International Journal of Forecasting 28 (2012) 722–738

Contents lists available at SciVerse ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Fast sparse regression and classification

Jerome H. Friedman

Department of Statistics, Stanford University, Stanford, CA 94305, United States

ARTICLE INFO

Keywords:
Regression
Classification
Regularization
Sparsity
Variable selection
Bridge-regression
Lasso
Elastic net
l_p-norm penalization

ABSTRACT

Many present day applications of statistical learning involve large numbers of predictor variables. Often, that number is much larger than the number of cases or observations available for training the learning algorithm. In such situations, traditional methods fail. Recently, new techniques have been developed, based on regularization, which can often produce accurate models in these settings. This paper describes the basic principles underlying the method of regularization, then focuses on those methods which exploit the sparsity of the predicting model. The potential merits of these methods are then explored by example.

1. Introduction

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

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

which is considered to be a random sample from some joint (population) distribution with probability density $p(\boldsymbol{x}, y)$. The random variable y is the "outcome" or "response" and $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ 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 $\boldsymbol{a} = \{a_0, a_1, \ldots, a_n\}$ of the linear model

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

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

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

over future predictions $\boldsymbol{x}, y \backsim p(\boldsymbol{x}, y)$. 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 the squared error

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

and the Bernoulli negative log-likelihood

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

associated with logistic regression. The negative log-likelihood representing any probability model can be characterized by a corresponding loss criterion.

For a specified loss criterion, the optimal parameter values are from Eq. (3)

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

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

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

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

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

E-mail address: jhf@stanford.edu.

0169-2070/$ – see front matter
doi:10.1016/j.ijforecast.2012.05.001

2. Regularization

It is well known that $\hat{\boldsymbol{a}}$ in Eqs. (7) and (8) often provides a poor estimate of $\boldsymbol{a}^*$; that is, $R(\hat{\boldsymbol{a}}) \gg R(\boldsymbol{a}^*)$ (Eq. (3)). 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 in Eq. (7) when Eq. (8) is evaluated on different random samples drawn from the population distribution. A common remedy is to modify Eq. (7) in order to stabilize the estimates by placing a restriction on the joint solution values. That is,

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

Here, $P(\boldsymbol{a})$ is a non-negative function of the parameters specifying the form of the constraint, and $t \geq 0$ regulates its strength. For a given data set (Eq. (1)), the loss criterion $L(y, F)$ in Eqs. (3) and (8), and the constraint function $P(\boldsymbol{a})$, the solution to Eq. (9) depends only on the value chosen for t. Varying its value induces a family of solutions, with each member being indexed by a particular value of $t \in [0, P(\hat{\boldsymbol{a}})]$ (Eq. (7)). This same family of solutions can be obtained through the equivalent (penalized) formulation of Eq. (9):

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

where $P(\boldsymbol{a})$ is the constraining function in Eq. (9), here called a penalty, and $\lambda > 0$ regulates its strength. Setting $\lambda = \infty$ produces the totally constrained solution ($t = 0$), whereas $\lambda = 0$ yields the unrestricted solution ($t \geq P(\hat{\boldsymbol{a}})$). Each value of $0 \leq \lambda \leq \infty$ in Eq. (10) produces one of the solutions $0 \leq t \leq P(\hat{\boldsymbol{a}})$ in Eq. (9), with smaller values of λ corresponding to larger values of t. Thus, Eq. (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 $\boldsymbol{a}^*$ (Eq. (6)) also represent a point in the parameter space. For a given penalty, the goal is to find a point λ^* on its path such that the corresponding solution $\hat{\boldsymbol{a}}(\lambda^*)$ is closest to $\boldsymbol{a}^*$, where the distance is characterized by the prediction risk in Eq. (3)

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

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

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

through

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

where $\tilde{R}(\boldsymbol{a})$ is a surrogate model selection criterion computed from the training data in Eq. (1), whose minimum is intended to approximate that of the actual risk (Eq. (3)).

There are a wide variety of model selection criteria available, each developed for a particular combination of loss (Eq. (3)) and penalty $P(\boldsymbol{a})$. Among the most general, being applicable to any loss-penalty combination, is cross-validation. The data are partitioned randomly 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 $p(\boldsymbol{x}, y)$ to compute the corresponding (estimated) risk in Eq. (3). These estimates are then used in Eq. (13) to obtain the estimate $\hat{\lambda}$. Sometimes the risk used in Eq. (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 $\hat{\boldsymbol{a}}(\lambda)$ in the parameter space such that some of the points on that path are close to the point $\boldsymbol{a}^*$ (Eq. (6)) representing the optimal solution. If no points on the path are close to $\boldsymbol{a}^*$, as measured by Eq. (11), then no model selection procedure can produce accurate estimates $\hat{\boldsymbol{a}}(\hat{\lambda})$. Since the path produced by Eq. (10) depends on the data, different randomly drawn data sets (Eq. (1)) will produce different paths for the same penalty. Thus, the paths are themselves random, and one seeks a penalty $P(\boldsymbol{a})$ that produces paths $\hat{\boldsymbol{a}}(\lambda)$ such that

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

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

2.3. Sparsity

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

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

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

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

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

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

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

2.3.1. Power family

One such family of penalties is the power family, defined as

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

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

Using squared-error loss (Eq. (4)), special cases of Eqs. (8), (10) and (18) include several popular regularized regression methods, namely $\gamma = 2$: ridge-regression; $\gamma = 1$: lasso; and $\gamma = 0$: all-subsets regression. Ridge-regression (Horel & Kennard, 1970) produces dense solutions, $S(\hat{\boldsymbol{a}}(\lambda)) \simeq 0$ (Eq. (15)), over its entire path $\infty \geq \lambda \geq 0$, while heavily shrinking the coefficient absolute values $|\hat{a}_j(\lambda)| \ll |a_j^*|$ 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 which are 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 (Eq. (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 values of γ. Note that for $\gamma \geq 1$, all penalties in the power family are convex functions of their argument $\boldsymbol{a}$, so that with convex risk $\hat{R}(\boldsymbol{a})$ (Eq. (8)), the problems represented by Eq. (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 (Eq. (18)) is not the only possibility for bridging all-subsets and ridge regression, with the lasso in between them. For bridging the lasso and ridge-regression, Zou and Hastie (2005) proposed the elastic net family of penalties, which can be expressed as

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

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 between these two extremes.

An extension of this family to non-convex members which produces sparser solutions than the lasso is

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

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 Eq. (18)). Values of β between these two extremes bridge all-subsets and the lasso, so that the entire family (Eqs. (19) and (20)) bridges all-subsets and ridge-regression for $0 < \beta \leq 2$.

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

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

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

The power (Eq. (18)) and generalized elastic net (Eqs. (19) and (20)) 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 $[\partial P_\gamma(\boldsymbol{a})/\partial|a_j|]_{a_j=0} = 0$, and for $\gamma < 1$, $[\partial P_\gamma(\boldsymbol{a})/\partial|a_j|]_{a_j=0} = \infty$. 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 $\hat{\boldsymbol{a}}_\gamma(\lambda)$ to have discontinuities as a function of λ for all non-convex members. For the generalized elastic net, $[\partial P_\beta(\boldsymbol{a})/\partial|a_j|]_{a_j=0}$ is non-zero and finite for all $0 < \beta < 2$. This causes the coefficients to enter (initially become non-zero) sequentially, with decreasing values of 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 the variance) of the coefficient estimates (Fan & Li, 2001).

3. Direct path seeking

A principal limitation of the bridge-regression strategy (Eqs. (16) and (17)) is the computational burden of obtaining the solutions to Eq. (16) for an adequate number of different penalties, and corresponding path points at which to perform Eq. (17). One approach that mitigates this burden is direct path seeking. The goal is to construct a path sequentially, directly in the parameter space, that closely approximates that for a given loss (Eq. (8)) and penalty $P(\boldsymbol{a})$, without having to repeatedly solve numerical optimization problems (Eq. (10)).

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

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

Here, $\boldsymbol{d}(\nu)$ is a vector characterizing a direction in the parameter space, and $\Delta\nu > 0$ is a specified distance in that direction.

