Fast Sparse Regression and Classification

Jerome H. Friedman[*]

July 28, 2008

Abstract

Regularized regression and classification methods fit a linear model to data, based on some loss criterion, subject to a constraint on the coefficient values. As special cases, ridge-regression, the lasso, and subset selection all use squared-error loss with different particular constraint choices. For large problems the general choice of loss–constraint combinations is usually limited by the computation required to obtain the corresponding solution estimates, especially when non convex constraints are used to induce very sparse solutions. A fast algorithm is presented that produces solutions that closely approximate those for any convex loss and a wide variety of convex and non convex constraints, permitting application to very large problems. The benefits of this generality are illustrated by examples.

Key words and phrases: regression, classification, regularization, sparsity, variable selection, bridge–regression, lasso, elastic net, l_p–norm penalization.

1 Introduction

Linear structural models are among the most popular for fitting data. One is given N observations of the form

$$\{y_i, \mathbf{x}_i\}_1^N = \{y_i, x_{i1}, \cdot\cdot\cdot, x_{in}\}_1^N \tag{1}$$

considered to be a random sample form some joint (population) distribution with probability density $p(\mathbf{x}, y)$. The random variable y is the "outcome" or "response" and $\mathbf{x} = \{x_1, \cdot\cdot\cdot, x_n\}$ are the predictor variables. These predictors may be the original measured variables and/or selected functions constructed from them. The goal is to estimate the joint values for the parameters $\mathbf{a} = \{a_0, a_1, \cdot\cdot\cdot, a_n\}$ of the linear model

$$F(\mathbf{x}; \mathbf{a}) = a_0 + \sum_{j=1}^{n} a_j x_j \tag{2}$$

for predicting y given $\mathbf{x}$, that minimize the expected loss ("risk")

$$R(\mathbf{a}) = E_{\mathbf{x},y} L(y, F(\mathbf{x}; \mathbf{a})) \tag{3}$$

over future predictions $\mathbf{x}, y \backsim p(\mathbf{x}, y)$. Here $L(y, F)$ is a loss criterion that specifies the cost of predicting the value F when the actual value is y. Popular loss criteria include squared–error

$$L(y, F) = (y - F)^2, \tag{4}$$

and Bernoulli negative log–likelihood

$$L(y, F) = \log(1 + e^{-yF}), \quad y \in \{-1, 1\} \tag{5}$$

[*] Department of Statistics, Stanford University, Stanford, CA 94305 (jhf@stanford.edu)

associated with logistic regression. For a specified loss criterion the optimal parameter values are from (3)

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} R(\mathbf{a}). \tag{6}$$

Since the population probability density $p(\mathbf{x}, y)$ is unknown, a common practice is to substitute an empirical estimate of the expected value in (3) based on the available data (1) yielding

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \hat{R}(\mathbf{a}) \tag{7}$$

as an estimate for $\mathbf{a}^*$, where

$$\hat{R}(\mathbf{a}) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i,\, a_0 + \sum_{j=1}^{n} a_j x_{ij}\right). \tag{8}$$

2 Regularization

It is well known that $\hat{\mathbf{a}}$ (7) (8) often provides poor estimates of $\mathbf{a}^*$; that is $R(\hat{\mathbf{a}}) >> R(\mathbf{a}^*)$. This is especially the case when the sample size N is not large compared to the number of parameters $(n+1)$. This is caused by the high variability of the estimates (7) when (8) is evaluated on different random samples drawn from the population distribution. A common remedy is to modify (7) in order to stabilize the estimates by placing a restriction on the joint solution values. That is,

$$\hat{\mathbf{a}}(t) = \arg\min_{\mathbf{a}} \hat{R}(\mathbf{a}) \;\; \text{s.t.} \;\; P(\mathbf{a}) \leq t. \tag{9}$$

Here $P(\mathbf{a})$ is a non negative function of the parameters specifying the form of the constraint and $t \geq 0$ regulates its strength. Setting $t = 0$ produces maximum restriction by requiring the solution values to exactly satisfy $P(\mathbf{a}) = 0$, thereby producing the least variance. Setting $t \geq P(\hat{\mathbf{a}})$ produces the unrestricted solution (7) with maximal variance. Intermediate values $0 < t < P(\hat{\mathbf{a}})$ provide degrees of restriction between these two extremes, thereby regulating the stability (variance) of the estimates $\hat{\mathbf{a}}(t)$ with respect to different training samples (1) drawn from $p(\mathbf{x}, y)$.

For a given data set (1), loss criterion $L(y, F)$ (3) (8), and constraint function $P(\mathbf{a})$, the solution to (9) depends only on the value chosen for t. Varying its value induces a family of solutions, each member being indexed by a particular value of $t \in [0, P(\hat{\mathbf{a}})]$. This same family of solutions can be obtained through the equivalent (penalized) formulation of (9)

$$\hat{\mathbf{a}}(\lambda) = \arg\min_{\mathbf{a}} [\hat{R}(\mathbf{a}) + \lambda \cdot P(\mathbf{a})] \tag{10}$$

where $P(\mathbf{a})$ is the constraining function in (9), here called the penalty, and $\lambda > 0$ regulates its strength. Setting $\lambda = \infty$ produces the totally constrained solution $(t = 0)$ whereas $\lambda = 0$ yields the unrestricted solution $(t \geq P(\hat{\mathbf{a}}))$. Each value of $0 \leq \lambda \leq \infty$ in (10) produces one of the solutions $0 \leq t \leq P(\hat{\mathbf{a}})$ in (9) with smaller values of λ corresponding to larger values of t. Thus (10) produces a family of estimates in which each member of the family is indexed by a particular value for the strength parameter λ. This family lies on a one–dimensional path of finite length in the $(n+1)$–dimensional space of all joint parameter values.

2.1 Model selection

The optimal parameter values $\mathbf{a}^*$ (6) also represent a point in the parameter space. For a given penalty, the goal is to find a point on its corresponding path $\hat{\mathbf{a}}(\lambda^*)$ that is closest to $\mathbf{a}^*$, where distance is characterized by the prediction risk (3)

$$D(\mathbf{a}, \mathbf{a}^*) = R(\mathbf{a}) - R(\mathbf{a}^*). \tag{11}$$

This is a classic model selection problem where one attempts to obtain an estimate $\hat{\lambda}$ for the optimal value of the strength parameter

$$\lambda^* = \arg \min_{0 \leq \lambda \leq \infty} R(\hat{\mathbf{a}}(\lambda)) \tag{12}$$

through

$$\hat{\lambda} = \arg \min_{0 \leq \lambda \leq \infty} \tilde{R}(\hat{\mathbf{a}}(\lambda)) \tag{13}$$

where $\tilde{R}(\mathbf{a})$ is a surrogate model selection criterion whose minimum is intended to approximate that for the actual risk (3).

There are a wide variety of model selection criteria each developed for a particular combination of loss (3) and penalty $P(\mathbf{a})$. Among the most general, applicable to any loss–penalty combination, is cross–validation. The data are randomly partitioned into two subsets (learning and test). The path is constructed using only the learning sample. The test sample is then used as an empirical surrogate for the population density $p(\mathbf{x}, y)$ to compute the corresponding (estimated) risk in (3). These estimates are then used in (13) to obtain the estimate $\hat{\lambda}$. Sometimes the risk used in (13) is estimated by averaging over several (K) such partitions ("K–fold" cross–validation).

2.2 Penalty Selection

Given a model selection procedure, the goal is to construct a path $\hat{\mathbf{a}}(\lambda)$ in parameter space such that some of the points on that path are close to the point $\mathbf{a}^*$ (6) representing the optimal solution. If no points on the path come close to $\mathbf{a}^*$, as measured by (11), then no model selection procedure can produce accurate estimates $\hat{\mathbf{a}}(\hat{\lambda})$. Since the path produced by (10) depends on the data, different randomly drawn data sets (1) will produce different paths for the same penalty. Thus the paths are themselves random, and one seeks a penalty $P(\mathbf{a})$ that produces paths $\hat{\mathbf{a}}(\lambda)$ such that

$$\left[E_T R(\hat{\mathbf{a}}(\lambda^*)) - R(\mathbf{a}^*)\right] / R(\mathbf{a}^*) = \text{ small} \tag{14}$$

with T being repeated data samples (1) drawn randomly from the joint density $p(\mathbf{x}, y)$, and λ^* is given by (12). This will depend on the particular $\mathbf{a}^*$ (6) associated with the application. Therefore, penalty choice is governed by whatever is known about the properties of $\mathbf{a}^*$.

2.3 Sparsity

One property of $\mathbf{a}^*$ that is often suspected is sparsity. That is, only a small fraction of the input variables $\{x_j\}_1^n$ are influencing predictions, with the identities of those influential variables being unknown. The degree of sparsity $S(\mathbf{a})$ of a parameter vector $\mathbf{a}$ can be defined as

$$S(\mathbf{a}) = \frac{1}{n} \sum_{k=1}^{n} I(\mid a_k \mid \, \leq \eta \cdot \max_j \mid a_j \mid) \tag{15}$$

with $\eta << 1$. If the predictor variables are all standardized to have similar scales then $S(\mathbf{a}^*)$ represents the fraction of non influential variables characterizing the problem.

If $\hat{\mathbf{a}}(\lambda^*) \simeq \mathbf{a}^*$ (14) then $S(\hat{\mathbf{a}}(\lambda^*)) \simeq S(\mathbf{a}^*)$, and in the absence of other information it is reasonable to choose a penalty that produces solutions $\hat{\mathbf{a}}(\lambda)$ with sparsity similar to that of $\mathbf{a}^*$ at $\lambda = \lambda^*$. Since the actual sparsity of $\mathbf{a}^*$ is generally unknown, one can define a family of penalties $P_\gamma(\mathbf{a})$, where γ indexes particular penalties in the family that produce solutions of differing sparseness, and then use model selection (Section 2.1) to jointly choose good values for γ and λ. That is,

$$\hat{\mathbf{a}}_\gamma(\lambda) = \arg \min_{\mathbf{a}} [\hat{R}(\mathbf{a}) + \lambda \cdot P_\gamma(\mathbf{a})] \tag{16}$$

$$(\hat{\gamma}, \hat{\lambda}) = \arg \min_{\gamma, \lambda} \tilde{R}(\hat{\mathbf{a}}_\gamma(\lambda)). \tag{17}$$

This approach is referred to as "bridge-regression" (Frank and Friedman 1993).

2.3.1 Power family

One such family of penalties is the power family defined as

$$P_\gamma(\mathbf{a}) = \sum_{j=1}^{n} |\, a_j \,|^\gamma; \quad \gamma \geq 0. \tag{18}$$

This is the l_γ–norm of the parameter vector $\mathbf{a}$ raised to the γ power.

Using squared–error loss (4), special cases of (8) (10) (18) include several popular regularized regression methods, namely $\gamma = 2$: ridge–regression, $\gamma = 1$: lasso , $\gamma = 0$: all–subsets regression. Ridge–regression (Horel and Kennard 1970) produces dense solutions, $S(\hat{\mathbf{a}}(\lambda)) \simeq 0$ (15), over its entire path $\infty \leq \lambda \leq 0$ while heavily shrinking the coefficient absolute values $|\, \hat{a}_j(\lambda) \,| << |\, a_j^* \,|$ for larger values of $|\, a_j^* \,|$ and λ. At the other extreme, all–subsets regression produces the sparsest solutions along its path (set of distinct points) by forcing many of the coefficient estimates to be zero and applying no shrinkage to the non zero estimates. The number of non zero coefficient estimates is regulated by the value of λ; larger values of λ produce fewer non zero coefficients. The lasso (Tibshirani 1996) produces paths intermediate between these two extremes, setting some coefficients to zero and applying shrinkage to the absolute values of the others. As λ increases along the path both the degree of shrinkage and the number of zero valued coefficients increase.

For $0 \leq \gamma \leq 2$ the power family (18) represents a continuum of penalties between all–subsets regression (sparsest solutions) and ridge–regression (dense solutions). For $\gamma > 1$ all coefficient estimates are strictly non zero at all points along the path, $\{|\, \hat{a}_j(\lambda) \,| \, > 0\}_1^n$ for $0 \leq \lambda < \infty$. However, their dispersion (coefficient of variation) at corresponding path points decreases with increasing γ. Note that for $\gamma \geq 1$ all penalties in the power family are convex functions of their argument $\mathbf{a}$, so that for convex risk $\hat{R}(\mathbf{a})$ (8) the problems represented by (16) are convex optimizations. For $\gamma < 1$ the penalties are non convex requiring (more difficult) non convex optimization techniques.

2.3.2 Generalized elastic net

The power family (18) is not the only possibility for bridging all–subsets and ridge regression. For bridging the lasso and ridge–regression Zou and Hastie 2005 proposed the elastic net family of penalties which can be expressed as

$$P_\beta(\mathbf{a}) = \sum_{j=1}^{n} (\beta - 1)\, a_j^2/2 + (2 - \beta)\, |\, a_j \,|\,; \quad 1 \leq \beta \leq 2. \tag{19}$$

Here the parameter β indexes family members with $\beta = 2$ yielding ridge–regression and $\beta = 1$ the lasso. For $1 < \beta < 2$, penalties in this family represent a mixture of the ridge and lasso penalties generating alternatives in between these two extremes.

An extension of this family to non convex members producing sparser solutions than the lasso is

$$P_\beta(\mathbf{a}) = \sum_{j=1}^{n} \log((1 - \beta)\ |\, a_j \,| + \beta)\,; \quad 0 < \beta < 1. \tag{20}$$

As $\beta \to 0$ this approaches the all–subsets penalty ($\gamma = 0$ in (18)) and as $\beta \to 1$ it yields the lasso penalty ($\gamma = 1$ in (18)). Values of β between these extremes bridge all–subsets and the lasso, so that the entire family (19) (20) bridges all–subsets and ridge–regression for $0 < \beta \leq 2$.

For the power family (18) members indexed by a value for γ are "dual" to those indexed by $2 - \gamma$ in the sense that

$$\frac{\partial P_\gamma(\mathbf{a})}{\partial a_k} = \left[\frac{\partial P_{2-\gamma}(\mathbf{a})}{\partial a_k}\right]^{-1}.$$

The choice (20) maintains this duality between the members of the generalized elastic net (19) (20) indexed by β and $2 - \beta$.

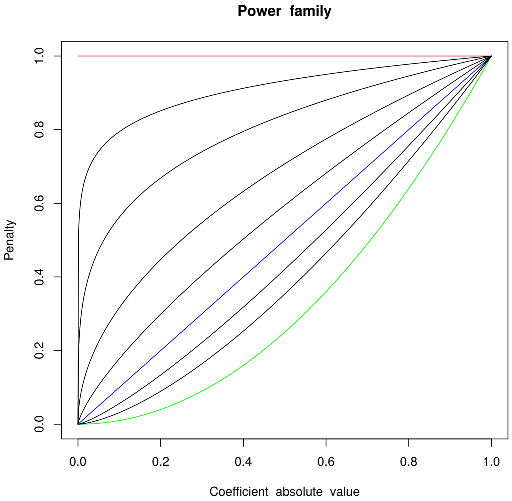

Figure 1: Power family penalties as a function of coefficient absolute value.

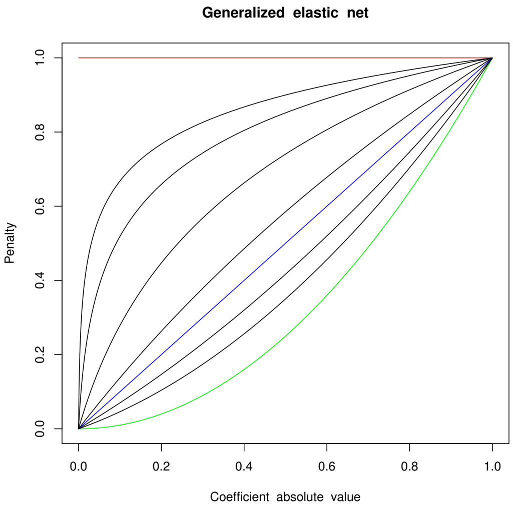

Figure 2: Generalized elastic net penalties as a function of coefficient absolute value.

