Regression Shrinkage and Selection via the Elastic Net, with Applications to Microarrays

Hui Zou and Trevor Hastie *
Department of Statistics, Stanford University

December 5, 2003

Abstract

We propose the elastic net, a new regression shrinkage and selection method. Real data and a simulation study show that the elastic net often outperforms the lasso, while it enjoys a similar sparsity of representation. In addition, the elastic net encourages a grouping effect, where strong correlated predictors are kept in the model. The elastic net is particularly useful in the analysis of microarray data in which the number of genes (predictors) is much bigger than the number of samples (observations). We show how the elastic net can be used to construct a classification rule and do automatic gene selection at the same time in microarray data, where the lasso is not very satisfied. We also propose an efficient algorithm for solving the elastic net based on the recently invented LARS algorithm.

keywords: Gene selection; Grouping effect; Lasso; LARS algorithm; Microarray classification.

1 Introduction

We consider the usual linear regression model: given p predictors $\mathbf{X}_1, \cdots, \mathbf{X}_p$, we predict the response Y by a linear model

$$\hat{Y} = \hat{\beta}_0 + \mathbf{X}_1\hat{\beta}_1 + \cdots + \mathbf{X}_p\hat{\beta}_p. \quad (1)$$

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

Given a data set, a model-fitting procedure gives the vector of coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \cdots, \hat{\beta}_p)$. For example, the ordinary least squares (OLS) estimates are obtained by minimizing the residual sum squares (RSS). Other variants such as ridge regression and the lasso often improve on OLS. The criteria for evaluating the quality of a model will differ according to the circumstances. Typically the following two aspects are important

- Accuracy of prediction on future data: it is hard to defend a model that predicts poorly.
- Interpretation of the model: scientists prefer a simpler model because it puts more light on the relations between 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. Many techniques have been proposed to improve OLS, such as ridge regression, best subset selection and the lasso. Ridge regression minimizes RSS subject to a bound on the L_2 norm of the coefficients. By doing so, ridge regression shrinks the coefficients continuously toward zero. It achieves its better prediction performance through a bias-variance trade-off. However, ridge regression always keeps all the predictors in the model, so it cannot produce a parsimonious model. Best subset selection minimizes the RSS subject to the number of non-zero coefficients equals some $k, k \leq p$. Obviously best subset selection produces a sparse model, however it is extremely variable because of its inherent discreteness.

Tibshirani (1996) proposed a promising method called the lasso. The lasso minimizes RSS subject to a bound on the L_1 norm of the coefficients. Due to the nature of the L_1 penalty, the lasso does both continuous shrinkage and automatic variable selection at the same time. So the lasso possess some of the good features of both ridge regression and best subset selection. Although the lasso has shown success in many situations, it has some limitations. Consider the following three scenarios:

1. If the number of predictors p is bigger than the number of observations n, the lasso is not well-defined unless the bound on the L_1 norm of the coefficients is smaller than a certain value. Moreover, it at most selects n variables because of the nature of the convex optimization problem. This seems to be a limiting feature for a regularization method.

2. 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.

3. For usual $n > p$ situations, if there exist high correlations among predictors, the prediction performance of the lasso is dominated by ridge regression by a good margin.

Scenarios (1) and (2) make the lasso an inappropriate variable selection method in some situations. We illustrate our points by considering the regression problem in microarray gene expression data. A typical microarray data set has thousands of predictors (genes) and less than 100 samples. For those genes sharing the same biological "pathway", the correlations among them can be high. We think of those genes as forming a group. The ideal gene selection method shall be able to do two things: eliminate the trivial genes, and automatically include whole groups into the model once one gene amongst them is selected. We call this "grouped selection". Because we are dealing with the $p \gg n$ case and grouped variables, the lasso is not the ideal method. As for prediction performance, scenario (3) is not rare in regression problems. So there is quite a lot of room for improving the prediction power of the lasso.

Our goal is to find a model-fitting procedure that works as well as the lasso whenever the lasso does the best, and can fix the problems highlighted above, i.e., it should mimic the ideal gene selection method in scenarios (1) and (2), especially with microarray data, and it should have a better prediction performance than the lasso in scenario (3).

In this paper we propose a new regression method which we call the *elastic net*. Similar to the lasso, the elastic net simultaneously does automatic variable selection and continuous shrinkage. Meanwhile the elastic net has the "grouped selection" ability. It is like a stretchable fishing net that retains "all" the large "fish". Simulation studies and real data examples show that the elastic net significantly improves on the lasso in terms of prediction accuracy. The elastic net is particularly useful for microarray data analysis, where the lasso is not very satisfactory. We use the elastic net to do microarray classification and automatic gene selection, and obtain good results.

The paper is organized as the follows. Section 2 introduces the elastic net penalty and the naive elastic net which is a penalized least squares method using the elastic net penalty. We give some results on the grouping effect of the naive elastic net, and show it is equivalent to a lasso type problem.

In Section 3, we discuss the strength and drawbacks of the naive elastic net, which leads us to the elastic net. We argue that the elastic net keeps the strength of the naive elastic and eliminates its deficiency, hence the elastic net is the desired method to achieve our goal. We also address the computation issues and show how to select the tuning parameters of the elastic net. As demonstrations, prostate cancer data is used to illustrate our methodology in Section 4. Simulation results are presented in Section 5. Section 6 shows the application of the elastic net to classification and gene selection in microarray gene-expression data. Leukemia data is used as an illustration example. Proofs of the theorems are given in the appendix.

2 Naive Elastic Net

2.1 Definition

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

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

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

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

where

$$|\boldsymbol{\beta}|^2 = \sum_{j=1}^{p} \beta_j^2 \text{ and } |\boldsymbol{\beta}|_1 = \sum_{j=1}^{p} |\beta_j|.$$

Then the naive elastic net estimator $\hat{\boldsymbol{\beta}}$ is chosen by minimizing the naive elastic net criterion,

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

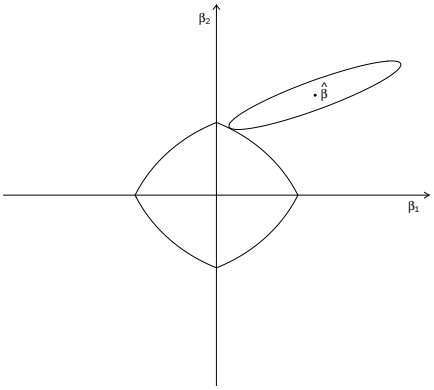

Figure 1: *The curve around origin is the contour plot of the elastic net penalty with* $p = 2, \alpha = 1$ *and* $t = 0.5$. *We see singularities at the vertexes and the edges are strictly convex. The strength of convexity varies with* α. *The oval shape curve shows the contour of* L_2 *loss with* $\hat{\beta}^{ols}$ *as its center.*

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

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

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. $\forall \alpha \in (0,1)$, the elastic net penalty function is singular (without first derivative) at 0 and it is strictly convex. Figure 1 is an example of the contour plot of the elastic net penalty in 2-dimensional space.

