Adaptive regularization using the entire solution surface

BY S. WU, X. SHEN AND C. J. GEYER

School of Statistics, University of Minnesota, 313 Ford Hall, 224 Church Street S. E., Minneapolis, Minnesota 55455, U.S.A.

swu@stat.umn.edu  xshen@stat.umn.edu  charlie@stat.umn.edu

SUMMARY

Several sparseness penalties have been suggested for delivery of good predictive performance in automatic variable selection within the framework of regularization. All assume that the true model is sparse. We propose a penalty, a convex combination of the \( L_1 \)- and \( L_\infty \)-norms, that adapts to a variety of situations including sparseness and nonsparseness, grouping and nongrouping. The proposed penalty performs grouping and adaptive regularization. In addition, we introduce a novel homotopy algorithm utilizing subgradients for developing regularization solution surfaces involving multiple regularizers. This permits efficient computation and adaptive tuning. Numerical experiments are conducted using simulation. In simulated and real examples, the proposed penalty compares well against popular alternatives.

Some key words: Homotopy; Lasso; \( L_1 \)-norm; \( L_\infty \)-norm; Subgradient; Support vector machine; Variable grouping and selection.

1. INTRODUCTION

There has been a great interest in various sparseness penalties for high-dimensional analysis, particularly the \( L_1 \)-penalty, a feature of which is that it permits automatic variable selection, especially when the number of candidate variables greatly exceeds the sample size. This, however, requires the true model be sparse, which is difficult if not impossible to verify in many situations. To seek high-dimensional structures leading to high predictive performance, we introduce a new penalty that adapts to a variety of situations including sparseness and nonsparseness, grouping and nongrouping. This new penalty, together with our new solution surface algorithm, yields adaptive regularization.

The \( L_2 \)-penalty has been used in regression (Hoerl & Kennard, 1970) and support vector machines (Vapnik, 1995). The \( L_1 \)-penalty has been used in least-squares regression (Tibshirani, 1996), support vector machines (Bradley & Mangasarian, 1998) and sparse overcomplete representations (Donoho et al., 2006). The elastic net penalty, a convex combination of the \( L_1 \)- and \( L_2 \)-penalties, encourages grouping of highly correlated predictors (Zou & Hastie, 2005; Wang et al., 2006). Other relevant penalties include simultaneous variable selection and clustering (Bondell & Reich, 2008; Liu & Wu, 2007; Wang & Zhu, 2008), the Dantzig selector (Candès & Tao, 2007; Bickel et al., 2008) and sure independence screening (Fan & Lv, 2008).

To deliver high predictive performance, especially in a high-dimensional situation, we propose a new penalty, a convex combination of the \( L_1 \)- and the \( L_\infty \)-penalties. This penalty not only enables automatic variable selection but also seeks grouping among predictors to enhance predictive performance. Furthermore, it permits efficient computation.
For high-dimensional data analysis, efficient computation requires realizing high predictive performance through tuning. Towards this end, we develop a subdifferential-based homotopy method for computing entire solution surfaces, in addition to subgradient surfaces. This approach differs dramatically from the existing Kuhn–Tucker method; see §3 for details.

For model selection, we propose a model selection criterion for prediction error in least-squares regression, where the designs can be random or fixed. The criterion can be expressed in terms of a covariance penalty plus a correction term taking into account extrapolation from random designs.

2. Methodology

2.1. Regularization

Consider a problem of estimating \( f \) based on a random sample \((x_i, y_i), \ldots, (x_n, y_n)\), where \( x_i = (x_{i1}, \ldots, x_{ip}) \) is a \( p \)-dimensional vector and \( y_i \) is a scalar. We estimate \( f \) by minimizing the empirical loss \( \sum_{i=1}^{n} l(y_i, f(x_i)) \) over a candidate function class \( f \in \mathcal{F} \).

Regularization is often employed to prevent overfitting in estimating \( f \). To regularize parameters, penalties are added to the loss \( l(\cdot, \cdot) \) for various purposes. Thus the regularized loss is

\[
\sum_{i=1}^{n} l(y_i, f(x_i)) + \lambda^T J(f),
\]

where \( \lambda \) is a vector of nonnegative tuning parameters, \( J(f) \) is a vector of penalties regularizing \( f \) and \( ^T \) denotes the transpose. The framework (1) covers least-squares regression with \( l(y_i, f(x_i)) = (y_i - f(x_i))^2 \) and support vector machine classification with \( l(y_i, f(x_i)) = |1 - y_i f(x_i)|_+ \), where \( x_+ \) denotes the nonnegative part of \( x \).

The penalty function \( J(f) \) is designed to achieve specific objectives. In variable selection, when \( f(x) = \beta^T h(x) \), where \( h(x) \) is a vector-valued basis function, \( J(f) \) can be chosen to be the \( L_1 \)-norm \( \|\beta\|_1 = \sum_{j=1}^{p} |\beta_j| \). Then (1) leads to the lasso (Tibshirani, 1996) when \( l(\cdot, \cdot) \) is the least-squares loss. The advantage of using the \( L_1 \)-penalty in (1) is that it performs automatic variable selection by yielding a sparse solution of (1): the \( L_1 \)-penalty can control a model’s complexity effectively even when the dimension of \( \beta \) greatly exceeds the sample size, cf. Wang & Shen (2007) for classification. When \( J(f) \) is chosen to be the \( L_2 \)-penalty, (1) shrinks \( \beta \) by grouping \( \beta_j s \) corresponding to highly correlated variables; unfortunately, it does not produce a sparse model; cf. Zou & Hastie (2005) for regression and Wang et al. (2006) for classification.

To seek a sparse model, Zou & Hastie (2005) propose the elastic net penalty \( J(f) \), a convex combination of \( L_1 \) - and \( L_2 \) - penalties, which combines the advantages of both. Despite its performance, the elastic net penalty has two aspects requiring further attention. First, grouping predictors alone in regression or classification may not deliver good predictive performance in variable selection when the response is ignored, because correlations among predictors can be irrelevant to outcomes of the response. See the example in §6. Second, this penalty does not permit an efficient algorithm through a piecewise linear regularization solution surface.

In a recent paper, Bondell & Reich (2008) propose a method using the \( L_1 \) and pairwise \( L_\infty \) - penalties for variable selection and grouping. However, efficient solution algorithms do not exist due to difficulty in treating overcomplete representation of the penalty.

2.2. The adaptive penalty

To enhance predictive performance of the method of regularization, we propose a new penalty to achieve three goals. First, the penalty is adaptive in that it adapts to a variety of situations
including both sparse or nonsparse situations. Second, it seeks grouping among predictors in variable selection, for better predictive accuracy. Third, it permits efficient computation through a homotopy approach (Allgower & Georg, 2003).

The proposed penalty, which we call the $L_1 L_\infty$-penalty, has the form in the case $f(x) = \beta^T h(x)$,

$$J(f) = J(\beta) = (1 - \alpha)\|\beta\|_1 + \alpha\|\beta\|_\infty$$

where the $L_\infty$-norm is $\|\beta\|_\infty = \max_{1 \leq j \leq p} |\beta_j|$. This penalty retains the advantages of the two extreme penalties within the class $L_p$ ($1 \leq p \leq \infty$). The $L_1$-penalty is sparse and thresholds some coefficients at zero, whereas the $L_\infty$-penalty is nonsparse and groups highly correlated predictors that are relevant to the response. To adapt to the degree of sparseness, $\alpha$ can be tuned. $L_1 L_\infty$ appears to be more adaptive than other penalties mentioned above. Most importantly, (2) is piecewise linear in $\beta$. This is critical for efficient computation through the method of homotopy; see §3 for details. In contrast, the elastic net penalty is quadratic in $\beta$, which explains its computational limitation discussed above.

Placing (2) into (1), we obtain our regularized loss $\sum_{i=1}^n l[y_i, \beta^T h(x_i)] + \tau [(1 - \alpha)\|\beta\|_1 + \alpha\|\beta\|_\infty]$, equivalently

$$\sum_{j=1}^n l[y_j, \beta^T h(x_j)] + \lambda_1\|\beta\|_1 + \lambda_\infty\|\beta\|_\infty,$$

where $\lambda_1 = \tau (1 - \alpha)$ and $\lambda_\infty = \tau \alpha$ are nonnegative tuning parameters.

3. REGULARIZATION SOLUTION SURFACES
3.1. General algorithm

The minimizer of (3) $\hat{\beta} \equiv \hat{\beta}_\lambda$, if exists, is a function of $\lambda = (\lambda_1, \lambda_\infty)^T$. Consequently, computing an entire solution surface $\lambda \mapsto \hat{\beta}_\lambda$ is critical for adaptive tuning.

In the literature, there exist several solution path algorithms computing an entire solution path for one-dimensional $\lambda$, cf. Efron et al. (2004), Rosset & Zhu (2007) and Park & Hastie (2007) for regression, and Zhu et al. (2003), Hastie et al. (2004) and Wang et al. (2006) for classification. These algorithms use the Kuhn–Tucker conditions for tracking the piecewise linear solution path in the one-dimensional case. Unfortunately, there does not seem to exist such an algorithm in higher dimensions, partly because of difficulty of applying the Kuhn–Tucker conditions when many slack variables are involved.

