SOME PERSPECTIVES OF SPARSE STATISTICAL MODELING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF STATISTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Hui Zou

May 2005

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Trevor Hastie Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Robert Tibshirani

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Bradley Efron

Approved for the University Committee on Graduate Studies.

Abstract

In this thesis we develop some new sparse modeling techniques and related theory. We first point out the fundamental drawbacks of the lasso in some scenarios: (1) the number of predictors (greatly) exceeds the number of observations; (2) the predictors are highly correlated and form "groups". A typical example where these scenarios naturally occur is the gene selection problem in microarray analysis. We then propose the elastic net, a new regularization and variable selection method, to further improve upon the lasso. We 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 is much bigger that the number of samples. We also propose an algorithm called LARS-EN for efficiently computing the entire elastic net regularization path, much like the LARS algorithm does for the lasso.

The second part of the thesis shows a nice application of the elastic net in deriving principal components with sparse loadings. We propose a principled approach called SPCA to modify PCA based on a novel sparse PCA criterion in which an elastic net constraint is used to produce sparse loadings. To solve the optimization problem in SPCA, we consider an alternating algorithm which iterates between the elastic net and the reduced rank Procrustes rotation. SPCA allows flexible control of the sparse structure of the resulting loadings and has the ability of identifying important variables.

In the third part of the thesis, we study the degrees of freedom of the lasso in the framework of SURE theory. We prove that the number of non-zero coefficients is an unbiased

estimate for the degrees of freedom of the lasso—a conclusion requires no special assumption on the predictors. Our analysis also provides mathematical support for a related conjecture by Efron et al. (2004). As an application, various model selection criteria—C_p, AIC and BIC—are defined, which, along with the LARS algorithm, provide a principled and efficient approach to obtaining the optimal Lasso fit with the computational efforts of a single ordinary least-squares fit. The degrees of freedom of the elastic net can be obtained by similar arguments.

Acknowledgments

First of all, I wish to take this opportunity to express my great appreciation to my advisor Professor Trevor Hastie. I benefit so much from his deep insight into statistics, his professionalism and his constant encouragement. Trevor has been very generous and patient to me. I feel extremely lucky to have Trevor as my thesis advisor.

I am deeply grateful to Professor Robert Tibshirani for his invaluable advice and helpful comments. I am also very grateful to Professor Bradley Efron for teaching me elegant statistics and writing important recommendation letters to support my job application. My sincere thanks also go to Professor Jerome Friedman and Professor Art Owen for joining my oral committee and providing many suggestions on my research.

I wish to thank other faculty members in the statistics department for their excellent lectures. I thank my friends for their sincere help. I will always cherish our friendship.

Finally, I want to thank my wife Shufeng. Without her love and support, this work would not have been completed. I would like to dedicate this thesis to my parents for their boundless love.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the practice of statistical modeling, it is often desirable to have an accurate predictive model with a sparse representation. Modern data sets usually have a large number of predictors, hence parsimony is especially an important issue. Best-subset selection is a conventional method of variable selection. It is now well-known that best-subset selection has two serious drawbacks. First, it is extremely variable because of its inherent discreteness. Secondly, when the number of variables is large, best-subset selection is computationally infeasible, because the total number of the subset models grows exponentially. To ease the computational burden, some greedy stepwise algorithms, such as backward selection, are used as a surrogate to best-subset selection. However, these kind of greedy algorithms are notorious for their poor performance.

The lasso (Tibshirani 1996) opens a new door to variable selection by using the L_1 penalty in the model fitting criterion. Due to the nature of the L_1 penalty, the lasso performs continuous shrinkage and variable selection simultaneously. Thus the lasso possesses the nice properties of both the L_2 (ridge) penalization and best-subset selection. It is forcefully argued that the automatic feature selection property makes the lasso a better choice than the L_2 penalization in high dimensional problems, especially when there are lots of redundant noise features (Friedman, Hastie, Rosset, Tibshirani & Zhu 2004), although the L_2 regularization has been widely used in various learning problems such as smoothing splines

(Wahba 1990), the support vector machine (Vapnik 1995) and neural networks where the L_2 regularization is called *weight decay*(Hastie, Tibshirani & Friedman 2001). An L_1 method called *basis pursuit* was also used in signal processing (Chen, Donoho & Saunders 2001). There are many theoretical work to prove the superiority of the L_1 penalization in sparse settings. Donoho, Johnstone, Kerkyacharian & Picard (1995) prove the near minimax optimality of soft-thresholding (L_1 shrinkage with orthogonal predictors). It is also shown that the L_1 approach is able to discover the "right" sparse representation of the model under certain conditions (Donoho & Huo 2001, Donoho & Elad 2002, Donoho 2004).

Although it has shown success in many situations, the lasso may produce unsatisfactory results in some scenarios: (1) the number of predictors (greatly) exceeds the number of observations; (2) the predictors are highly correlated and form "groups". A typical example where the two scenarios are common is the gene selection problem in microarray analysis. In this thesis we develop new sparse modeling techniques which keep the promising properties of the L_1 method and fix the two drawbacks highlighted above.

In Chapter 2 we propose the elastic net, a new regularization and automatic variable selection method, to further improve upon the lasso. We 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 is much bigger that the number of samples. We also propose an algorithm called LARS-EN for efficiently computing the entire elastic net regularization path, much like the LARS algorithm does for the lasso.

Sparse representation is also important in unsupervised learning. Principal component analysis (PCA) is widely used unsupervised learning tool in data processing and dimensionality reduction. However, PCA suffers from the fact that each principal component is a linear combination of all the original variables, thus it is often difficult to interpret the results. A naive method is often used to obtain sparse loadings by artificially setting small loadings to zero, which can be potentially misleading in various respects. In Chapter 3 we show a nice application of the elastic net in deriving principal components with sparse

loadings. We consider a principled approach called SPCA to modify PCA based on a new sparse PCA criterion in which an elastic net constraint is used to produce sparse loadings. To solve the optimization problem in SPCA, we consider an efficient alternating algorithm which iterates between the elastic net and the reduced rank Procrustes rotation. As a principled procedure, SPCA enjoys advantages in several aspects, including computational efficiency, high explained variance and an ability in identifying important variables.

Chapter 4 concerns the *effective degrees of freedom* of the lasso, which is very important for understanding the model complexity of the lasso and very useful in selecting the "best" lasso model. Degrees of freedom are well studied for linear procedures, where $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$ and the smoother matrix $\mathbf{S}$ is free of $\mathbf{y}$. In this case $df(\mathbf{S}) = \mathrm{tr}(\mathbf{S})$. However, because of the nonlinear nature of the L_1 penalization, these nice results for linear smoothers are not directly applicable. Efron et al. (2004) presented a conjecture to approximate the degrees of freedom of the lasso. We study the degrees of freedom of the lasso in the framework of Stein's unbiased risk estimation (Stein 1981). We prove that the number of non-zero coefficients is an unbiased estimate for the degrees of freedom of the lasso—a conclusion requires no special assumption on the predictors. Our analysis also provides mathematical support for the conjecture by Efron et al. (2004). Our results can be used to define various model selection criteria—C_p, AIC and BIC—which, along with the LARS algorithm, provide a principled and efficient approach to obtaining the optimal Lasso fit with the computational efforts of a single ordinary least-squares fit. The degrees of freedom of the elastic net can be obtained by similar arguments.

A summary of the thesis is given in Chapter 5.

Chapter 2

The Elastic Net

In this Chapter we propose the elastic net, a new regularization and variable selection method. We 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 the LARS algorithm does for the lasso.

2.1 Introduction and Motivation

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

$$\hat{\mathbf{y}} = \hat{\beta}_0 + \mathbf{x}_1\hat{\beta}_1 + \cdots + \mathbf{x}_p\hat{\beta}_p. \tag{2.1}$$

A model-fitting procedure produces 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). 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 relationship 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. Penalization techniques have been proposed to improve OLS. For example, ridge regression (Hoerl & Kennard 1988) minimizes RSS 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 on the other hand 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 a L_1 penalty on the regression coefficients. Due 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 & Friedman 1993) and found 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 due to its sparse representation.

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

1. 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.

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. See Section 2.2.3.

3. For usual $n > p$ situations, if there exist high correlations among predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression (Tibshirani 1996) .

Scenarios (1) and (2) 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 less than 100 samples. For those genes sharing the same biological "pathway", the correlations among them can be high (Segal, Dahlquist & 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, Hastie, Johnstone & Tibshirani 2004), and it lacks the ability to reveal the grouping information. As for prediction performance, scenario (3) is not rare in regression problems. So it is possible to further strengthen 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 highlighted above; i.e., it should mimic the ideal variable selection method in scenarios (1) and (2), especially with microarray data, and it should deliver better prediction performance than the lasso in scenario (3).

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 is able to select groups of correlated variables. It is like a stretchable fishing net that retains "all the big fish". In Section 2.2 we define the *naive elastic net*, which is a penalized least squares method using a novel *elastic net penalty*. We discuss the grouping

effect caused by the elastic net penalty. In Section 2.3, we show that this naive procedure tends to over-shrink in regression problems. We then introduce the *elastic net*, which corrects this problem. An efficient LARS-EN algorithm is proposed for computing the entire elastic net regularization paths with the computational effort of a single OLS fit. Prostate cancer data is used to illustrate our methodology in Section 2.4, and simulation results comparing the lasso and the elastic net are presented in Section 2.5. Section 2.6 shows an application of the elastic net to classification and gene selection in a Leukemia microarray problem.

2.2 Naive Elastic Net

2.2.1 Definition

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

$$\sum_{i=1}^{n} y_i = 0, \quad \sum_{i=1}^{n} x_{ij} = 0, \text{ and } \sum_{i=1}^{n} x_{ij}^2 = 1, \text{ for } j = 1, 2, \ldots, p. \tag{2.2}$$

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

$$L(\lambda_1, \lambda_2, \boldsymbol{\beta}) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_2 \|\boldsymbol{\beta}\|^2 + \lambda_1 \|\boldsymbol{\beta}\|_1, \tag{2.3}$$

where

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

The naive elastic net estimator $\hat{\boldsymbol{\beta}}$ is the minimizer of (2.3):

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} L(\lambda_1, \lambda_2, \boldsymbol{\beta}) . \tag{2.4}$$

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 (2.3) is equivalent to the optimization problem:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2, \text{ subject to } (1-\alpha)\|\boldsymbol{\beta}\|_1 + \alpha\|\boldsymbol{\beta}\|^2 \leq t \text{ for some t.} \tag{2.5}$$

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 naive elastic net becomes simple ridge regression. In this paper, we only consider $\alpha < 1$. 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 possessing the characteristics of both the lasso and ridge. Note that the lasso penalty ($\alpha = 0$) is convex but not strictly convex. These arguments can be seen clearly from Figure 2.1.

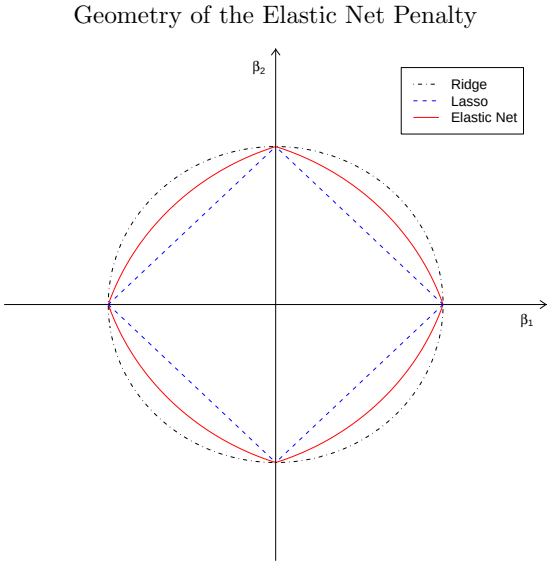

Figure 2.1: *2-dimensional contour plots (level=1). The outermost contour shows the shape of the ridge penalty while the diamond shaped curve is the contour of the lasso penalty. The solid curve is the contour plot of the elastic net penalty with* $\alpha = 0.5$. *We see singularities at the vertexes and the edges are strictly convex. The strength of convexity varies with* α.

2.2.2 Solution

We now develop a method to solve the naive elastic net problem efficiently. It turns out that minimizing (2.3) is equivalent to a lasso type optimization problem. This fact implies that the naive elastic net also enjoys the computational advantage of the lasso.

Lemma 2.1. *Given data set* $(\mathbf{y}, \mathbf{X})$ *and* (λ_1, λ_2)*, define an artificial data set* $(\mathbf{y}^*, \mathbf{X}^*)$ *by*

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

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*

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

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 2.1 says that we can transform the naive elastic net problem to 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 the naive elastic net can potentially select all p predictors in all situations. This important property overcomes the limitations of the lasso described in scenario (1). Lemma 2.1 also shows that the naive elastic net can perform an automatic variable selection in a fashion similar to the lasso. In the next section we show that the naive elastic net has the ability of selecting "grouped" variables, a property not shared by the lasso.

In the case of an orthogonal design, it is straightforward to show that with parameters

(λ_1, λ_2), the naive elastic net solution is

$$\hat{\beta}_i(\text{naive elastic net}) = \frac{\left(\left|\hat{\beta}_i(\text{ols})\right| - \frac{\lambda_1}{2}\right)_+}{1+\lambda_2} \text{sgn}\left(\hat{\beta}_i(\text{ols})\right). \tag{2.6}$$

where $\hat{\boldsymbol{\beta}}(\text{ols}) = \mathbf{X}^{\mathrm{T}}\mathbf{y}$ and z_+ denotes the positive part, which is z if $z > 0$, else 0. 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 $\hat{\beta}_i(\text{lasso}) = \left(\left|\hat{\beta}_i(\text{ols})\right| - \frac{\lambda_1}{2}\right)_+ \text{sgn}\left(\hat{\beta}_i(\text{ols})\right)$. Figure 2.2 shows the operational characteristics of the three penalization methods in an orthogonal design, where the naive elastic net can be viewed as a two-stage procedure: a ridge-type direct shrinkage followed by a lasso-type thresholding.

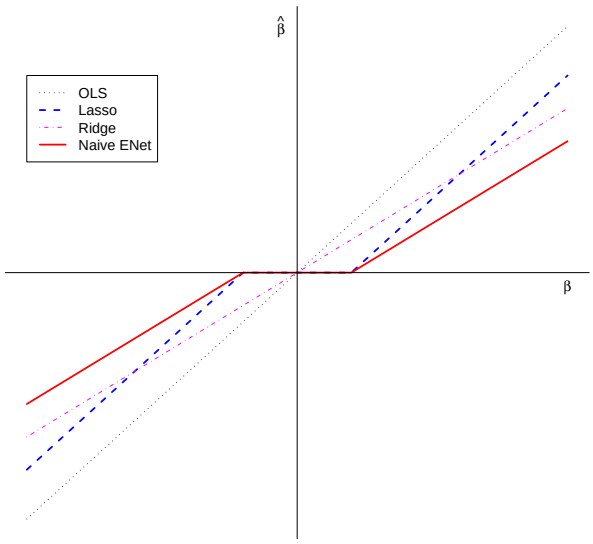

Figure 2.2: *Exact solutions for the lasso, ridge and the naive elastic net (naive ENet) in an orthogonal design. Shrinkage parameters are* $\lambda_1 = 2, \lambda_2 = 1$.

2.2.3 The grouping effect

In the "large p, small n" problem (West, Blanchette, Dressman, Huang, Ishida, Spang, Zuzan, Marks & Nevins 2001), the "grouped variables" situation is a particularly important

concern, which has been addressed a number of times in the literature. For example, principal component analysis (PCA) has been used to construct methods for finding a set of highly correlated genes in Hastie, Tibshirani, Eisen, Brown, Ross, Scherf, Weinstein, Alizadeh, Staudt & Botstein (2000) and Díaz-Uriarte (2003). Tree harvesting (Hastie, Tibshirani, Botstein & Brown 2003) uses supervised learning methods to select groups of predictive genes found by hierarchical clustering. Using an algorithmic approach, Dettling & Bühlmann (2004) perform the clustering and supervised learning together. A careful study by Segal, Dahlquist & Conklin (2003) strongly motivates the use of regularized regression procedure to find the grouped genes. We consider the generic penalization method

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda J(\boldsymbol{\beta}) \tag{2.7}$$

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 sign change 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 2.2. *Assume* $\mathbf{x}_i = \mathbf{x}_j$, $i, j \in \{1, \ldots, p\}$.

1. *If* $J(\cdot)$ *is strictly convex, then* $\hat{\beta}_i = \hat{\beta}_j \ \forall \ \lambda > 0$.

2. *If* $J(\boldsymbol{\beta}) = |\boldsymbol{\beta}|_1$, *then* $\hat{\beta}_i \hat{\beta}_j \geq 0$ *and* $\hat{\boldsymbol{\beta}}^*$ *is another minimizer of (2.7), 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 2.2 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 2.1. *Given data* $(\mathbf{y}, \mathbf{X})$ *and parameters* (λ_1, λ_2)*, the response* $\mathbf{y}$ *is centered and the predictors* $\mathbf{X}$ *are standardized. Let* $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ *be the naive elastic net estimate. Suppose* $\hat{\beta}_i(\lambda_1, \lambda_2)\,\hat{\beta}_j(\lambda_1, \lambda_2) > 0$*. Define*

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

then $D_{\lambda_1,\lambda_2}(i,j) \leq \frac{1}{\lambda_2}\sqrt{2(1-\rho)}$*, where* $\rho = \mathbf{x}_i^T\mathbf{x}_j$*, the sample correlation.*

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 $-\mathbf{x}_j$), Theorem 2.1 says the difference between the coefficient paths of predictor i and predictor j is almost 0. The upper bound in the above inequality provides quantitative description for the grouping effect of the naive elastic net.

The lasso does not possess the grouping effect. Scenario (2) in Section 2.1 occurs frequently in practice. A theoretical explanation is given in Efron, Hastie, Johnstone & Tibshirani (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 get $\left|\hat{\beta}_1 - \hat{\beta}_2\right| = |cos(\theta)|$, where θ is the angle between $\mathbf{y}$ and $\mathbf{x}_1 - \mathbf{x}_2$. It is easy to construct examples such that $\rho = \text{cor}(\mathbf{x}_1, \mathbf{x}_2) \to 1$ but $cos(\theta)$ does not vanish.

Although Theorem 2.1 works with squared error loss, the group effect of the elastic net is an inherent property of the penalty, holding under other popular loss functions. Let ϕ be an arbitrary loss function and we consider the problem of ϕ risk minimization with the elastic net regularization

$$(\hat{\alpha}, \hat{\beta}) = \arg\min_{\alpha,\beta} \sum_{k=1}^{n} \phi\left(y_k, (\alpha + \mathbf{x}_k^T\boldsymbol{\beta})\right) + \lambda_2\|\boldsymbol{\beta}\|^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \tag{2.8}$$

In regression analysis, we take

$$\phi\left(y_k, (\alpha + \mathbf{x}_k^T\boldsymbol{\beta})\right) = \phi\left(y_k - (\alpha + \mathbf{x}_k^T\boldsymbol{\beta})\right); \tag{2.9}$$

and for classification we can let ϕ

$$\phi\left(y_k, (\alpha + \mathbf{x}_k^T\boldsymbol{\beta})\right) = \phi\left(y_k(\alpha + \mathbf{x}_k^T\boldsymbol{\beta})\right), \tag{2.10}$$

where y is coded as 1 or -1.

It is easy to check that Lemma 2.2 also holds for a general loss function. Furthermore, we have the following theorem concerning the grouping effect caused by the elastic net penalty.

Theorem 2.2. *Assume that the loss function ϕ is Lipschitz, i.e.,*

$$|\phi(t_1) - \phi(t_2)| \leq M\,|t_1 - t_2| \quad \textit{for some positive } M.$$

Then $\forall$ a pair of (i, j), we have

$$|\hat{\beta}_i - \hat{\beta}_j| \leq \frac{M}{\lambda_2}\|\mathbf{x}_i - \mathbf{x}_j\|_1.$$

The above inequality holds for all $\lambda_1 \geq 0$.

If the predictors $\mathbf{x}$ are centered and normalized, we have

$$|\hat{\beta}_i - \hat{\beta}_j| \leq \frac{\sqrt{n}M}{\lambda_2}\sqrt{2(1-\rho)}.$$

where $\rho = \mathrm{cor}(\mathbf{x}_i, \mathbf{x}_j)$.

The Lipschitz condition is easily satisfied by many popular loss functions used in practice. For example, the support vector machine loss is Lipschitz with $M = 1$. So Theorem 2.2 is actually a very useful result.

2.2.4 Bayesian connections and the L_q penalty

Bridge regression (Frank & Friedman 1993, Fu 1998) has $J(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_q^q = \sum_{j=1}^p |\boldsymbol{\beta}_j|^q$ in (2.7), which is a generalization of both the lasso ($q = 1$) and ridge ($q = 2$). The bridge estimator can be viewed as the Bayes posterior mode under the prior

$$p_{\lambda,q}(\boldsymbol{\beta}) = C(\lambda, q) \exp(-\lambda \|\boldsymbol{\beta}\|_q^q). \tag{2.11}$$

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

$$p_{\lambda,\alpha}(\boldsymbol{\beta}) = C(\lambda, \alpha) \exp(-\lambda(\alpha \|\boldsymbol{\beta}\|^2 + (1 - \alpha)\|\boldsymbol{\beta}\|_1)); \tag{2.12}$$

a compromise between the Gaussian and Laplacian priors. Although bridge 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, while the bridge does not. Fan & Li (2001) prove that in the L_q ($q \geq 1$) penalty family, only the lasso penalty ($q = 1$) can produce a sparse solution. Bridge ($1 < q < 2$) always keeps all predictors in the model, as does ridge. Since automatic variable selection via penalization is a primary objective of this article, L_q ($1 < q < 2$) penalization is not a candidate.

2.3 Elastic Net

2.3.1 Deficiency of the naive elastic net

As an automatic variable selection method, the naive elastic net overcomes the limitations of the lasso in scenarios (1) and (2). However, empirical evidence (see Sections 2.4 and 2.5) shows that the naive elastic net does not perform satisfactorily unless it is very close to either ridge or the lasso. This is the reason we call it *naive*.

In the regression prediction setting, an accurate penalization method achieves good prediction performance through the bias-variance trade-off. 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. 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 naive elastic net by correcting this double-shrinkage.

2.3.2 The elastic net estimate

We follow the notation in Section 2.2.2. Given data $(\mathbf{y}, \mathbf{X})$, penalty parameter (λ_1, λ_2), and augmented data $(\mathbf{y}^*, \mathbf{X}^*)$, the naive elastic net solves a lasso type problem

$$\hat{\boldsymbol{\beta}}^* = \arg\min_{\boldsymbol{\beta}^*} \|\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta}^*\|^2 + \frac{\lambda_1}{\sqrt{1+\lambda_2}}\|\boldsymbol{\beta}^*\|_1. \tag{2.13}$$

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

$$\hat{\boldsymbol{\beta}}\,(\text{elastic net}) = \sqrt{1+\lambda_2}\hat{\boldsymbol{\beta}}^*. \tag{2.14}$$

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

$$\hat{\boldsymbol{\beta}}(\text{elastic net}) = (1+\lambda_2)\hat{\boldsymbol{\beta}}(\text{naive elastic net}). \tag{2.15}$$

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

Such a scaling transformation preserves the variable-selection property of the naive elastic net, and is the simplest way to undo shrinkage. Hence all the good properties of the naive elastic net described in Section 2.2 hold for the elastic net. Empirically we have found the elastic net performs very well when compared with the lasso and ridge.

We have another justification for choosing $1 + \lambda_2$ as the scaling factor. Consider the exact solution of the naive elastic net when the predictors are orthogonal. The lasso is known to be minimax optimal (Donoho, Johnstone, Kerkyacharian & Picard 1995) in this case, which implies the naive 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

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

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

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

We can rewrite $\mathbf{R}$ as

$$\mathbf{R} \;= \tfrac{1}{1+\lambda_2}\mathbf{R}^* = \tfrac{1}{1+\lambda_2}\begin{bmatrix} 1 & \frac{\rho_{12}}{(1+\lambda_2)} & \cdot & \frac{\rho_{1p}}{(1+\lambda_2)} \\ & 1 & \cdot & \cdot \\ & & 1 & \frac{\rho_{p-1,p}}{(1+\lambda_2)} \\ & & & 1 \end{bmatrix}^{-1} \mathbf{X}^T. \tag{2.16}$$

$\mathbf{R}^*$ is like the usual OLS operator except the correlations are shrunk by factor $\frac{1}{1+\lambda_2}$, which we call de-correlation. Hence from (2.16) we can interpret the ridge operator as de-correlation followed by direct scaling shrinkage.