Figures 1 and 2 compare the power and generalized elastic net families by plotting the partial contribution of each coefficient a_j to the penalty as a function of $|\, a_j \,|$ for selected values of $0 \leq \gamma \leq 2$ (Fig. 1) and for selected values of $0 \leq \beta \leq 2$ (Fig. 2). One sees that the two families produce a similar spectrum of penalties. The principal differences occur at very small coefficient values $|\, a_j \,| \simeq 0$. For $\gamma > 1$ all members of the power family have $[\partial P_\gamma(\mathbf{a})/\partial \,|\, a_j \,|]_{a_j=0} = 0$, and for $\gamma < 1$, $[\partial P_\gamma(\mathbf{a})/\partial \,|\, a_j \,|]_{a_j=0} = \infty$. The former causes all coefficient estimates to be non zero at every point on the path for all convex members except the lasso, and the latter property causes the coefficient paths $\hat{\mathbf{a}}_\gamma(\lambda)$ to have discontinuities as a function of λ for all non convex members. For the generalized elastic net $[\partial P_\beta(\mathbf{a})/\partial \,|\, a_j \,|]_{a_j=0}$ is non zero and finite for all $0 < \beta < 2$. This causes the coefficients to enter (initially become non zero) sequentially with decreasing λ for all $\beta < 2$. It also produces strictly continuous paths for $\beta \geq 1/2$, and smaller discontinuities (jumps) for $0 < \beta < 1/2$. This increases the stability (reduces variance) of the coefficient estimates (Fan and Li 2001).

3 Direct path seeking

A principal limitation of the bridge-regression strategy (16) (17) is the computational burden of obtaining the solutions to (16) for an adequate number of different penalties and corresponding path points at which to perform (17). One approach that mitigates this burden is direct path seeking. The goal is to sequentially construct a path directly in the parameter space that closely approximates that for a given penalty $P(\mathbf{a})$, without having to repeatedly solve numerical optimization problems.

With direct path seeking, solution points on the path $\hat{\mathbf{a}}(\nu)$ are indexed by path length ν. Starting at $\nu = 0$ with some initial point $\hat{\mathbf{a}}(0)$ (usually $\hat{\mathbf{a}}(0) = 0$) each successive point $\hat{\mathbf{a}}(\nu + \Delta\nu)$ is obtained from the previous one $\hat{\mathbf{a}}(\nu)$ by

$$\hat{\mathbf{a}}(\nu + \Delta\nu) = \hat{\mathbf{a}}(\nu) + \mathbf{d}(\nu) \cdot \Delta\nu; \quad \nu \leftarrow \nu + \Delta\nu. \tag{21}$$

Here $\mathbf{d}(\nu)$ is a vector characterizing a direction in the parameter space and $\Delta\nu > 0$ is a specified distance along that direction. These iterations continue until a point $\nu_{\max}$ of minimum empirical risk (8) is reached

$$\nu_{\max} = \arg\min_{\nu > 0} \hat{R}(\hat{\mathbf{a}}(\nu)). \tag{22}$$

This procedure (21) can be viewed as a numerical optimization method for minimizing the empirical risk (8) (22). However the focus here is on the *path* traversed by the procedure from its starting point $\hat{\mathbf{a}}(0)$ to the end point $\hat{\mathbf{a}}(\nu_{\max})$. Different path seeking methods, each intended for a particular loss–penalty combination, specify different prescriptions for calculating $\mathbf{d}(\nu)$ and $\Delta\nu$ at each path point $\hat{\mathbf{a}}(\nu)$. The path produced by this *single* numerical optimization (21) is intended to approximate that produced by (10), for the corresponding loss–penalty combination, obtained by repeatedly solving a large number of numerical optimizations for a sequence of λ values.

Popular path seekers based on squared–error loss (4) include partial least squares regression (PLS, Wold *et al* 1984) which approximates the ridge–regression path (Frank and Friedman 1993), forward stepwise regression intended to approximate the all–subsets path, and least angle regression (Efron *et al* 2004) approximating the lasso path. Gradient boosting (Friedman 2001, Hastie *et al* 2007) is another direct path seeker for the lasso that can be used with any convex loss criterion.

3.1 Generalized path seeking

In order to perform bridge regression (16) (17), fast methods are required for inducing (approximate) paths for a wide variety of penalties, such as all those in the power (18) or generalized elastic net (19) (20) families. In addition, it would be desirable to be able to employ a variety of loss criteria inducing risk functions (8) corresponding to likelihoods for a variety of probability models.

Consider penalties $P(\mathbf{a})$ for which

$$\left\{ \frac{\partial P(\mathbf{a})}{\partial \,|\, a_j \,|} > 0 \right\}_1^n \tag{23}$$

for all values of $\mathbf{a}$. These conditions define a class of penalties where each member in the class is a monotone increasing function of the absolute value of each of its arguments. All members of the power family (18) and generalized elastic net (19) (20) are included in this class. The SCAD penalty (Fan and Li 2001), the MC+ family (Zhang 2007), the grouped lasso (Yuan and Lin 2006), the CAP family (Zhao, Rocha, and Yu 2006), the grouped bridge family (Huang *et al* 2007) and some (reparameterized) smoothness inducing penalties are also included in this class, along with many other penalties that have been, or have yet to be, proposed. The following generalized path seeking algorithm (GPS) can be used to approximate the path corresponding to any penalty in this class in conjunction with any (differentiable) convex loss.

Let ν measure length along the path and $\Delta\nu > 0$ be a *small* increment. Define

$$g_j(\nu) = -\left[\frac{\partial \hat{R}(\mathbf{a})}{\partial a_j}\right]_{\mathbf{a}=\hat{\mathbf{a}}(\nu)}, \tag{24}$$

$$p_j(\nu) = \left[\frac{\partial P(\mathbf{a})}{\partial \,|\, a_j \,|}\right]_{\mathbf{a}=\hat{\mathbf{a}}(\nu)} \tag{25}$$

and

$$\lambda_j(\nu) = g_j(\nu)\,/p_j(\nu). \tag{26}$$

Here $g_j(\nu)$ is the jth component of the negative gradient of the empirical risk (8) evaluated at the path point $\hat{\mathbf{a}}(\nu)$, and $p_j(\nu)$ is the corresponding component of the gradient of $P(\mathbf{a})$ with respect to $|\, a_j \,|$. Note that by assumption (23) all $\{p_j(\nu) > 0\}_1^n$. The components of the vector $\lambda(\nu)$ are the component wise ratios of these two gradients at $\hat{\mathbf{a}}(\nu)$. These lamdas (26) are used to drive the generalized path seeking (GPS) algorithm.

GPS Algorithm

```
1   Initialize: $\nu = 0$; $\{\hat{a}_j(0) = 0\}_1^n$
2   Loop {
3       Compute $\{\lambda_j(\nu)\}_1^n$
4       $S = \{j \mid \lambda_j(\nu) \cdot \hat{a}_j(\nu) < 0\}$
5       if ($S =$ empty) $j^* = \arg\max_j |\lambda_j(\nu)|$
6       else $j^* = \arg\max_{j \in S} |\lambda_j(\nu)|$
7       $\hat{a}_{j^*}(\nu + \Delta\nu) = \hat{a}_{j^*}(\nu) + \Delta\nu \cdot sign(\lambda_{j^*}(\nu)\,)$
8       $\{\hat{a}_j(\nu + \Delta\nu) = \hat{a}_j(\nu)\}_{j \neq j^*}$
9       $\nu \leftarrow \nu + \Delta\nu$
10  } Until $\lambda(\nu) = 0$
```

Line 1 initializes the path. At each step the vector $\lambda(\nu)$ is computed via (24–26) (line 3). At line 4, those non zero coefficients $\hat{a}_j(\nu) \neq 0$ with sign opposite to that of their corresponding $\lambda_j(\nu)$ are identified. If there are none (usual case) the coefficient corresponding to the largest component of $\lambda(\nu)$ in absolute value is selected (line 5). If one or more $\lambda_j(\nu) \cdot \hat{a}_j(\nu) < 0$, then the coefficient with corresponding largest $|\lambda_j(\nu)|$ within this subset is instead selected (line 6). The selected coefficient $\hat{a}_{j^*}(\nu)$ is then incriminated by a small amount in the direction of the sign of its corresponding $\lambda_{j^*}(\nu)$ (line 7) with all other coefficients remaining unchanged (line 8), producing the solution for the next path point $\nu + \Delta\nu$ (line 9). Iterations continue until all components of $\lambda(\nu)$ are zero (line 10). Since each step (lines 7–8) reduces the empirical risk (8), $\hat{R}(\hat{\mathbf{a}}(\nu + \Delta\nu)) < \hat{R}(\hat{\mathbf{a}}(\nu))$, the algorithm will reach an unregularized solution (7) where all $\{\lambda_j(\nu) = 0\}_1^n$ (23–26).

3.2 Motivation

In this section motivation is provided to explain why one might expect the GPS algorithm to closely track the paths produced by (9) (10) for convex risk (8) and penalties satisfying (23). Actual comparisons are presented in Section 4.

Consider the constrained formulation (9). Let $\hat{\mathbf{a}}(t)$ be a solution to (9) at a path point indexed by a value of the constraint threshold t, and $\hat{\mathbf{a}}(t + \Delta t)$ be the solution when the constraint is relaxed by a small amount $\Delta t > 0$. Then $\Delta\hat{\mathbf{a}}(t) = \hat{\mathbf{a}}(t + \Delta t) - \hat{\mathbf{a}}(t)$ is the solution to

$$\Delta\hat{\mathbf{a}}(t) = \arg\min_{\Delta\mathbf{a}} [\hat{R}(\hat{\mathbf{a}}(t) + \Delta\mathbf{a}) - \hat{R}(\hat{\mathbf{a}}(t))] \quad \text{s.t. } P(\hat{\mathbf{a}}(t) + \Delta\mathbf{a}) - P(\hat{\mathbf{a}}(t)) \leq \Delta t. \tag{27}$$

Suppose the path $\hat{\mathbf{a}}(t)$ is a continuous function of t

$$\left\{ \left| \frac{d\hat{a}_j(t)}{dt} \right| < \infty \right\}_1^n, \quad t > 0. \tag{28}$$

Then as $\Delta t \to 0$, assuming (23), (27) can be expressed to first order

$$\begin{aligned} \Delta\hat{\mathbf{a}}(t) = \arg \max_{\{\Delta a_j\}_1^n} & \sum_{j=1}^n g_j(t) \cdot \Delta a_j \\ \text{s.t.} \ \sum_{\hat{a}_j(t)=0} p_j(t) \cdot |\,\Delta a_j\,| &+ \sum_{\hat{a}_j(t)\neq 0} p_j(t) \cdot sign(\hat{a}_j(t)) \cdot \Delta a_j \ \leq \ \Delta t \end{aligned} \tag{29}$$

where

$$g_j(t) = -\left[\frac{\partial \hat{R}(\mathbf{a})}{\partial a_j} \right]_{\mathbf{a}=\hat{\mathbf{a}}(t)}$$

and

$$p_j(t) = \left[\frac{\partial P(\mathbf{a})}{\partial\,|\,a_j\,|} \right]_{\mathbf{a}=\hat{\mathbf{a}}(t)}.$$

Furthermore, suppose that all coefficient paths $\{\hat{a}_j(t)\}_1^n$ are monotonic functions of t

$$\{|\,\hat{a}_j(t+\Delta t)|\, \geq |\,\hat{a}_j(t)\,|\,\}_1^n \tag{30}$$

so that $\{sign(\hat{a}_j(t)) = sign(\,\Delta\hat{a}_j(t))\}_{\hat{a}_j(t)\neq 0}$. Under this (additional) constraint (29) becomes

$$\Delta\hat{\mathbf{a}}(t) = \arg\max_{\{\Delta a_j\}_1^n} \sum_{j=1}^n g_j(t)\cdot \Delta a_j \quad \text{s.t.} \quad \sum_{j=1}^n p_j(t)\cdot |\,\Delta a_j\,| \,\leq \Delta t.$$

This is a linear programming problem with solution

$$\begin{aligned} j^*(t) &= \arg\max_{1\leq j\leq n}\, |\,g_j(t)\,|\,/p_j(t) \\ \Delta\hat{a}_{j^*}(t) &= [g_{j^*}(t)\,/p_{j^*}(t)]\cdot \Delta t \\ \{\Delta\hat{a}_j(t) &= 0\}_{j\neq j^*}. \end{aligned} \tag{31}$$

From (24–26) one sees that the GPS algorithm (lines 5 and 7–8) follows the strategy implied by (31) provided $sign(\lambda_j(\nu)) = sign(\hat{a}_j(t))$ for all $\hat{a}_j(t) \neq 0$. This will be the case at all points for which the GPS and exact paths coincide, as a consequence of the Karush-Kuhn-Tucker (KKT) optimality conditions

$$\lambda_j(t) = \lambda(t)\cdot sign(\hat{a}_j(t)), \quad \hat{a}_j(t)\neq 0, \tag{32}$$

where $\lambda(t) > 0$ is the value of λ in (10) corresponding to t. At the beginning, the exact ($t = 0$) and GPS ($\nu = 0$) paths coincide by construction (line 1). Therefore as long as the exact path (9) remains continuous (28) and monotonic (30) for $t \leq t_0$, in the limit $\Delta\nu \to 0$ ($\Delta t \to 0$) the GPS and exact paths will coincide for $t \leq t_0$.

If the exact path (10) is continuous and monotonic over its entire extent ($\infty \leq \lambda \leq 0$), as is often the case, then the GPS algorithm produces the exact path ($0 \leq \nu \leq \nu_{\max}$) (22) as $\Delta\nu \to 0$. A sufficient (but far from necessary) condition for such total monotonicity is orthogonality of the predictor variables over the training sample (1)

$$\sum_{i=1}^N x_{ij}x_{ik} = 0,\; j\neq k. \tag{33}$$

In this case the GPS algorithm produces the exact path provided the latter is continuous.

3.2.1 Discontinuity

With the generalized elastic net family (19) (20), all members for which $\beta \geq 1/2$ produce continuous paths. For $\beta < 1/2$ the paths are not continuous. There can be jumps at those points (λ values (10)) where each successive variable enters (coefficient initially becomes non zero). This is caused by the variables entering with finite non zero coefficient values at those points. This is illustrated in Fig. 3 for $\beta \in \{0.5, 0.4, 0.25, 0.1\}$ in the case of orthogonal (standardized) predictor variables (33) and squared–error loss (4). Here the exact path solutions for nine coefficients are shown as thick (red) points plotted in terms of fraction of explained risk (8)

$$r(\lambda) = [\hat{R}(\hat{\mathbf{a}}(\infty)) - \hat{R}(\hat{\mathbf{a}}(\lambda))]/\hat{R}(\hat{\mathbf{a}}(\infty)) \tag{34}$$

which is monotonically increasing with decreasing λ along the path $\infty \leq \lambda \leq 0$. For squared–error loss $r(\lambda)$ is the fraction of explained variance $R^2(\lambda)$ of the data values $\{y_i\}_1^N$ (1) at each path point indexed by λ. In Fig. 3 the abscissa is the fraction of explainable variance $r(\lambda)/r(0)$.