To compute a high-dimensional solution surface, we use the subdifferential approach (Rockafellar & Wets, 2003), a systematic method of handling nonsmooth functions, which has several advantages. First, the properties of nonsmooth functions at nondifferentiable points are completely characterized by subgradients. Second, subgradients can be tracked as well as $\hat{\beta}_\lambda$, facilitating computation in multi-dimensional situations.

First we describe a general algorithm for a convex objective function $G_\lambda(\beta)$, and then apply it to two specific situations.

A subgradient of $G_\lambda$ at $\beta$ is any vector $b \in \mathbb{R}^p$ satisfying $G_\lambda(\gamma) \geq G_\lambda(\beta) + b^T(\gamma - \beta)$, for all $\gamma \in \mathbb{R}^p$, and the subdifferential of $G_\lambda$ at $\beta$, denoted by $\partial G_\lambda(\beta)$, is the set of all such $b$. The subdifferential of a convex function is a nonempty, convex, compact set, and $0 \in \partial G_\lambda(\hat{\beta}_\lambda)$ is a necessary and sufficient condition that $\hat{\beta}_\lambda$ is a global minimizer of $G_\lambda$ (Rockafellar & Wets, 2003, pp. 308–11).

We give two examples, which will be used in our algorithms. The subgradient of $\beta \mapsto \|\beta\|_1$ at $\hat{\beta}_\lambda$ is a vector $b^1_\lambda$ whose components satisfy $b^1_{\lambda,j} = \text{sign}(\hat{\beta}_{\lambda,j})$ if $\hat{\beta}_{\lambda,j} \neq 0$ and $-1 \leq b^1_{\lambda,j} \leq 1$
otherwise, where \( \text{sign}(x) = 1 \) if \( x > 0 \), equals 0 if \( x = 0 \) and equals \(-1\) otherwise. The subgradient of \( \beta \mapsto \|\beta\|_\infty \) (Rockafellar, 1970, p. 215) at \( \hat{\beta}_\lambda \neq 0 \) is a vector \( b^\infty_\lambda \) whose components satisfy (a)

\[
\sum_{j=1}^{p} |b^\infty_{\lambda,j}| = 1; 
\]

(b) \( b^\infty_{\lambda,j} = 0 \) when \( |\hat{\beta}_{\lambda,j}| < \|\hat{\beta}_\lambda\|_\infty \); (c) \( |b^\infty_{\lambda,j}| \leq 1 \); and (d) \( \text{sign}(b^\infty_{\lambda,j}) \text{sign}(\hat{\beta}_{\lambda,j}) \geq 0 \). The subgradient of \( \beta \mapsto \|\beta\|_\infty \) at \( \hat{\beta}_\lambda = 0 \) is a vector \( b^\infty_\lambda \) whose components satisfy \( \sum_{j=1}^{p} |b^\infty_{\lambda,j}| \leq 1 \).

We encapsulate the case splitting in these characterizations by partitioning the indices for each into strongly active, weakly active and inactive sets. An index \( j \) is strongly active for the \( L_1 \) term at \( \lambda \) if \( |\hat{\beta}_{\lambda,j}| = 0 \) and \( |b^1_{\lambda,j}| < 1 \) while an index \( j \) is weakly active for the \( L_1 \) term at \( \lambda \) if \( |\hat{\beta}_{\lambda,j}| = 0 \) and \( |b^1_{\lambda,j}| = 1 \). An index \( j \) is strongly active for the \( L_\infty \) term at \( \lambda \) if \( |\hat{\beta}_{\lambda,j}| = \|\hat{\beta}_\lambda\|_\infty \) and \( b^\infty_{\lambda,j} \neq 0 \) while an index \( j \) is weakly active for the \( L_\infty \) term at \( \lambda \) if \( |\hat{\beta}_{\lambda,j}| = \|\hat{\beta}_\lambda\|_\infty \) and \( b^\infty_{\lambda,j} = 0 \). An index \( j \) is inactive if it is not one of these active cases.

For the least-squares loss, we can write the optimality condition \( 0 \in \partial G_\lambda(\hat{\beta}_\lambda) \) as

\[
-2 \sum_{i=1}^{n} x_{ij}(y_i - x_{ij}^T \hat{\beta}_\lambda) + \lambda_1 b^1_{\lambda,j} + \lambda_\infty b^\infty_{\lambda,j} = 0 \quad (j = 1, \ldots, p). 
\]

Knowledge of the index sets determines certain components of \( \hat{\beta}_\lambda, b^1_\lambda \) and \( b^\infty_\lambda \) and allows (5) to determine the rest. If the result satisfies the characterization of the subgradients and index sets, then this is the solution. Otherwise we must check with other index sets again until we identify the correct index sets at \( \lambda \).

Since this check process is slow, we use more properties of \( \hat{\beta}_\lambda, b^1_\lambda \) and \( b^\infty_\lambda \). Each is piecewise linear in a known smooth function of \( \lambda \); see Theorem 1 below. Moreover, the change in the index sets at a transition point where the slope of some piecewise linear function changes is usually regular: one index goes from strongly active to weakly active to inactive or vice versa as \( \lambda \) goes through the transition point. In this regular case, no check process is necessary. If multiple indices are weakly active simultaneously, then the check process is necessary. More specifically we let one weakly active index be either strongly active or inactive, and repeat this process with other weakly active indices until we obtain the new index sets satisfying (5). Thus it is easy to compute the solution surface proceeding along straight lines in \( \lambda \) space.

The optimality condition becomes more complicated for the support vector machine because the support vector machine hinge loss, \( l(y_i, f(x_i)) = \{1 - y_i(\sum_{j=1}^{p} x_{ij}(\hat{\beta}_j + \beta_0))\}_+ \), is nonsmooth and also needs subgradients and index sets. Then (5) is replaced by (6) below, but the general principles are the same. The solutions and subgradients are piecewise constant in \( \lambda \) or piecewise linear in a known smooth function \( \lambda \); see Theorem 2 below. When the index sets are fixed, (6) determines solutions and subgradients. The behaviour of subgradients at transition points is usually regular, so checking index sets is unnecessary when the solution surface is followed along straight lines in \( \lambda \) space.

Following the above discussion, to compute \( \lambda \mapsto \hat{\beta}_\lambda \), we specify a set of evaluation points at which \( \hat{\beta}_\lambda \) will be computed. Starting from any evaluation point \( \lambda \) and evaluating \( \hat{\beta}_\lambda \) at all evaluation points by moving along straight lines from one evaluation point to another yields the entire solution surface.

This leads to a new homotopy algorithm.
Algorithm 1.

_Step 1._ At an initial $\lambda = \lambda^0$, compute $\hat{\beta}_\lambda$, which initializes the strongly active, weakly active and inactive sets.

_Step 2._ At a current point, either evaluation or transition, compute the directional derivative of $\hat{\beta}_\lambda$, $b^1_\lambda$ and $b^\infty_\lambda$ and in the support vector machine the subgradient of the loss function, along a line towards the next evaluation point. If the current point is a transition point, then the check process is applied. Determine the next transition point along the direction.

_Step 3._ At the current point, if no transition occurs before reaching the evaluation point, then extrapolate linearly to compute the value of $\hat{\beta}_\lambda$ at the evaluation point from the current point. Update the current point by the evaluation point when no transition occurs, otherwise, by the next transition point and update the corresponding index sets. Terminate if $\hat{\beta}_\lambda$ has been computed at all evaluation points, otherwise, go to Step 2.

Differ $\lambda$ can produce the same solution; in particular, for sufficiently large $\lambda_1$ or $\lambda_\infty$, the solution is $\hat{\beta}_\lambda = 0$. By construction, Algorithm 1 identifies the unique global $\hat{\beta}_\lambda$; see §3.2 for the specification of evaluation points.

3.2. Least-squares regression

In least-squares regression, as $\lambda$ varies, $\hat{\beta}_\lambda$, $b^1_\lambda$ and $b^\infty_\lambda$ change to satisfy (5), resulting in the piecewise linearity of $\hat{\beta}_\lambda$, $b^1_\lambda$ and $b^\infty_\lambda$.

**Theorem 1.** Suppose $\lambda$ is not a transition point. Then the solution $\hat{\beta}_\lambda$ is piecewise linear in $\lambda$ while $b^1_\lambda$ and $b^\infty_\lambda$ are piecewise linear in $(\lambda_1/\lambda_\infty, 1/\lambda_\infty)^T$ and $(\lambda_\infty/\lambda_1, 1/\lambda_1)^T$, respectively.

