1) a four-class linear classification example with two "strong" important variables and two "weak" important variables for all the classes; (2) a four-class linear classification example with different important variables across classes; (3) a three-class high-dimensional example with two important variables for all the classes and 198 noise variables.

For comparison, we consider four soft classifiers and six hard classifiers. Four soft classification methods are L_1 logistic regression (L1 LR), group-L_1 logistic regression (GroupL1 LR), supSCAD logistic regression (supSCAD LR), and the composite MCP logistic regression (comp-MCP LR). Six hard classification methods are the standard MSVM (L2 MSVM), L_1 MSVM (L1 MSVM), supSCAD MSVM/MPSVM, and the composite MCP MSVM/MPSVM. The Bayes rule is also implemented as the reference. All simulations are conducted using Matlab (MATLAB, 2014) and Tomlab (Holmström et al., 2010), an optimization environment within Matlab.

For each experiment, we generate a training data set of size n and a test data set of size n' from the same distribution. We use the training data to train the classifiers and select the best tuning parameter from a series of λ values, and use the test data to evaluate performance of the estimated classifiers. A total of 100 simulations are conducted under all settings. Each classifier is evaluated in terms of its prediction and variable selection accuracy. For a soft classifier, we also examine its performance in estimating the conditional class probabilities.

To choose an appropriate λ for penalized logistic regression, we use a BIC selection criterion

$$BIC(\lambda) = -\frac{2}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i = k)\log \hat{p}_k(\mathbf{x}_i) + \frac{1}{n}\log(n) \times df,$$

where df is the number of nonzero coefficients in $\hat{\boldsymbol{\beta}}$. According to Zou (2006), the number of nonzero coefficients is an unbiased estimate for the degrees of freedom of LASSO-type methods. We adopt this result here as an approximated solution. For the regularized MSVM/MPSVM, we choose the best λ by five-fold cross validation (CV). Both BIC and CV search the best tuning parameter λ over a grid: $\log_2(\lambda) = -10, -9, \ldots, 10$. The optimum λ is identified by the least BIC or CV score. When ties occur, the larger λ is used for higher sparsity. For the tuning parameter a in comp-MCP, Breheny and Huang (2009) suggested using $a = 30$ in logistic regression and $a = 3$ in linear regression. However, in our simulations, we found that $a = 3$ had better performance in almost all the settings and therefore used $a = 3$ throughout the paper.

The true conditional class probabilities $p_k(\mathbf{x}), k = 1, \ldots, K$ are known in simulation. We evaluate the probability estimation by three criteria using the test set:

- $P1$ error: $\frac{1}{n'}\sum_{i=1}^{n'}\sum_{k=1}^{K}|\hat{p}_k(\mathbf{x}_i^{test}) - p_k(\mathbf{x}_i^{test})|$
- $P2$ error: $\frac{1}{n'}\sum_{i=1}^{n'}\sum_{k=1}^{K}(\hat{p}_k(\mathbf{x}_i^{test}) - p_k(\mathbf{x}_i^{test}))^2$
- Empirical Generalized Kullback-Leibler (EGKL) loss: $\frac{1}{n'}\sum_{i=1}^{n'}\sum_{k=1}^{K} p_k(\mathbf{x}_i^{test})\log\frac{p_k(\mathbf{x}_i^{test})}{\hat{p}_k(\mathbf{x}_i^{test})}$.

Each classifier consists of K discriminant functions. And there are totally $K \times d$ coefficients in the model. To assess variable selection performance, we use the following criteria:

- *Correct Zero*: the number of zero estimates which are truly zero
- *Incorrect Zero*: the number of zero estimates which are truly nonzero
- *Model Size*: the number of covariates which are selected in the final model
- *Correct Model*: the frequency of selecting the correct model over 100 simulations

Table 1: Summary and comparison of simulation results in Example 1.

Method	Correct Zeros	Incorrect Zeros	Model Size	Correct Model	Testing Error[a]
L1 LR	59.61	0.45	9.89	10	0.32
GroupL1 LR	61.04	0.08	4.73	46	0.31
supSCAD LR	61.92	0.72	4.34	70	0.32
comp-MCP LR	62.41	6.12	2.98	14	0.38
L2 MSVM	0	0	20	0	0.37
L1 MSVM	10.05	0.08	19.05	0	0.37
supSCAD MSVM	50.28	0.32	7.35	60	0.33
supSCAD MPSVM	53.10	0	6.73	45	0.30
comp-MCP MSVM	21.67	0.36	15.68	5	0.37
comp-MCP MPSVM	33.75	0.04	14.86	0	0.33
Bayes rule	64	0	4	100	0.29

[a]Entries in this column have standard errors in the range of 0.001 to 0.003.