If $\boldsymbol{d}(\nu)$ and $\Delta\nu$ are chosen so that the empirical risk (Eq. (8)) is reduced at each step, $\hat{R}(\hat{\boldsymbol{a}}(\nu + \Delta\nu)) < \hat{R}(\hat{\boldsymbol{a}}(\nu))$, then continued iterations of Eq. (21) will eventually reach a point $\hat{\boldsymbol{a}}(\nu_{\max})$ that minimizes the empirical risk. This procedure (Eq. (21)) can thus be viewed as a numerical optimization method for minimizing the empirical risk (Eq. (8)). However, the focus here is on the *path* traversed by the procedure from its starting point $\hat{\boldsymbol{a}}(0)$ to the end point $\hat{\boldsymbol{a}}(\nu_{\max})$. Different path-seeking methods, each intended for a particular loss-penalty combination, specify different prescriptions for calculating $\boldsymbol{d}(\nu)$ and $\Delta\nu$ at each path point $\hat{\boldsymbol{a}}(\nu)$. The path traversed through parameter space by this *single* numerical optimization (Eq. (21)) is intended to approximate that produced by Eq. (10), $\infty \geq \lambda \geq 0$, for the corresponding loss-penalty combination. This avoids the many numerical optimizations required for solving Eq. (10) for a large number of λ values.

Popular path seekers based on the squared-error loss (Eq. (4)) include partial least squares regression (PLS, see Wold, Ruhe, Wold, & Dunn III, 1984), which approximates the ridge-regression path (Frank & Friedman, 1993); forward stepwise regression, which is intended to approximate the all-subsets path; and least angle regression (Efron, Hastie, Johnstone, & Tibshirani, 2004), which approximates the lasso path. Gradient boosting (Friedman, 2001; Hastie, Taylor, Tibshirani, & Walther, 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 (Eqs. (16) and (17)), fast methods are required for inducing (approximate) paths for a wide variety of penalties, such as all of those in the power (Eq. (18)) or generalized elastic net (Eqs. (19) and (20)) families. In addition, it would be desirable to be able to employ a variety of loss criteria inducing risk functions (Eq. (8)) corresponding to likelihoods for a variety of probability models.

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

$$\left\{\frac{\partial P(\boldsymbol{a})}{\partial |a_j|} > 0\right\}_1^n, \tag{22}$$

for all values of $\boldsymbol{a}$. These conditions define a class of penalties where each member in the class is a monotonically increasing function of the absolute value of each of its arguments. All members of the power (Eq. (18)) and generalized elastic net (Eqs. (19) and (20)) families are included in this class. The SCAD penalty (Fan & Li, 2001), the MC+ family (Zhang, 2007), the group lasso (Yuan & Lin, 2006), the CAP family (Zhao, Rocha, & Yu, 2006), the grouped bridge family (Huang, Ma, Xie, & Zhang, 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

$$g_j(\nu) = -\left[\frac{\partial \hat{R}(\boldsymbol{a})}{\partial a_j}\right]_{\boldsymbol{a}=\hat{\boldsymbol{a}}(\nu)}, \tag{23}$$

$$p_j(\nu) = \left[\frac{\partial P(\boldsymbol{a})}{\partial |a_j|}\right]_{\boldsymbol{a}=\hat{\boldsymbol{a}}(\nu)}, \tag{24}$$

and

$$\lambda_j(\nu) = g_j(\nu)/p_j(\nu). \tag{25}$$

Here, $g_j(\nu)$ is the jth component of the negative gradient of the empirical risk (Eq. (8)) evaluated at the path point $\hat{\boldsymbol{a}}(\nu)$, and $p_j(\nu)$ is the corresponding component of the gradient of $P(\boldsymbol{a})$ with respect to $|a_j|$. Note that by the assumption in Eq. (22), all $\{p_j(\nu) > 0\}_1^n$. The components of the vector $\boldsymbol{\lambda}(\nu)$ are the component-wise ratios of these two gradients at $\hat{\boldsymbol{a}}(\nu)$. These lambdas (Eq. (25)) 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 $S = \{j | \lambda_j(\nu) \cdot \hat{a}_j(\nu) < 0\}$
5 if ($S =$ empty), $j^* = \arg\max_j |\lambda_j(\nu)|$
6 else $j^* = \arg\max_{j \in S} |\lambda_j(\nu)|$
7 $\hat{a}_{j^*}(\nu + \Delta\nu) = \hat{a}_{j^*}(\nu) + \Delta\nu \cdot \text{sign}(\lambda_{j^*}(\nu))$
8 $\{\hat{a}_j(\nu + \Delta\nu) = \hat{a}_j(\nu)\}_{j \neq j^*}$
9 $\nu \leftarrow \nu + \Delta\nu$
10 } Until $\boldsymbol{\lambda}(\nu) = 0$

Line 1 initializes the path. At each step, the vector $\boldsymbol{\lambda}(\nu)$ is computed via Eqs. (23)–(25) (line 3). At line 4, those nonzero coefficients $\hat{a}_j(\nu) \neq 0$ which have a 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 $\boldsymbol{\lambda}(\nu)$ (in absolute value) is selected (line 5). If one or more $\lambda_j(\nu) \cdot \hat{a}_j(\nu) < 0$, then the coefficient with the largest corresponding $|\lambda_j(\nu)|$ within this subset is selected instead (line 6). The selected coefficient $\hat{a}_{j^*}(\nu)$ is then incremented 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), thus producing the solution for the next path point $\nu + \Delta\nu$ (line 9). Iterations continue until all components of $\boldsymbol{\lambda}(\nu)$ are zero (line 10). Since each step (lines 7–8) reduces the empirical risk (Eq. (8)), $\hat{R}(\hat{\boldsymbol{a}}(\nu + \Delta\nu)) < \hat{R}(\hat{\boldsymbol{a}}(\nu))$, the algorithm will reach an unregularized solution (Eq. (7)) where all $\{\lambda_j(\nu) = 0\}_1^n$ (Eqs. (22)–(25)).

3.2. Motivation

In this section, it is explained why one might expect the GPS algorithm to track the paths produced by Eqs. (9) and (10) closely for convex risk (Eq. (8)) and penalties satisfying Eq. (22). The actual comparisons are presented in Section 4.

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

Then $\Delta\hat{\boldsymbol{a}}(t) = \hat{\boldsymbol{a}}(t + \Delta t) - \hat{\boldsymbol{a}}(t)$ is the solution to

$$\Delta\hat{\boldsymbol{a}}(t) = \arg\min_{\Delta\boldsymbol{a}}[\hat{R}(\hat{\boldsymbol{a}}(t) + \Delta\boldsymbol{a}) - \hat{R}(\hat{\boldsymbol{a}}(t))] \quad \text{s.t. } P(\hat{\boldsymbol{a}}(t) + \Delta\boldsymbol{a}) - P(\hat{\boldsymbol{a}}(t)) \leq \Delta t. \tag{26}$$

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

$$\left\{\left|\frac{d\hat{a}_j(t)}{dt}\right| < \infty\right\}_1^n, \quad t > 0. \tag{27}$$

Then, as $\Delta t \to 0$, assuming Eq. (22), Eq. (26) can be expressed in first order

$$\Delta\hat{\boldsymbol{a}}(t) = \arg\max_{\{\Delta a_j\}_1^n} \sum_{j=1}^n g_j(t)\cdot\Delta a_j \quad \text{s.t.} \sum_{\hat{a}_j(t)=0} p_j(t)\cdot|\Delta a_j| + \sum_{\hat{a}_j(t)\neq 0} p_j(t)\cdot\text{sign}(\hat{a}_j(t))\cdot\Delta a_j \leq \Delta t, \tag{28}$$

where

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

and

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

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

$$\{|\hat{a}_j(t + \Delta t)| \geq |\hat{a}_j(t)|\}_1^n, \tag{29}$$

so that $\{\text{sign}(\hat{a}_j(t)) = \text{sign}(\Delta\hat{a}_j(t))\}_{\hat{a}_j(t)\neq 0}$. Under this (additional) constraint, Eq. (28) becomes

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

This is a linear programming problem with solution

$$j^*(t) = \arg\max_{1\leq j\leq n} |g_j(t)|/p_j(t) \quad \Delta\hat{a}_{j^*}(t) = [g_{j^*}(t)/p_{j^*}(t)]\cdot\Delta t \quad \{\Delta\hat{a}_j(t) = 0\}_{j\neq j^*}. \tag{30}$$

From Eqs. (23)–(25), one sees that the GPS algorithm (lines 5 and 7–8) follows the strategy implied by Eq. (30), provided that $\text{sign}(\lambda_j(\nu)) = \text{sign}(\hat{a}_j(t))$ for all $\hat{a}_j(t) \neq 0$. This will be the case at all points where the GPS and exact paths coincide, as a consequence of the Karush–Kuhn–Tucker (KKT) optimality conditions

$$\lambda_j(t) = \lambda(t)\cdot\text{sign}(\hat{a}_j(t)), \quad \hat{a}_j(t) \neq 0, \tag{31}$$

where $\lambda(t) > 0$ is the value of λ in Eq. (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 (Eq. (9)) remains continuous (Eq. (27)) and monotonic (Eq. (29)) for $t \leq t_0$, the GPS and exact paths will coincide for $t \leq t_0$ in the limit $\Delta\nu \to 0$ ($\Delta t \to 0$).