Let $S^1_\lambda$ and $W^1_\lambda$ denote the strongly and weakly active sets for $\|\hat{\beta}_\lambda\|_1$, similarly $S^\infty_\lambda$ and $W^\infty_\lambda$ for $\|\hat{\beta}_\lambda\|_\infty$, and $I_\lambda$ the inactive set. Also, let $A^\infty_\lambda = S^\infty_\lambda \cup W^\infty_\lambda$ and $A^1_\lambda = S^1_\lambda \cup W^1_\lambda$.

As an initial evaluation point, we take $\lambda^0 = (\lambda^0_1, 0)^T$ where $\lambda^0_1 = 2 \max_j |\sum_{i=1}^n x_i j y_i|$. In (5), if $\lambda_1 > \lambda^0_1$ and $\lambda_\infty = 0$, then $|b^1_{\lambda, j}| < 1$ for all $j$, hence $\hat{\beta}_\lambda = 0$. Thus if $\lambda_1 = \lambda^0_1$ and $\lambda_\infty = 0$, then a coefficient $\hat{\beta}_{\lambda, j}$ becomes nonzero as $|b^1_{\lambda, j}| = 1$. Indeed $|\hat{\beta}_{\lambda, j}^0| = \|\hat{\beta}_{\lambda, j}\|_\infty$ and $|b^\infty_{\lambda, j}| = 1$ because $\hat{\beta}_{\lambda, j}^0$ is the only nonzero variable. Consequently, initial index sets become $A^\infty_0 = \{j\}$, $I_0 = \emptyset$ and $A^1_0 = (A^\infty_0)^c$, where $A^c$ denotes the complement of $A$.

To specify other evaluation points, we compute $\lambda^1_1$ and $\lambda^0_\infty = 2 \sum_{j=1}^p \sum_{i=1}^n |x_i j y_i|$ where $\hat{\beta}_{(\lambda^1_1, 0)^T} = 0$ for $\lambda_1 > \lambda^1_1$ and $\hat{\beta}_{(0, \lambda^0_\infty)^T} = 0$ for $\lambda_\infty > \lambda^0_\infty$, respectively. In the $\lambda$-plane, then, we locate a set of evaluation points that are uniformly distributed in the rectangle with four corners $(0, 0)$, $(\lambda^1_1, 0)$, $(\lambda^0_1, \lambda^0_\infty)$ and $(0, \lambda^0_\infty)$. Starting from $\lambda = (\lambda^0_1, 0)^T$, $\hat{\beta}_\lambda$ can be evaluated through the moving process: (a) move $\hat{\beta}_\lambda$ along the $\lambda_1$ axis by decreasing $\lambda_1$ until reaching the $\lambda_\infty$ axis; (b) move $\hat{\beta}_\lambda$ along the $\lambda_\infty$ axis by increasing $\lambda_\infty$ to the next evaluation point; (c) move $\hat{\beta}_\lambda$ parallel to the $\lambda_1$ axis by increasing $\lambda_1$ until reaching $\lambda_\lambda$ where $\hat{\beta}_\lambda = 0$; (d) move $\hat{\beta}_\lambda$ to the nearest evaluation point with $\lambda_1 < \lambda_1$ and $\lambda_\infty > \lambda_\infty$; and (e) iterate (a)–(d) to reach $(0, \lambda^0_\infty)$.

Given index sets, directional derivatives of $\hat{\beta}_\lambda$, $b^1_\lambda$ and $b^\infty_\lambda$ are obtained by solving the derivatives with respect to $\lambda$ of (4) and (A1)–(A3). In moving along a direction, a transition occurs when one of the following events occurs: (i) an index $j$ in $I_\lambda$ moves to $W^\infty_\lambda$ when $|\hat{\beta}_{\lambda, j}|$ becomes $\|\hat{\beta}_\lambda\|_\infty$ retaining $|b^\infty_{\lambda, j}| = 0$; (ii) an index $j$ in $S^\infty_\lambda$ moves to $W^\infty_\lambda$ when $|b^\infty_{\lambda, j}|$ becomes 0 retaining $|\hat{\beta}_{\lambda, j}| = \|\hat{\beta}_\lambda\|_\infty$; (iii) an index $j$ in $I_\lambda$ moves to $W^1_\lambda$ when $|\hat{\beta}_{\lambda, j}|$ becomes 0 retaining $|b^1_{\lambda, j}| = 1$; or (iv) an index $j$ in $S^1_\lambda$ moves to $W^1_\lambda$ when $|b^1_{\lambda, j}|$ becomes 1 retaining $|\hat{\beta}_{\lambda, j}| = 0$.

Then Step 3 of Algorithm 1 is applied.
3.3. Classification

We now apply (2) to the hinge loss \( l(y, f(x)) = \{1 - y_i(\sum_{j=1}^{p} x_{ij}\beta_j + \beta_0)\}_+ \) with \( y_i \in \{1, -1\}. \)

In addition to the index sets for \( \|\beta\|_1 \) and \( \|\beta\|_\infty \), the index sets for the hinge loss are specified. Let \( z_i(\beta, \beta_0) = 1 - y_i(\sum_{j=1}^{p} x_{ij}\beta_j + \beta_0) \) and \( \alpha_{\lambda,i} \) denote subgradients at \( z_{\lambda,i} \equiv z_i(\hat{\beta}_\lambda, \hat{\beta}_\lambda, 0) \) of \( z_{\lambda,i} \mapsto (z_{\lambda,i})_+ \). Then \( \alpha_{\lambda,i} = 0 \) if \( z_{\lambda,i} < 0 \); \( = 1 \) if \( z_{\lambda,i} > 0 \); and \( \in [0, 1] \) otherwise \( (i = 1, \ldots, n) \).

From this characterization, the strongly active hinge set is \( S^H_\lambda = \{i : z_{\lambda,i} = 0 \text{ and } 0 < \alpha_{\lambda,i} < 1\} \).

The left weakly active hinge and right weakly active hinge sets are \( \mathcal{L}_\lambda = \{i : z_{\lambda,i} > 0 \text{ and } \alpha_{\lambda,i} = 1\} \) and \( \mathcal{R}_\lambda = \{i : z_{\lambda,i} < 0 \text{ and } \alpha_{\lambda,i} = 0\} \). Let \( \mathcal{H}_\lambda = S^H_\lambda \cup \mathcal{L}_\lambda \cup \mathcal{R}_\lambda \). The left and right inactive sets are defined as \( \mathcal{L}_\lambda = \{i : z_{\lambda,i} > 0 \text{ and } \alpha_{\lambda,i} = 1\} \) and \( \mathcal{R}_\lambda = \{i : z_{\lambda,i} < 0 \text{ and } \alpha_{\lambda,i} = 0\} \).

The optimality condition \( 0 \in \partial G_\lambda(\hat{\beta}_\lambda) \) can be written as

\[
- \sum_{i=1}^{n} \alpha_{\lambda,i} y_i x_{ij} + \lambda_1 b_{\lambda,j}^1 + \lambda_\infty b_{\lambda,j}^\infty = 0 \quad (j = 1, \ldots, p), \quad \sum_{i=1}^{n} \alpha_{\lambda,i} y_i = 0. \tag{6}
\]

As \( \lambda \) varies, \( \hat{\beta}_\lambda, b_{\lambda,j}^1, b_{\lambda,j}^\infty \) and \( \alpha_{\lambda} = (\alpha_{\lambda,1}, \ldots, \alpha_{\lambda,n})^T \) change to satisfy (6), leading to the following theorem.

**Theorem 2.** Suppose \( \lambda \) is not a transition point. Then the solutions \( \hat{\beta}_\lambda \) and \( \hat{\beta}_\lambda, 0 \) are piecewise constant in \( \lambda \). Furthermore, \( \alpha_{\lambda} \) is piecewise linear in \( \lambda \) while \( b_{\lambda,j}^\infty \) and \( b_{\lambda,j}^1 \) are piecewise linear in \( (\lambda_1/\lambda_\infty, 1/\lambda_\infty)^T \) and \( (\lambda_\infty/\lambda_1, 1/\lambda_1)^T \), respectively.

In this case, minimizing (3) for the hinge loss numerically at \( \lambda^0 \) yields an initial solution and its corresponding index sets.

Directional derivatives of \( b_{\lambda,j}^1, b_{\lambda,j}^\infty \) and \( \alpha_{\lambda} \) are obtained by solving the derivatives with respect to \( \lambda \) of (4) and (A4)–(A7).

However, (6) does not determine \( \hat{\beta}_\lambda \) and \( \hat{\beta}_\lambda, 0 \), so to track them, we use the hinge relationship

\[
\hat{\beta}_{\lambda,0} + \sum_{j \in \mathcal{I}_\lambda} x_{ij} \hat{\beta}_{\lambda,j} + \sum_{j \in \mathcal{A}_\lambda^\infty} x_{ij} \text{sign}(\hat{\beta}_{\lambda,j}) \|\hat{\beta}_\lambda\|_\infty = y_i \quad (i \in \mathcal{H}_\lambda).
\]

These are \( \text{card}(\mathcal{H}_\lambda) \) equations to be solved for \( \text{card}(\mathcal{I}_\lambda) + 2 \) unknowns, \( \beta_{\lambda,j} \ (j \in \mathcal{I}_\lambda), \hat{\beta}_\lambda, 0 \) and \( \|\hat{\beta}_\lambda\|_\infty \). To obtain \( \hat{\beta}_\lambda \) and \( \hat{\beta}_\lambda, 0 \), we apply the cardinality relationship \( \text{card}(\mathcal{H}_\lambda) = \text{card}(\mathcal{I}_\lambda) + 2 \) as explained in the proof of Theorem 2.

