Variable Selection via A Combination of the L_0 and L_1 Penalties

Yufeng Liu and Yichao Wu*

March 27, 2007

Abstract

Variable selection is an important aspect of high-dimensional statistical modelling, particularly in regression and classification. In the regularization framework, various penalty functions are used to perform variable selection by putting relatively large penalties on small coefficients. The L_1 penalty is a popular choice because of its convexity, but it produces biased estimates for the large coefficients. The L_0 penalty is attractive for variable selection since it directly penalizes the number of non-zero coefficients. However, the optimization involved is discontinuous and nonconvex, and therefore it is very challenging to implement. Moreover, its solution may not be stable. In this article, we propose a new penalty which combines the L_0 and L_1 penalties. We implement this new penalty by developing a global optimization algorithm using Mixed Integer Programming (MIP). We compare this combined penalty with several other penalties via simulated examples as well as real applications. The results show that the new penalty outperforms both the L_0 and L_1 penalties in terms of variable selection while maintaining good prediction accuracy.

Key Words: Mixed integer programming, regression, regularization, SVM, variable selection.

*Yufeng Liu is Assistant Professor, Department of Statistics and Operations Research, Carolina Center for Genome Sciences, University of North Carolina, Chapel Hill, NC 27599 (Email: yfliu@email.unc.edu); Yichao Wu is Research Associate, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544 (Email: yichaowu@princeton.edu).

1 Introduction

Regression and classification are two commonly used techniques in statistics. In linear regression with p explanatory variables $x_1, \cdots, x_p$, the continuous response y may be predicted by $\hat{y} = \hat{b} + x_1\hat{w}_1 + \cdots + x_p\hat{w}_p$. While, in binary linear classification, the class label $y \in \{\pm 1\}$ can be predicted by $\hat{y} =$ sign $(\hat{b} + x_1\hat{w}_1 + \cdots + x_p\hat{w}_p)$. The vector of coefficients $\hat{\mathbf{w}} = (\hat{w}_1, \cdots, \hat{w}_p)^T$ and the intercept $\hat{b}$ are estimated from a training dataset using some model-fitting procedure, e.g., by minimizing the residual sum of squares (RSS), the ordinary least squares (OLS) regression yields the best linear unbiased estimator (BLUE) of $\mathbf{w}$. The criteria for evaluating the quality of a model may vary, with the goal often being both interpretability of the model and its prediction accuracy. In practice, a large number of candidate predictors are available for modelling, but keeping irrelevant variables in the model makes interpretation difficult and may decrease the predictability of the resulting model. Therefore, variable selection is an important aspect of the model building process.

Ridge regression, which minimizes RSS subject to the constraint $\sum_{j=1}^{p} |w_j|^2 \leq t;\ t > 0$, was first introduced by Hoerl and Kennard (1970). Frank and Friedman (1993) extended this idea to bridge regression by generalizing the constraint to $L_q(\mathbf{w}) = \sum_{j=1}^{p} |w_j|^q \leq t$ with $q > 0$. Bridge regression can perform variable selection only for $q \leq 1$, c.f., Knight and Fu (2000). The case $q = 1$ corresponds to the LASSO, which tends to produce a sparsity of nonzero coefficients in the resulting model (Tibshirani, 1996). However, the LASSO produces biased estimates for the large coefficients. Donoho and Johnstone (1994) and Donoho et al. (1995) also discussed the shrinkage property of the L_1 penalty in the wavelet setting.

Intuitively, the L_0 penalty $L_0(\mathbf{w}) = \sum_{j=1}^{p} I(w_j \neq 0)$, the limit of the L_q penalty as $q \to 0$, penalizes the number of nonzero coefficients directly and thus is desirable for variable selection.

However, due to its nonconvexity and discontinuity at the origin, it is very hard to implement the corresponding optimization problem. In addition, the solution using the L_0 penalty may be unstable because it may not be identifiable.

Similar methods have been proposed for classification, for example the Support Vector Machine (SVM). The SVM was first introduced by Vapnik and his co-authors (see Vapnik, 1998) and has now enjoyed great popularity in both the statistics and machine learning communities. The original idea of the standard SVM (the L_2 SVM) is to search for the optimal separating hyperplane with maximum separation between two classes. Such separation is measured by the so-called geometric margin, which is closely related to the L_2 penalty (Cristianini and Shawe-Taylor, 1999). To perform variable selection, Bradley and Mangarsarian (1998) investigated the SVM using the L_1 penalty. Zhu et al. (2003) proposed an algorithm for calculating the solution path of the L_1 SVM as a function of its tuning parameter.

In this paper, we propose a new penalty which combines the L_0 and L_1 penalties. We show that this new penalty retains the advantages of both the L_0 and L_1 penalties. It can deliver better variable selection than the L_1 penalty while yielding a more stable model than the L_0 penalty. A general regularization scheme for both regression and classification is considered. A new approach is proposed to implement the new penalty effectively. In particular, we convert the corresponding optimization problem into a mixed integer programming (MIP) problem. Then we utilize a state-of-the art algorithm for MIP from the operations research community to find the global optimizer.

The rest of this article is organized as follows. In Section 2, we consider the general regularization framework of regression and classification. The new combined penalty is introduced and its properties are discussed in Section 3. Section 4 introduces MIP for handling the new

penalty. Simulation studies in Section 5 and real examples in Section 6 show that our method produces effective prediction and variable selection simultaneously. Some discussion is given in Section 7. The Appendix collects the lemmas and technical proofs.

2 Regularization

In both regression and classification, we are given a training dataset $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i = (x_{1i}, \cdots, x_{pi})^T \in \mathbf{R}^p$. In linear regression, the response $y_i \in \mathbf{R}$ is continuous and the goal is to estimate a linear function $f(\mathbf{x}) = \hat{b} + \mathbf{x}^T\hat{\mathbf{w}}$ to predict y using regressor $\mathbf{x}$. For binary linear classification with the class label $y \in \{-1, 1\}$, we want to find a classification function $f(\mathbf{x}) = \hat{b} + \mathbf{x}^T\hat{\mathbf{w}}$ and use sign(f) as the classification rule. For simplicity, we focus on linear classification and regression. However, one can extend the idea to the nonlinear case via basis expansion.

Under the regularization framework, the objective function of both penalized regression and classification can be expressed as:

$$\min_f \ \frac{1}{n}\sum_{i=1}^n l(f(\mathbf{x}_i), y_i) + \lambda J(f), \tag{1}$$

where $J(f)$ is a regularization term serving as the roughness penalty of f, l is the loss function, and $\lambda > 0$ is a tuning parameter which controls the trade-off between the data fit measured by l and the complexity of f in terms of $J(f)$. For example, bridge regression uses $J(f) = L_q(\mathbf{w})$. Least squares regression (LSR) and least absolute deviation regression (LADR) use the square loss $l(f(\mathbf{x}_i), y_i) = (y_i - f(\mathbf{x}_i))^2$ and the absolute loss $l(f(\mathbf{x}_i), y_i) = |y_i - f(\mathbf{x}_i)|$ respectively. The classification method SVM uses the hinge loss $l(f(\mathbf{x}_i), y_i) = [1 - y_i f(\mathbf{x}_i)]_+$, where $[u]_+ = u$ if $u \geq 0$ and 0 otherwise.

