J. R. Statist. Soc. B (2005)
67, *Part* 2, *pp.* 301–320

Regularization and variable selection via the elastic net

Hui Zou and Trevor Hastie

Stanford University, USA

[Received December 2003. Final revision September 2004]

Summary. We propose the elastic net, a new regularization and variable selection method. Real world data and a simulation study show that the elastic net often outperforms the lasso, while enjoying a similar sparsity of representation. In addition, the elastic net encourages a grouping effect, where strongly correlated predictors tend to be in or out of the model together. The elastic net is particularly useful when the number of predictors (p) is much bigger than the number of observations (n). By contrast, the lasso is not a very satisfactory variable selection method in the $p \gg n$ case. An algorithm called LARS-EN is proposed for computing elastic net regularization paths efficiently, much like algorithm LARS does for the lasso.

Keywords: Grouping effect; LARS algorithm; Lasso; Penalization; $p \gg n$ problem; Variable selection

1. Introduction and motivation

We consider the usual linear regression model: given p predictors $\mathbf{x}_1, \ldots, \mathbf{x}_p$, the response $\mathbf{y}$ is predicted by

$$\hat{\mathbf{y}} = \hat{\beta}_0 + \mathbf{x}_1\hat{\beta}_1 + \ldots + \mathbf{x}_p\hat{\beta}_p. \tag{1}$$

A model fitting procedure produces the vector of coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)$. For example, the ordinary least squares (OLS) estimates are obtained by minimizing the residual sum of squares. The criteria for evaluating the quality of a model will differ according to the circumstances. Typically the following two aspects are important:

(a) accuracy of prediction on future data—it is difficult to defend a model that predicts poorly;
(b) interpretation of the model—scientists prefer a simpler model because it puts more light on the relationship between the response and covariates. Parsimony is especially an important issue when the number of predictors is large.

It is well known that OLS often does poorly in both prediction and interpretation. Penalization techniques have been proposed to improve OLS. For example, ridge regression (Hoerl and Kennard, 1988) minimizes the residual sum of squares subject to a bound on the L_2-norm of the coefficients. As a continuous shrinkage method, ridge regression achieves its better prediction performance through a bias–variance trade-off. However, ridge regression cannot produce a parsimonious model, for it always keeps all the predictors in the model. Best subset selection in

Address for correspondence: Trevor Hastie, Department of Statistics, Stanford University, Stanford, CA 94305, USA.
E-mail: hastie@stanford.edu

© 2005 Royal Statistical Society 1369–7412/05/67301

contrast produces a sparse model, but it is extremely variable because of its inherent discreteness, as addressed by Breiman (1996).

A promising technique called the lasso was proposed by Tibshirani (1996). The lasso is a penalized least squares method imposing an L_1-penalty on the regression coefficients. Owing to the nature of the L_1-penalty, the lasso does both continuous shrinkage and automatic variable selection simultaneously. Tibshirani (1996) and Fu (1998) compared the prediction performance of the lasso, ridge and bridge regression (Frank and Friedman, 1993) and found that none of them uniformly dominates the other two. However, as variable selection becomes increasingly important in modern data analysis, the lasso is much more appealing owing to its sparse representation.

Although the lasso has shown success in many situations, it has some limitations. Consider the following three scenarios.

(a) In the $p > n$ case, the lasso selects at most n variables before it saturates, because of the nature of the convex optimization problem. This seems to be a limiting feature for a variable selection method. Moreover, the lasso is not well defined unless the bound on the L_1-norm of the coefficients is smaller than a certain value.
(b) If there is a group of variables among which the pairwise correlations are very high, then the lasso tends to select only one variable from the group and does not care which one is selected. See Section 2.3.
(c) For usual $n > p$ situations, if there are high correlations between predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression (Tibshirani, 1996).

Scenarios (a) and (b) make the lasso an inappropriate variable selection method in some situations. We illustrate our points by considering the gene selection problem in microarray data analysis. A typical microarray data set has many thousands of predictors (genes) and often fewer than 100 samples. For those genes sharing the same biological 'pathway', the correlations between them can be high (Segal and Conklin, 2003). We think of those genes as forming a group. The ideal gene selection method should be able to do two things: eliminate the trivial genes and automatically include whole groups into the model once one gene among them is selected ('grouped selection'). For this kind of $p \gg n$ and grouped variables situation, the lasso is not the ideal method, because it can only select at most n variables out of p candidates (Efron *et al.*, 2004), and it lacks the ability to reveal the grouping information. As for prediction performance, scenario (c) is not rare in regression problems. So it is possible to strengthen further the prediction power of the lasso.

Our goal is to find a new method that works as well as the lasso whenever the lasso does the best, and can fix the problems that were highlighted above, i.e. it should mimic the ideal variable selection method in scenarios (a) and (b), especially with microarray data, and it should deliver better prediction performance than the lasso in scenario (c).

In this paper we propose a new regularization technique which we call the *elastic net*. Similar to the lasso, the elastic net simultaneously does automatic variable selection and continuous shrinkage, and it can select groups of correlated variables. It is like a stretchable fishing net that retains 'all the big fish'. Simulation studies and real data examples show that the elastic net often outperforms the lasso in terms of prediction accuracy.

In Section 2 we define the *naïve elastic net*, which is a penalized least squares method using a novel *elastic net penalty*. We discuss the grouping effect that is caused by the elastic net penalty. In Section 3, we show that this naïve procedure tends to overshrink in regression problems. We then introduce the elastic net, which corrects this problem. An efficient algorithm LARS-EN is

proposed for computing the entire elastic net regularization paths with the computational effort of a single OLS fit. Prostate cancer data are used to illustrate our methodology in Section 4, and simulation results comparing the lasso and the elastic net are presented in Section 5. Section 6 shows an application of the elastic net to classification and gene selection in a leukaemia microarray problem.

2. Naïve elastic net

2.1. Definition

Suppose that the data set has n observations with p predictors. Let $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ be the response and $\mathbf{X} = (\mathbf{x}_1 | \ldots | \mathbf{x}_p)$ be the model matrix, where $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^{\mathrm{T}}$, $j = 1, \ldots, p$, are the predictors. After a location and scale transformation, we can assume that the response is centred and the predictors are standardized,

$$\sum_{i=1}^{n} y_i = 0, \qquad \sum_{i=1}^{n} x_{ij} = 0 \qquad \text{and} \quad \sum_{i=1}^{n} x_{ij}^2 = 1, \qquad \text{for } j = 1, 2, \ldots, p. \tag{2}$$

For any fixed non-negative λ_1 and λ_2, we define the naïve elastic net criterion

$$L(\lambda_1, \lambda_2, \boldsymbol{\beta}) = |\mathbf{y} - \mathbf{X}\boldsymbol{\beta}|^2 + \lambda_2 |\boldsymbol{\beta}|^2 + \lambda_1 |\boldsymbol{\beta}|_1, \tag{3}$$

where

$$|\boldsymbol{\beta}|^2 = \sum_{j=1}^{p} \beta_j^2,$$

$$|\boldsymbol{\beta}|_1 = \sum_{j=1}^{p} |\beta_j|.$$

The naïve elastic net estimator $\hat{\boldsymbol{\beta}}$ is the minimizer of equation (3):

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \{L(\lambda_1, \lambda_2, \boldsymbol{\beta})\}. \tag{4}$$

This procedure can be viewed as a penalized least squares method. Let $\alpha = \lambda_2/(\lambda_1 + \lambda_2)$; then solving $\hat{\boldsymbol{\beta}}$ in equation (3) is equivalent to the optimization problem

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} |\mathbf{y} - \mathbf{X}\boldsymbol{\beta}|^2, \qquad \text{subject to } (1-\alpha)\,|\boldsymbol{\beta}|_1 + \alpha\,|\boldsymbol{\beta}|^2 \leqslant t \text{ for some } t. \tag{5}$$

We call the function $(1-\alpha)\,|\boldsymbol{\beta}|_1 + \alpha\,|\boldsymbol{\beta}|^2$ the elastic net penalty, which is a convex combination of the lasso and ridge penalty. When $\alpha = 1$, the naïve elastic net becomes simple ridge regression. In this paper, we consider only $\alpha < 1$. For all $\alpha \in [0, 1)$, the elastic net penalty function is singular (without first derivative) at 0 and it is strictly convex for all $\alpha > 0$, thus having the characteristics of both the lasso and ridge regression. Note that the lasso penalty ($\alpha = 0$) is convex but not strictly convex. These arguments can be seen clearly from Fig. 1.

2.2. Solution

We now develop a method to solve the naïve elastic net problem efficiently. It turns out that minimizing equation (3) is equivalent to a lasso-type optimization problem. This fact implies that the naïve elastic net also enjoys the computational advantage of the lasso.