In moving along a homotopy direction, a transition occurs when one of the events (i)–(iv) in §3.2 occurs or one of the following events occurs: (i) an index \( i \) in \( \mathcal{L}_\lambda \) moves to \( \mathcal{W}_{\lambda}^{LH} \) when \( z_{\lambda,i} \) becomes 0 retaining \( \alpha_{\lambda,i} = 1 \); (ii) an index \( i \) in \( S^H_\lambda \) moves to \( \mathcal{W}_{\lambda}^{LH} \) when \( \alpha_{\lambda,i} \) becomes 1 retaining \( z_{\lambda,i} = 0 \); (iii) an index \( i \) in \( \mathcal{R}_\lambda \) moves to \( \mathcal{W}_{\lambda}^{RH} \) when \( \alpha_{\lambda,i} \) becomes 0 retaining \( z_{\lambda,i} = 0 \); or (iv) an index \( i \) in \( S^H_\lambda \) moves to \( \mathcal{W}_{\lambda}^{RH} \) when \( \alpha_{\lambda,i} \) becomes 0 retaining \( z_{\lambda,i} = 0 \).

Evaluation points are located as in the least-squares problem and then Step 3 of Algorithm 1 is applied.

4. Choice of Tuning Parameter

This section is devoted to model selection, particularly for selection of the optimal tuning parameter \( \lambda \). Specifically, we focus our attention on least-squares regression and binary classification, where the design points can be fixed or random. Our focus is on estimation of the prediction error and generalization error.
Consider a regression model \( Y_i = \mu(X_i) + e_i \) \( (i = 1, \ldots, n) \), with \( \mu(x) = x^T \beta \), where \( X_i \) follows an unknown distribution \( P \), and \( e_i \) is the random error with \( E(e_i) = 0 \) and \( \text{var}(e_i) = \sigma^2 \), and \( e_i \) is independent of \( X_i \), for all \( i \). Let \( \hat{\mu}_i(x) \) be an estimate of \( \mu_i(x) \) obtained from (5), based on the sample \( (X^n, Y^n) = (X_i, Y_i)_{i=1}^n \).

The performance of \( \hat{\mu}_\lambda \) is evaluated by the prediction error, \( \text{PE}(\hat{\mu}_\lambda) = E\{Y - \hat{\mu}_\lambda(X)\}^2 \), where the expectation \( E \) is taken over \( (X, Y) \). This prediction error measures predictive performance with respect to not only \( Y \) but also \( X \), which differs from the conventional conditional prediction error given \( X \). See Breiman & Spector (1992) for a detailed discussion of the difference between this \( \text{PE}(\hat{\mu}_\lambda) \) and the conditional prediction error.

To derive a model selection criterion, we apply an argument similar to that in Theorem 1 of Wang & Shen (2006a), to yield an approximately optimal unbiased estimator of \( \text{PE}(\hat{\mu}_\lambda) \) in the form of

\[
\text{OPE}(\hat{\mu}_\lambda) = n^{-1} \sum_{i=1}^n \{Y_i - \hat{\mu}_\lambda(X_i)\}^2 + 2n^{-1} \sum_{i=1}^n \text{cov}(Y_i, \hat{\mu}_\lambda(X_i) | X^n)
\]

\[+ D_{1n}(X^n, \hat{\mu}_\lambda) + D_{2n}(X^n),
\]

where

\[
D_{1n} = E\{E(Y | X) - \hat{\mu}_\lambda(X)^2\} - n^{-1} \sum_{i=1}^n \{E(Y_i | X_i) - \hat{\mu}_\lambda(X_i)^2 \mid X^n}\]

and

\[
D_{2n} = E\{\text{var}(Y | X)\} - n^{-1} \sum_{i=1}^n \text{var}(Y_i | X_i).
\]

For comparison, it suffices to use \( \text{OPE}(\hat{\mu}_\lambda) - D_{2n} \) because the term \( D_{2n} \) is independent of \( \hat{\mu}_\lambda \). This leads to our proposed model selection criterion, denoted by the generalized degrees of freedom:

\[
\text{GDF}(\hat{\mu}_\lambda) = n^{-1} \sum_{i=1}^n \{Y_i - \hat{\mu}_\lambda(X_i)\}^2 + 2n^{-1} \sum_{i=1}^n \text{cov}(Y_i, \hat{\mu}_\lambda(X_i) | X^n) + \hat{D}_{1n}(X^n, \hat{\mu}_\lambda).
\]

In the case of fixed design, \( \hat{D}_{1n} \equiv 0 \) and hence (7) reduces to the covariance penalty, denoted \( C_p(\hat{\mu}_\lambda) \).

To estimate \( \sum_{i=1}^n \text{cov}(Y_i, \hat{\mu}_\lambda(X_i) | X^n) \), we define the degrees of freedom \( \text{df}(\hat{\mu}_\lambda) \) for \( \hat{\mu}_\lambda \) to be \( \sum_{i=1}^n \text{cov}(Y_i, \hat{\mu}_\lambda(X_i) | X^n) / \sigma^2 \).

**Theorem 3.** For the \( L_1 \) \( L_\infty \) estimate, \( \text{df}(\hat{\mu}_\lambda) = E\{\text{card} (\hat{I}_\lambda)\} + 1 \), which is an unbiased estimate of \( \text{df}(\hat{\mu}_\lambda) \).

Therefore \( \sum_{i=1}^n \text{cov}(Y_i, \hat{\mu}_\lambda(X_i) | X^n) \) can be estimated through \( \hat{\text{df}}(\hat{\mu}_\lambda) \sigma^2 \). When \( \sigma^2 \) is unknown, an approximately unbiased estimate \( \hat{\sigma}^2 \) is used; see § 4 of Efron et al. (2004).

For \( \hat{D}_{1n} \), we apply the data perturbation technique in Shen & Huang (2006). First, we perturb \( X_i \) to generate pseudo-data \( X^*_i = X_i + \tau(X_i - \bar{X}_i) \) \( (i = 1, \ldots, n) \), where \( \bar{X}_i \) is sampled from its empirical distribution and \( 0 \leq \tau \leq 1 \) is the perturbation size. Second, we perturb \( Y_i \) to yield \( Y^*_i = Y_i + \tau(\bar{Y}_i - Y_i) \) \( (i = 1, \ldots, n) \) with \( \bar{Y}_i \sim N(Y_i, \sigma^2) \). The first term \( E\{E(Y | X) - \hat{\mu}_\lambda(X)^2\} \) in \( D_{1n} \) is estimated by \( n^{-1} \sum_{i=1}^n \{\hat{\mu}_\lambda(X_i) - \hat{\mu}^*_\lambda(X_i)\}^2 \), where \( \hat{\mu}^*_\lambda \) is estimated through \( \{X_i^*, Y_i^*, Y^*_i\}_{i=1}^n \), while the second term \( n^{-1} \sum_{i=1}^n \{E(Y_i | X_i) - \hat{\mu}_\lambda(X_i)^2 \mid X^n\} \) in \( D_{1n} \) is estimated by \( n^{-1} \sum_{i=1}^n \{\hat{\mu}_\lambda(X_i^*) - \hat{\mu}^*_\lambda(X_i^*)\}^2 \). Consequently, (7) is obtained by the perturbed data, and can be computed via Monte Carlo approximation as described in Wang & Shen (2006a).
In what follows, we fix $\tau = 0.5$ throughout; see Shen & Huang (2006) for a sensitivity study with regard to the choice of $\tau$.

In classification, a model selection criterion that is similar to (7) has been obtained in Wang & Shen (2006a) through a different data perturbation scheme. The reader may consult their paper for more details.

5. NUMERICAL STUDIES

5.1. Simulated examples

We now demonstrate the effectiveness of the proposed penalty and compare it against the elastic net, lasso and the $L_\infty$-penalty through simulated examples. In least-squares regression and binary classification, we examine the case of small $p$ and large $n$ and additionally consider the case of large $p$ and small $n$ where the number of candidate variables $p$ can greatly exceed the sample size $n$, which is of great current interest.