If the exact path (Eq. (10)) is continuous and monotonic over its entire extent ($\infty \geq \lambda \geq 0$), as is often the case, then the GPS algorithm produces the exact path ($0 \leq \nu \leq \nu_{\max}$) 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 (Eq. (1)):

$$\sum_{i=1}^N x_{ij}x_{ik} = 0, \quad j \neq k. \tag{32}$$

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

3.2.1. Discontinuity

With the generalized elastic net family (Eqs. (19) and (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 in Eq. (10)) where each successive variable enters (coefficient initially becomes non-zero). This is caused by the variables entering at those points with finite non-zero coefficient values. This is illustrated in Fig. 1 for $\beta \in \{0.5, 0.4, 0.25, 0.1\}$ in the case of orthogonal (standardized) predictor variables (Eq. (32)) and squared-error loss (Eq. (4)). Here, the exact path solutions for nine coefficients are shown as thick (red) lines, plotted in terms of the fraction of the risk explained (Eq. (8))

$$r(\lambda) = [\hat{R}(\hat{\boldsymbol{a}}(\infty)) - \hat{R}(\hat{\boldsymbol{a}}(\lambda))]/\hat{R}(\hat{\boldsymbol{a}}(\infty)), \tag{33}$$

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

As Fig. 1 shows, the coefficient paths for $\beta = 0.5$ (upper left panel) are continuous. For $\beta < 0.5$, discontinuities appear in the coefficient paths; there are values of $r(\lambda)$ at which no exact solution exists. As β becomes smaller, the discontinuities increase in both 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 $r(\lambda)$ (Eq. (33)).

The thin (black) curves in Fig. 1 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 as $\Delta\nu \to 0$. 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 track the exact paths fairly closely (but not exactly) 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

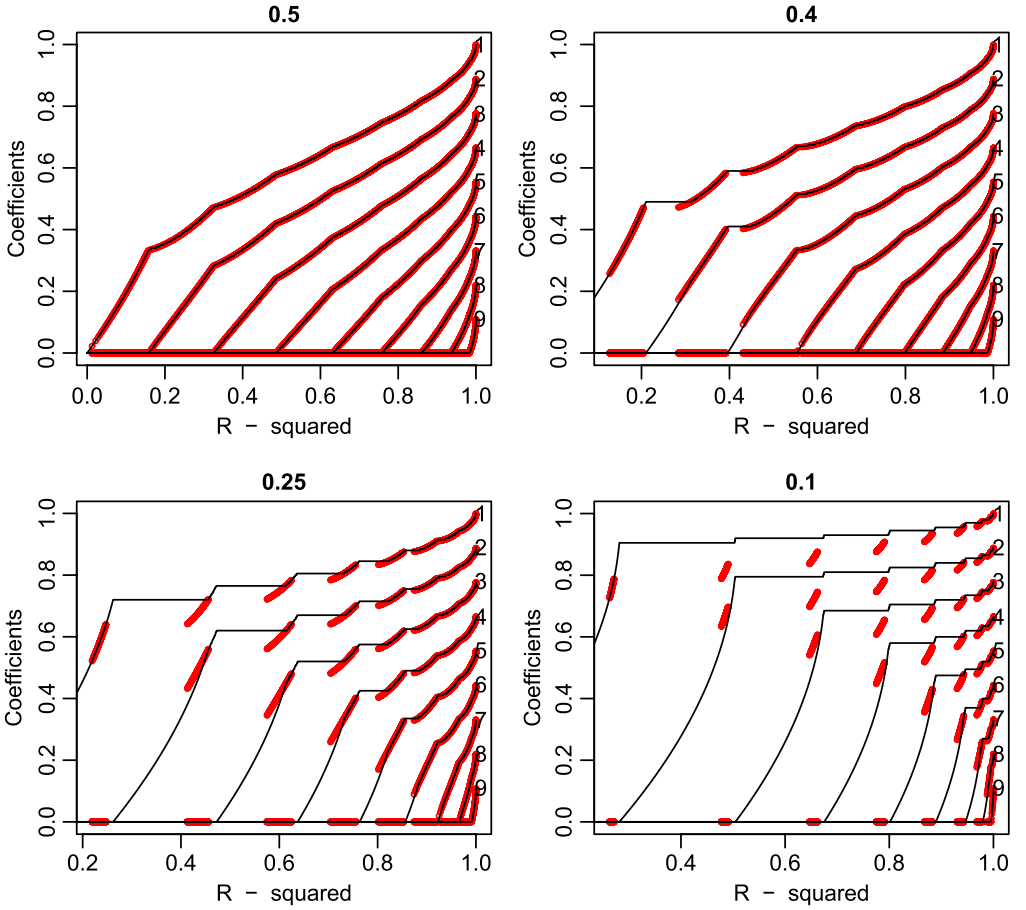

Fig. 1. Exact (thick red) and GPS (thin black) coefficient paths for elastic net non-convex penalties $\beta \in \{0.5, 0.4, 0.25, 0.1\}$, with orthogonal predictors. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

representing solutions for the latter. At other path points, GPS provides continuous paths that interpolate between the corresponding exact solution points.

As was 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 in the orthogonal case here (Eq. (32)).

3.2.2. Non-monotonicity

When all exact coefficient paths $\hat{a}_j(t)$ (Eq. (9)) are continuous, the GPS paths coincide with the exact ones as long as all $\hat{a}_j(t)$ remain monotonic (Eq. (29)). In this case, from the KKT conditions (Eq. (31)), one has

$$\lambda_j(\nu) \cdot \text{sign}(\hat{a}_j(\nu)) = \max_k |\lambda_k(\nu)|, \quad \hat{a}_j(\nu) \neq 0, \tag{34}$$

for all non-zero GPS coefficients. If the exact path for some coefficient $\hat{a}_j(t)$ becomes non-monotonic ($|\hat{a}_j(t + \Delta t)| < |\hat{a}_j(t)|$) at some point t_0 (ν_0), then Eq. (31) remains valid for $t > t_0$ for all coefficients on the exact path, whereas Eq. (34) need not hold for all coefficients on the GPS path. There may be no single variable increment (lines 7–8) that produces the exact solution for $\nu > \nu_0$. So long as $\text{sign}(\lambda_j(\nu)) = \text{sign}(\hat{a}_j(\nu))$, GPS will continue to monotonically update the other coefficients that satisfy Eq. (34), leaving $\hat{a}_j(\nu)$ constant as ν increases, since $|\lambda_j(\nu)| < \max_k |\lambda_k(\nu)|$. This continues until $\lambda_j(\nu)$ changes sign. At that point, $\lambda_j(\nu) \cdot \hat{a}_j(\nu) < 0$, and the GPS algorithm (line 6) chooses that coefficient $j^* = j$ for updating. This update (line 7) causes $|\hat{a}_j(\nu + \Delta\nu)| < |\hat{a}_j(\nu)|$, thereby (belatedly) introducing non-monotonicity into the GPS path for $\hat{a}_j(\nu)$. As long as $\text{sign}(\lambda_j(\nu)) \neq \text{sign}(\hat{a}_j(\nu))$, the coefficient $\hat{a}_j(\nu)$ will continue to decrease in absolute value for successive steps, while the other coefficients $\{\hat{a}_j(\nu)\}_{j \notin S}$ remain constant. This continues until either $\hat{a}_j(\nu)$ or $\lambda_j(\nu)$ changes sign, at which point $\text{sign}(\lambda_j(\nu)) = \text{sign}(\hat{a}_j(\nu))$, and updating proceeds in the usual manner (line 5).

4. Examples

In this section, applications of the GPS algorithm to data using various loss-penalty combinations are presented, and compared to the exact paths for the convex penalties.

4.1. Least-squares regression: diabetes data

This data set, used by Efron et al. (2004), consists of $n = 10$ predictor variables and $N = 442$ observations.