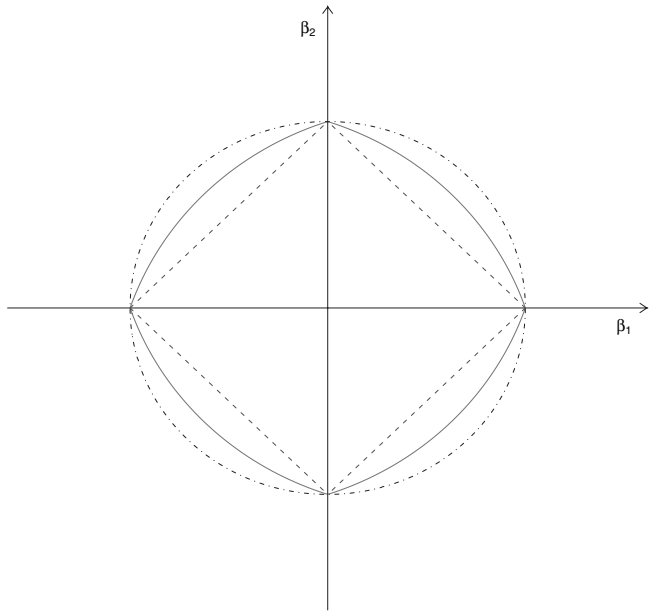

Fig. 1. Two-dimensional contour plots (level 1) (· - - - -, shape of the ridge penalty; -------, contour of the lasso penalty; ———, contour of the elastic net penalty with $\alpha = 0.5$): we see that singularities at the vertices and the edges are strictly convex; the strength of convexity varies with α

Lemma 1. Given data set $(\mathbf{y}, \mathbf{X})$ and (λ_1, λ_2), define an artificial data set $(\mathbf{y}^*, \mathbf{X}^*)$ by

$$\mathbf{X}^*_{(n+p)\times p} = (1+\lambda_2)^{-1/2} \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda_2}\mathbf{I} \end{pmatrix}, \qquad \mathbf{y}^*_{(n+p)} = \begin{pmatrix} \mathbf{y} \\ 0 \end{pmatrix}.$$

Let $\gamma = \lambda_1/\sqrt{(1+\lambda_2)}$ and $\boldsymbol{\beta}^* = \sqrt{(1+\lambda_2)}\boldsymbol{\beta}$. Then the naïve elastic net criterion can be written as

$$L(\gamma, \boldsymbol{\beta}) = L(\gamma, \boldsymbol{\beta}^*) = \left|\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta}^*\right|^2 + \gamma \left|\boldsymbol{\beta}^*\right|_1.$$

Let

$$\hat{\boldsymbol{\beta}}^* = \arg\min_{\boldsymbol{\beta}^*} L\{(\gamma, \boldsymbol{\beta}^*)\};$$

then

$$\hat{\boldsymbol{\beta}} = \frac{1}{\sqrt{(1+\lambda_2)}}\hat{\boldsymbol{\beta}}^*.$$

The proof is just simple algebra, which we omit. Lemma 1 says that we can transform the naïve elastic net problem into an equivalent lasso problem on augmented data. Note that the sample size in the augmented problem is $n + p$ and $\mathbf{X}^*$ has rank p, which means that the naïve elastic net can potentially select all p predictors in all situations. This important property overcomes the limitations of the lasso that were described in scenario (a). Lemma 1 also shows that the naïve elastic net can perform an automatic variable selection in a fashion similar to the lasso. In the next section we show that the naïve elastic net has the ability of selecting 'grouped' variables, a property that is not shared by the lasso.

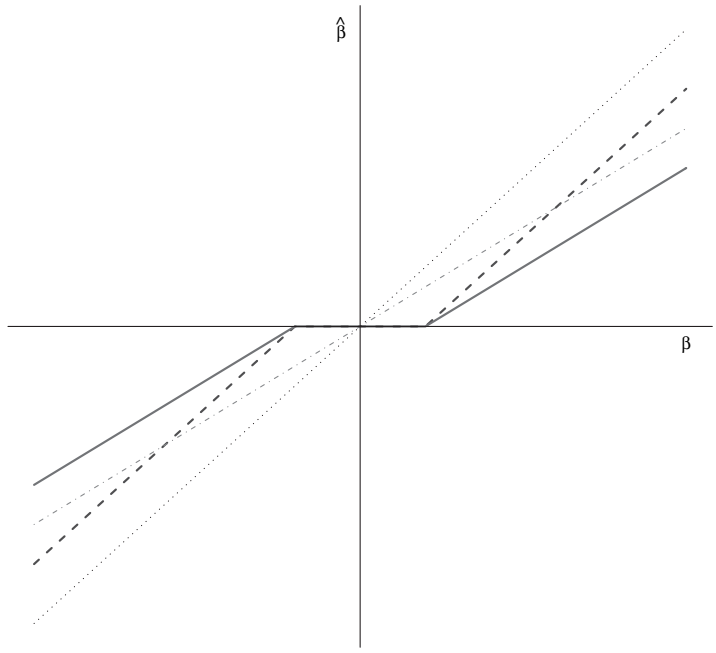

Fig. 2. Exact solutions for the lasso (-------), ridge regression (· · · · · ·) and the naïve elastic net (———) in an orthogonal design (· · · · · · ·, OLS): the shrinkage parameters are $\lambda_1 = 2$ and $\lambda_2 = 1$

In the case of an orthogonal design, it is straightforward to show that with parameters (λ_1, λ_2) the naïve elastic net solution is

$$\hat{\beta}_i(\text{naïve elastic net}) = \frac{(|\hat{\beta}_i(\text{OLS})| - \lambda_1/2)_+}{1+\lambda_2}\text{sgn}\{\hat{\beta}_i(\text{OLS})\}, \tag{6}$$

where $\hat{\boldsymbol{\beta}}(\text{OLS}) = \mathbf{X}^{\text{T}}\mathbf{y}$ and z_+ denotes the positive part, which is z if $z > 0$ and 0 otherwise. The solution of ridge regression with parameter λ_2 is given by $\hat{\boldsymbol{\beta}}(\text{ridge}) = \hat{\boldsymbol{\beta}}(\text{OLS})/(1+\lambda_2)$, and the lasso solution with parameter λ_1 is

$$\hat{\beta}_i(\text{lasso}) = (|\hat{\beta}_i(\text{OLS})| - \lambda_1/2)_+ \,\text{sgn}\{\hat{\beta}_i(\text{OLS})\}.$$

Fig. 2 shows the operational characteristics of the three penalization methods in an orthogonal design, where the naïve elastic net can be viewed as a two-stage procedure: a ridge-type direct shrinkage followed by a lasso-type thresholding.

2.3. The grouping effect

In the 'large p, small n' problem (West *et al.*, 2001), the 'grouped variables' situation is a particularly important concern, which has been addressed many times in the literature. For example, principal component analysis has been used to construct methods for finding a set of highly correlated genes in Hastie *et al.* (2000) and Díaz-Uriarte (2003). Tree harvesting (Hastie *et al.*, 2003) uses supervised learning methods to select groups of predictive genes found by hierarchical clustering. Using an algorithmic approach, Dettling and Bühlmann (2004) performed the clustering and supervised learning together. A careful study by Segal and Conklin (2003) strongly motivates the use of a regularized regression procedure to find the grouped genes. We consider the generic penalization method

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} |\mathbf{y} - \mathbf{X}\boldsymbol{\beta}|^2 + \lambda\, J(\boldsymbol{\beta}) \tag{7}$$

where $J(\cdot)$ is positive valued for $\boldsymbol{\beta} \neq 0$.

Qualitatively speaking, a regression method exhibits the grouping effect if the regression coefficients of a group of highly correlated variables tend to be equal (up to a change of sign if negatively correlated). In particular, in the extreme situation where some variables are exactly identical, the regression method should assign identical coefficients to the identical variables.

Lemma 2. Assume that $\mathbf{x}_i = \mathbf{x}_j$, $i, j \in \{1, \ldots, p\}$.

(a) If $J(\cdot)$ is strictly convex, then $\hat{\beta}_i = \hat{\beta}_j$, $\forall \lambda > 0$.
(b) If $J(\boldsymbol{\beta}) = |\boldsymbol{\beta}|_1$, then $\hat{\beta}_i \hat{\beta}_j \geqslant 0$ and $\hat{\boldsymbol{\beta}}^*$ is another minimizer of equation (7), where

$$\hat{\beta}_k^* = \begin{cases} \hat{\beta}_k & \text{if } k \neq i \text{ and } k \neq j, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (s) & \text{if } k = i, \\ (\hat{\beta}_i + \hat{\beta}_j) \cdot (1 - s) & \text{if } k = j, \end{cases}$$

for any $s \in [0, 1]$.

Lemma 2 shows a clear distinction between *strictly* convex penalty functions and the lasso penalty. Strict convexity guarantees the grouping effect in the extreme situation with identical predictors. In contrast the lasso does not even have a unique solution. The elastic net penalty with $\lambda_2 > 0$ is strictly convex, thus enjoying the property in assertion (1).

Theorem 1. Given data $(\mathbf{y}, \mathbf{X})$ and parameters (λ_1, λ_2), the response $\mathbf{y}$ is centred and the predictors $\mathbf{X}$ are standardized. Let $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ be the naïve elastic net estimate. Suppose that $\hat{\beta}_i(\lambda_1, \lambda_2)\, \hat{\beta}_j(\lambda_1, \lambda_2) > 0$. Define

$$D_{\lambda_1, \lambda_2}(i, j) = \frac{1}{|\mathbf{y}|_1} |\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)|;$$

then

$$D_{\lambda_1, \lambda_2}(i, j) \leqslant \frac{1}{\lambda_2} \sqrt{\{2(1 - \rho)\}},$$

where $\rho = \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j$, the sample correlation.