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

Theorem 2.3. *Given data* $(\mathbf{y}, \mathbf{X})$ *and* (λ_1, λ_2)*, then the elastic net estimates* $\hat{\boldsymbol{\beta}}$ *are given by*

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^T \left(\frac{\mathbf{X}^T\mathbf{X} + \lambda_2 \mathbf{I}}{1+\lambda_2} \right) \boldsymbol{\beta} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\beta} + \lambda_1 \|\boldsymbol{\beta}\|_1. \tag{2.17}$$

It is easy to see that

$$\hat{\boldsymbol{\beta}}(\text{lasso}) = \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^T (\mathbf{X}^T\mathbf{X})\boldsymbol{\beta} - 2\mathbf{y}^T\mathbf{X}\boldsymbol{\beta} + \lambda_1 \|\boldsymbol{\beta}\|_1. \tag{2.18}$$

Hence Theorem 2.3 interprets the elastic net as a stabilized version of the lasso. Note that $\widehat{\Sigma} = \mathbf{X}^T\mathbf{X}$ is a sample version of the correlation matrix (Σ) and $\frac{\mathbf{X}^T\mathbf{X}+\lambda_2\mathbf{I}}{1+\lambda_2} = (1-\gamma)\widehat{\Sigma} + \gamma\mathbf{I}$ with $\gamma = \frac{\lambda_2}{1+\lambda_2}$ shrinks $\widehat{\Sigma}$ toward the identity matrix. Together (2.17) and (2.18) say that rescaling after the elastic net penalization is mathematically equivalent to replacing $\widehat{\Sigma}$ with its shrunk version in the lasso. In linear discriminant analysis, prediction accuracy can often be improved by replacing $\widehat{\Sigma}$ by a shrunken estimate (Friedman 1989, Hastie, Tibshirani & Friedman 2001). Likewise we improve the lasso by regularizing $\widehat{\Sigma}$ in (2.18).

2.3.3 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 2.3, $\hat{\boldsymbol{\beta}} \to \hat{\boldsymbol{\beta}}(\infty)$ as $\lambda_2 \to \infty$, where

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

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

$$\hat{\boldsymbol{\beta}}(\infty)_i = \left(\left|\mathbf{y}^T\mathbf{x}_i\right| - \frac{\lambda_1}{2} \right)_+ \text{sgn}\left(\mathbf{y}^T\mathbf{x}_i\right), \quad i = 1, 2, \ldots, p. \tag{2.19}$$

Observe that $\mathbf{y}^T\mathbf{x}_i$ is the univariate regression coefficient of the i-th predictor, $\hat{\boldsymbol{\beta}}(\infty)$ are the estimates by applying soft-thresholding on univariate regression coefficients, thus (2.19) is called univariate soft-thresholding (UST).

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, Tibshirani & Chu 2001) and the nearest shrunken centroids (NSC) classifier (Tibshirani, Hastie, Narasimhan & Chu 2002), and have shown good empirical performance. The elastic net naturally bridges the lasso and UST.

2.3.4 Computation: the LARS-EN algorithm

We propose an efficient algorithm called LARS-EN to efficiently solve the elastic net, which is based on the recently proposed LARS algorithm of Efron, Hastie, Johnstone & Tibshirani (2004) (referred to as the LAR paper henceforth). In the LAR paper, the authors proved that starting from zero, the lasso solution paths grow piecewise linearly in a predictable way. They proposed a new algorithm called LARS to efficiently solve the entire lasso solution path using the same order of computations as a single OLS fit. The piecewise linearity of the lasso solution path was first proved by Osborne, Presnell & Turlach (2000), and they also described an efficient algorithm that calculates the complete lasso solution path. By Lemma 2.1, for each fixed λ_2 the elastic net problem is equivalent to a lasso problem on the augmented data set. So the LARS algorithm can be directly used to efficiently create the *entire elastic net solution path* 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 things down a lot.

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 the LAR 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 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 the LARS algorithm. It is also economical to 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. In the $p \gg n$ case, such as with 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 $O(m^3 + pm^2)$ operations.

2.3.5 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 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 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 is always valued within $[0, 1]$. In the LARS algorithm the lasso is described as a forward stage-wise additive fitting procedure and shown to be (almost) identical to ϵ-L_2 boosting (Efron, Hastie, Johnstone & Tibshirani 2004). This new view adopts the number of steps k of the LARS algorithm as a tuning parameter for the lasso. For each fixed λ_2, the elastic net is solved by the LARS-EN algorithm, 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 piece-wise elastic net/lasso solution paths as shown in Figure 2.3.

There are well-established methods for choosing such tuning parameters (Hastie, Tibshirani & Friedman 2001, Chapter 7). If only training data are available, 10-fold cross-validation 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 2-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, the LARS-EN algorithm produces the entire solution path of the elastic net. The other tuning parameter (λ_1, s or k) is selected by 10-fold CV. The chosen λ_2 is the one giving the smallest CV error.

For each λ_2, the computational cost of 10-fold CV is the same as ten OLS fits. Thus the 2-D 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 $n = 30$ and $p = 5000$, if we do not want more than 200 variables in the final model, we may stop the LARS-EN algorithm after 500 steps and only consider the best k within 500.

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

2.4 Prostate Cancer Example.

The data in this example come from a study of prostate cancer (Stamey, Kabalin, Mcneal, Johnstone, Redwine & Yang 1989). 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).

OLS, ridge regression, the lasso, the naive 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 10-fold cross-validation 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 2.4 clearly shows the elastic net as 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 and fails to do variable selection. The lasso

Table 2.1: *Prostate cancer data: comparing different methods*

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)

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. We also see in this case that the elastic net is actually UST, because the selected λ is very big (1000). This can be considered as a piece of empirical evidence supporting UST. Figure 2.3 displays the lasso and the elastic net solution paths.

If we check the correlation matrix of these eight predictors, we see there are a number of 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 improves on OLS, the elastic net will improve the lasso. We demonstrate this point by simulations in the next section.

2.5 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

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

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

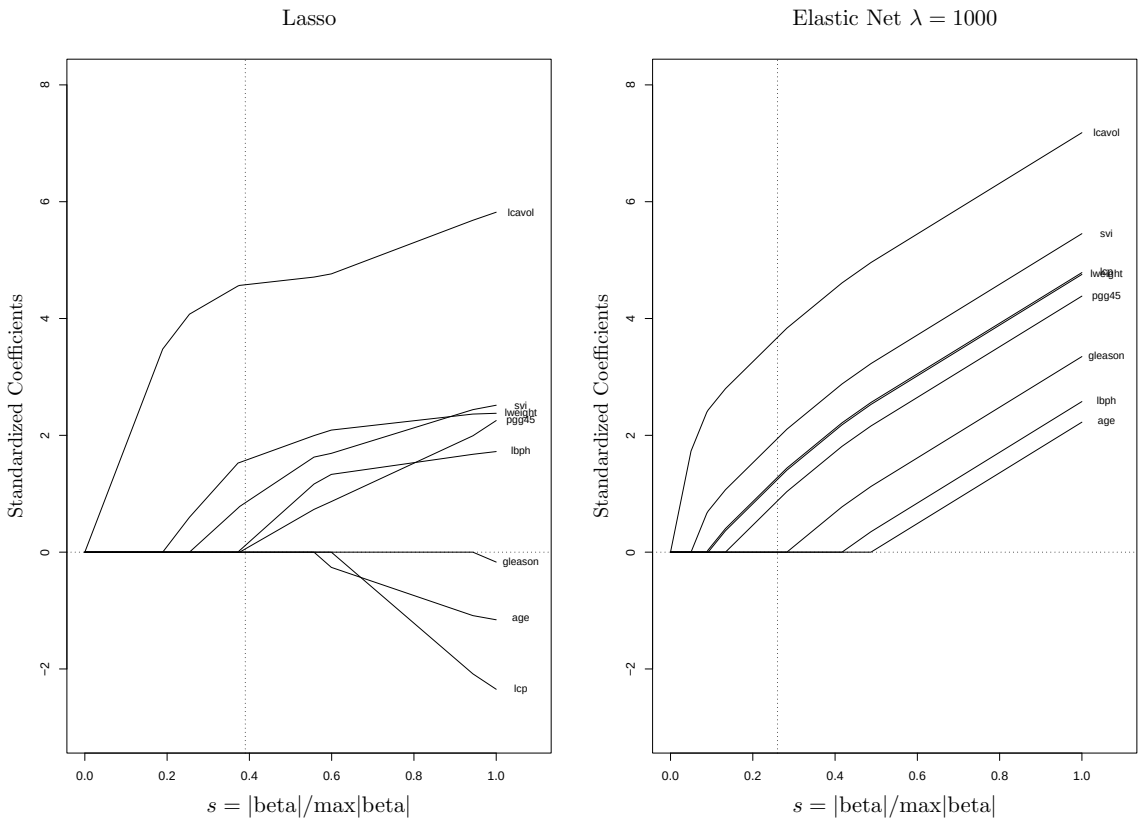

Figure 2.3: *The left panel shows the lasso estimates as a function of s, and the right panel shows the elastic net estimates as a function of s. Both of them are piecewise linear, which is a key property of our efficient algorithm. The solution paths also show the elastic net is identical to univariate soft-thresholding in this example. In both plots the vertical dotted line indicates the selected final model.*

Within each example, our simulated data consists 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 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.

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 $\text{cor}(i, j) = (0.5)^{|i-j|}$.

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 $\sigma = 15$; $\text{cor}(i, j) = 0.5$ 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:

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

where ϵ_i^x are independent identically distributed (i.i.d.) $N(0, 0.01)$, $i = 1, \ldots, 15$. 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 only select the 15 true features and set the coefficients of the 25 noise features to 0.

Table 2.2: *Median MSE for the simulated examples and four methods based on 50 replications. The numbers in parentheses are the corresponding standard errors (of the medians) estimated using the bootstrap with $B = 500$ resamplings on the fifty MSEs.*

Method	*Ex.1*	*Ex.2*	*Ex.3*	*Ex.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	4.49 (0.46)	2.84 (0.27)	39.5 (1.80)	64.5 (4.78)
Naive elastic net	5.70 (0.41)	2.73 (0.23)	41.0 (2.13)	45.9 (3.72)

Table 2.3: *Median 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

Table 2.5 and Figure 2.4 (Box-plots) summarize the prediction results. First we see that the naive elastic net either has very poor performance (in example 1) or behaves almost identical to either ridge regression (in example 2 and 3) or the lasso (in example 4). In all examples, the elastic net is significantly more accurate than the lasso, even when the lasso is doing much better than ridge. The reductions of the prediction error in the four examples are 18%, 18%, 13% and 27%, respectively. The simulation results indicate that the elastic net dominates the lasso under collinearity.

Table 2.5 shows that the elastic net produces sparse solutions. The elastic net tends to select more variables than the lasso does, due 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 $N(Z_1 + 0.1 \cdot Z_2, 1)$. Suppose we only observe

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

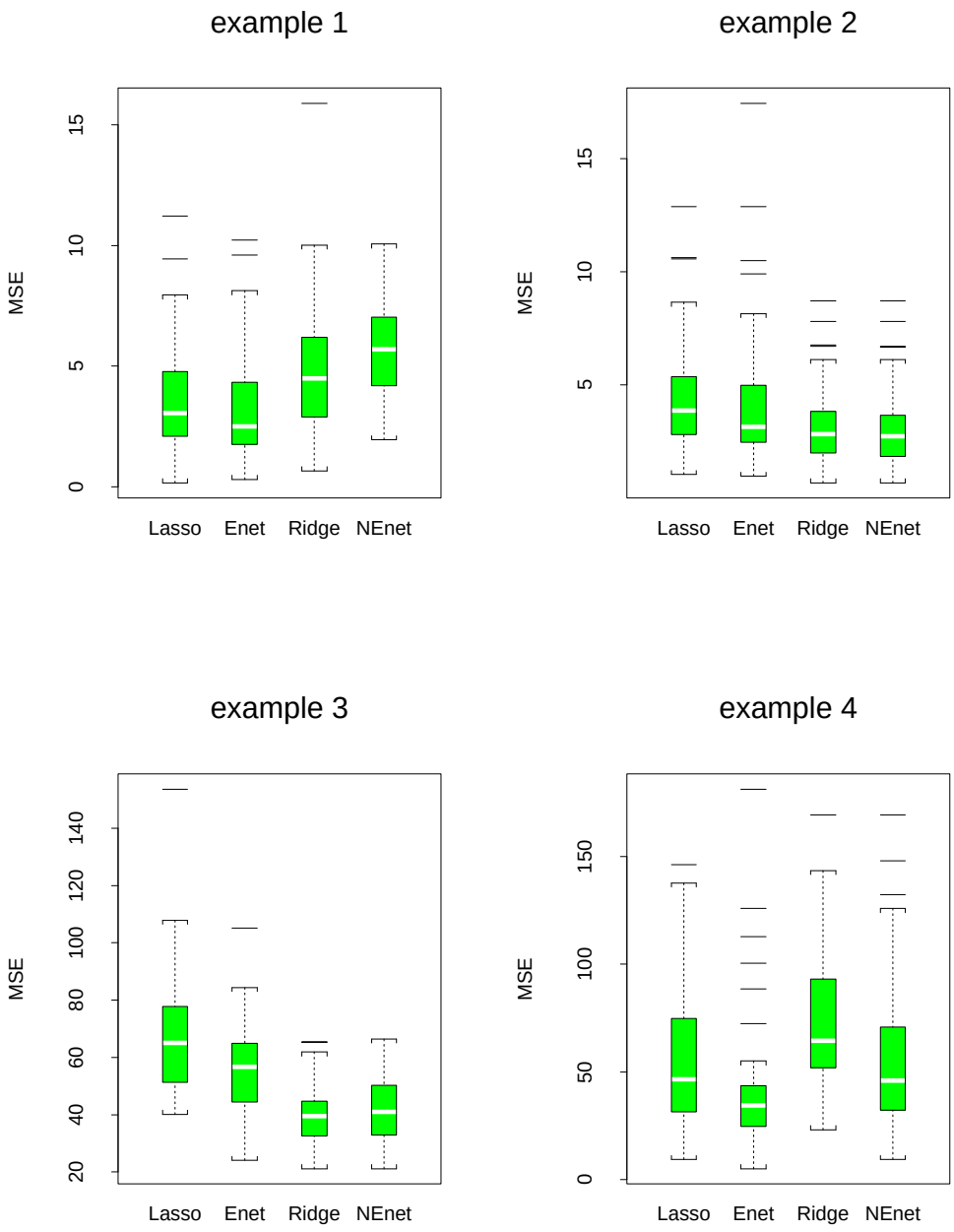

Figure 2.4: *Comparing prediction accuracy of the lasso, the elastic net (Enet), ridge and the naive elastic net (NEnet). The elastic net outperforms the lasso in all four examples.*

where ϵ_i are i.i.d. $N(0, \frac{1}{16})$. One hundred observations were generated from this model. The variables $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ form a group whose underlying factor is Z_1, and $\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6$ form 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. Figure 2.5 compares the solution paths of the lasso and the elastic net.

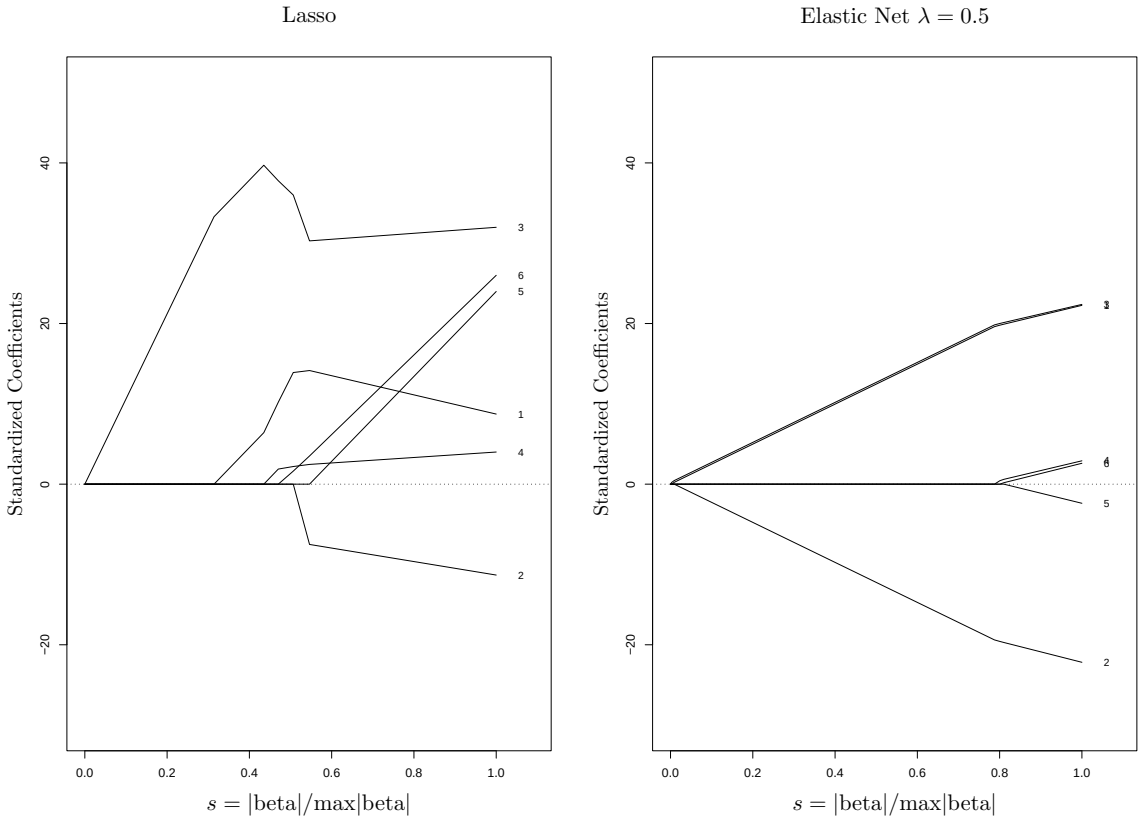

Figure 2.5: *The left and right panel show the lasso and the elastic net ($\lambda_2 = 0.5$) solution paths respectively. The lasso paths are unstable and the plot does not reveal any correlation information by itself. In contrast, the elastic net has much smoother solution paths, while clearly showing 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.*

2.6 Microarray Classification and Gene Selection

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 classification method should have the following properties:

1. Gene selection should be *built into* the procedure.
2. It should not be limited by the fact that $p \gg n$.
3. For those genes sharing the same biological "pathway", it should be able to automatically include whole groups into the model once one gene 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 (1) but fails both (2) and (3). The support vector machine (SVM) (Guyon, Weston, Barnhill & Vapnik 2002) and penalized logistic regression (PLR) (Zhu & Hastie 2004) are very successful classifiers, but they cannot do gene selection automatically and both use either univariate ranking (UR) (Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loh, Downing & Caligiuri 1999) or recursive feature elimination (RFE) (Guyon, Weston, Barnhill & Vapnik 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 leukemia data to illustrate the elastic net classifier.

The leukemia data consist of 7129 genes and 72 samples (Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loh, Downing & Caligiuri 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 a 0-1 response y. The classification function is $I(\text{fitted value} > 0.5)$,

Table 2.4: *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	3/38	0/34	45

where $I(\cdot)$ is the indicator function. We used 10-fold cross-validation to select the tuning parameters.

We used pre-screening to make the computation more manageable. Each time a model is fit, we first select the 1000 most "significant" genes as the predictors, according to their t-statistic scores (Tibshirani, Hastie, Narasimhan & Chu 2002). Note that this screening is done separately in each training fold in the cross-validation. In practice, this screening does not affect the results, because we stop the elastic net path relatively early, at a stage when the screened variables are unlikely to be in the model.

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

We stopped the LARS-EN algorithm after 200 steps. As can be seen from Figure 2.6, using the number of steps k in the LARS-EN algorithm as the tuning parameter, the elastic net classifier ($\lambda = 0.01, k = 82$) gives 10-fold CV error 3/38 and the test error 0/34 with 45 genes selected. Figure 2.7 displays the elastic net solution paths and the gene selection results. Table 2.6 compares the elastic net with several competitors including Golubs' method, the support vector machine (SVM), penalized logistic regression (PLR), nearest shrunken centroid (NSC) (Tibshirani, Hastie, Narasimhan & Chu 2002). The elastic net gives the best classification, and it has an *internal* gene selection facility.

2.7 Summary

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.

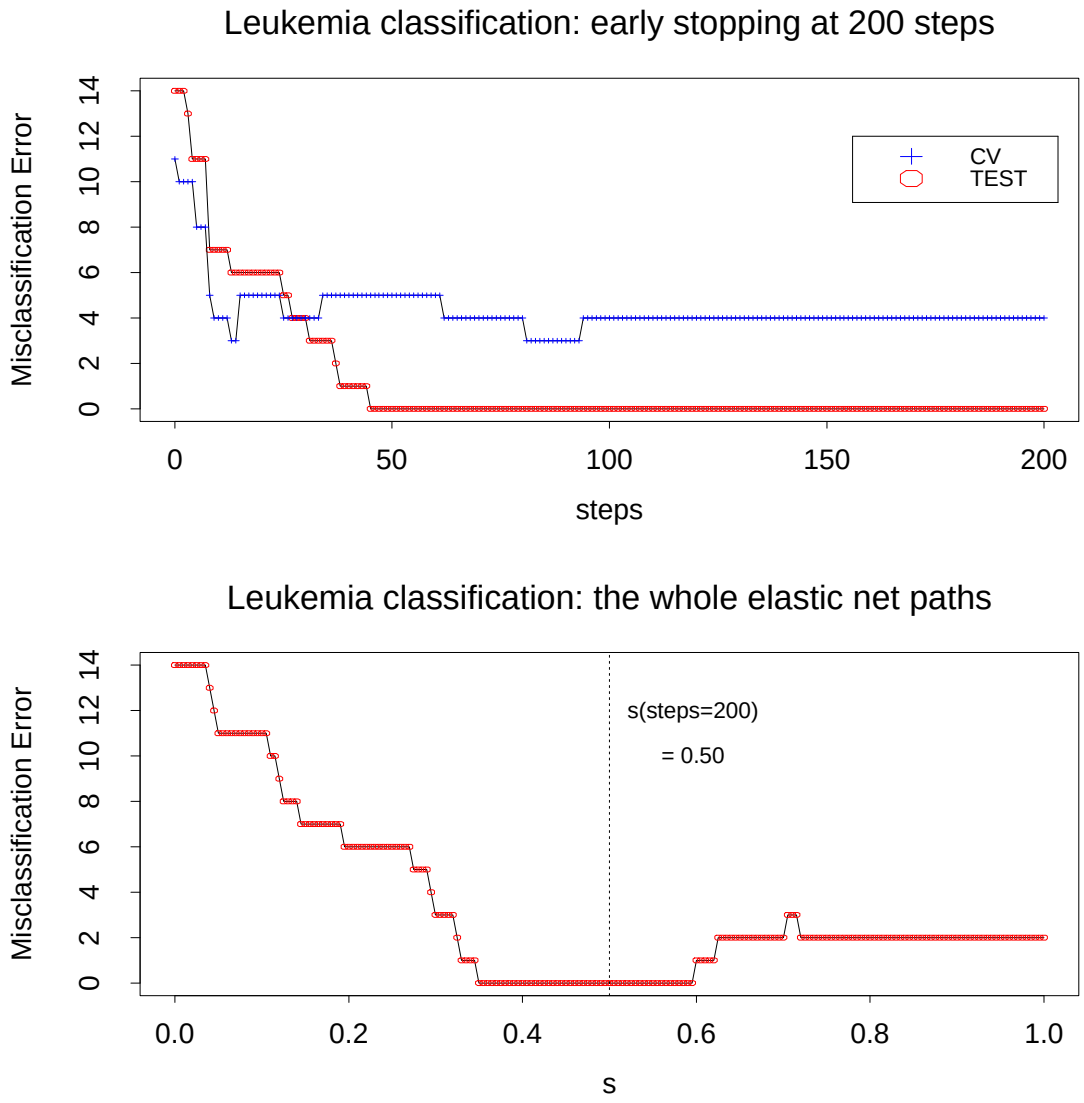

Figure 2.6: *Leukemia classification and gene selection by the elastic net($\lambda = 0.01$). The early stopping strategy (upper plot) finds the optimal classifier with much less computational cost than the lower. With early stopping, the number of steps is much more convenient than s, the fraction of L_1 norm, since computing s depends on the fit at the last step of the LARS-EN algorithm. The actual values of s are not available in 10-fold cross-validation if the LARS-EN algorithm is stopped early. On the training set, steps=200 is equivalent to $s = 0.50$, indicated by the broken vertical line in the lower plot.*

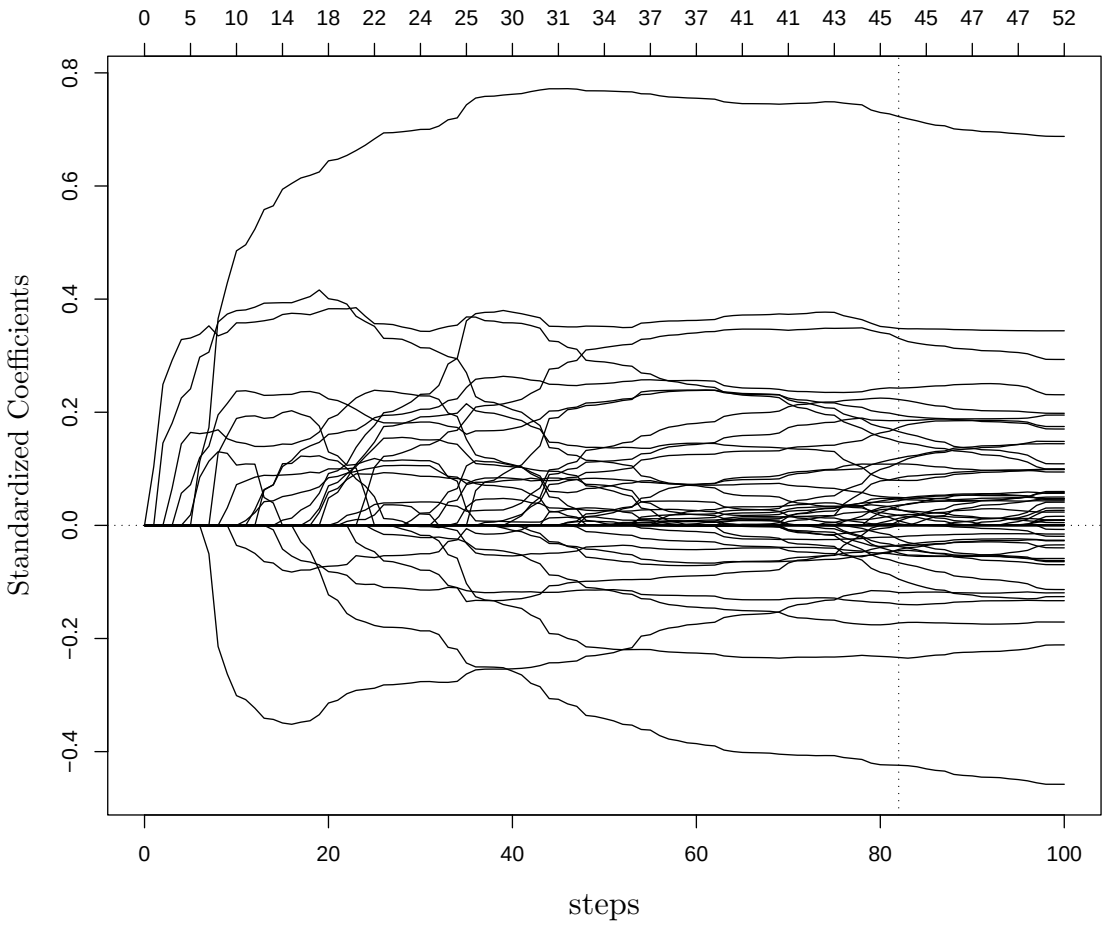

Figure 2.7: *Leukemia data: the elastic net coefficients paths (up to $k = 100$). The labels on the top indicate the number of nonzero coefficients (selected genes) at each step. The optimal elastic net model is given by the fit at step eighty-two with 45 selected genes. Note that the size of training 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 10-fold CV. If a bigger λ is used, the grouping effect will be stronger.*

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 misclassification error, and it does automatic gene selection.