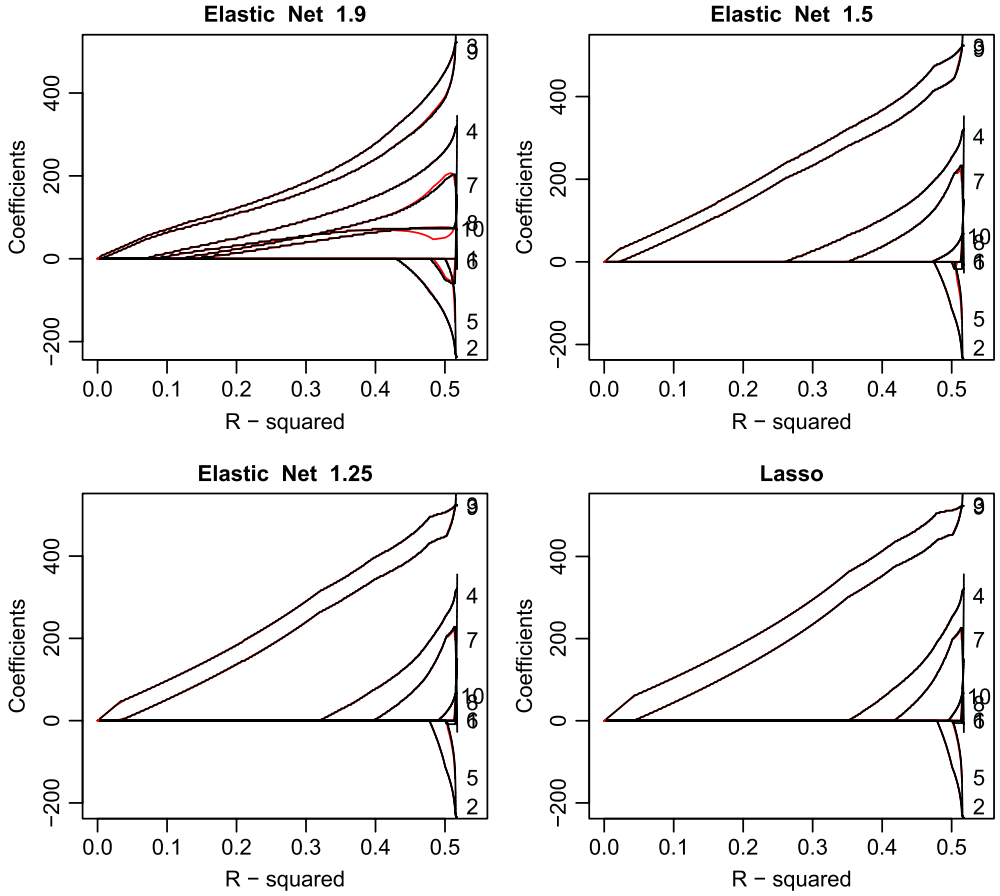

Fig. 2. Exact (red) and GPS (black) paths for the diabetes data, using convex elastic net penalties $\beta \in \{1.9, 1.5, 1.25, 1.0\}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The outcome variable is numeric, so the squared-error loss in Eq. (4) was employed.

Fig. 2 shows the ten coefficient paths as a function of R^2 (Eq. (33)) for elastic net penalties (Eq. (19)), with $\beta = 1.9$ (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, while the black ones are the corresponding GPS paths. For $\beta = 1.9$, slight differences are seen to occur near the end of the path, where several of the exact coefficient paths become non-monotonic and the GPS paths exhibit the behavior described in Section 3.2.2. For the other penalties, the differences are seen to be smaller. As β decreases, the solutions become sparser, in that there tend to be fewer non-zero coefficients for the same degree of data fit, as measured by R^2. That is,

$$S(\hat{\boldsymbol{a}}_\beta(R^2)) \geq S(\hat{\boldsymbol{a}}_{\beta'}(R^2)), \quad \beta < \beta', \tag{35}$$

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

Fig. 3 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, one sees a similar pattern of further increasing sparsity (Eq. (35)) as $\beta < 1$ decreases.

The red numbers in the lower right panel of Fig. 3 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 include in the model next. 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 based on which 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.

Fig. 4 shows the number of non-zero coefficients (logarithmic scale) as a function R^2 along the path for a larger set of generalized elastic net penalties. This number is inversely related to sparsity (Eq. (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 are identical to that of $\beta = 0$

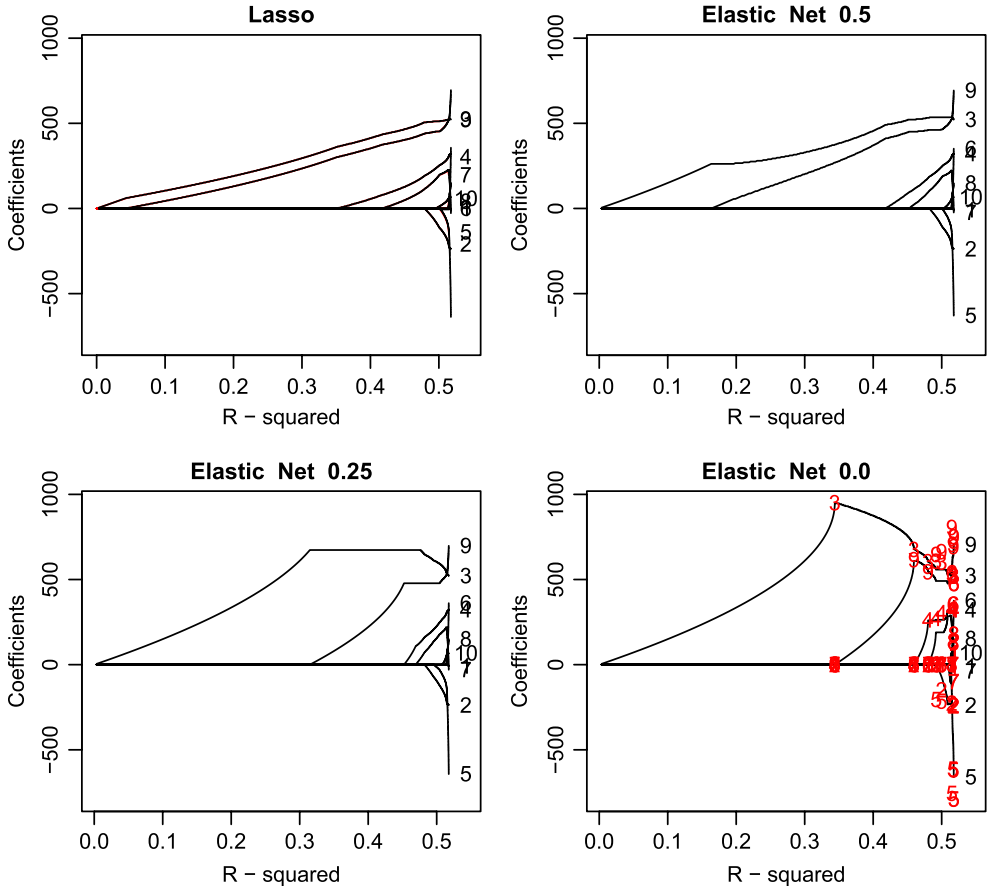

Fig. 3. Paths for the Lasso, and GPS non-convex elastic net penalties $\beta \in \{0.5, 0.25, 0.0\}$, for the diabetes data. The red numbers indicate the forward stepwise solutions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

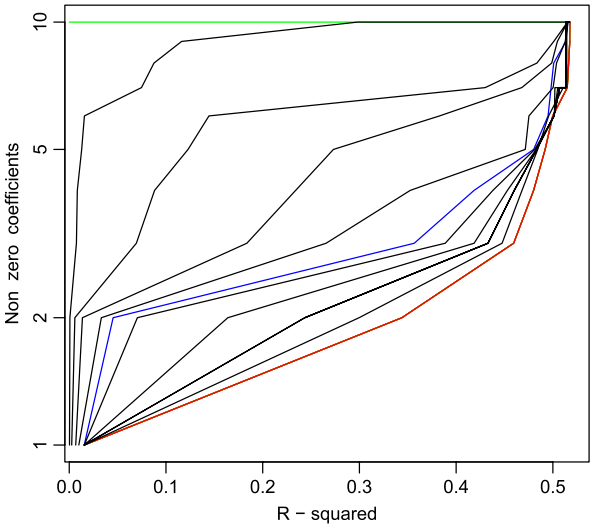

Fig. 4. Number of non-zero coefficient estimates along the respective paths for the diabetes data using elastic net penalties $\beta \in \{2.0$ (ridge, green), 1.99, 1.9, 1.7, 1.5, 1.0 (lasso, blue), 0.7, 0.5, 0.4, 0.3, 0.0$\}$ and stepwise (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

GPS here. The results for $\beta = 1$ (lasso) and $\beta = 2$ (ridge-regression) are highlighted as well (blue and green). From Fig. 4, one sees that at $R^2 \simeq 0.45$, stepwise regression enters three variables, the lasso four variables, and ridge-regression all ten variables. Using these curves as an inverse measure of sparsity, one sees a strict monotonicity