The unitless quantity $D_{\lambda_1, \lambda_2}(i, j)$ describes the difference between the coefficient paths of predictors i and j. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are highly correlated, i.e. $\rho \doteq 1$ (if $\rho \doteq -1$ then consider $-\mathbf{x}_j$), theorem 1 says that the difference between the coefficient paths of predictor i and predictor j is almost 0. The upper bound in the above inequality provides a quantitative description for the grouping effect of the naïve elastic net.

The lasso does not have the grouping effect. Scenario (b) in Section 1 occurs frequently in practice. A theoretical explanation is given in Efron *et al.* (2004). For a simpler illustration, let us consider the linear model with $p = 2$. Tibshirani (1996) gave the explicit expression for $(\hat{\beta}_1, \hat{\beta}_2)$, from which we easily obtain that $|\hat{\beta}_1 - \hat{\beta}_2| = |\cos(\theta)|$, where θ is the angle between $\mathbf{y}$ and $\mathbf{x}_1 - \mathbf{x}_2$. It is easy to construct examples such that $\rho = \mathrm{corr}(\mathbf{x}_1, \mathbf{x}_2) \to 1$ but $\cos(\theta)$ does not vanish.

2.4. Bayesian connections and the L_q-penalty

Bridge regression (Frank and Friedman, 1993; Fu, 1998) has $J(\boldsymbol{\beta}) = |\boldsymbol{\beta}|_q^q = \Sigma_{j=1}^p |\boldsymbol{\beta}_j|^q$ in equation (7), which is a generalization of both the lasso ($q = 1$) and ridge regression ($q = 2$). The

bridge estimator can be viewed as the Bayes posterior mode under the prior

$$p_{\lambda,q}(\boldsymbol{\beta}) = C(\lambda, q)\exp(-\lambda|\boldsymbol{\beta}|_q^q). \tag{8}$$

Ridge regression ($q = 2$) corresponds to a Gaussian prior and the lasso ($q = 1$) a Laplacian (or double-exponential) prior. The elastic net penalty corresponds to a new prior given by

$$p_{\lambda,\alpha}(\boldsymbol{\beta}) = C(\lambda, \alpha)\exp[-\lambda\{\alpha|\boldsymbol{\beta}|^2 + (1-\alpha)|\boldsymbol{\beta}|_1\}], \tag{9}$$

a compromise between the Gaussian and Laplacian priors. Although bridge regression with $1 < q < 2$ will have many similarities with the elastic net, there is a fundamental difference between them. The elastic net produces *sparse* solutions, whereas bridge regression does not. Fan and Li (2001) proved that, in the L_q ($q \geqslant 1$) penalty family, only the lasso penalty ($q = 1$) can produce a sparse solution. Bridge regression ($1 < q < 2$) always keeps all predictors in the model, as does ridge regression. Since automatic variable selection via penalization is a primary objective of this paper, L_q ($1 < q < 2$) penalization is not a candidate.

3. Elastic net

3.1. Deficiency of the naïve elastic net

As an automatic variable selection method, the naïve elastic net overcomes the limitations of the lasso in scenarios (a) and (b). However, empirical evidence (see Sections 4 and 5) shows that the naïve elastic net does not perform satisfactorily unless it is very close to either ridge regression or the lasso. This is why we call it *naïve*.

In the regression prediction setting, an accurate penalization method achieves good prediction performance through the bias–variance trade-off. The naïve elastic net estimator is a two-stage procedure: for each fixed λ_2 we first find the ridge regression coefficients, and then we do the lasso-type shrinkage along the lasso coefficient solution paths. It appears to incur a double amount of shrinkage. Double shrinkage does not help to reduce the variances much and introduces unnecessary extra bias, compared with pure lasso or ridge shrinkage. In the next section we improve the prediction performance of the naïve elastic net by correcting this double shrinkage.

3.2. The elastic net estimate

We follow the notation in Section 2.2. Given data $(\mathbf{y}, \mathbf{X})$, penalty parameter (λ_1, λ_2) and augmented data $(\mathbf{y}^*, \mathbf{X}^*)$, the naïve elastic net solves a lasso-type problem

$$\hat{\boldsymbol{\beta}}^* = \arg\min_{\boldsymbol{\beta}^*} |\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta}^*|^2 + \frac{\lambda_1}{\sqrt{(1+\lambda_2)}}|\boldsymbol{\beta}^*|_1. \tag{10}$$

The elastic net (corrected) estimates $\hat{\boldsymbol{\beta}}$ are defined by

$$\hat{\boldsymbol{\beta}}(\text{elastic net}) = \sqrt{(1+\lambda_2)}\hat{\boldsymbol{\beta}}^*. \tag{11}$$

Recall that $\hat{\boldsymbol{\beta}}(\text{naïve elastic net}) = \{1/\sqrt{(1+\lambda_2)}\}\hat{\boldsymbol{\beta}}^*$; thus

$$\hat{\boldsymbol{\beta}}(\text{elastic net}) = (1+\lambda_2)\,\hat{\boldsymbol{\beta}}(\text{naïve elastic net}). \tag{12}$$

Hence the elastic net coefficient is a rescaled naïve elastic net coefficient.

Such a scaling transformation preserves the variable selection property of the naïve elastic net and is the simplest way to undo shrinkage. Hence all the good properties of the naïve elastic

net that were described in Section 2 hold for the elastic net. Empirically we have found that the elastic net performs very well when compared with the lasso and ridge regression.

We have another justification for choosing $1+\lambda_2$ as the scaling factor. Consider the exact solution of the naïve elastic net when the predictors are orthogonal. The lasso is known to be minimax optimal (Donoho *et al.*, 1995) in this case, which implies that the naïve elastic net is not optimal. After scaling by $1+\lambda_2$, the elastic net automatically achieves minimax optimality.

A strong motivation for the $(1+\lambda_2)$-rescaling comes from a decomposition of the ridge operator. Since the predictors $\mathbf{X}$ are standardized, we have

$$\mathbf{X}^{\mathrm{T}}\mathbf{X} = \begin{pmatrix} 1 & \rho_{12} & . & \rho_{1p} \\ & 1 & . & . \\ & & 1 & \rho_{p-1,\,p} \\ & & & 1 \end{pmatrix}_{p\times p},$$

where $\rho_{i,j}$ is sample correlation. Ridge estimates with parameter λ_2 are given by $\hat{\boldsymbol{\beta}}(\text{ridge}) = \mathbf{R}\mathbf{y}$, with

$$\mathbf{R} = (\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda_2\mathbf{I})^{-1}\mathbf{X}^{\mathrm{T}}.$$

We can rewrite $\mathbf{R}$ as

$$\mathbf{R} = \frac{1}{1+\lambda_2}\mathbf{R}^* = \frac{1}{1+\lambda_2}\begin{pmatrix} 1 & \frac{\rho_{12}}{1+\lambda_2} & . & \frac{\rho_{1p}}{1+\lambda_2} \\ & 1 & . & . \\ & & 1 & \frac{\rho_{p-1,\,p}}{1+\lambda_2} \\ & & & 1 \end{pmatrix}^{-1}\mathbf{X}^{\mathrm{T}}. \tag{13}$$

$\mathbf{R}^*$ is like the usual OLS operator except that the correlations are shrunk by the factor $1/(1+\lambda_2)$, which we call decorrelation. Hence from equation (13) we can interpret the ridge operator as decorrelation followed by direct scaling shrinkage.

This decomposition suggests that the grouping effect of ridge regression is caused by the decorrelation step. When we combine the grouping effect of ridge regression with the lasso, the direct $1/(1+\lambda_2)$ shrinkage step is not needed and is removed by rescaling. Although ridge regression requires $1/(1+\lambda_2)$ shrinkage to control the estimation variance effectively, in our new method, we can rely on the lasso shrinkage to control the variance and to obtain sparsity.

From now on, let $\hat{\boldsymbol{\beta}}$ stand for $\hat{\boldsymbol{\beta}}$(elastic net). The next theorem gives another presentation of the elastic net, in which the decorrelation argument is more explicit.

Theorem 2. Given data $(\mathbf{y}, \mathbf{X})$ and (λ_1, λ_2), then the elastic net estimates $\hat{\boldsymbol{\beta}}$ are given by

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda_2\mathbf{I}}{1+\lambda_2}\right)\boldsymbol{\beta} - 2\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} + \lambda_1|\boldsymbol{\beta}|_1. \tag{14}$$

It is easy to see that

$$\hat{\boldsymbol{\beta}}(\text{lasso}) = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{X})\boldsymbol{\beta} - 2\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} + \lambda_1|\boldsymbol{\beta}|_1. \tag{15}$$

Hence theorem 2 interprets the elastic net as a stabilized version of the lasso. Note that $\hat{\Sigma} = \mathbf{X}^{\mathrm{T}}\mathbf{X}$ is a sample version of the correlation matrix Σ and

$$\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda_2\mathbf{I}}{1+\lambda_2} = (1-\gamma)\hat{\Sigma} + \gamma\mathbf{I}$$

with $\gamma = \lambda_2/(1+\lambda_2)$ shrinks $\hat{\Sigma}$ towards the identity matrix. Together equations (14) and (15) say that rescaling after the elastic net penalization is mathematically equivalent to replacing $\hat{\Sigma}$ with its shrunken version in the lasso. In linear discriminant analysis, the prediction accuracy can often be improved by replacing $\hat{\Sigma}$ by a shrunken estimate (Friedman, 1989; Hastie *et al.*, 2001). Likewise we improve the lasso by regularizing $\hat{\Sigma}$ in equation (15).