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, Tibshirani & Friedman 2001, Friedman, Hastie, Rosset, Tibshirani & Zhu 2004). Our results offer other insights into the lasso, and ways to improve it.

2.8 Proofs of Lemma 2.2 and Theorems 2.1-2.3

Proof of Lemma 2.2. Part (1): Fix $\lambda > 0$. If $\hat{\beta}_i \neq \hat{\beta}_j$, let us consider $\hat{\boldsymbol{\beta}}^*$ as follows

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

Because $\mathbf{x}_i = \mathbf{x}_j$, it is obvious that $\mathbf{X}\hat{\boldsymbol{\beta}}^* = \mathbf{X}\hat{\boldsymbol{\beta}}$, 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 $J\left(\hat{\boldsymbol{\beta}}^*\right) < J\left(\hat{\boldsymbol{\beta}}\right)$. Therefore $\hat{\boldsymbol{\beta}}$ cannot be the minimizer of (2.7), a contradiction. So we must have $\hat{\beta}_i = \hat{\beta}_j$.

Part (2): If $\hat{\beta}_i \hat{\beta}_j < 0$, consider the same $\hat{\boldsymbol{\beta}}^*$ again. We see $\|\hat{\boldsymbol{\beta}}^*\|_1 < \|\hat{\boldsymbol{\beta}}\|_1$, so $\hat{\boldsymbol{\beta}}$ cannot be a lasso solution. The rest can be directly verified by the definition of the lasso, thus omitted. □

Proof of Theorem 2.1. 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, we have $\operatorname{sgn}\left(\hat{\beta}_i(\lambda_1, \lambda_2)\right) = \operatorname{sgn}\left(\hat{\beta}_j(\lambda_1, \lambda_2)\right)$. Because of (2.4), $\hat{\boldsymbol{\beta}}(\lambda_1, \lambda_2)$ satisfies

$$\left.\frac{\partial L(\lambda_1, \lambda_2, \boldsymbol{\beta})}{\partial \beta_k}\right|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}(\lambda_1,\lambda_2)} = 0 \quad \text{if } \hat{\beta}_k(\lambda_1, \lambda_2) \neq 0. \tag{2.20}$$

Hence we have

$$-2\mathbf{x}_i^T\left(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}\left(\lambda_1,\lambda_2\right)\right)+\lambda_1\text{sgn}\left(\hat{\beta}_i\left(\lambda_1,\lambda_2\right)\right)+2\lambda_2\hat{\beta}_i\left(\lambda_1,\lambda_2\right)=0, \tag{2.21}$$

$$-2\mathbf{x}_j^T\left(\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}\left(\lambda_1,\lambda_2\right)\right)+\lambda_1\text{sgn}\left(\hat{\beta}_j\left(\lambda_1,\lambda_2\right)\right)+2\lambda_2\hat{\beta}_j\left(\lambda_1,\lambda_2\right)=0. \tag{2.22}$$

Subtracting (2.21) from (2.22) gives

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

which is equivalent to

$$\hat{\beta}_i\left(\lambda_1,\lambda_2\right)-\hat{\beta}_j\left(\lambda_1,\lambda_2\right)=\frac{1}{\lambda_2}(\mathbf{x}_i^T-\mathbf{x}_j^T)\hat{r}\left(\lambda_1,\lambda_2\right), \tag{2.23}$$

where $\hat{r}\left(\lambda_1,\lambda_2\right)=\mathbf{y}-\mathbf{x}\hat{\boldsymbol{\beta}}\left(\lambda_1,\lambda_2\right)$ is the residual vector. Since $\mathbf{X}$ are standardized, $\|\mathbf{x}_i-\mathbf{x}_j\|^2=2(1-\rho)$ where $\rho=\mathbf{x}_i^T\mathbf{x}_j$. By (2.4) we must have

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

So $\|\hat{r}\left(\lambda_1,\lambda_2\right)\|\leq\|\mathbf{y}\|$. Then (2.23) implies

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

□

Proof of Theorem 2.2. We denote $C(\alpha,\boldsymbol{\beta})=\sum_{k=1}^n\phi\left(y_k(\alpha+\mathbf{x}_k^T\boldsymbol{\beta})\right)+\lambda_2\|\boldsymbol{\beta}\|^2+\lambda_1\|\boldsymbol{\beta}\|_1$. Consider another estimates with $\hat{\alpha}^*=\hat{\alpha}$ and

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

By definition, $C(\hat{\alpha}^*, \hat{\boldsymbol{\beta}}^*) \geq C(\hat{\alpha}, \hat{\boldsymbol{\beta}})$. Hence we have

$$\sum_{k=1}^{n} \left[\phi\left(y_k, (\hat{\alpha}^* + \mathbf{x}_k^T \hat{\boldsymbol{\beta}}^*)\right) - \phi\left(y_k, (\hat{\alpha} + \mathbf{x}_k^T \hat{\boldsymbol{\beta}})\right)\right] + \lambda_2 \left[\|\hat{\boldsymbol{\beta}}^*\|^2 - \|\hat{\boldsymbol{\beta}}\|^2\right] + \lambda_1 \left[\|\hat{\boldsymbol{\beta}}^*\|_1 - \|\hat{\boldsymbol{\beta}}\|_1\right] \geq 0 \tag{2.24}$$

On the other hand, we know that

$$\begin{aligned} \|\hat{\boldsymbol{\beta}}^*\|_1 - \|\hat{\boldsymbol{\beta}}\|_1 &= |\hat{\beta}_i^*| + |\hat{\beta}_j^*| - |\hat{\beta}_i| - |\hat{\beta}_j| \\ &= |\hat{\beta}_i + \hat{\beta}_j| - |\hat{\beta}_i| - |\hat{\beta}_j| \\ &\leq 0 \end{aligned} \tag{2.25}$$

$$\begin{aligned} \|\hat{\boldsymbol{\beta}}^*\|^2 - \|\hat{\boldsymbol{\beta}}\|^2 &= |\hat{\beta}_i^*|^2 + |\hat{\beta}_j^*|^2 - |\hat{\beta}_i|^2 - |\hat{\beta}_j|^2 \\ &= -\frac{|\hat{\beta}_i - \hat{\beta}_j|^2}{2} \end{aligned} \tag{2.26}$$

$$\begin{aligned} & \sum_{k=1}^{n} \left[\phi\left(y_k, (\hat{\alpha}^* + \mathbf{x}_k^T \hat{\boldsymbol{\beta}}^*)\right) - \phi\left(y_k, (\hat{\alpha} + \mathbf{x}_k^T \hat{\boldsymbol{\beta}})\right)\right] \\ \leq & \sum_{k=1}^{n} \left|\phi\left(y_k, (\hat{\alpha}^* + \mathbf{x}_k^T \hat{\boldsymbol{\beta}}^*)\right) - \phi\left(y_k, (\hat{\alpha} + \mathbf{x}_k^T \hat{\boldsymbol{\beta}})\right)\right| \\ \leq & \sum_{k=1}^{n} M \left|(\hat{\alpha}^* + \mathbf{x}_k^T \hat{\boldsymbol{\beta}}^*) - (\hat{\alpha} + \mathbf{x}_k^T \hat{\boldsymbol{\beta}})\right| \qquad (2.27) \\ = & \sum_{k=1}^{n} M \left|\mathbf{x}_k^T (\hat{\boldsymbol{\beta}}^* - \hat{\boldsymbol{\beta}})\right| \\ = & \sum_{k=1}^{n} M \left|(\mathbf{x}_{k,i} - \mathbf{x}_{k,j}) \frac{1}{2} (\hat{\beta}_i - \hat{\beta}_j)\right| \\ = & \frac{M}{2} \|\mathbf{x}_i - \mathbf{x}_j\|_1 |\hat{\beta}_i - \hat{\beta}_j| \qquad (2.28) \end{aligned}$$

where inequality (2.27) uses the Lipschitz assumption. Together (2.24), (2.25), (2.26) and

(2.28) imply that

$$0 \leq \frac{M}{2}\|\mathbf{x}_i - \mathbf{x}_j\|_1|\hat{\beta}_i - \hat{\beta}_j| - \frac{\lambda}{2}|\hat{\beta}_i^* - \hat{\beta}_j^*|^2 \tag{2.29}$$

Thus the first inequality in Theorem 2.2 is obtained by rearranging (2.29). For the second inequality, we simply use the inequality $\|\mathbf{x}_i - \mathbf{x}_j\|_1 \leq \sqrt{n}\sqrt{\|\mathbf{x}_i - \mathbf{x}_j\|^2} = \sqrt{n}\sqrt{2(1-\rho)}$.

□

Proof of Theorem 2.3. Let $\hat{\boldsymbol{\beta}}$ be the elastic net estimates. By definition and (2.13) we have

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \|\mathbf{y}^* - \mathbf{X}^* \frac{\boldsymbol{\beta}}{\sqrt{1+\lambda_2}}\|^2 + \frac{\lambda_1}{\sqrt{1+\lambda_2}}\|\frac{\boldsymbol{\beta}}{\sqrt{1+\lambda_2}}\|_1 \\ &= \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^T \left(\frac{\mathbf{X}^{*T}\mathbf{X}^*}{1+\lambda_2}\right)\boldsymbol{\beta} - 2\frac{\mathbf{y}^{*T}X^*}{\sqrt{1+\lambda_2}} + \mathbf{y}^{*T}\mathbf{y}^* + \frac{\lambda_1 |\boldsymbol{\beta}|_1}{1+\lambda_2}. \end{aligned} \tag{2.30}$$

Substituting the identities

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

into (2.30), we have

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

□

Chapter 3

Sparse Principal Component Analysis

Principal component analysis (PCA) is widely used in data processing and dimensionality reduction. However, PCA suffers from the fact that each principal component is a linear combination of all the original variables, thus it is often difficult to interpret the results. In this Chapter we introduce a new method called sparse principal component analysis (SPCA) using the *lasso* (*elastic net*) to produce modified principal components with sparse loadings. We first show that PCA can be formulated as a regression-type optimization problem; sparse loadings are then obtained by imposing the lasso (elastic net) constraint on the regression coefficients. Efficient algorithms are proposed to fit our SPCA models for both regular multivariate data and gene expression arrays. We also give a new formula to compute the total variance of modified principal components. As illustrations, SPCA is applied to real and simulated data with encouraging results.

3.1 Background

Principal component analysis (PCA) (Jolliffe 1986) is a popular data processing and dimension reduction technique, with numerous applications in engineering, biological and social

science. Some interesting examples include handwritten zip code classification (Hastie, Tibshirani & Friedman 2001) and human face recognition (Hancock, Burton & Bruce 1996). Recently PCA has been used in gene expression data analysis (Alter, Brown & Botstein 2000). Hastie, Tibshirani, Eisen, Brown, Ross, Scherf, Weinstein, Alizadeh, Staudt & Botstein (2000) proposed the so-called *Gene Shaving* techniques using PCA to cluster highly variable and coherent genes in microarray data sets.

PCA seeks the linear combinations of the original variables such that the derived variables capture maximal variance. PCA can be computed via the singular value decomposition (SVD) of the data matrix. In detail, let the data $\mathbf{X}$ be a $n \times p$ matrix, where n and p are the number of observations and the number of variables, respectively. Without loss of generality, assume the column means of $\mathbf{X}$ are all 0. Let the SVD of $\mathbf{X}$ be

$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T. \tag{3.1}$$

$\mathbf{Z} = \mathbf{U}\mathbf{D}$ are the principal components (PCs), and the columns of $\mathbf{V}$ are the corresponding loadings of the principal components. The sample variance of the i-th PC is $\mathbf{D}_{ii}^2/n$. In gene expression data the standardized PCs $\mathbf{U}$ are called the *eigen-arrays* and $\mathbf{V}$ are the *eigen-genes* (Alter, Brown & Botstein 2000). Usually the first q ($q \ll \min(n, p)$) PCs are chosen to represent the data, thus a great dimensionality reduction is achieved.

The success of PCA is due to the following two important optimal properties:

1. principal components sequentially capture the maximum variability among the columns of $\mathbf{X}$, thus guaranteeing minimal information loss;

2. principal components are uncorrelated, so we can talk about one principal component without referring to others.

However, PCA also has an obvious drawback, i.e., each PC is a linear combination of all p variables and the loadings are typically nonzero. This makes it often difficult to interpret the derived PCs. Rotation techniques are commonly used to help practitioners to interpret principal components (Jolliffe 1995). Vines (2000) considered simple principal components

by restricting the loadings to take values from a small set of allowable integers such as 0, 1 and -1.

We feel it is desirable not only to achieve the dimensionality reduction but also to reduce the number of explicitly used variables. An ad hoc way to achieve this is to artificially set the loadings with absolute values smaller than a threshold to zero. This informal thresholding approach is frequently used in practice, but can be potentially misleading in various respects (Cadima & Jolliffe 1995). McCabe (1984) presented an alternative to PCA which found a subset of *principal variables*. Jolliffe, Trendafilov & Uddin (2003) introduced SCoTLASS to get modified principal components with possible zero loadings.

The same interpretation issues arise in multiple linear regression, where the response is predicted by a linear combination of the predictors. Interpretable models are obtained via variable selection. The *lasso* (Tibshirani 1996) is a promising variable selection technique, simultaneously producing accurate and sparse models. The *elastic net* introduced in Chapter 2 has some advantages over the lasso. In this Chapter we introduce a new approach for estimating PCs with sparse loadings, which we call sparse principal component analysis (SPCA). SPCA is built on the fact that PCA can be written as a regression-type optimization problem, with a quadratic penalty; the lasso penalty (via the elastic net) can then be directly integrated into the regression criterion, leading to a modified PCA with sparse loadings.

The methodological details of SPCA are presented in Section 3.2. We present an efficient algorithm for fitting the SPCA model. We also derive an appropriate expression for representing the variance explained by modified principal components. In Section 3.3 we consider a special case of the SPCA algorithm for handling gene expression arrays efficiently. The proposed methodology is illustrated by using real data and simulation examples in Section 3.4. Section 3.6 contains proofs of Theorems supporting the SPCA methodology.

3.2 Motivation and Details of SPCA

In both lasso and elastic net, the sparse coefficients are a direct consequence of the L_1 penalty, and do not depend on the squared error loss function. Jolliffe, Trendafilov & Uddin (2003) proposed SCoTLASS, an interesting procedure that obtains sparse loadings by directly imposing an L_1 constraint on PCA. SCoTLASS successively maximizes the variance

$$a_k^T(\mathbf{X}^T\mathbf{X})a_k \tag{3.2}$$

subject to

$$a_k^T a_k = 1 \quad \text{and (for} \quad k \geq 2) \quad a_h^T a_k = 0, \quad h < k; \tag{3.3}$$

and the extra constraints

$$\sum_{j=1}^{p} |a_{kj}| \leq t \tag{3.4}$$

for some tuning parameter t. Although sufficiently small t yields some exact zero loadings, there is not much guidance with SCoTLASS in choosing an appropriate value for t. One could try several t values, but the high computational cost of SCoTLASS makes this an impractical solution. This high computational cost is probably due to the fact that SCoTLASS is not a convex optimization problem. Moreover, the examples in Jolliffe, Trendafilov & Uddin (2003) show that the loadings obtained by SCoTLASS are not sparse enough when one requires a high percentage of explained variance.

We consider a different approach to modifying PCA. We first show how PCA can be recast exactly in terms of (ridge) regression problem. We then introduce the lasso penalty by changing this ridge regression to an elastic-net regression.

3.2.1 Direct sparse approximations

We first discuss a simple regression approach to PCA. Observe that each PC is a linear combination of the p variables, thus its loadings can be recovered by regressing the PC on the p variables.

Theorem 3.1. *For each i, denote by $Z_i = \mathbf{U}_i\mathbf{D}_{ii}$ the i-th principal component. Consider a positive λ and the ridge estimates $\hat{\beta}_{\text{ridge}}$ given by*

$$\hat{\beta}_{\text{ridge}} = \arg\min_{\beta} \|Z_i - \mathbf{X}\beta\|^2 + \lambda\|\beta\|^2. \tag{3.5}$$

Let $\hat{v} = \frac{\hat{\beta}_{\text{ridge}}}{\|\hat{\beta}_{\text{ridge}}\|}$, then $\hat{v} = V_i$.

The theme of this simple theorem is to show the connection between PCA and a regression method. Regressing PCs on variables was discussed in Cadima & Jolliffe (1995), where they focused on approximating PCs by a subset of k variables. We extend it to a more general case of ridge regression in order to handle all kinds of data, especially gene expression data. Obviously, when $n > p$ and $\mathbf{X}$ is a full rank matrix, the theorem does not require a positive λ. Note that if $p > n$ and $\lambda = 0$, ordinary multiple regression has no unique solution that is exactly V_i. The same happens when $n > p$ and $\mathbf{X}$ is not a full rank matrix. However, PCA always gives a unique solution in all situations. As shown in Theorem 3.1, this indeterminancy is eliminated by the positive ridge penalty ($\lambda\|\beta\|^2$). Note that after normalization the coefficients are independent of λ, therefore the ridge penalty is not used to penalize the regression coefficients but to ensure the reconstruction of principal components.

Now let us add the L_1 penalty to (3.5) and consider the following optimization problem

$$\hat{\beta} = \arg\min_{\beta} \|Z_i - \mathbf{X}\beta\|^2 + \lambda\|\beta\|^2 + \lambda_1\|\beta\|_1, \tag{3.6}$$

where $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$ is the 1-norm of β. We call $\widehat{V}_i = \frac{\hat{\beta}}{\|\hat{\beta}\|}$ an approximation to V_i, and $\mathbf{X}\widehat{V}_i$ the i-th approximated principal component. In Chapter 2 we call (3.6) *naive* elastic net which differs from the elastic net by a scaling factor $(1 + \lambda)$. Since we are using the normalized fitted coefficients, the scaling factor does not affect $\widehat{V}_i$. Clearly, large enough λ_1 gives a sparse $\hat{\beta}$, hence a sparse $\widehat{V}_i$. Given a fixed λ, (3.6) is efficiently solved for all λ_1 by using the LARS-EN algorithm. Thus we can flexibly choose a sparse approximation to the i-th principal component.

3.2.2 Sparse principal components based on the SPCA criterion

Theorem 1 depends on the results of PCA, so it is not a *genuine* alternative. However, it can be used in a two-stage exploratory analysis: first perform PCA, then use (3.6) to find suitable sparse approximations.

We now present a "self-contained" regression-type criterion to derive PCs. Let $\mathbf{x}_i$ denote the i-th row vector of the matrix $\mathbf{X}$. We first consider the leading principal component.

Theorem 3.2. *For any $\lambda > 0$, let*

$$\begin{aligned}(\hat{\alpha}, \hat{\beta}) &= \arg\min_{\alpha,\beta} \sum_{i=1}^{n} \|\mathbf{x}_i - \alpha\beta^T \mathbf{x}_i\|^2 + \lambda\|\beta\|^2 \\ &\quad \textit{subject to} \quad \|\alpha\|^2 = 1.\end{aligned} \tag{3.7}$$

Then $\hat{\beta} \propto V_1$.

The next theorem extends Theorem 3.2 to derive the whole sequence of PCs.

Theorem 3.3. *Suppose we are considering the first k principal components. Let $\mathbf{A}_{p\times k} = [\alpha_1, \cdots, \alpha_k]$ and $\mathbf{B}_{p\times k} = [\beta_1, \cdots, \beta_k]$. For any $\lambda > 0$, let*

$$\begin{aligned}(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) &= \arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T \mathbf{x}_i\|^2 + \lambda \sum_{j=1}^{k} \|\beta_j\|^2 \\ &\quad \textit{subject to} \quad \mathbf{A}^T\mathbf{A} = I_{k\times k}.\end{aligned} \tag{3.8}$$

Then $\hat{\beta}_j \propto V_j$ for $j = 1, 2, \ldots, k$.

Theorems 3.2 and 3.3 effectively transform the PCA problem to a regression-type problem. The critical element is the objective function $\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2$. If we restrict $\mathbf{B} = \mathbf{A}$, then

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{A}^T\mathbf{x}_i\|^2,$$

whose minimizer under the orthonormal constraint on $\mathbf{A}$ is exactly the first k loading vectors of ordinary PCA. This formulation arises in the "closest approximating linear manifold"

derivation of PCA (Hastie, Tibshirani & Friedman 2001, for example). Theorem 3.3 shows that we can still have exact PCA while relaxing the restriction $\mathbf{B} = \mathbf{A}$ and adding the ridge penalty term. As can be seen later, these generalizations enable us to flexibly modify PCA.

The proofs of Theorems 3.2 and 3.3 are given in the appendix; here we give an intuitive explanation. Note that

$$\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 = \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2 \tag{3.9}$$

Since $\mathbf{A}$ is orthonomal, let $\mathbf{A}_\perp$ be any orthonormal matrix such that $[\mathbf{A}; \mathbf{A}_\perp]$ is $p \times p$ orthonormal. Then we have

$$\begin{aligned} \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2 &= \|\mathbf{X}\mathbf{A}_\perp\|^2 + \|\mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\|^2 & (3.10) \\ &= \|\mathbf{X}\mathbf{A}_\perp\|^2 + \sum_{j=1}^{k} \|\mathbf{X}\alpha_j - \mathbf{X}\beta_j\|^2 & (3.11) \end{aligned}$$

Suppose $\mathbf{A}$ is given, then the optimal $\mathbf{B}$ minimizing (3.8) should minimize

$$\arg\min_{\mathbf{B}} \sum_{j=1}^{k} \{\|\mathbf{X}\alpha_j - \mathbf{X}\beta_j\|^2 + \lambda\|\beta_j\|^2\} \tag{3.12}$$

which is equivalent to k independent ridge regression problems. In particular, if $\mathbf{A}$ corresponds to the ordinary PCs, i.e., $\mathbf{A} = \mathbf{V}$, then by Theorem 1, we know that $\mathbf{B}$ should be proportional to $\mathbf{V}$. Actually, the above view points out an effective algorithm for solving (3.8), which is revisited in the next section.

We carry on the connection between PCA and regression, and use the lasso approach to produce sparse loadings ("regression coefficients"). For that purpose, we add the lasso

penalty into the criterion (3.8) and consider the following optimization problem

$$\begin{aligned}(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) &= \arg\min_{\mathbf{A},\mathbf{B}} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 + \lambda \sum_{j=1}^{k} \|\beta_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j}\|\beta_j\|_1 \\ &\text{subject to} \quad \mathbf{A}^T\mathbf{A} = I_{k\times k}.\end{aligned} \tag{3.13}$$

Whereas the same λ is used for all k components, different $\lambda_{1,j}$s are allowed for penalizing the loadings of different principal components. Again, if $p > n$, a positive λ is required in order to get exact PCA when the sparsity constraint (the lasso penalty) vanishes ($\lambda_{1,j} = 0$). We call (3.13) the SPCA criterion hereafter.

3.2.3 Numerical solution

We propose an alternating algorithm to minimize the SPCA criterion (3.13).