An alternative formulation of the regularization problem can be written as

$$\min_f \ \frac{1}{n}\sum_{i=1}^{n} l(f(\mathbf{x}_i), y_i) \qquad \text{subject to } J(f) \leq t;\ t > 0. \tag{2}$$

Problem (2) is closely related to problem (1). In fact, using the Lagrangian theory, one can show equivalence of the two problems with a one-to-one correspondence between λ and t if both l and $J(f)$ are convex (Boyd and Vandenberghe, 2004).

Our focus here is on the choice of penalty function $J(f)$. For a graphical visualization, we first consider the one-dimensional case, i.e., $p = 1$. Figure 1 plots penalty functions L_1 and L_2 in the top-left panel and the L_0 penalty in the top-right panel. Notice that, unlike L_2, both L_0 and L_1 are singular at the origin, which is a necessary condition for performing variable selection (Fan and Li, 2001). Another observation is that L_1 penalizes coefficients close to 0 more than L_2 does. As a result, it can perform variable selection but L_2 cannot. Compared to the L_0 penalty, L_1 places large penalties on coefficients far away from 0 thus producing estimation bias. Since L_0 directly penalizes the number of nonzero coefficients, it can be a better choice for the purpose of variable selection.

There are two drawbacks to the use of the L_0 penalty. One clear drawback is that the computation involved in (1) is difficult due to the discontinuity and nonconvexity of L_0 at the origin. Another disadvantage of L_0 is due to its scale invariance. The L_0 penalized procedure cannot distinguish between two solutions of the same size that yield identical total losses measured by the first term in (1). Such identifiability problems become more severe for the L_0 SVM in the completely separable case. In that situation, there are infinitely many candidates having total loss 0 and L_0 cannot identify the classifier with maximum separation.

To overcome the disadvantages of the L_0 and L_1 penalties, we propose to combine the two penalties. The new combined penalty is expected to retain the strengths of both the L_0 and L_1

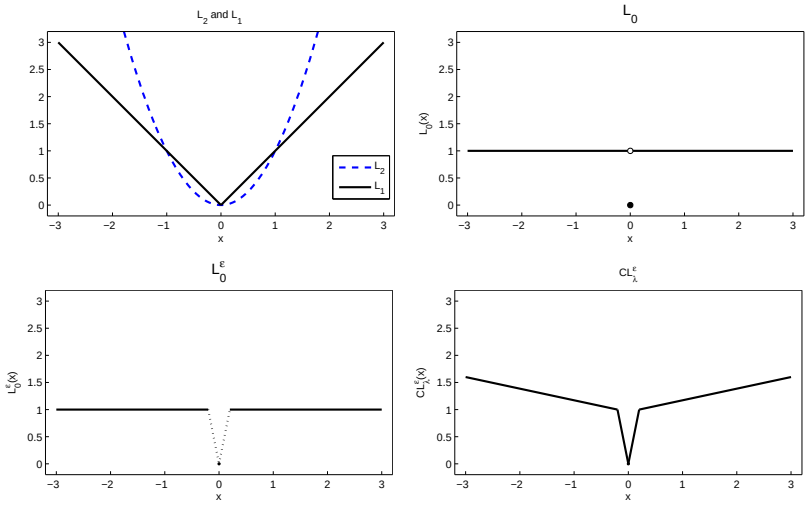

Figure 1: Plot of the L_2, L_1, L_0, L_0^ϵ, and CL_γ^ϵ penalties with $\epsilon = 0.2$, $\gamma = 0.3$, and $p = 1$.

penalties.

3 A New Combined Penalty

Discontinuity at the origin for L_0 makes the implementation of the corresponding optimization problem difficult. To make the computation easier, we consider a continuous approximation to L_0. Notice that $L_0(\theta) = 0$ if $\theta = 0$ and 1 otherwise. We approximate $L_0(\theta)$ by $L_0^\epsilon(\theta)$, defined to be 1 if $|\theta| \geq \epsilon$, and $|\theta|/\epsilon$ otherwise. Clearly, the limit of this approximation is $L_0(\theta)$ itself as $\epsilon \to 0$. Consequently, when $\epsilon > 0$ is small enough, $L_0^\epsilon(\theta)$ is a good approximation to $L_0(\theta)$ (see the bottom-left panel of Figure 1). For $p > 1$, we approximate $L_0(\mathbf{w}) = \sum_{j=1}^p L_0(w_j)$ by $L_0^\epsilon(\mathbf{w}) = \sum_{j=1}^p L_0^\epsilon(w_j)$.

To deal with the possible instability of L_0 due to its scale invariance, we now consider a combined penalty, denoted as CL_γ^ϵ, by combining the L_0^ϵ and L_1 penalties as follows:

$$CL_\gamma^\epsilon(\mathbf{w}) = (1-\gamma)L_0^\epsilon(\mathbf{w}) + \gamma L_1(\mathbf{w}), \tag{3}$$

where $0 \leq \gamma \leq 1$ is a weighting parameter. Clearly CL_γ^ϵ is a more general penalty, with

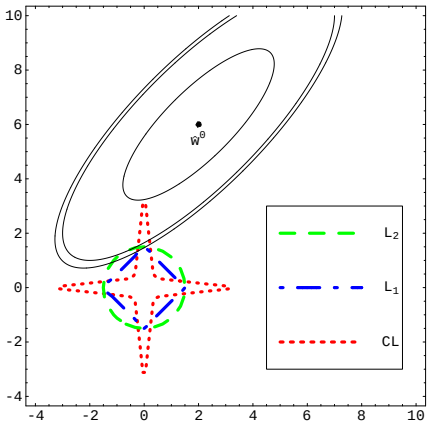

Figure 2: Estimation of the LSR problem (5) with different penalties satisfying $J(\mathbf{x}^T\boldsymbol{\phi}) \leq 1.5$. The dashed, dot-dashed, and dotted curves represent the restriction boundaries of L_2, L_1, and $CL_{0.2}^{0.5}$ respectively. The solid elliptical curves centered at the OLS estimator $\hat{\mathbf{w}}^0$ are the contours of $h(\boldsymbol{\phi})$ at three different levels.

$CL_1^\epsilon = L_1$ and $CL_0^\epsilon = L_0^\epsilon$ as special cases. In this section, we study properties of CL_γ^ϵ in the penalized LSR setting.

Consider a linear regression model $Y_i = \mathbf{x}_i^T\mathbf{w} + \varepsilon_i$, where the ε_i's are i.i.d. random variables with mean 0 and variance σ^2; $i = 1, \ldots, n$. Without loss of generality, we assume that both the covariates and the response are centered to have mean 0, thus the intercept term is 0. Under the regularization framework (1), we estimate $\mathbf{w}$ by $\hat{\mathbf{w}}_n = \text{argmin}_{\boldsymbol{\phi}} Z_n(\boldsymbol{\phi})$ with

$$Z_n(\boldsymbol{\phi}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\phi})^2 + \frac{\lambda_n}{n}\sum_{j=1}^{p} CL_\gamma^\epsilon(\phi_j), \tag{4}$$

where $\lambda_n > 0$ is a tuning parameter and plays the same role as that of $n\lambda$ in (1).