3.3. Connections with univariate soft thresholding

The lasso is a special case of the elastic net with $\lambda_2 = 0$. The other interesting special case of the elastic net emerges when $\lambda_2 \to \infty$. By theorem 2, $\hat{\boldsymbol{\beta}} \to \hat{\boldsymbol{\beta}}(\infty)$ as $\lambda_2 \to \infty$, where

$$\hat{\boldsymbol{\beta}}(\infty) = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta} - 2\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta} + \lambda_1 |\boldsymbol{\beta}|_1.$$

$\hat{\boldsymbol{\beta}}(\infty)$ has a simple closed form

$$\hat{\boldsymbol{\beta}}(\infty)_i = \left(|\mathbf{y}^{\mathrm{T}}\mathbf{x}_i| - \frac{\lambda_1}{2}\right)_+ \operatorname{sgn}(\mathbf{y}^{\mathrm{T}}\mathbf{x}_i), \qquad i = 1, 2, \ldots, p. \tag{16}$$

Observe that $\mathbf{y}^{\mathrm{T}}\mathbf{x}_i$ is the univariate regression coefficient of the ith predictor and $\hat{\boldsymbol{\beta}}(\infty)$ are the estimates by applying soft thresholding on univariate regression coefficients; thus equation (16) is called univariate soft thresholding (UST).

UST totally ignores the dependence between predictors and treats them as independent variables. Although this may be considered illegitimate, UST and its variants are used in other methods such as significance analysis of microarrays (Tusher *et al.*, 2001) and the nearest shrunken centroids classifier (Tibshirani *et al.*, 2002), and have shown good empirical performance. The elastic net naturally bridges the lasso and UST.

3.4. Computation: the algorithm LARS-EN

We propose an efficient algorithm called LARS-EN to solve the elastic net efficiently, which is based on the recently proposed algorithm LARS of Efron *et al.* (2004). They proved that, starting from zero, the lasso solution paths grow piecewise linearly in a predictable way. They proposed a new algorithm called LARS to solve the entire lasso solution path efficiently by using the same order of computations as a single OLS fit. By lemma 1, for each fixed λ_2 the elastic net problem is equivalent to a lasso problem on the augmented data set. So algorithm LARS can be directly used to create the *entire elastic net solution path* efficiently with the computational efforts of a single OLS fit. Note, however, that for $p \gg n$ the augmented data set has $p+n$ 'observations' and p variables, which can slow the computation considerably.

We further facilitate the computation by taking advantage of the sparse structure of $\mathbf{X}^*$, which is crucial in the $p \gg n$ case. In detail, as outlined in Efron *et al.* (2004), at the kth step we need to invert the matrix $\mathbf{G}_{A_k} = \mathbf{X}_{A_k}^{*\mathrm{T}}\mathbf{X}_{A_k}^*$, where A_k is the active variable set. This is done efficiently by updating or downdating the Cholesky factorization of $\mathbf{G}_{A_{k-1}}$ that is found at the previous step. Note that

$$\mathbf{G}_A = \frac{1}{1+\lambda_2}(\mathbf{X}_A^{\mathrm{T}}\mathbf{X}_A + \lambda_2\mathbf{I})$$

for any index set A, so it amounts to updating or downdating the Cholesky factorization of $\mathbf{X}_{A_{k-1}}^{\mathrm{T}}\mathbf{X}_{A_{k-1}} + \lambda_2\mathbf{I}$. It turns out that we can use a simple formula to update the Cholesky factorization of $\mathbf{X}_{A_{k-1}}^{\mathrm{T}}\mathbf{X}_{A_{k-1}} + \lambda_2\mathbf{I}$, which is very similar to the formula that is used for updating the Cholesky factorization of $\mathbf{X}_{A_{k-1}}^{\mathrm{T}}\mathbf{X}_{A_{k-1}}$ (Golub and Van Loan, 1983). The exact same

downdating function can be used for downdating the Cholesky factorization of $\mathbf{X}_{A_{k-1}}^{\mathrm{T}}\mathbf{X}_{A_{k-1}} + \lambda_2\mathbf{I}$. In addition, when calculating the equiangular vector and the inner products of the non-active predictors with the current residuals, we can save computations by using the simple fact that $\mathbf{X}_j^*$ has $p-1$ zero elements. In a word, we do not explicitly use $\mathbf{X}^*$ to compute all the quantities in algorithm LARS. It is also economical to record only the non-zero coefficients and the active variables set at each LARS-EN step.

Algorithm LARS-EN sequentially updates the elastic net fits. In the $p \gg n$ case, such as with microarray data, it is not necessary to run the algorithm to the end (early stopping). Real data and simulated computational experiments show that the optimal results are achieved at an early stage of algorithm LARS-EN. If we stop the algorithm after m steps, then it requires $O(m^3 + pm^2)$ operations.

3.5. Choice of tuning parameters

We now discuss how to choose the type and value of the tuning parameter in the elastic net. Although we defined the elastic net by using (λ_1, λ_2), it is not the only choice as the tuning parameter. In the lasso, the conventional tuning parameter is the L_1-norm of the coefficients (t) or the fraction of the L_1-norm (s). By the proportional relationship between $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\beta}}^*$, we can also use (λ_2, s) or (λ_2, t) to parameterize the elastic net. The advantage of using (λ_2, s) is that s is always valued within $[0, 1]$. In algorithm LARS the lasso is described as a forward stagewise additive fitting procedure and shown to be (almost) identical to ε-L_2 boosting (Efron *et al.*, 2004). This new view adopts the number of steps k of algorithm LARS as a tuning parameter for the lasso. For each fixed λ_2, the elastic net is solved by our algorithm LARS-EN; hence similarly we can use the number of the LARS-EN steps (k) as the second tuning parameter besides λ_2. The above three types of tuning parameter correspond to three ways to interpret the piecewise elastic net or lasso solution paths as shown in Fig. 3.

There are well-established methods for choosing such tuning parameters (Hastie *et al.* (2001), chapter 7). If only training data are available, tenfold cross-validation (CV) is a popular method for estimating the prediction error and comparing different models, and we use it here. Note that there are two tuning parameters in the elastic net, so we need to cross-validate on a two-dimensional surface. Typically we first pick a (relatively small) grid of values for λ_2, say $(0, 0.01, 0.1, 1, 10, 100)$. Then, for each λ_2, algorithm LARS-EN produces the entire solution path of the elastic net. The other tuning parameter (λ_1, s or k) is selected by tenfold CV. The chosen λ_2 is the one giving the smallest CV error.

For each λ_2, the computational cost of tenfold CV is the same as 10 OLS fits. Thus two-dimensional CV is computationally thrifty in the usual $n > p$ setting. In the $p \gg n$ case, the cost grows linearly with p and is still manageable. Practically, early stopping is used to ease the computational burden. For example, suppose that $n = 30$ and $p = 5000$; if we do not want more than 200 variables in the final model, we may stop algorithm LARS-EN after 500 steps and consider only the best k within 500.

From now on we drop the subscript of λ_2 if s or k is the other parameter.

4. Prostate cancer example

The data in this example come from a study of prostate cancer (Stamey *et al.*, 1989). The predictors are eight clinical measures: log(cancer volume) (lcavol), log(prostate weight) (lweight), age, the logarithm of the amount of benign prostatic hyperplasia (lbph), seminal vesicle invasion

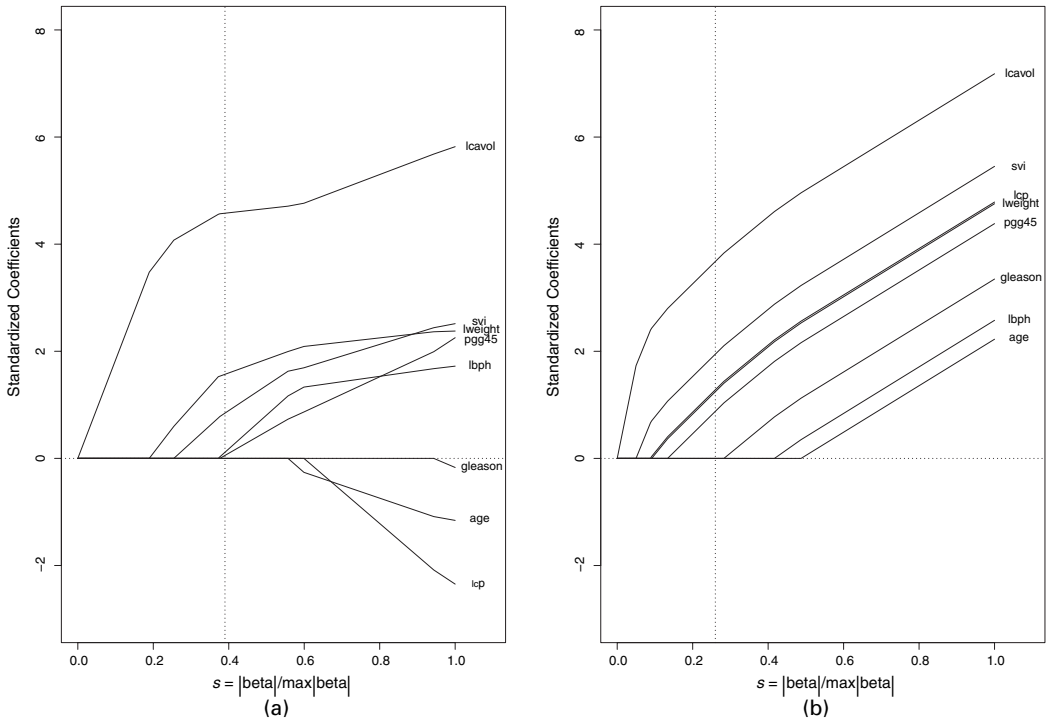

Fig. 3. (a) Lasso estimates as a function of s and (b) elastic net estimates ($\lambda = 1000$) as a function of s: both estimates are piecewise linear, which is a key property of our efficient algorithm; the solution paths also show that the elastic net is identical to univariate soft thresholding in this example (⋮, final model selected)

(svi), log(capsular penetration) (lcp), Gleason score (gleason) and percentage Gleason score 4 or 5 (pgg45). The response is the logarithm of prostate-specific antigen (lpsa).