It is known that for ridge regression, if the predictors are highly correlated then the corresponding coefficients tend to be equal (see Theorem 1 in Section 2.3). This gives us some flavor of the grouping effect even though ridge regression does not select variables. The lasso on the other hand can do variable selection. These characteristics are determined by their penalty functions. So the intuition behind the naive elastic net is that by mixing the ridge penalty and the lasso penalty, we hope to combine the strengths of the

lasso and ridge regression so that the new method can achieve our goals.

2.2 Solutions

Here we present a method to solve the naive elastic net problem. The basic idea is to reduce the naive elastic net problem to an equivalent lasso problem.

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

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

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

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

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 naive elastic net problem to an equivalent lasso problem on an artificial data sets. Note that in the equivalent artificial data sets, the sample size is $n+p$ and $\mathbf{X}^*$ has rank p, which means the naive elastic net can potentially select all p predictors in all situations. This important property overcomes the limitation of the lasso described in scenario (1).

In the case of an orthogonal design, the exact solution of ridge regression with parameter λ_2 is given by $\hat{\boldsymbol{\beta}}(\text{ridge}) = \frac{\hat{\boldsymbol{\beta}}^{ols}}{1+\lambda_2}$, where $\hat{\boldsymbol{\beta}}^{ols} = \mathbf{X}^T Y$, while the lasso solution is given by the soft-thresholding (Donoho et al. 1995), $\hat{\beta}_i(\text{lasso}) = \left(\left|\hat{\beta}_i^{ols}\right| - \frac{\lambda_1}{2}\right)_+ \text{sgn}\left(\hat{\beta}_i^{ols}\right)$, where "+" denotes the positive part. It is straightforward to show that with parameters (λ_1, λ_2), the naive elastic net solution is

$$\hat{\beta}_i(\text{naive elastic net}) = \frac{\left(\left|\hat{\beta}_i^{ols}\right| - \frac{\lambda_1}{2}\right)_+}{1+\lambda_2} \text{sgn}\left(\hat{\beta}_i^{ols}\right). \tag{6}$$

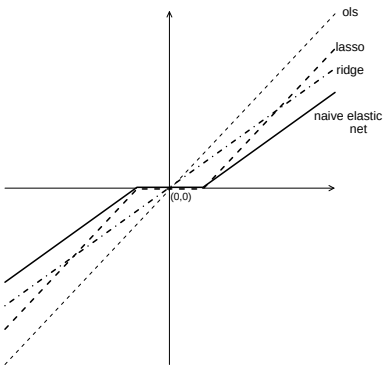

Figure 2: $\lambda_1 = 2, \lambda_2 = 1$. *Exact solutions for the lasso, ridge and the naive elastic net in an orthogonal design.*

(6) explains clearly how the naive elastic net works in the orthogonal design case: it first apples soft-thresholding on the OLS coefficients, then directly shrinks them by factor $1 + \lambda_2$ (see Figure 2).

2.3 Results on the grouping effect

In Section 2.1 we argued heuristically that the naive elastic net would tend to select variables in groups. Now we try to give some mathematical support to our argument.

Theorem 1 *Given data* $(Y, \mathbf{X})$ *and parameters* (λ_1, λ_2)*, the response* Y *is centered and the predictors* $\mathbf{X}$ *are standardized. Let* $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ *be the naive elastic net estimates. Define* $D_{\lambda_1,\lambda_2}(i,j) = \frac{1}{|Y|}\left|\left|\hat{\beta}_i(\lambda_1,\lambda_2)\right| - \left|\hat{\beta}_j(\lambda_1,\lambda_2)\right|\right|$.

1. *If* $\mathbf{X}_i = \mathbf{X}_j$*, then given any fixed* $\lambda_2 > 0$*,* $\hat{\beta}_i(\lambda_1,\lambda_2) = \hat{\beta}_j(\lambda_1,\lambda_2)$ *for all* $\lambda_1 \geq 0$*, i.e., the whole solution paths of* $\hat{\beta}_i$ *and* $\hat{\boldsymbol{\beta}}_j$ *are the same.*

2. $\forall \lambda_2 > 0$*, suppose* $\hat{\boldsymbol{\beta}}_i(\lambda_1,\lambda_2)\,\hat{\boldsymbol{\beta}}_j(\lambda_1,\lambda_2) > 0$*, then* $D_{\lambda_1,\lambda_2}(i,j) \leq \frac{1}{\lambda_2}\sqrt{2(1-\rho)}$*, where* $\rho = cor(\mathbf{X}_i, \mathbf{X}_j)$.

Remark 1: Conclusion (1) is a well-known result in ridge regression. From the proof (given in Appendix) we can see that this conclusion is always true for any loss function as long as the penalty function is strict convex.

Remark 2: The unit-less 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 $-\mathbf{X}_j$ instead), conclusion (2) says the difference between the coefficient paths of predictor i and predictor j is almost 0, which in certain aspects quantifies the grouping effect of the naive elastic net.

3 Elastic Net

3.1 Deficiency of the naive elastic net

The naive elastic net looks very promising so far. Due to the nature of the elastic net penalty, it seems to posses the good properties of both ridge regression and the lasso. However, real data examples and simulations (the results are given in Section 4 and Section 5) show that the naive elastic net does not perform satisfactorily when compared to ridge regression and the lasso. It behaves either like ridge regression, keeping all the variables in the model or very similar to the lasso, thus it fails to truly combine the good properties of ridge and the lasso. The naive 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 (which also depend on λ_2). The naive elastic net inherits the grouping effect from ridge, and we hope that it also enjoys the sparsity of lasso. But it incurs a double amount of shrinkage, since ridge regression also shrinks the coefficients at the beginning. Double shrinkage does not help to reduce the variances much and introduces unnecessary extra bias. When we select the tuning parameters according to MSE by some external methods such as cross-validation, we tend to end up with a result either close to ridge or close to the lasso. Therefore the double shrinkage is really where the problem lies.