B given A: For each j, let $Y_j^* = \mathbf{X}\alpha_j$. By the same analysis used in (3.10)–(3.12), we know that $\hat{\mathbf{B}} = [\hat{\beta}_1, \cdots, \hat{\beta}_k]$, where each $\hat{\beta}_j$ is an elastic net estimate

$$\hat{\beta}_j = \arg\min_{\beta_j} \|Y_j^* - \mathbf{X}\beta_j\|^2 + \lambda\|\beta_j\|^2 + \lambda_{1,j}\|\beta_j\|_1. \tag{3.14}$$

A given B: On the other hand, if $\mathbf{B}$ is fixed, then we can ignore the penalty part in (3.13) and only try to minimize $\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 = \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2$, subject to $\mathbf{A}^T\mathbf{A} = \mathbf{I}_{k\times k}$. The solution is obtained by a reduced rank form of the *Procrustes rotation,* given in Theorem 3.4 below. We compute the SVD

$$(\mathbf{X}^T\mathbf{X})\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \tag{3.15}$$

and set $\widehat{\mathbf{A}} = \mathbf{U}\mathbf{V}^T$.

Theorem 3.4 (Reduced Rank Procrustes Rotation). *Let* $\mathbf{M}_{n\times p}$ *and* $\mathbf{N}_{n\times k}$ *be two*

matrices. Consider the constrained minimization problem

$$\widehat{\mathbf{A}} = \arg\min_{\mathbf{A}} \|\mathbf{M} - \mathbf{N}\mathbf{A}^T\|^2 \quad \textit{subject to} \quad \mathbf{A}^T\mathbf{A} = \mathbf{I}_{k\times k}. \tag{3.16}$$

Suppose the SVD of $\mathbf{M}^T\mathbf{N}$ *is* $\mathbf{U}\mathbf{D}\mathbf{V}^T$*, then* $\widehat{\mathbf{A}} = \mathbf{U}\mathbf{V}^T$*.*

The usual Procrustes rotation (Mardia, Kent & Bibby 1979, for example) has $\mathbf{N}$ the same size as $\mathbf{M}$.

It is worth pointing out that to solve (3.14), we only need to know the Gram matrix $\mathbf{X}^T\mathbf{X}$, because

$$\begin{aligned} &\|Y_j^* - \mathbf{X}\beta_j\|^2 + \lambda\|\beta_j\|^2 + \lambda_{1,j}\|\beta_j\|_1 \\ &= (\alpha_j - \beta_j)^T\mathbf{X}^T\mathbf{X}(\alpha_j - \beta_j) + \lambda\|\beta_j\|^2 + \lambda_{1,j}\|\beta_j\|_1 \end{aligned} \tag{3.17}$$

The same is true of (3.15).

Now $\frac{1}{n}\mathbf{X}^T\mathbf{X}$ is the sample covariance matrix of $\mathbf{X}$. Therefore if $\mathbf{\Sigma}$, the covariance matrix of $\mathbf{X}$, is known, we can replace $\mathbf{X}^T\mathbf{X}$ with $\mathbf{\Sigma}$ in (3.17) and have a population version of SPCA. If $\mathbf{X}$ is standardized beforehand, then we use the (sample) correlation matrix, which is preferred when the scales of the variables are different.

Although (3.17) (with $\mathbf{\Sigma}$ instead of $\mathbf{X}^T\mathbf{X}$) is not quite an elastic net problem, we can easily turn it into one. Create the artificial response Y^{**} and $\mathbf{X}^{**}$ as follows

$$Y^{**} = \mathbf{\Sigma}^{\frac{1}{2}}\alpha_j \quad \mathbf{X}^{**} = \mathbf{\Sigma}^{\frac{1}{2}}, \tag{3.18}$$

then it is easy to check that

$$\hat{\beta}_j = \arg\min_{\beta} \|Y^{**} - \mathbf{X}^{**}\beta\|^2 + \lambda\|\beta\|^2 + \lambda_{1,j}\|\beta\|_1. \tag{3.19}$$

Algorithm 3.2.1 summarizes the steps of our SPCA procedure outlined above.

Some remarks:

1. Empirical evidence suggests that the output of the above algorithm does not change

Algorithm 3.2.1 *General SPCA Algorithm*

1. Let $\mathbf{A}$ start at $\mathbf{V}[\,, 1:k]$, the loadings of the first k ordinary principal components.
2. Given a fixed $\mathbf{A} = [\alpha_1, \cdots, \alpha_k]$, solve the following elastic net problem for $j = 1, 2, \ldots, k$
$$\beta_j = \arg\min_{\beta} (\alpha_j - \beta)^T \mathbf{X}^T \mathbf{X} (\alpha_j - \beta) + \lambda \|\beta\|^2 + \lambda_{1,j} \|\beta\|_1$$
3. For a fixed $\mathbf{B} = [\beta_1, \cdots, \beta_k]$, compute the SVD of $\mathbf{X}^T\mathbf{X}\mathbf{B} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, then update $\mathbf{A} = \mathbf{U}\mathbf{V}^T$.
4. Repeat steps 2–3, until convergence.
5. Normalization: $\widehat{V}_j = \frac{\beta_j}{\|\beta_j\|}$, $j = 1, \ldots, k$.

much as λ is varied. For $n > p$ data, the default choice of λ can be zero. Practically λ is chosen to be a small positive number to overcome potential collinearity problems in $\mathbf{X}$. Section 3.3 discusses the default choice of λ for data with thousands of variables, such as gene expression arrays.

2. In principle, we can try several combinations of $\{\lambda_{1,j}\}$ to figure out a good choice of the tuning parameters, since the above algorithm converges quite fast. There is a shortcut provided by the direct sparse approximation (3.6). The LARS-EN algorithm efficiently delivers a whole sequence of sparse approximations for each PC and the corresponding values of $\lambda_{1,j}$. Hence we can pick a $\lambda_{1,j}$ that gives a good compromise between variance and sparsity. When facing the variance-sparsity trade-off, we let variance have a higher priority.

3.2.4 Adjusted total variance

The ordinary principal components are uncorrelated and their loadings are orthogonal. Let $\widehat{\mathbf{\Sigma}} = \mathbf{X}^T\mathbf{X}$, then $\mathbf{V}^T\mathbf{V} = \mathbf{I}_k$ and $\mathbf{V}^T\widehat{\mathbf{\Sigma}}\mathbf{V}$ is diagonal. It is easy to check that it is only for ordinary principal components the the loadings can satisfy both conditions. In Jolliffe, Trendafilov & Uddin (2003) the loadings were forced to be orthogonal, so the uncorrelated property was sacrificed. SPCA does not explicitly impose the uncorrelated components

condition neither.

Let $\widehat{\mathbf{Z}}$ be the modified PCs. Usually the total variance explained by $\widehat{\mathbf{Z}}$ is calculated by $\mathrm{Tr}(\widehat{\mathbf{Z}}^T\widehat{\mathbf{Z}})$. This is reasonable when $\widehat{\mathbf{Z}}$ are uncorrelated. However, if they are correlated, $\mathrm{Tr}(\widehat{\mathbf{Z}}^T\widehat{\mathbf{Z}})$ is too optimistic for representing the total variance. Suppose $(\hat{Z}_i, i = 1, 2, \ldots, k)$ are the first k modified PCs by any method, and the $(k+1)$-th modified PC $\hat{Z}_{k+1}$ is obtained. We want to compute the total variance explained by the first $k + 1$ modified PCs, which should be the sum of the explained variance by the first k modified PCs and the additional variance from $\hat{Z}_{k+1}$. If $\hat{Z}_{k+1}$ is correlated with $(\hat{Z}_i, i = 1, 2, \ldots, k)$, then its variance contains contributions from $(\hat{Z}_i, i = 1, 2, \ldots, k)$, which should not be included into the total variance given the presence of $(\hat{Z}_i, i = 1, 2, \ldots, k)$.

Here we propose a new formula to compute the total variance explained by $\widehat{\mathbf{Z}}$, which takes into account the correlations among $\widehat{\mathbf{Z}}$. We use regression projection to remove the linear dependence between correlated components. Denote $\hat{Z}_{j\cdot 1,\ldots,j-1}$ the residual after adjusting $\hat{Z}_j$ for $\hat{Z}_1, \ldots, \hat{Z}_{j-1}$, that is

$$\hat{Z}_{j\cdot 1,\ldots,j-1} = \hat{Z}_j - \mathbf{H}_{1,\ldots,j-1}\hat{Z}_j, \tag{3.20}$$

where $\mathbf{H}_{1,\ldots,j-1}$ is the projection matrix on $\{\hat{Z}_i\}_1^{j-1}$. Then the adjusted variance of $\hat{Z}_j$ is $\|\hat{Z}_{j\cdot 1,\ldots,j-1}\|^2$, and the total explained variance is defined as $\sum_{j=1}^{k} \|\hat{Z}_{j\cdot 1,\ldots,j-1}\|^2$. When the modified PCs $\widehat{\mathbf{Z}}$ are uncorrelated, the new formula agrees with $\mathrm{Tr}(\widehat{\mathbf{Z}}^T\widehat{\mathbf{Z}})$.

Note that the above computations depend on the order of $\hat{Z}_i$. However, since we have a natural order in PCA, ordering is not an issue here. Using the QR decomposition, we can easily compute the adjusted variance. Suppose $\widehat{\mathbf{Z}} = \mathbf{QR}$, where $\mathbf{Q}$ is orthonormal and $\mathbf{R}$ is upper triangular. Then it is straightforward to see that

$$\|\hat{Z}_{j\cdot 1,\ldots,j-1}\|^2 = \mathbf{R}_{jj}^2. \tag{3.21}$$

Hence the explained total variance is equal to $\sum_{j=1}^{k} \mathbf{R}_{jj}^2$.

3.2.5 Computation complexity

PCA is computationally efficient for both $n > p$ or $p \gg n$ data. We separately discuss the computational cost of the general SPCA algorithm for $n > p$ and $p \gg n$.

1. $n > p$. Traditional multivariate data fit in this category. Note that although the SPCA criterion is defined using $\mathbf{X}$, it only depends on $\mathbf{X}$ via $\mathbf{X}^T\mathbf{X}$. A trick is to first compute the $p \times p$ matrix $\widehat{\mathbf{\Sigma}} = \mathbf{X}^T\mathbf{X}$ once for all, which requires np^2 operations. Then the same $\widehat{\mathbf{\Sigma}}$ is used at each step within the loop. Computing $\mathbf{X}^T\mathbf{X}\beta$ costs p^2k and the SVD of $\mathbf{X}^T\mathbf{X}\beta$ is of order $O(pk^2)$. Each elastic net solution requires at most $O(p^3)$ operations. Since $k \leq p$, the total computation cost is at most $np^2 + mO(p^3)$, where m is the number of iterations before convergence. Therefore the SPCA algorithm is able to efficiently handle data with huge n, as long as p is small (say $p < 100$).

2. $p \gg n$. Gene expression arrays are typical examples in this $p \gg n$ category. The trick of using $\widehat{\mathbf{\Sigma}}$ is no longer applicable, because $\widehat{\mathbf{\Sigma}}$ is a huge matrix ($p \times p$) in this case. The most consuming step is solving each elastic net, whose cost is of order $O(pnJ + J^3)$ for a positive finite λ, where J is the number of nonzero coefficients. Generally speaking the total cost is of order $mkO(pJn + J^3)$, which can be expensive for large J and p. Fortunately, as shown in the next section, there exists a special SPCA algorithm for efficiently dealing with $p \gg n$ data.

3.3 SPCA for $p \gg n$ and Gene Expression Arrays

For gene expression arrays the number of variables (genes) is typically much bigger than the number of samples (e.g $n = 10,000$, $p = 100$). Our general SPCA algorithm still fits this situation using a positive λ. However the computational cost is expensive when requiring a large number of nonzero loadings. It is desirable to simplify the general SPCA algorithm to boost the computation.

Observe that Theorem 3 is valid for all $\lambda > 0$, so in principle we can use any positive λ. It turns out that a thrifty solution emerges if $\lambda \to \infty$. Precisely, we have the following

theorem.

Theorem 3.5. *Let* $\widehat{V}_j(\lambda) = \frac{\hat{\beta}_j}{\|\hat{\beta}_j\|}$ *$(j = 1, \ldots, k)$ be the loadings derived from criterion (3.13). Let* $(\widehat{\mathbf{A}}, \widehat{\mathbf{B}})$ *be the solution of the optimization problem*

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}) = \arg\min_{\mathbf{A},\mathbf{B}} -2\mathrm{Tr}\left(\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{B}\right) + \sum_{j=1}^{k} \|\beta_j\|^2 + \sum_{j=1}^{k} \lambda_{1,j}\|\beta_j\|_1 \quad (3.22)$$
$$\textit{subject to} \quad \mathbf{A}^T\mathbf{A} = \mathbf{I}_{k\times k}.$$

When $\lambda \to \infty$, $\widehat{V}_j(\lambda) \to \frac{\hat{\beta}_j}{\|\hat{\beta}_j\|}$.

We can use the same alternating algorithm in Section 3.2.3 to solve (3.22), where we only need to replace the general elastic net problem with its special case ($\lambda = \infty$). Note that given $\mathbf{A}$,

$$\hat{\beta}_j = \arg\min_{\beta_j} -2\alpha_j^T(\mathbf{X}^T\mathbf{X})\beta_j + \|\beta_j\|^2 + \lambda_{1,j}\|\beta_j\|_1, \quad (3.23)$$

which has an explicit form solution given in (3.24).

Gene Expression Arrays SPCA Algorithm

Replacing step 2 in the general SPCA algorithm with

Step 2*: for $j = 1, 2, \ldots, k$

$$\beta_j = \left(\left|\alpha_j^T\mathbf{X}^T\mathbf{X}\right| - \frac{\lambda_{1,j}}{2}\right)_+ Sign(\alpha_j^T\mathbf{X}^T\mathbf{X}). \quad (3.24)$$

The operation in (3.24) is called soft-thresholding. Figure 3.1 gives an illustration of how the soft-thresholding rule operates. Recently soft-thresholding has become increasingly popular in the literature. For example, nearest shrunken centroids (Tibshirani, Hastie, Narasimhan & Chu 2002) adopts the soft-thresholding rule to simultaneously classify samples and select important genes in microarrays.

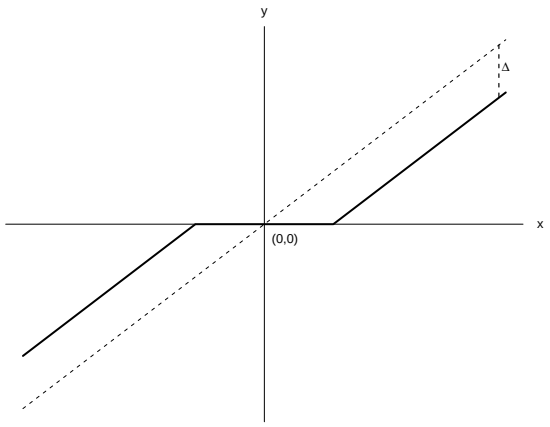

Figure 3.1: *An illustration of soft-thresholding rule* $y = (|x| - \Delta)_+ Sign(x)$ *with* $\Delta = 1$.

3.4 Examples

3.4.1 Pitprops data

The pitprops data first introduced in Jeffers (1967) has 180 observations and 13 measured variables. It is a classic example showing the difficulty of interpreting principal components. Jeffers (1967) tried to interpret the first six PCs. Jolliffe, Trendafilov & Uddin (2003) used their SCoTLASS to find the modified PCs. Table 3.1 presents the results of PCA, while Table 3.2 presents the modified PC loadings as computed by SCoTLASS and the adjusted variance computed using (3.21).

As a demonstration, we also considered the first six principal components. Since this is a usual $n \gg p$ data set, we set $\lambda = 0$. $\lambda_1 = (0.06, 0.16, 0.1, 0.5, 0.5, 0.5)$ were chosen according to Figure 3.2 such that each sparse approximation explained almost the same amount of variance as the ordinary PC did. Table 3.3 shows the obtained sparse loadings and the corresponding adjusted variance. Compared with the modified PCs of SCoTLASS, PCs by SPCA account for nearly the same amount of variance (75.8% vs. 78.2%) but with a much sparser loading structure. The important variables associated with the six PCs do

not overlap, which further makes the interpretations easier and clearer. It is interesting to note that in Table 3.3 even though the variance does not strictly monotonously decrease, the adjusted variance follows the right order. However, Table 3.2 shows that this is not true in SCoTLASS. It is also worthy of mention that the entire SPCA computation was done in seconds in `R`, while the implementation of SCoTLASS for each t was expensive (Jolliffe, Trendafilov & Uddin 2003). Optimizing SCoTLASS over several values of t is an even more difficult computational challenge.

Although the informal thresholding method, which we henceforth refer to as simple thresholding, has various drawbacks, it may serve as the benchmark for testing sparse PCs methods. A variant of simple thresholding is soft-thresholding. We found that when used in PCA, soft-thresholding performs very similarly to simple thresholding. Thus we omitted the results of soft-thresholding in this paper. Both SCoTLASS and SPCA were compared with simple thresholding. Table 3.4 presents the loadings and the corresponding variance explained by simple thresholding. To make the comparisons fair, we let the numbers of nonzero loadings obtained by simple thresholding match the results of SCoTLASS and SPCA, as shown in the top and bottom parts of Table 3.4, respectively. In terms of variance, it seems that simple thresholding is better than SCoTLASS and worse than SPCA. Moreover, the variables with non-zero loadings by SPCA are different to that chosen by simple thresholding for the first three PCs; while SCoTLASS seems to create a similar sparseness pattern as simple thresholding does, especially in the leading PC.

3.4.2 A synthetic example

Our synthetic example has three *hidden* factors

$$\begin{aligned} V_1 \sim N(0, 290), \quad & V_2 \sim N(0, 300) \\ V_3 = -0.3V_1 + 0.925V_2 + \epsilon, \quad & \epsilon \sim N(0, 1) \\ V_1, V_2 \text{ and } \epsilon \quad & \text{are independent.} \end{aligned}$$

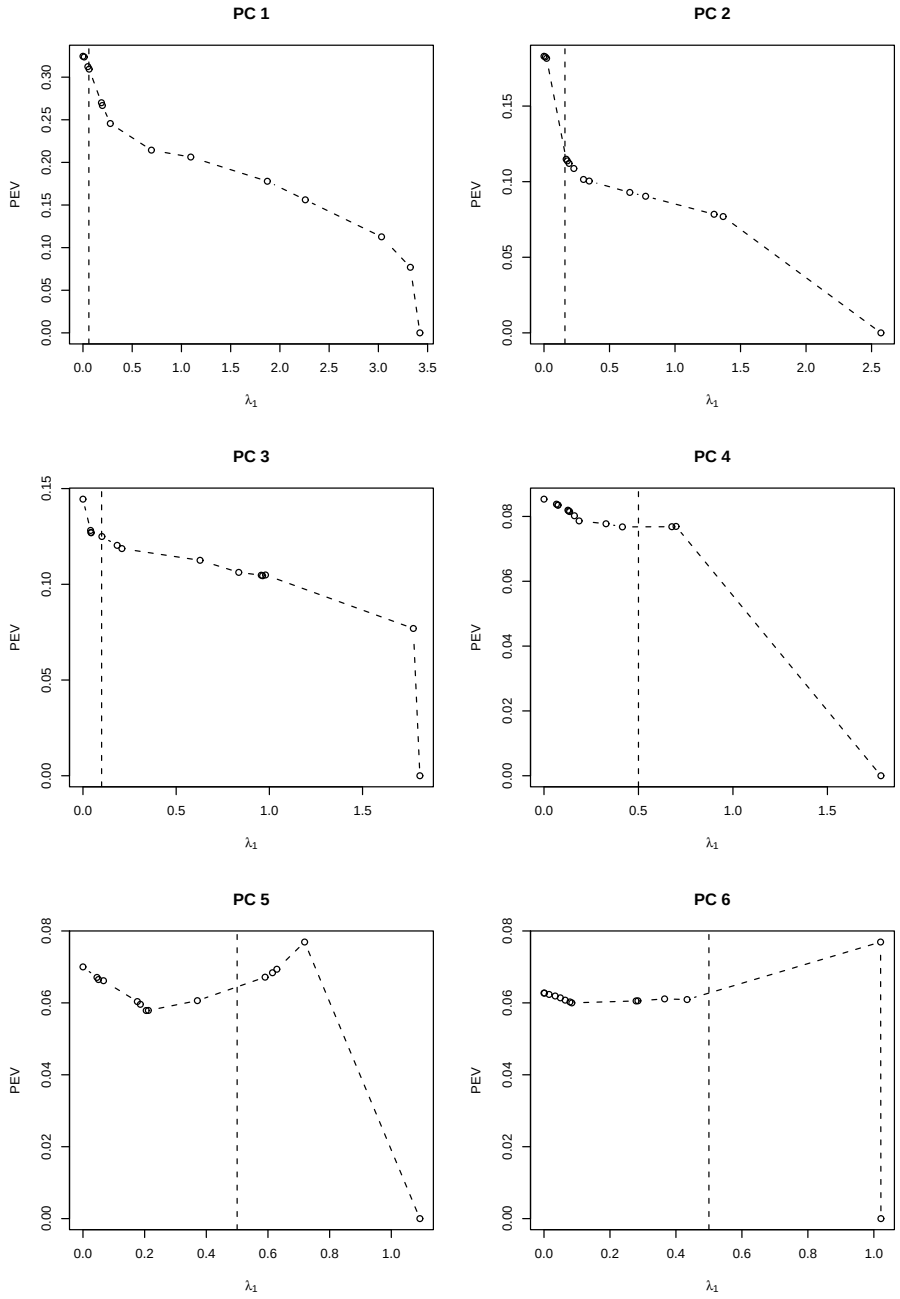

Figure 3.2: *Pitprops data: The sequences of sparse approximations to the first 6 principal components. The curves show the percentage of explained variance (PEV) as a function of* λ_1*. The vertical broken lines indicate the choice of* λ_1 *used in our SPCA analysis.*

Table 3.1: *Pitprops data: loadings of the first six principal components*

Variable	PC1	PC2	PC3	PC4	PC5	PC6
topdiam	-0.404	0.218	-0.207	0.091	-0.083	0.120
length	-0.406	0.186	-0.235	0.103	-0.113	0.163
moist	-0.124	0.541	0.141	-0.078	0.350	-0.276
testsg	-0.173	0.456	0.352	-0.055	0.356	-0.054
ovensg	-0.057	-0.170	0.481	-0.049	0.176	0.626
ringtop	-0.284	-0.014	0.475	0.063	-0.316	0.052
ringbut	-0.400	-0.190	0.253	0.065	-0.215	0.003
bowmax	-0.294	-0.189	-0.243	-0.286	0.185	-0.055
bowdist	-0.357	0.017	-0.208	-0.097	-0.106	0.034
whorls	-0.379	-0.248	-0.119	0.205	0.156	-0.173
clear	0.011	0.205	-0.070	-0.804	-0.343	0.175
knots	0.115	0.343	0.092	0.301	-0.600	-0.170
diaknot	0.113	0.309	-0.326	0.303	0.080	0.626
Variance (%)	32.4	18.3	14.4	8.5	7.0	6.3
Cumulative Variance (%)	32.4	50.7	65.1	73.6	80.6	86.9

Table 3.2: *Pitprops data: loadings of the first six modified PCs by SCoTLASS. Empty cells have zero loadings.*

$t = 1.75$						
Variable	PC1	PC2	PC3	PC4	PC5	PC6
topdiam	0.546	0.047	-0.087	0.066	-0.046	
length	0.568		-0.076	0.117	-0.081	
moist		0.641	-0.187	-0.127	0.009	0.017
testsg		0.641		-0.139		
ovensg			0.457		-0.614	-0.562
ringtop		0.356	0.348			-0.045
ringbut	0.279		0.325			
bowmax	0.132	-0.007		-0.589		
bowdist	0.376					0.065
whorls	0.376	-0.065		-0.067	0.189	-0.065
clear					-0.659	0.725
knots		0.206		0.771	0.040	0.003
diaknot			-0.718	0.013	-0.379	-0.384
Number of nonzero loadings	6	7	7	8	8	8
Variance (%)	27.2	16.4	14.8	9.4	7.1	7.9
Adjusted Variance (%)	27.2	15.3	14.4	7.1	6.7	7.5
Cumulative Adjusted Variance (%)	27.2	42.5	56.9	64.0	70.7	78.2

Table 3.3: *Pitprops data: loadings of the first six sparse PCs by SPCA. Empty cells have zero loadings.*

Variable	PC1	PC2	PC3	PC4	PC5	PC6
topdiam	-0.477					
length	-0.476					
moist		0.785				
testsg		0.620				
ovensg	0.177		0.640			
ringtop			0.589			
ringbut	-0.250		0.492			
bowmax	-0.344	-0.021				
bowdist	-0.416					
whorls	-0.400					
clear				-1		
knots		0.013			-1	
diaknot			-0.015			1
Number of nonzero loadings	7	4	4	1	1	1
Variance (%)	28.0	14.4	15.0	7.7	7.7	7.7
Adjusted Variance (%)	28.0	14.0	13.3	7.4	6.8	6.2
Cumulative Adjusted Variance (%)	28.0	42.0	55.3	62.7	69.5	75.8

Then 10 observable variables are constructed as follows

$$
\begin{aligned}
X_i &= V_1 + \epsilon_i^1, \quad \epsilon_i^1 \sim N(0,1), \quad i = 1,2,3,4, \\
X_i &= V_2 + \epsilon_i^2, \quad \epsilon_i^2 \sim N(0,1), \quad i = 5,6,7,8, \\
X_i &= V_3 + \epsilon_i^3, \quad \epsilon_i^3 \sim N(0,1), \quad i = 9,10, \\
\{\epsilon_i^j\} &\text{ are independent}, \qquad j = 1,2,3 \qquad i = 1,\cdots,10.
\end{aligned}
$$

We used the exact covariance matrix of $(X_1, \ldots, X_{10})$ to perform PCA, SPCA and simple thresholding (in the population setting).