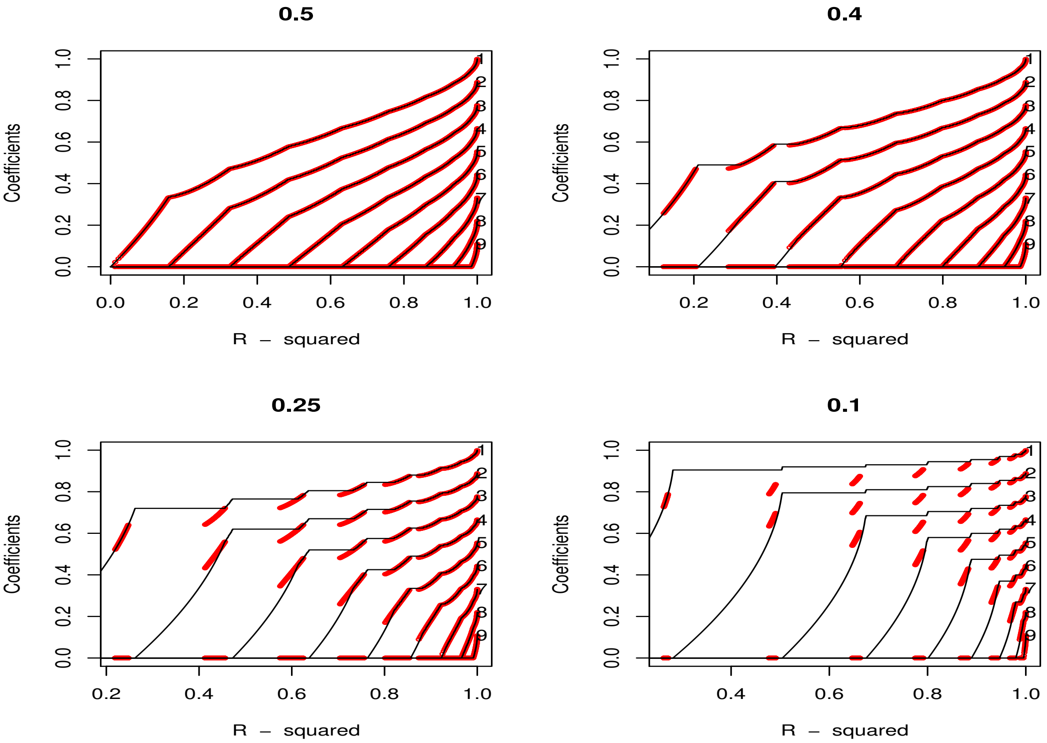

Figure 3: Exact (red) and GPS (black) paths for elastic net non convex penalties $\beta \in \{0.5, 0.4, 0.25, 0.1\}$ with orthogonal predictors.

As seen in Fig. 3 the coefficient paths for $\beta = 0.5$ (upper left panel) are continuous. For $\beta = 0.4$ (upper right panel) discontinuities appear in the coefficient paths for $0 \leq r(\lambda) \lesssim 0.6$; there are values of $r(\lambda)$ in this range at which no exact solution exists. For smaller values of β (lower panels) the discontinuities increase in magnitude and number. For $\beta = 0$ representing all–subsets regression (not shown) there are no continuous sections of the exact path and it reduces to a set of discrete points for each coefficient as a function of $r(\lambda)$ (34).

The thin (black) curves in Fig. 3 show the corresponding paths produced by the GPS algorithm for the same penalty. By construction these paths are continuous at all points for all coefficients. For $\beta = 1/2$ the GPS and exact paths coincide at all points as a consequence of the continuity of the latter. For $\beta < 1/2$ the GPS and exact paths coincide in those regions where the exact paths for *all* coefficients are continuous. In regions where this is not the case the GPS algorithm provides continuous approximations that fairly closely (but not exactly) track the exact paths where solutions for the latter exist. The sparseness properties of the two sets of paths are seen to be quite similar. In the case $\beta = 0$ (all–subsets, not shown) the GPS and exact paths coincide at the (discrete) points representing solutions for the latter. At other path points GPS provides continuous paths that interpolate between the corresponding exact solution points.

As pointed out by Fan and Li 2001, discontinuities in the coefficient paths are undesirable because they lead to instability (increased variance) in the coefficient estimates. In this sense the continuous GPS paths might be preferred on statistical grounds even when the exact paths can be calculated as here in the orthogonal case (33).

3.2.2 Non monotonicity

When all exact coefficient paths $\hat{a}_j(t)$ (9) are continuous, the GPS paths coincide with the exact ones as long as all $\hat{a}_j(t)$ remain monotonic (30). In this case one has from the KKT conditions

(32)

$$\lambda_j(\nu) \cdot sign(\hat{a}_j(\nu)) = \max_k \mid \lambda_k(\nu) \mid, \quad \hat{a}_j(\nu) \neq 0, \tag{35}$$

for all non zero GPS coefficients. If at some point t_0 (ν_0) one or more exact paths $\hat{a}_j(t)$ become non monotonic ($\mid \hat{a}_j(t+\Delta t) \mid \, < \mid \hat{a}_j(t) \mid$), then (32) remains valid for $t > t_0$, whereas (35) need not hold for all GPS coefficient paths. There may be no single variable increment (lines 7–8) that produces the exact solution for $\nu > \nu_0$. So long as all $sign(\lambda_j(\nu)) = sign(\hat{a}_j(\nu))$, $\hat{a}_j(\nu) \neq 0$, GPS will continue to monotonically update the coefficients that satisfy (35), leaving those for which $\mid \lambda_j(\nu) \mid \, < \max_k \, \mid \lambda_k(\nu) \mid$ constant. These are the coefficients for which the exact solutions have become non monotonic. This continues until the corresponding $\lambda_j(\nu)$ for one or more of these variables changes sign. At that point $\lambda_j(\nu) \cdot sign(\hat{a}_j(\nu)) < 0$ and the GPS algorithm (line 6) chooses the coefficient $\hat{a}_{j^*}(\nu)$ corresponding to the most negative $\lambda_j(\nu) \cdot sign(\hat{a}_j(\nu))$ for updating. This update (line 7) causes $\mid \hat{a}_{j^*}(\nu+\Delta\nu) \mid \, < \mid \hat{a}_{j^*}(\nu) \mid$ thereby (belatedly) introducing non monotonicity into the GPS paths of these coefficients.

As long as the set S (line 4) is not empty the coefficients $\{\hat{a}_j(\nu)\}_{j \in S}$ will continue to decrease in absolute value for successive steps while the other coefficients $\{\hat{a}_j(\nu)\}_{j \notin S}$ remain constant. Each update (line 7) causes $\mid \lambda_{j^*}(\nu+\Delta\nu) \mid \, < \mid \lambda_{j^*}(\nu) \mid$ since to first order

$$\Delta \mid \lambda_{j^*}(\nu) \mid \, = -[(h_{j^*}(\nu) + \lambda_{j^*}(\nu) \; q_{j^*}(\nu) \, sign(\hat{a}_{j^*}(\nu)))/p_{j^*}(\nu)] \cdot \Delta\nu \tag{36}$$

where $h_{j^*}(\nu)$ and $q_{j^*}(\nu)$ are the corresponding diagonal elements of the Hessians of $\hat{R}(\mathbf{a})$ (8) and penalty $P(\mathbf{a})$ respectively. Since $h_{j^*}(\nu) > 0$ by convexity, $p_{j^*}(\nu) > 0$ by assumption (23) (25), and $\mid \lambda_{j^*}(\nu) \mid$ is small when $\lambda_{j^*}(\nu)$ initially becomes negative, this quantity (36) is initially small and negative and stays that way as a result of (36). Thus the largest $\mid \lambda_j(\nu) \mid$, $j \in S$, is decreased at each step (line 7) until another (if any) $\mid \lambda_l(\nu) \mid$, $l \in S$, becomes larger. In this way, all coefficients $\{\hat{a}_j(\nu)\}_{j \in S}$ are repeatedly updated, reducing their corresponding $\mid \hat{a}_j(\nu) \mid$ until either $\hat{a}_j(\nu)$ or $\lambda_j(\nu)$ changes sign, or the end of the path is reached (all $\{\lambda_j(\nu) = 0\}_1^n$).

4 Examples

In this section applications of the GPS algorithm to data using generalized elastic net penalties (19) (20) are presented, and compared to the exact paths for the convex members ($\beta \geq 1$).

4.1 Least-squares regression: diabetes data

This data set, used in Efron *et al* 2004, consists of $n = 10$ predictor variables and $N = 442$ observations. The outcome variable is numeric so that squared–error loss (4) was employed.

Figure 4 shows the ten coefficient paths as a function of R^2 (34) for $\beta = 1.9$ (upper left), $\beta = 1.5$ (upper right), $\beta = 1.25$ (lower left), and $\beta = 1$ (lasso, lower right). The red curves are the exact paths, whereas the black ones are the corresponding GPS paths. For $\beta = 1.9$ slight differences are seen to occur for $R^2 \gtrsim 0.45$ where one of the exact coefficient paths becomes non monotonic. For the other penalties the differences are seen to be smaller. As β decreases the solutions become sparser in that for the same degree of data fit as measured by R^2 there tend to be fewer non zero coefficients. That is

$$S(\hat{\mathbf{a}}_\beta(R^2)) \geq S(\hat{\mathbf{a}}_{\beta'}(R^2)), \quad \beta < \beta' \tag{37}$$

with $S(\mathbf{a})$ given by (15) for $\eta = 0$.

Figure 5 shows the GPS paths for the lasso (upper left) and for several non convex generalized elastic net penalties, $\beta = 0.5$ (upper right), $\beta = 0.25$ (lower left), and $\beta = 0$ (lower right), plotted on the same vertical scale. Here on sees a similar pattern of further increasing sparsity (37) as $\beta < 1$ decreases.

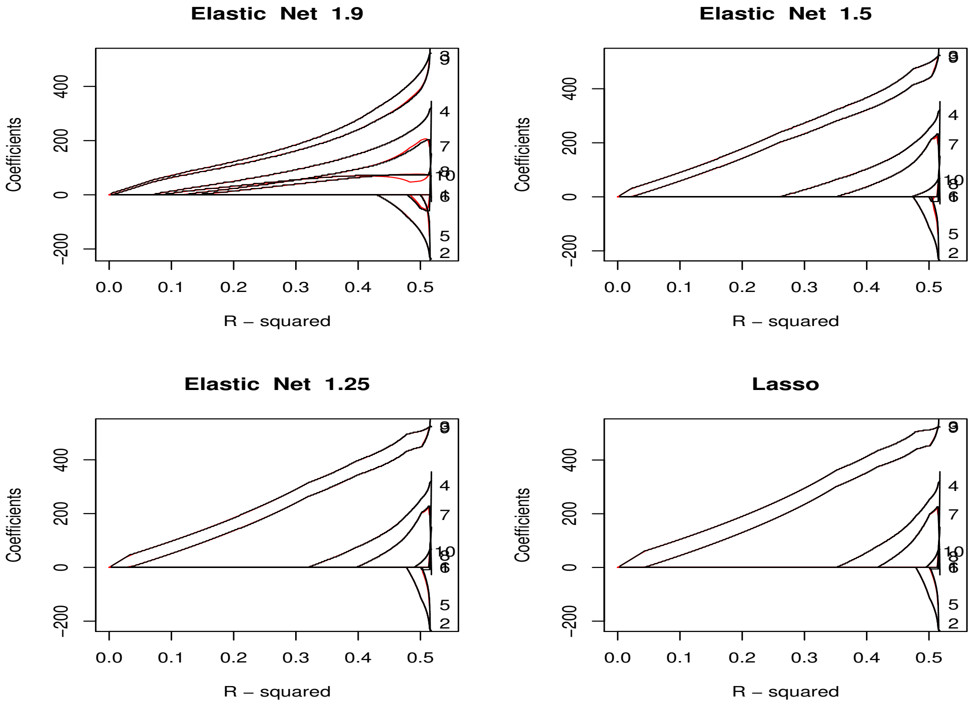

Figure 4: Exact (red) and GPS (black) paths for the diabetes data using convex elastic net penalties $\beta \in \{1.9, 1.5, 1.25, 1.0\}$.

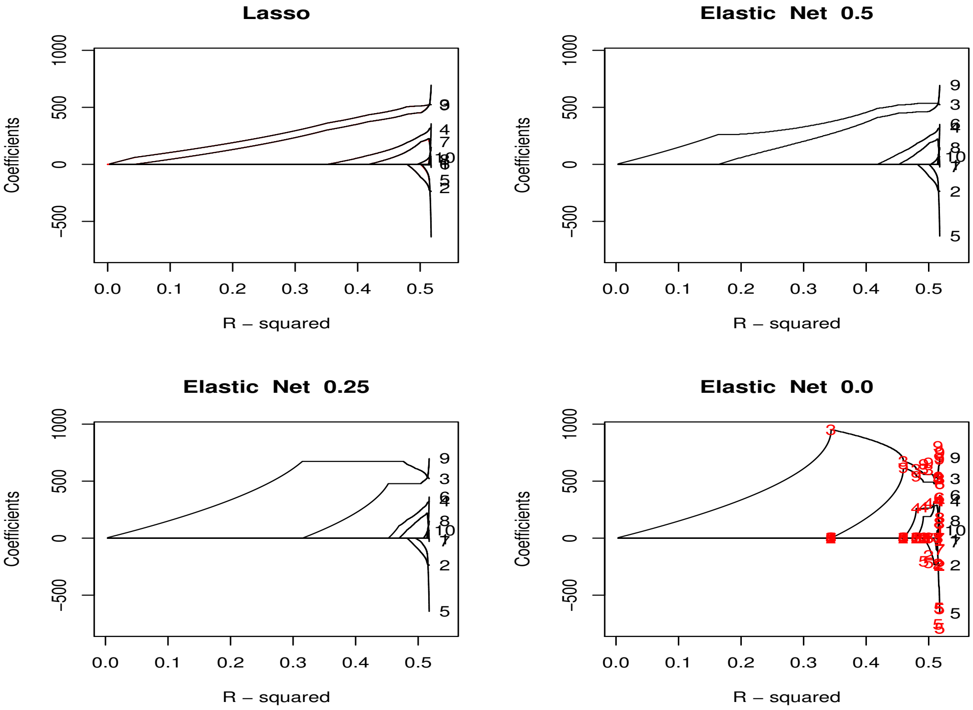

Figure 5: Paths for Lasso, and GPS non convex elastic net penalties $\beta \in \{0.5, 0.25, 0.0\}$, for the diabetes data. The red numbers indicate the forward stepwise solutions.

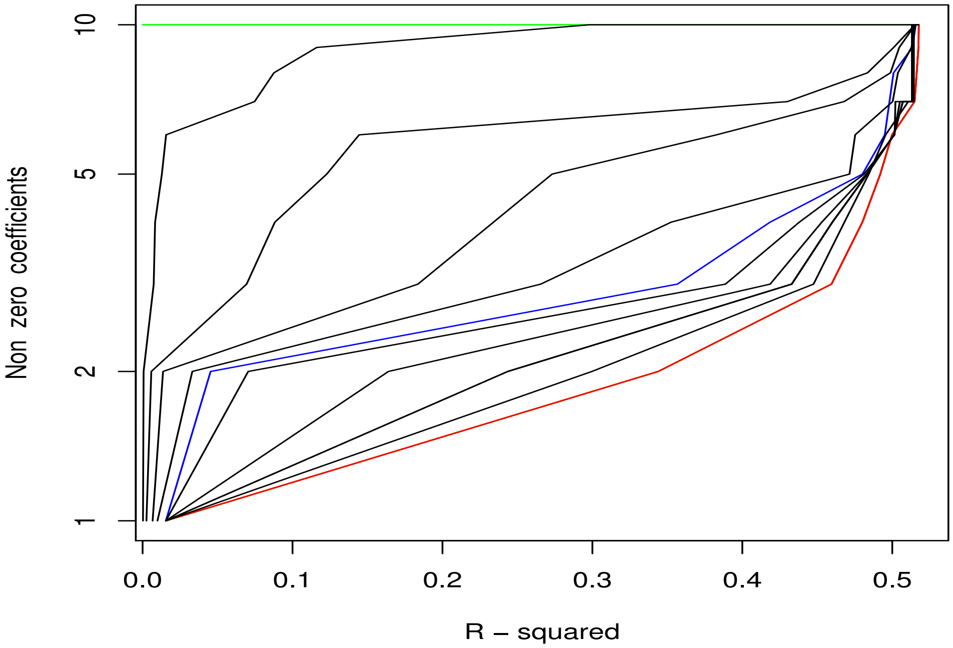

Figure 6: Number of non zero coefficient estimates along respective paths for diabetes data using elastic net penalties $\beta \in \{2.0$(ridge, green), 1.99, 1.9, 1.7, 1.5, 1.0 (lasso, blue), 0.7, 0.5, 0.4, 0.3, 0.0$\}$ and stepwise (red).