OLS, ridge regression, the lasso, the naïve elastic net and the elastic net were all applied to these data. The prostate cancer data were divided into two parts: a training set with 67 observations and a test set with 30 observations. Model fitting and tuning parameter selection by tenfold CV were carried out on the training data. We then compared the performance of those methods by computing their prediction mean-squared error on the test data.

Table 1 clearly shows that the elastic net is the winner among all the competitors in terms of both prediction accuracy and sparsity. OLS is the worst method. The naïve elastic net performs identically to ridge regression in this example and fails to do variable selection. The lasso includes lcavol, lweight lbph, svi and pgg45 in the final model, whereas the elastic net selects

Table 1. Prostate cancer data: comparing different methods

Method	*Parameter(s)*	*Test mean-squared error*	*Variables selected*
OLS		0.586 (0.184)	All
Ridge regression	$\lambda = 1$	0.566 (0.188)	All
Lasso	$s = 0.39$	0.499 (0.161)	(1,2,4,5,8)
Naïve elastic net	$\lambda = 1$, $s = 1$	0.566 (0.188)	All
Elastic net	$\lambda = 1000$, $s = 0.26$	0.381 (0.105)	(1,2,5,6,8)

lcavol, lweight, svi, lcp and pgg45. The prediction error of the elastic net is about 24% lower than that of the lasso. We also see in this case that the elastic net is actually UST, because the λ selected is very big (1000). This can be considered as a piece of empirical evidence supporting UST. Fig. 3 displays the lasso and the elastic net solution paths.

If we check the correlation matrix of these eight predictors, we see that there are some medium correlations, although the highest is 0.76 (between pgg45 and gleason). We have seen that the elastic net dominates the lasso by a good margin. In other words, the lasso is hurt by the high correlation. We conjecture that, whenever ridge regression improves on OLS, the elastic net will improve the lasso. We demonstrate this point by simulations in the next section.

5. A simulation study

The purpose of this simulation is to show that the elastic net not only dominates the lasso in terms of prediction accuracy but also is a better variable selection procedure than the lasso. We simulate data from the true model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sigma\varepsilon, \qquad \varepsilon \sim N(0, 1).$$

Four examples are presented here. The first three examples were used in the original lasso paper (Tibshirani, 1996), to compare the prediction performance of the lasso and ridge regression systematically. The fourth example creates a grouped variable situation.

Within each example, our simulated data consist of a training set, an independent validation set and an independent test set. Models were fitted on training data only, and the validation data were used to select the tuning parameters. We computed the test error (the mean-squared error) on the test data set. We use the notation $\cdot/\cdot/\cdot$ to describe the number of observations in the training, validation and test set respectively, e.g. 20/20/200. Here are the details of the four scenarios.

(a) In example 1, we simulated 50 data sets consisting of 20/20/200 observations and eight predictors. We let $\boldsymbol{\beta} = (3, 1.5, 0, 0, 2, 0, 0, 0)$ and $\sigma = 3$. The pairwise correlation between $\mathbf{x}_i$ and $\mathbf{x}_j$ was set to be $\text{corr}(i, j) = 0.5^{|i-j|}$.

(b) Example 2 is the same as example 1, except that $\beta_j = 0.85$ for all j.

(c) In example 3, we simulated 50 data sets consisting of 100/100/400 observations and 40 predictors. We set

$$\boldsymbol{\beta} = (\underbrace{0, \ldots, 0}_{10}, \underbrace{2, \ldots, 2}_{10}, \underbrace{0, \ldots, 0}_{10}, \underbrace{2, \ldots, 2}_{10})$$

and $\sigma = 15$; $\text{corr}(i, j) = 0.5$ for all i and j.

(d) In example 4 we simulated 50 data sets consisting of 50/50/400 observations and 40 predictors. We chose

$$\boldsymbol{\beta} = (\underbrace{3, \ldots, 3}_{15}, \underbrace{0, \ldots, 0}_{25})$$

and $\sigma = 15$. The predictors $\mathbf{X}$ were generated as follows:

$$\begin{aligned}
\mathbf{x}_i &= Z_1 + \varepsilon_i^x, \quad Z_1 \sim N(0, 1), \quad i = 1, \ldots, 5, \\
\mathbf{x}_i &= Z_2 + \varepsilon_i^x, \quad Z_2 \sim N(0, 1), \quad i = 6, \ldots, 10, \\
\mathbf{x}_i &= Z_3 + \varepsilon_i^x, \quad Z_3 \sim N(0, 1), \quad i = 11, \ldots, 15, \\
\mathbf{x}_i &\sim N(0, 1), \quad \mathbf{x}_i \text{ independent identically distributed}, \quad i = 16, \ldots, 40,
\end{aligned}$$

Table 2. Median mean-squared errors for the simulated examples and four methods based on 50 replications†

Method	*Results for the following examples:*			
	Example 1	*Example 2*	*Example 3*	*Example 4*
Lasso	3.06 (0.31)	3.87 (0.38)	65.0 (2.82)	46.6 (3.96)
Elastic net	2.51 (0.29)	3.16 (0.27)	56.6 (1.75)	34.5 (1.64)
Ridge regression	4.49 (0.46)	2.84 (0.27)	39.5 (1.80)	64.5 (4.78)
Naïve elastic net	5.70 (0.41)	2.73 (0.23)	41.0 (2.13)	45.9 (3.72)

†The numbers in parentheses are the corresponding standard errors (of the medians) estimated by using the bootstrap with $B = 500$ resamplings on the 50 mean-squared errors.

where ε_i^x are independent identically distributed $N(0, 0.01)$, $i = 1, \ldots, 15$. In this model, we have three equally important groups, and within each group there are five members. There are also 25 pure noise features. An ideal method would select only the 15 true features and set the coefficients of the 25 noise features to 0.

Table 2 and Fig. 4 (box plots) summarize the prediction results. First we see that the naïve elastic net either has a very poor performance (in example 1) or behaves almost identically to either ridge regression (in examples 2 and 3) or the lasso (in example 4). In all the examples, the elastic net is significantly more accurate than the lasso, even when the lasso is doing much better than ridge regression. The reductions in the prediction error in examples 1, 2, 3 and 4 are 18%, 18%, 13% and 27% respectively. The simulation results indicate that the elastic net dominates the lasso under collinearity.

Table 3 shows that the elastic net produces sparse solutions. The elastic net tends to select more variables than the lasso does, owing to the grouping effect. In example 4 where grouped selection is required, the elastic net behaves like the 'oracle'. The additional 'grouped selection' ability makes the elastic net a better variable selection method than the lasso.

Here is an idealized example showing the important differences between the elastic net and the lasso. Let Z_1 and Z_2 be two independent $U(0, 20)$ variables. The response $\mathbf{y}$ is generated as $N(Z_1 + 0.1Z_2, 1)$. Suppose that we observe only

$$\mathbf{x}_1 = Z_1 + \varepsilon_1, \qquad \mathbf{x}_2 = -Z_1 + \varepsilon_2, \qquad \mathbf{x}_3 = Z_1 + \varepsilon_3,$$
$$\mathbf{x}_4 = Z_2 + \varepsilon_4, \qquad \mathbf{x}_5 = -Z_2 + \varepsilon_5, \qquad \mathbf{x}_6 = Z_2 + \varepsilon_6,$$

where ε_i are independent identically distributed $N(0, 1/16)$. 100 observations were generated from this model. The variables $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$ form a group whose underlying factor is Z_1, and $\mathbf{x}_4$, $\mathbf{x}_5$ and $\mathbf{x}_6$ form a second group whose underlying factor is Z_2. The within-group correlations are almost 1 and the between-group correlations are almost 0. An oracle would identify the Z_1-group as the important variates. Fig. 5 compares the solution paths of the lasso and the elastic net.

6. Microarray classification and gene selection

A typical microarray data set has thousands of genes and fewer than 100 samples. Because of the unique structure of the microarray data, we feel that a good classification method should have the following properties.