In each example, a training sample is generated together with an independent test sample. The solution $\hat{\beta}_p$ is computed on a training sample, and its predictive performance is evaluated on a test sample. To adaptively tune, we compute $\hat{\beta}_p$ through a regularization solution surface by applying Algorithm 1. For the $L_1 L_\infty$-penalty, we locate 200 evaluation points on the $\lambda$-plane as described in Section 3.2. For the $L_2$-component of the elastic net and the $L_2$-penalty, we choose a set of uniform grid points between $10^{-3}$ and $10^3$. For a fair comparison, the number of grid points for the $L_2$-component of the elastic net and $L_2$-penalty is fixed to be that of evaluation points on the $\lambda_\infty$ axis in the $L_1 L_\infty$-penalty. In each example, we compute the test error with $\hat{\beta}_p$ for the squared loss in regression and the 0–1 loss in classification by obtaining the minimizer $\hat{\beta}$ of the generalized degrees of freedom.

In least-squares regression, five simulated examples are examined. The first three examples are modified from those in Tibshirani (1996) and Zou & Hastie (2005), the fourth one is taken from Yuan & Lin (2006) and the last one considers the case of large $p$ and small $n$. In each example, a linear model is used, where the response $Y_i$ is generated from

$$Y_i = X_i^T \beta + e_i, \quad e_i \sim N(0, \sigma^2) \quad (i = 1, \ldots, n),$$

(8)

where $X_i = (X_{i1}, \ldots, X_{ip})^T$ is a vector of predictors, and is independent of $e_i$. For each example, a training sample of size 50 and a test sample of size $10^2$ are generated. Details of the five examples are as follows.

**Example 1.** In (8), $X_i$ is sampled from $N(0, \Sigma)$ with $p = 10$, where the $jk$th element of $\Sigma$ is $0.5^{|j-k|}$. Here $\beta = (3, -1.5, 0, 0, 1, 0, 0, 2, 0)^T$ and $\sigma = 3$.

**Example 2.** As Example 1 except that $\beta = (0.85, \ldots, 0.85)^T$.

**Example 3.** In (8), $X_i$ is sampled from $N(0, \Sigma)$ with $p = 20$, where the diagonal and off-diagonal elements of $\Sigma$ are 1 and 0.5, respectively. Here $\beta = (0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2, 2, 2)^T$ and $\sigma = 9$.

**Example 4.** In (8), let $X_i = (W_{i1}, W_{i2}^2, W_{i3}^3, \ldots, W_{i5}, W_{i5}^2, W_{i5}^3)^T$, where $W_{ik} = (U_{ik} + V)/\sqrt{2}$ $(k = 1, \ldots, 5)$ and $U_{i1}, \ldots, U_{i5}$ and $V$ are generated from $N(0, 1)$ independently. Here $\beta = (0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 2/3, -1, 1/3, 0, 0, 0)^T$ and $\sigma = 3$.

**Example 5.** In (8), $X_i$ is sampled from $N(0, \sigma)$ with $p = 200$, where the $jk$th element of $\Sigma$ is $0.5^{|j-k|}$. Here elements 1–5 of $\beta$ are 3, elements 6–10 of $\beta$ are $-1.5$, elements 11–15 are 1, elements 16–20 are 2 and the rest are zeros.
Table 1. Simulation results for least-squares regression. Averaged test errors, standard errors in parentheses and the number of distinct nonzero coefficients in curly brackets of the four methods with optimal tuning by the oracle and the generalized degrees of freedom, based on 100 simulation replications. The smallest error over the four methods is in italics.

<table>
<thead>
<tr>
<th>Example</th>
<th>Criterion</th>
<th>Method</th>
<th>L1L∞</th>
<th>Elastic</th>
<th>Lasso</th>
<th>L∞</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Oracle</td>
<td>10.4(0.1)</td>
<td>10.3(0.1)</td>
<td>10.4(0.1)</td>
<td>11.0(0.1)</td>
<td></td>
</tr>
<tr>
<td>GDF</td>
<td>10.1(0.2)</td>
<td>11.1(0.2)</td>
<td>11.0(0.1)</td>
<td>11.6(0.2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Oracle</td>
<td>9.7(0.1)</td>
<td>10.9(0.1)</td>
<td>10.9(0.1)</td>
<td>9.7(0.1)</td>
<td></td>
</tr>
<tr>
<td>GDF</td>
<td>10.4(0.1)</td>
<td>11.9(0.1)</td>
<td>11.3(0.1)</td>
<td>10.1(0.1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Oracle</td>
<td>86.9(0.4)</td>
<td>91.1(0.6)</td>
<td>91.5(0.6)</td>
<td>88.8(0.4)</td>
<td></td>
</tr>
<tr>
<td>GDF</td>
<td>90.8(0.6)</td>
<td>97.3(0.8)</td>
<td>93.6(0.7)</td>
<td>92.4(0.6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Oracle</td>
<td>10.4(0.1)</td>
<td>10.7(0.1)</td>
<td>10.8(0.1)</td>
<td>11.5(0.2)</td>
<td></td>
</tr>
<tr>
<td>GDF</td>
<td>11.4(0.1)</td>
<td>11.6(0.1)</td>
<td>11.4(0.2)</td>
<td>12.2(0.2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Oracle</td>
<td>35.2(1.2)</td>
<td>34.3(1.2)</td>
<td>39.8(1.1)</td>
<td>72.3(0.7)</td>
<td></td>
</tr>
<tr>
<td>GDF</td>
<td>39.4(1.8)</td>
<td>39.8(2.0)</td>
<td>42.5(1.1)</td>
<td>78.7(1.1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GDF, generalized degrees of freedom.

Table 1 indicates that the \( L_1 L_\infty \)-penalty performs as well as the lasso and elastic net penalties in both the large \( n \) small \( p \) and large \( p \) small \( n \) sparse cases and as well as the \( L_\infty \)-penalty in the nonsparse case, adapting to a variety of situations by changing the value of \( \lambda \). Interestingly, it outperforms the others in the sparse grouped predictors case, implying that its grouping property provides dimensionality reduction. As a result, it tends to identify a simpler model. In fact, the number of distinct nonzero coefficients identified by the \( L_1 L_\infty \)-penalty is close to those of the elastic net and lasso in the sparse situation, and becomes much smaller in the nonsparse situation.

With regard to the quality of estimation of \( \text{PE}(\hat{\mu}_\lambda) = E\{Y - \hat{\mu}_\lambda(X)\}^2 \), (7) performs well as compared to the oracle test error that is the minimum value of the empirical \( \text{PE}(\hat{\mu}_\lambda) \) evaluated through the test samples on the prespecified \( \lambda \) values.

In classification, three examples are examined, slightly modified from those used in Wang et al. (2006). In the case of small \( p \) and large \( n \), we generate a training sample of size \( n = 50 \) and \( p = 10 \), with 50% of them having the positive class. In the case of large \( p \) and small \( n \), we take \( u = 50 \) and \( p = 300 \). In each example, a test sample of size \( 10^4 \) is used to evaluate the performance of each method after adaptive tuning through the generalized degrees of freedom.

**Example 1.** First, \( X_i \) is generated from \( N(\mu, I_{p \times p}) \) with \( \mu = (0.5, 0.5, 0.5, 0.5, 0, \ldots, 0)^T \in \mathbb{R}^p \) and assign \( Y_i = 1 \) \((i = 1, \ldots, [n/2]) \). Second, \( X_i \) is generated from \( N(-\mu, I_{p \times p}) \) and assign \( Y_i = -1 \) \((i = [n/2], 1, \ldots, n) \).

**Example 2.** First, \( X_i \) is generated from \( N(\mu, \Sigma) \) with \( \mu = (1, 1, 1, 1, 0, \ldots, 0)^T \in \mathbb{R}^p \), and assign \( Y_i = 1 \) \((i = 1, \ldots, [n/2]) \). Second, \( X_i \) is generated from \( N(-\mu, \Sigma) \) and assign \( Y_i = -1 \) \((i = [n/2], 1, \ldots, n) \). Here

\[
\Sigma = \begin{pmatrix}
\Sigma^* & 0_{5 \times (p-5)} \\
0_{(p-5) \times 5} & I_{(p-5) \times (p-5)}
\end{pmatrix},
\]

where the diagonal and the off-diagonal elements of \( \Sigma^* \) equal 1 and 0.8, respectively.

**Example 3.** As Example 2 except that the \( jk \)th component of \( \Sigma^* \) is \( 0.8|j-k| \).
Table 2. Simulation results for binary support vector machine classification. Averaged test errors and standard errors in parentheses of the four support vector machines with optimal tuning by the generalized degrees of freedom over 100 simulation replications. The smallest error over the four methods is in italics.