3.2 The elastic net estimates

We outline a remedy for the poor performance of the naive elastic net. Let us follow the notation of Section 2.2. Given data $(Y, \mathbf{X})$ and penalty parameter (λ_1, λ_2), after introducing the equivalent artificial data set $(Y^*, \mathbf{X}^*)$, we solve a lasso type problem

$$\hat{\boldsymbol{\beta}}^* = \arg\min_{\boldsymbol{\beta}^*} |Y^* - \mathbf{X}^*\boldsymbol{\beta}^*|^2 + \frac{\lambda_1}{\sqrt{1+\lambda_2}} |\boldsymbol{\beta}^*|_1 . \tag{7}$$

The elastic net estimates $\hat{\boldsymbol{\beta}}$ are defined as

$$\hat{\boldsymbol{\beta}}\,(\text{elastic net}) = \sqrt{1+\lambda_2}\hat{\boldsymbol{\beta}}^* . \tag{8}$$

Recall that $\hat{\boldsymbol{\beta}}(\text{naive elastic net}) = \frac{1}{\sqrt{1+\lambda_2}}\hat{\boldsymbol{\beta}}^*$, thus

$$\hat{\boldsymbol{\beta}}(\text{elastic net}) = (1+\lambda_2)\hat{\boldsymbol{\beta}}(\text{naive elastic net}). \tag{9}$$

Hence the elastic net estimate is nothing but a rescaled naive elastic net estimate.

Why choose $1+\lambda_2$ as the scaling factor? One obvious motivation is from the exact solution of the naive elastic net with an orthogonal design. After scaling correction by $1+\lambda_2$ the elastic net estimate is identical to the lasso estimate, which is known to be minimax optimal (Donoho et al. 1995). Another motivation for the scaling correction comes from a decomposition of the ridge operator. Since $\mathbf{X}$ are standardized beforehand, we have

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

Then for ridge regression with parameter λ_2,

$$\hat{\beta}(\text{ridge}) = \mathbf{R}Y,$$

where $\mathbf{R}$ is the ridge operator

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

We can rewrite $\mathbf{R}$ as

$$\mathbf{R} = \frac{1}{1+\lambda_2}\mathbf{R}^* = \frac{1}{1+\lambda_2}\begin{bmatrix} 1 & \frac{\rho_{12}}{(1+\lambda_2)} & \cdot & \frac{\rho_{1p}}{(1+\lambda_2)} \\ & 1 & \cdot & \cdot \\ & & 1 & \frac{\rho_{p-1,p}}{(1+\lambda_2)} \\ & & & 1 \end{bmatrix}^{-1} \mathbf{X}^T. \tag{10}$$

$\mathbf{R}^*$ is like the usual OLS operator except the correlations are shrunk by factor $\frac{1}{1+\lambda_2}$, which we call de-correlation. (10) is regarded as the decomposition of ridge operator: ridge regression is equivalent to de-correlation followed by direct scaling shrinkage. So the intuition of the scaling correction is that if we re-scale the naive elastic net estimator by factor $1+\lambda_2$, we may eliminate the ridge shrinkage effect while still keep the de-correlation part of ridge regression, which is responsible for the grouping effect, then we can rely on the lasso shrinkage to achieve good prediction performance and sparsity. In the later sections, real data and simulations confirm these arguments. Hence the elastic net is the desired method to achieve our goals.

3.3 Connection with univariate soft-thresholding

Let $\hat{\beta}$ stand for $\hat{\beta}$ (elastic net). The next theorem gives another presentation of the elastic net, in which the de-correlation argument is more explicit.

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

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \beta^T \left(\frac{\mathbf{X}^T\mathbf{X} + \lambda_2\mathbf{I}}{1+\lambda_2} \right) \beta - 2Y^T\mathbf{X}\boldsymbol{\beta} + \lambda_1 |\boldsymbol{\beta}|_1 . \tag{11}$$

It is easy to see that

$$\hat{\boldsymbol{\beta}}(\text{lasso}) = \arg\min_{\boldsymbol{\beta}} \beta^T \left(\mathbf{X}^T\mathbf{X} \right) \boldsymbol{\beta} - 2Y^T\mathbf{X}\boldsymbol{\beta} + \lambda_1 |\boldsymbol{\beta}|_1 .$$

So the de-correlation distinguishes the elastic net from the lasso.

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

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

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

$$\hat{\beta}(\infty)_i = \left(\left|Y^T\mathbf{X}_i\right| - \frac{\lambda_1}{2}\right)_+ \operatorname{sgn}\left(Y^T\mathbf{X}_i\right), \quad i = 1, 2, \ldots, p. \tag{12}$$

Observe that $Y^T\mathbf{X}_i$ is the univariate regression coefficient of the i-th predictor, hence $\hat{\boldsymbol{\beta}}(\infty)$ is the estimator by applying soft-thresholding on univariate regression coefficients. We call $\hat{\boldsymbol{\beta}}(\infty)$ the univariate soft-thresholding (UST) estimates.

UST totally ignores the dependence among predictors, and treats them as independent variables. Although this may be considered illegitimate, UST and its variants are used in other methods such as SAM (Tusher et al. 2001) and nearest shrunken centroids (NSC) classifier (Tibshirani et al. 2002) and have successful empirical performances. The elastic net naturally bridges the lasso and UST, which may help to shed some light on UST and its variants.

3.4 Computations: the LARS-EN algorithm

We propose an efficient algorithm to efficiently solve the entire elastic net solution paths for any fixed λ_2. Our algorithm is based on the recently proposed LARS algorithm of Efron et al. (2004) (referred to as the LARS paper hereafter), thus we call it LARS-EN algorithm. In the LARS paper, the authors proved that starting from 0, the lasso solution paths grow piecewise linearly in a predictable way, and based on this they proposed a new algorithm called LARS to efficiently solve the whole lasso solution paths in the same order of computations as a single OLS fit.

By lemma 1, for each fixed λ_2 we transform the elastic net problem to an equivalent lasso problem on the artificial data set. We also made some modifications to the original LARS algorithm in order to take advantage of the sparse structure of $\mathbf{X}^*$, which is crucial in the $p \gg n$ case.

In detail, as outlined in the LARS paper, at the k-th step we need to invert the matrix $\mathbf{G}_{A_k} = \mathbf{X}_{A_k}^{*T}\mathbf{X}_{A_k}^*$, where A_k is the active variable set. This is done efficiently by updating or downdating the Cholesky factorization of $\mathbf{G}_{A_{k-1}}$ found at the previous step. Note that $\mathbf{G}_A = \frac{1}{1+\lambda_2}\left(\mathbf{X}_A^T\mathbf{X}_A + \lambda_2\mathbf{I}\right)$ for any index set A, so it amounts to updating or downdating the Cholesky factorization of $\mathbf{X}_{A_{k-1}}^T\mathbf{X}_{A_{k-1}} + \lambda_2\mathbf{I}$. It turns out that one can use a simple formula to update the Cholesky factorization of $\mathbf{X}_{A_{k-1}}^T\mathbf{X}_{A_{k-1}} + \lambda_2\mathbf{I}$, which is very similar to the formula used for updating the Cholesky factorization of

$\mathbf{X}_{A_{k-1}}^T \mathbf{X}_{A_{k-1}}$ (Golub & Van Loan 1983). The exact same downdating function can be used for downdating the Cholesky factorization of $\mathbf{X}_{A_{k-1}}^T \mathbf{X}_{A_{k-1}} + \lambda_2 \mathbf{I}$. 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 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 the LARS algorithm. It is also economical that we only record the non-zero coefficients and the active variables set at each LARS-EN step.