The variance of the three underlying factors is 290, 300 and 283.8, respectively. The numbers of variables associated with the three factors are 4, 4 and 2. Therefore V_2 and V_1 are almost equally important, and they are much more important than V_3. The first two PCs together explain 99.6% of the total variance. These facts suggest that we only need to consider two derived variables with *"correct"* sparse representations. Ideally, the first

Table 3.4: *Pitprops data: loadings of the first six modified PCs by simple thresholding. Empty cells have zero loadings.*

Simple thresholding vs. SCoTLASS						
Variable	PC1	PC2	PC3	PC4	PC5	PC6
topdiam	-0.439	0.234		0.092		0.120
length	-0.441		-0.253	0.104		0.164
moist		0.582			0.361	-0.277
testsg		0.490	0.379		0.367	
ovensg			0.517		0.182	0.629
ringtop			0.511		-0.326	
ringbut	-0.435		0.272		-0.222	
bowmax	-0.319		-0.261	-0.288	0.191	
bowdist	-0.388			-0.098		
whorls	-0.412	-0.267		0.207		-0.174
clear		0.221		-0.812	-0.354	0.176
knots		0.369		0.304	-0.620	-0.171
diaknot		0.332	-0.350	0.306		0.629
Number of nonzero loadings	6	7	7	8	8	8
Variance (%)	28.9	16.6	14.2	8.6	6.9	6.3
Adjusted Variance (%)	28.9	16.5	14.0	8.5	6.7	6.2
Cumulative Adjusted Variance (%)	28.9	45.4	59.4	67.9	74.6	80.8

Simple thresholding vs. SPCA						
Variable	PC1	PC2	PC3	PC4	PC5	PC6
topdiam	-0.420					
length	-0.422					
moist		0.640				
testsg		0.540	0.425			
ovensg			0.580			
ringtop	-0.296		0.573			
ringbut	-0.416					
bowmax	-0.305					
bowdist	-0.370					
whorls	-0.394					
clear				-1		
knots		0.406			-1	
diaknot		0.365	-0.393			1
Number of nonzero loadings	7	4	4	1	1	1
Variance (%)	30.7	14.8	13.6	7.7	7.7	7.7
Adjusted Variance (%)	30.7	14.7	11.1	7.6	5.2	3.6
Cumulative Adjusted Variance (%)	30.7	45.4	56.5	64.1	68.3	71.9

derived variable should recover the factor V_2 only using (X_5, X_6, X_7, X_8), and the second derived variable should recover the factor V_1 only using (X_1, X_2, X_3, X_4). In fact, if we sequentially maximize the variance of the first two derived variables under the orthonormal constraint, while restricting the numbers of nonzero loadings to four, then the first derived variable uniformly assigns nonzero loadings on (X_5, X_6, X_7, X_8); and the second derived variable uniformly assigns nonzero loadings on (X_1, X_2, X_3, X_4).

Both SPCA ($\lambda = 0$) and simple thresholding were carried out by using the oracle information that the ideal sparse representations use only four variables. Table 3.5 summarizes the comparison results. Clearly, SPCA correctly identifies the sets of important variables. In fact, SPCA delivers the ideal sparse representations of the first two principal components. Mathematically, it is easy to show that if $t = 2$ is used, SCoTLASS is also able to find the same sparse solution. In this example, both SPCA and SCoTLASS produce the ideal sparse PCs, which may be explained by the fact that both methods explicitly use the lasso penalty.

In contrast, simple thresholding incorrectly includes X_9, X_{10} in the most important variables. The variance explained by simple thresholding is also lower than that by SPCA, although the relative difference is small (less than 5%). Due to the high correlation between V_2 and V_3, variables X_9, X_{10} achieve loadings which are even higher than those of the true important variables (X_5, X_6, X_7, X_8). Thus the truth is disguised by the high correlation. On the other hand, simple thresholding correctly discovers the second factor, because V_1 has a low correlation with V_3.

3.4.3 Ramaswamy data

An important task in microarray analysis is to find a set of genes which are biologically relevant to the outcome (e.g. tumor type or survival time). PCA (or SVD) has been a popular tool for this purpose. Many gene-clustering methods in the literature use PCA (or SVD) as a building block. For example, *gene shaving* (Hastie, Tibshirani, Eisen, Brown, Ross, Scherf, Weinstein, Alizadeh, Staudt & Botstein 2000) uses an iterative principal component shaving algorithm to identify subsets of coherent genes. Here we consider another approach

Table 3.5: *Results of the simulation example: loadings and variance.*

	PCA			SPCA ($\lambda = 0$)		Simple	Thresholding
	PC1	PC2	PC3	PC1	PC2	PC1	PC2
X_1	0.116	-0.478	-0.087	0	0.5	0	-0.5
X_2	0.116	-0.478	-0.087	0	0.5	0	-0.5
X_3	0.116	-0.478	-0.087	0	0.5	0	-0.5
X_4	0.116	-0.478	-0.087	0	0.5	0	-0.5
X_5	-0.395	-0.145	0.270	0.5	0	0	0
X_6	-0.395	-0.145	0.270	0.5	0	0	0
X_7	-0.395	-0.145	0.270	0.5	0	-0.497	0
X_8	-0.395	-0.145	0.270	0.5	0	-0.497	0
X_9	-0.401	0.010	-0.582	0	0	-0.503	0
X_{10}	-0.401	0.010	-0.582	0	0	-0.503	0
Adjusted Variance (%)	60.0	39.6	0.08	40.9	39.5	38.8	38.6

to gene selection through SPCA. The idea is intuitive: if the (sparse) principal component can explain a large part of the total variance of gene expression levels, then the subset of genes representing the principal component are considered important.

We illustrate the sparse PC selection method on Ramaswamy's data (Ramaswamy, Tamayo, Rifkin, Mukheriee, Yeang, Angelo, Ladd, Reich, Latulippe, Mesirov, Poggio, Gerald, Loda, Lander & Golub 2001) which has 16063 ($p = 16063$) genes and 144 ($n = 144$) samples. Its first principal component explains 46% of the total variance. For microarray data like this, it appears that SCoTLASS cannot be practically useful for finding sparse PCs. We applied SPCA ($\lambda = \infty$) to find the leading sparse PC. A sequence of values for λ_1 were used such that the number of nonzero loadings varied over a wide range. As displayed in Figure 3.3, the percentage of explained variance decreases at a slow rate, as the sparsity increases. As few as 2.5% of these 16063 genes can sufficiently construct the leading principal component with an affordable loss of explained variance (from 46% to 40%). Simple thresholding was also applied to this data. It seems that when using the same number of genes, simple thresholding always explains slightly higher variance than SPCA does. Among the same number of selected genes by SPCA and simple thresholding, there are about 2% different genes, and this difference rate is quite consistent.

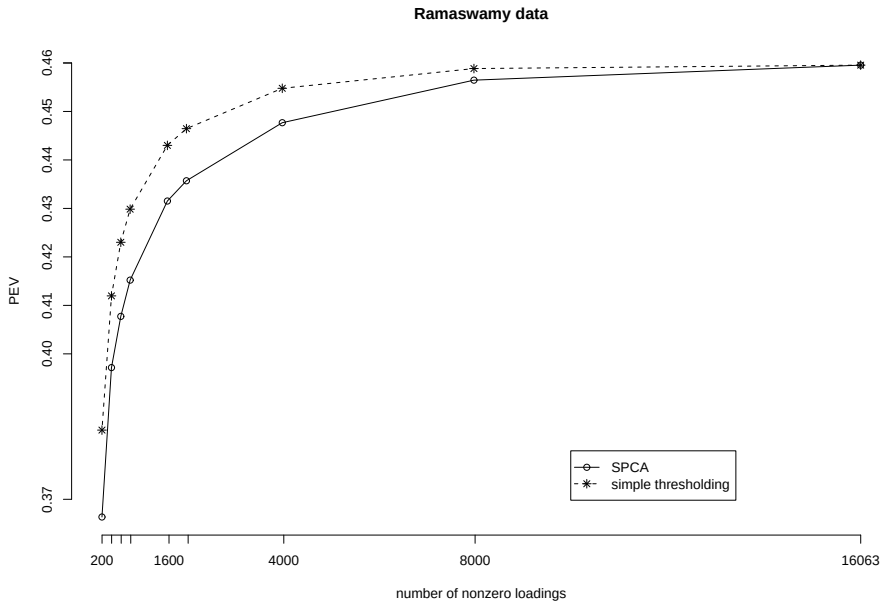

Figure 3.3: *The sparse leading principal component: percentage of explained variance versus sparsity. Simple thresholding and SPCA have similar performances. However, there still exists consistent difference in the selected genes (the ones with nonzero loadings).*

3.5 Discussion

It has been a long standing interest to have a formal approach to derive principal components with sparse loadings. From a practical point of view, a good method to achieve the sparseness goal should (at least) possess the following properties.

- Without any sparsity constraint, the method should reduce to PCA.
- It should be computationally efficient for both small p and big p data.
- It should avoid misidentifying the important variables.

The often-used simple thresholding approach is not criterion based. However, this informal method seems to possess the first two of the desirable properties listed above. If the explained variance and sparsity are the only concerns, simple thresholding is a reasonable approach, and it is extremely convenient. We have shown that simple thresholding can work well with gene expression arrays. The serious problem with simple thresholding is that it

can misidentify the real important variables. Nevertheless, simple thresholding is regarded as a benchmark for any potentially better method.

Using the lasso constraint in PCA, SCoTLASS successfully derives sparse loadings. However, SCoTLASS is not computationally efficient, and it lacks a good rule to pick its tuning parameter. In addition, it is not feasible to apply SCoTLASS to gene expression arrays, where PCA is a quite popular tool.

In this Chapter we have developed SPCA using our SPCA criterion (3.13). This new criterion gives exact PCA results when its sparsity (lasso) penalty term vanishes. SPCA allows flexible control on the sparse structure of the resulting loadings. Unified efficient algorithms have been proposed to compute SPCA solutions for both regular multivariate data and gene expression arrays. As a principled procedure, SPCA enjoys advantages in several aspects, including computational efficiency, high explained variance and an ability in identifying important variables.

3.6 Proofs of Theorems 3.1-3.5

Proof of Theorem 3.1. Using $\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{D}^2\mathbf{V}^T$ and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$, we have

$$\hat{\beta}_{\text{ridge}} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T(\mathbf{X}V_i) = V_i\frac{\mathbf{D}_{ii}^2}{\mathbf{D}_{ii}^2 + \lambda}. \tag{3.25}$$

Hence $\hat{v} = V_i$. □

Note that since Theorem 3.2 is a special case of Theorem 3.3, we will not prove it separately. We first provide a lemma.

Lemma 3.1. *Consider the ridge regression criterion*

$$C_\lambda(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|^2.$$

Then if $\hat{\beta} = \arg\min_\beta C_\lambda(\beta)$,

$$C_\lambda(\hat{\beta}) = \mathbf{y}^T(\mathbf{I} - \mathbf{S}_\lambda)\mathbf{y},$$

where $\mathbf{S}_\lambda$ *is the ridge operator*

$$\mathbf{S}_\lambda = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T.$$

Proof of Lemma 3.1. Differentiating C_λ wrt β, we get that

$$-\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\beta}) + \lambda\hat{\beta} = 0.$$

Pre-multiplication by $\hat{\beta}^T$ and re-arrangement gives $\lambda\|\hat{\beta}\|^2 = (\mathbf{y} - \mathbf{X}\hat{\beta})^T\mathbf{X}\hat{\beta}$. Since

$$\|\mathbf{y} - \mathbf{X}\hat{\beta}\|^2 = (\mathbf{y} - \mathbf{X}\hat{\beta})^T\mathbf{y} - (\mathbf{y} - \mathbf{X}\hat{\beta})^T\mathbf{X}\hat{\beta},$$

$C_\lambda(\hat{\beta}) = (\mathbf{y} - \mathbf{X}\hat{\beta})^T\mathbf{y}$. The result follows since the "fitted values" $\mathbf{X}\hat{\beta} = \mathbf{S}_\lambda\mathbf{y}$. □

Proof of Theorem 3.3. We use the notation introduced in Section 3.2: $\mathbf{A} = [\alpha_1, \cdots, \alpha_k]$ and $\mathbf{B} = [\beta_1, \cdots, \beta_k]$. Let

$$C_\lambda(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 + \lambda\sum_{j=1}^{k} \|\beta_j\|^2.$$

As in (3.10) we have

$$\begin{aligned} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\mathbf{B}^T\mathbf{x}_i\|^2 &= \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}^T\|^2 && (3.26)\\ &= \|\mathbf{X}\mathbf{A}_\perp\|^2 + \|\mathbf{X}\mathbf{A} - \mathbf{X}\mathbf{B}\|^2. && (3.27) \end{aligned}$$

Hence, with $\mathbf{A}$ fixed, solving

$$\arg\min_{\mathbf{B}} C_\lambda(\mathbf{A}, \mathbf{B})$$

is equivalent to solving the series of ridge regressions

$$\arg\min_{\{\beta_j\}_1^k} \sum_{j=1}^{k} \{\|\mathbf{X}\alpha_j - \mathbf{X}\beta_j\|^2 + \lambda\|\beta_j\|^2\}.$$

It is easy to show that

$$\widehat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\mathbf{A}, \tag{3.28}$$

and using Lemma 3.1 and (3.26) we have that the partially optimized penalized criterion is given by

$$C_\lambda(\mathbf{A}, \widehat{\mathbf{B}}) = \|\mathbf{X}\mathbf{A}_\perp\|^2 + \mathrm{Tr}\left((\mathbf{X}\mathbf{A})^T(\mathbf{I} - \mathbf{S}_\lambda)(\mathbf{X}\mathbf{A})\right). \tag{3.29}$$

Rearranging the terms, we get

$$C_\lambda(\mathbf{A}, \widehat{\mathbf{B}}) = \mathrm{Tr}(\mathbf{X}^T\mathbf{X}) - \mathrm{Tr}(\mathbf{A}^T\mathbf{X}^T\mathbf{S}_\lambda\mathbf{X}\mathbf{A}), \tag{3.30}$$

which must be minimized wrt $\mathbf{A}$ with $\mathbf{A}^T\mathbf{A} = \mathbf{I}$. Hence $\mathbf{A}$ should be taken to be the largest k eigenvectors of $\mathbf{X}^T\mathbf{S}_\lambda\mathbf{X}$. If the SVD of $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, it is easy to show that $\mathbf{X}^T\mathbf{S}_\lambda\mathbf{X} = \mathbf{V}\mathbf{D}^2(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{D}^2\mathbf{V}^T$, hence $\widehat{\mathbf{A}} = \mathbf{V}[, 1:k]$. Likewise, plugging the SVD of $\mathbf{X}$ into (3.28), we see that each of the $\hat{\beta}_j$ are scaled elements of the corresponding V_j. □

Proof of Theorem 3.4. We expand the matrix norm

$$\|\mathbf{M} - \mathbf{N}\mathbf{A}^T\|^2 \;=\; \mathrm{Tr}(\mathbf{M}^T\mathbf{M}) - 2\mathrm{Tr}(\mathbf{M}^T\mathbf{N}\mathbf{A}^T) + \mathrm{Tr}(\mathbf{A}\mathbf{N}^T\mathbf{N}\mathbf{A}^T). \tag{3.31}$$

Since $\mathbf{A}^T\mathbf{A} = \mathbf{I}$, the last term is equal to $\mathrm{Tr}(\mathbf{N}^T\mathbf{N})$, and hence we need to maximize (minus half) the middle term. With the SVD $\mathbf{M}^T\mathbf{N} = \mathbf{U}\mathbf{D}\mathbf{V}^T$, this middle term becomes

$$\begin{aligned} \mathrm{Tr}(\mathbf{M}^T\mathbf{N}\mathbf{A}^T) &= \mathrm{Tr}(\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{A}^T) && (3.32)\\ &= \mathrm{Tr}(\mathbf{U}\mathbf{D}\mathbf{A}^{*T}) && (3.33)\\ &= \mathrm{Tr}(\mathbf{A}^{*T}\mathbf{U}\mathbf{D}), && (3.34) \end{aligned}$$

where $\mathbf{A}^* = \mathbf{A}\mathbf{V}$, and since $\mathbf{V}$ is $k \times k$ orthonormal, $\mathbf{A}^{*T}\mathbf{A}^* = \mathbf{I}$. Now since $\mathbf{D}$ is diagonal, (3.34) is maximized when the diagonal of $\mathbf{A}^{*T}\mathbf{U}$ is positive and maximum. By Cauchy-Schwartz, this is achieved when $\mathbf{A}^* = \mathbf{U}$, in which case the diagonal elements are all 1. Hence $\hat{\mathbf{A}} = \mathbf{U}\mathbf{V}^T$.

□

Proof of Theorem 3.5. Let $\widehat{\mathbf{B}}^* = [\hat{\beta}_1^*, \cdots, \hat{\beta}_k^*]$ with $\hat{\beta}^* = (1+\lambda)\hat{\beta}$, then $\hat{V}_i(\lambda) = \frac{\hat{\beta}_i^*}{||\hat{\beta}_i^*||}$. On the other hand, $\hat{\beta} = \frac{\hat{\beta}^*}{1+\lambda}$ means that

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}^*) = \arg\min_{\mathbf{A},\mathbf{B}} C_{\lambda,\lambda_1}(\mathbf{A},\mathbf{B}) \quad \text{subject to} \quad \mathbf{A}^T\mathbf{A} = \mathbf{I}_{k\times k}, \tag{3.35}$$

where

$$C_{\lambda,\lambda_1}(\mathbf{A},\mathbf{B}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{A}\frac{\mathbf{B}^T}{1+\lambda}\mathbf{x}_i\|^2 + \lambda\sum_{j=1}^{k}\|\frac{\beta_j}{1+\lambda}\|^2 + \sum_{j=1}^{k}\lambda_{1,j}\|\frac{\beta_j}{1+\lambda}\|_1. \tag{3.36}$$

Since

$$\sum_{j=1}^{k}\|\frac{\beta_j}{1+\lambda}\|^2 = \frac{1}{(1+\lambda)^2}\mathrm{Tr}(\mathbf{B}^T\mathbf{B}),$$

and

$$\begin{aligned}\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{A}\frac{\mathbf{B}^T}{1+\lambda}\mathbf{x}_i\|^2 &= \mathrm{Tr}\left((\mathbf{X} - \mathbf{X}\frac{\mathbf{B}}{1+\lambda}\mathbf{A}^T)^T(\mathbf{X} - \mathbf{X}\frac{\mathbf{B}}{1+\lambda}\mathbf{A}^T)\right)\\ &= \mathrm{Tr}\left(\mathbf{X}^T\mathbf{X}\right) + \frac{1}{(1+\lambda)^2}\mathrm{Tr}\left(\mathbf{B}^T\mathbf{X}^T\mathbf{X}\mathbf{B}\right) - \frac{2}{1+\lambda}\mathrm{Tr}\left(\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{B}\right).\end{aligned}$$

Thus we have

$$C_{\lambda,\lambda_1}(\mathbf{A},\mathbf{B}) = \mathrm{Tr}\left(\mathbf{X}^T\mathbf{X}\right) + \frac{1}{1+\lambda}\left(\mathrm{Tr}\left(\mathbf{B}^T\frac{\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}}{1+\lambda}\mathbf{B}\right) - 2\mathrm{Tr}\left(\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{B}\right) + \sum_{j=1}^{k}\lambda_{1,j}\|\beta_j\|_1\right),$$

which implies that

$$(\widehat{\mathbf{A}}, \widehat{\mathbf{B}}^*) = \arg\min_{\mathbf{A},\mathbf{B}} \mathrm{Tr}\left(\mathbf{B}^T\frac{\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}}{1+\lambda}\mathbf{B}\right) - 2\mathrm{Tr}\left(\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{B}\right) + \sum_{j=1}^{k}\lambda_{1,j}\|\beta_j\|_1, \tag{3.37}$$

subject to $\mathbf{A}^T\mathbf{A} = \mathbf{I}_{k\times k}$. As $\lambda \to \infty$, $\mathrm{Tr}\left(\mathbf{B}^T\frac{\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I}}{1+\lambda}\mathbf{B}\right) \to \mathrm{Tr}(\mathbf{B}^T\mathbf{B}) = \sum_{j=1}^{k}\|\beta_j\|^2$. Thus (3.37) approaches (3.22) and the conclusion of Theorem 3.5 follows.

□

Chapter 4

Degrees of Freedom of the Lasso

In this Chapter we study the degrees of freedom of the lasso in the framework of Stein's unbiased risk estimation (SURE). We show that the number of non-zero coefficients is an unbiased estimate for the degrees of freedom of the lasso—a conclusion that requires no special assumption on the predictors. Our analysis also provides mathematical support for a related conjecture by Efron et al. (2004). As an application, various model selection criteria—C_p, AIC and BIC—are defined, which, along with the LARS algorithm, provide a principled and efficient approach to obtaining the optimal lasso fit with the computational efforts of a single ordinary least-squares fit. We propose the use of BIC-lasso shrinkage if the lasso is primarily used as a variable selection method.

4.1 Introduction

Modern data sets typically have a large number of observations and predictors. A typical goal in model fitting is to achieve good prediction accuracy with a sparse representation of the predictors in the model.

The lasso is a promising automatic model building technique, simultaneously producing accurate and parsimonious models (Tibshirani 1996). Suppose $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the response vector and $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})^T, j = 1, \ldots, p$ are the linearly independent predictors. Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_p]$ be the predictor matrix. The lasso estimates for the coefficients of a

linear model solve

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j\|^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq t. \tag{4.1}$$

Or equivalently

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{4.2}$$

where λ is a non-negative regularization parameter. Without loss of generality we assume that the data are centered, so the intercept is not included in the above model. There is a one-one correspondence (generally depending on the data) between t and λ making the optimization problems in (4.1) and (4.2) equivalent. The second term in (4.2) is called the 1-norm penalty and λ is called as the lasso regularization parameter. Since the $Loss+Penalty$ formulation is common in the statistical community, we use the representation (4.2). Figure 4.1 displays the lasso estimates as a function of λ using the diabetes data (Efron, Hastie, Johnstone & Tibshirani 2004). As can be seen from Figure 4.1 (the left plot), the lasso continuously shrinks the coefficients toward zero as λ increases; and some coefficients are shrunk to exact zero if λ is sufficiently large. In addition, the shrinkage often improves the prediction accuracy due to the bias-variance trade-off. Thus the lasso simultaneously achieves accuracy and sparsity.

Generally speaking, the purpose of regularization is to control the complexity of the fitted model (Hastie, Tibshirani & Friedman 2001). The least regularized lasso ($\lambda = 0$) corresponds to Ordinary Least Squares (OLS); while the most regularized lasso uses $\lambda = \infty$, yielding a constant fit. So the model complexity is reduced via shrinkage. However, the effect of the lasso shrinkage is not very clear except for these two extreme cases. An informative measurement of model complexity is the *effective degrees of freedom* (Hastie & Tibshirani 1990). The profile of degrees of freedom clearly shows that how the model complexity is controlled by shrinkage. The degrees of freedom also plays an important role in estimating the prediction accuracy of the fitted model, which helps us pick an optimal model among all the possible candidates, e.g. the optimal choice of λ in the lasso. Thus it

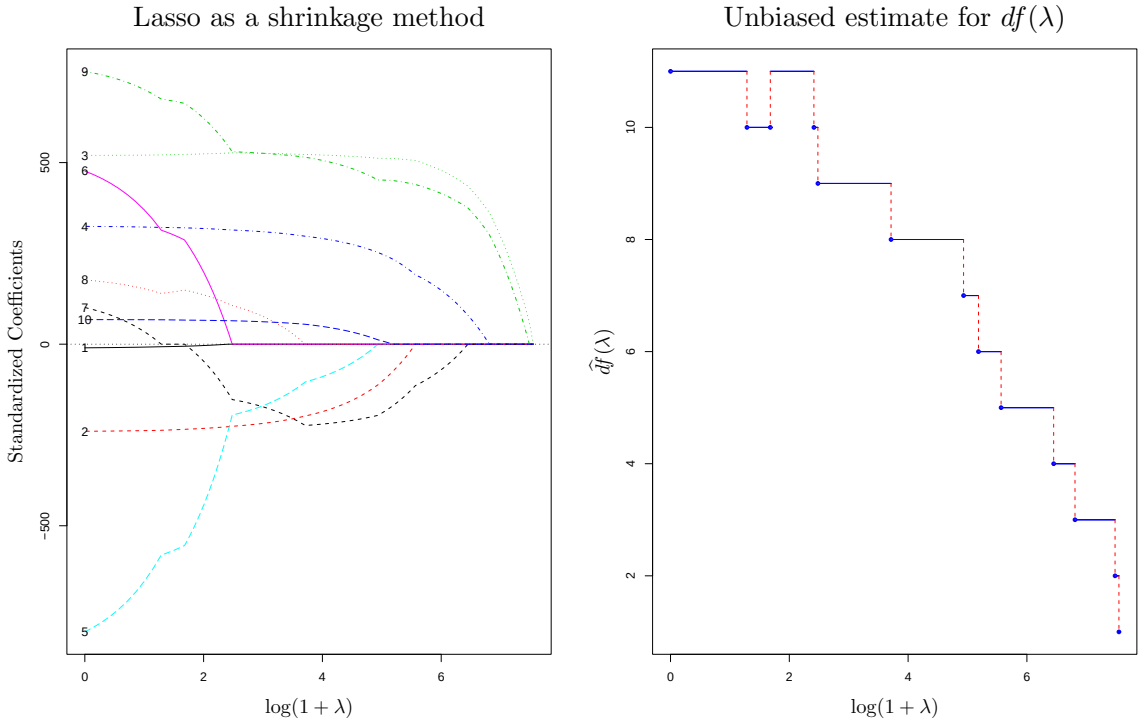

Figure 4.1: *Diabetes data with 10 predictors. The left panel shows the lasso coefficients estimates* $\hat{\beta}_j, j = 1, 2, \ldots, 10$*, for the diabetes study. The diabetes data were standardized. The lasso coefficients estimates are piece-wise linear functions of* λ *(Efron et al. 2004), hence they are piece-wise non-linear as functions of* $\log(1+\lambda)$*. The right panel shows the curve of the proposed unbiased estimate for the degrees of freedom of the lasso, whose piece-wise constant property is basically determined by the piece-wise linearity of* $\hat{\beta}$*.*

is desirable to know what is the degrees of freedom of the lasso for a given regularization parameter λ, or $df(\lambda)$. This is an interesting problem of both theoretical and practical importance.