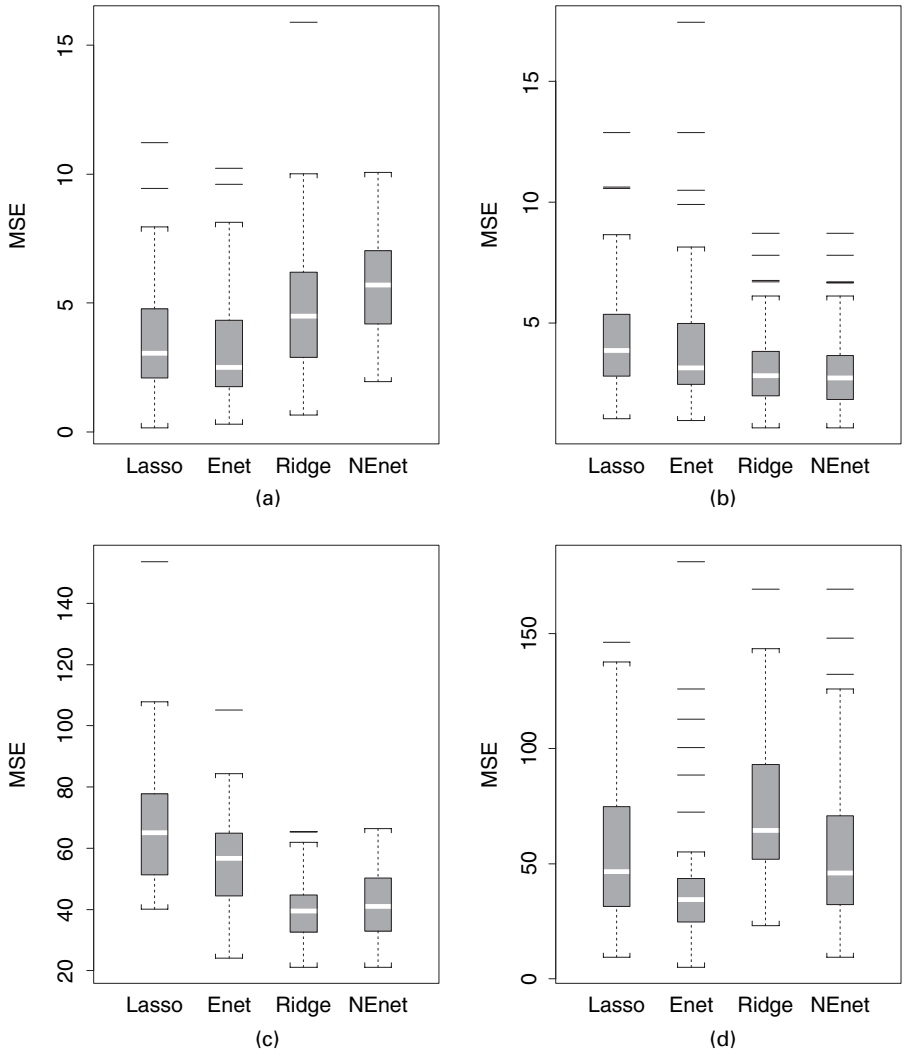

Fig. 4. Comparing the accuracy of prediction of the lasso, the elasic net (Enet), ridge regression and the naïve elastic net (NEnet) (the elastic net outperforms the lasso in all four examples): (a) example 1; (b) example 2; (c) example 3; (d) example 4

(a) Gene selection should be *built into* the procedure.
(b) It should not be limited by the fact that $p \gg n$.
(c) For those genes sharing the same biological pathway, it should be able to include whole groups into the model automatically once one gene among them is selected.

From published results in this domain, it appears that many classifiers achieve similar low classification error rates. But many of these methods do not select genes in a satisfactory way. Most of the popular classifiers fail with respect to at least one of the above properties. The lasso is good at (a) but fails both (b) and (c). The support vector machine (Guyon *et al.*, 2002) and penalized logistic regression (Zhu and Hastie, 2004) are very successful classifiers, but they cannot do gene selection automatically and both use either univariate ranking (Golub *et al.*,

Table 3. Median number of non-zero coefficients

Method	*Results for the following examples:*			
	Example 1	*Example 2*	*Example 3*	*Example 4*
Lasso	5	6	24	11
Elastic net	6	7	27	16

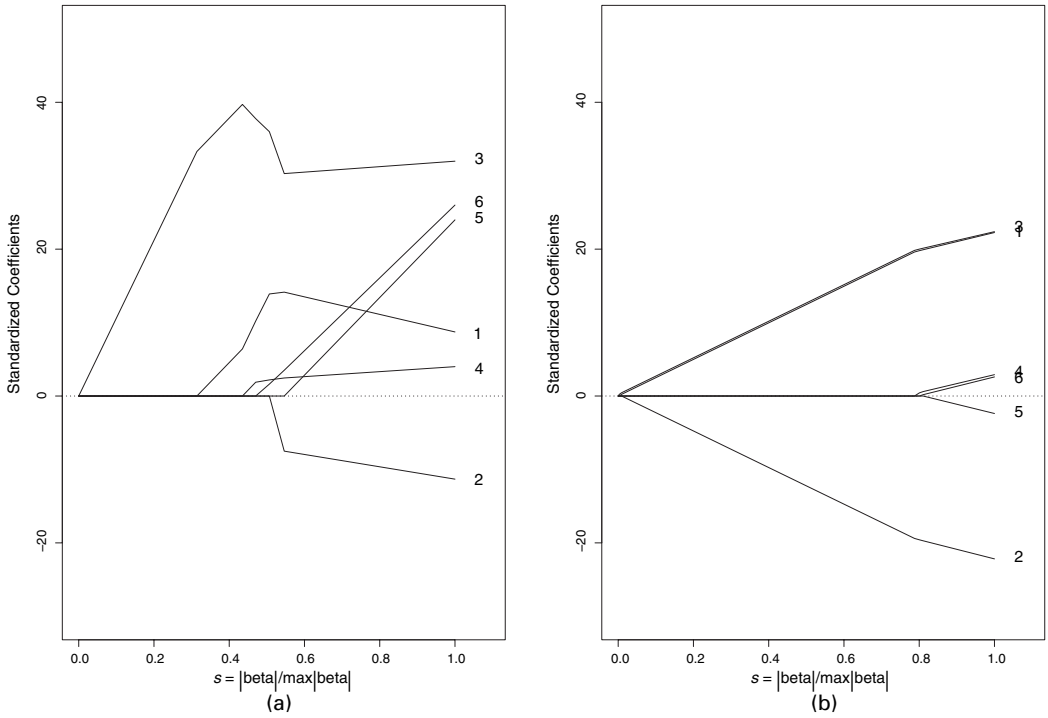

Fig. 5. (a) Lasso and (b) elastic net ($\lambda_2 = 0.5$) solution paths: the lasso paths are unstable and (a) does not reveal any correction information by itself; in contrast, the elastic net has much smoother solution paths, while clearly showing the 'grouped selection'—$\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ are in one 'significant' group and $\mathbf{x}_4$, $\mathbf{x}_5$ and $\mathbf{x}_6$ are in the other 'trivial' group; the decorrelation yields the grouping effect and stabilizes the lasso solution

1999) or recursive feature elimination (Guyon *et al.*, 2002) to reduce the number of genes in the final model.

As an automatic variable selection method, the elastic net naturally overcomes the difficulty of $p \gg n$ and has the ability to do grouped selection. We use the leukaemia data to illustrate the elastic net classifier.

The leukaemia data consist of 7129 genes and 72 samples (Golub *et al.*, 1999). In the training data set, there are 38 samples, among which 27 are type 1 leukaemia (acute lymphoblastic leukaemia) and 11 are type 2 leukaemia (acute myeloid leukaemia). The goal is to construct a diagnostic rule based on the expression level of those 7219 genes to predict the type of leukaemia. The remaining 34 samples are used to test the prediction accuracy of the diagnostic rule. To apply the elastic net, we first coded the type of leukaemia as a 0–1 response y. The classification

function is $I(\text{fitted value} > 0.5)$, where $I(\cdot)$ is the indicator function. We used tenfold CV to select the tuning parameters.