3.1 Geometry of the New Penalty

To illustrate CL_γ^ϵ, we first study its properties under (2) from a geometric point of view. Denote $\mathbf{X}$ as the $n \times p$ design matrix, and $\mathbf{y} = (y_1, y_2, \cdots, y_n)^T$. Then the first term in (4) becomes the quadratic function $h(\boldsymbol{\phi}) = \frac{1}{n}(\boldsymbol{\phi} - \hat{\mathbf{w}}^0)^T\mathbf{X}^T\mathbf{X}(\boldsymbol{\phi} - \hat{\mathbf{w}}^0) + \text{constant}$, where $\hat{\mathbf{w}}^0 = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ is

the OLS estimator. As a result, (2) becomes

$$\min_{\boldsymbol{\phi}} h(\boldsymbol{\phi}) \text{ subject to } J(\mathbf{x}^T\boldsymbol{\phi}) \leq t. \tag{5}$$

Figure 2 gives a graphical demonstration of (5) for $p = 2$. The solid elliptical curves (centered at the OLS estimates) are the contours of the function $h(\boldsymbol{\phi})$, and the dotted, dashed, and dot-dashed lines represent the boundaries of $J(\mathbf{x}^T\boldsymbol{\phi}) \leq 1.5$ with $J(\mathbf{x}^T\boldsymbol{\phi}) = CL_\gamma^\epsilon(\boldsymbol{\phi})$, $L_2(\boldsymbol{\phi})$, and $L_1(\boldsymbol{\phi})$ respectively. The solution of (5) corresponds to the place that the elliptical contours first touch the feasible regions constrained by the penalty function. Notice that the feasible regions $J(\mathbf{x}^T\boldsymbol{\phi}) \leq t$ of both CL_γ^ϵ and L_1 have corners. Hence, the first hit often happens at these corners, and as a result, the solution has a zero coefficient. From the graph, we can easily see that the feasible region of CL_γ^ϵ has sharper corners than that of L_1, and thus CL_γ^ϵ can perform variable selection more easily. For the L_2 penalty, the feasible region is a circle which does not have any corners, hence illustrating its lack of variable selection.

3.2 Some Asymptotic Properties

To study the limiting behavior of (4), we assume the following regularity conditions for the design matrix. In particular, $A_n = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^T \to A$, where A is a nonnegative definite matrix and $\frac{1}{n}\max_{1\leq i\leq n}\mathbf{x}_i^T\mathbf{x}_i \to 0$. Then consistency of $\hat{\mathbf{w}}_n$ can be established by the following theorem, provided that $\lambda_n = o(n)$.

Theorem 1 *If A is nonsingular and $\lambda_n/n \to \lambda_0 \geq 0$, then $\hat{\boldsymbol{w}}_n \to_p$ argmin(Z), where*

$$Z(\boldsymbol{\phi}) = (\boldsymbol{\phi} - \boldsymbol{w})^T A(\boldsymbol{\phi} - \boldsymbol{w}) + \lambda_0 \sum_{j=1}^p CL_\gamma^\epsilon(\phi_j).$$

Thus if $\lambda_n = o(n)$, argmin$(Z) = \boldsymbol{w}$ and so $\hat{\boldsymbol{w}}_n$ is consistent.

Theorem 1 suggests that $\lambda_n = o(n)$ is sufficient for consistency, but λ_n needs to grow more slowly for $\sqrt{n}$-consistency of $\hat{\mathbf{w}}_n$ as shown in the following theorem.

Theorem 2 *If $\lambda_n/n^{1/2} \to \lambda_0 \geq 0$ and A is nonsingular, then*

$$\sqrt{n}(\hat{\boldsymbol{w}}_n - \boldsymbol{w}) \to_d argmin(V),$$

where $V(\boldsymbol{u}) = -2\boldsymbol{u}^T \boldsymbol{W} + \boldsymbol{u}^T A\boldsymbol{u} + \gamma \left\{ \lambda_0 \sum_{j=1}^p \left[u_j sign(w_j) I(w_j \neq 0) + |u_j| \, I(w_j = 0) \right] \right\} +$
$(1-\gamma) \left\{ \frac{\lambda_0}{\epsilon} \sum_{j=1}^p \left[|u_j| \, I(w_j = 0) + u_j sign(w_j) I(0 < |w_j| < \epsilon) + \frac{sign(w_j)u_j - |u_j|}{2} I(|w_j| = \epsilon) \right] \right\}$ *and* $\boldsymbol{W}$ *has a* $N(\boldsymbol{0}, \sigma^2 A)$ *distribution.*

Theorem 2 is analogous to the asymptotic results for the bridge regression by Knight and Fu (2000). If $\lambda_0 = 0$, then $\text{argmin}(V) = A^{-1}\mathbf{W}$ and $\hat{\mathbf{w}}_n$ is asymptotically unbiased. For $\lambda_0 > 0$, the last two summation terms of $V(\mathbf{u})$ correspond to the L_1 and L_0^ϵ components of CL_γ^ϵ respectively. When $|w_j| > \epsilon$; $j = 1, \cdots, p$, $V(\mathbf{u}) = -2\mathbf{u}^T\mathbf{W} + \mathbf{u}^T A\mathbf{u} + \gamma[\lambda_0 \sum_{j=1}^p u_j \text{sign}(w_j) I(w_j \neq 0)]$, and thus w_j is estimated with asymptotic bias. This bias comes from the L_1 component with the weight γ. Hence, if we compare two procedures using the new penalty with different weights, the procedure with the smaller γ produces less estimation bias than the other one. Asymptotically, when the true $w_j = 0$, the estimator will be exactly zero with high probability due to the large penalty put on the term $|u_j| I(w_j = 0)$ in $V(\mathbf{u})$. In fact, the shrinkage term from L_0^ϵ is much bigger than that from L_1, and consequently better shrinkage towards zero by CL_γ^ϵ can be expected.

4 Solving the New Penalty through MIP

With the new penalty and $f(\mathbf{x}) = b + \mathbf{x}^T\mathbf{w}$, (1) becomes

$$\min_{\mathbf{w},b} \ \frac{1}{n} \sum_{i=1}^n l(b + \mathbf{x}_i^T \mathbf{w}, y_i) + \lambda \sum_{j=1}^p CL_\gamma^\epsilon(w_j). \tag{6}$$

The computation of (6) is rather challenging since CL_γ^ϵ is nonconvex. In this section, we utilize the piecewise linearity of CL_γ^ϵ and convert problem (6) into a MIP problem. To this end, we first introduce MIP in Section 4.1 and then propose a solution to (6) in Section 4.2.

4.1 MIP

A linear (quadratic) program is the optimization of a linear (quadratic) objective function subject to linear constraints. When some variables are restricted to take only integer values, it becomes a MIP problem which can be further classified as mixed-integer linear programming (MILP) or mixed-integer quadratic programming (MIQP). Explicitly, a MILP problem of n variables $\mathbf{z} = (z_1, z_2, \cdots, z_n)^T$ can be formulated as follows:

$$\min_{\mathbf{z}} \sum_{j \in I \cup U} a_j z_j, \tag{7}$$

subject to $D\mathbf{z} \leq \mathbf{b}$, $\mathbf{l} \leq \mathbf{z} \leq \mathbf{u}$, $z_j \in \mathbf{R}$ for $j \in I$, $z_j \in \mathbf{Z}$ for $j \in U$, and $I \cup U = \{1, 2, \cdots, n\}$. In this formulation, D, an $m \times n$ matrix, and $\mathbf{b}$, a vector of length m, are used to specify linear equality and inequality constraints on $\mathbf{z}$, and the a_j's are fixed constants. Vectors $\mathbf{l}$ and $\mathbf{u}$ are of length n and used as the lower and upper bounds of $\mathbf{z}$. Sets U and I are the index sets of continuous and integer variables z_i's. If the index set I is empty, (7) reduces to a linear programming problem.