The red numbers in the lower right panel of Fig. 5 represent the (discrete) path points for forward stepwise regression. Here one sees that the $\beta = 0$ GPS and stepwise paths coincide at the stepwise solutions. At other points the GPS paths are continuous, interpolating between the stepwise solutions. This is not always the case. For $\beta = 0$ the GPS paths interpolate the discrete path points generated by "*state*wise" regression. At each step, statewise regression successively selects the variable not in the model that is most correlated with the current residuals to next include in the model. It then performs a full multiple regression on the current variable set to obtain the solution coefficients. As a variable selection technique this can be slightly less aggressive than forward *step*wise regression which selects each successive variable that gives the best multiple regression fit, given the variables that have already entered. In many situations the two procedures give identical results (as here), but this is not always the case. However, the results of the two procedures are seldom very different especially for the larger estimated coefficients.

Figure 6 shows the number of non zero coefficients as a function R^2 along the path for a larger set of generalized elastic net penalties. This number is inversely related to sparsity (15) for $\eta = 0$. The results for each penalty are connected by straight lines to aid visualization. Results for forward stepwise regression (red) are also included, which here are identical to that of $\beta = 0$ GPS. Results for $\beta = 1$ (lasso) and $\beta = 2$ (ridge–regression) are highlighted as well (blue and green). From Fig. 6 one sees that at $R^2 \simeq 0.45$, stepwise regression enters 3 variables, the lasso 4 variables, and ridge–regression all 10 variables. Using these curves as an inverse measure of sparsity, one sees a strict monotonicity among the members of this family. Smaller β produces sparser solutions at every point on the path as indexed by degree of data fit (R^2).

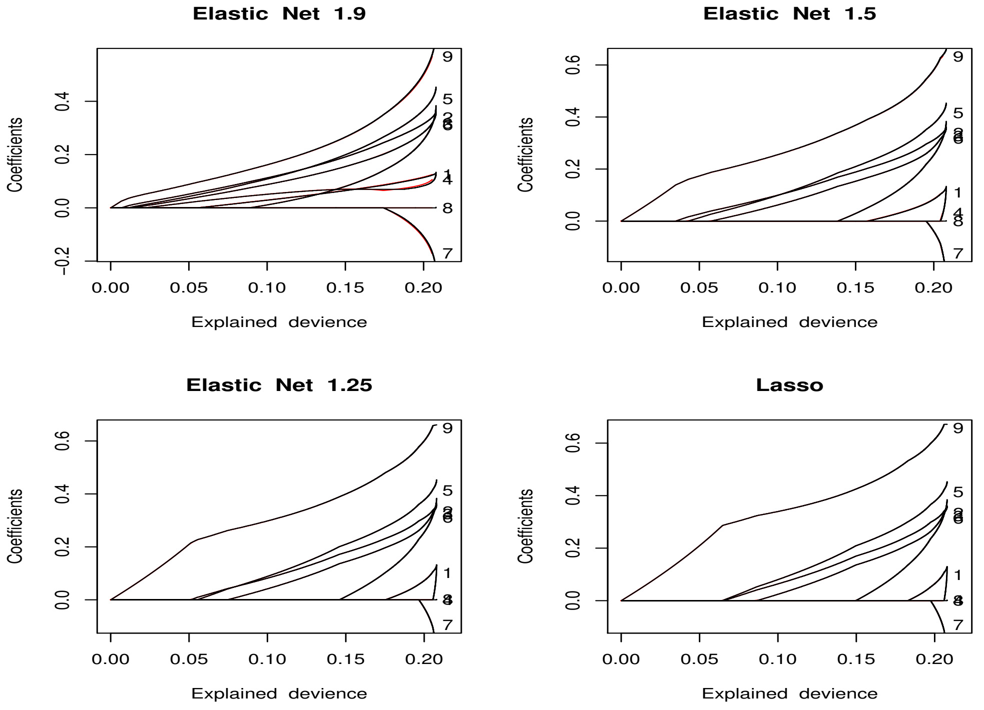

Figure 7: Exact (red) and GPS (black) paths for the heart transplant data using convex elastic net penalties $\beta \in \{1.9, 1.5, 1.25, 1.0\}$.

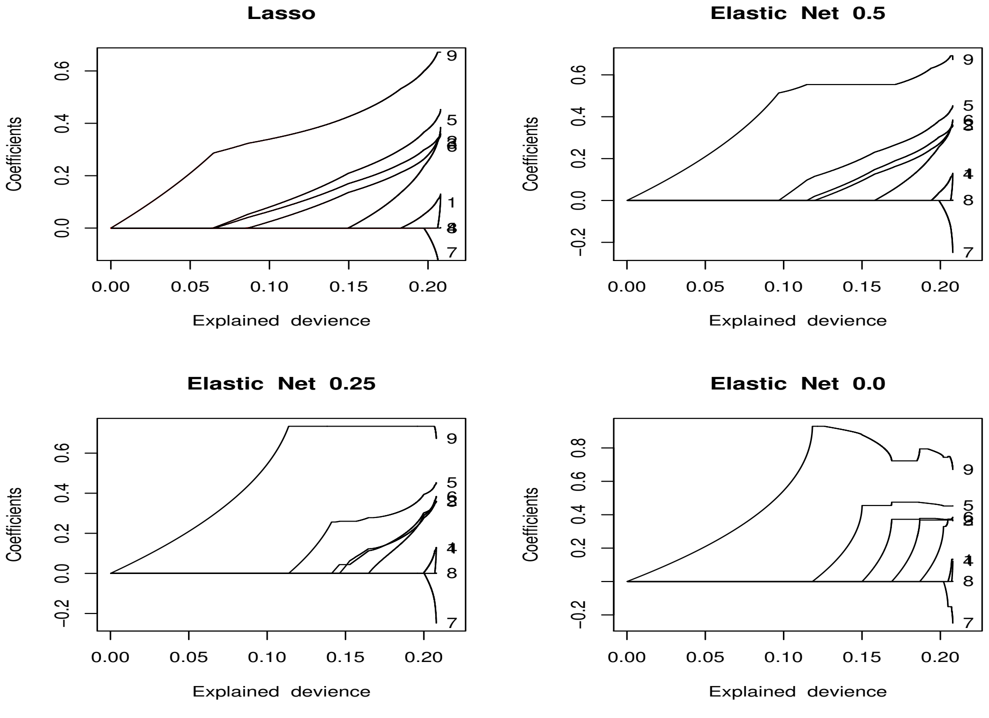

Figure 8: Paths for lasso, and GPS elastic net non convex penalties $\beta \in \{0.5, 0.25, 0.0\}$, for the heart transplant data.

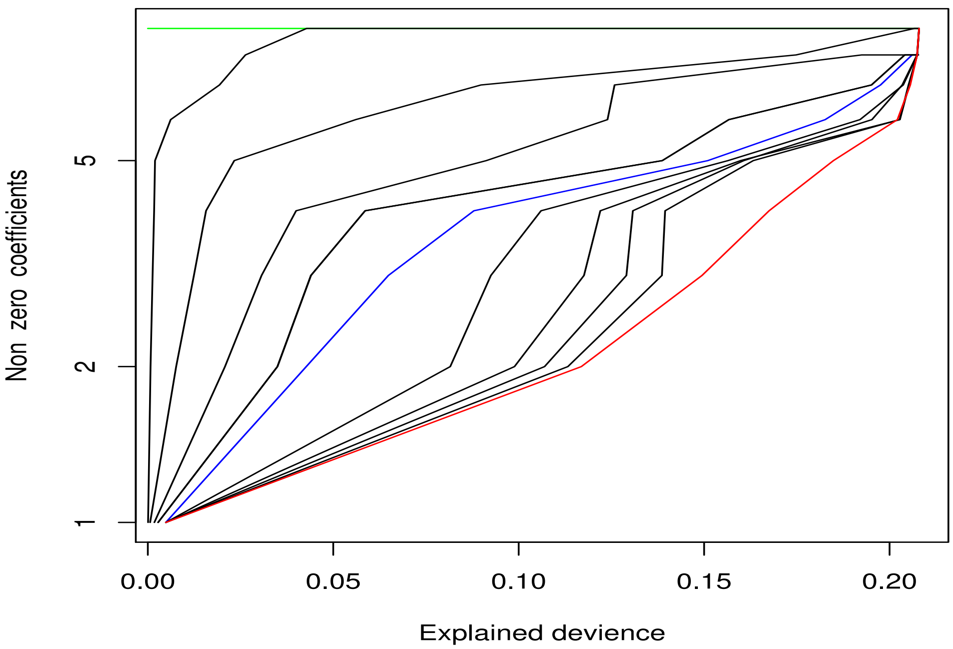

Figure 9: Number of non zero coefficient estimates along respective paths for heart transplant data using elastic net penalties $\beta \in \{2.0$(ridge, green), 1.99, 1.9, 1.7, 1.5, 1.0 (lasso, blue), 0.7, 0.5, 0.4, 0.3, 0.0 (red)$\}$

4.2 Logistic regression: South African heart transplant data

This data set was presented in Hastie, Tibshirani and Friedman 2001. It has $n = 9$ predictor variables and $N = 462$ observations. The outcome variable is binary so logistic loss (5) is appropriate.

Figure 7 compares the exact and GPS coefficient paths for selected convex members of the generalized elastic net for this data set. The paths are here indexed by fraction of explained deviance (5) (8) (34). As with squared–error loss (Fig. 4) the GPS paths closely track those for the exact solutions and become sparser for smaller β.

Figure 8 repeats the lasso for comparison, and shows the results for the same non convex penalties as in Fig. 5. Again sparsity is seen to continue to increase with decreasing $\beta < 1$. Figure 9 shows the number of non zero coefficients as a function of explained deviance along the path. As in the squared–error loss case (Fig. 6) there is a strict monotonicity; smaller β produces sparser solutions at every point on the path as indexed by degree of data fit, here measured by explained deviance.

4.3 Least-squares regression: under-determined problem

The above two examples are highly over-determined in that the number of observations N is much larger than the number of predictor variables n. In such cases regularization is much less important than it is for highly under-determined problems where $N << n$. In this section a highly under-determined regression problem is considered. There are $N = 200$ observations and $n = 10000$ predictor variables. The data are simulated from the model

$$y_i = \sum_{j=1}^{n} a_j^* x_{ij} + \varepsilon_i \tag{38}$$

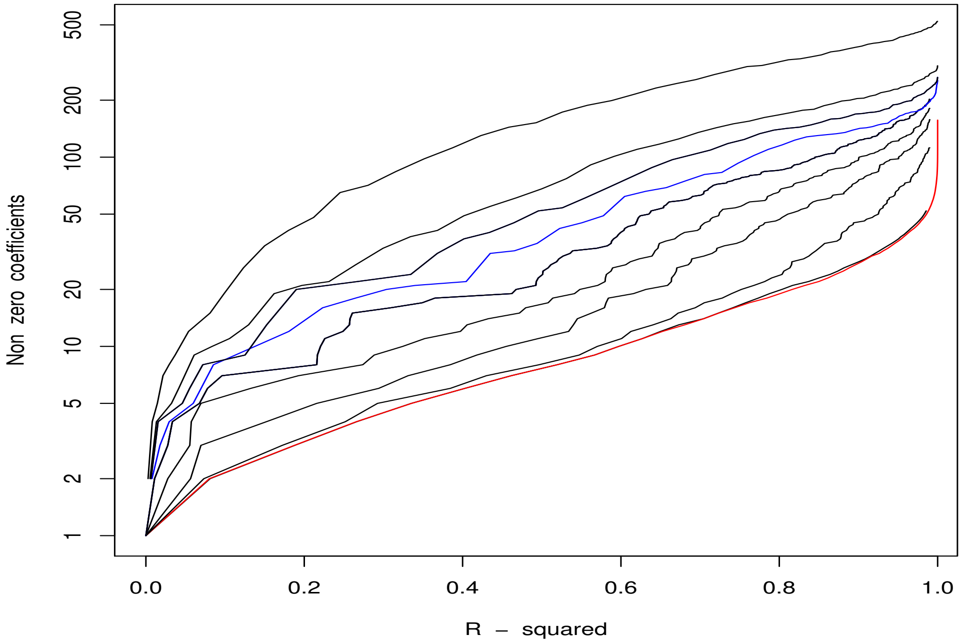

Figure 10: Number of non zero coefficient estimates along respective paths for the under-determined regression example ($n = 10000$, $N = 200$) using elastic net penalties $\beta \in \{1.9,\ 1.7,\ 1.5,\ 1.0$ (lasso, blue), $0.5,\ 0.3,\ 0.2,\ 0.1,\ 0.0\}$, and stepwise (red).

where the predictor variables are randomly drawn from a normal distribution $\mathbf{x}_i \sim N(0, \mathbf{C})$; with covariance matrix elements $C_{jj} = 1,\ \ C_{jk} = 0.4,\ j \neq k$. The random error is also normally distributed $\varepsilon_i \sim N(0, \sigma^2)$, with the value of σ set to produce a 3/1 signal to noise ratio. The optimal coefficient vector $\mathbf{a}^*$ (6) has 30 non zero coefficients with uniformly distributed absolute values $|\,a_j^*| = [31 - j]_+$, and alternating signs $sign(a_{j+1}^*) = -sign(a_j^*)$, $1 \leq j \leq 29$.

Figure 10 shows the number of non zero coefficients as a function of R^2 along the path for forward stepwise regression (red) and a selected set of generalized elastic net penalties. The $\beta = 1$ (lasso) penalty is colored blue. One sees the same monotonic relation between the value of β and the sparsity of the induced path. At $R^2 = 0.9$ on the training data, stepwise and $\beta = 0$ GPS have 15 non zero coefficients, the lasso has 120, and $\beta = 1.9$ elastic net has almost 400. Thus, by varying β one can exercise sharp control over the sparsity of the induced solutions. Note that the $\beta = 0$ and stepwise results are here slightly different for $R^2 > 0.75$.

4.4 Logistic regression: under-determined problem

The data for this problem are similar to that in the previous section. There are $N = 200$ observations and $n = 10000$ predictor variables generated from the same model. The outcome variable has two values $y \in \{-1, 1\}$ with the log–odds given by

$$\log[\Pr(y = 1)/\Pr(y = -1)] = s \cdot \sum_{j=1}^{n} a_j^* x_{ij}. \tag{39}$$

The value for s was chosen to produce a Bayes error rate of 0.05. Here the optimal coefficient vector has 15 non zero coefficients with uniformly distributed absolute values and alternating signs.