<table>
<thead>
<tr>
<th>Example</th>
<th>n &amp; p</th>
<th>( L_1 L_\infty )</th>
<th>( L_1 )</th>
<th>( L_2 )</th>
<th>( L_\infty )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( n \gg p )</td>
<td>0.133 (0.001)</td>
<td>0.143 (0.001)</td>
<td>0.145 (0.001)</td>
<td>0.145 (0.002)</td>
</tr>
<tr>
<td></td>
<td>( n \ll p )</td>
<td>0.167 (0.003)</td>
<td>0.197 (0.005)</td>
<td>0.321 (0.003)</td>
<td>0.366 (0.005)</td>
</tr>
<tr>
<td>2</td>
<td>( n \gg p )</td>
<td>0.138 (0.001)</td>
<td>0.142 (0.001)</td>
<td>0.139 (0.001)</td>
<td>0.143 (0.001)</td>
</tr>
<tr>
<td></td>
<td>( n \ll p )</td>
<td>0.140 (0.001)</td>
<td>0.147 (0.001)</td>
<td>0.175 (0.001)</td>
<td>0.298 (0.005)</td>
</tr>
<tr>
<td>3</td>
<td>( n \gg p )</td>
<td>0.118 (0.001)</td>
<td>0.120 (0.001)</td>
<td>0.121 (0.001)</td>
<td>0.123 (0.001)</td>
</tr>
<tr>
<td></td>
<td>( n \ll p )</td>
<td>0.121 (0.001)</td>
<td>0.131 (0.002)</td>
<td>0.162 (0.001)</td>
<td>0.283 (0.005)</td>
</tr>
</tbody>
</table>

Table 2 shows that the \( L_1 L_\infty \) support vector machine outperforms its competitors in each case. This suggests that the \( L_1 L_\infty \)-penalty goes beyond its \( L_1 \) and \( L_\infty \) counterparts in terms of adaptation. However, the improvement varies over the competitors, with the largest amount occurring in the case of large \( p \) and small \( n \).

5.2. Applications

The Wisconsin Breast Cancer Dataset, collected at the University of Wisconsin Hospitals, develops a prediction model for discriminating benign from malignant breast tissue samples through nine clinical diagnostic characteristics. These characteristics are assigned integer values through nine clinical diagnostic characteristics. These characteristics are assigned integer values 1, \ldots, 10, with lower values indicating the most normal states. Detailed descriptions of the Wisconsin Breast Cancer Dataset can be found in Wolberg & Mangasarian (1990).

For the Wisconsin Breast Cancer Dataset, we apply the \( L_1 \), \( L_2 \), \( L_\infty \) and \( L_1 L_\infty \) support vector machines. To crossvalidate our analysis, we randomly divide the 682 tissue samples into equal halves for training and testing. Averaged test errors and standard errors in parentheses over 100 random partitions for the \( L_1 L_\infty \), \( L_1 \), \( L_2 \) and \( L_\infty \) support vector machines are 0.025(0.001), 0.028(0.001), 0.026(0.001) and 0.027(0.001), respectively. The \( L_1 L_\infty \) support vector machine outperforms its competitors in terms of predictive accuracy. It appears that the Wisconsin Breast Cancer Dataset is a nonsparse case as the \( L_1 \) support vector machine performs worst.

The leukaemia DNA microarray dataset studied in Golub et al. (1999) concerns prediction of two types of acute leukaemia, lymphoblastic and myeloid, through gene expression profiles. Of particular interest is selecting a subset of genes, among 7129 candidate genes, as a prediction marker of acute leukaemia. For the 7129 genes, 1059 genes remain after a prescreening process consisting of thresholding, filtering and standardization (Dudoit et al., 2002). The data contain 72 tissue samples of the two types of acute leukaemia, among which 57 samples are lymphoblastic, together with expression profiles of 1059 candidate genes. Details can be found at http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

For the data, we apply the four support vector machines to 38 training samples, and use an additional 34 for testing as in Golub et al. (1999). The test errors and the numbers of selected genes in parentheses for the \( L_1 L_\infty \), \( L_1 \), \( L_2 \) and \( L_\infty \) support vector machines are 0/34(65), 1/34(16), 2/34(1059) and 2/34(1059), respectively. The \( L_1 L_\infty \) support vector machine performs best in terms of predictive accuracy, while identifying 65 important genes. The \( L_1 \) support vector machine selects 16 important genes. In fact, the maximum number of important genes that can be selected by the \( L_1 \) support vector machine is no greater than the training sample size 38, which may be too small to be realistic, whereas the \( L_2 \) and \( L_\infty \) support vector machines select all 1059 genes.
genes. In a similar study, the elastic net selects 78 out of 2308 (Wang et al., 2006). This result is comparable to what we obtain here.

6. Theory

This section investigates statistical aspects of the grouping, adaptation and shrinkage properties of the $L_1 L_\infty$-penalty.

**Theorem 4.** In least-squares regression, let $c_j(\hat{\beta}_x) = -\sum_{i=1}^n x_{ij}(y_i - x_i^T \hat{\beta}_x)$ denote the correlation between predictors and residuals. For any $\lambda$ and $j = 1, \ldots, p$, if $|c_j(\hat{\beta}_x)| > \lambda_1/2$ and $\hat{\beta}_j \neq 0$, then $\hat{\beta}_{x,j} = \text{sign}(c_j(\hat{\beta}_x)) \|\hat{\beta}_x\|_\infty$. Furthermore, in case of orthonormal predictors,

$$\hat{\beta}_{x,j} = \begin{cases} (\sum_{i=1}^n x_{ij}y_i - \frac{\lambda}{2})_+ \text{sign}(c_j(\hat{\beta}_x)), & |c_j(\hat{\beta}_x)| \leq \frac{\lambda}{2}; \\ \frac{1}{\text{card}(A^c)} \{ \sum_{j \in A^c} (\sum_{i=1}^n x_{ij}y_i - \frac{\lambda}{2}) - \frac{\lambda}{2} \} \text{sign}(c_j(\hat{\beta}_x)), & |c_j(\hat{\beta}_x)| > \frac{\lambda}{2}. \end{cases}$$

Theorem 4 says that $\hat{\beta}_{x,j}$ is grouped at $\text{sign}(c_j(\hat{\beta}_x)) \|\hat{\beta}_x\|_\infty$ if the sample correlation between the predictor $x_j$ and the residuals exceeds $\lambda_1/2$. When $|c_j(\hat{\beta}_x)| > \lambda_1/2$, as suggested in the orthonormal case, $\sum_{i=1}^n x_{ij}y_i$, the $j$th component of the ordinary least-squares estimate, gets shrunk by both the $L_1$- and $L_\infty$-components of the $L_1 L_\infty$-penalty, and $\hat{\beta}_{x,j}$ is pulled down to equal $\text{sign}(c_j(\hat{\beta}_x)) \|\hat{\beta}_x\|_\infty$. When $|c_j(\hat{\beta}_x)| \leq \lambda_1/2$, the $L_1 L_\infty$-penalty performs a lasso-type thresholding (Tibshirani, 1996). Consequently, the grouping property is incorporated into shrinkage, which enables the $L_1 L_\infty$-penalty to yield a simple model regardless of sparseness. In the sparse situation, $\lambda_1$ can be greater than $\lambda_\infty$ to yield some coefficients to be shrunken to zero. In the nonsparse case, the opposite occurs, and the $L_1 L_\infty$-penalty forces some predictors to be grouped. In both cases, the $L_1 L_\infty$-penalty produces a simple model.

As mentioned in §2-1, grouping by the elastic net uses the correlations among predictors while grouping by the $L_1 L_\infty$-penalty deals with $c(\hat{\beta}_x)$. It is clear that the correlations among the predictors do not determine the correlations between the predictors and the response. Thus grouping by the former may not reduce estimation variance and may result in degradation of variable selection. To confirm this intuition, we examine a simple example. We sample $X_1$ from $\text{Un}[-10, 10]$, and let $X_k = e \times X_1$ ($k = 2, \ldots, 10$), where $e$ follows $N(3, 1)$ and is independent of $X_1$. Then the response $Y$ is $X_1 + e$, where $e \sim N(0, 3)$. This makes $X_1$ and $X_k$ ($k = 2, \ldots, 10$) highly correlated but $Y$ is conditionally independent of $(X_2, \ldots, X_{10})$ given $X_1$. Consequently, the selected model should contain $X_1$ only. We generate 100 datasets, each with a training and test sample of 50 and 500 observations. Tuning is performed as in §5.

The averaged test errors and the standard errors in parentheses for the $L_1 L_\infty$-penalty and the elastic net are $3.450(0.151)$ and $5.373(0.272)$, respectively. It shows that predictive performance of the elastic net penalty is worse than the $L_1 L_\infty$-penalty. In fact, the elastic net penalty selects an average of 8.19 variables while the $L_1 L_\infty$-penalty selects 4.52. This example demonstrates that prediction accuracy and variable selection are not directly related with the correlations among predictors.

The next theorem explains how the grouping property of the $L_1 L_\infty$-penalty leads to adaptive regularization.