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

3.5 Choice of tuning parameters

Each combination (λ_1, λ_2) gives a unique elastic net solution, but (λ_1, λ_2) is not the only choice for the tuning parameters. 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 relation 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 always takes values from 0 to 1. In microarray data situation, the early stopping strategy is used. For example, if $n = 30$ and $p = 5000$, we may stop the LARS-EN algorithm after 500 steps. The best parameter are chosen according to the partial solution paths.

How do we pick the optimal parameters? Ideally we would like to choose the right parameters to minimize the risk on future data, such as population mean squared error (MSE). In practice people minimize an estimate of the risk, such as $\mathcal{C}_p$ statistic, or cross-validation error. For more details on this topic, see Chapter 7 of Hastie et al. (2001). In this work, we use 10-fold cross-validation to estimate the population risk. Note that there are actually two tuning parameters in the elastic net, so we cross-validate on a 2 dimensional surface. Typically we first pick a grid of λ_2, say $(0, 0.01, 0.1, 1, 10, 100)$, then for each λ_2 the LARS-EN algorithm produces the entire solution paths of the elastic net. The other tuning parameter is selected according the entire or partial (if early stopped) solution paths. If λ_1 and λ_2 do not appear together,

we will omit the subscript of λ_2.

4 Prostate Cancer Data Example

In this example, the predictors are eight clinical measures: log cancer volume (lcavol), log prostate weight (lweight), age, log of the amount of benign prostatic hyperplasia (lbph), seminal vesicle invasion (svi), log capsular penetration (lcp), Gleason score (gleason) and percentage Gleason score 4 or 5 (pgg45). The response is the log of prostate specific antigen (lpsa). The data came from a study (Stamey et al. 1989). They examined the correlation between the level of prostate specific antigen and those eight clinical measures.

OLS, ridge regression, the lasso, the naive elastic net and the elastic net were applied to these data. The prostate cancer data were divided into two parts: training set with 67 observations, and test set with 30 observations. Model fitting and tuning parameter selection by 10-fold cross-validation were done 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 the elastic net is the winner among all competitors in terms of both prediction accuracy and sparsity. OLS is the worst method. The naive elastic net is identical to ridge regression in this example. It fails to do variable selection. The lasso includes lcavol, lweight lbph, svi, and pgg45 in the final model, while the elastic net selects lcavol, lweight, svi, lcp, and pgg45. The prediction error of the elastic net is about 24 percent lower than that of the lasso. Also we see in this case the elastic net is actually UST because the selected λ is very big (1000). This can be considered as an empirical evidence to support UST.

If we check the correlation matrix of these eight predictors, we see there are a bunch of medium correlations although the highest is 0.76 (between pgg45 and gleason). We have seen that the elastic net is doing better than lasso by a good margin. In other words, the prediction power of the lasso gets hurt by the correlations. We conjecture that whenever ridge improves OLS the elastic net improves the lasso. We further demonstrate this point by simulations in the next section. Figure 3 displays the lasso and the elastic net solution paths.

Table 1: *Results of different methods applied to the prostate cancer data*

Method	Parameter(s)	Test MSE	Variables Selected
OLS		0.586 (0.184)	all
Ridge	$\lambda = 1$	0.566 (0.188)	all
Lasso	$s = 0.39$	0.499 (0.161)	(1,2,4,5,8)
Naive 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)

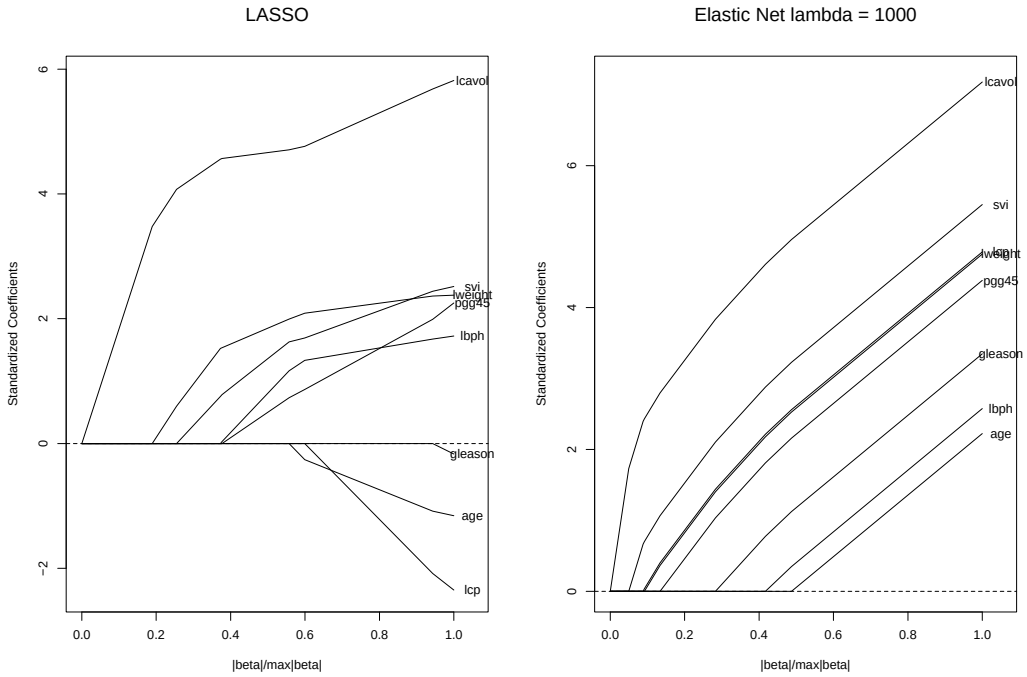

Figure 3: *The left panel shows the lasso estimates as a function of s, and the right panel shows the elastic net estimates as a function of s. Both of them are piecewise linear, which is a key property of our efficient algorithm. The solution paths also show the elastic net is identical to univariate soft-thresholding in this example.*

5 A Simulation Study

The purpose of simulation study 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 generate data from the true model

$$Y = \mathbf{X}\beta + \sigma\epsilon, \quad \epsilon \sim N(0,1).$$

Four examples are presented here. Within each example, our simulated data consisted 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 (mean squared errors) on the test data set. Notation $\cdot/\cdot/\cdot$ was used to describe the number of observations in the training, validation and test set respectively. For instance, 20/20/200 means there are 20 obs. in the training set, 20 obs. in the validation set and 200 obs. in the test set.

Example 1: We simulated 50 data sets consisting of 20/20/200 observations and 8 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 $cor(i,j) = (0.5)^{|i-j|}$.