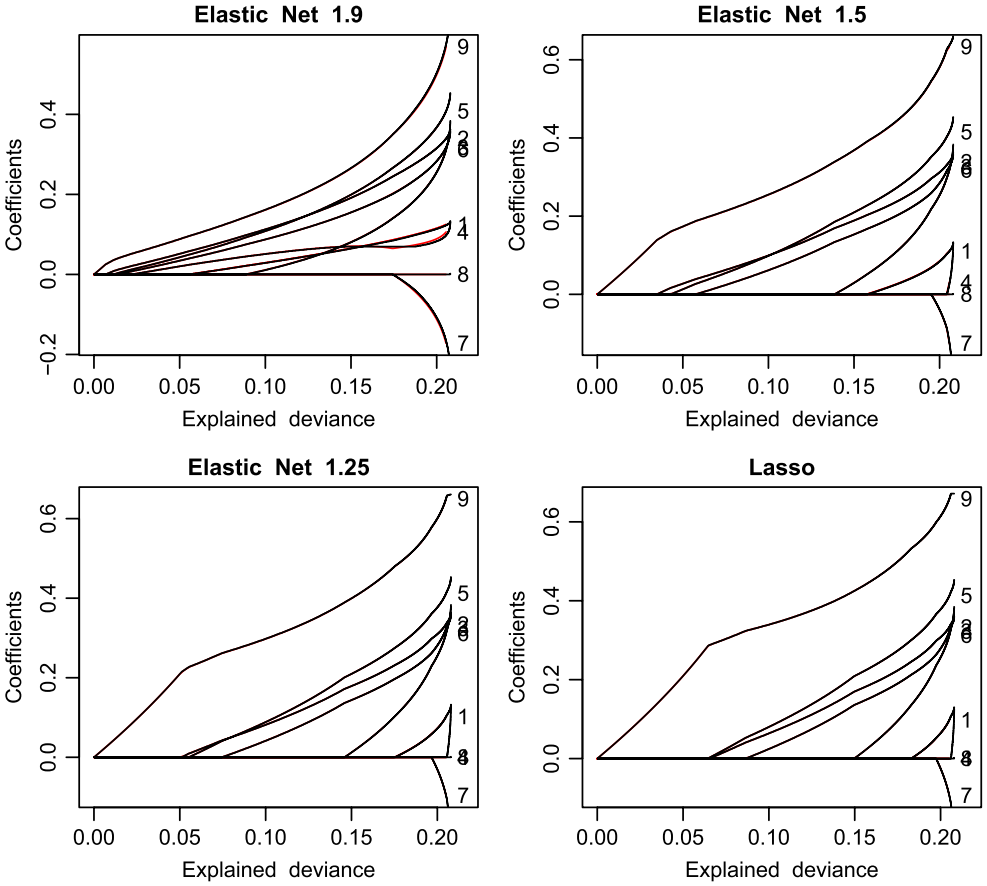

Fig. 5. Exact (red) and GPS (black) coefficient paths for the heart transplant data using convex elastic net penalties $\beta \in \{1.9, 1.5, 1.25, 1.0\}$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

among the members of this family: a smaller β produces sparser solutions at every point on the path, as indexed by the degree of data fit (R^2).

4.2. Logistic regression: South African heart transplant data

This data set was presented by Hastie, Tibshirani, and Friedman (2001). It has $n = 9$ predictor variables and $N = 462$ observations. The outcome variable is binary, so logistic loss (Eq. (5)) is appropriate. Fig. 5 compares the exact (red) and GPS (black) coefficient paths for selected convex members of the generalized elastic net (Eq. (19)) for this data set. The paths here are indexed by the fraction of deviance explained (Eqs. (5)–(8) and (33)). One sees that the coefficient paths produced by the GPS procedure track those for the exact solutions closely. They are identical for $\beta \in \{1.5, 1.25, 1.0 \text{ (lasso)}\}$, owing to the monotonic dependence of the coefficient paths on λ, but there is a slight discrepancy for $\beta = 1.9$, where the path for the coefficient of the fourth variable becomes non-monotonic. Note that here, as in the previous example, the solutions become sparser as β decreases.

Fig. 6 repeats the lasso for comparison, and shows the GPS coefficient paths for various non-convex members of the generalized elastic net (Eq. (20)). Sparsity is seen to continue to increase at each path point as β decreases, with the sparsest solutions being obtained at $\beta = 0$.

4.3. Least-squares regression: under-determined problem

The previous two examples are highly over-determined, in that the number of observations N is considerably larger than the number of predictor variables n. Regularization is less important in such cases than it is for highly under-determined problems where $N \ll n$. In this section, a highly under-determined regression problem is considered. There are $N = 200$ observations and $n = 10{,}000$ predictor variables. The data are simulated from the model

$$y_i = \sum_{j=1}^{n} a_j^* x_{ij} + \varepsilon_i, \tag{36}$$

where the predictor variables are drawn randomly from a normal distribution $\boldsymbol{x}_i \sim N(0, \boldsymbol{C})$ 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 $\boldsymbol{a}^*$ (Eq. (6)) has 30 non-zero coefficients with uniformly distributed absolute values $|a_j^*| = [31 - j]_+$, and alternating signs $\text{sign}(a_{j+1}^*) = -\text{sign}(a_j^*)$, $1 \leq j \leq 29$.

Fig. 7 shows the number of non-zero coefficients (logarithmic scale) as a function of R^2 (Eq. (33)) along the path for forward stepwise regression (red) and a selected

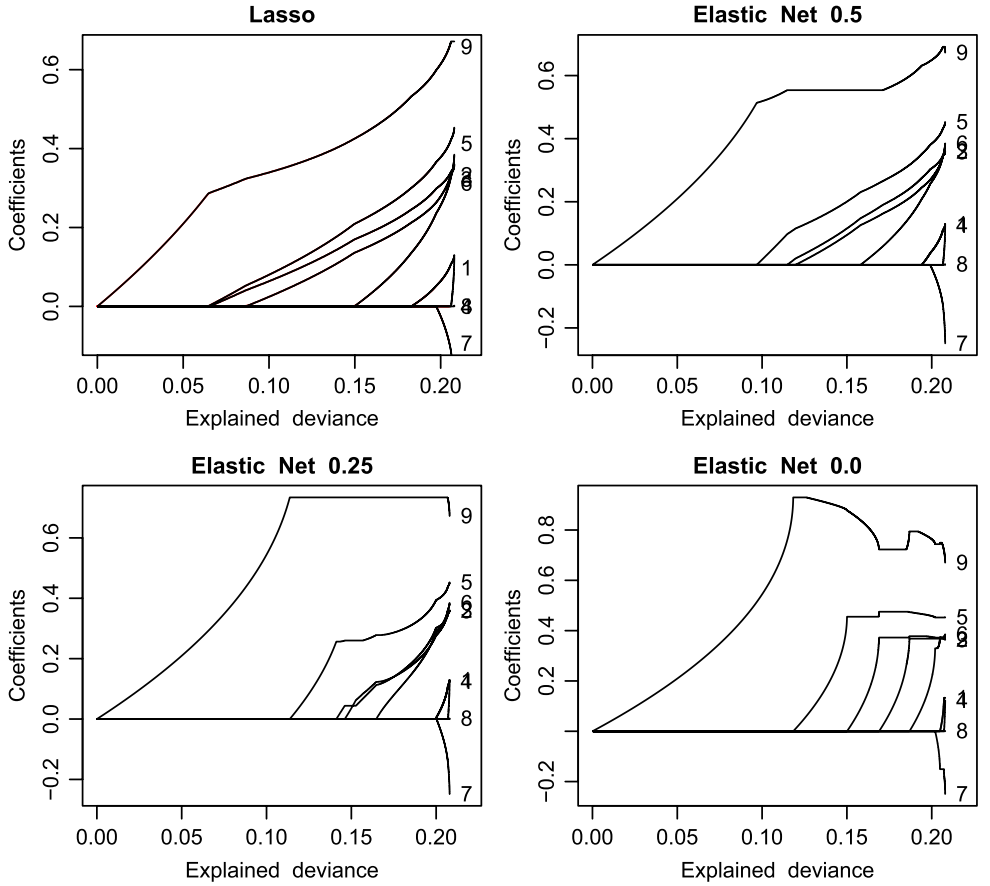

Fig. 6. Coefficient paths for the lasso, and GPS elastic net non-convex penalties $\beta \in \{0.5, 0.25, 0.0\}$, for the heart transplant data.