Figure 11 shows the number of non zero coefficients as a function of fraction of deviance explained along the path for the same selected set of generalized elastic net penalties as in

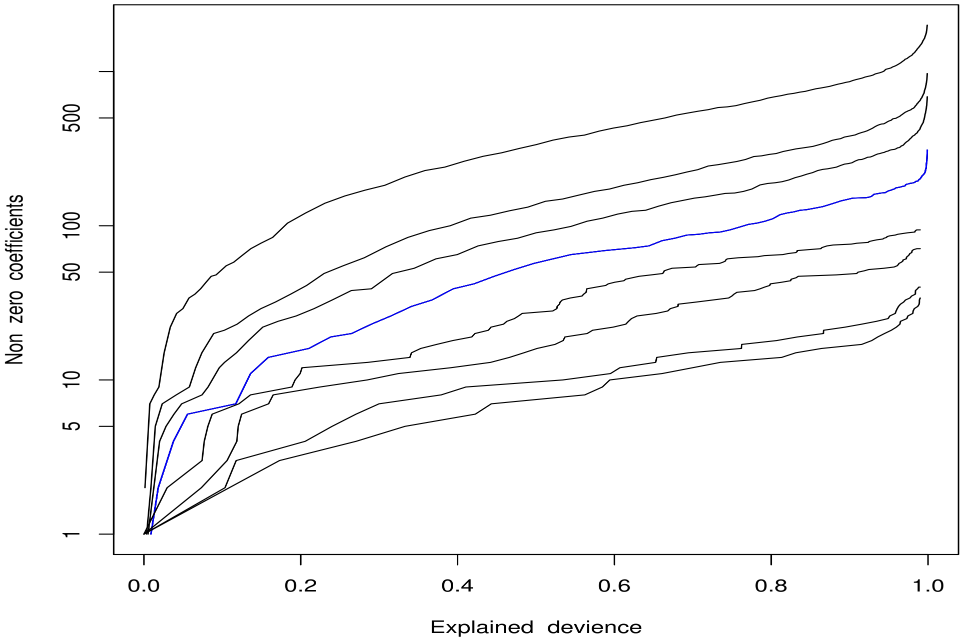

Figure 11: Number of non zero coefficient estimates along respective paths for the under-determined logistic regression example ($n = 10000$, $N = 200$) using elastic net penalties $\beta \in \{1.9, 1.7, 1.5, 1.0$ (lasso, blue), $0.7, 0.5, 0.3, 0.0\}$,

Fig. 10, with $\beta = 1$ (lasso) colored blue. Again the sparsity of the solutions along the path is monotonically related the value of β. At 95% explained deviance, the $\beta = 0$ path includes 10 non zero coefficients, the lasso has 120, and the $\beta = 1.9$ elastic net path has almost 1000.

5 Speed

As the examples illustrate, one can control the sparsity of regularized regression solutions (9) (10) by appropriately selecting penalties $P(\mathbf{a})$. Since the sparsity (15) of of the optimal coefficients (6) is generally unknown, bridge-regression (16) (17) can be used to estimate the best penalty, provided the resulting computational burden is not excessive.

Table 1

Time in seconds for computing 500 path points for GPS and exact convex algorithms with $n = 10000$, $N = 200$.

Penalty β	Sq–err GPS	Sq–err exact	Logistic GPS	Logistic exact
0.0	0.37	1.82*	1.43	
0.1	0.47		1.43	
0.2	0.55		1.44	
0.5	0.62		1.44	
1.0	0.56	3.58	1.34	6.16
1.5	0.69	2.97	1.35	5.42

* forward stepwise

Table 1 shows the computation time is seconds (column 2) required by the GPS algorithm to generate paths (500 points) for the $n = 10000$, $N = 200$ problem described in Section 4.3, for several generalized elastic net penalties (19) (20) as indexed by β (column 1). The third column (rows 5 and 6) show corresponding exact path times (500 path points) for the convex penalties ($\beta \geq 1$) using the fastest known convex optimization methods for these particular problems (Friedman *et al* 2007). The entry in the first row of column 3 is for the (approximate) stepwise path.

The corresponding entries in the last two columns of Table 1 are for the logistic regression problem of Section 4.4. To facilitate comparison with the squared–error results (columns 2 and 3) the optimal coefficient vector $\mathbf{a}^*$ used in (39) was here taken to have 30 non zero coefficients with uniformly distributed absolute values and alternating signs. Again the entries in column 6 are the exact path times using the fastest convex optimization method for elastic net logistic regression (Friedman, Hastie and Tibshirani 2008).

As see from Table 1 bridge-regression is quite feasible for problems of this size. Performing 10–fold cross-validation to evaluate 500 path points for each of these six penalties would require 35 seconds for squared–error loss regression and 84 seconds for logistic regression. This is equivalent to solving 30000 optimization problems in (16), most of which are non convex. For the convex penalties ($\beta \geq 1$) one can (if desired) use the corresponding fast exact algorithms here increasing the times to 51 seconds and 170 seconds respectively, which are still quite feasible.

For $n >> N$ the computation for GPS scales roughly as $n \cdot N$ so that bridge-regression with much larger problems is still quite feasible. GPS computation is roughly independent of the particular loss–penalty combination used. As seen in Table 1 however, logistic regression is somewhat slower than squared–error loss due to the computation of transcendental functions required to evaluate the gradient (24). For the convex elastic net (including the lasso) special fast exact algorithms are available that are competitive in speed with GPS as seen in Table 1. However, there are no such competitive algorithms for the non convex members (20). In general non convex optimization is far more difficult than convex optimization. It often requires a convergent series of iterated convex optimizations that may converge to suboptimal local minima. And even in the convex realm there are many loss–penalty combinations for which special fast exact algorithms competitive with GPS do not exist.

6 Utility

The results presented in Section 5 show that bridge-regression using GPS is computationally tractable for fairly large problems. In this section its potential statistical advantages are investigated.

6.1 Least-squares regression

For regression the (lack of) quality of a particular coefficient path $\hat{\mathbf{a}}(\rho)$, as indexed by its path points ρ, can be measured by

$$\min_{\rho} [R(\hat{\mathbf{a}}(\rho)) - R(\mathbf{a}^*)]/R(\mathbf{a}^*) \tag{40}$$

where $R(\mathbf{a}^*)$ is the minimum possible risk associated with the problem (6). This quantity is the minimal distance (11) between points on the path and the optimal solution $\mathbf{a}^*$, scaled by $1/R(\mathbf{a}^*)$. As discussed in Section 2.1, paths $\hat{\mathbf{a}}(\rho)$ that produce smaller values for (40) have the potential for producing more accurate predictions given a model selection procedure such as cross–validation.

Figure 12 shows the distribution of (40) (boxplots) for paths produced by several squared–error loss (4) regression methods, based on 50 data sets randomly drawn from the model described in Section 4.3. The methods are (left to right) forward stepwise regression, GPS using (non convex) generalized elastic net penalties $\beta \in \{0.0, 0.1, 0.2, 0.5\}$ (20), and the exact paths produced by the lasso ($\beta = 1$) and elastic net with $\beta = 1.5$.

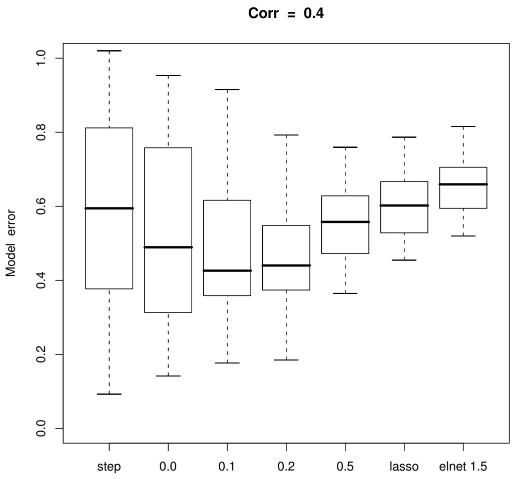

Figure 12: Inaccuracy of stepwise, several non convex elastic net GPS, and convex exact paths, over 50 simulated regression data sets with $n = 10000$, $N = 200$ (Section 6.1).

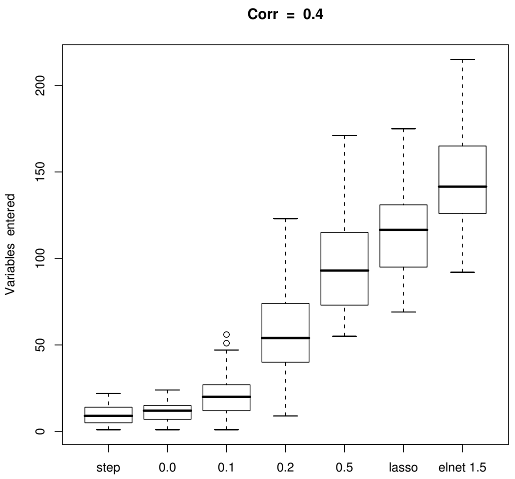

Figure 13: Number of non zero coefficients at optimal solutions for stepwise, several non convex elastic net GPS, and convex exact paths, over the 50 simulated data sets.

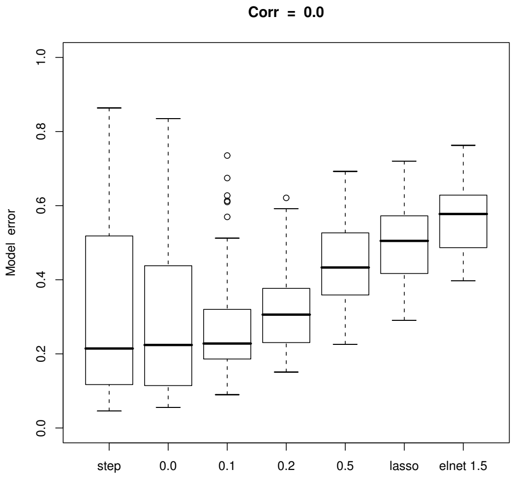

Figure 14: Inaccuracy of stepwise, several non convex elastic net GPS, and convex exact paths over 50 simulated data sets with population uncorrelated predictors.

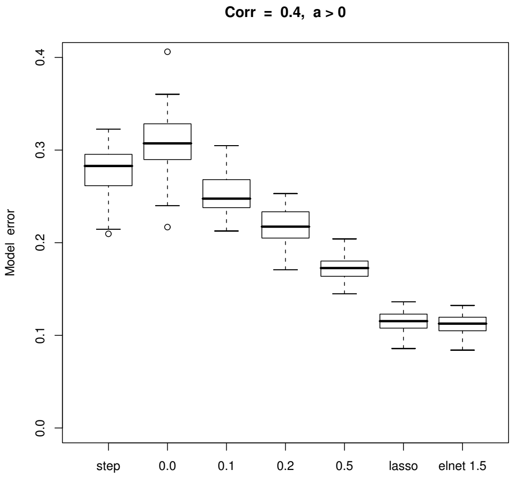

Figure 15: Inaccuracy of stepwise, several non convex elastic net GPS, and convex exact paths, over 50 simulated data sets, all optimal coefficients having the same sign.

From Fig. 12 one sees that the lasso and $\beta = 1.5$ elastic net consistently yield inferior paths on this very sparse problem (30 out of 10000 true non zero coefficients). The forward stepwise procedure yields paths of similar accuracy (40) to the lasso on average, but with much more variability. Stepwise paths based on some data sets are considerably better than those produced by the lasso, and some others are considerably worse. For the GPS paths variability decreases with increasing β. Expected performance is best for $\beta = 0.1$ or $\beta = 0.2$, with the latter having less variability.

Figure 13 shows the distribution of the number of non zero coefficients at the optimal path points minimizing (40) for each of the respective methods. Here one sees that forward stepwise regression typically has around 12 non zero coefficients at its optimal solutions. The slightly less aggressive $\beta = 0$ GPS paths average 14. As β increases, the optimal points on GPS paths tend towards less sparsity. The lasso and $\beta = 1.5$ elastic net produce even denser optimal solutions, typically involving 125 and 150 non zero coefficients respectively. Here one sees that penalty choice closely controls the sparsity of the solutions with sparse ($\beta = 0.1$ or 0.2), but not the sparsest ($\beta = 0$ or stepwise), being the best.

Figure 14 show results analogous to those in Fig. 12 for a slightly modified problem. Here for each of the 50 data sets, the predictor variables are drawn from a standard normal distribution, $\mathbf{x}_i \sim N(0, \mathbf{I})$. That is, the variables are uncorrelated with respect to their (population) joint distribution. All other aspects of the generating model are the same. For this problem all methods produce better paths, as measured by (40), with the sparsest procedures improving the most. Again the forward stepwise procedure produces the least stable paths in terms of variability, with the $\beta = 0$ GPS path being almost as unstable. The stability of the paths produced by the other procedures are all about the same. The results for this problem are qualitatively similar to those shown in Fig. 12, with the best stability–expected performance trade–off appearing to be for the $\beta = 0.1$ GPS path. The distributions of the number of non zero coefficients for the optimal solutions of each of the methods (not shown) is quite similar to that shown in Fig. 13.

Figure 15 shows analogous results to those in Fig. 12 for a slightly different modification of the problem. Here the joint distribution of the predictor variables is as described in Section 4.3, as is all other aspects of the model, except that the signs of the optimal non zero coefficients are taken to be the same ($sign(a^*_{j+1}) = sign(a^*_j) > 0$), instead of alternating. Here one sees a very different pattern of results. All methods produce much more stable paths. However their relative quality (40) is reversed; the worst methods in the previous two problems are here the best and vice versa. The lasso and the $\beta = 1.5$ elastic net paths here dramatically out–perform methods producing sparser solutions. The distributions of the number of (optimal solution) non zero coefficients for each of the methods (not shown) is for this problem again quite similar to that shown in Fig. 13. The optimal $\beta = 1.5$ elastic net solutions, typically involving 150 non zero coefficients, are far more accurate than methods producing much sparser solutions, even though the population optimal coefficient vector $\mathbf{a}^*$ has only 30 non zero entries.

The results shown in Figs. 12–14 show that the accuracy of a given method for the same population joint distribution can strongly depend on the particular training data set realized from that distribution. This is especially the case for methods that induce very sparse paths. In Figs. 12 and 14 one sees that the sparsest methods often produce much better solutions than denser methods on particular data sets, and much worse on others. Thus, comparisons based on one or a small number of data sets (simulated or real) can be highly misleading.

A somewhat surprising result from the above examples is that the optimal sparsity of the *estimated* coefficients depends upon more than just the sparsity of the optimal coefficients $\mathbf{a}^*$ (6) characterizing the problem. In all three examples the sparsity (15) of $\mathbf{a}^*$ was the same; 30 out of 10000 non zero coefficients. In fact, all $\{|\, a^*_j \,|\}_1^n$ were identical. For the situation shown in Fig. 12 the best solutions typically involved 50 non zero coefficients whereas for that shown in Fig. 14 penalties producing around 20 were best. For the situation shown in Fig. 15 the densest method being considered produced the most accurate solutions with 150 non zero coefficients on average.

The penalties used here depend only on the absolute values of the coefficients. One might then expect that the best penalty would depend mainly on the relative absolute values of the optimal coefficients $\{|\, a^*_j \,|\}_1^n$ (6). Thus, as discussed in Section 2.3, this knowledge (if available) would drive penalty choice. The examples presented here show that this is not the case. The relative signs of the optimal coefficients as well as the correlational structure of the predictor variable distribution also influence which such penalty is best. For example, methods that induce

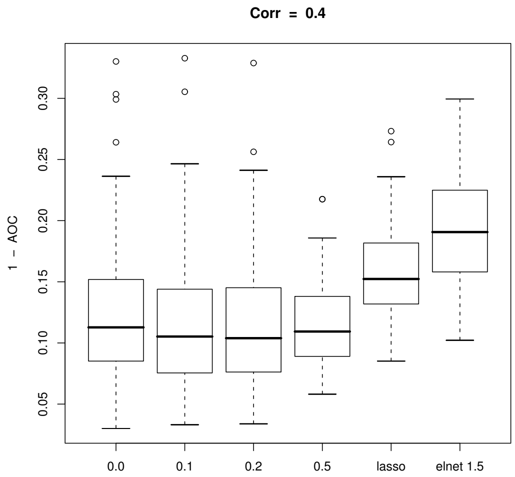

Figure 16: 1-AUC of several non convex elastic net GPS, and convex exact paths, over 50 simulated logistic regression data sets with $n = 10000$, $N = 200$ (Section 6.2).