Example 1 (Strong and weak signals)**.** Assume $K = 4$ and $d = 20$. For each class $k = 1, \ldots, 4$, we generate the first two components of $\mathbf{x}$ from $N(\mu_k, 3I_2)$, where $\mu_1 = (\sqrt{2}, \sqrt{2})^T$, $\mu_2 = (-\sqrt{2}, \sqrt{2})^T$, $\mu_3 = (-\sqrt{2}, -\sqrt{2})^T$, $\mu_4 = (\sqrt{2}, -\sqrt{2})^T$, and I_2 is a 2×2 identity matrix. Then, for each class k, we generate two additionally "weak" important variables from $N(\nu_k, I_2)$, where $\nu_1 = (0.5, 0.5)^T$, $\nu_2 = (-0.5, 0.5)^T$, $\nu_3 = (0.5, -0.5)^T$, $\nu_4 = (-0.5, -0.5)^T$. The remaining sixteen variables are noise and generated i.i.d. from $N(0, 1)$. The sample size is $n = 200$ and the test size is $n' = 40,000$.

Table 1 shows that, among all methods, supSCAD LR identified the true model most frequently, 70 out of 100, while supSCAD MPSVM produced the lowest classification error rate at 0.30. The column of "Incorrect Zeros" shows that, all of the penalized methods except supSCAD MPSVM missed the two weak signals sometimes, and the comp-MCP LR missed the weak signals most often having the smallest average model size of 2.98 and the largest average Incorrect-zero of 6.12. This over-shrinkage result may be due to the double penalty of the composite MCP, at both the group and individual levels.

The probability estimation results for all the simulations are summarized in Table 2. The supSCAD LR was overall the best among all the procedures and it has the smallest values in P1, P2 and EGKL.

Example 2 (Different important variables across classes)**.** This example considers a case where the set of important variables varies are from class to class. Let $d = 20$. For the input $\mathbf{x}$, its first four elements x_1, x_2, x_3, x_4 are generated uniformly from a 4-dimension ball with radius 3, i.e. $\{(x_1, x_2, x_3, x_4)^T \in R^4 : \sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2} \leqslant 3\}$, and the remaining $x_5, \ldots, x_{20}$ are generated *i.i.d.* from $N(0, 1)$. Furthermore, define the four discriminating functions as $f_1 = -3x_1 + 3x_4$, $f_2 = 3x_1 + 3x_2$, $f_3 = -3x_2 + 3x_3$, and $f_4 = -3x_3 - 3x_4$. We set $p_k(x) = P(Y = k|X = x) \propto \exp(f_k(x))$, $k = 1, 2, 3, 4$ and kept $n = 200$ and $n' = 40,000$. Among the first 4 important variables, x_1 is non-informative for distinguishing class 3 vs class 4; x_2 is non-informative for distinguishing class 1 vs class 4; x_3 is non-informative for distinguishing class 1 vs class 2; and x_4 is non-informative for distinguishing class 2 vs class 3.

Table 2: Probability Estimation for all three simulation examples.

	Example 1			Example 2			Example 3		
Methods[a]	P1	P2	EGKL	P1	P2	EGKL	P1	P2	EGKL
L1 LR	0.36	0.06	0.12	0.38	0.07	0.17	0.30	0.04	0.11
GroupL1 LR	0.39	0.06	0.13	0.48	0.11	0.24	0.30	0.05	0.11
supSCAD LR	0.24	0.04	0.07	0.21	0.03	0.07	0.32	0.05	0.12
comp-MCP LR	0.44	0.10	0.19	0.20	0.03	0.07	0.32	0.05	0.12

[a]Entries in this column have standard errors in the range of 0.001 to 0.02.

Table 3: Summary and comparison of simulation results in Example 2.

Method	Correct Zeros	Incorrect Zeros	Model Size	Correct Model	Testing Error[a]
L1 LR	67.75	0	7.44	5	0.13
GroupL1 LR	62.48	0	4.39	72	0.13
supSCAD LR	64.00	0	4	100	0.12
comp-MCP LR	71.71	0	4	100	0.11
L2 MSVM	0	0	20	0	0.36
L1 MSVM	22.82	0.16	17.12	4	0.32
supSCAD MSVM	27.17	0.2	13.43	4	0.33
supSCAD MPSVM	41.49	0	9.63	2	0.19
comp-MCP MSVM	36.87	0.42	12.23	15	0.32
comp-MCP MPSVM	47.77	0	11.94	24	0.19
Bayes rule	72	0	4	100	0.11

[a]Entries in this column have standard errors in the range of 0.001 to 0.007.

The supSCAD focuses on the group-level selection and it does not imposes penalty at the individual level. By design, this example has important variables with zero coefficients for certain classes, which favors the bi-level selection procedures such as the comp-MCP most. However, supSCAD LR still gave satisfactory results. The top half of Table 3 shows that both comp-MCP and supSCAD LR did the perfect job in model selection. Besides getting the lowest test error as the Bayes rule, the comp-MCP LR also identified most correct zeros in the first four important variables. In the bottom half, the comp-MCP MPSVM had the smallest test error and the largest frequency of selecting the correct model.