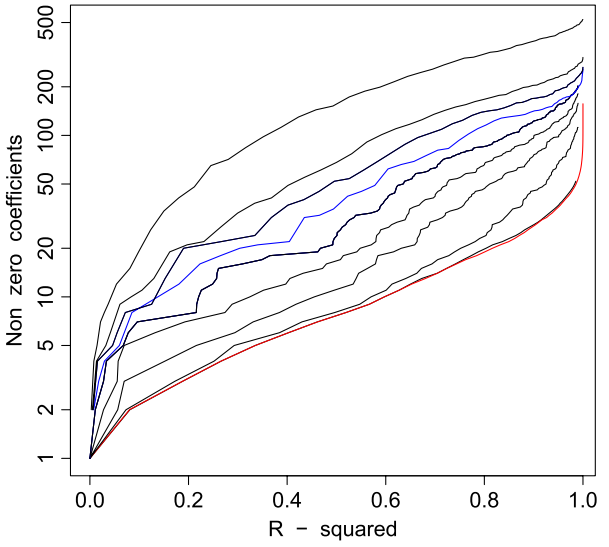

Fig. 7. Number of non-zero coefficient estimates along respective paths for the under-determined regression example ($n = 10{,}000$, $N = 200$) using elastic net penalties $\beta \in \{1.9, 1.7, 1.5, 1.0$ (lasso, blue), $0.5, 0.3, 0.2, 0.1, 0.0\}$, and stepwise (red), top to bottom. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

set of generalized elastic net penalties using squared-error loss. The $\beta = 1$ (lasso) penalty is colored blue. One sees a clear monotonic relationship between the value of β and the sparsity of the corresponding solutions at each path point indexed by R^2 on the training data. For example, at $R^2 = 0.9$, stepwise and $\beta = 0$ GPS have 15 non-zero

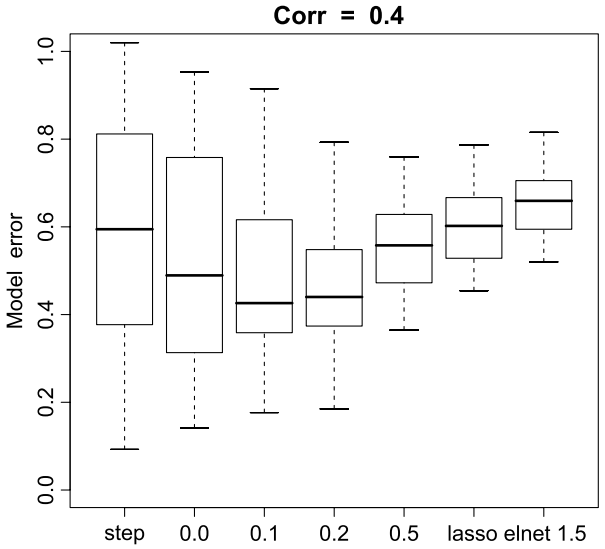

Fig. 8. Inaccuracy of stepwise, several non-convex elastic net GPS, and convex exact paths, over 50 simulated regression data sets with $n = 10{,}000$, $N = 200$ and population correlated predictors (Section 4.3).

coefficients, the lasso has 120, and the $\beta = 1.9$ elastic net has almost 400. Thus, by varying the penalty parameter β (Eqs. (19) and (20)), one can exercise strong control over the sparsity of the induced solutions along its path.

5. Utility

The simplicity of the GPS algorithm makes bridge-regression computationally tractable for fairly large problems. In this section, its potential statistical advantages are investigated.

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

$$\min_{\rho}[R(\hat{\boldsymbol{a}}(\rho)) - R(\boldsymbol{a}^*)]/R(\boldsymbol{a}^*), \tag{37}$$

where $R(\boldsymbol{a}^*)$ is the minimum possible risk associated with the problem (Eq. (6)). This quantity is the minimum distance (Eq. (11)) between points on the path and the optimal solution $\boldsymbol{a}^*$, scaled by $1/R(\boldsymbol{a}^*)$. As was discussed in Section 2.2, paths $\hat{\boldsymbol{a}}(\rho)$ that produce smaller values for Eq. (37) have the potential to produce more accurate predictions, given a model selection procedure such as cross-validation.

Fig. 8 shows the distribution of Eq. (37) (boxplots) for paths produced by several squared-error loss (Eq. (4)) regression methods, applied to 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 $\beta \in \{0.0, 0.1, 0.2, 0.5\}$ (Eq. (20)), and the exact paths produced by the lasso ($\beta = 1$) and the elastic net with $\beta = 1.5$ (Eq. (19)).

From Fig. 8, one sees that the lasso and $\beta = 1.5$ elastic net consistently yield inferior paths on this very sparse problem (30 out of 10,000 true non-zero coefficients). The forward stepwise procedure yields paths of similar accuracy (Eq. (37)) to the lasso on average, but with much more variability. Stepwise paths based on some of the data sets are considerably better than those produced by the lasso, while those from other data sets are considerably worse. For the GPS paths, the variability decreases with an increasing β. The expected performance is best for $\beta = 0.1$ or $\beta = 0.2$, with the latter having less variability.

Fig. 9 shows the distribution over the 50 data sets of the number of non-zero coefficients at the optimal path points minimizing Eq. (37) for each of the respective methods. Here, one sees that forward stepwise regression typically has around 12 non-zero coefficients in its optimal solutions. The slightly less aggressive $\beta = 0$ GPS paths average 14. As β increases, the optimal points on the GPS paths tend towards a reduced 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 respective optimal solutions, with sparse ($\beta = 0.1$ or 0.2), but not the sparsest ($\beta = 0$ or stepwise), being the best (Fig. 8).

Fig. 10 shows results analogous to those in Fig. 8 for a slightly modified problem. Here, for each of the 50 data sets, the predictor variables are drawn from a standard normal distribution, $\boldsymbol{x}_i \sim N(0, \boldsymbol{I})$. 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 Eq. (37), 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 paths produced by the other procedures are all about equally stable. The results for this problem are qualitatively similar to those shown in Fig. 8, with the best stability-expected performance trade-off appearing to be for the $\beta = 0.1$ GPS path. The distributions of the numbers of non-zero coefficients for the optimal solution of each of

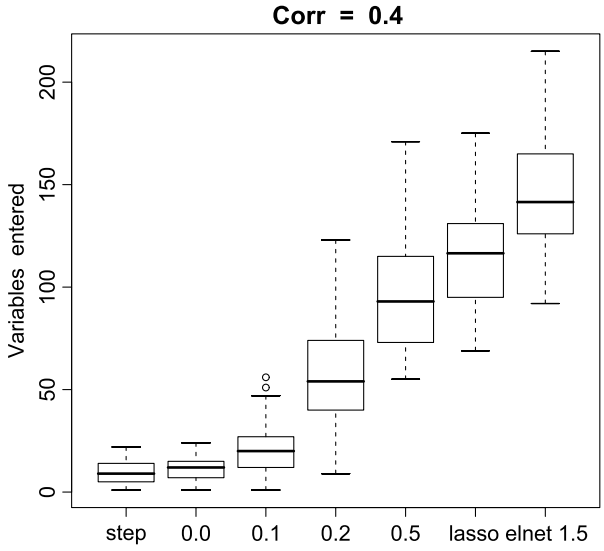

Fig. 9. Number of non-zero coefficients at optimal solutions for stepwise, several non-convex elastic net GPS, and convex exact paths, over the 50 simulated data sets.

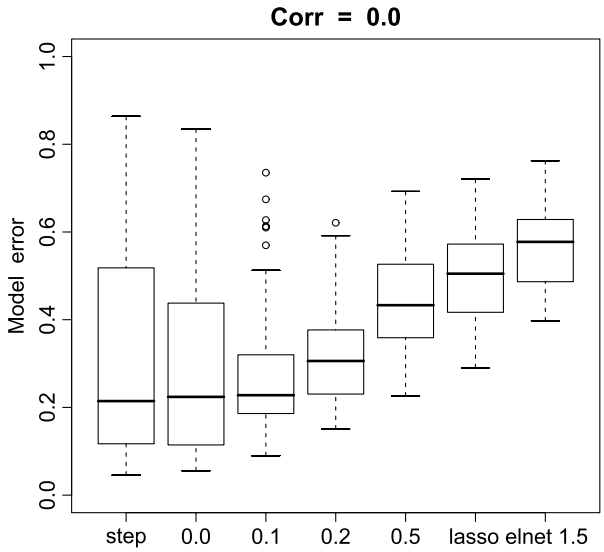

Fig. 10. Inaccuracy of stepwise, several non-convex elastic net GPS, and convex exact paths over 50 simulated data sets with population uncorrelated predictor variables.

the methods (not shown) are quite similar to that shown in Fig. 9.