MIP, as an active research area in operations research, has been studied for many years. Its applications include airline scheduling, production planning, survivability of telecommunication networks and many others. There exist numerous methods to solve MIP, among which the most commonly used is the branch and bound algorithm. The essence of this algorithm is to solve the relaxed problem by dropping the integer constraint(s). If the relaxed problem is infeasible, the original MIP problem is infeasible as well. If the solution coincidentally satisfies the integer constraint(s), then the optimal solution is found. Otherwise one needs to branch, recursively solve the resulting subproblems, and improve the lower bound of the objective by pruning out certain parts of the feasible solution spaces. The algorithm stops when the optimal solution attaining the lower bound is obtained. More details on MIP can be found in Garfinkel and

Nemhauser (1972) and Nemhauser and Wolsey (1999). For statistical applications, Liu and Wu (2006) proposed to use MIP to solve ψ-learning for classification problems.

The complexity of MIP can be affected by many factors including the size of the problem, the numerical characteristics of the data, the algorithm adopted, and the computational hardware. When it is impractical to compute the optimal solution, one can settle for a good (but not necessarily optimal) one by setting a restriction on the computing time. A detailed list of software for MIP can be found at http://www-fp.mcs.anl.gov/otc/Guide/SoftwareGuide/. In this article, all examples were computed using either the commercial optimization software CPLEX with the AMPL interface (Fourer et al., 2003) or a free C package GLPK of GNU available at http://www.gnu.org/software/glpk/.

4.2 Converting CL_γ^ϵ Penalized Regression/SVM into MIP

The piecewise linear function CL_γ^ϵ defined via

$$CL_\gamma^\epsilon(\theta) = (1-\gamma)L_0^\epsilon(\theta) + \gamma\,|\theta| = \begin{cases} (1-\gamma) + \gamma\,|\theta| & \text{when } |\theta| \geq \epsilon; \\ ((1-\gamma)/\epsilon + \gamma)\,|\theta| & \text{when } |\theta| < \epsilon, \end{cases} \tag{8}$$

can be reformulated as follows. Let M be chosen large enough such that $\theta \in [-M, M]$ for all feasible values of θ. Partition the interval $[-M, M]$ into four subintervals $[-M, -\epsilon]$, $[-\epsilon, 0]$, $[0, \epsilon]$, and $[\epsilon, M]$ with binary variables δ_2, δ_3, and δ_4 to specify the subinterval containing θ. Construct the corresponding four variables $\theta_1 \in [0, M-\epsilon]$, $\theta_2 \in [0, \epsilon]$, $\theta_3 \in [0, \epsilon]$, and $\theta_4 \in [0, M-\epsilon]$ and set $\theta = -M + \theta_1 + \theta_2 + \theta_3 + \theta_4$. Then one can show that

$$CL_\gamma^\epsilon(\theta) = (1-\gamma)(1 - \frac{\theta_2}{\epsilon} + \frac{\theta_3}{\epsilon}) + \gamma(M - \theta_1 - \theta_2 + \theta_3 + \theta_4), \tag{9}$$

subject to the following constraints:

$$\begin{aligned}
\delta_k &\in \{0,\,1\}, && \text{for } k\text{=2,3,4,} && (10)\\
\delta_{k+1} &\leq \delta_k, && \text{for } k\text{=2,3,} && (11)\\
0 &\leq \theta_k, && \text{for } k\text{=1,2,3,4,} && (12)\\
\theta_1 &\leq (M-\epsilon), && && (13)\\
\theta_k &\leq \epsilon\delta_k, && \text{for } k\text{=2,3,} && (14)\\
\theta_4 &\leq (M-\epsilon)\delta_4, && && (15)\\
-\theta_1 &\leq (M-\epsilon)(1-2\delta_2), && && (16)\\
-\theta_k &\leq \epsilon(1-2\delta_{k+1}), && \text{for } k\text{=2,3,} && (17)\\
\theta &= -M+\theta_1+\theta_2+\theta_3+\theta_4. && && (18)
\end{aligned}$$

To prove (9), we consider the four possible cases satisfying constraints (10) and (11):

- Case 1 ($\delta_2=\delta_3=\delta_4=0$): Constraints (14) and (15) imply $\theta_2=\theta_3=\theta_4=0$. While (12), (13) and (16) imply $\theta_1\in[0,M-\epsilon]$. These together with (18) yield $\theta=-M+\theta_1\in[-M,-\epsilon]$ and $CL^{\epsilon}_{\gamma}(\theta)=(1-\gamma)+\gamma(M-\theta_1)=(1-\gamma)+\gamma\,|\theta|$.

- Case 2 ($\delta_2=1,\delta_3=\delta_4=0$): Following the constraints gives $\theta_1=M-\epsilon$, $\theta_2\in[0,\epsilon]$, and $\theta_3=\theta_4=0$. Hence $\theta=-\epsilon+\theta_2\in[-\epsilon,0]$ and $CL^{\epsilon}_{\gamma}(\theta)=(1-\gamma)(1-\theta_2/\epsilon)+\gamma(M-(M-\epsilon)-\theta_2)=(1-\gamma)\,|\theta|\,/\epsilon+\gamma\,|\theta|$.

- Case 3 ($\delta_2=\delta_3=1,\delta_4=0$): This corresponds to $\theta_1=M-\epsilon$, $\theta_2=\epsilon$, $\theta_3\in[0,\epsilon]$, and $\theta_4=0$. So $\theta=\theta_3\in[0,\epsilon]$ and $CL^{\epsilon}_{\gamma}(\theta)=(1-\gamma)\theta_3/\epsilon+\gamma\theta_3=(1-\gamma)\,|\theta|\,/\epsilon+\gamma\,|\theta|$.

- Case 4 ($\delta_2=\delta_3=\delta_4=1$): This implies $\theta_1=M-\epsilon$, $\theta_2=\epsilon$, $\theta_3=\epsilon$, and $\theta_4\in[0,M-\epsilon]$. We then have $\theta=\epsilon+\theta_4\in[\epsilon,M]$ and $CL_{\gamma}=(1-\gamma)+\gamma(\epsilon+\theta_4)=(1-\gamma)+\gamma\,|\theta|$.

Hence, the equivalence given by (9) is satisfied.

After plugging $J(f) = CL^{\epsilon}_{\gamma}(\mathbf{w})$ into (1) and using the above reformulation of CL^{ϵ}_{γ}, we can get the following mixed integer optimization problem:

$$\min_{\theta_{\mathbf{1}},\theta_{\mathbf{2}},\theta_{\mathbf{3}},\theta_{\mathbf{4}},\mathbf{w},b} \frac{1}{n}\sum_{i=1}^{n} l(b+\mathbf{x}_i^T\mathbf{w}, y_i) + \lambda \sum_{j=1}^{p}\left[(1-\gamma)(1-\frac{\theta_2^j}{\epsilon}+\frac{\theta_3^j}{\epsilon}) + \gamma(M-\theta_1^j-\theta_2^j+\theta_3^j+\theta_4^j)\right], \tag{19}$$

subject to p sets of constraints (10) -(18), with the j-th set corresponding to w_j.