**Theorem 5.** In least-squares regression if $\hat{\beta}_x = 0$, then the regularized loss (3) reduces to

$$\sum_{i=1}^n (y_i - x_i^T \hat{\beta}_x^e)^2 + \lambda \sum_{j=1}^{\text{card}(I_x) + 1} |\hat{\beta}_{x,j}^e|,$$
where \( x_i^c = \{x_{ik}, \ldots, x_{ik_{\text{card}(\mathcal{I}_k)}}\}, \sum_{j \in \mathcal{A}_\infty^k} x_{ij} \text{sign}(\hat{\beta}_{k,j}) \}^T \), and \( \hat{\beta}_k^c = (\hat{\beta}_{k,1}, \ldots, \hat{\beta}_{k,\text{card}(\mathcal{I}_k)}, \|\hat{\beta}_k\|_\infty)^T \) with \( \{k_1, \ldots, k_{\text{card}(\mathcal{I}_k)}\} \in \mathcal{I}_k \), and \( \lambda^c \) is a vector whose first card(\( \mathcal{I}_k \)) elements are \( \lambda_1 \) and the last element is card(\( \mathcal{A}_{\infty}^k \)) \( \lambda_1 + \lambda_\infty \); \( \mathcal{I}_k \) and \( \mathcal{A}_{\infty}^k \) are defined in §3.2.

Theorem 5 says that \( \|\hat{\beta}_k\|_\infty \) is regularized by card(\( \mathcal{A}_{\infty}^k \)) \( \lambda_1 + \lambda_\infty \) while \( \hat{\beta}_{k,j}, j \in \mathcal{I}_k \) is controlled by \( \lambda_1 \). Because card(\( \mathcal{A}_{\infty}^k \)) \( \lambda_1 + \lambda_\infty \) > \( \lambda_1 \), it indicates the \( L_1L_\infty \) -penalty regularizes \( \|\hat{\beta}_k\|_\infty \) more than the components of \( \hat{\beta}_k \) in \( \mathcal{I}_k \). In other words, the \( L_1L_\infty \) -penalty achieves adaptive regularization based on \( c_j(\hat{\beta}_k) \) because any \( \hat{\beta}_{k,j} \) with \( c_j(\hat{\beta}_k) > \lambda_1/2 \) is grouped at sign(\( c_j(\hat{\beta}_k) \)) \( \|\hat{\beta}_k\|_\infty \).

**ACKNOWLEDGEMENT**

This research was supported in part by grants from the U.S. National Science Foundation and National Institutes of Health.

**APPENDIX**

*Proof of Theorem 1.* Since \( \lambda \) is not a transition point, the index sets remain unchanged in an interval. Then it suffices to prove the piecewise linearity of \( \hat{\beta}_k, b_{\infty}^k \) and \( b_1^k \) in this interval. We can write (5) at \( \lambda \) as

\[
0 = -2 \sum_{i=1}^{n} x_{ij} \left\{ y_i - \sum_{k \in \mathcal{I}_k} x_{ik} \hat{\beta}_{k,j} - \sum_{k \in \mathcal{A}_{\infty}^k} x_{ik} \text{sign}(\hat{\beta}_{k,j}) \|\hat{\beta}_k\|_\infty \right\} \\
+ \lambda_1 \text{sign}(\hat{\beta}_{k,j}) + \lambda_\infty \text{sign}(\hat{\beta}_{k,j}) \|b_{\infty,j}^k\|_\infty \quad (j \in \mathcal{A}_{\infty}^k), \quad (A1)
\]

\[
0 = -2 \sum_{i=1}^{n} x_{ij} \left\{ y_i - \sum_{k \in \mathcal{I}_k} x_{ik} \hat{\beta}_{k,j} - \sum_{k \in \mathcal{A}_{\infty}^k} x_{ik} \text{sign}(\hat{\beta}_{k,j}) \|\hat{\beta}_k\|_\infty \right\} \\
+ \lambda_1 \text{sign}(\hat{\beta}_{k,j}) \quad (j \in \mathcal{I}_k), \quad (A2)
\]

\[
0 = -2 \sum_{i=1}^{n} x_{ij} \left\{ y_i - \sum_{k \in \mathcal{I}_k} x_{ik} \hat{\beta}_{k,j} - \sum_{k \in \mathcal{A}_{\infty}^k} x_{ik} \text{sign}(\hat{\beta}_{k,j}) \|\hat{\beta}_k\|_\infty \right\} \\
+ \lambda_1 b_{1,j}^k \quad (j \in \mathcal{A}_1^k). \quad (A3)
\]

We first give the proof for \( \hat{\beta}_k \). Eliminating \( b_{\infty,j}^k \) from (A1) through (4), we obtain

\[-2 \sum_{j \in \mathcal{A}_{\infty}^k} \text{sign}(\hat{\beta}_{k,j}) \sum_{i=1}^{n} x_{ij} \left\{ y_i - \sum_{k \in \mathcal{I}_k} x_{ik} \hat{\beta}_{k,j} - \sum_{k \in \mathcal{A}_{\infty}^k} x_{ik} \text{sign}(\hat{\beta}_{k,j}) \|\hat{\beta}_k\|_\infty \right\} + \text{card}(\mathcal{A}_{\infty}^k) \lambda_1 + \lambda_\infty = 0.
\]

Solving this equation and (A2) for \( \hat{\beta}_{k,j} \) (\( j \in \mathcal{I}_k \)) and \( \|\hat{\beta}_k\|_\infty \) shows that they are linear in \( \lambda \). This proves that \( \hat{\beta}_k \) is piecewise linear in \( \lambda \). Using this solution for \( \hat{\beta}_k \), we can solve (A1) for \( b_{\infty,j}^k \) (\( j \in \mathcal{A}_{\infty}^k \)) and (A3) for \( b_{1,j}^k \) (\( j \in \mathcal{A}_1^k \)), and this implies that \( b_{\infty}^k \) is piecewise linear in \((\lambda_1/\lambda_\infty, 1/\lambda_\infty)^T\) and \( b_1^k \) is piecewise linear in \((\lambda_\infty/\lambda_1, 1/\lambda_1)^T\). □
Proof of Theorem 2. Since \( \lambda \) is not a transition point, the index sets remain unchanged in an interval. In this interval, (6) can be written as

\[
- \sum_{i \in \mathcal{H}_k} \alpha_{i,j} y_i x_{ij} - \sum_{i \in \mathcal{L}_k} y_i x_{ij} + \lambda_1 \text{sign}(\hat{\beta}_{\lambda,j}) + \lambda_\infty \text{sign}(\hat{\beta}_{\lambda,j}) |b_{\lambda,j}^\infty| = 0 \quad (j \in A_{\lambda}^\infty), \tag{A4}
\]

\[
- \sum_{i \in \mathcal{H}_k} \alpha_{i,j} y_i x_{ij} - \sum_{i \in \mathcal{L}_k} y_i x_{ij} + \lambda_1 \text{sign}(\hat{\beta}_{\lambda,j}) = 0 \quad (j \in \mathcal{I}_k), \tag{A5}
\]

\[
- \sum_{i \in \mathcal{H}_k} \alpha_{i,j} y_i x_{ij} - \sum_{i \in \mathcal{L}_k} y_i x_{ij} + \lambda_1 b_{\lambda,j}^1 = 0 \quad (j \in A_{\lambda}^1). \tag{A6}
\]

\[
\sum_{i \in \mathcal{H}_k} \alpha_{i,j} y_i + \sum_{i \in \mathcal{L}_k} y_i = 0. \tag{A7}
\]

For these card(\( \mathcal{H}_k \)) equations to solve for card(\( \mathcal{I}_k \)) + 2 number of unknowns \( \hat{\beta}_{\lambda,j} \) \( (j \in \mathcal{I}_k) \), \( \hat{\beta}_{\lambda,0} \) and \( \|\hat{\beta}_{\lambda}\|_\infty \), we impose card(\( \mathcal{H}_k \)) = card(\( \mathcal{I}_k \)) + 2 as in Wang & Shen (2006b). From the fact that this system of equations is independent of \( \hat{\beta}_{\lambda,j} \) \( (j \in \mathcal{I}_k) \), \( \hat{\beta}_{\lambda,0} \) and \( \|\hat{\beta}_{\lambda}\|_\infty \), it follows that \( \hat{\beta}_{\lambda} \) and \( \hat{\beta}_{\lambda,0} \) are piecewise constants. Eliminating \( b_{\lambda,j}^\infty \) from (A4) through (4), we obtain

\[
\sum_{j \in A_{\lambda}^\infty} \text{sign}(\hat{\beta}_{\lambda,j}) \left( \sum_{i \in \mathcal{H}_k} \alpha_{i,j} y_i x_{ij} - \sum_{i \in \mathcal{L}_k} y_i x_{ij} \right) + \text{card}(A_{\lambda}^\infty) \lambda_1 + \lambda_\infty = 0.
\]

Solving this equation, (A5) and (A7) for \( \alpha_{i,j} \) \( (i \in \mathcal{H}_k) \) yields that they are linear in \( \lambda \), which establishes that \( \alpha_{i,j} \) is piecewise linear in \( \lambda \). Using these solutions for \( \hat{\beta}_{\lambda} \) and \( \alpha_{i,j} \) and solving (A4) for \( b_{\lambda,j}^\infty \) \( (j \in A_{\lambda}^\infty) \) and (A6) for \( b_{\lambda,j}^1 \) \( (j \in A_{\lambda}^1) \) implies that \( b_{\lambda}^\infty \) is piecewise linear in \( \lambda_1/\lambda_\infty, 1/\lambda_\infty \) and \( b_{\lambda,j}^1 \) is piecewise linear in \( \lambda_1/\lambda_\infty, 1/\lambda_\infty \). \( \square \)