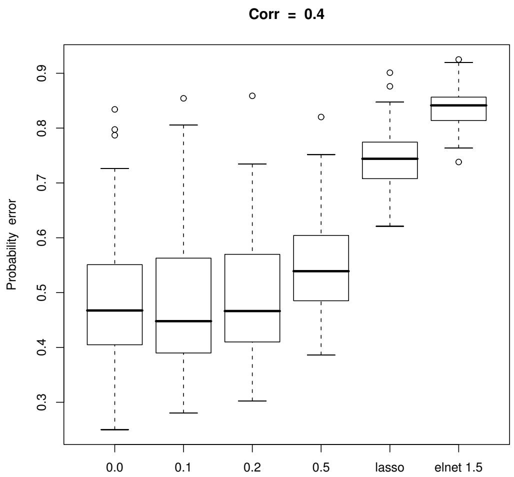

Figure 17: Probability estimation error of several non convex elastic net GPS, and convex exact paths, over the 50 simulated logistic regression data sets.

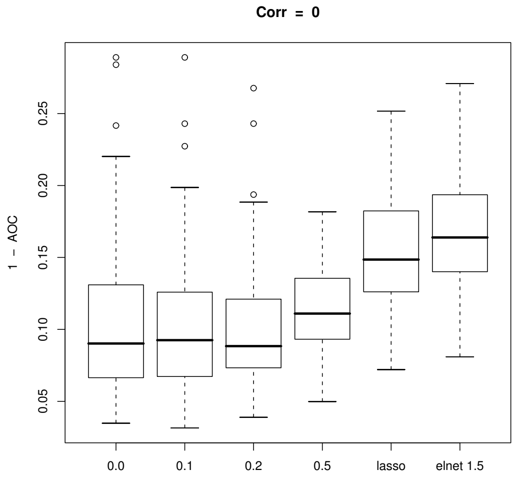

Figure 18: 1-AUC of several non convex elastic net GPS, and convex exact paths, over 50 simulated logistic regression data sets ($n = 10000$, $N = 200$), with population uncorrelated predictors.

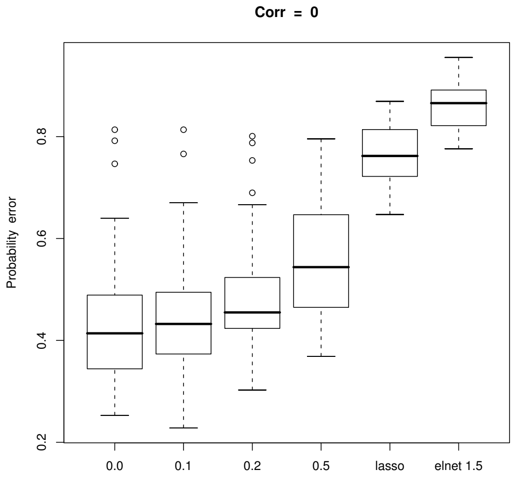

Figure 19: Probability estimation error of several non convex elastic net GPS, and convex exact paths, over 50 simulated logistic regression data sets, with population uncorrelated predictors.

sparser solutions than the lasso are not always better than the lasso solutions, even in sparse situations as characterized by the optimal coefficients $\mathbf{a}^*$. Even with knowledge of the latter, the other aspects of the problem that influence choice of a good penalty are likely to be unknown. Thus, using bridge regression to aid penalty choice can be helpful.

6.2 Logistic regression

In this section classification problems for which the outcome variable is dichotomous, $y \in \{-1, 1\}$, are considered. In this case logistic regression based on loss (5) is usually employed. An often used figure of merit for a classifier is the area under its ROC curve (AUC). The AUC of a prediction function F is the probability its value for a randomly chosen positive instance ($y = 1$) is larger than a randomly chosen negative one ($y = -1$) . Using this measure, the (lack of) quality for classification of a particular coefficient path $\hat{\mathbf{a}}(\rho)$, as indexed by its path points ρ, would be

$$\min_{\rho}[1 - AUC(\hat{\mathbf{a}}(\rho))] \tag{41}$$

where $AUC(\mathbf{a})$ is the AUC of the function $F(\mathbf{x}; \mathbf{a})$ (2).

Figure 16 shows the distribution of (41) for paths produced by several logistic regression methods, based on 50 data sets randomly drawn from the model described in Section 4.4. The methods are (left to right) GPS using non convex generalized elastic net penalties $\beta \in \{0.0, 0.1, 0.2, 0.5\}$ (20), and the exact paths produced by the convex lasso ($\beta = 1$) and elastic net with $\beta = 1.5$. Here one sees very little differentiation between the GPS paths based on the various non convex penalties. The exact lasso and $\beta = 1.5$ elastic net paths are somewhat inferior.

In addition to classification, logistic regression is often used to estimate the probability that an instance realizes a positive outcome, $\Pr(y = 1 \,|\, \mathbf{x})$. This is related to the prediction function $F(\mathbf{x}; \mathbf{a})$ through

$$\Pr(y = 1 \,|\, \mathbf{x}) = Q(\mathbf{x}, \mathbf{a}) = \left(1 + e^{-F(\mathbf{x};\mathbf{a})}\right)^{-1}. \tag{42}$$

A measure of the lack of quality for probability estimation is the relative average absolute error

$$E_{\mathbf{x}} \,|\, Q(\mathbf{x}, \mathbf{a}^*) - Q(\mathbf{x}, \hat{\mathbf{a}}(\rho^*)) \,|\, / \, E_{\mathbf{x}} \,|Q(\mathbf{x}, \mathbf{a}^*) - 0.5 \,| \tag{43}$$

where ρ^* is the minimizer of 41. The denominator in (43) is the average absolute error in always predicting the null probability $\Pr(y = 1) = 0.5$.

Figure 17 shows the distributions of (43) for the same set of methods. Here one sees a sharper differentiation with the sparest methods producing considerably more accurate probability estimates.

Figures 18 and 19 show the corresponding results for (population) uncorrelated predictor variables, $\mathbf{x}_i \sim N(0, \mathbf{I})$. In this case one sees results similar to those in Figs. 16 and 17 with a little sharper distinction between the methods. Here the methods producing the sparsest paths perform best.

Finally, Figs. 20 and 21 show results for (population) correlated predictor variables and all optimal non zero coefficients $\{a_j^*\}_1^{30}$ having the same sign. The reversal seen in the regression examples (Fig. 15) is evident here as well. In terms of probability estimation error (43), and especially AUC (41), the denser methods perform considerably better than sparser ones here, in direct contrast to the other two situations (Figs. 16–19).

The results for logistic regression (Figs. 16–21) reflect those for least-squares regression (Figs. 12–15). The best method (penalty) can strongly depend on the characteristics of the application at hand. In particular, the best sparsity for the coefficient estimates depends on aspects of the problem other than just the sparsity of the optimal coefficients $\mathbf{a}^*$ (6). Estimating non zero values for many coefficients for which $a_j^* = 0$ can sometimes dramatically increase prediction accuracy (Figs. 15, 20, 21).

7 Post–processing selectors

As illustrated in Section 6, there are some applications where methods producing sparser solutions than is possible with convex penalties achieve higher prediction accuracy. One reason

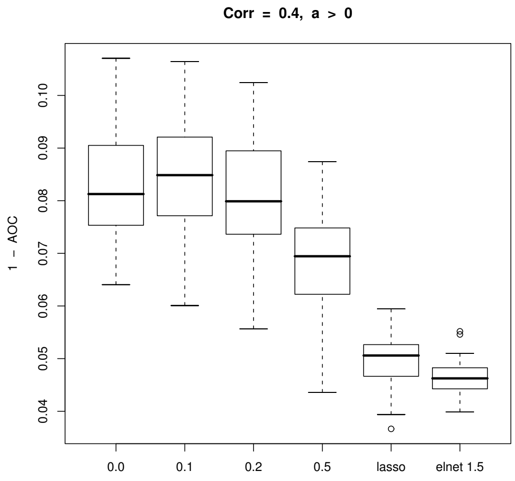

Figure 20: 1 - AUC of several non convex elastic net GPS, and convex exact paths, over 50 simulated logistic regression data sets ($n = 10000$, $N = 200$), with all optimal coefficients having the same sign.

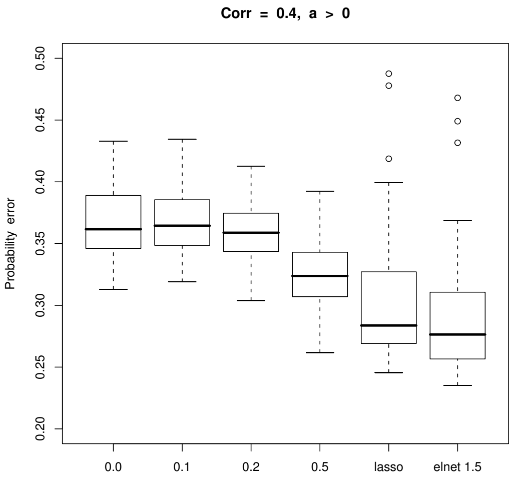

Figure 21: Probability estimation error of several non convex elastic net GPS, and convex exact paths, over 50 simulated logistic regression data sets, with all optimal coefficients having the same sign.

for this is the shrinkage of the absolute values of the coefficient estimates inherent in most regularized regression procedures (10). For the generalized elastic net penalties the value of λ controls both the number of non zero coefficients (sparsity) and the degree of shrinkage of the absolute values of those estimated to be non zero. For path points λ with the same number of non zero coefficients (same sparsity), penalties indexed by larger values of β induce more shrinkage. Thus, there is a sparsity–shrinkage trade–off associated with choice of penalty (β).

For convex penalties ($\beta \geq 1$) high sparsity solutions tend to involve heavy shrinkage of their non zero coefficient estimates. This can induce large bias if the optimal coefficients $\mathbf{a}^*$ are also very sparse. In order to reduce this bias, the optimal solutions (40) (41) (43) for these convex penalties trade decreased sparsity for decreased shrinkage in an attempt to overcome this bias. Non convex penalties ($\beta < 1$) shrink less for the same sparsity thereby producing sparser less biased optimal solutions as illustrated in Fig. 13. This can sometimes improve performance as illustrated in Figs. 12, 14, 16–19.

With uncorrelated predictor variables (33) the paths induced by all generalized elastic net penalties have the property that at all path points of the same sparsity (15), the identities of the non zero solution coefficients are identical. That is, they all enter the variables into their respective regression models in the same order. The only differences are the degree of shrinkage of the absolute values of the corresponding coefficients. Penalties indexed by smaller values of β shrink less. Thus, when non convex penalties ($\beta < 1$) perform better it is due solely to the effect of this decreased shrinkage. Therefore in these situations the performance of convex methods can be improved by simply modifying the shrinkage prescription at each of their path points.

When the predictor variables are correlated on the training sample different generalized elastic net penalties do not necessarily enter variables in the same order. That is, path points of the same sparsity need not involve the same non zero coefficients. However, if the correlations are not large one might expect different penalties not to differ too much in this regard, especially for the most important variables affecting the predictions.

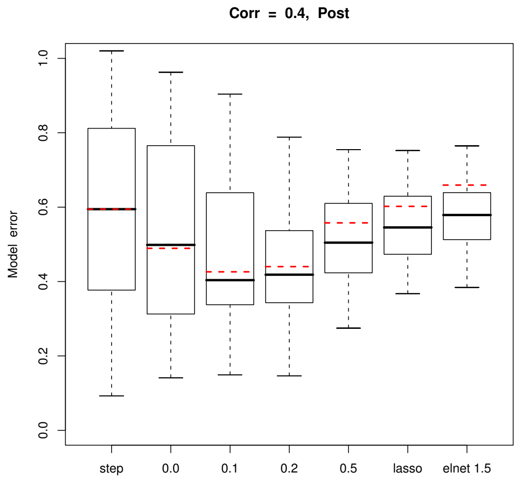

Figure 22: Results for Fig. 12 data when methods are used as selectors. Red lines are medians from Fig. 12 distributions.

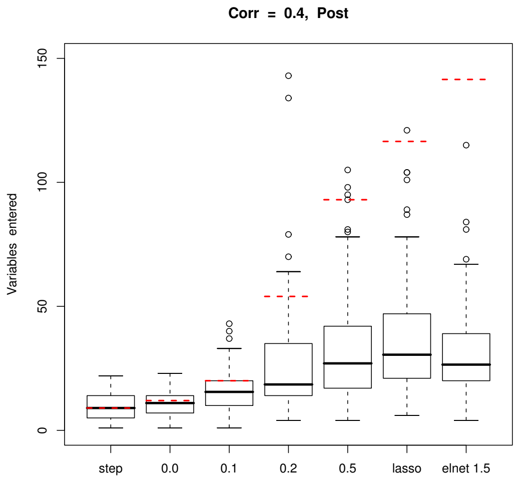

Figure 23: Results for Fig. 13 data when methods are used as selectors. Red lines are medians from Fig. 13 distributions.

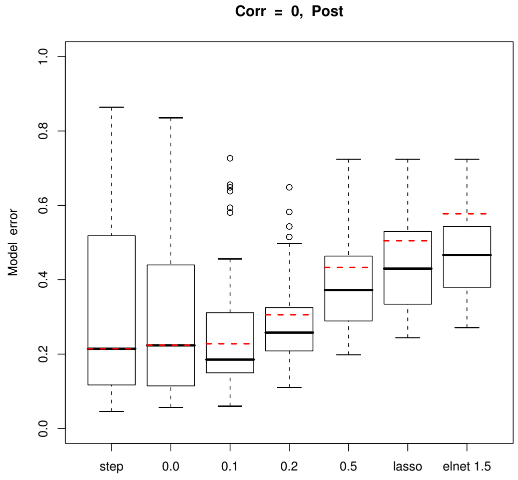

Figure 24: Results for Fig. 14 data when methods are used as selectors. Red lines are medians from Fig. 14 distributions.

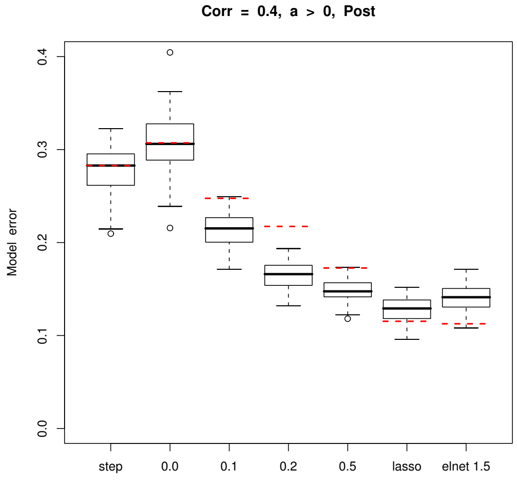

Figure 25: Results for Fig. 15 data when methods are used as selectors. Red lines are medians from Fig. 15 distributions.

These considerations suggest that the performance of a convex method such as the lasso might be improved by using it as a "selector". At each path point λ, the "active" variables corresponding to the non zero coefficients $A(\lambda) = \{j \,|\, \hat{a}_j(\lambda) \neq 0\}$ are identified. Then a different regression procedure, producing less shrinkage, is used to estimate the coefficient values $\{\tilde{a}_j(\lambda)\}_{j \in A(\lambda)}$ of these active variables. The overall solution at the path point λ is then taken to be $\{\hat{a}_j(\lambda) = \tilde{a}_j(\lambda)\}_{j \in A(\lambda)}$ and $\{\hat{a}_j(\lambda) = 0\}_{j \notin A(\lambda)}$. In this way the selector identifies the non zero coefficients and the post–regression procedure determines their values, at each path point. Often an unregularized regression (7) (8) is used for this the second "post–processing" step.