Overall there are $3p$ integer variables embedded in (19). It is known that introducing too many integer variables in optimization may affect the computing speed of MIP considerably. Fortunately, we can take advantage of the symmetry of CL^{ϵ}_{γ} to reduce the number of integer variables from $3p$ to p. Specifically, $CL^{\epsilon}_{\gamma}(\theta)$ can also be represented as

$$CL^{\epsilon}_{\gamma}(\theta) = (1-\gamma)\frac{\xi_1}{\epsilon} + \gamma\xi,$$

subject to the following constraints

$$\theta \le \xi, \quad -\theta \le \xi, \qquad \xi = \xi_1 + \xi_2, \quad \delta \in \{0,1\}, \tag{20}$$

$$-\xi_1 \le \epsilon(1-2\delta), \quad 0 \le \xi_1 \le \epsilon, \qquad 0 \le \xi_2 \le (M-\epsilon)\delta. \tag{21}$$

Then (19) becomes

$$\min_{\xi_{\mathbf{1}},\xi_{\mathbf{2}}\mathbf{w},b} \frac{1}{n}\sum_{i=1}^{n} l(b+\mathbf{x}_i^T\mathbf{w}, y_i) + \lambda \sum_{j=1}^{p}\left[(1-\gamma)\frac{\xi_1^j}{\epsilon} + \gamma\xi^j\right], \tag{22}$$

subject to p sets of constraints (20)-(21) corresponding to the p components of $\mathbf{w}$. Problem (22) becomes either MILP or MIQP, depending on the choice of the loss function l.

5 Simulation Study

In this section, we examine the performance of the proposed CL^{ϵ}_{γ} penalty (in terms of variable selection as well as prediction accuracy) via simulations. Eight different penalties $CL^{\epsilon}_{0}(= L^{\epsilon}_{0})$,

$CL_{0.01}^{\epsilon}$, $CL_{0.1}^{\epsilon}$, $CL_{0.3}^{\epsilon}$, $CL_{0.5}^{\epsilon}$, $CL_{0.8}^{\epsilon}$, $CL_{1}^{\epsilon}(= L_1)$, and L_2 are selected for comparison. We examine the results of SVM in Sections 5.1 and 5.2, and LADR and LSR in Section 5.3. Tuning sets (with the same sizes as training data sets) are generated for all examples. The tuning parameter λ is selected via a discrete search with 40 candidates in the tuning set. Testing/prediction errors are estimated via independent testing sets of size 20,000, generated in the same manner as the corresponding training sets. In all numerical examples, parameters M and ϵ, as defined in Section 3, are set to be $1,000$ and 0.01 respectively. Our numerical experience suggests that these choices work well for most problems.

Table 1: Simulation results of Example 5.1.1 based on 100 replications.

		No. of	Avg. No. of 0 coef.		
Penalty	Testing Error	Correct models*	Correct	Incorrect	avg. solve time
L_0^{ϵ}	0.0109 (0.0067)	96	7.96	0.00	0.4123
$CL_{0.01}^{\epsilon}$	0.0061 (0.0016)	96	7.96	0.00	0.6156
$CL_{0.1}^{\epsilon}$	0.0054 (0.0010)	77	7.76	0.00	0.3668
$CL_{0.3}^{\epsilon}$	0.0053 (0.0010)	86	7.85	0.00	0.2534
$CL_{0.5}^{\epsilon}$	0.0053 (0.0008)	82	7.78	0.00	0.1954
$CL_{0.8}^{\epsilon}$	0.0055 (0.0010)	83	7.71	0.00	0.1252
L_1	0.0072 (0.0020)	81	7.30	0.00	0.0512
L_2	0.0083 (0.0028)	0	0.00	0.00	0.0892

* A correct model here means a model only uses x_1 and x_2.

Table 2: Simulation results of Example 5.1.2 based on 100 replications.

		No. of	Avg. No. of 0 coef.		
Penalty	Testing Error	Correct model*	Correct	Incorrect	avg. solve time
L_0^{ϵ}	0.0998 (0.0048)	56	6.71	0.00	0.5141
$CL_{0.01}^{\epsilon}$	0.0992 (0.0048)	65	7.00	0.00	0.5332
$CL_{0.1}^{\epsilon}$	0.0983 (0.0045)	65	7.19	0.00	0.5074
$CL_{0.3}^{\epsilon}$	0.0980 (0.0040)	51	7.09	0.00	0.4517
$CL_{0.5}^{\epsilon}$	0.0979 (0.0034)	46	6.87	0.00	0.4075
$CL_{0.8}^{\epsilon}$	0.0984 (0.0036)	43	6.80	0.00	0.2805
L_1	0.1001 (0.0042)	22	4.78	0.00	0.0578
L_2	0.1085 (0.0075)	0	0.00	0.00	0.1101

* A correct model here means a model only uses x_1 and x_2.

5.1 Linear SVM

For classification problems, we generate 100 random samples of size 200 with ten-dimensional variables $(x_1, x_2, \cdots, x_{10})^T \in \mathbf{R}^{10}$. Binary classification is considered with response $y \in \{-1, 1\}$. The data are generated as follows. First, we obtain class label y with $P(Y = 1) = 0.5$. Then given y, $(x_1, x_2)^T$ is generated from a two-dimensional mixed normal distribution $0.5N((1.5y, 0)^T, \sigma^2 I_2) + 0.5N((0, 1.5y)^T, \sigma^2 I_2)$, where I_2 is a 2×2 identity matrix. Finally, the remaining eight variables $x_j; j = 3, 4, \cdots, 10$, are generated independently from $N(0, 9\sigma^2)$. Notice that among these ten dimensional variables, the first two are important variables while the others are random noise variables.

Example 5.1.1. The parameter σ is set to be 0.4. In this case, most training samples are linearly separable. The testing error, the number of correct models selected, as well as the average number of correct/incorrect 0 coefficients over 100 replications are summarized in Table 1. The numbers in parentheses are the standard errors of the corresponding estimates. The last column reports the average solving time (in seconds).

Example 5.1.2. We set σ to be 0.8. In contrast to Example 5.1.1, most training samples are linearly nonseparable in this case. The corresponding results are given in Table 2.

According to Tables 1 and 2, we can see that L_0^ϵ and $CL_{0.01}^\epsilon$ give much better results in terms of variable selection than L_1 does. Among all procedures, the correct nonzero estimates have the same signs as those in the Bayes classifier $\hat{y} = \text{sign}(x_1 + x_2)$. In terms of testing errors, the CL_γ^ϵ SVMs give smaller errors in both examples for $\gamma \in (0, 1)$. We note that the L_0^ϵ SVM has much larger testing error than the rest in Example 5.1.1 since L_0^ϵ cannot identify the solution with maximum separation (as discussed in Section 3). The CL_γ^ϵ SVMs with $\gamma > 0$, in contrast to the L_0^ϵ SVM, do not have such difficulty. In terms of computational time, the L_1 SVM is

Table 3: Simulation results of Example 5.2 based on 100 replications.