Fig. 11 shows results analogous to those in Fig. 8 for a slightly different modification of the problem. Here, the joint distribution of the predictor variables is as described in Section 4.3, as are all other aspects of the generating model, except that the signs of the optimal non-zero coefficients are taken to be the same ($\text{sign}(a^*_{j+1}) = \text{sign}(a^*_j) > 0$), instead of alternating. This gives a very different pattern of results. All methods produce much more stable paths. However, their relative qualities (Eq. (37)) are reversed; the worst methods in the previous two problems are the best here, and vice versa. The lasso and the $\beta = 1.5$ elastic net paths dramatically out-perform methods which produce sparser solutions. For this problem, the distributions of the numbers of (optimal solution) non-zero coefficients for each of the methods (not shown) are again quite similar to that shown in Fig. 9. The optimal $\beta = 1.5$ elastic net solutions, which typically involve 150 non-zero coefficients, are far more accurate than methods producing much sparser solutions, even though the population optimal coefficient vector $\boldsymbol{a}^*$ has only 30 non-zero entries.

5.1. Discussion

The results shown in Figs. 8–11 show that the accuracy of a given method for the same population joint distribution can depend strongly on the particular training data set realized from that distribution. This is especially

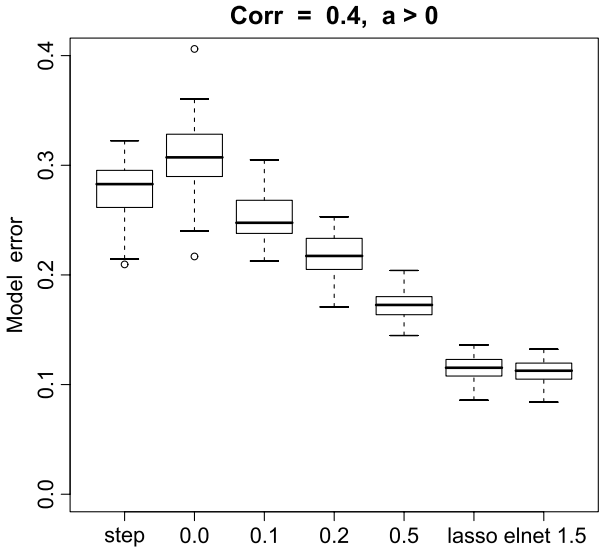

Fig. 11. Inaccuracy of stepwise, several non-convex elastic net GPS, and convex exact paths over 50 simulated data sets with population correlated predictors and with all optimal coefficients having the same sign.

true for methods that induce very sparse paths. In Figs. 8 and 10, one sees that the sparsest methods often produce much better solutions than denser methods on certain data sets, and much worse solutions on others. Thus, comparisons based on one or a small number of data sets (whether simulated or real) can be highly misleading.

A somewhat surprising result from the examples above is that the optimal sparsity of the *estimated* coefficients depends upon more than just the sparsity of the optimal coefficients $\boldsymbol{a}^*$ (Eq. (6)) characterizing the problem. In all three examples, the sparsity (Eq. (15)) of $\boldsymbol{a}^*$ was the same; 30 out of 10,000 non-zero coefficients. In fact, all $\{|a_j^*|\}_1^n$ were identical. For the situation shown in Fig. 8, the best solutions typically involved 50 non-zero coefficients, whereas for that shown in Fig. 10, penalties producing around 20 were best. For the situation shown in Fig. 11, 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 therefore expect that the best penalty would depend mainly on the relative absolute values of the optimal coefficients $\{|a_j^*|\}_1^n$ (Eq. (6)). Thus, as was discussed in Section 2.3, this knowledge (if available) would drive penalty choice. The examples presented here show that this is not necessarily 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 sparser solutions than the lasso are not always better than the lasso solutions, even in sparse situations, such as those characterized by the optimal coefficients $\boldsymbol{a}^*$. Even with a knowledge of the latter, the other aspects of the problem that influence the choice of a good penalty are likely to be unknown. Thus, using bridge regression to aid penalty choice can be helpful.

6. Variable selection

A common practice in regression or classification problems is to employ a two-stage process. One first attempts to identify the relevant predictor variables, then performs an ordinary regression (Eqs. (7) and (8)) using only the variables so selected. There are a wide variety of variable selection methods. One which is frequently used involves employing Eq. (10) with a sparsity inducing penalty $P(\boldsymbol{a})$. At each path point $\lambda(\infty \geq \lambda \geq 0)$, the variables corresponding to the non-zero coefficients $\{x_j|\hat{a}_j(\lambda) \neq 0\}$ are selected for the post regression (Eqs. (7) and (8)). This induces a sequence of models with a generally increasing number of variables selected, as the value of λ decreases. A model selection procedure (Section 2.1) can then be used to to estimate the best one in this sequence. In this way, the regularized procedure (Eq. (10)) serves as a "selector" for the post regression (Eqs. (7) and (8)).

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

Figs. 12–15 show the corresponding results of using the various GPS and exact convex procedures as selectors in the set of situations represented in Figs. 8–11, respectively. The dashed (red) lines on each boxplot represent the medians of the corresponding distributions in Figs. 8–11. The results for forward stepwise regression are (by construction) identical, and are repeated for comparison.

From Fig. 13, one sees that the selector-based optimal solutions (Eq. (37)) are sparser than those for the

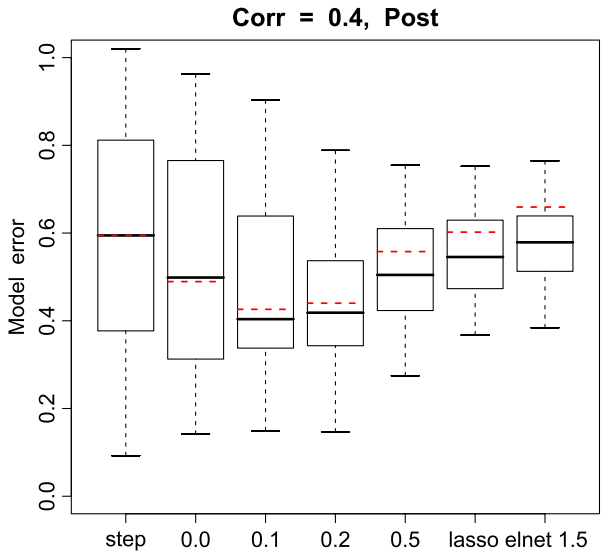

Fig. 12. Results for Fig. 8 data when the respective methods are used as variable selectors. The dashed red lines are the corresponding medians from the Fig. 8 distributions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

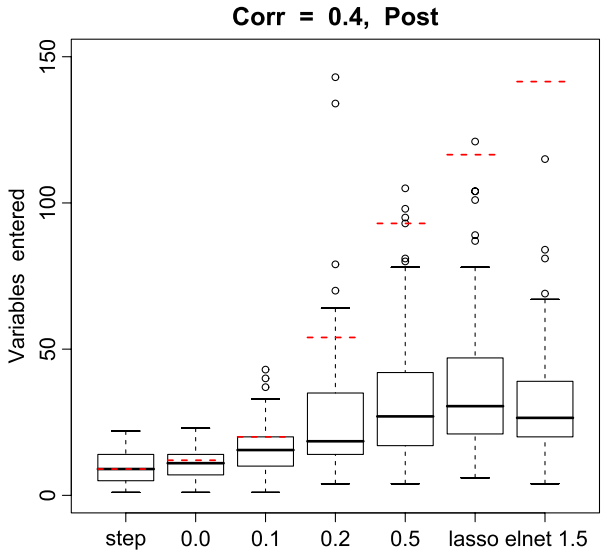

Fig. 13. Number of non-zero coefficients at the optimal solutions when the methods shown in Fig. 9 are used as selectors. The dashed red lines are the corresponding medians from the Fig. 9 distributions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

corresponding direct solutions (Fig. 9) 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 when the corresponding methods are used directly. This is seen to increase the accuracy in those situations (Figs. 8 and 10) where the sparser direct methods ($\beta < 1$) provide superior results. Again, this accuracy increase is more pronounced for larger values of 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 shown in Fig. 11, where the direct convex methods were seen to provide the best performance, using them as selectors *decreases* their accuracy (Fig. 15). In this situation, using the $\beta < 1$ GPS procedures as selectors improves their performance, but not enough to compete with the direct convex methods which produce the densest solutions.