We used prescreening to make the computation more manageable. Each time that a model is fitted, we first select the 1000 most 'significant' genes as the predictors, according to their t-statistic scores (Tibshirani *et al.*, 2002). Note that this screening is done separately in each training fold in the CV. In practice, this screening does not affect the results, because we stop

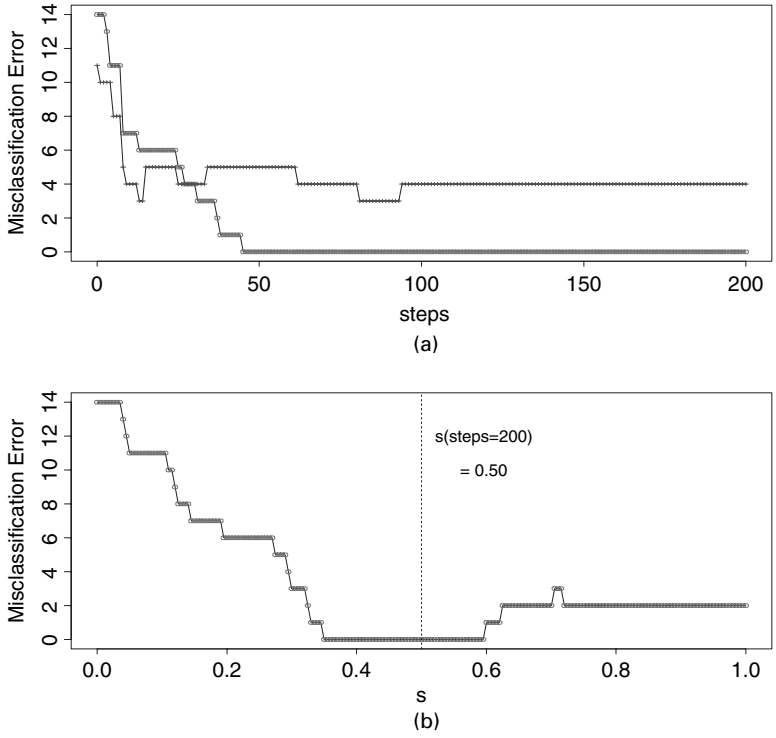

Fig. 6. Leukaemia classification and gene selection by the elastic net ($\lambda = 0.01$): (a) the early stopping strategy at 200 steps finds the optimal classifier with much less computational cost than (b) the whole elastic net paths; with early stopping, the number of steps is much more convenient than s, the fraction of L_1-norm, since computing s depends on the fit at the last step of algorithm LARS-EN; the actual values of s are not available in tenfold CV (+) if the algorithm is stopped early; on the training set, 200 steps are equivalent to $s = 0.50$ (⋮)

Table 4. Summary of the leukaemia classification results

Method	*Tenfold CV error*	*Test error*	*Number of genes*
Golub	3/38	4/34	50
Support vector maching–recursive feature elimination	2/38	1/34	31
Penalized logistic regression–recursive feature elimination	2/38	1/34	26
Nearest shrunken centroids	2/38	2/34	21
Elastic net	3/38	0/34	45

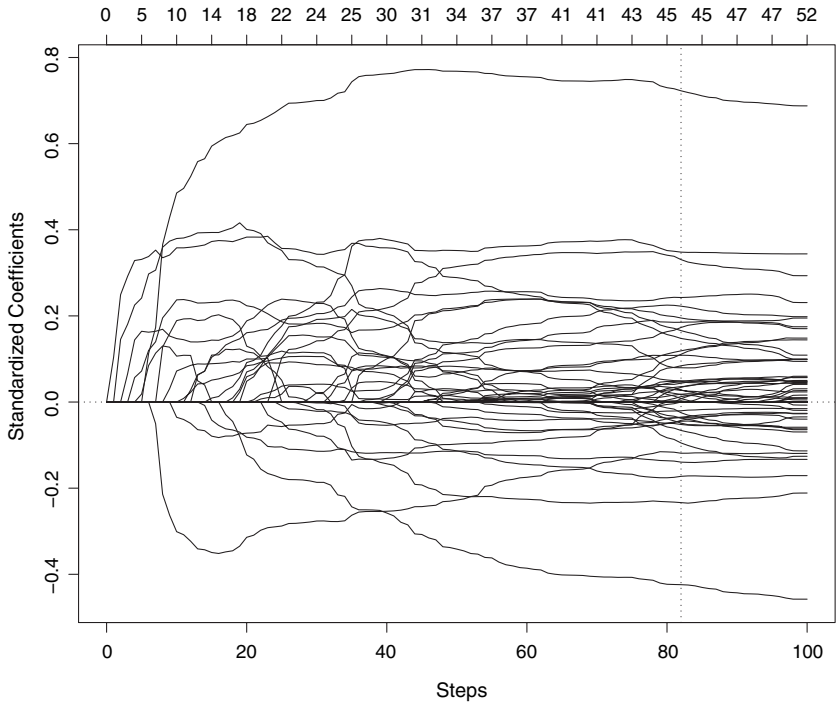

Fig. 7. Leukaemia data—elastic net coefficients paths (up to $k = 100$): the numbers on the top indicate the number of non-zero coefficients (selected genes) at each step; the optimal elastic net model is given by the fit at step 82 (⋮) with 45 selected genes; note that the size of the training set is 38, so the lasso can at most select 38 genes; in contrast, the elastic net selected more than 38 genes, not limited by the sample size; $\lambda = 0.01$ is chosen by tenfold CV; if a bigger λ is used, the grouping effect will be stronger

the elastic net path relatively early, at a stage when the screened variables are unlikely to be in the model.

All the prescreening, fitting and tuning were done using only the training set and the classification error is evaluated on the test data.

We stopped algorithm LARS-EN after 200 steps. As can be seen from Fig. 6, using the number of steps k in the algorithm as the tuning parameter, the elastic net classifier ($\lambda = 0.01$ and $k = 82$) gives a tenfold CV error of 3/38 and a test error of 0/34 with 45 genes selected. Fig. 7 displays the elastic net solution paths and the gene selection results. Table 4 compares the elastic net with several competitors including Golub's method, the support vector machine, penalized logistic regression and the nearest shrunken centroid (Tibshirani *et al.*, 2002). The elastic net gives the best classification, and it has an *internal* gene selection facility.

7. Discussion

We have proposed the elastic net, a novel shrinkage and selection method. The elastic net produces a sparse model with good prediction accuracy, while encouraging a grouping effect. The empirical results and simulations demonstrate the good performance of the elastic net and its superiority over the lasso. When used as a (two-class) classification method, the elastic net appears to perform well on microarray data in terms of the misclassification error, and it does automatic gene selection.

Although our methodology is motivated by regression problems, the elastic net penalty can be used in classification problems with any consistent (Zhang, 2004) loss functions, including the L_2-loss which we have considered here and binomial deviance. Some nice properties of the elastic net are better understood in the classification paradigm. For example, Fig. 6 is a familiar picture in boosting: the test error keeps decreasing and reaches a long flat region and then slightly increases (Hastie *et al.*, 2001). This is no coincidence. In fact we have discovered that the elastic net penalty has a close connection with the maximum margin explanation (Rosset *et al.*, 2004) to the success of the support vector machine and boosting. Thus Fig. 6 has a nice margin-based explanation. We have made some progress in using the elastic net penalty in classification, which will be reported in a future paper.

We view the elastic net as a generalization of the lasso, which has been shown to be a valuable tool for model fitting and feature extraction. Recently the lasso was used to explain the success of boosting: boosting performs a high dimensional lasso without explicitly using the lasso penalty (Hastie *et al.*, 2001; Friedman *et al.*, 2004). Our results offer other insights into the lasso, and ways to improve it.

Acknowledgements

We thank Rob Tibshirani and Ji Zhu for helpful comments, and an Associate Editor and referee for their useful comments and references. Trevor Hastie was partially supported by grant DMS-0204162 from the National Science Foundation and grant RO1-EB0011988-08 from the National Institutes of Health. Hui Zou was supported by grant DMS-0204162 from the National Science Foundation.

Appendix A: Proofs

A.1. Proof of lemma 2

A.1.1. Part (1)

Fix $\lambda > 0$. If $\hat{\beta}_i \neq \hat{\beta}_j$, let us consider $\hat{\boldsymbol{\beta}}^*$ as follows:

$$\hat{\beta}_k^* = \begin{cases} \hat{\beta}_k & \text{if } k \neq i \text{ and } k \neq j, \\ \frac{1}{2}(\hat{\beta}_i + \hat{\beta}_j) & \text{if } k = i \text{ or } k = j. \end{cases}$$

Because $\mathbf{x}_i = \mathbf{x}_j$, it is obvious that $\mathbf{X}\hat{\boldsymbol{\beta}}^* = \mathbf{X}\hat{\boldsymbol{\beta}}$; thus $|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^*|^2 = |\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}|^2$. However, $J(\cdot)$ is strictly convex, so we have $J(\hat{\boldsymbol{\beta}}^*) < J(\hat{\boldsymbol{\beta}})$. Therefore $\hat{\boldsymbol{\beta}}$ cannot be the minimizer of equation (7), which is a contradiction. So we must have $\hat{\beta}_i = \hat{\beta}_j$.

A.1.2. Part (2)

If $\hat{\beta}_i \hat{\beta}_j < 0$, consider the same $\hat{\boldsymbol{\beta}}^*$ again. We see that $|\hat{\boldsymbol{\beta}}^*| < |\hat{\boldsymbol{\beta}}|$, so $\hat{\boldsymbol{\beta}}$ cannot be a lasso solution. The rest can be directly verified by the definition of the lasso, which is thus omitted.