Example 3 (High-dimensional classification)**.** We consider a similar example to Example 1 in Zhang and Liu (2014). A three-class dataset was generated with $P(Y = k) = 1/3$, and $X|Y = k \sim N(\mu_k, \sigma^2 I_2)$ for $k = 1, 2, 3$, where the first two covariates of μ_ks were equally distributed on the unit circle specifically are $(\cos(\pi/3), \sin(\pi/3))$, $(\cos(\pi), \sin(\pi))$, and $(\cos(5\pi/3), \sin(5\pi/3))$, respectively, and the remaining 198 covariates were *i.i.d.* noise generated from $N(0, 0.5)$. The error variance σ^2 was chosen to have the Bayes error$= 0.1$. We had $n = 200$ and $n' = 40,000$.

Table 4 shows that all the penalized LR methods did very well in class prediction. In particular, the supSCAD LR distinguished the two important predictors from the other 198

Table 4: Summary and comparison of simulation results in Example 3.

Method	Correct Zeros	Incorrect Zeros	Model Size	Correct Model	Testing Error[a]
L1 LR	594.61	2.00	2.39	78	0.10
GroupL1 LR	591.65	0	2.79	48	0.10
supSCAD LR	594	0	2.00	100	0.10
comp-MCP LR	594.47	0	2.09	91	0.10
L2 MSVM	0	0	200	0	0.36
L1 MSVM	563.48	0	16.71	22	0.12
supSCAD MSVM	589.05	0	4.39	60	0.11
supSCAD MPSVM	569.51	0	13.7	32	0.11
comp-MCP MSVM	579.59	0	8.82	2	0.12
comp-MCP MPSVM	574.98	0	11.12	87	0.11
Bayes rule	595	0	2	100	0.10

[a]Entries in this column have standard errors in the range of 0.001 to 0.003.

noise variables each time. Although the comp-MCP LR selected the correct model with a slightly lower frequency, it had larger Correct Zero numbers since it identified the zero coefficient of the second important predictor in the second class sometimes. In this example, the L1 LR method outperformed the GroupL1 LR in model selection, however it was the only method that had non-zero "Incorrect Zeros", i.e., it missed some important predictions occasionally. In high-dimensional settings, the benefit of penalization became more obvious. Additionally, the composite-MCP seemed to perform unstably with MSVM and MPSVM in terms of variable selection. A possible explanation is that there are two tuning parameters involved and it is difficult to tune these parameters precisely altogether.

6 Real Data Applications

We consider two real data examples: the Small Round Blue Cell Tumors data set and the Semeion Handwritten Digit data set.

Small Round Blue Cell Tumors (SRBCTs) SRBCTs in childhood can be categorized into 4 classes: neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL), and the Ewing family of tumors (EWS) (Khan et al., 2001), and Burkitt lymphoma (BL) is a subset of NHL. The SRBCTs data set is a cDNA microarray data with 63 training samples and 20 independent test samples. The four tumor class distributions in the training and test sets are given in Table 5. After filtering, the expression of 2308 genes out of 6567 genes are given at http://research.nhgri.nih.gov/microarray/Supplement/.

Before applying any classification method, we first standardize the data set based on gene expression means and standard deviations from the training set. Therefore, the gene expressions in training have mean zero and standard deviation one. Next, we rank all the genes based on their marginal separation power in the training set (Dudoit et al., 2002). Specifically, the relevance measure for gene g is defined to be the ratio of the between-class sum of squares to the

Table 5: Class distribution of the SRBCTs data.

Dataset	NB	RMS	BL	EWS	Total
Training	12	20	8	23	63
Test	6	5	3	6	20

Table 6: Classification results for two real datasets.

Method	L1 LR	GroupL1 LR	supSCAD LR	comp-MCP LR	L2 MSVM
SRBCT	0	0.05	0	0	0
Semeion	0.37	0.35	0.11	0.12	0.20
Method	L1 MSVM	supSCAD MSVM	supSCAD MPSVM	comp-MCP MSVM	comp-MCP MPSVM
SRBCT	0.05	0.10	0.05	0.05	0.05
Semeion	0.27	0.25	0.19	0.27	0.19

Table 7: Class distribution of Semeion Handwritten Digit data.

Dataset	0	1	2	3	4	5	6	7	8	9	Total
Training	135	135	130	130	135	130	135	130	130	130	1320
Test	26	27	29	29	26	29	26	28	25	28	273

within-class sum of squares as follows: $R(g) = \frac{\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i=k)(\bar{x}_{.g}^{(k)}-\bar{x}_{.g})^2}{\sum_{i=1}^{n}\sum_{k=1}^{K} I(y_i=k)(x_{ig}-\bar{x}_{.g}^{(k)})^2}$, where n is the size of the training set, $\bar{x}_{.g}^{(k)}$ denotes the average expression level of gene g for class k observations, and $\bar{x}_{.g}$ is the overall mean expression level of gene g in the training set. We then select the top 100 and bottom 100 genes as the prediction covariates according to their relevance measure R values.