Generally, convex selectors are employed due to the unattractive computational aspects associated with non convex optimization. With GPS, paths corresponding to non convex penalties ($\beta < 1$) can be obtained with computation similar to that of convex ones ($\beta \geq 1$) (Table 1), thereby expanding the pool of eligible selectors. In this section the utility of selectors based on non convex generalized elastic net penalties ($\beta < 1$) is examined, and compared to that of convex selectors ($\beta \geq 1$), using unregularized regression (7) (8) as the post–processor.

7.1 Least-squares regression

Figures 22–25 show the corresponding results of using the various GPS and exact convex procedures as selectors in the same set of situations represented in Figs. 12–15, respectively. The dashed (red) lines on each boxplot represents the medians of the corresponding distributions in Figs. 12–15. The results for forward stepwise regression are (by construction) identical and repeated for comparison. The results for $\beta = 0$ GPS are very similar since this GPS path interpolates the unregularized *state*wise regression solutions at path points where each successive variable enters (Section 4.1).

From Fig. 23 one sees that the selector based optimal solutions (40) are sparser than those for the corresponding direct solutions (Fig. 13) for all $\beta > 0$, with this effect being more pronounced as β increases. For the convex procedures ($\beta \geq 1$) the optimal selector solutions typically involve 25 non zero coefficients rather than around 120 for their corresponding direct methods. This is seen to increase accuracy in those situations (Figs. 22, 24) where the sparser direct methods ($\beta < 1$) provide superior results. Again, this accuracy increase is more pronounced for larger β, improving the convex methods the most. However, using these convex methods as selectors does not result in enough improvement to be competitive with the best non convex methods, especially when the latter are themselves used as selectors.

For the situation in Fig. 15 where the direct convex methods were seen to provide the best performance, using them as selectors decreases their accuracy (Fig. 25). In this situation using the $\beta < 1$ GPS procedures as selectors improves their performance, but not enough to compete with the direct convex methods.

As with the direct methods, the success of the selector strategy in improving performance is seen to depend on more than just the sparsity of the optimal coefficients $\mathbf{a}^*$. Other factors including their signs and the correlational structure of the predictor variable distribution are seen to be relevant. The results shown in Figs. 22–25 indicate that the best direct methods give rise to the best selectors. Namely, when the sparser direct methods are superior it is because they are better selectors as well as better shrinkers. This also suggests that one can reduce computation by estimating the best direct method through bridge-regression (16) (17) and then use its path as the selector to ascertain whether or not accuracy is improved.

7.2 Logistic regression

Figures 26–31 show the results of using GPS and exact convex paths, based on the logistic regression loss (5), as selectors for an unregularized logistic regression post–processor. Each successive figure represents the same situation as depicted in the corresponding successive Figs. 16–21 respectively. The dashed (red) lines are the medians of the corresponding distributions in Figs. 16–21.

For the situation depicted in Figs. 16 and 17 one sees in Fig. 26 that using the non convex ($\beta < 1$) GPS methods as selectors results in no improvement in AUC (41) over their corresponding direct methods. There is substantial improvement in AUC with the convex methods however. From Fig. 27 one sees that the probability estimates (43) are improved for all methods when used as selectors, being especially dramatic for the convex ones. These effects are the same for the uncorrelated case, $\mathbf{x}_i \sim N(0, \mathbf{I})$, shown in Figs. 28 and 29. In the case of correlated predictor variables and all optimal coefficients $\{a_j^*\}_1^n$ having the same sign (Figs. 30 and 31) one again sees a complete reversal. For logistic regression, using each method as a selector *decreases* its accuracy.

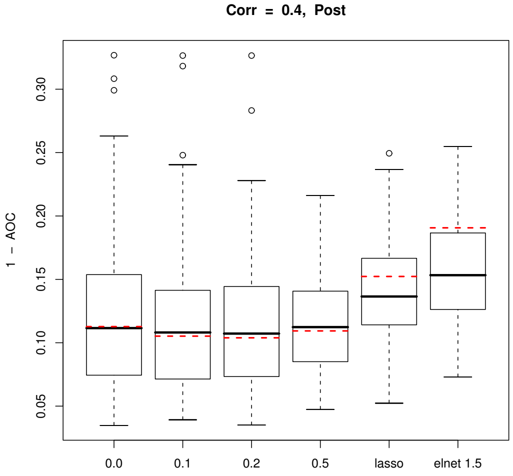

Figure 26: Results for Fig. 16 data when methods are used as selectors. Red lines are medians from Fig. 16 distributions.

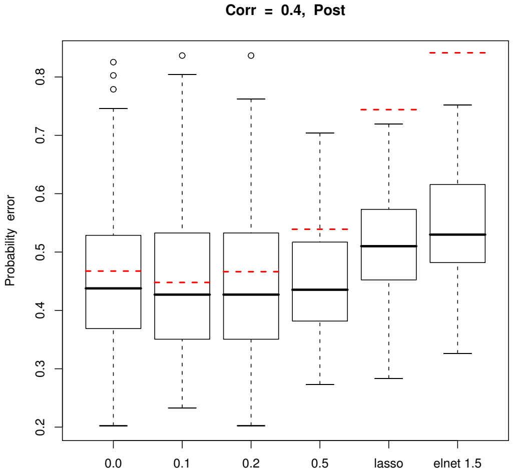

Figure 27: Results for Fig. 17 data when methods are used as selectors. Red lines are medians from Fig. 17 distributions.

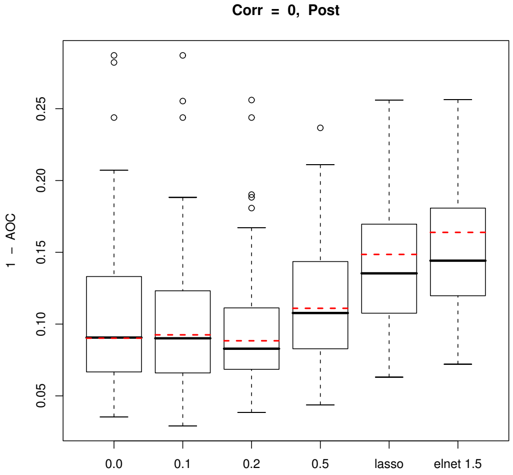

Figure 28: Results for Fig. 18 data when methods are used as selectors. Red lines are medians from Fig. 18 distributions.

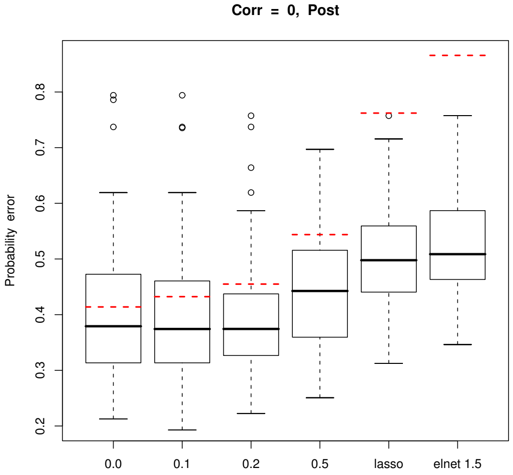

Figure 29: Results for Fig. 19 data when methods are used as selectors. Red lines are medians from Fig. 19 distributions.

both in terms of AUC (41) and probability estimation error (43). The convex methods, which are best in this situation, are damaged the most.

Results for using GPS non convex and exact convex logistic regression as selectors are somewhat similar to those for least-squares regression (Section 7.1). The convex methods are substantially improved in those situations friendly to the sparser non convex methods. In situations where the direct convex methods are best (Figs. 30 and 31) selectors based on all methods perform substantially worse that their direct counterparts. For logistic regression, as with least-squares, the best direct methods serve as the best selectors.

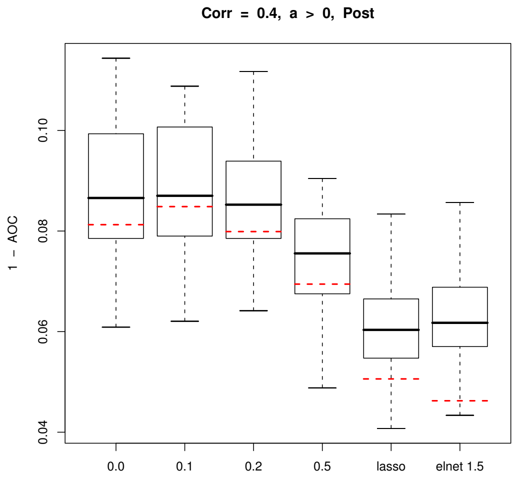

Figure 30: Results for Fig. 20 data when methods are used as selectors. Red lines are medians from Fig. 20 distributions.

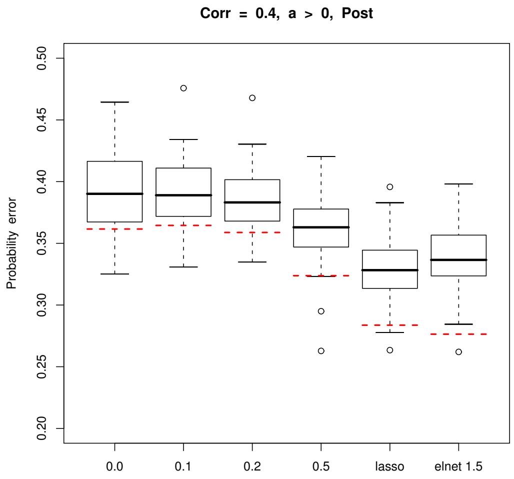

Figure 31: Results for Fig. 21 data when methods are used as selectors. Red lines are medians from Fig. 21 distributions.

8 Empirical bridge regression

The accuracy results presented in Sections 6 and 7 are based on perfect model selection using the known optimal coefficients $\mathbf{a}^*$ (6) for each problem. This is appropriate for assessing the quality of paths produced by different penalties without reference to specific model selection procedures. However in practice $\mathbf{a}^*$ is unknown and one must use an empirical model selection technique (Section 2.1) to simultaneously estimate the best penalty and corresponding path point (17). Since empirical model selection is an imperfect process results obtained in actual applications may be somewhat less optimistic. Here this effect is investigated using 20–fold cross–validation for model selection.

The upper left panel of Fig. 32 shows results for least–squares regression corresponding to those in Figs. 14 and 24 (population uncorrelated predictor variables). The first box plot (las-dir) shows the distribution over the 50 data sets used in Fig. 14 of

$$[R(\hat{\mathbf{a}}(\hat{\rho})) - R(\mathbf{a}^*)]/R(\mathbf{a}^*)$$

for the lasso, where $\hat{\rho}$ is the 20–fold cross–validation estimate of the optimal path point. The second box plot (las-sel) shows the corresponding distribution when the lasso is used as a selector for unregularized least-squares regression. The third box plot (brg-dir) shows the distribution over the same 50 data sets of

$$[R(\hat{\mathbf{a}}_{\hat{\beta}}(\hat{\rho})) - R(\mathbf{a}^*)]/R(\mathbf{a}^*)$$

where the penalty indexed by $\hat{\beta}$ and the corresponding path point $\hat{\rho}$ are jointly estimated by 20–fold cross–validation. The fourth box plot (brg-sel) shows the corresponding distribution when the methods are used as selectors. The upper right panel of Fig. 30 shows the frequency distribution of the penalties $\hat{\beta}$ selected by direct bridge regression over the 50 data sets.

The lower left panel of Fig. 32 shows corresponding results for logistic regression, on the 50 data sets used in Figs. 18 and 28, for the same set of procedures. Here (lack of) quality is assessed by $1 - AUC$, where AUC is the area under the ROC curve. The lower right panel shows the frequency distribution of penalties $\hat{\beta}$ selected by direct logistic bridge regression.

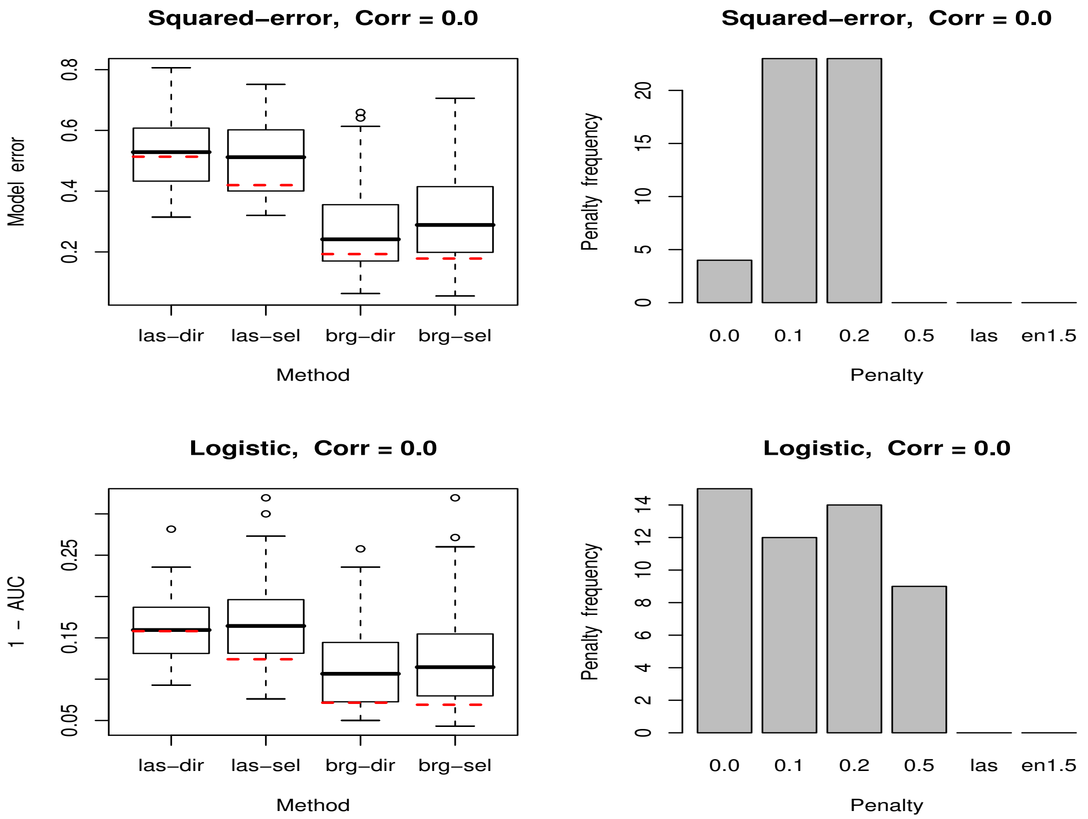

Figure 32: Results for the lasso, lasso selector, bridge regression, and bridge selector using empirical model selection based on 20–fold cross-validation. The upper panels are for the 50 regression data sets used in Fig. 14, and the lower are for the 50 logistic regression data sets used in Fig. 18. The left panels show the distributions of lack of accuracy (risk, $1 - AUC$). The right panels show the frequency distribution of selected penalties by direct bridge regression over the respective 50 data sets. The red dashed lines are the medians of the corresponding perfect model selection distributions.

The red dashed lines in Fig. 32 represent the medians of the corresponding distributions using perfect model selection.

From the upper left panel of Fig. 32 one sees that for the direct lasso, empirical model selection based on cross-validation increases the prediction risk by only 3% over using perfect model selection. For direct bridge regression the corresponding increase is around 25%. With the selector based methods there is a much larger increase in prediction risk caused by empirical model selection: 22% for the lasso selector and 62% for the bridge selector. With perfect model selection the selector strategy substantially increased accuracy for the lasso and slightly for bridge regression in this example, whereas with cross-validation this is not the case. This effect may be caused by the inherent discontinuities in the selector induced paths. There are finite jumps in all coefficient paths at those points where each successive variable enters the model. As discussed in Section 3.2.1 this causes increased instability in the presence of the intrinsic randomness of empirical model selection. The direct lasso and GPS paths are strictly continuous over their entire extent, thereby inducing less instability as a result of this randomness.