Example 2: Same as example 1, except $\beta_j = 0.85$ for all j.

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 $\sigma = 15$; $cor(i,j) = 0.5$ for all i,j.

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 $\mathbf{X}$ are generated as the follows:

$$\begin{aligned}
X_i &= Z_1 + \epsilon_i^x, \quad Z_1 \sim N(0,1), \quad i = 1,\cdots,5,\\
X_i &= Z_2 + \epsilon_i^x, \quad Z_2 \sim N(0,1), \quad i = 6,\cdots,10,\\
X_i &= Z_3 + \epsilon_i^x, \quad Z_3 \sim N(0,1), \quad i = 11,\cdots,15,\\
X_i &\sim N(0,1), \qquad X_i \quad \text{i.i.d} \qquad i = 16,\ldots,40,
\end{aligned}$$

where ϵ_i^x are iid $N(0, 0.01)$, $i = 1,\ldots,15$. In this model, we have 3 equally important groups, within each group there are 5 members.

There are also 25 pure noise features. An ideal method would only select the 15 true features and set the coefficients of the 25 noise features to be 0.

OLS performs much worse than ridge and the lasso in all four examples, so we do not show its results. Table 2 and Figure 4 (Box-plots) summarize the prediction accuracy comparison results. Table 3 shows the variable selection results. In examples 1 and 4 the lasso performs better than ridge, while in examples 2 and 3 ridge regression does better than the lasso. The naive elastic net either has very poor performance (in example 1) or behaves very similar to ridge regression (in example 2 and 3) or the lasso (in example 4). In all examples, the elastic net predicts better than the lasso, even when the lasso is doing much better than ridge. The reductions of the prediction error in four examples are 18%, 18%, 13% and 27%, respectively. The simulation results indicate that just like ridge dominates OLS, the elastic net dominates the lasso by a good margin when there are high correlations or many moderate pairwise correlations. Meanwhile the elastic net still keeps a sparse representation similar to the lasso. In addition, the elastic net tends to select more variables than the lasso in scenarios 1 and 2, which is due to the grouping effect. In example 4, 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 $unif(0, 20)$. Response Y is generated by $Y = Z_1 + 0.1Z_2 + N(0, 1)$. Suppose we only observe

$$\mathbf{X}_1 = Z_1 + \epsilon_1, \quad \mathbf{X}_2 = -Z_1 + \epsilon_2, \quad \mathbf{X}_3 = Z_1 + \epsilon_3,$$
$$\mathbf{X}_4 = Z_2 + \epsilon_4, \quad \mathbf{X}_5 = -Z_2 + \epsilon_5, \quad \mathbf{X}_6 = Z_2 + \epsilon_6,$$

where ϵ_i are iid $N(0, \frac{1}{16})$. 100 data were generated from this model. So $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ form a group whose underlying factor is Z_1, and $\mathbf{X}_4, \mathbf{X}_5, \mathbf{X}_6$ form the other group whose underlying factor is Z_2. The within group correlations are almost 1 and the between group correlations are almost 0. 100 data were generated. Figure 5 displays the solution paths of the lasso and the elastic net.

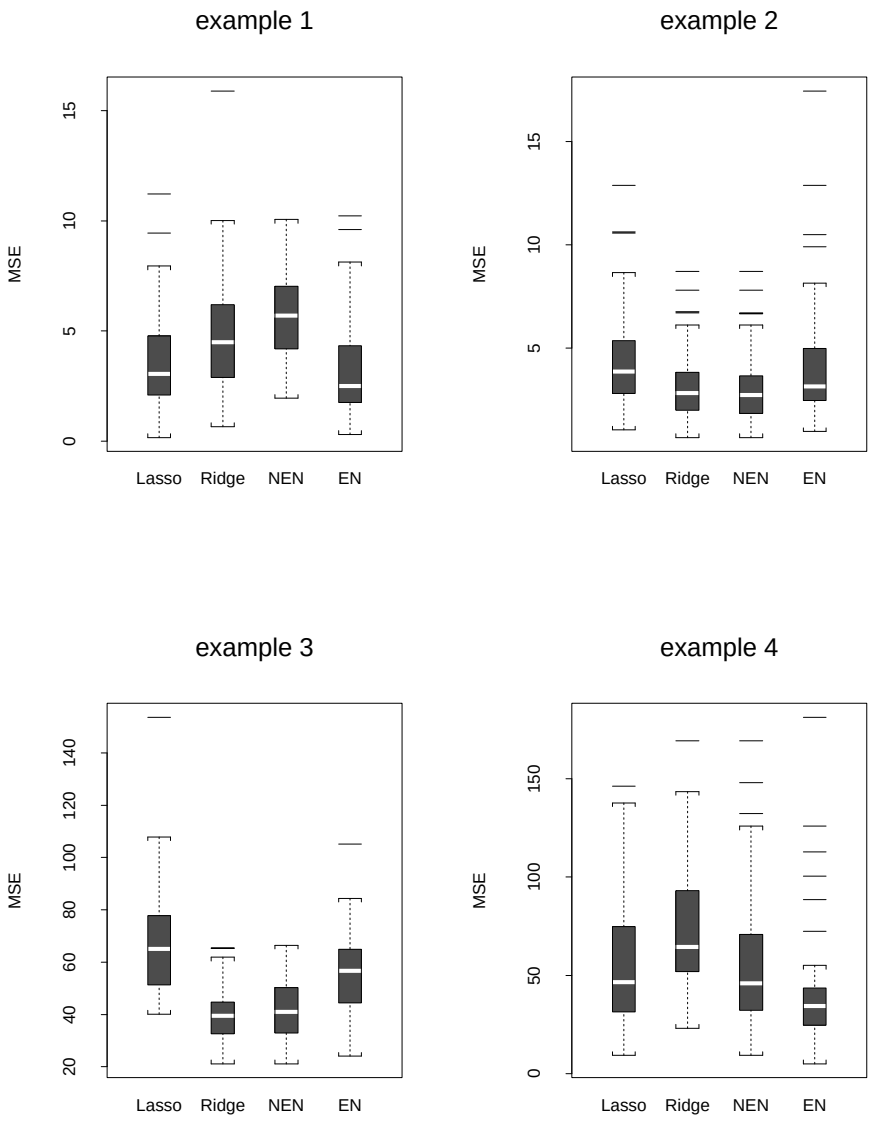

Figure 4: *Comparing prediction accuracy of the lasso, ridge, the naive elastic net(NEN) and the elastic net(EN). The elastic net outperforms the lasso in all four examples.*