We apply the four penalized logistic regression approaches and six MSVMs to the training data of 200 genes and 63 training samples. Leave-one-out cross validation is used to select the optimal tuning parameter, and then the trained models are used to predict the class labels for 20 test samples. Results are summarized and compared in the first row of Table 6. It is observed that the LR-based methods have overall better classification accuracy than the SVM-based methods for this data set, and the L1 LR, supSCAD LR, and the comp-MCP LR methods all have the test error zero.

Semeion Handwritten Digit Data This data set is available from the University of California Irvine machine learning repository website. It consists of 1593 handwritten digits (0–9), each of which was scanned and stretched in a square box 16×16 in a gray scale of 256 values (i.e. 256 attributes). Then each pixel of each image was scaled into a Boolean (1/0) value of a fixed threshold. The whole data set is roughly equally distributed among the 10 classes. We split the data into six groups with roughly equal size and class distribution, with one group used for testing and the remaining five groups used for training. We performed 5-fold CV to select the best tuning parameter. The class distributions of the training and testing sets are summarized in Table 7, and the classification results are the second row of Table 6. For this data set, the supSCAD LR gives the lowest classification error 0.11.

7 Discussion

In this paper, we propose and study a new class of regularization methods for variable selection in multiclass classification problems. This new penalty, supSCAD, enhances sparse learning in multi-classification by retaining the merits from both SCAD and supnorm penalties. It can incorporate the natural group effects of the coefficients associated with the same covariate to construct more parsimonious classifiers with desired oracle properties.

To tackle the numerical challenge of non-differentiability and non-convexity of the objective function, we have proposed an efficient iterative algorithm based on the local linear approximation (LLA) and DCA. For multiclass probability estimation, supSCAD is applied to multinomial logistic regression to conduct variable selection and conditional probability estimation simultaneously. An optimization procedure involving quadratic approximation to the multinomial log-likelihood function and nested DCA/LLA for the supSCAD penalty is developed and evaluated by numeric experiments. We further extend the penalty to the multi-category SVM framework and develop supSCAD MSVM/MPSVM, which demonstrate competitive performance compared to other regularized MSVM in simulated and real data examples.

The major underlying assumption of the supSCAD penalty is that the covariates across different classes can be naturally grouped for each predictor. Though this assumption may not hold for very complex problems, we find that it can still achieve reasonably good results even when the assumption is violated. For further improvement, our supSCAD penalty can be easily extended to incorporate within-group sparsity structure by imposing additional penalty on individual coefficients, such as LASSO or adaptive LASSO penalty, e.g., $\sum_{j=1}^{d} J_\lambda(\|\boldsymbol{\beta}_{(j)}\|_\infty)+\lambda_c \sum_{k=1}^{K}\sum_{j=1}^{d}|\beta_{kj}|$. However, choosing the extra tuning parameter λ_c may cause additional computation cost. Its theoretical and computational properties are interesting for investigation in future work.

Supplementary Material

A zip file includes all the computation code and data for the numerical experiments is available.

Appendix

Before presenting proofs of the theorems, we first state some regularity conditions, which mostly follow Fan et al. (2004). The reparametrized multinomial log-likelihood $\tilde{\mathcal{L}}$ and its associated true parameter vector $\beta^\star$ are defined in (7) and (8).

Regularity Conditions

1. The observations $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, are i.i.d. with multinomial distribution $(\pi_1, \ldots, \pi_K)$, $1 > \pi_k > 0$, $\sum_{k=1}^{K} \pi_k = 1$.
2. The Fisher information matrix $I(\boldsymbol{\beta}) = E\{(\frac{\partial \tilde{\mathcal{L}}}{\partial \boldsymbol{\beta}})(\frac{\partial \tilde{\mathcal{L}}}{\partial \boldsymbol{\beta}})^T\}$ is finite and positive definite at $\boldsymbol{\beta} = \boldsymbol{\beta}^\star$ for all n observations. For $j, k = 1, 2, \ldots, d$,

$$E\left\{\left(\frac{\partial \tilde{\mathcal{L}}}{\partial \beta_j}\right)\left(\frac{\partial \tilde{\mathcal{L}}}{\partial \beta_k}\right)\right\}^2 < C_1 < \infty$$

and

$$E\left\{\left(\frac{\partial \tilde{\mathcal{L}}^2}{\partial \beta_j \partial \beta_k}\right)\right\}^2 < C_2 < \infty.$$

3. There is a sufficient large enough open set w that contains $\boldsymbol{\beta}^\star$ such that for almost all n observations the density admits all third derivatives $\frac{\partial \tilde{\mathcal{L}}}{\partial\beta_j\beta_k\beta_l}$ for all $\boldsymbol{\beta}^\star \in w$, and

$$\left|\frac{\partial \tilde{\mathcal{L}}^3}{\partial\beta_{kj}\partial\beta_{k_1j_1}\partial\beta_{k_2j_2}}\right| \leqslant M(\mathbf{x}) < \infty$$

and

$$E_{\boldsymbol{\beta}^\star}\big[M(\mathbf{x})\big] < \infty$$

for all $n,\ k,\ j,\ k_1,\ j_1,\ k_2,\ j_2$.