Qualitative results for the logistic regression example (lower left panel of Fig. 32) are similar to those for squared–error loss (upper left). One sees a much larger increase in $1 - AUC$ due to empirical model selection for the selector based methods than for their direct shrinking counterparts. In the presence of perfect model selection both the logistic lasso and logistic bridge regression are improved when used as selectors, whereas with 20–fold cross-validation that trend is reversed.

The two right panels of Fig. 32 illustrate that model selection based on 20–fold cross-validation is behaving reasonably in these examples. Comparison with Figs. 14 and 18 shows that the best penalty or one close to it in performance is chosen for all of the 50 data sets.

The effects of empirical model selection for the other examples with population correlated predictor variables are qualitatively similar to those shown in Fig. 32 for population uncorrelated predictors. Using 20–fold cross-validation rather than perfect model selection causes considerably more damage to the selector based procedures. As a result they are not competitive with the direct shrinking methods either for the lasso or bridge regression in those examples. As is the case here, direct bridge regression substantially out-performed the lasso in the population correlated examples as well, but by a somewhat smaller amount that reflected in Fig. 32.

9 Miscellany

In this section several additional aspects of the GPS method are discussed.

9.1 Unpenalized parameters

In some applications one may not wish to apply a penalty to all of the parameters. For example, the value of the intercept a_0 (2) is seldom penalized. Let $\{a_l\}_{l\in L}$ be a specified set of unpenalized parameters, where $L \subset \{1,2,\cdots,n\}$. Then $\{p_l(\nu)=0\}_{l\in L}$ (25) at all path points ν. This violates (23), which can be remedied by setting $\{p_l(\nu)=\varepsilon\}_{l\in L}$ so that the corresponding lamdas (26) become $\{\lambda_l(\nu)=g_l(\nu)/\varepsilon\}_{l\in L}$, with ε being a very small positive quantity. Each such $\lambda_l(\nu)$will have a very large absolute value unless its corresponding $|\,g_l(\nu))\,| \lesssim \varepsilon$. When one or more $|\,g_l(\nu))\,|$, $l \in L$, becomes larger than $\varepsilon \cdot \max_{j\notin L} |\,\lambda_j(\nu)\,|$ the coefficient corresponding to the largest among them $\hat{a}_{j^*}(\nu)$ is chosen for modification by the GPS algorithm (line 5 or line 6). This causes $|\,g_{j^*}(\nu+\Delta\nu)\,| < |\,g_{j^*}(\nu)\,|$ since to first order

$$\Delta|\,g_{j^*}(\nu)\,| = -h_{j^*}(\nu)\cdot\Delta\nu.$$

Here $h_{j^*}(\nu)$ is the j^*th diagonal element of the Hessian of the risk $\hat{R}(\mathbf{a})$ (8) evaluated at $\hat{\mathbf{a}}(\nu)$, which is positive for a strictly convex risk. Thus repeated steps of the algorithm maintain $\{|\,g_l(\nu))\,| \simeq \varepsilon\}_{l\in L}$. This in turn maintains

$$\{\hat{a}_l(\nu)\}_{l\in L} \simeq \arg\min_{\{a_l\}_{l\in L}} \left(\hat{R}(\mathbf{a})\,|\,\{\hat{a}_j(\nu)\}_{j\notin L}\right).$$

In the case of squared–error loss (4), centering the predictor variables to all have zero means causes the derivative $g_0(\nu)$ (24) corresponding to the intercept a_0 to be zero at all ν. Thus, $\lambda_0(\nu)=0$ and the intercept is never updated. For other losses this need not be the case. The GPS algorithm will update $\hat{a}_0(\nu)$ whenever $|\,g_0(\nu))\,| \gtrsim \varepsilon\cdot\max_{j\neq 0}|\,\lambda_j(\nu)\,|$.

For many losses it is possible to rapidly solve

$$\hat{a}_0(\nu) = \arg\min_{a_0}\left(\hat{R}(\mathbf{a})\,|\,\{\hat{a}_j(\nu)\}_1^n\right). \tag{44}$$

When this is the case (for example with the logistic loss (5)) applying (44) at every iteration will increase the speed of the GPS algorithm by reducing the number of steps.

9.2 Weighted parameters

More generally, one may wish to penalize some parameters less heavily than others. With additive penalties

$$P(\mathbf{a}) = \sum_{j=1}^{n} P_j(a_j) \tag{45}$$

this can be accomplished by applying a weight $1/v_j > 0$ to each corresponding term $P_j(a_j)$ in the penalty. Parameters a_j corresponding to larger values of v_j will be less heavily penalized than those with smaller values. Thus, values of their corresponding $\hat{a}_j(\nu)$ will be less constrained by the penalty and influenced more by the data distribution. In this sense one can consider v_j as a weight for variable x_j characterizing its presumed a priori importance.

For some penalties a similar effect can be obtained by scaling each (standardized) variable x_j by a factor $s_j > 0$. This is equivalent to dividing its corresponding coefficient in the penalty by s_j $(a_j \rightarrow a_j/s_j)$. For the power family (18) this is the same as applying the weight $v_j = s_j^{\gamma}$. For mixture penalties such as the generalized elastic net (19) (20) applying differential scaling to the predictor variables produces a more complex effect. Variables with larger scales are penalized less. However, the form of the penalty becomes different for each scaled variable. For $\beta \notin \{0, 1, 2\}$, variables with larger scales receive penalties more closely related to the lasso $(\beta = 1)$ than those with smaller scales. If this is not desired one can use standardized variables and directly apply the variable weight $(1/v_j)$ to the respective terms in the penalty to reflect presumed a priori importance.

9.3 Penalty carpentry

For additive penalties (45) each $P_j(a)$ us usually taken to be the same function $P_j(a_j) = P(a_j)$, symmetric about $a_j = 0$. This is not a requirement. Each $P_j(a_j)$ in (45) can be a different function so long as (23) is met. For example, some of the coefficients can be subjected to a ridge penalty, others to a lasso or subset selection penalty. Also, (23) does not require that each $P_j(a_j)$ be a symmetric function. For example, one could use a convex penalty for positive coefficient values and a concave one for negative values. Also, the relative strength of the positive and negative parts of the penalty can be different so as to preferentially discourage large coefficients of a particular sign.

For example, suppose that for a given loss $L(y, F)$ and additive penalty (45) one wishes to construct the path under the constraint that one or more selected coefficient values be non negative at all points ν. That is

$$\{\hat{a}_j(\nu) \geq 0\}_{j \in J} \tag{46}$$

where J is the set of indices corresponding to the selected coefficients. Employing the GPS algorithm, the modified penalty

$$\tilde{P}(\mathbf{a}) = \sum_{j \notin J} P_j(a_j) + \sum_{j \in J} [P_j(a_j)\, I(a_j \geq 0) + \infty \cdot |\, a_j \,|\, I(a_j < 0)] \tag{47}$$

will produce a path corresponding to (45) under the constraints (46). In over-determined problems this path converges to the solution to (7) (8) under (46) at its end point where a valid descent step reducing the risk (8) no longer exists.

9.4 Step size

For a given loss $L(y, F)$ (3) and penalty $P(\mathbf{a})$ (10) the only parameter associated with the GPS algorithm is $\Delta\nu$ (line 7). Its value regulates the size of the steps $\nu \leftarrow \nu + \Delta\nu$ as the iterations proceed. This in turn regulates the number of iterations required to reach the end of the path. Larger values of $\Delta\nu$ require fewer iterations to traverse the entire path. As $\Delta\nu \rightarrow 0$, the steps produced by the GPS algorithm approach a smooth continuous path in parameter space. Finite values of $\Delta\nu$ produce a less smooth sequence of points that lie close to this path where the degree of closeness (smoothness) depends on the value of $\Delta\nu$ and the the number of non zero coefficients along the path. More non zero coefficients generally require smaller values of $\Delta\nu$ for the same smoothness since the increments (line 7) are shared among more coefficients so that each one is updated less frequently.

The strategy used in the current implementation is to choose the size of the step $\Delta\nu$ at each path point ν so as to reduce the empirical risk (8) by a fixed fraction

$$[\hat{R}(\hat{\mathbf{a}}(\nu)) - \hat{R}(\hat{\mathbf{a}}(\nu + \Delta\nu))]/\hat{R}(\hat{\mathbf{a}}(\nu)) = \varepsilon$$

at that point. The default value is $\varepsilon = 0.01$, but other values may be appropriate in different applications.

10 Related work

There is a large literature pertaining to regularized regression and classification. Most work involves the use of convex loss functions with convex penalties so that the overall criterion (10) is convex. Standard algorithms for convex optimization problems (Boyd and Vandenberghe 2004) can then be employed to repeatedly solve (10) for a sequence of λ–values. For squared–error loss and the lasso penalty, special one-at-a-time coordinate descent algorithms have been developed (Daubechines, DeFrise and De Mol 2004, Wu and Lang 2008) that are much faster than general convex optimizers for this special case. The method was extended to the full convex elastic net family of penalties by Var der Kooij 2007 and Friedman *et al* 2007. The one-at-a-time coordinate descent strategy was applied to regularized logistic and multinomial regression by Balaji *et al* 2005 and Genkin, Lewis, and Madigan 2007, and was further generalized to the convex elastic net family by Friedman, Hastie, and Tibshirani 2008. These one-at-a-time coordinate descent algorithms are currently the fastest methods for these particular convex problems and, as seen in Table 1, their speed can rival that of GPS for those special loss–penalty combinations.

In order to obtain sparser solutions than the lasso with squared–error loss Fan and Li 2001 proposed the non convex piecewise quadratic SCAD penalty. They use an iterative approximate Newton–Raphson method for solving (10) at each λ–value. Lin and Wu 2007 proposed a family of non convex penalties bridging subset selection and the lasso consisting of an adjustable mixture of those two penalties. They use a mixed integer programming technique to solve (10) for square–error loss at each path point. Neither of these methods are speed competitive with GPS. However, the GPS algorithm can be used to approximate paths based on these penalties, for any convex loss. Zhang 2007 proposed the MC+ family of non convex piecewise quadratic penalties and provides a fast algorithm for generating paths for squared–error loss. Again, the GPS algorithm can be used to generalize this method to any convex loss, for example logistic regression (5).

Direct path seeking algorithms for producing paths sparser than the lasso have been proposed by Buhlmann and Yu 2006 based on a modification of squared–error loss boosting (Friedman 2001). This procedure has connections to the sparsity inducing non negative garrote (Breiman 1995).

Again for squared–error loss, a variety of post–processing strategies using convex selectors have been proposed to create sparser paths. The relaxed lasso (Meinshausen 2007) and VISA (Radchenko and James 2008) use the lasso as the basic selector. The Dantzig selector (Candes and Tao 2007) uses a different convex constrained procedure for variable selection along its path with properties similar to the lasso. Although generally faster than exact methods based on non convex penalties, these selectors are still considerably slower than GPS based on those penalties. Also GPS is not limited to squared–error loss, and itself can be used to produce non convex selectors with improved selection performance, as illustrated in Section 7. However, as shown in Section 8, using these selectors for regression need not lead to improved *prediction* accuracy over their direct shrinking counterparts in the presence of empirical model selection.

Rosset 2003 proposed a direct path seeking algorithm for the convex members of the power family (18) ($\gamma \geq 1$) based on boosting, and illustrated its use in approximating ridge penalty ($\gamma = 2$) solutions. The GPS method is a generalization of Rosset's proposal that more closely approximates exact paths for convex penalties, and extends application to non convex penalties.

11 Discussion

The principal advantages of using GPS to generate paths based on chosen loss–penalty combinations are simplicity, generality, and speed. The same basic algorithm can be used with a wide variety of penalty and loss criteria without the need to develop specialized search strategies for each such combination. One can concentrate on the statistical merits of the resulting regularized procedure with less concern for computational complexity. The speed of GPS extends the application of regularized regression to very large problems using any convex loss with any penalty satisfying (23).

As seen in Section 6, the best penalty for any given application can strongly depend on various different aspects of the particular problem. These include the actual sparsity of the optimal coefficients $\mathbf{a}^*$ (6), their relative signs, and the correlational structure of the predictor variable distribution. Since some or all of these properties are usually unknown, bridge-regression (16) (17) can aid in penalty choice. Again the speed of GPS makes this possible for large problems.

12 Acknowledgements

Helpful discussions with Trevor Hastie, Rob Tibshirani and Saharon Rosset are gratefully acknowledged. This work was partially supported by the National Science Foundation under grant DMS-97-64431.

References

[1] Balaji, K., Carlin, L., Figueiredo, M. A. T., and Hartemink. A. J. (2005). Sparse multinomial logistic regression: fast algorithms and general bounds. *IEEE Trans. Pattern Analysis and Machine Intelligence* **27**, 957.

[2] Boyd. S. and Vandenberghe, L. (2004). *Convex Optimization.* Cambridge University Press.

[3] Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics* **37**, 373–384.

[4] Buhlmann, P. and Yu, B. (2006). Sparse Boosting. *Journal of Machine Learning Research* **7**, 1001-1024).

[5] Candes, E. and Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n (with discussion). *Annals of Statistics* **35**, 2313–2351.

[6] Daubechines, I., Defrise, M. and De Mol, C. (2004). An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics* **57**, 1413–1457.

[7] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression (with discussion). *Annals of Statistics* **32**, 407–499.

[8] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* **96**, 1348–1360.

[9] Frank, I. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools (with discussion). *Technometrics* **35**, 109–148.

[10] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics* **29**, 1189–1232.

[11] Friedman, J. H., Hastie, T., Höfling, H. and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics* **1**, 302-332.

[12] Friedman, J. H., Hastie, T. and Tibshirani, R. (2008). Regularized paths for generalized linear models via coordinate descent. Stanford University, Dept. of Statistics technical report.

[13] Genkin, A., Lewis, D., Madigan, D. (2007). Large–scale Bayesian logistic regression for text categorization. *Technometrics* **49**, 291–304.

[14] Hastie, T., Tibshirani, R., and Friedman, J. H. (2001), *Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer-Verlag, New York.

[15] Hastie, T., Taylor, J., Tibshirani, R. and Walther, G. (2007). Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics* **1**, 1–29.

[16] Horel, A. E. and Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* **12**, 55–67.

[17] Huang, J., Ma, S., Xie, H., and Zhang, C-H. (2007). A group bridge approach for variable selection. The University of Iowa, Dept. of Statistics technical report No. 376.

[18] Lin, Y. and Wu, Y. (2007). Variable selection via a combination of the L_0 and L_1 penalties. *J. Computational and Graphical Statistics*, **16**, 782–798.

[19] Meinshausen, N. (2007). Relaxed lasso. *Computational statistics and Data Analysis* **52**, 374–393.

[20] Radchenko, P. and James, G. (2008). Variable Inclusion and Shrinkage Algorithms. *J. Amer. Statist. Assoc.* (to appear).

[21] Rosset, S. (2003). Topics in regularization and boosting. Ph. D. Thesis, Dept. of Statistics, Stanford University.

[22] Tibshirani, R. (1996). Regularization shrinkage and selection via the lasso. *J. Roy. Statist. Soc.* B **58**, 267–288.

[23] Van der Kooij, A. (2007). Prediction accuracy and stability of regression with optimal scaling transformations. Ph. D Thesis, Dept. of Data Theory, Leiden University.

[24] Wold, S., Ruhe, A., Wold, H. and W. J. Dunn III (1984). The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM Journal of Scientific and Statistical Computing* **5**, 735–742.

[25] Wu, T. and Lange, K. (2008) Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics* 2, 224-244.

[26] Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. Roy. Statist. Soc.* B **68**, 49-67.

[27] Zhang, C-H. (2007). Penalized linear unbiased selection. Rutgers University, Dept. of Statistics technical report No. 2007-003.

[28] Zhao, P., Rocha, G., Yu, B. (2006). Grouped and hierarchical model selection through composite absolute penalties. University of California, Berkeley, Dept. of Statistics technical report No. 703.

[29] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. Roy. Statist. Soc.* B **67**, 301-320.