Proof of Theorem 3. This proof employs Theorems 1 and 2 in Zou et al. (2007) and Theorem 5. Following the notation in Theorem 5, let \( X^\delta \) be the matrix whose rows are \( x_{i}^T \) \( (i = 1, \ldots, n) \). If \( \lambda \) is not a transition point, then \( d\left(\sum_{i=1}^{n} y_i - x_i^T \hat{\beta}_{\lambda}^\delta \right) + \sum_{j=1}^{\text{card}(\mathbb{I}_{\lambda}^T)+1} \beta_{\lambda,j} \left| \hat{\beta}_{\lambda,j}^\delta \right| / \lambda_1 \hat{\beta}_{\lambda,j}^\delta / \lambda_1 + \lambda_\infty = 0 \) yields \( \hat{\beta}_{\lambda}^\delta(y) = (X^T X)^{-1} (Y - \Delta y) / 2 \), where the vector \( \Delta y \) = \( [\text{sign}(\hat{\beta}_{\lambda,1}), \ldots, \text{sign}(\hat{\beta}_{\lambda,\text{card}(\mathbb{I}_{\lambda})})]_{\lambda_1 + \lambda_\infty} \). Observe that \( \hat{\mu}_{\lambda,j}(y) = X^T \hat{\beta}_{\lambda,j}^\delta(y) = P_{\lambda}(y) y - W_{\lambda,j}(y) \), where \( P_{\lambda}(y) = X^T (X^T X)^{-1} X^T \) and \( W_{\lambda,j}(y) = X^T (X^T X)^{-1} X^T \Delta y / 2 \).

Now we compute an infinitesimal change of \( \hat{\mu}_{\lambda,j}(y) \), when \( \lambda_1 \) changes infinitesimally, which is essentially to apply Stein’s lemma (Stein, 1981). By Theorem 1 in Zou et al. (2007), there exists a sufficiently small \( \epsilon \) such that \( \|\Delta y\|_2 < \epsilon \) keeping the index sets unchanged. Accordingly, for such a sufficiently small change of \( y \), we have \( P_{\lambda}(y + \Delta y) = P_{\lambda}(y) \) and \( W_{\lambda,j}(y + \Delta y) = W_{\lambda,j}(y) \), and hence \( d\hat{\mu}_{\lambda,j}(y) / \partial y \approx P_{\lambda}(y) \). By Theorem 2 in Zou et al. (2007), \( \hat{\mu}_{\lambda,j}(y) \) is almost differentiable with respect to \( y \). Then by Stein’s lemma, we obtain \( d\hat{\mu}_{\lambda,j} = \text{tr}(\partial \hat{\mu}_{\lambda,j}(y) / \partial y) = \text{tr}(P_{\lambda}(y) = \text{card}(\mathbb{I}_{\lambda}) + 1 + d\hat{\mu}_{\lambda,j} = E[\text{card}(\mathbb{I}_{\lambda}) + 1]. \) \( \square \)

Proof of Theorem 4. Suppose \( |c_j(\hat{\beta}_{\lambda})| > \lambda_1/2 \). Then from (5), \( |b_{\lambda,j}^1| < 1 \) implies \( |b_{\lambda,j}^\infty| > 0 \). On the other hand, \( |b_{\lambda,j}^1| < 1 \) implies \( |\hat{\beta}_{\lambda,j}^1| = 0 \) and hence \( |\hat{\beta}_{\lambda,j}^\infty| < \|\hat{\beta}_{\lambda}\|_\infty \), which means \( b_{\lambda,j}^\infty = 0 \) because \( \hat{\beta}_{\lambda} \neq 0 \) by the assumption. Thus \( |b_{\lambda,j}^1| < 1 \) does not satisfy \( |c_j(\hat{\beta}_{\lambda})| > \lambda_1/2 \), and we must have \( |b_{\lambda,j}^1| = 1 \).

Then \( |b_{\lambda,j}^\infty| > 0 \) and hence \( |\hat{\beta}_{\lambda,j}^\infty| = \|\hat{\beta}_{\lambda}\|_\infty \). Now the characteristics of \( b_{\lambda,j}^1 \) and \( b_{\lambda,j}^\infty \) imply sign(\( \hat{\beta}_{\lambda,j}^\delta \)) = sign(\( b_{\lambda,j}^1 \)) \( (j \in A_{\lambda}^\infty) \). Then from (5), we obtain sign(\( \hat{\beta}_{\lambda,j}^\delta \)) = sign(\( c_j(\hat{\beta}_{\lambda}) \)) \( (j \in A_{\lambda}^\infty) \). Since \( |c_j(\hat{\beta}_{\lambda})| > \lambda_1/2 \) is equivalent to \( j \in A_{\lambda}^\infty, \hat{\beta}_{\lambda,j} = \text{sign}(c_j(\hat{\beta}_{\lambda})) \|\hat{\beta}_{\lambda}\|_\infty \) if \( |c_j(\hat{\beta}_{\lambda})| > \lambda_1/2 \).
In the orthonormal case, (A1)–(A3) become

\[-2 \sum_{i=1}^{n} x_{ij}y_{i} + 2\hat{\beta}_{k,j} + \lambda_{1}\text{sign}(\hat{\beta}_{k,j}) + \lambda_{\infty}\text{sign}(\hat{\beta}_{k,j})|b_{\infty}^{\infty}| = 0 \quad (j \in \mathcal{A}^{\infty}_{k}), \quad (A8)\]

\[-2 \sum_{i=1}^{n} x_{ij}y_{i} + 2\hat{\beta}_{k,j} + \lambda_{1}\text{sign}(\hat{\beta}_{k,j}) = 0 \quad (j \in \mathcal{I}_{k}), \quad (A9)\]

\[-2 \sum_{i=1}^{n} x_{ij}y_{i} + \lambda_{1}b_{1}^{\infty} = 0 \quad (j \in \mathcal{A}^{1}_{k}). \quad (A10)\]

Applying \(\text{sign}(\hat{\beta}_{k,j}) = \text{sign}(\sum_{i=1}^{n} x_{ij}y_{i}) \quad (j \in \mathcal{I}_{k})\), from (A9) we get \(\hat{\beta}_{k,j} = \sum_{i=1}^{n} x_{ij}y_{i} - (\lambda_{1}/2)\text{sign}(\sum_{i=1}^{n} x_{ij}y_{i}) \quad (j \in \mathcal{I}_{k})\). The requirement \(|b_{1}^{\infty}| \leq 1\) in (A10) yields \(|\sum_{i=1}^{n} x_{ij}y_{i}| \leq \lambda_{1}/2\) if and only if \(j \in \mathcal{A}^{1}_{k}\) because \(|\sum_{i=1}^{n} x_{ij}y_{i}| > \lambda_{1}/2\) implies \(\hat{\beta}_{k,j} = 0\). Since \(j \in \mathcal{I}_{k} \cup \mathcal{A}^{1}_{k}\) is equivalent to \(|c_{j}(\hat{\beta}_{k})| \leq \lambda_{1}/2\), under \(|c_{j}(\hat{\beta}_{k})| \leq \lambda_{1}/2\) we obtain

\[
\hat{\beta}_{k,j} = \left(\sum_{i=1}^{n} x_{ij}y_{i} - \frac{\lambda_{1}}{2}\right) / |c_{j}(\hat{\beta}_{k})|. \]

Observe that \(\hat{\beta}_{k,j} = \|\hat{\beta}_{k}\|_{\infty}\text{sign}(\sum_{i=1}^{n} x_{ij}y_{i}) \quad (j \in \mathcal{A}^{\infty}_{k})\). This allows us to write (A8) as \(-2\|\sum_{i=1}^{n} x_{ij}y_{i}| + 2\|\hat{\beta}_{k}\|_{\infty} + \lambda_{1} + \lambda_{\infty}|b_{\infty}^{\infty}| = 0 \quad (j \in \mathcal{A}^{\infty}_{k})\). Now through (4), under \(|c_{j}(\hat{\beta}_{k})| > \lambda_{1}/2\) we obtain

\[
\hat{\beta}_{k,j} = \frac{1}{\text{card}(\mathcal{A}^{\infty}_{k})} \left\{ \sum_{j \in \mathcal{A}^{\infty}_{k}} \left( \sum_{i=1}^{n} x_{ij}y_{i} - \frac{\lambda_{1}}{2} \right) / \left( -\frac{\lambda_{1}}{2} \right) \right\} \text{sign}(c_{j}(\hat{\beta}_{k})). \]

**Proof of Theorem 5.** The proof is straightforward hence omitted.

\[\square\]

**REFERENCES**


Adaptive regularization using the entire solution surface


[Received November 2007. Revised January 2009]