4. Let the first s_n values of $\boldsymbol{\beta}$ be nonzero, and the rest $d_n - s_n$ values be zero. Then the $\beta_1, \beta_2, \ldots, \beta_s$ satisfy

$$\min_{1\leqslant j\leqslant s_n} |\beta_j|/\lambda \to \infty \text{ as } \mathrm{n} \to \infty.$$

Proof of Theorem 1

Proof. To prove Theorem 1, it is enough to show that for any given $\varepsilon > 0$, there exists a large enough constant C such that

$$P\Big\{\inf_{||\mathbf{u}||=C} \tilde{R}\big(\boldsymbol{\beta}^\star + \mathbf{u}\sqrt{d_n/n}\big) > \tilde{R}\big(\boldsymbol{\beta}^\star\big)\Big\} \geqslant 1-\varepsilon, \qquad (\star)$$

which implies that, with probability at least $1-\varepsilon$, there exists a local minimum in the ball $\{\boldsymbol{\beta}^\star + \mathbf{u}\sqrt{d_n/n} : ||\mathbf{u}|| \leqslant C\}$. This in turn implies that there exists a local minimizer such that $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\star\| = O_p(\sqrt{d_n/n})$, which is exactly what we want to show. Notice that

$$\begin{aligned}
&\tilde{R}\big(\boldsymbol{\beta}^\star + \mathbf{u}\sqrt{d_n/n}\big) - \tilde{R}\big(\boldsymbol{\beta}^\star\big)\\
&\quad\geqslant -\big(\tilde{\mathcal{L}}\big(\boldsymbol{\beta}^\star + \mathbf{u}\sqrt{d_n/n}\big) - \tilde{\mathcal{L}}\big(\boldsymbol{\beta}^\star\big)\big)\\
&\qquad + n\sum_{j=1}^{s}\Bigg[J_{\lambda_n}\Bigg(\max\Bigg\{\Bigg|\beta^\star_{1j} + \frac{u_{1j}\sqrt{d_n}}{\sqrt{n}}\Bigg|, \Bigg|\beta^\star_{2j} + \frac{u_{2j}\sqrt{d_n}}{\sqrt{n}}\Bigg|, \ldots, \Bigg|\beta^\star_{(K-1),j} + \frac{u_{K-1,j}\sqrt{d_n}}{\sqrt{n}}\Bigg|,\\
&\qquad\qquad \Bigg|\sum_{l=1}^{K-1}\Bigg(\beta^\star_{lj} + \frac{u_{lj}\sqrt{d_n}}{\sqrt{n}}\Bigg)\Bigg|\Bigg\}\Bigg) - J_{\lambda_n}\Bigg(\max\Bigg\{|\beta^\star_{1j}|, |\beta^\star_{2j}|, \ldots, |\beta^\star_{(K-1),j}|, \Bigg|\sum_{l=1}^{K-1}\beta^\star_{lj}\Bigg|\Bigg\}\Bigg)\Bigg]
\end{aligned}$$

following the proof in Fan et al. (2004)

$$= -\tilde{\mathcal{L}}'\big(\boldsymbol{\beta}^\star\big)^T\frac{\mathbf{u}\sqrt{d_n}}{\sqrt{n}} + \frac{d_n}{2}\mathbf{u}^T I\big(\boldsymbol{\beta}^\star\big)\mathbf{u}\big\{1+o_p(1)\big\} + D_3 = D_1 + D_2 + D_3$$

Note that $\tilde{\mathcal{L}}'(\boldsymbol{\beta}^\star)^T = O_p(\sqrt{d_n n})$, thus D_2 is asymptotic positive and it dominates D_1 by choosing a sufficiently large C. Since the supSCAD penalty is flat for coefficients of magnitude larger than $a\lambda_n$ as $n \to \infty$,

$$\begin{aligned}
D_3 = n\sum_{j=1}^{s_n}\Bigg[&J_{\lambda_n}\Bigg(\max\Bigg\{\Bigg|\beta^\star_{1j} + \frac{u_{1j}}{\sqrt{n}}\Bigg|, \Bigg|\beta^\star_{2j} + \frac{u_{2j}}{\sqrt{n}}\Bigg|, \ldots, \Bigg|\beta^\star_{(K-1),j} + \frac{u_{K-1,j}}{\sqrt{n}}\Bigg|, \Bigg|\sum_{l=1}^{K-1}\Bigg(\beta^\star_{lj} + \frac{u_{lj}}{\sqrt{n}}\Bigg)\Bigg|\Bigg\}\Bigg)\\
&- J_{\lambda_n}\Bigg(\max\Bigg\{|\beta^\star_{1j}|, |\beta^\star_{2j}|, \ldots, |\beta^\star_{(K-1),j}|, \Bigg|\sum_{l=1}^{K-1}\beta^\star_{lj}\Bigg|\Bigg\}\Bigg)\Bigg] = 0.
\end{aligned}$$