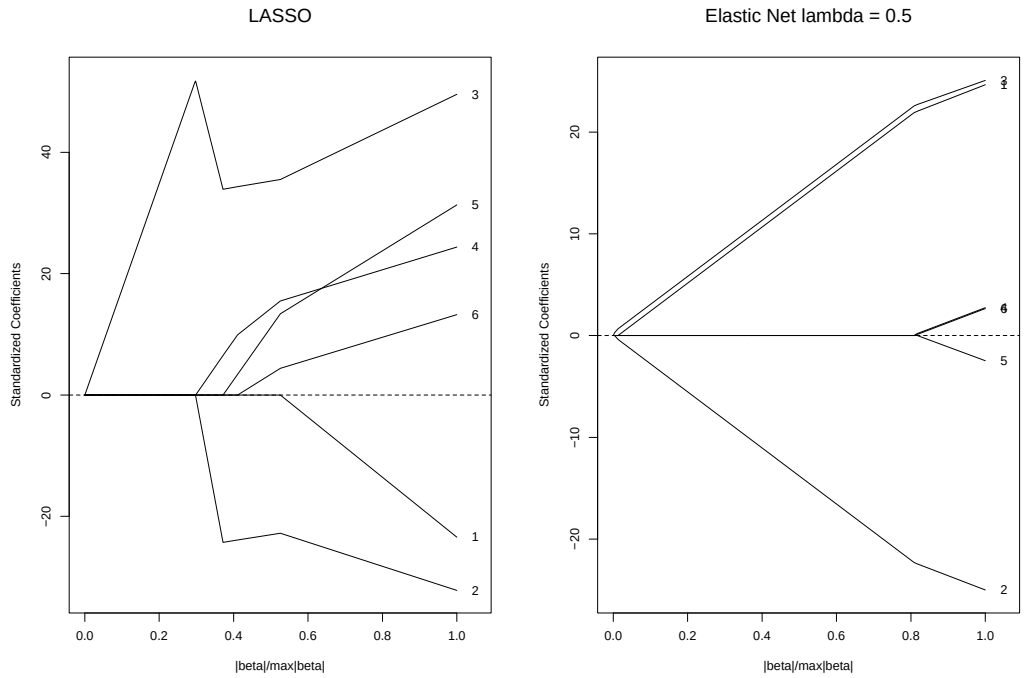

Figure 5: *The left and right panel show the lasso and the elastic net ($\lambda_2 = 0.5$) solution paths respectively. As can be seen from the lasso solution plot, $\mathbf{X}_3$ and $\mathbf{X}_2$ are considered the most important variables in the lasso fit, but their paths are jumpy. The lasso plot does not reveal any correlation information by itself. In contrast, the elastic net has much smoother solution paths, while clearly showing the "grouped selection": $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ are in one "significant" group and $\mathbf{X}_4, \mathbf{X}_5, \mathbf{X}_6$ are in the other "trivial" group. The de-correlation yields grouping effect and stabilizes the lasso solution.*

Table 2: Median of MSE

Method	*Ex.1*	*Ex.2*	*Ex.3*	*Ex.4*
Lasso	3.06	3.87	65.0	46.6
Ridge	4.49	2.84	39.5	64.5
Naive elastic net	5.70	2.73	41.0	45.9
Elastic net	2.51	3.16	56.6	34.5

Table 3: Median of the number of non-zero coefficients

Method	*Ex.1*	*Ex.2*	*Ex.3*	*Ex.4*
Lasso	5	6	24	11
Elastic net	6	7	27	16

6 Microarray Data Example: Leukemia Classification

There are many good classification/regression methods for microarray analysis in terms of prediction performance, but they all require an *external* procedure to select important genes. For example, support vector machine and penalized logistic regression (Zhu & Hastie 2003) both use either univariate ranking (UR) (Golub et al. 1999) or recursive feature elimination (RFE) (Guyon et al. 2002) to reduce the number of genes in their final model. We prefer a method which can do classification/regression and gene selection simultaneously. A typical microarray data set has thousands of genes and less than 100 samples. Because of the unique structure of the microarray data, we feel a good method should have the following properties:

1. Gene selection should be *built into* the procedure.
2. It should not be limited by the fact $p \gg n$.
3. For those genes sharing the same biological "pathway", it should be able to automatically include whole groups into the model once one gene amongst them is selected.

As we have pointed out, even thought it is a promising variable selection method in the usual $n > p$ setting, the lasso fails to do both (1) and (2). 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". Those properties make the elastic net a good candidate for that purpose.

Let us consider a real microarray data example. The leukemia data have 7129 genes and 72 samples (Golub et al. 1999). In the training data set, there are 38 samples, among which 27 are type 1 leukemia (ALL) and 11 are type 2 leukemia (AML). The goal is to construct a diagnostic rule based on the expression level of those 7219 genes to predict the type of leukemia. 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 leukemia as 0-1 variable. Then the 0-1 response y was fitted by the elastic net. The classification function is $I(\text{fitted value} > 0.5)$, where $I(\cdot)$ is the indicator function. We optimized the elastic net by 10-fold cross-validation on the training data, using (λ, s) as the tuning parameter. To make the computation easier, we pre-screened out 1000 most "significant" genes as the predictors, according to their t-statistic used in SAM (Tusher et al. 2001). The pre-screening step is not crucial in our method.

The elastic net with $\lambda = 0.01$ and $s = 0.46$ selects 87 genes with 10-fold cross-validation error 1/38 and test error 0/34. If we sacrifice one more cross-validation error by using the one standard error rule, then we choose $s = 0.41$, which reduces the number of selected genes to 53 and still has 0 test error. As mentioned in Section 3.5 we can adopt the early stopping strategy to facilitate the computation. We stopped the LARS-EN algorithm after 200 steps. If using the steps of LARS-EN algorithm as the tuning parameter, $steps = 85$ gives 10-fold cross-validation error 2/38 and the test error 0/34 with 52 genes selected. Figure 6 shows the classification results, and Table 4 summarizes the results of several other competitors including Golubs' method, support vector machine (SVM), penalized logistic regression (PLR), nearest shrunken centroid (NSC) (Tibshirani et al. 2002). The elastic net gives the best classification results, and it is an *internal* gene selection method. We applied the elastic net to other microarray data and got good results too.