Degrees of freedom are well studied for linear procedures. For example, the degrees of freedom in multiple linear regression exactly equals the number of predictors. A generalization is made for all linear smoothers (Hastie & Tibshirani 1990), where the fitted vector is written as $\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$ and the smoother matrix $\mathbf{S}$ is free of $\mathbf{y}$. Then $df(\mathbf{S}) = \mathrm{tr}(\mathbf{S})$ (see Section 4.2). A leading example is ridge regression (Hoerl & Kennard 1988) with $\mathbf{S} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}$. These results rely on the convenient expressions for representing linear smoothers. Unfortunately, the explicit expression of the lasso fit is not available (at least so far) due to the nonlinear nature of the lasso, thus the nice results for linear smoothers are not directly applicable.

Efron, Hastie, Johnstone & Tibshirani (2004) (referred to as the LAR paper henceforth) propose *Least Angle Regression* (LARS), a new stage-wise model building algorithm. They show that a simple modification of LARS yields the entire lasso solution path with the computational cost of a single OLS fit. LARS describes the lasso as a forward stage-wise model fitting process. Starting at zero, the lasso fits are sequentially updated till reaching the OLS fit, while being piece-wise linear between successive steps. The updates follow the current *equiangular direction*. Figure 4.2 shows how the lasso estimates evolve step by step.

From the forward stage-wise point of view, it is natural to consider the number of steps as the meta parameter to control the model complexity. In the LAR paper, it is shown that under a *"positive cone" condition*, the degrees of freedom of LARS equals the number of steps, i.e., $df(\hat{\boldsymbol{\mu}}_k) = k$, where $\hat{\boldsymbol{\mu}}_k$ is the fit at step k. Since the lasso and LARS coincide under the positive cone condition, the remarkable formula also holds for the lasso. Under general situations $df(\hat{\boldsymbol{\mu}}_k)$ is still well approximated by k for LARS. However, this simple approximation cannot be true in general for the lasso because the total number of lasso steps can exceed the number of predictors. This usually happens when some variables are temporally dropped (coefficients cross zero) during the LARS process, and they are eventually included into the full OLS model. For instance, the LARS algorithm takes 12

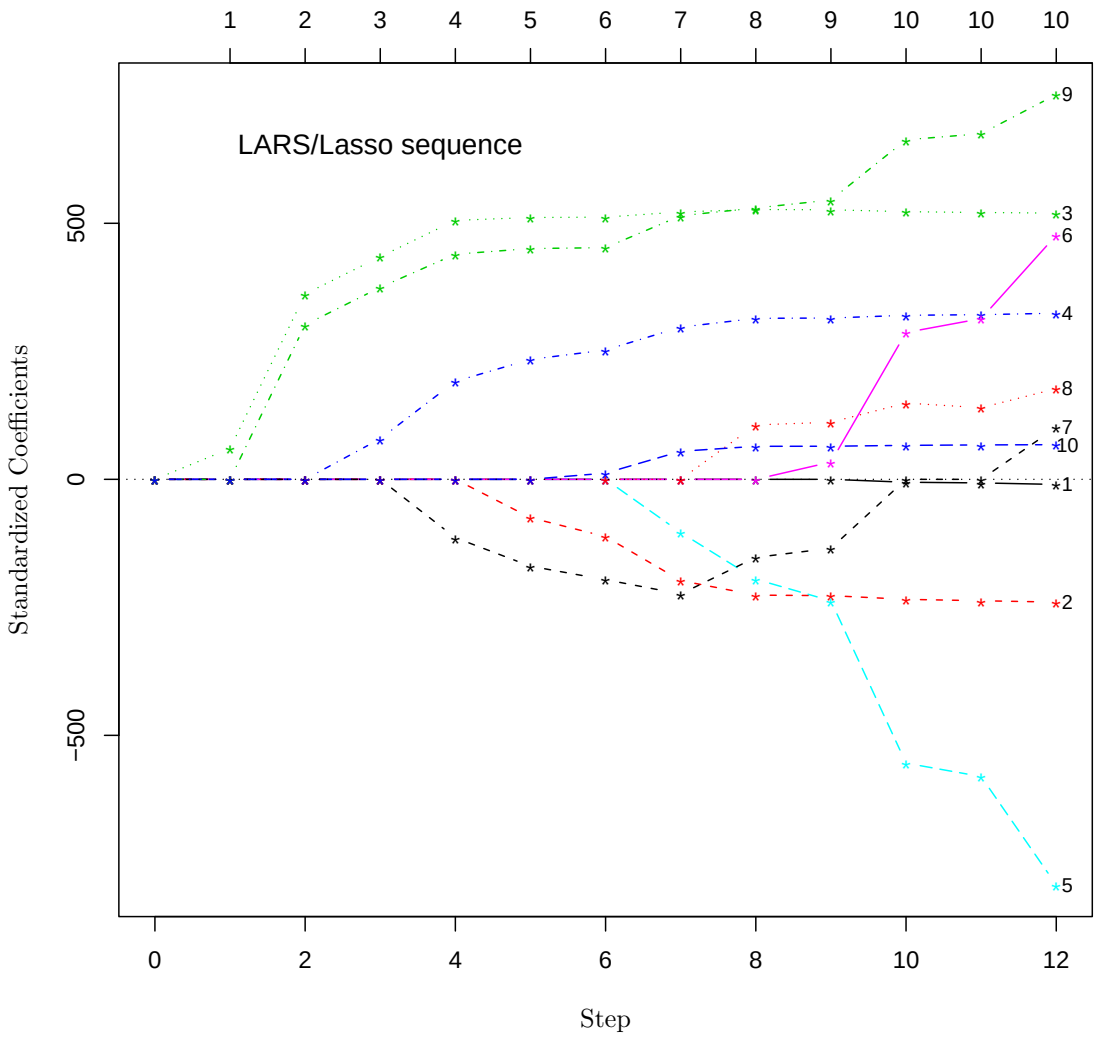

Figure 4.2: *Diabetes data with 10 predictors: the growth paths of the lasso coefficients estimates as the LARS algorithm moves forward. On the top of the graph, we display the number of non-zero coefficients at each step.*

lasso steps to reach the OLS fit as shown in Figure 4.2, but the number of predictors is 10. For the degrees of freedom of the lasso under general conditions, Efron, Hastie, Johnstone & Tibshirani (2004) presented the following conjecture.

Conjecture [EHJT04]: *Starting at step 0, let m_k be the index of the last model in the lasso sequence containing k predictors. Then $df(\hat{\boldsymbol{\mu}}_{m_k}) \doteq k$.*

In this Chapter we study the degrees of freedom of the lasso using Stein's unbiased risk estimation (SURE) theory (Stein 1981). The lasso exhibits the backward penalization and forward growth pictures, which consequently induces two different ways to describe its degrees of freedom. With the representation (4.2), we show that for any given λ the number of non-zero predictors in the model is an unbiased estimate for the degrees of freedom, and no special assumption on the predictors is required, e.g. the positive cone condition. The right panel in Figure 4.1 displays the unbiased estimate for the degrees of freedom as a function of λ on diabetes data (with 10 predictors). If the lasso is viewed as a forward stage-wise process, our analysis provides mathematical support for the above conjecture.

We first briefly review the SURE theory in Section 4.2. Main results and proofs are presented in Section 4.3. In Section 4.4, model selection criteria are constructed using the degrees of freedom to adaptively select the optimal lasso fit. We address the difference between two types of optimality: adaptive in prediction and adaptive in variable selection. Discussions are in Section 4.5. Proofs of lemmas are presented in Section 4.6.

4.2 Stein's Unbiased Risk Estimation

We begin with a brief introduction to the Stein's unbiased risk estimation (SURE) theory (Stein 1981) which is the foundation of our analysis. The readers are referred to Efron (2004) for detailed discussions and recent references on SURE.

Given a model fitting method δ, let $\hat{\boldsymbol{\mu}} = \delta(\mathbf{y})$ represent its fit. We assume a homoskedastic model, i.e., given the $\mathbf{x}$'s, $\mathbf{y}$ is generated according to

$$\mathbf{y} \sim (\boldsymbol{\mu}, \sigma^2 \mathbf{I}), \tag{4.3}$$

where $\boldsymbol{\mu}$ is the true mean vector and σ^2 is the common variance. The focus is how accurate δ can be in predicting future data. Suppose $\mathbf{y}^{new}$ is a new response vector generated from (4.3), then under the squared-error loss, the prediction risk is $E\Big\{\|\hat{\boldsymbol{\mu}} - \mathbf{y}^{new}\|^2\Big\}/n$. Efron (2004) shows that

$$E\{\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2\} = E\{\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2 - n\sigma^2\} + 2\sum_{i=1}^{n} \mathrm{cov}(\hat{\boldsymbol{\mu}}_i, y_i). \tag{4.4}$$

The last term of (4.4) is called the *optimism* of the estimator $\hat{\boldsymbol{\mu}}$ (Efron 1986). Identity (4.4) also gives a natural definition of the *degrees of freedom* for an estimator $\hat{\boldsymbol{\mu}} = \delta(\mathbf{y})$,

$$df(\hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} \mathrm{cov}(\hat{\boldsymbol{\mu}}_i, y_i)/\sigma^2. \tag{4.5}$$

If δ is a linear smoother, i.e., $\hat{\boldsymbol{\mu}} = \mathbf{S}\mathbf{y}$ for some matrix $\mathbf{S}$ independent of $\mathbf{y}$, then it is easy to verify that since $\mathrm{cov}(\hat{\boldsymbol{\mu}}, \mathbf{y}) = \sigma^2 S$, $df(\hat{\boldsymbol{\mu}}) = \mathrm{tr}(\mathbf{S})$, which coincides with the definition given by Hastie & Tibshirani (1990). By (4.4) we obtain

$$E\{\|\hat{\boldsymbol{\mu}} - \mathbf{y}^{new}\|^2\} = E\{\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2 + 2df(\hat{\boldsymbol{\mu}})\ \sigma^2\}. \tag{4.6}$$

Thus we can define a C_p-type statistic

$$C_p(\hat{\boldsymbol{\mu}}) = \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2}{n} + \frac{2df(\hat{\boldsymbol{\mu}})}{n}\sigma^2 \tag{4.7}$$

which is an unbiased estimator of the true prediction error. When σ^2 is unknown, it is replaced with an unbiased estimate.

Stein proves an extremely useful formula to simplify (4.5), which is often referred to as Stein's Lemma (Stein 1981). According to Stein, a function $g : \mathbb{R}^n \to \mathbb{R}$ is said to be *almost differentiable* if there is a function $f : \mathbb{R}^n \to \mathbb{R}^n$ such that

$$g(x + u) - g(x) = \int_0^1 u^T f(x + tu)dt \tag{4.8}$$

for a.e. $x \in \mathbb{R}^n$, each $u \in \mathbb{R}^n$.

Lemma 4.1 (Stein's Lemma). *Suppose that $\hat{\boldsymbol{\mu}} : \mathbb{R}^n \to \mathbb{R}^n$ is almost differentiable and denote $\nabla \cdot \hat{\boldsymbol{\mu}} = \sum_{i=1}^n \partial \hat{\boldsymbol{\mu}}_i / \partial y_i$. If $\mathbf{y} \sim N_n(\boldsymbol{\mu}, \sigma^2 \mathbf{I})$, then*

$$\sum_{i=1}^{n} \operatorname{cov}(\hat{\boldsymbol{\mu}}_i, y_i)/\sigma^2 = E[\nabla \cdot \hat{\boldsymbol{\mu}}]. \tag{4.9}$$

In many applications $\nabla \cdot \hat{\boldsymbol{\mu}}$ is shown to be a constant; for example, with $\hat{\boldsymbol{\mu}} = S\mathbf{y}$, $\nabla \cdot \hat{\boldsymbol{\mu}} = \operatorname{tr}(S)$. Thus the degrees of freedom is easily obtained. Even if $\nabla \cdot \hat{\boldsymbol{\mu}}$ depends on y, Stein's Lemma says

$$\widehat{df}(\hat{\boldsymbol{\mu}}) = \nabla \cdot \hat{\boldsymbol{\mu}} \tag{4.10}$$

is an unbiased estimate for the degrees of freedom $df(\hat{\boldsymbol{\mu}})$. In the spirit of SURE, we can use

$$C_p^*(\hat{\boldsymbol{\mu}}) = \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2}{n} + \frac{2\widehat{df}(\hat{\boldsymbol{\mu}})}{n}\, \sigma^2 \tag{4.11}$$

as an unbiased estimate for the true risk. It is worth mentioning that in some situations verifying the almost differentiability of $\hat{\boldsymbol{\mu}}$ is not easy.

Even though Stein's Lemma assumes normality, the essence of (4.9) only requires homoskedasticity (4.3) and the almost differentiability of $\hat{\boldsymbol{\mu}}$; its justification can be made by a "delta method" argument (Efron, Hastie, Johnstone & Tibshirani 2004). After all, $df(\hat{\boldsymbol{\mu}})$ is about the self-influence of $\mathbf{y}$ on the fit, and $\nabla \cdot \hat{\boldsymbol{\mu}}$ is a natural candidate for that purpose. Meyer & Woodroofe (2000) discussed the degrees of freedom in shape-restricted regression and argued that the divergence formula (4.10) provides a measure of the effective dimension.

4.3 Main Theorems

We adopt the SURE framework with the lasso fit. Let $\hat{\boldsymbol{\mu}}_\lambda$ be the lasso fit using the representation (4.2). Similarly, let $\hat{\boldsymbol{\mu}}_m$ be the lasso fit at step m in the LARS algorithm. For convenience, we also let $df(\lambda)$ and $df(m)$ stand for $df(\hat{\boldsymbol{\mu}}_\lambda)$ and $df(\hat{\boldsymbol{\mu}}_m)$, respectively.

The following matrix representation of Stein's Lemma is helpful. Let $\frac{\partial \hat{\boldsymbol{\mu}}}{\partial \mathbf{y}}$ be a $n \times n$

matrix whose elements are

$$\left(\frac{\partial \hat{\boldsymbol{\mu}}}{\partial \mathbf{y}}\right)_{i,j} = \frac{\partial \hat{\boldsymbol{\mu}}_i}{\partial y_j} \quad i,j = 1,2,\ldots,n. \tag{4.12}$$

Then we can write

$$\nabla \cdot \hat{\boldsymbol{\mu}} = \mathrm{tr}\left(\frac{\partial \hat{\boldsymbol{\mu}}}{\partial \mathbf{y}}\right). \tag{4.13}$$

Suppose $\mathbf{M}$ is a matrix with p columns. Let $\mathcal{S}$ be a subset of the indices $\{1,2,\ldots,p\}$. Denote by $\mathbf{M}_{\mathcal{S}}$ the sub-matrix

$$\mathbf{M}_{\mathcal{S}} = [\cdots m_j \cdots]_{j\in\mathcal{S}}, \tag{4.14}$$

where m_j is the j-th column of $\mathbf{M}$. Similarly, define $\beta_{\mathcal{S}} = (\cdots \beta_j \cdots)_{j\in\mathcal{S}}$ for any vector β of length p. Let $\mathrm{Sgn}(\cdot)$ be the sign function:

$$\mathrm{Sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

4.3.1 Results and data examples

Our results are stated as follows. Denote the set of non-zero elements of $\hat{\beta}_\lambda$ as $\mathcal{B}(\lambda)$, then

$$df(\lambda) = E[|\mathcal{B}_\lambda|] \tag{4.15}$$

where $|\mathcal{B}_\lambda|$ means the size of $\mathcal{B}(\lambda)$. Hence $\widehat{df}(\lambda) = |\mathcal{B}_\lambda|$ is an unbiased estimate for $df(\lambda)$. The identity (4.15) holds for all $\mathbf{X}$, requiring no special assumption.

We also provide mathematical support for the conjecture in Section 2.1. Actually we argue that if m_k is a lasso step containing k non-zero predictors, then $\widehat{df}(m_k) = k$ is a good estimate for $df(m_k)$. Note that m_k is not necessary the last lasso step containing k non-zero predictors. So the result includes the conjecture as a special case. However, we

show in Section 4.4 that the last step choice is superior in the lasso model selection. We let m_k^{last} and m_k^{first} denote the last and first lasso step containing exact k non-zero predictors, respectively.

Before delving into the detail of theoretical analysis, we check the validity of our arguments by a simulation study. Here is the outline of the simulation. We take the 64 predictors in the diabetes data which include the quadratic terms and interactions of the original 10 predictors. The positive cone condition is violated on the 64 predictors (Efron, Hastie, Johnstone & Tibshirani 2004). The response vector $\mathbf{y}$ was used to fit a OLS model. We computed the OLS estimates $\hat{\beta}_{ols}$ and $\hat{\sigma}^2_{ols}$. Then we considered a synthetic model

$$\mathbf{y}^* = \mathbf{X}\beta + N(0,1)\sigma, \tag{4.16}$$

where $\beta = \hat{\beta}_{ols}$ and $\sigma = \hat{\sigma}_{ols}$.

Given the synthetic model, the degrees of freedom of the lasso (both $df(\lambda)$ and $df(m_k)$) can be numerically evaluated by Monte Carlo methods. For $b = 1, 2, \ldots, B$, we independently simulated $\mathbf{y}^*(b)$ from (4.16). For a given λ, by the definition of $df(\lambda)$, we need to evaluate

$$\text{cov}_i = E[(\hat{\boldsymbol{\mu}}_{\lambda,i} - E[\hat{\boldsymbol{\mu}}_{\lambda,i}])(\mathbf{y}_i^* - (\mathbf{X}\beta)_i)]. \tag{4.17}$$

Then $df(\lambda) = \sum_{i=1}^n \text{cov}_i/\sigma^2$. Since $E[\mathbf{y}_i^*] = (\mathbf{X}\beta)_i$ and note that

$$\text{cov}_i = E[(\hat{\boldsymbol{\mu}}_{\lambda,i} - a_i)(\mathbf{y}_i^* - (\mathbf{X}\beta)_i)] \tag{4.18}$$

for any fixed known constant a_i. Then we compute

$$\widehat{\text{cov}}_i = \frac{\sum_{b=1}^B \left(\hat{\boldsymbol{\mu}}_{\lambda,i}(b) - a_i\right)\left(\mathbf{y}_i^*(b) - (\mathbf{X}\beta)_i\right)}{B} \tag{4.19}$$

and $df(\lambda) = \sum_{i=1}^n \widehat{\text{cov}}_i/\sigma^2$. Typically $a_i = 0$ is used in Monte Carlo calculation. In this work we use $a_i = (\mathbf{X}\beta)_i$, for it gives a Monte Carlo estimate for $df(\lambda)$ with smaller variance than that by $a_i = 0$. On the other hand, for a fixed λ, each $\mathbf{y}^*(b)$ gave the lasso fit $\hat{\boldsymbol{\mu}}_\lambda(b)$

and the df estimate $\widehat{df}(\lambda)_b$. Then we evaluated $E[|\mathcal{B}_\lambda|]$ by $\sum_{b=1}^{B} \widehat{df}(\lambda)_b/B$. Similarly, we computed $df(m_k)$ by replacing $\hat{\boldsymbol{\mu}}_\lambda(b)$ with $\hat{\boldsymbol{\mu}}_{m_k}(b)$. We are interested in $E[|\mathcal{B}_\lambda|] - df(\lambda)$ and $k - df(m_k)$. Standard errors were calculated based on the B replications.

Figure 4.3 is a very convincing picture for the identity (4.15). Figure 4.4 shows that $df(m_k)$ is well approximated by k even when the positive cone condition is failed. The simple approximation works pretty well for both m_k^{last} and m_k^{first}.

In Figure 4.4, it appears that $k - df(m_k)$ is not exactly zero for some k. We would like to check if the bias is real. Furthermore, if the bias is real, then we would like to explore the relation between the bias $k - df(m_k)$ and the signal/noise ratio. In the synthetic model (4.16) the signal/noise ratio $\frac{\text{Var}(\mathbf{X}\hat{\beta}_{ols})}{\hat{\sigma}^2_{ols}}$ is about 1.25. We repeated the same simulation procedure with $(\beta = 0, \sigma = 1)$ and $(\beta = \hat{\beta}_{ols}, \sigma = \frac{\hat{\sigma}_{ols}}{10})$ in the synthetic model. The corresponding signal/noise ratios are zero and 125, respectively. Thus the simulation covers broad scenarios.

As shown clearly in Figure 4.5, the bias $k - df(m_k)$ is truly non-zero for some k. Thus the positive cone condition seems to be sufficient and necessary for turning the approximation into an exact result. However, even if the bias exists, its maximum magnitude is less than one, regardless the size of the signal/noise ratio. So k is a very good estimate for $df(m_k)$. An interesting observation is that k tends to underestimate $df(m_k^{\text{last}})$ and overestimate $df(m_k^{\text{first}})$. In addition, we observe that $k - df(m_k^{\text{last}}) \doteq df(m_k^{\text{first}}) - k$.

4.3.2 Theorems on $df(\lambda)$

Let $\mathcal{B} = \{j : \text{Sgn}(\beta)_j \neq 0\}$ be the *active set* of β where $\text{Sgn}(\beta)$ is the sign vector of β given by $\text{Sgn}(\beta)_j = \text{Sgn}(\beta_j)$. We denote the active set of $\hat{\beta}(\lambda)$ as $\mathcal{B}(\lambda)$ and the corresponding sign vector $\text{Sgn}(\hat{\beta}(\lambda))$ as $\text{Sgn}(\lambda)$. We do not distinguish the index of a predictor and the predictor itself.