Based on the above, $\tilde{R}(\boldsymbol{\beta}^\star + \mathbf{u}\sqrt{d_n/n}) - \tilde{R}(\boldsymbol{\beta}^\star)$ is dominated by D_2. Hence, by choosing a sufficient large C $(\star)$ holds. □

Proof of Lemma 1

Proof. As long as the $\max\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{K-1,j}|, |\sum_{m=1}^{K-1}\beta_{mj}|\}$ is zero, then each component in $\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{K-1,j}|, |\sum_{m=1}^{K-1}\beta_{mj}|\}$ is zero.

It is sufficient to show that with probability tending to 1 as $n \to \infty$, for any $\boldsymbol{\beta}_+$ satisfying $||\boldsymbol{\beta}_+ - \boldsymbol{\beta}^\star_+|| = O_p(\sqrt{d_n/n})$ and any constant C, for $j = s_n + 1, \ldots, d$,

$$\frac{\partial \tilde{R}(\boldsymbol{\beta})}{\partial \beta^r_{kj}} > 0, \text{ for } 0 \leqslant \beta_{kj} \leqslant C\sqrt{d_n/n} \text{ and } \beta_{kj} = \max\left\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{K-1,j}|, \left|\sum_{m=1}^{K-1}\beta_{mj}\right|\right\}$$

$$\frac{\partial \tilde{R}(\boldsymbol{\beta})}{\partial \beta^l_{kj}} < 0, \text{ for } -C\sqrt{d_n/n} \leqslant \beta_{kj} \leqslant 0 \text{ and } \beta_{kj} = -\max\left\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{K-1,j}|, \left|\sum_{m=1}^{K-1}\beta_{mj}\right|\right\}$$

where $\frac{\partial \tilde{R}}{\partial \beta^r_{kj}}$ and $\frac{\partial \tilde{R}}{\partial \beta^l_{kj}}$ denote the right and left hand partial derivative respectively.

$$\frac{\partial \tilde{R}}{\partial \beta_{kj}} = -\frac{\partial \tilde{\mathcal{L}}}{\partial \beta_{kj}} + nJ'_{\lambda_n}(|\beta_{kj}|)sgn(\beta_{kj}) \text{ when } |\beta_{kj}| = \max\left\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{K-1,j}|, \left|\sum_{m=1}^{K-1}\beta_{mj}\right|\right\}$$

By Taylor expansion,

$$\begin{aligned}\frac{\partial \tilde{\mathcal{L}}(\boldsymbol{\beta})}{\partial \beta_{kj}} &= \frac{\partial \tilde{\mathcal{L}}(\boldsymbol{\beta}^\star)}{\partial \beta_{kj}} + \sum_{j_1=1}^{d}\sum_{k_1=1}^{K-1}\frac{\partial^2 \tilde{\mathcal{L}}(\boldsymbol{\beta}^\star)}{\partial \beta_{kj}\partial \beta_{k_1 j_1}}\left(\beta_{k_1 j_1} - \beta^\star_{k_1 j_1}\right) \\ &\quad + \sum_{j_1=1}^{d_n}\sum_{k_1=1}^{K-1}\sum_{j_2=1}^{d_n}\sum_{k_2=1}^{K-1}\frac{\partial^3 \tilde{\mathcal{L}}(\boldsymbol{\beta}^*)}{\partial \beta_{kj}\partial \beta_{k_1 j_1}\partial \beta_{k_2 j_2}}\left(\beta_{k_1 j_1} - \beta^\star_{k_1 j_1}\right)\left(\beta_{k_2 j_2} - \beta^\star_{k_2 j_2}\right)\end{aligned}$$

where $\boldsymbol{\beta}^*$ lies between $\boldsymbol{\beta}$ and $\boldsymbol{\beta}^\star$. Note that $\frac{\partial \tilde{\mathcal{L}}(\boldsymbol{\beta}^\star)}{\partial \beta_{kj}} = O_p(\sqrt{d_n n})$ and $\frac{1}{n}\frac{\partial^2 \tilde{\mathcal{L}}(\boldsymbol{\beta}^\star)}{\partial \beta_{kj}\partial \beta_{k_1 j_1}} = E\{\frac{\partial^2 \tilde{\mathcal{L}}(\boldsymbol{\beta}^\star)}{\partial \beta_{kj}\partial \beta_{k_1 j_1}}\} + o_p(1)$. We have

$$\begin{aligned}\frac{\partial \tilde{R}(\boldsymbol{\beta})}{\partial \beta^r_{kj}} &= n\lambda_n\left\{O_p(\sqrt{d_n/n}/\lambda_n) + \frac{J'_{\lambda_n}(|\beta_{kj}|)}{\lambda_n}\right\} \text{ if } \beta_{kj} > 0 \\ \frac{\partial \tilde{R}(\boldsymbol{\beta})}{\partial \beta^l_{kj}} &= n\lambda_n\left\{O_p(\sqrt{d_n/n}/\lambda_n) - \frac{J'_{\lambda_n}(|\beta_{kj}|)}{\lambda_n}\right\} \text{ if } \beta_{kj} < 0\end{aligned}$$

In both cases, the second term dominates the first term.