Penalty	Testing Error	No. of correct model*	Avg. No. of 0 coef. Correct	Incorrect	avg. solve time
L_0^ϵ	0.0323 (0.0091)	87	8.63	0.01	1.0122
$CL_{0.01}^\epsilon$	0.0318 (0.0072)	80	8.63	0.00	0.9698
$CL_{0.1}^\epsilon$	0.0325 (0.0070)	86	8.76	0.00	0.8221
$CL_{0.3}^\epsilon$	0.0364 (0.0072)	59	8.15	0.00	0.6924
$CL_{0.5}^\epsilon$	0.0395 (0.0091)	46	7.58	0.01	0.6307
$CL_{0.8}^\epsilon$	0.0443 (0.0088)	18	5.77	0.03	0.5768
L_1	0.0470 (0.0096)	0	2.80	0.01	0.0810
L_2	0.0537 (0.0099)	0	0.00	0.00	0.1033

* A correct model here means a model only uses x_1, x_1^3, and x_2.

the fastest one to compute. As expected, the CL_γ^ϵ SVMs take longer to compute especially for smaller γ's, but they are still quite reasonable. Overall, the CL_γ^ϵ SVMs with small positive γ's have the best performance in terms of both classification accuracy and variable selection.

5.2 Nonlinear SVM via Basis Expansion

In this example, we examine the performance of the new method in nonlinear classification via basis pursuit. To this end, we generate 100 random samples of size 200 with input $\mathbf{x} = (x_1, x_2, x_3, x_4, x_1^2, x_2^2, x_3^2, x_4^2, x_1^3, x_2^3, x_3^3, x_4^3)^T$ where $x_i \sim$ Uniform[-2,2]; $i = 1, 2, 3, 4$. Let $f(\mathbf{x}) = x_1 - x_1^3 - x_2$. For each input $\mathbf{x}$, the corresponding label y is generated according to the following two steps. First, set $y = 1$ if $f(\mathbf{x}) > 0$ and $y = -1$ if $f(\mathbf{x}) \leq 0$. Then, for those inputs with $yf(\mathbf{x}) < 0.5$, we change y to $-y$ with probability 0.1 to make the training sample nonseparable. From the data generation mechanism, we can easily see that there are only three important variables (x_1, x_1^3 and x_2), and the remaining nine variables are noise.

We calculate the classifiers using all twelve variables. The results are summarized in Table 3. From this table, we can see that the L_0^ϵ, $CL_{0.01}^\epsilon$, and $CL_{0.1}^\epsilon$ SVMs select the correct models much more frequently and have smaller testing errors than the other SVMs. In addition, similar to the previous examples, the L_2 SVM fails to perform variable selection since it does not have

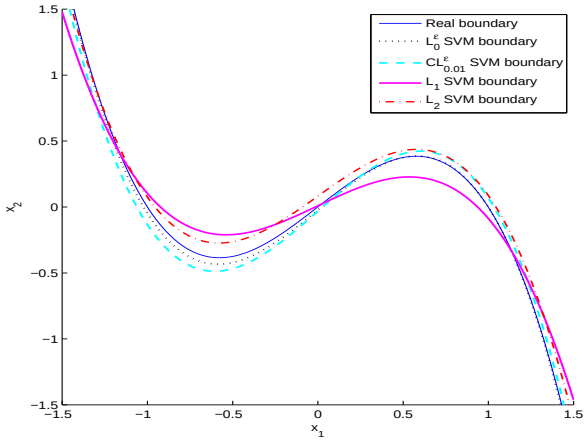

Figure 3: Projected boundaries of one particular random sample in Example 5.2

any zero coefficients. For a visual illustration, we plot the classification boundaries of the L_0^ϵ, $CL_{0.01}^\epsilon$, L_1, and L_2 SVM classifiers from one particular sample projected on the plane spanned by x_1 and x_2 in Figure 3. The Bayes decision boundary is also plotted for comparison. From the plot, we can conclude that the boundaries yielded by the $CL_{0.01}^\epsilon$ and L_0^ϵ SVMs are much closer to the Bayes boundary.

Table 4: Simulation results for Example 5.3.1 based on 100 replications.

		No. of	Avg. No. of 0 coef.		
Penalty	Mean absolute error	correct model*	Correct	Incorrect	avg. solve time
L_0^ϵ	1.6238 (0.0234)	58	4.03	0.00	3.1338
$CL_{0.01}^\epsilon$	1.6239 (0.0234)	58	4.02	0.00	3.0458
$CL_{0.1}^\epsilon$	1.6217 (0.0226)	56	4.08	0.00	2.6462
$CL_{0.3}^\epsilon$	1.6209 (0.0210)	59	4.17	0.00	2.2990
$CL_{0.5}^\epsilon$	1.6220 (0.0213)	55	4.10	0.00	2.0878
$CL_{0.8}^\epsilon$	1.6253 (0.0220)	57	3.99	0.00	1.8703
L_1	1.6353 (0.0255)	3	1.97	0.00	0.2512
L_2	1.6586 (0.0335)	0	0.00	0.00	0.2256

* A correct model here means a model only uses x_1, x_2, and x_5.

5.3 Regression

To study the effects of different penalties on regression, we generate 100 datasets consisting of 200 observations from the linear model $Y = \mathbf{x}^T\mathbf{w} + 2\varepsilon$, where $\mathbf{w} = (3, 1.5, 0, 0, 2, 0, 0, 0)^T$. Each

Table 5: Simulation results for Example 5.3.2 based on 100 replications.

Penalty	Mean square error	No. of correct model*	Avg. No. of 0 coef.		avg. solve time
			Correct	Incorrect	
L_0^ϵ	4.0884 (0.0821)	49	4.10	0.00	0.5874
$CL_{0.01}^\epsilon$	4.0880 (0.0820)	43	4.03	0.00	0.5932
$CL_{0.1}^\epsilon$	4.0869 (0.0787)	42	4.04	0.00	0.5875
$CL_{0.3}^\epsilon$	4.0858 (0.0796)	45	4.01	0.00	0.5697
$CL_{0.5}^\epsilon$	4.0872 (0.0799)	44	3.98	0.00	0.5566
$CL_{0.8}^\epsilon$	4.0976 (0.0828)	54	3.99	0.00	0.5289
L_1	4.1448 (0.0981)	2	1.70	0.00	0.1374
L_2	4.2324 (0.1486)	0	0.00	0.00	0.1385

* A correct model here means a model only uses x_1, x_2, and x_5.

component of $\mathbf{x}$ is from $N(0,1)$ with the correlation between x_i and x_j being $0.5^{|i-j|}$, and ε is from $N(0,1)$ and is independent of $\mathbf{x}$. This model was previously used in Tibshirani (1996).

Example 5.3.1. We fit the CL_γ^ϵ LADR with the absolute loss function for $\gamma = 0$, 0.01, 0.1, 0.3, 0.5, 0.8, and 1 as well as the L_2 LADR. Results are summarized in Table 4. The second column reports the predictive mean absolute error and its corresponding standard deviation in parenthesis, and the last column gives the average solving time (in seconds).

Example 5.3.2. LSR with square loss function and eight different penalties are fitted on the same data as in Example 5.3.1. Table 5 displays the corresponding simulation results.

Tables 4 and 5 indicate that in these regression problems, the CL_γ^ϵ penalties with $\gamma < 1$ outperform the L_1 and L_2 penalties both in terms of the prediction errors and variable selection. For computational times, the CL_γ^ϵ penalties with $\gamma < 1$ take a little longer to compute than the L_1 and L_2 penalties. Among all procedures, the correct nonzero estimates have the same signs as those of the true coefficients.

6 Application

The Ionosphere data collected by a system in Goose Bay, Labrador is considered. The complete dataset of size 351 can be obtained at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/ionosphere.