Firstly, let us review some characteristics of the lasso solution. For a given response vector $\mathbf{y}$, there are a sequence of λ's:

$$\lambda_0 > \lambda_1 > \lambda_2 \cdots > \lambda_K = 0 \quad \text{such that:} \tag{4.20}$$

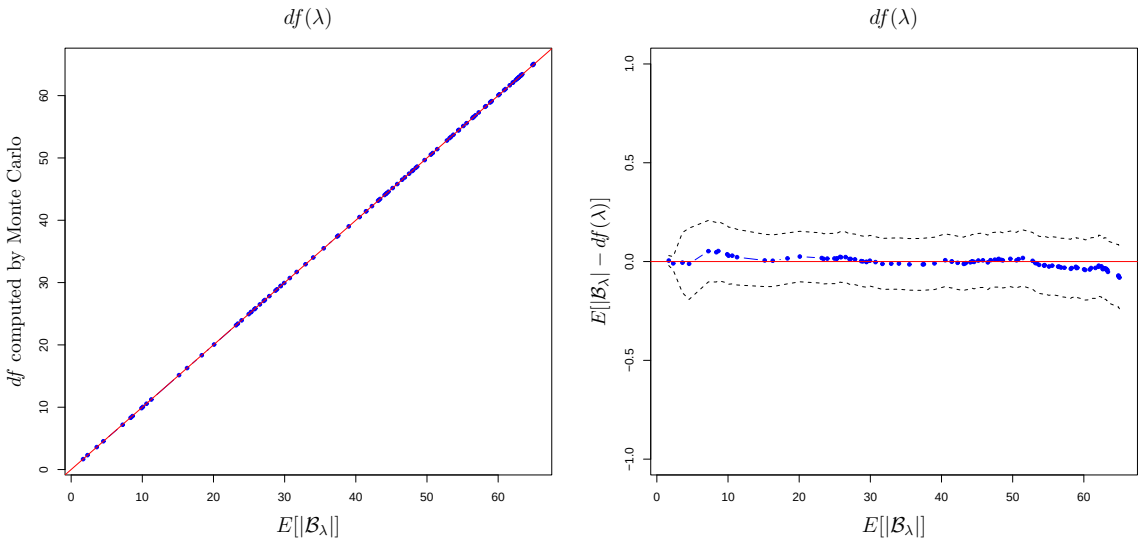

Figure 4.3: *The synthetic model with the 64 predictors in the diabetes data. In the left panel we compare* $E[|\mathcal{B}_\lambda|]$ *with the true degrees of freedom* $df(\lambda)$ *based on* $B = 20000$ *Monte Carlo simulations. The solid line is the* 45^0 *line (the perfect match). The right panel shows the estimation bias and its point-wise* 95% *confidence intervals indicated by the thin dashed lines. Note that the zero horizontal line is well inside the confidence intervals.*

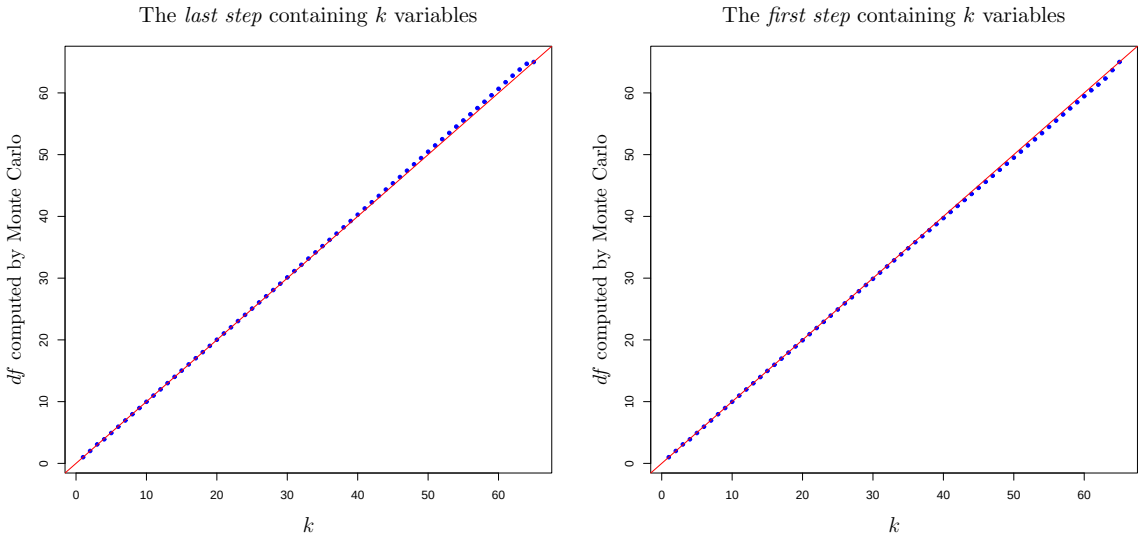

Figure 4.4: *The synthetic model with the 64 predictors in the diabetes data. We compare* $\widehat{df}(m_k)$ *with the true degrees of freedom* $df(m_k)$ *based on* $B = 20000$ *Monte Carlo simulations. We consider two choices of* m_k*: in the left panel* m_k *is the last lasso step containing exact* k *non-zero variables, while the right panel chooses the first lasso step containing exact* k *non-zero variables. As can be seen from the plots, our formula works pretty well in both cases.*

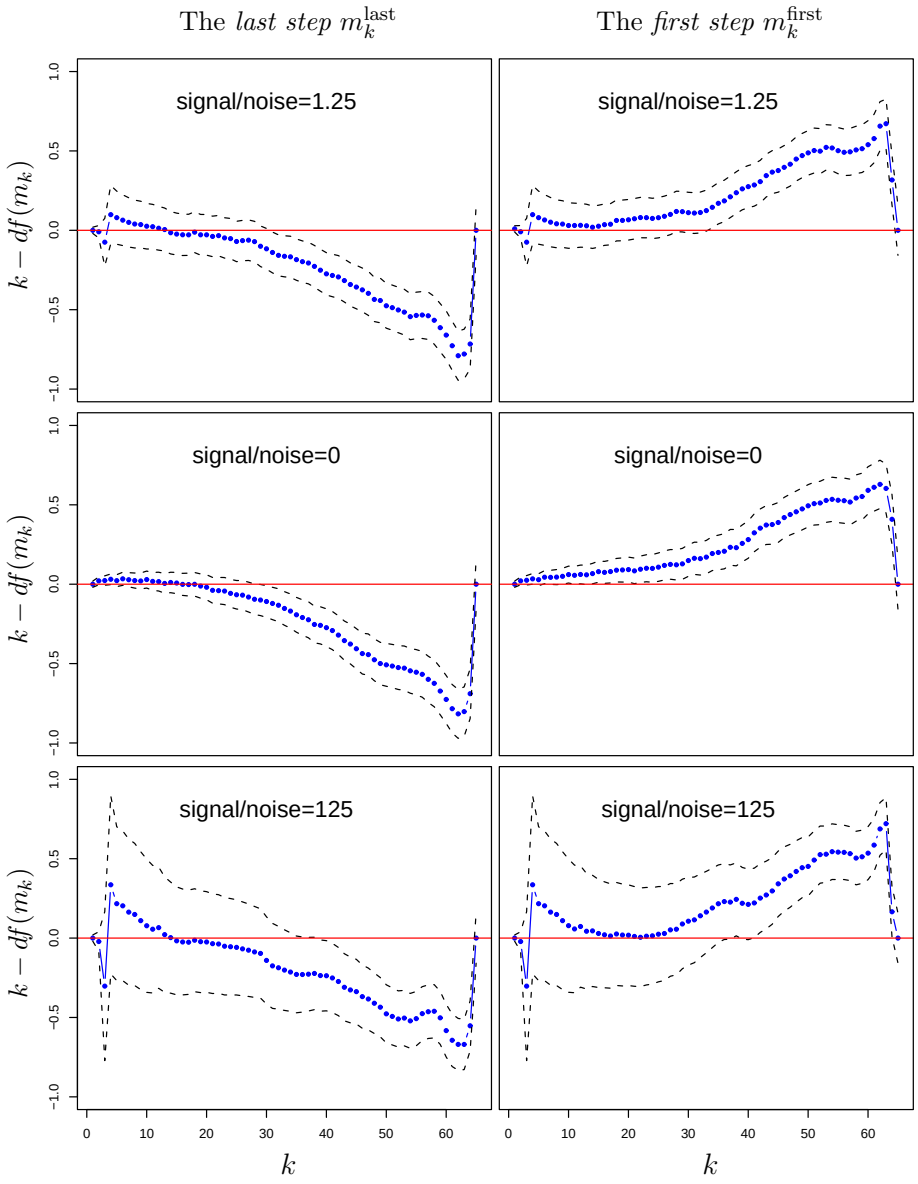

Figure 4.5: $B = 20000$ *replications were used to assess the bias of* $\widehat{df}(m_k) = k$. *The* 95% *point-wise confidence intervals are indicated by the thin dashed lines. Under the positive cone condition, it is exactly the true degrees of freedom* $df(m_k)$. *This simulation suggests that when the positive cone condition is violated,* $df(m_k) \neq k$ *for some* k. *However, the bias is small (the maximum absolute bias is about 0.8). It seems that* k *tends to underestimate* $df(m_k^{\text{last}})$ *and overestimate* $df(m_k^{\text{first}})$. *In addition, we observe that* $k - df(m_k^{\text{last}}) \doteq df(m_k^{\text{first}}) - k$. *The most important message is that the magnitude of the bias is always less than one, regardless the size of the signal/noise ratio.*

- For all $\lambda > \lambda_0$, $\hat{\beta}(\lambda) = 0$.
- In the interior of the interval $(\lambda_{m+1}, \lambda_m)$, the active set $\mathcal{B}(\lambda)$ and the sign vector $\mathrm{Sgn}(\lambda)_{\mathcal{B}(\lambda)}$ are constant with respect to λ. Thus we write them as $\mathcal{B}_m$ and Sgn_m for convenience.

The active set changes at each λ_m. When λ decreases from $\lambda = \lambda_m - 0$, some predictors with zero coefficients at λ_m are about to have non-zero coefficients, thus they join the active set $\mathcal{B}_m$. However, as λ approaches $\lambda_{m+1} + 0$ there are possibly some predictors in $\mathcal{B}_m$ whose coefficients reach zero. Hence we call $\{\lambda_m\}$ the *transition points*.

We shall proceed by proving the following lemmas (proofs are given in the appendix).

Lemma 4.2. *Suppose $\lambda \in (\lambda_{m+1}, \lambda_m)$. $\hat{\beta}(\lambda)$ are the lasso coefficient estimates. Then we have*

$$\hat{\beta}(\lambda)_{\mathcal{B}_m} = \left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m}\right)^{-1} \left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{y} - \frac{\lambda}{2} \mathrm{Sgn}_m\right). \tag{4.21}$$

Lemma 4.3. *Consider the transition points λ_m and λ_{m+1}, $\lambda_{m+1} \geq 0$. $\mathcal{B}_m$ is the active set in $(\lambda_{m+1}, \lambda_m)$. Suppose i_{add} is an index added into $\mathcal{B}_m$ at λ_m and its index in $\mathcal{B}_m$ is i^*, i.e., $i_{add} = (\mathcal{B}_m)_{i^*}$. Denote by $(a)_k$ the k-th element of the vector a. We can express the transition point λ_m as follows:*

$$\lambda_m = \frac{2\left(\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m}\right)^{-1} \mathbf{X}_{\mathcal{B}_m}^T \mathbf{y}\right)_{i^*}}{\left(\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m}\right)^{-1} \mathrm{Sgn}_m\right)_{i^*}} \tag{4.22}$$

Moreover, if j_{drop} is a dropped (if there is any) index at λ_{m+1} and $j_{drop} = (\mathcal{B}_m)_{j^*}$, then λ_{m+1} can be written as:

$$\lambda_{m+1} = \frac{2\left(\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m}\right)^{-1} \mathbf{X}_{\mathcal{B}_m}^T \mathbf{y}\right)_{j^*}}{\left(\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m}\right)^{-1} \mathrm{Sgn}_m\right)_{j^*}} \tag{4.23}$$

Lemma 4.4. *$\forall\ \lambda > 0$, $\exists$ a null set $\mathcal{N}_\lambda$ which is a finite collection of hyperplanes in $\mathbb{R}^n$. Let $\mathcal{G}_\lambda = \mathbb{R}^n \setminus \mathcal{N}_\lambda$. Then $\forall \mathbf{y}\ \in \mathcal{G}_\lambda$, λ is not any of the transition points, i.e., $\lambda \notin \{\lambda(\mathbf{y})_m\}$.*

Lemma 4.5. *$\forall\lambda$, $\hat{\beta}_\lambda(\mathbf{y})$ is a continuous function of $\mathbf{y}$.*

Lemma 4.6. *Fix any $\lambda > 0$, consider $\mathbf{y} \in \mathcal{G}_\lambda$ as defined in Lemma 4.4. The active set $\mathcal{B}(\lambda)$ and the sign vector $\mathrm{Sgn}(\lambda)$ are locally constant with respect to $\mathbf{y}$.*

Theorem 4.1. *Let $\mathcal{G}_0 = \mathbb{R}^n$. Fix an arbitrary $\lambda \geq 0$. On the set $\mathcal{G}_\lambda$ with full measure as defined in Lemma 4.4, the lasso fit $\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y})$ is uniformly Lipschitz. Precisely,*

$$\|\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y}+\Delta\mathbf{y}) - \hat{\boldsymbol{\mu}}_\lambda(\mathbf{y})\| \leq \|\Delta\mathbf{y}\| \quad \textit{for sufficiently small } \Delta\mathbf{y} \tag{4.24}$$

Moreover, we have the divergence formula

$$\nabla \cdot \hat{\boldsymbol{\mu}}_\lambda(\mathbf{y}) = |\mathcal{B}_\lambda|. \tag{4.25}$$

Proof. If $\lambda = 0$, then the lasso fit is just the OLS fit. The conclusions are easy to verify. So we focus on $\lambda > 0$. Fix a $\mathbf{y}$. Choose a small enough ϵ such that $\mathrm{Ball}(\mathbf{y}, \epsilon) \subset \mathcal{G}_\lambda$.

Since λ is not any transition point, using (4.21) we observe

$$\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y}) = \mathbf{X}\hat{\beta}(\mathbf{y}) = \mathbf{H}_\lambda(\mathbf{y})\mathbf{y} - \lambda\boldsymbol{\omega}_\lambda(\mathbf{y}), \tag{4.26}$$

where $\mathbf{H}_\lambda(\mathbf{y}) = \mathbf{X}_{\mathcal{B}_\lambda}(\mathbf{X}_{\mathcal{B}_\lambda}^T\mathbf{X}_{\mathcal{B}_\lambda})^{-1}\mathbf{X}_{\mathcal{B}_\lambda}^T$ is the projection matrix on the space $\mathbf{X}_{\mathcal{B}_\lambda}$ and $\omega_\lambda(\mathbf{y}) = \frac{1}{2}\,\mathbf{X}_{\mathcal{B}_\lambda}(\mathbf{X}_{\mathcal{B}_\lambda}^T\mathbf{X}_{\mathcal{B}_\lambda})^{-1}\mathrm{Sgn}_{\mathcal{B}_\lambda}$. Consider $\|\Delta\mathbf{y}\| < \epsilon$. Similarly, we get

$$\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y}+\Delta\mathbf{y}) = \mathbf{H}_\lambda(\mathbf{y}+\Delta\mathbf{y})(\mathbf{y}+\Delta\mathbf{y}) - \lambda\boldsymbol{\omega}_\lambda(\mathbf{y}+\Delta\mathbf{y}). \tag{4.27}$$

Lemma 4.6 says that we can further let ϵ be sufficiently small such that both the effective set $\mathcal{B}_\lambda$ and the sign vector Sgn_λ stay constant in $\mathrm{Ball}(\mathbf{y}, \epsilon)$. Now fix ϵ. Hence if $\|\Delta\mathbf{y}\| < \epsilon$, then

$$\mathbf{H}_\lambda(\mathbf{y}+\Delta\mathbf{y}) = \mathbf{H}_\lambda(\mathbf{y}) \quad \text{and} \quad \boldsymbol{\omega}_\lambda(\mathbf{y}+\Delta\mathbf{y}) = \boldsymbol{\omega}_\lambda(\mathbf{y}). \tag{4.28}$$

Then (4.26) and (4.27) give

$$\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y}+\Delta\mathbf{y}) - \hat{\boldsymbol{\mu}}_\lambda(\mathbf{y}) = \mathbf{H}_\lambda(\mathbf{y})\Delta\mathbf{y}. \tag{4.29}$$

But since $\|\mathbf{H}_\lambda(\mathbf{y})\Delta\mathbf{y}\| \leq \|\Delta\mathbf{y}\|$, (4.24) is proved.

By the local constancy of $H(\mathbf{y})$ and $\omega(\mathbf{y})$, we have

$$\frac{\partial\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y})}{\partial\mathbf{y}} = \mathbf{H}_\lambda(\mathbf{y}). \tag{4.30}$$

Then the trace formula (4.13) implies

$$\nabla\cdot\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y}) = \mathrm{tr}\left(\mathbf{H}_\lambda(\mathbf{y})\right) = |\mathcal{B}_\lambda|. \tag{4.31}$$

□

By standard analysis arguments, it is easy to check the following proposition

Proposition *Let $f:\mathbb{R}^n\to\mathbb{R}^n$ and suppose f is uniformly Lipschitz on $\mathcal{G}=\mathbb{R}^n\setminus\mathcal{N}$ where $\mathcal{N}$ is a finite set of hyperplanes. If f is continuous, then f is uniformly Lipschitz on $\mathbb{R}^n$.*

Theorem 4.2. *$\forall\ \lambda$ the lasso fit $\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y})$ is uniformly Lipschitz. The degrees of freedom of $\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y})$ equal the expectation of the effective set $\mathcal{B}_\lambda$, i.e.,*

$$df(\lambda) = E\left[|\mathcal{B}_\lambda|\right]. \tag{4.32}$$

Proof. The proof is trivial for $\lambda = 0$. We only consider $\lambda > 0$. By Theorem 4.1 and the proposition, we conclude that $\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y})$ is uniformly Lipschitz. Therefore $\hat{\boldsymbol{\mu}}_\lambda(\mathbf{y})$ is almost differentiable, see Meyer & Woodroofe (2000) and Efron, Hastie, Johnstone & Tibshirani (2004). Then (4.32) is obtained by Stein's Lemma and the divergence formula (4.25). □

4.3.3 $df(m_k)$ and the conjecture

In this section we provide mathematical support for the conjecture in Section 2.1. The conjecture becomes a simple fact for two trivial cases $k = 0$ and $k = p$, thus we only need to consider $k = 1,\ldots,(p-1)$. Our arguments rely on the details of the LARS algorithm. For the sake of clarity, we first briefly describe the LARS algorithm. The readers are referred to the LAR paper (Efron, Hastie, Johnstone & Tibshirani 2004) for the complete description.

The LARS algorithm sequentially updates the lasso estimate in a predictable way. Initially (the 0 step), let $\hat{\beta}_0 = 0,\ A_0 = \varnothing$. Suppose that $\hat{\beta}_m$ is the vector of current lasso coefficient estimates. Then $\hat{\boldsymbol{\mu}}_m = \mathbf{X}\hat{\beta}_m$ and $\hat{r}_m = \mathbf{y} - \hat{\boldsymbol{\mu}}_m$ are the current fit and residual vectors. We say $\hat{c} = \mathbf{X}^T\hat{r}_m$ is the vector of current correlations. Define

$$\widehat{C} = \max_j\{|\hat{c}|\} \quad \mathcal{W}_m = \{j : |\hat{c}_j| = \widehat{C} \text{ and } j \in A_m^c\}. \tag{4.33}$$

Then $\lambda_m = 2\widehat{C}$. Define the current active set $\mathcal{A} = A_m \cup \mathcal{W}_m$ and the signed matrix

$$X_{\mathcal{A}}^{\text{sign}} = (\cdots \text{Sgn}(\hat{c}_j)\mathbf{x}_j \cdots)_{j\in\mathcal{A}}. \tag{4.34}$$

Let $\mathcal{G}_{\mathcal{A}} = \left(X_{\mathcal{A}}^{\text{sign}}\right)^T X_{\mathcal{A}}^{\text{sign}}$. $1_{\mathcal{A}}$ is a vector of 1's of length $|\mathcal{A}|$. Then we compute the *equiangular vector*

$$\mathbf{u}_{\mathcal{A}} = X_{\mathcal{A}}^{\text{sign}} w_{\mathcal{A}} \quad \text{with} \quad w_{\mathcal{A}} = DG_{\mathcal{A}}^{-1}1_{\mathcal{A}}, \tag{4.35}$$

where $D = (1_{\mathcal{A}}^T\mathcal{G}_{\mathcal{A}}^{-1}1_{\mathcal{A}})^{-\frac{1}{2}}$. Let the inner product vector $\mathbf{a} = \mathbf{X}^T\mathbf{u}_{\mathcal{A}}$ and

$$\widehat{\gamma} = \min_{j\in\mathcal{A}^c}{}^{+}\left\{\frac{\widehat{C} - \widehat{c}_j}{D - a_j}, \frac{\widehat{C} + \widehat{c}_j}{D + a_j}\right\}. \tag{4.36}$$

For $j \in \mathcal{A}$ we compute $d_j = \text{Sgn}(\hat{c}_j)w_{\mathcal{A}_j}$ and $\gamma_j = -(\hat{\beta}_m)_j/d_j$. Define

$$\widetilde{\gamma} = \min_{\gamma_j>0}\{\gamma_j\} \quad \text{and} \quad \mathcal{V}_m = \{j : \gamma_j = \widetilde{\gamma}\ j \in \mathcal{A}\}. \tag{4.37}$$

The lasso coefficient estimates are updated by

$$(\hat{\beta}_{m+1})_j = (\hat{\beta}_m)_j + \min\{\widehat{\gamma}, \widetilde{\gamma}\}d_j \quad \text{for } j \in \mathcal{A}. \tag{4.38}$$

The set A_m is also updated. If $\widehat{\gamma} < \widetilde{\gamma}$ then $A_{m+1} = \mathcal{A}$. Otherwise $A_{m+1} = \mathcal{A}\backslash\mathcal{V}_m$.

Let q_m be the indicator of whether $\mathcal{V}_m$ is dropped or not. Define $q_m\mathcal{V}_m = \mathcal{V}_m$ if $q_m = 1$, otherwise $q_m\mathcal{V}_m = \varnothing$; and conventionally let $\mathcal{V}_{-1} = \varnothing$ and $q_{-1}\mathcal{V}_{-1} = \varnothing$. Considering the

active set $\mathcal{B}_\lambda$ as a function of λ, we summarize the following facts

$$|\mathcal{B}_\lambda| = |\mathcal{B}_{\lambda_m}| + |\mathcal{W}_m| \quad \text{if } \lambda_m < \lambda < \lambda_{m+1}, \tag{4.39}$$

$$|\mathcal{B}_{\lambda_{m+1}}| = |\mathcal{B}_{\lambda_m}| + |\mathcal{W}_m| - |q_m \mathcal{V}_m|. \tag{4.40}$$

In the LARS algorithm, the lasso is regarded as one kind of forward stage-wise method for which the number of steps is often used as an effective regularization parameter. For each k, $k \in \{1, 2, \ldots, (p-1)\}$, we seek the models with k non-zero predictors. Let

$$\Lambda_k = \{m : |\mathcal{B}_{\lambda_m}| = k\}. \tag{4.41}$$

The conjecture is asking for the fit using $m_k^{\text{last}} = \sup(\Lambda_k)$. However, it may happen that for some k there is no such m with $|\mathcal{B}_{\lambda_m}| = k$. For example, if $\mathbf{y}$ is an equiangular vector of all $\{\mathbf{X}_j\}$, then the lasso estimates become the OLS estimates after just one step. So $\Lambda_k = \varnothing$ for $k = 2, \ldots, (p-1)$. The next Lemma concerns this type of situation. Basically, it shows that the *"one at a time"* condition (Efron et al. 2004) holds almost everywhere, therefore Λ_k is not empty for all k a.s.

Lemma 4.7. *$\exists$ a set $\widetilde{\mathcal{N}}_0$ which is a collection of finite many hyperplanes in $\mathbb{R}^n$. $\forall \mathbf{y} \in \mathbb{R}^n \setminus \widetilde{\mathcal{N}}_0$,*

$$|\mathcal{W}_m(\mathbf{y})| = 1 \quad \textit{and} \quad |q_m \mathcal{V}_m(\mathbf{y})| \leq 1 \quad \forall\ m = 0, 1, \ldots, K(\mathbf{y}). \tag{4.42}$$

Corollary 4.1. *$\forall\ \mathbf{y} \in \mathbb{R}^n \setminus \widetilde{\mathcal{N}}_0$, Λ_k is not empty for all k, $k = 0, 1, \ldots, p$.*

Proof. This is a direct consequence of Lemma 4.7 and (4.39), (4.40). □

The next theorem presents an expression for the lasso fit at each transition point, which helps us compute the divergence of $\hat{\boldsymbol{\mu}}_{m_k}(\mathbf{y})$.

Theorem 4.3. *Let $\hat{\boldsymbol{\mu}}_m(\mathbf{y})$ be the lasso fit at the transition point λ_m, $\lambda_m > 0$. Then for*

any $i \in \mathcal{W}_m$, we can write $\hat{\boldsymbol{\mu}}(m)$ as follows

$$\hat{\boldsymbol{\mu}}_m(\mathbf{y}) = \left\{ \mathbf{H}_{\mathcal{B}(\lambda_m)} - \frac{\mathbf{X}^T_{\mathcal{B}(\lambda_m)} \left(\mathbf{X}^T_{\mathcal{B}(\lambda_m)} \mathbf{X}_{\mathcal{B}(\lambda_m)} \right) \operatorname{Sgn}(\lambda_m) \mathbf{x}_i^T (\mathbf{I} - \mathbf{H}_{\mathcal{B}(\lambda_m)})}{\operatorname{Sgn}_i - \mathbf{x}_i^T \mathbf{X}^T_{\mathcal{B}(\lambda_m)} \left(\mathbf{X}^T_{\mathcal{B}(\lambda_m)} \mathbf{X}_{\mathcal{B}(\lambda_m)} \right) \operatorname{Sgn}(\lambda_m)} \right\} \mathbf{y} \quad (4.43)$$

$$=: \mathbf{S}_m(\mathbf{y})\mathbf{y} \quad (4.44)$$

where $\mathbf{H}_{\mathcal{B}(\lambda_m)}$ is the projection matrix on the subspace of $\mathbf{X}_{\mathcal{B}(\lambda_m)}$. Moreover

$$\operatorname{tr}\left(\mathbf{S}_m(\mathbf{y})\right) = |\mathcal{B}(\lambda_m)|. \quad (4.45)$$

Proof. Note that $\hat{\beta}(\lambda)$ is continuous on λ. Using (18) in Lemma 2 and taking the limit of $\lambda \to \lambda_m$, we have

$$-2\mathbf{x}_j^T \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \hat{\beta}(\lambda_m)_j \right) + \lambda_m \operatorname{Sgn}(\hat{\beta}(\lambda_m)_j) = 0, \quad \text{for } j \in \mathcal{B}(\lambda_m). \quad (4.46)$$

However, $\sum_{j=1}^{p} \mathbf{x}_j \hat{\beta}(\lambda_m)_j = \sum_{j \in \mathcal{B}(\lambda_m)} \mathbf{x}_j \hat{\beta}(\lambda_m)_j$. Thus we have

$$\hat{\beta}(\lambda_m) = \left(\mathbf{X}^T_{\mathcal{B}(\lambda_m)} \mathbf{X}_{\mathcal{B}(\lambda_m)} \right)^{-1} \left(\mathbf{X}^T_{\mathcal{B}(\lambda_m)} \mathbf{y} - \frac{\lambda_m}{2} \operatorname{Sgn}(\lambda_m) \right). \quad (4.47)$$

Hence

$$\begin{aligned} \hat{\boldsymbol{\mu}}_m(\mathbf{y}) &= \mathbf{X}_{\mathcal{B}(\lambda_m)} \left(\mathbf{X}^T_{\mathcal{B}(\lambda_m)} \mathbf{X}_{\mathcal{B}(\lambda_m)} \right)^{-1} \left(\mathbf{X}^T_{\mathcal{B}(\lambda_m)} \mathbf{y} - \frac{\lambda_m}{2} \operatorname{Sgn}(\lambda_m) \right) \\ &= \mathbf{H}_{\mathcal{B}(\lambda_m)} \mathbf{y} - \mathbf{X}_{\mathcal{B}(\lambda_m)} \left(\mathbf{X}^T_{\mathcal{B}(\lambda_m)} \mathbf{X}_{\mathcal{B}(\lambda_m)} \right)^{-1} \operatorname{Sgn}(\lambda_m) \frac{\lambda_m}{2}. \end{aligned} \quad (4.48)$$

Since $i \in \mathcal{W}_m$, we must have the *equiangular* condition

$$\operatorname{Sgn}_i \mathbf{x}_i^T \left(\mathbf{y} - \hat{\boldsymbol{\mu}}(m) \right) = \frac{\lambda_m}{2}. \quad (4.49)$$

Substituting (4.48) into (4.49), we solve $\frac{\lambda_m}{2}$ and obtain

$$\frac{\lambda_m}{2} = \frac{\mathbf{x}_i^T \left(\mathbf{I} - \mathbf{H}_{\mathcal{B}(\lambda_m)}\right) \mathbf{y}}{\mathrm{Sgn}_i - \mathbf{x}_i^T \mathbf{X}_{\mathcal{B}(\lambda_m)}^T \left(\mathbf{X}_{\mathcal{B}(\lambda_m)}^T \mathbf{X}_{\mathcal{B}(\lambda_m)}\right) \mathrm{Sgn}(\lambda_m)}. \tag{4.50}$$

Then putting (4.50) back to (4.48) yields (4.43).