7 Discussion

We have proposed the elastic net, a novel shrinkage and selection regression method. The elastic net produces a sparse model with good prediction accuracy, while encouraging a grouping effect. The empirical results and simula-

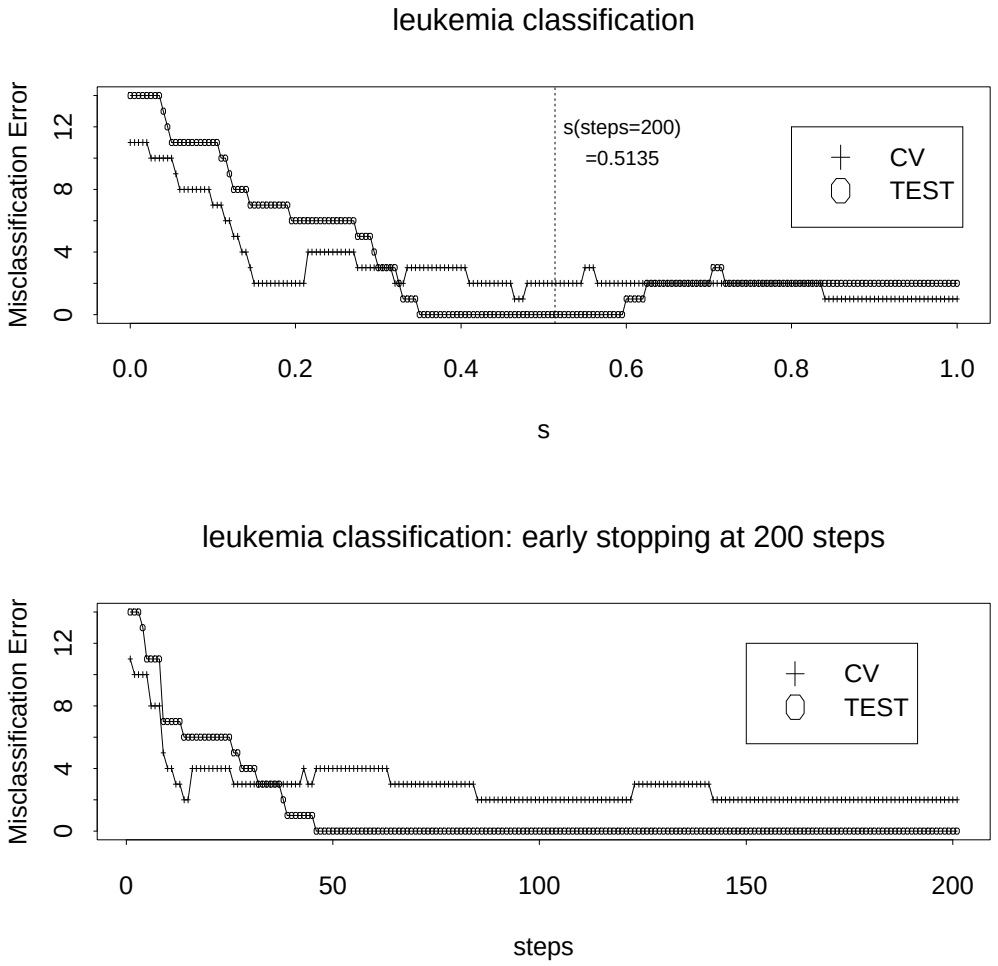

Figure 6: *Leukemia classification and gene selection by the elastic net($\lambda = 0.01$). The thin line shows the 10-fold cross-validation error, and the test error corresponds to the heavy line. The upper plot uses the whole elastic net solution paths, and the lower plot uses the early stopping strategy (stopped at 200 steps). We see these two methods give very similar results. Since computing s depends on the fit at the last step of the LARS-EN algorithm, when we use the early stopping strategy, the actual values of s are not available in 10-fold cross-validation. That is why we use steps as the regularization parameter. On the training set, steps=200 is equivalent to $s = 0.5135$, indicated by the broken vertical line in the upper plot. Note the one-one mapping $s = s(steps)$ is data-dependent, which means $s(steps = 200)$ may not be 0.5135 for another data set.*

Table 4: Summary of leukemia classification results

Method	*10-fold CV error*	*Test error*	*No. of genes*
Golub	3/38	4/34	50
SVM RFE	2/38	1/34	31
PLR RFE	2/38	1/34	26
NSC	2/38	2/34	21
Elastic net ($\lambda = 0.01, s = 0.46$)	1/38	0/34	87
Elastic net ($\lambda = 0.01, s = 0.41$)	2/38	0/34	53
Elastic net ($\lambda = 0.01, steps = 85$)	2/38	0/34	52

tions demonstrate the good performance of the elastic net and its superiority over the lasso. When used as a classification method, the elastic net performs very well in the analysis of microarray data in terms of misclassification error and automatic gene selection.

The elastic net estimate, being a combination of a lasso and ridge estimates, enjoys the good properties of both of the methods. Combining is not a new idea in the statistics literature. Some famous examples are Bagging (Breiman 1996), Boosting (Freund & Schapire 1997) and Bayesian model averaging (Hoeting et al. 1999). Those techniques share the same traditional combining strategy: the combined estimator is a convex combination of a list of estimators with cleverly chosen combining weights so that the combined estimator performs (much) better than any estimator in the list. However, the combining strategy in our work is very different. Instead of directly combining the lasso and ridge estimates, we combine the lasso and ridge penalties; and the combined estimator is a penalized estimator using the combined penalty, followed by some post-process adjustment (re-scaling in this work). To our knowledge, this combining strategy has not been reported in the literature.

We consider the elastic net as a generalization of the lasso, and do not suggest pushing the lasso away. In fact, the lasso is a valuable tool for model fitting and feature extraction . Recently the lasso was used to explain the success of Boosting. It is argued that Boosting performs a high-dimensional lasso without explicitly using the lasso penalty. See Hastie et al. (2001), Efron et al. (2004) and Friedman et al. (2004). We believe it is beneficial to have a better understanding of the properties of the lasso.

Acknowledgements

We thank Rob Tibshirani and Ji Zhu for helpful comments. 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: Proofs

Proof of Theorem 1

Part (1): Fix $\lambda_2 > 0$. $\forall \lambda_1 \geq 0$, by definition we have

$$\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2) = \arg\min_{\boldsymbol{\beta}} L(\lambda_1, \lambda_2, \boldsymbol{\beta}), \tag{13}$$

where $L(\lambda_1, \lambda_2, \boldsymbol{\beta})$ is defined in (3).

If $\hat{\beta}_i(\lambda_1, \lambda_2) \neq \hat{\beta}_j(\lambda_1, \lambda_2)$, let us consider an alternative estimates $\hat{\boldsymbol{\beta}}^*(\lambda_1, \lambda_2)$

$$\hat{\beta}^*_k(\lambda_1, \lambda_2) = \begin{cases} \hat{\beta}_k(\lambda_1, \lambda_2) & \text{if } k \neq i \text{ and } k \neq j \\ \frac{1}{2}\left(\hat{\beta}_i(\lambda_1, \lambda_2) + \hat{\beta}_j(\lambda_1, \lambda_2)\right) & \text{if } k = i \text{ or } k = j. \end{cases}$$

Because $\mathbf{X}_i = \mathbf{X}_j$, it is obvious that $\mathbf{X}\hat{\boldsymbol{\beta}}^*(\lambda_1, \lambda_2) = \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$, thus

$$\left|Y - \mathbf{X}\hat{\boldsymbol{\beta}}^*(\lambda_1, \lambda_2,)\right|^2 = \left|Y - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\right|^2.$$

On the other hand, we have

$$\left|\hat{\boldsymbol{\beta}}^*(\lambda_1, \lambda_2)\right|_1 \leq \left|\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\right|_1 \quad \text{and} \quad \left|\hat{\boldsymbol{\beta}}^*(\lambda_1, \lambda_2)\right|^2 < \left|\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\right|^2.$$

Therefore $L\left(\lambda_1, \lambda_2, \hat{\boldsymbol{\beta}}^*(\lambda_1, \lambda_2)\right) < L\left(\lambda_1, \lambda_2, \hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\right)$, which contradicts (13). So we must have $\hat{\beta}_i(\lambda_1, \lambda_2) = \hat{\beta}_j(\lambda_1, \lambda_2)$.