As with the direct methods, the success of the selector strategy in improving the performance is seen to depend on more than just the sparsity of the optimal coefficients $\boldsymbol{a}^*$; other factors, including their signs and the correlational structure of the predictor variable distribution, are seen to be relevant too. The results shown in Figs. 12–15 indicate that the best direct methods give rise to the best

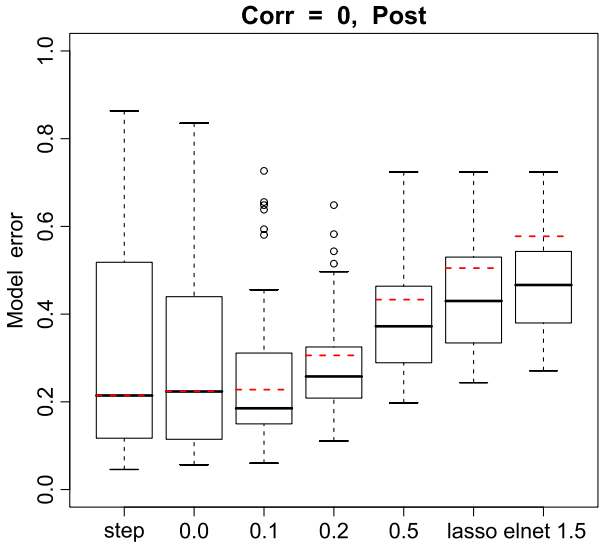

Fig. 14. Results for Fig. 10 data when the respective methods are used as selectors. The dashed red lines are the corresponding medians from the Fig. 10 distributions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

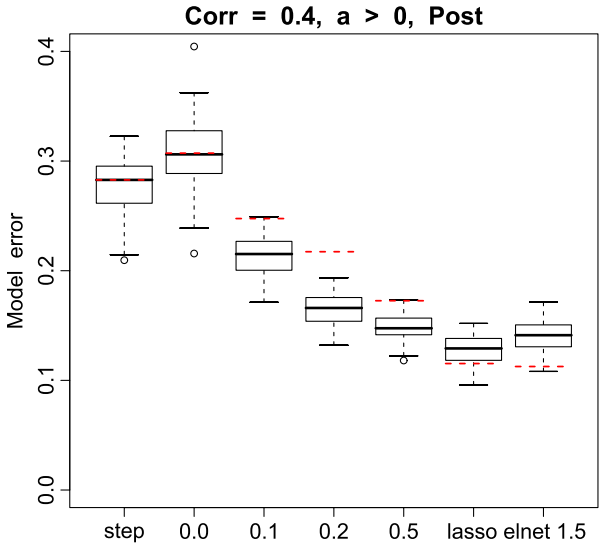

Fig. 15. Results for Fig. 11 data when the respective methods are used as selectors. The dashed red lines are the corresponding medians from the Fig. 11 distributions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

selectors. That is, when the sparser direct methods are superior, it is due at least in part to their better variable selection properties. For all of the methods, the results shown in Figs. 12–15 are based on the same procedure (Eqs. (7) and (8)) for estimating the coefficients, given the selected variables for each potential model; and thus, the differences in performance seen there reflect solely on the ability of each of the respective methods to select the best variables for prediction.

Fig. 14 shows that, for the case of population uncorrelated predictor variables, the highly non-convex $\beta = 0.1$ generalized elastic net penalty performed best for variable selection, being more than twice as effective as the lasso, as measured by Eq. (37). This is somewhat surprising, because they would have the same performances for *sample* uncorrelated predictors. In fact, the results for all of the procedures considered here would be the same in that case, namely that of all subsets regression. This fact is often used as the motivation for the use of convex methods for variable selection in low correlation situations. The results shown in Fig. 14 suggest that, even in population uncorrelated settings, the empirical data correlations induced by sampling can be large enough to severely degrade the performance of the lasso as a variable selector in highly underdetermined problems, such as that represented in Fig. 14 ($N = 200, n = 10{,}000$).

Note that the most aggressive selector considered here, forward stepwise regression, is also inferior to the less aggressive $\beta = 0.1$ generalized elastic net procedure. This is due to the higher degree of variability in the order in which it enters the predictor variables over the 50 data sets.

7. 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 (Eq. (2)) is seldom penalized. Let $\{a_l\}_{l\in L}$ be a specified set of unpenalized parameters, where $L \subset \{1, 2, \ldots, n\}$. Then $\{p_l(\nu) = 0\}_{l\in L}$ (Eq. (24)) at all path points ν. This violates Eq. (22), which can be remedied by setting $\{p_l(\nu) = \varepsilon\}_{l\in L}$, so that the corresponding lambdas (Eq. (25)) become $\{\lambda_l(\nu) = g_l(\nu)/\varepsilon\}_{l\in L}$, with ε being a very small positive quantity. Each such an $\lambda_l(\nu)$ 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 $\hat{R}(\boldsymbol{a})$ (Eq. (8)) evaluated at $\hat{\boldsymbol{a}}(\nu)$, 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

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

In the case of squared-error loss (Eq. (4)), forcing the predictor variables to all have zero means causes the derivative $g_0(\nu)$ (Eq. (23)) 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

$$\hat{a}_0(\nu) = \arg\min_{a_0} \left(\hat{R}(\boldsymbol{a})|\{\hat{a}_j(\nu)\}_1^n\right). \tag{38}$$

When this is the case (for example, with the logistic loss in Eq. (5)), applying Eq. (38) at every iteration will increase the speed of the GPS algorithm by reducing the number of steps.

8. Step size

For a given loss $L(y, F)$ (Eq. (3)) and penalty $P(\boldsymbol{a})$ (Eq. (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 $\Delta\nu \to 0$, 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 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 (Eq. (8)) by a fixed fraction

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

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

9. Related work

There is a large body of literature pertaining to regularized regression and classification. Most of the work involves the use of convex loss functions with convex penalties, so that the overall criterion (Eq. (10)) is convex. Standard algorithms for convex optimization problems (Boyd & Vandenberghe, 2004) can then be employed to solve Eq. (10) repeatedly 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, & De Mol, 2004; Wu & Lange, 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 (Eq. (19)) by Friedman, Hastie, Höfling, and Tibshirani (2007) and Van der Kooij (2007). The one-at-a-time coordinate descent strategy was applied to regularized logistic and multinomial regression by Genkin, Lewis, and Madigan (2007) and Krishnapuram, Carlin, Figueiredo, and Hartemink (2005), 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 their speed can rival that of GPS for these 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 Eq. (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 Eq. (10) for square-error loss at each path point. Neither of these methods is 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 provided a fast algorithm for generating paths for squared-error loss. Mazumder, Friedman, and Hastie (2011) applied the coordinate descent method to certain non-convex penalties (including MC+) with squared-error loss, using a special penalty relaxation strategy to direct solutions to good local optima. Again, the GPS algorithm can be used to generalize these methods to any convex loss and a wider selection of penalties.

Direct path seeking algorithms for producing paths which are 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 is related 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 for creating sparser paths. The relaxed lasso (Meinshausen, 2007) and VISA (Radchenko & James, 2008) use the lasso as the basic selector. The Dantzig selector (Candes & Tao, 2007) uses a different convex constrained procedure for variable selection along its path, with properties similar to the lasso. Although these selectors are generally faster than exact methods based on non-convex penalties, they 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 was illustrated in Section 6.

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

10. Discussion

The principal advantages of using GPS to generate paths based on chosen loss-penalty combinations are its 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 Eq. (22).

As was seen in Section 5, the best penalty for any given application can depend strongly on various different aspects of the particular problem. These include the actual sparsity of the optimal coefficients $\boldsymbol{a}^*$ (Eq. (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 (Eqs. (16) and (17)) can aid in the penalty choice. Again, the speed of GPS makes this feasible for large problems.

Acknowledgments

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

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

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

Buhlmann, P., & Yu, B. (2006). Sparse boosting. *Journal of Machine Learning Research*, *7*, 1001–1024.

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

Daubechines, I., DeFrise, M., & 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.

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

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

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

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

Friedman, J. H., Hastie, T., Höfling, H., & Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics*, *1*, 302–332.

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

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

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

Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). *Elements of statistical learning: data mining, inference and prediction*. New York: Springer-Verlag.

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

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

Krishnapuram, B., Carlin, L., Figueiredo, M. A. T., & Hartemink, A. J. (2005). Sparse multinomial logistic regression: fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*, 957–968.

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

Mazumder, R., Friedman, J. H., & Hastie, T. (2011). SparseNet: coordinate descent with non-convex penalties. *Journal of the American Statistical Association*, *106*(495), 1125–1138.

Meinshausen, N. (2007). Relaxed lasso. *Computational Statistics and Data Analysis*, *52*, 374–393.

Radchenko, P., & James, G. (2008). Variable inclusion and shrinkage algorithms. *Journal of the American Statistical Association*, *103*, 1304–1315.

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

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

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

Wold, S., Ruhe, A., Wold, H., & Dunn III, W. J. (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.

Wu, T., & Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics*, *2*, 224–244.

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

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

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.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, *67*, 301–320.