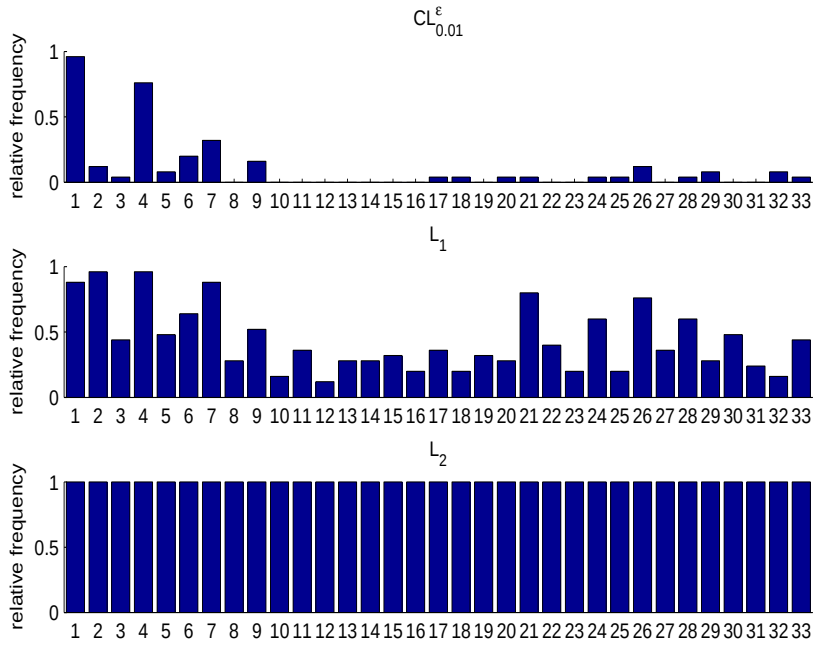

Figure 4: Plot of proportions of the 33 variables selected corresponding to the $CL^{\epsilon}_{0.01}$, L_1, and L_2 SVMs based on 25 random training samples.

In this dataset, the targets are free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not and their signals pass through the ionosphere (Sigillito et al., 1989). There are 34 continuous predictor variables and the response attribute is either "good" or "bad" radar returns. We exclude the second predictor variable since it takes a constant value 0 for all the observations and we use the remaining 33 to predict the response attribute.

To explore the performance of different procedures, we randomly divide the dataset into three sets of equal sizes for training, tuning, and testing. We repeat this procedure 25 times; the results of variable selection by the $CL^{\epsilon}_{0.01}$, L_1, and L_2 SVMs are plotted in Figure 4. As shown in the plot, the $CL^{\epsilon}_{0.01}$ SVM delivers the sparsest model with the 1st and 4th variables frequently selected. The L_1 SVM performs effective variable selection but the result is not as

sparse as the $CL^{\epsilon}_{0.01}$ SVM. Due to the incapability of the L_2 penalty in variable selection, the variable selecting proportions are all 1's for the L_2 SVM. Furthermore, the average testing errors of the three methods, the $CL^{\epsilon}_{0.01}$, L_1, and L_2 SVMs, are 0.1412, 0.1456, and 0.1525 respectively (with corresponding standard errors are 0.0296, 0.0269, and 0.0323). Therefore, the $CL^{\epsilon}_{0.01}$ SVM uses the sparest model to deliver the best prediction accuracy.

7 Discussion

In this article, we have discussed variable selection using a new combined penalty for both SVM classification and linear regression. A new global algorithm using MIP from operations research to solve the corresponding difficult nonconvex optimization problem is introduced. Simulations in Section 5 and real applications in Section 6 have shown significant improvements of the new penalty over the L_1 penalty. We hope this research helps to connect different research areas between statistics and operations research. Moreover, we believe our algorithm can be useful for further studies of the relationships among different penalties.

We have shown that the new penalty CL^{ϵ}_{γ} with $\gamma \in (0, 1)$ can perform better variable selection than the L_1 penalty and produce more stable models than the L_0 penalty. Ideally, one can treat γ as a tuning parameter but the computation involved can be very intensive. In practice, we suggest to select a relatively small γ, for example 0.01. Our numerical examples indicate that such a choice indeed gives very good performance.

Although our numerical results show the advantage of the CL^{ϵ}_{γ} penalty over the L_1 penalty in terms of variable selection, solving a regularization problem with the L_1 penalty is more efficient. We have demonstrated in the paper that our proposed algorithm using MIP for the CL^{ϵ}_{γ} penalty can solve problems with moderate sizes effectively. However, it can be slow for large-scale problems. Since the CL^{ϵ}_{γ} penalty can be decomposed as the difference of two convex

functions, the difference convex algorithm (DCA) (Liu et al., 2005) may be proven useful here. More investigations and simplifications are needed.

Appendix

Proof of Theorem 1: To prove the theorem, we need to show that

$$\sum_{\boldsymbol{\phi}\in K} \left| Z_n(\boldsymbol{\phi}) - Z(\boldsymbol{\phi}) - \sigma^2 \right| \to_p 0 \tag{23}$$

for any compact set K and that

$$\hat{\mathbf{w}}_n = O_p(1). \tag{24}$$

Then under (23) and (24), we have $\text{argmin}(Z_n) \to_p \text{argmin}(Z)$. (23) is easy to obtain by the pointwise convergence in probability of $Z_n(\boldsymbol{\phi})$. To prove (24), note that $Z_n(\boldsymbol{\phi}) \geq \frac{1}{n}\sum_{i=1}^n (Y_i - \mathbf{x}_i^T\boldsymbol{\phi})^2 = Z_n^{(0)}(\boldsymbol{\phi})$ for all $\boldsymbol{\phi}$. Since $\text{argmin}(Z_n^{(0)}) = O_p(1)$, it follows that $\text{argmin}(Z_n) = O_p(1)$.

Lemma 1 *The global minimizer of* $g(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T A\boldsymbol{u} + \boldsymbol{W}^T\boldsymbol{u} - \sum_{i=1}^p d_i |u_i|$ *is almost surely unique, where* $\boldsymbol{u}$ *is a vector of length* p*,* A *is a positive definite* $p\times p$ *matrix,* $\boldsymbol{W}$ *is a vector of* p *continuous random variables, and* $d_i > 0$*;* $i = 1, 2, \cdots, p$*, are positive constants.*

The lemma can be proved using the quadratic properties of $g(\mathbf{u})$ and details of the proof are omitted.