If $|\kappa_j| = |\sum_{m=1}^{K-1}\beta_{mj}| = \max\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{K-1,j}|, |\sum_{m=1}^{K-1}\beta_{mj}|\}$, then

$$\frac{\partial \tilde{R}}{\partial \kappa_j} = -\frac{\partial \tilde{\mathcal{L}}}{\partial \beta_{kj}}\frac{\partial \beta_{kj}}{\partial \kappa_j} + nJ'_{\lambda_n}(|\kappa_j|)sgn(\kappa_j) = -\frac{\partial \tilde{\mathcal{L}}}{\partial \kappa_j} + nJ'_{\lambda_n}(|\kappa_j|)sgn(\kappa_j).$$

Therefore,

$$\begin{aligned}\frac{\partial \tilde{R}(\boldsymbol{\beta})}{\partial \kappa^r_j} &= n\lambda_n\left\{O_p(\sqrt{d_n/n}/\lambda_n) + \frac{J'_{\lambda_n}(|\kappa_j|)}{\lambda_n}\right\} \text{ if } \kappa_j > 0 \\ \frac{\partial \tilde{R}(\boldsymbol{\beta})}{\partial \kappa^l_j} &= n\lambda_n\left\{O_p(\sqrt{d_n/n}/\lambda_n) - \frac{J'_{\lambda_n}(|\kappa_j|)}{\lambda_n}\right\} \text{ if } \kappa_j < 0\end{aligned}$$

The second term still dominates the first term. Thus the result of Lemma 1 follows. □

Proof of Theorem 2

Proof. Part 1 holds by Lemma 1. For Part 2

$$\begin{aligned}
&\lambda_n \leqslant a^{-1} \max_{1\leqslant j\leqslant s} \max\left\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{K-1,j}|, \left|\sum_{m=1}^{K-1}\beta_{mj}\right|\right\} \\
&\quad\Rightarrow \max_{1\leqslant j\leqslant s} \max\left\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{K-1,j}|, \left|\sum_{m=1}^{K-1}\beta_{mj}\right|\right\} \geqslant a\lambda_n \\
&\quad\Rightarrow J_{\lambda_n}\left(\max\left\{|\beta_{1j}|, |\beta_{2j}|, \ldots, |\beta_{K-1,j}|, \left|\sum_{m=1}^{K-1}\beta_{mj}\right|\right\}\right) = J_{\lambda_n}(a\lambda_n) = \frac{(a+1)^2\lambda_n^2}{2}.
\end{aligned}$$

Therefore $\arg\min \tilde{R}(\boldsymbol{\beta}) = \arg\min -\tilde{\mathcal{L}}(\boldsymbol{\beta})$. The desired result follows. □

Algorithm 1 for supSCAD multinomial logistic regression

Algorithm 1

Initialize $\boldsymbol{\beta}^{(0)}$.
for (t = 0 to MaxIteration)
Compute $Q(\boldsymbol{\beta}^{(t)})$ in (9).
Use DCA/LLA to solve the problem (10); denote the solution by $\boldsymbol{\beta}^{(t+1)}$.
Evaluate the objective function given in (7) at $\boldsymbol{\beta}^{(t+1)}$.
if (the stopping criterion is satisfied), Break; **end**
end

Algorithm 2 for supSCAD MSVM

1. **Initialization:** $\boldsymbol{\beta}_0^{(0)} = 0$, $\boldsymbol{\beta}^{(0)} = 0$, $\boldsymbol{\xi}^{(0)} = 0$, $\boldsymbol{\eta}^{(0)} = 0$, $t = 0$.
2. **Repeat:** solve $\boldsymbol{\beta}_0^{(t+1)}$, $\boldsymbol{\beta}^{(t+1)}$, $\boldsymbol{\xi}^{(t+1)}$, and $\boldsymbol{\eta}^{(t+1)}$ from (13), $t = t + 1$.
3. **Stop:** $\boldsymbol{\beta}_0^{(t+1)}$ and $\boldsymbol{\beta}^{(t+1)}$ meet the rule of convergence.