A.2. Proof of theorem 1

If $\hat{\beta}_i(\lambda_1, \lambda_2)\ \hat{\beta}_j(\lambda_1, \lambda_2) > 0$, then both $\hat{\beta}_i(\lambda_1, \lambda_2)$ and $\hat{\beta}_j(\lambda_1, \lambda_2)$ are non-zero, and we have $\operatorname{sgn}\{\hat{\beta}_i(\lambda_1, \lambda_2)\} = \operatorname{sgn}\{\hat{\beta}_j(\lambda_1, \lambda_2)\}$. Because of equation (4), $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ satisfies

$$\left.\frac{\partial L(\lambda_1, \lambda_2, \boldsymbol{\beta})}{\partial \beta_k}\right|_{\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)} = 0 \qquad \text{if } \hat{\beta}_k(\lambda_1, \lambda_2) \neq 0. \tag{17}$$

Hence we have

$$-2\mathbf{x}_i^{\mathrm{T}}\{\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\} + \lambda_1 \operatorname{sgn}\{\hat{\beta}_i(\lambda_1, \lambda_2)\} + 2\lambda_2\, \hat{\beta}_i(\lambda_1, \lambda_2) = 0, \tag{18}$$

$$-2\mathbf{x}_j^{\mathrm{T}}\{\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1,\lambda_2)\}+\lambda_1\,\mathrm{sgn}\{\hat{\beta}_j(\lambda_1,\lambda_2)\}+2\lambda_2\,\hat{\beta}_j(\lambda_1,\lambda_2)=0. \tag{19}$$

Subtracting equation (18) from equation (19) gives

$$(\mathbf{x}_j^{\mathrm{T}}-\mathbf{x}_i^{\mathrm{T}})\{\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1,\lambda_2)\}+\lambda_2\{\hat{\beta}_i(\lambda_1,\lambda_2)-\hat{\beta}_j(\lambda_1,\lambda_2)\}=0,$$

which is equivalent to

$$\hat{\beta}_i(\lambda_1,\lambda_2)-\hat{\beta}_j(\lambda_1,\lambda_2)=\frac{1}{\lambda_2}(\mathbf{x}_i^{\mathrm{T}}-\mathbf{x}_j^{\mathrm{T}})\,\hat{r}(\lambda_1,\lambda_2), \tag{20}$$

where $\hat{r}(\lambda_1,\lambda_2)=\mathbf{y}-\mathbf{x}\hat{\boldsymbol{\beta}}(\lambda_1,\lambda_2)$ is the residual vector. Since $\mathbf{X}$ are standardized, $|\mathbf{x}_i-\mathbf{x}_j|^2=2(1-\rho)$ where $\rho=\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$. By equation (4) we must have

$$L\{\lambda_1,\lambda_2,\hat{\boldsymbol{\beta}}(\lambda_1,\lambda_2)\}\leqslant L(\lambda_1,\lambda_2,\boldsymbol{\beta}=0),$$

i.e.

$$|\hat{r}(\lambda_1,\lambda_2)|^2+\lambda_2|\hat{\beta}(\lambda_1,\lambda_2)|^2+\lambda_1|\hat{\boldsymbol{\beta}}(\lambda_1,\lambda_2)|_1\leqslant|\mathbf{y}|^2.$$

So $|\hat{r}(\lambda_1,\lambda_2)|\leqslant|\mathbf{y}|$. Then equation (20) implies that

$$D_{\lambda_1,\lambda_2}(i,j)\leqslant\frac{1}{\lambda_2}\frac{|\hat{\mathbf{r}}(\lambda_1,\lambda_2)|}{|\mathbf{y}|}|\mathbf{x}_i-\mathbf{x}_j|\leqslant\frac{1}{\lambda_2}\sqrt{\{2(1-\rho)\}}.$$

A.3. Proof of theorem 2

Let $\hat{\boldsymbol{\beta}}$ be the elastic net estimates. By definition and equation (10) we have

$$\begin{aligned}\hat{\boldsymbol{\beta}}&=\arg\min_{\boldsymbol{\beta}}\left|\mathbf{y}^*-\mathbf{X}^*\frac{\boldsymbol{\beta}}{\sqrt{(1+\lambda_2)}}\right|^2+\frac{\lambda_1}{\sqrt{(1+\lambda_2)}}\left|\frac{\boldsymbol{\beta}}{\sqrt{(1+\lambda_2)}}\right|_1\\&=\arg\min_{\boldsymbol{\beta}}\boldsymbol{\beta}^{\mathrm{T}}\left(\frac{\mathbf{X}^{*\mathrm{T}}\mathbf{X}^*}{1+\lambda_2}\right)\boldsymbol{\beta}-2\frac{\mathbf{y}^{*\mathrm{T}}X^*}{\sqrt{(1+\lambda_2)}}+\mathbf{y}^{*\mathrm{T}}\mathbf{y}^*+\frac{\lambda_1|\beta|_1}{1+\lambda_2}.\end{aligned} \tag{21}$$

Substituting the identities

$$\mathbf{X}^{*\mathrm{T}}\mathbf{X}^*=\left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda_2}{1+\lambda_2}\right),$$
$$\mathbf{y}^{*\mathrm{T}}\mathbf{X}^*=\frac{\mathbf{y}^{\mathrm{T}}\mathbf{X}}{\sqrt{(1+\lambda_2)}},$$
$$\mathbf{y}^{*\mathrm{T}}\mathbf{y}^*=\mathbf{y}^{\mathrm{T}}\mathbf{y}$$

into equation (21), we have

$$\begin{aligned}\hat{\boldsymbol{\beta}}&=\arg\min_{\boldsymbol{\beta}}\frac{1}{1+\lambda_2}\left\{\boldsymbol{\beta}^{\mathrm{T}}\left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda_2\mathbf{I}}{1+\lambda_2}\right)\boldsymbol{\beta}-2\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta}+\lambda_1|\boldsymbol{\beta}|_1\right\}+\mathbf{y}^{\mathrm{T}}\mathbf{y}\\&=\arg\min_{\boldsymbol{\beta}}\boldsymbol{\beta}^{\mathrm{T}}\left(\frac{\mathbf{X}^{\mathrm{T}}\mathbf{X}+\lambda_2\mathbf{I}}{1+\lambda_2}\right)\boldsymbol{\beta}-2\mathbf{y}^{\mathrm{T}}\mathbf{X}\boldsymbol{\beta}+\lambda_1|\boldsymbol{\beta}|_1.\end{aligned}$$

References

Breiman, L. (1996) Heuristics of instability and stabilization in model selection. *Ann. Statist.*, **24**, 2350–2383.

Dettling, M. and Bühlmann, P. (2004) Finding predictive gene groups from microarray data. *J. Multiv. Anal.*, **90**, 106–131.

Díaz-Uriarte, R. (2003) A simple method for finding molecular signatures from gene expression data. *Technical Report*. Spanish National Cancer Center. (Available from `http://www.arxiv.org/abs/q-bio.QM/0401043`.)

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G. and Picard, D. (1995) Wavelet shrinkage: asymptopia (with discussion)? *J. R. Statist. Soc.* B, **57**, 301–369.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004) Least angle regression. *Ann. Statist.*, **32**, 407–499.

Fan, J. and Li, R. (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Statist. Ass.*, **96**, 1348–1360.
Frank, I. and Friedman, J. (1993) A statistical view of some chemometrics regression tools. *Technometrics*, **35**, 109–148.
Friedman, J. (1989) Regularized discriminant analysis. *J. Am. Statist. Ass.*, **84**, 249–266.
Friedman, J., Hastie, T., Rosset, S., Tibshirani, R. and Zhu, J. (2004) Discussion of boosting papers. *Ann. Statist.*, **32**, 102–107.
Fu, W. (1998) Penalized regression: the bridge versus the lasso. *J. Computnl Graph. Statist.*, **7**, 397–416.
Golub, G. and Van Loan, C. (1983) *Matrix Computations*. Baltimore: Johns Hopkins University Press.
Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J. and Caligiuri, M. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 513–536.
Guyon, I., Weston, J., Barnhill, S. and Vapnik, V. (2002) Gene selection for cancer classification using support vector machines. *Mach. Learn.*, **46**, 389–422.
Hastie, T., Tibshirani, R., Botstein, D. and Brown, P. (2003) Supervised harvesting of expression trees. *Genome Biol.*, **2**, 0003.1–0003.12.
Hastie, T., Tibshirani, R., Eisen, M., Brown, P., Ross, D., Scherf, U., Weinstein, J., Alizadeh, A., Staudt, L. and Botstein, D. (2000) 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biol.*, **1**, 1–21.
Hastie, T., Tibshirani, R. and Friedman, J. (2001) *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. New York: Springer.
Hoerl, A. and Kennard, R. (1988) Ridge regression. In *Encyclopedia of Statistical Sciences*, vol. 8, pp. 129–136. New York: Wiley.
Rosset, S., Zhu, J. and Hastie, T. (2004) Boosting as a regularized path to a maximum margin classifier. *J. Mach. Learn. Res.*, **5**, 941–973.
Segal, M., Dahlquist, K. and Conklin, B. (2003) Regression approach for microarray data analysis. *J. Computnl Biol.*, **10**, 961–980.
Stamey, T., Kabalin, J., McNeal, J., Johnstone, I., Freiha, F., Redwine, E. and Yang, N. (1989) Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate ii: radical prostatectomy treated patients. *J. Urol.*, **16**, 1076–1083.
Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.* B, **58**, 267–288.
Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, C. (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natn. Acad. Sci. USA*, **99**, 6567–6572.
Tusher, V., Tibshirani, R. and Chu, C. (2001) Significance analysis of microarrays applied to transcriptional responses to ionizing radiation. *Proc. Natn. Acad. Sci. USA*, **98**, 5116–5121.
West, M., Blanchettem, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Marks, J. and Nevins, J. (2001) Predicting the clinical status of human breast cancer using gene expression profiles. *Proc. Natn. Acad. Sci. USA*, **98**, 11462–11467.
Zhang, T. (2004) Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statist.*, **32**, 469–475.
Zhu, J. and Hastie, T. (2004) Classification of gene microarrays by penalized logistic regression. *Biostatistics*, **5**, 427–444.