Proof of Theorem 2: The proof is similar to that of Knight and Fu (2000). Define

$$V_n(\mathbf{u}) = \sum_{i=1}^n \left[(\varepsilon_i - \mathbf{u}^T\mathbf{x}_i/\sqrt{n})^2 - \varepsilon_i^2\right] + \lambda_n \sum_{j=1}^p \left[CL_\gamma(w_j + u_j/\sqrt{n}) - CL_\gamma(w_j)\right],$$

where $\mathbf{u} = (u_1, \cdots, u_p)^T$ and note that V_n is minimized at $\sqrt{n}(\hat{\mathbf{w}}_n - \mathbf{w})$. First note that $\sum_{i=1}^n \left[(\varepsilon_i - \mathbf{u}^T\mathbf{x}_i/\sqrt{n})^2 - \varepsilon_i^2\right] \to_d -2\mathbf{u}^T\mathbf{W} + \mathbf{u}^T A\mathbf{u}$ with finite-dimensional convergence holding

trivially. For sufficiently large n, we have

$$
\begin{aligned}
& \lambda_n(CL_\gamma(w_j + u_j/\sqrt{n}) - CL_\gamma(w_j)) \\
= \;& (1-\gamma)\lambda_n(L_0^\epsilon(w_j + u_j/\sqrt{n}) - L_0^\epsilon(w_j)) + \gamma\lambda_n(L_1(w_j + u_j/\sqrt{n}) - L_1(w_j)) \\
\to \;& \frac{1-\gamma}{\epsilon}\lambda_0\{|u_j|\, I(w_j = 0) + u_j \mathrm{sign}(w_j) I(0 < |w_j| < \epsilon) + \mathrm{sign}(w_j) u_j I(w_j u_j < 0) I(|w_j| = \epsilon)\} \\
& + \gamma\lambda_0\{u_j \mathrm{sign}(w_j) I(w_j \neq 0) + |u_j|\, I(w_j = 0)\} \\
= \;& \frac{1-\gamma}{\epsilon}\lambda_0\{|u_j|\, I(w_j = 0) + u_j \mathrm{sign}(w_j) I(0 < |w_j| < \epsilon) + \frac{\mathrm{sign}(w_j)u_j - |u_j|}{2} I(|w_j| = \epsilon)\} \\
& + \gamma\lambda_0\{u_j \mathrm{sign}(w_j) I(w_j \neq 0) + |u_j|\, I(w_j = 0)\} \qquad (25) \\
= \;& \lambda_0 f(\gamma, \epsilon, u_j, w_j),
\end{aligned}
$$

where $f(\gamma, \epsilon, u_j, w_j) = \gamma\{u_j \mathrm{sign}(w_j) I(w_j \neq 0) + |u_j|\, I(w_j = 0)\} + \frac{1-\gamma}{\epsilon}\{|u_j|\, I(w_j = 0) + u_j \mathrm{sign}(w_j) I(0 < |w_j| < \epsilon) + \frac{\mathrm{sign}(w_j)u_j - |u_j|}{2} I(|w_j| = \epsilon)\}$. Hence, we have $V_n(\,\cdot\,) \to_d V(\,\cdot\,)$.

To prove that $\mathrm{argmin}(V_n) \to_d \mathrm{argmin}(V)$, it suffices to show that $\mathrm{argmin}(V)$ is unique with probability one and $\mathrm{argmin}(V_n) = O_p(1)$ [Kim and Pollard (1990)]. Using (25), for all $\mathbf{u}$ and sufficiently large n, there exists a δ such that

$$
\begin{aligned}
V_n(\mathbf{u}) &= \sum_{i=1}^{n} \left[(\varepsilon_i - \mathbf{u}^T \mathbf{x}_i/\sqrt{n})^2 - \varepsilon_i^2\right] + \lambda_n \sum_{j=1}^{p} \left[CL_\gamma(w_j + u_j/\sqrt{n}) - CL_\gamma(w_j)\right] \\
&\geq \sum_{i=1}^{n} \left[(\varepsilon_i - \mathbf{u}^T \mathbf{x}_i/\sqrt{n})^2 - \varepsilon_i^2\right] - (\lambda_0 + \delta) \sum_{j=1}^{p} |f(\gamma, \epsilon, u_j, w_j)| = V_n^{(l)}(\mathbf{u}).
\end{aligned}
$$

Since u_j appears in $f(\gamma, \epsilon, u_j, w_j)$ with order of at most one, which grows slower than the quadratic terms in $V_n^{(l)}$, we have $\mathrm{argmin}(V_n^{(l)}) = O_p(1)$. Hence $\mathrm{argmin}(V_n) = O_p(1)$.

What remains is to show that $\mathrm{argmin}(V)$ is unique with probability one. Notice that every term in $f(\gamma, \epsilon, u_i, w_j)$ is convex except $-\frac{(1-\gamma)|u_j|}{2\epsilon} I(|w_j| = \epsilon)$. The uniqueness of the global minimizer of $V(\,\cdot\,)$ can then be established by Lemma 1. The desired result then follows.

Acknowledgements

The authors would like to thank Howard Bondell, Edward Carlstein, J. S. Marron, Gabor Pataki, Scott Provan, and Jon W. Tolle for their helpful discussions and comments. The authors are indebted to the editor, the associate editor, and two referees, whose helpful comments and suggestions led to a much improved presentation. Yufeng Liu's research was partially supported by the National Science Foundation Grant DMS-0606577, the UNC Junior Faculty Development Award, and the UNC University Research Council Small Grant Program.

References

Boyd, S. and Vandenberghe, L. (2004), "Convex optimization". *Cambridge University Press*.

Bradley, P. and Mangasarian, O. (1998), "Feature selection via concave minimization and support vector machines", In J Shavlik(eds), *ICML'98*. Morgan Kaufmann.

Cristianini, N. and Shawe-Taylor, J. (1999), "An introduction to support vector machines and other kernel-based learning methods". Cambridge University Press.

Donoho, D. L. and Johnstone, I. (1994), "Ideal spatial adaptation by wavelet shrinkage", *Biometrika*, **81**, 425-455.

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G., and Picard, D. (1995), "Wavelet shrinkage; asymptopia?", *JRSS*, Ser. B, **35**, 109-148.

Fan, J. and Li, R. (2001), "Variable selection via nonconcave penalized likelihood and its oracle properties", *JASA*, **96**, 456, 1348-1360.

Frank, I. E. and Friedman, J. H. (1993), "An statistical view of some chemometrics regression tools", *Technometrics*, **35**, 109-135.

Fourer, R., Gay, D.M., and Kernighan, B.W. (2003), "AMPL: A Modeling Language for Math-

ematical Programming". *Duxbury Press.*

Garfinkel, R. S and Nemhauser, G. L. (1972), "Integer programming", *Wiley.*

Hoerl, A. E. and Kennard, R. W. (1970). "Ridge regression: biased estimation for nonorthogonal problems", *Technometrics*, **12**, 55-67.

Kim, J. and Pollard, D. (1990), "Cube root asymptotics", *Ann. Stat.*, **18**, 191-219.

Knight, K. and Fu, W. J. (2000), "Asymptotics for lasso-type estimators", *Ann. Stat.*, **28**(5), 1356-1378.

Liu, Y., Shen, X., and Doss, H. (2005). Multicategory ψ-learning and support vector machine: computational tools. *Journal of Computational and Graphical Statistics,* **14**(1): 219-236.

Liu, Y. and Wu, Y. (2006). Optimizing ψ-learning via mixed integer programming. *Statistica Sinica*, **16**, 2, 441-457.

Nemhauser, G.L. and Wolsey, L.A. (1999), "Integer and combinatorial optimization", *Wiley.*

Sigillito, V. G., Wing, S. P., Hutton, L. V., and Baker, K. B. (1989), "Classification of radar returns from the ionosphere using neural networks", *Johns Hopkins APL Technical Digest*, **10**, 262-266.

Tibshirani, R. J. (1996), "Regression Shrinkage and Selection via the Lasso", *JRSS*, Ser. B, **58**, 267-288.

Vapnik, V. (1998), "Statistical learning theory", *Wiley.*

Zhu, J., Hastie, T., Rosset, S., and Tibshirani, R. (2003), "1-norm support vector machines", *Neural Information Processing Systems*, **16.**