□

Part (2): 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. Moreover, we have

$$\begin{aligned} \operatorname{sgn}\left(\hat{\beta}_i(\lambda_1, \lambda_2)\right) &= \operatorname{sgn}\left(\hat{\beta}_j(\lambda_1, \lambda_2)\right), \\ D_{\lambda_1,\lambda_2}(i,j) &= \tfrac{1}{|Y|}\left|\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)\right|. \end{aligned}$$

Because of (13), $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ satisfies

$$\frac{\partial L(\lambda_1, \lambda_2, \boldsymbol{\beta})}{\partial \beta_k}\Big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}(\lambda_1,\lambda_2)} = 0 \quad \text{if } \hat{\beta}_k(\lambda_1, \lambda_2) \neq 0. \tag{14}$$

Hence we have

$$-2\mathbf{X}_i^T\left(Y - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\right) + \lambda_1 \text{sgn}\left(\hat{\beta}_i(\lambda_1, \lambda_2)\right) + 2\lambda_2\hat{\beta}_i(\lambda_1, \lambda_2) = 0, \tag{15}$$

$$-2\mathbf{X}_j^T\left(Y - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\right) + \lambda_1 \text{sgn}\left(\hat{\beta}_j(\lambda_1, \lambda_2)\right) + 2\lambda_2\hat{\beta}_j(\lambda_1, \lambda_2) = 0. \tag{16}$$

Substracting (15) from (16) gives

$$(\mathbf{X}_j^T - \mathbf{X}_i^T)\left(Y - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\right) + \lambda_2\left(\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)\right) = 0,$$

which is equivalent to

$$\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2) = \frac{1}{\lambda_2}(\mathbf{X}_i^T - \mathbf{X}_j^T)\hat{r}(\lambda_1, \lambda_2), \tag{17}$$

where $\hat{r}(\lambda_1, \lambda_2) = Y - \mathbf{X}\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ is the residual vector. Since $\mathbf{X}$ are standardized, $|\mathbf{X}_i - \mathbf{X}_j|^2 = 2(1-\rho)$ where $\rho = cor(\mathbf{X}_i, \mathbf{X}_j)$. By (13) we must have

$$L\left(\lambda_1, \lambda_2, \hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\right) \leq L(\lambda_1, \lambda_2, \boldsymbol{\beta} = 0),$$
$$\text{i.e., } |\hat{\boldsymbol{r}}(\lambda_1, \lambda_2)|^2 + \lambda_2\left|\hat{\beta}(\lambda_1, \lambda_2)\right|^2 + \lambda_1\left|\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)\right|_1 \leq |Y|^2.$$

So $|\hat{\boldsymbol{r}}(\lambda_1, \lambda_2)| \leq |Y|$. Then (17) implies

$$D_{\lambda_1,\lambda_2}(i, j) \leq \frac{1}{\lambda_2}\frac{|\hat{\boldsymbol{r}}(\lambda_1, \lambda_2)|}{|Y|}|\mathbf{X}_i - \mathbf{X}_j| \leq \frac{1}{\lambda_2}\sqrt{2(1-\rho)}.$$

□

Proof of Theorem 2

Let $\hat{\beta}$ be the elastic net estimates. By definition and (7) we have

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}}\left|Y^* - \mathbf{X}^*\frac{\boldsymbol{\beta}}{\sqrt{1+\lambda_2}}\right|^2 + \frac{\lambda_1}{\sqrt{1+\lambda_2}}\left|\frac{\boldsymbol{\beta}}{\sqrt{1+\lambda_2}}\right|_1 \\ &= \arg\min_{\boldsymbol{\beta}}\boldsymbol{\beta}^T\left(\frac{\mathbf{X}^{*T}\mathbf{X}^*}{1+\lambda_2}\right)\boldsymbol{\beta} - 2\frac{Y^{*T}X^*}{\sqrt{1+\lambda_2}} + Y^{*T}Y^* + \frac{\lambda_1|\beta|_1}{1+\lambda_2}. \end{aligned} \tag{18}$$

Substituting the identities

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

into (18), we have

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

□

References

Breiman, L. (1996), 'Bagging predictors', *Machine Learing* **24**, 123–140.

Donoho, D., Johnstone, I., Kerkyacharian, G. & Picard, D. (1995), 'Wavelet shrinkage: asymptopia? (with discussion)', *J.R. Statist. Soc. B* **57**, 301–337.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), 'Least angle regression', *The Annals of Statistics* **32**, In press.

Freund, Y. & Schapire, R. (1997), 'A decision-theoretic generalization of online learning and an application to boosting', *Journal of Computer and System Sciences* **55**, 119–139.

Friedman, J., Hastie, T., Rosset, S., Tibshirani, R. & Zhu, J. (2004), 'Discussion of boosting papers', *The Annals of Statistics* **32**, In press.

Golub, G. & Van Loan, C. (1983), *Matrix computations*, Johns Hopkins University Press.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J. & 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. & Vapnik, V. (2002), 'Gene selection for cancer classification using support vector machines', *Machine Learning* **46**, 389–422.

Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The Elements of Statistical Learning; Data mining, Inference and Prediction*, Springer Verlag, New York.

Hoeting, J., Madigan, D., Raftery, A. & Volinsky, C. (1999), 'Bayesian model averaging: A tutorial (with discussion)', *Statistical Science* **14**, 382–417.

Stamey, T., Kabalin, J., Mcneal, J., Johnstone, I., F. F., Redwine, E. & 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. & Chu, G. (2002), 'Diagnosis of multiple cancer types by shrunken centroids of gene expression', *Proceedings of the National Academy of Sciences* **99**, 6567–6572.

Tusher, V., Tibshirani, R. & Chu, C. (2001), 'Significance analysis of microarrays applied to transcriptional responses to ionizing radiation', *Proceedings of the National Academy of Sciences* **98**, 5116–5121.

Zhu, J. & Hastie, T. (2003), Classification of gene microarrys by penalized logistic regression, Technical report, Department of Statistic, Stanford University.