Computation formula

Similarly, we can use the DCA/LLA approximation to compute supSCAD MPSVM by solving a series of quadratic programming (QP) problems:

$$\begin{aligned}
\textbf{DCA}\ &\min_{\boldsymbol{\beta}_0,\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\eta}} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\delta_{ik}\left[\beta_{k0} + \boldsymbol{\beta}_k^T\mathbf{x}_i + 1\right]^2 + \lambda\sum_{j=1}^{d}\eta_j - \sum_{j=1}^{d}J'_{\lambda,2}\left(\eta_j^{(t)}\right)\left(\eta_j - \eta_j^{(t)}\right) \\
\textbf{LLA}\ &\min_{\boldsymbol{\beta}_0,\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\eta}} \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\delta_{ik}\left[\beta_{k0} + \boldsymbol{\beta}_k^T\mathbf{x}_i + 1\right]^2 + \sum_{j=1}^{d}J'_{\lambda}\left(\eta_j^{(t)}\right)\cdot\eta_j \\
&\text{subject to } \sum_{k=1}^{K}\beta_{k0} = 0, \quad \sum_{k=1}^{K}\beta_{kj} = 0,\ j = 1,\ldots,d, \\
&\boldsymbol{\beta}_{(j)} \leqslant \eta_j\mathbf{1}_K,\ -\boldsymbol{\beta}_{(j)} \leqslant \eta_j\mathbf{1}_K,\ j = 1,\ldots,d.
\end{aligned}$$

References

Bradley PS, Mangasarian OL (1998). Feature selection via concave minimization and support vector machines. In: *ICML*, volume 98, 82–90.

Breheny P, Huang J (2009). Penalized methods for bi-level variable selection. *Statistics and Its Interface*, 2(3): 369.

Crammer K, Singer Y (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2: 265–292.

Dudoit S, Fridlyand J, Speed TP (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457): 77–87.

Fan J, Li R (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456): 1348–1360.

Fan J, Peng H, et al. (2004). Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics*, 32(3): 928–961.

Hastie T, Tibshirani R, Friedman J (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.

Holmström K, Göran AO, Edvall MM (2010). User's Guide for Tomlab 7.

Huang J, Breheny P, Ma S (2012). A selective review of group selection in high-dimensional models. *Statistical Science*, 27(4): 481–499.

Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M, Westermann F, et al. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6): 673–679.

Lange K, Hunter DR, Yang I (2000). Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1): 1–20.

Le Thi Hoai A, Tao PD (1997). Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. *Journal of Global Optimization*, 11(3): 253–285.

Lee Y, Lin Y, Wahba G (2004). Multicategory support vector machines: theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465): 67–81.

Liu Y, Shen X (2006). Multicategory ψ-learning. *Journal of the American Statistical Association*, 101(474): 500–509.

Liu Y, Yuan M (2011). Reinforced multicategory support vector machines. *Journal of Computational and Graphical Statistics*, 20(4): 901–919.

Mangasarian O, Wild E (2001). Proximal support vector machine classifiers. *Proceedings KDD-2001: Knowledge Discovery and Data Mining*. Citeseer.

MATLAB (2014). *version 8.3 (R2014a)*. The MathWorks Inc., Natick, Massachusetts.

McCullagh P, Nelder JA (1989). *Generalized Linear Models*, 2nd edition. Chapman and Hall, London, UK.

Suykens JA, Vandewalle J (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3): 293–300.

Tang Y, Zhang HH (2006). Multiclass proximal support vector machines. *Journal of Computational and Graphical Statistics*, 15(2): 339–355.

Tibshirani R (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B, Methodological*, 58(1): 267–288.

Tutz G, Pößnecker W, Uhlmann L (2015). Variable selection in general multinomial logit models. *Computational Statistics & Data Analysis*, 82: 207–222.

Vapnik V (1998). *Statistical Learning Theory*. Wiley-Interscience, New York.

Vapnik VN (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag.

Wang L, Shen X (2007). On l 1-norm multiclass support vector machines: Methodology and theory. *Journal of the American Statistical Association*, 102(478): 583–594.

Weston J, Watkins C, et al. (1999). Support vector machines for multi-class pattern recognition. In: *Esann*, volume 99, 219–224.

Wu Y, Liu Y (2009). Variable selection in quantile regression. *Statistica Sinica*, 19(2): 801–817.

Yuan M, Lin Y (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B, Statistical Methodology*, 68(1): 49–67.

Zhang C, Liu Y (2014). Multicategory angle-based large-margin classification. *Biometrika*, 101(3): 625–640.

Zhang CH, et al. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2): 894–942.

Zhang HH, Ahn J, Lin X, Park C (2006). Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, 22(1): 88–95.

Zhang HH, Liu Y, Wu Y, Zhu J, et al. (2008). Variable selection for the multicategory svm via adaptive sup-norm regularization. *Electronic Journal of Statistics*, 2: 149–167.

Zhao P, Rocha G, Yu B, et al. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 37(6A): 3468–3497.

Zou H (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476): 1418–1429.

Zou H, Li R (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, 36(4): 1509.

Zou H, Yuan M (2008). The f_∞-norm support vector machine. *Statistica Sinica*, 18(1): 379–398.