Using the identity $\mathrm{tr}(AB) = \mathrm{tr}(BA)$, we observe

$$\begin{aligned}
\mathrm{tr}\left(\mathbf{S}_m(\mathbf{y}) - \mathbf{H}_{\mathcal{B}(\lambda_m)}\right) &= \mathrm{tr}\left(\frac{\mathbf{X}_{\mathcal{B}(\lambda_m)}^T \left(\mathbf{X}_{\mathcal{B}(\lambda_m)}^T \mathbf{X}_{\mathcal{B}(\lambda_m)}\right) \mathrm{Sgn}(\lambda_m) \mathbf{x}_i^T (\mathbf{I} - \mathbf{H}_{\mathcal{B}(\lambda_m)})}{\mathrm{Sgn}_i - \mathbf{x}_i^T \mathbf{X}_{\mathcal{B}(\lambda_m)}^T \left(\mathbf{X}_{\mathcal{B}(\lambda_m)}^T \mathbf{X}_{\mathcal{B}(\lambda_m)}\right) \mathrm{Sgn}(\lambda_m)}\right) \\
&= \mathrm{tr}\left(\frac{\left(\mathbf{X}_{\mathcal{B}(\lambda_m)}^T \mathbf{X}_{\mathcal{B}(\lambda_m)}\right) \mathrm{Sgn}(\lambda_m) \mathbf{x}_i^T (\mathbf{I} - H_{\mathcal{B}(\lambda_m)}) \mathbf{X}_{\mathcal{B}(\lambda_m)}^T}{\mathrm{Sgn}_i - \mathbf{x}_i^T \mathbf{X}_{\mathcal{B}(\lambda_m)}^T \left(\mathbf{X}_{\mathcal{B}(\lambda_m)}^T \mathbf{X}_{\mathcal{B}(\lambda_m)}\right) \mathrm{Sgn}(\lambda_m)}\right) \\
&= \mathrm{tr}\,(0) = 0.
\end{aligned}$$

So $\mathrm{tr}\left(\mathbf{S}_m(\mathbf{y})\right) = \mathrm{tr}\left(\mathbf{H}_{\mathcal{B}(\lambda_m)}\right) = |\mathcal{B}(\lambda_m)|$. □

Definition 4.1. $\mathbf{y} \in \mathbb{R}^n \setminus \widetilde{\mathcal{N}}_0$ *is said to be a locally stable point for* Λ_k*, if* $\forall\ \mathbf{y}'$ *such that* $\|\mathbf{y}' - \mathbf{y}\| \leq \epsilon(\mathbf{y})$ *for a small enough* $\epsilon(\mathbf{y})$*, the effective set* $\mathcal{B}_{\lambda_m}(\mathbf{y}') = \mathcal{B}_{\lambda_m}(\mathbf{y})$*, for all* $m \in \Lambda_k$*. Let* $LS(\Lambda_k)$ *be the set of all locally stable points for* Λ_k*.*

Theorem 4.4. *If* $\mathbf{y} \in LS(\Lambda_k)$*, then we have the divergence formula* $\nabla \cdot \hat{\boldsymbol{\mu}}_m(\mathbf{y}) = k$ *which holds for all* $m \in \Lambda_k$ *including* $m_k = \sup(\Lambda_k)$*, the choice in the conjecture.*

Proof. The conclusion immediately follows definition 1 and Theorem 4.3. □

Points in $LS(\Lambda_k)$ are the majority of $\mathbb{R}^n$. Under the positive cone condition, $LS(\Lambda_k)$ is a set of full measure for all k. In fact the positive cone condition implies a stronger conclusion.

Definition 4.2. $\mathbf{y}$ *is said to be a strong locally stable point if* $\forall\ \mathbf{y}'$ *such that* $\|\mathbf{y}' - \mathbf{y}\| \leq \epsilon(\mathbf{y})$ *for a small enough* $\epsilon(\mathbf{y})$*, the effective set* $\mathcal{B}_{\lambda_m}(\mathbf{y}') = \mathcal{B}_{\lambda_m}(\mathbf{y})$*, for all* $m = 0, 1, \ldots, K(\mathbf{y})$*.*

Lemma 4.8. *Let* $\widetilde{\mathcal{N}}_1 = \left\{\mathbf{y} : \widehat{\gamma}(\mathbf{y}) = \widetilde{\gamma}(\mathbf{y}) \quad \text{for some } m, m \in \{0, 1, \ldots, K(\mathbf{y})\}\right\}$. $\forall\ \mathbf{y} \in$ *the interior of* $\mathbb{R}^n \setminus (\widetilde{\mathcal{N}}_0 \cup \widetilde{\mathcal{N}}_1)$, $\mathbf{y}$ *is a strong locally stable point. In particular, the positive cone condition implies* $\widetilde{\mathcal{N}}_1 = \varnothing$.

LARS is a discrete procedure by its definition, but the lasso is a continuous shrinkage method. So it also makes sense to talk about fractional lasso steps in the LARS algorithm, e.g. what is the lasso fit at 3.5 steps? Under the positive cone condition, we can generalize the result of Theorem 4 in the LAR paper to the case of non-integer steps.

Corollary 4.2. *Under the positive cone condition* $df(\hat{\boldsymbol{\mu}}_s) = s$ *for all real valued* s: $0 \le s \le p$.

Proof. Let $k \le s < k+1$, $s = k + r$ for some $r \in [0, 1)$. According to the LARS algorithm, the lasso fit is linearly interpolated between steps k and $k+1$. So $\hat{\boldsymbol{\mu}}_s = \hat{\boldsymbol{\mu}}_k \cdot (1-r) + \hat{\boldsymbol{\mu}}_{k+1} \cdot r$. Then by definition (4.5) and the fact cov is a linear operator, we have

$$\begin{aligned} df(\hat{\boldsymbol{\mu}}_s) &= df(\hat{\boldsymbol{\mu}}_k) \cdot (1-r) + df(\hat{\boldsymbol{\mu}}_{k+1}) \cdot r \\ &= k \cdot (1-r) + (k+1) \cdot r = s. \end{aligned} \tag{4.51}$$

In (4.51) we have used the positive cone condition and Theorem 4 in the LAR paper. □

4.4 Adaptive Lasso Shrinkage

4.4.1 Model selection criteria

For any regularization method an important issue is to find a good choice of the regularization parameter such that the corresponding model is optimal according to some criterion, e.g. minimizing the prediction risk. For this purpose, model selection criteria have been proposed in the literature to compare different models. Famous examples are AIC (Akaike 1973) and BIC (Schwartz 1978). Mallows's C_p (Mallows 1973) is very similar to AIC and a whole class of AIC or C_p-type criteria are provided by SURE theory (Stein 1981). In Efron (2004) C_p and SURE are summarized as covariance penalty methods for estimating the prediction

error, and are shown to offer substantially better accuracy than cross-validation and related nonparametric methods, if one is willing to assume the model is correct.

In the previous section we have derived the degrees of freedom of the lasso for both types of regularization: λ and m_k. Although the exact value of $df(\lambda)$ is still unknown, our formula provides a convenient unbiased estimate. In the spirit of SURE theory, the unbiased estimate for $df(\lambda)$ suffices to provide an unbiased estimate for the prediction error of $\hat{\boldsymbol{\mu}}_\lambda$. If we choose m_k as the regularization parameter, the good approximation $df(\hat{\boldsymbol{\mu}}_{m_k}) \doteq k$ also well serves the SURE purpose. Therefore an estimate for the prediction error of $\hat{\boldsymbol{\mu}}$ ($pe(\hat{\boldsymbol{\mu}})$) is

$$\widehat{pe}(\hat{\boldsymbol{\mu}}) = \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2}{n} + \frac{2}{n}\widehat{df}(\hat{\boldsymbol{\mu}})\ \sigma^2, \tag{4.52}$$

where $\widehat{df}$ is either $\widehat{df}(\lambda)$ or $\widehat{df}(m_k)$, depending on the type of regularization. When σ^2 is unknown, it is usually replaced with an estimate based on the largest model.

Equation (4.52) equivalently derives AIC for the lasso

$$\mathrm{AIC}(\hat{\boldsymbol{\mu}}) = \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2}{n\sigma^2} + \frac{2}{n}\ \widehat{df}(\hat{\boldsymbol{\mu}}). \tag{4.53}$$

Selecting the lasso model by AIC is called *AIC-Lasso shrinkage*. Following the usual definition of BIC, we propose BIC for the lasso as follows

$$\mathrm{BIC}(\hat{\boldsymbol{\mu}}) = \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}\|^2}{n\sigma^2} + \frac{\log(n)}{n}\ \widehat{df}(\hat{\boldsymbol{\mu}}). \tag{4.54}$$

Similarly the lasso model selection by BIC is called *BIC-Lasso shrinkage*.

AIC and BIC possess different asymptotic optimality. It is well known that if the true regression function is not in the candidate models, the model selected by AIC asymptotically achieves the smallest average squared error among the candidates; and the AIC estimator of the regression function converges at the minimax optimal rate whether the true regression function is in the candidate models or not, see Shao (1997), Yang (2005) and references therein. On the other hand, BIC is well known for its consistency in selecting the true model (Shao 1997). If the true model is in the candidate list, the probability of selecting

Table 4.1: *The simulation example: the probability of discovering the exact true model by AIC and BIC Lasso shrinkage. The calculation is based on 2000 replications. Compared with AIC-Lasso shrinkage, BIC-Lasso shrinkage has a much higher probability of identifying the ground truth.*

n	AIC	BIC
100	0.162	0.451
500	0.181	0.623
1000	0.193	0.686
2000	0.184	0.702

Table 4.2: *The simulation example: the median of the number of non-zero variables selected by AIC and BIC Lasso shrinkage based on 2000 replications. One can see that AIC-Lasso shrinkage is conservative in variable selection and BIC-Lasso shrinkage tends to find models with the right size.*

n	AIC	BIC
100	5	4
500	5	3
1000	5	3
2000	5	3

the true model by BIC approaches one as the sample size $n \to \infty$. Considering the case where the true underlying model is sparse, BIC-Lasso shrinkage is adaptive in variable selection. However, AIC-Lasso shrinkage tends to include more non-zero predictors than the truth. The conservative nature of AIC is a familiar result in linear regression. Hence BIC-Lasso shrinkage is more appropriate than AIC-Lasso shrinkage when variable selection is the primary concern in applying the lasso.

Here we show a simulation example to demonstrate the above argument. We simulated response vectors $\mathbf{y}$ from a linear model: $\mathbf{y} = \mathbf{X}\beta + N(0,1)$ where $\beta = (3, 1.5, 0, 0, 2, 0, 0, 0)$. Predictors $\{\mathbf{x}_i\}$ are multivariate normal vectors with pairwise correlation $\text{cor}(i,j) = (0.1)^{|i-j|}$ and the variance of each $\mathbf{x}_i$ is one. For each estimate $\hat{\beta}$, it is said to discover the exact true model if $\{\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_5\}$ are non-zero and the rest coefficients are all zero. Table 1 shows the probability of discovering the exact true model using AIC-Lasso shrinkage and BIC-Lasso shrinkage. In this example both AIC and BIC always select the true predictors $\{1, 2, 5\}$ in all the 2000 replications, but AIC tends to include other variables as real factors as shown in Table 2. In contrast to AIC, BIC has a much lower false positive rate.

One may think of combining the good properties of AIC and BIC into a new criterion. Although this proposal sounds quite reasonable, a surprising result is proved that any model selection criterion cannot be consistent and optimal in average squared error at the same time (Yang 2005). In other words, any model selection criterion must sacrifice either prediction optimality or consistency.

4.4.2 Computation

Using either AIC or BIC to find the optimal lasso model, we are facing an optimization problem

$$\lambda(\text{optimal}) = \arg\min_{\lambda} \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}_\lambda\|^2}{n\sigma^2} + \frac{w_n}{n}\, \widehat{df}(\lambda), \tag{4.55}$$

where $w_n = 2$ for AIC and $w_n = \log(n)$ for BIC. Since the LARS algorithm efficiently solves the lasso solution for all λ, finding $\lambda(\text{optimal})$ is attainable in principle. In fact, we show that $\lambda(\text{optimal})$ is one of the transition points, which further facilitates the searching procedure.

Theorem 4.5. *To find* $\lambda(\text{optimal})$*, we only need to solve*

$$m^* = \arg\min_{m} \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}_{\lambda_m}\|^2}{n\sigma^2} + \frac{w_n}{n}\, \widehat{df}(\lambda_m) \tag{4.56}$$

then $\lambda(\text{optimal}) = \lambda_{m^*}$.

Proof. Let us consider $\lambda \in (\lambda_{m+1}, \lambda_m)$. By (4.21) we have

$$\mathbf{y} - \hat{\boldsymbol{\mu}}_\lambda = (\mathbf{I} - H_{\mathcal{B}_m})\mathbf{y} + \frac{\lambda}{2}\, \mathbf{X}_{\mathcal{B}_m}(\mathbf{X}_{\mathcal{B}_m}^T\mathbf{X}_{\mathcal{B}_m})^{-1}\text{Sgn}_m \tag{4.57}$$

$$\|\mathbf{y} - \hat{\boldsymbol{\mu}}_\lambda\|^2 = \mathbf{y}^T(\mathbf{I} - H_{\mathcal{B}_m})\mathbf{y} + \frac{\lambda^2}{4}\, \text{Sgn}_m^T(\mathbf{X}_{\mathcal{B}_m}^T\mathbf{X}_{\mathcal{B}_m})^{-1}\text{Sgn}_m \tag{4.58}$$

where $H_{\mathcal{B}_m} = \mathbf{X}_{\mathcal{B}_m}(\mathbf{X}_{\mathcal{B}_m}^T\mathbf{X}_{\mathcal{B}_m})^{-1}\mathbf{X}_{\mathcal{B}_m}^T$. Thus we can conclude that $\|\mathbf{y} - \hat{\boldsymbol{\mu}}_\lambda\|^2$ is strictly increasing in the interval $(\lambda_{m+1}, \lambda_m)$. Moreover, the lasso estimates are continuous on λ, hence

$$\|\mathbf{y} - \hat{\boldsymbol{\mu}}_{\lambda_m}\|^2 > \|\mathbf{y} - \hat{\boldsymbol{\mu}}_\lambda\|^2 > \|\mathbf{y} - \hat{\boldsymbol{\mu}}_{\lambda_{m+1}}\|^2. \tag{4.59}$$

On the other hand, note that $\widehat{df}(\lambda) = |\mathcal{B}_m| \; \forall \; \lambda \in (\lambda_{m+1}, \lambda_m)$ and $|\mathcal{B}_m| \geq |\mathcal{B}(\lambda_{m+1})|$. Therefore the optimal choice of λ in $[\lambda_{m+1}, \lambda_m)$ is λ_{m+1}, which means $\lambda(\text{optimal}) \in \{\lambda_m\}, m = 0, 1, 2, \ldots, K$. □

According to Theorem 4.5, the optimal lasso model is immediately selected once the entire lasso solution path is solved by the LARS algorithm, which has the cost of a single least squares fit.

If we consider the best lasso fit in the forward stage-wise modeling picture (like Figure 4.2), inequality (4.59) explains the superiority of the choice of m_k in the conjecture. Let m_k be the last lasso step containing k non-zero predictors. Suppose m'_k is another lasso step containing k non-zero predictors, then $\widehat{df}(\hat{\boldsymbol{\mu}}(m'_k)) = k = \widehat{df}(\hat{\boldsymbol{\mu}}(m_k))$. However, $m'_k < m_k$ gives $\|\mathbf{y} - \hat{\boldsymbol{\mu}}_{m_k}\|^2 < \|\mathbf{y} - \hat{\boldsymbol{\mu}}_{m'_k}\|^2$. Then by the C_p statistic, we see that $\hat{\boldsymbol{\mu}}(m_k)$ is always more accurate than $\hat{\boldsymbol{\mu}}(m'_k)$, while using the same number of non-zero predictors. Using k as the tuning parameter of the lasso, we need to find $k(\text{optimal})$ such that

$$k(\text{optimal}) = \arg\min_k \frac{\|\mathbf{y} - \hat{\boldsymbol{\mu}}_{m_k}\|^2}{n\sigma^2} + \frac{w_n}{n} k. \tag{4.60}$$

Once $\lambda^* = \lambda(\text{optimal})$ and $k^* = k(\text{optimal})$ are found, we fix them as the regularization parameters for fitting the lasso on future data. Using the fixed k^* means the fit on future data is $\hat{\boldsymbol{\mu}}_{m_{k^*}}$, while the fit using the fixed λ^* is $\hat{\boldsymbol{\mu}}_{\lambda^*}$. It is easy to see that the selected models by (4.55) and (4.60) coincide on the training data, i.e., $\hat{\boldsymbol{\mu}}_{\lambda^*} = \hat{\boldsymbol{\mu}}_{m_{k^*}}$.

Figure 4.6 displays the C_p (equivalently AIC) and BIC estimates of risk using the diabetes data. The models selected by C_p are the same as those selected in the LAR paper. With 10 predictors, C_p and BIC select the same model using 7 non-zero covariates. With 64 predictors, C_p selects a model using 15 covariates, while BIC selects a model with 11 covariates.

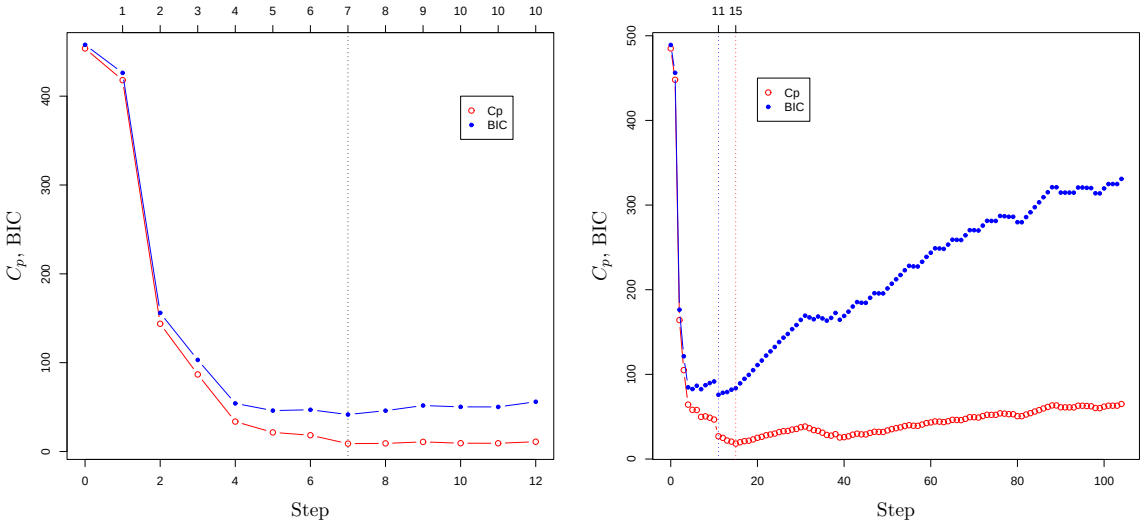

Figure 4.6: *The diabetes data. C_p and BIC estimates of risk with 10 (left) and 64 (right) predictors. In the left panel C_p and BIC select the same model with 7 non-zero coefficients. In the right panel, C_p selects a model with 15 non-zero coefficients and BIC selects a model with 11 non-zero coefficients.*

4.5 Discussion

4.5.1 Smoothed *df* estimate

It is interesting to note that the true degrees of freedom is a strictly decreasing function of λ, as shown in Figure 4.7. However, the unbiased estimate $\widehat{df}(\lambda)$ is not necessarily monotone, although its global trend is monotonically decreasing. The same phenomenon is also shown in the right panel of Figure 4.1. The non-monotonicity of $\widehat{df}(\lambda)$ is due to the fact that some variables can be dropped during the LARS/lasso process.

An interesting question is that whether there is a smoothed estimate $\widehat{df}^*(\lambda)$ such that $\widehat{df}^*(\lambda)$ is a smooth decreasing function and keeps the unbiased property, i.e.,

$$df(\lambda) = E[\widehat{df}^*(\lambda)] \tag{4.61}$$

holds for all λ. This is a future research topic.

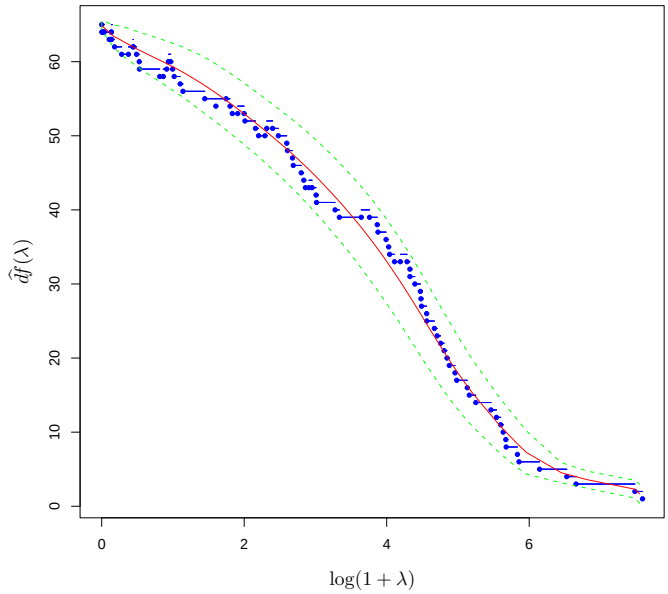

Figure 4.7: *The dotted line is the curve of estimated degrees of freedom* ($\widehat{df}(\lambda)$ *vs.* $\log(1+\lambda)$)*, using a typical realization* $\mathbf{y}^*$ *generated by the synthetic model (4.16). The smooth curve shows the true degrees of freedom* $df(\lambda)$ *obtained by averaging* 20000 *estimated curves. One can see that the estimated df curve is piece-wise constant and non-monotone, while the true df curve is smooth and monotone. The two thin broken lines correspond to* $df(\lambda) \pm 2\sqrt{Var(\widehat{df}(\lambda))}$*, where* $Var(\widehat{df}(\lambda))$ *is calculated from the* $B = 20000$ *replications.*

4.5.2 *df* of the elastic net

The elastic net is a generalized version of the lasso and offers many additional advantages. Our analysis in this Chapter can be used to derive the degrees of freedom of the elastic net, since the elastic net is equivalent to a lasso-type problem on an "augmented data set" (Lemma 2.1). The (naive) elastic net estimates are given by

$$\hat{\beta} = \arg\min_{\beta} \|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j\|^2 + \lambda_2 \|\boldsymbol{\beta}\|^2 + \lambda_1 \|\boldsymbol{\beta}\|_1, \tag{4.62}$$

where λ_1 and λ_2 are non-negative regularization parameters.

We consider the degrees of freedom ($df(\lambda_1, \lambda_2)$) of the elastic net with a fixed (λ_1, λ_2) pair. It is straightforward to check that by similar arguments in Section 4.32, we have an unbiased estimate for ($df(\lambda_1, \lambda_2)$) as follows

$$\widehat{df} = \mathrm{Tr}\left(\mathbf{H}_{\lambda_2}(\mathcal{A})\right) \tag{4.63}$$

where $\mathcal{A}$ is *the active set* and

$$\mathbf{H}_{\lambda_2}(\mathcal{A}) = \mathbf{X}_{\mathcal{A}} \left(\mathbf{X}_{\mathcal{A}}^T \mathbf{X}_{\mathcal{A}} + \lambda_2 \mathbf{I}\right)^{-1} \mathbf{X}_{\mathcal{A}}^T. \tag{4.64}$$

One can see equations (4.63) and (4.64) by combining the results in Section 4.32 and Lemma 2.1 in Chapter 2.

4.6 Proofs of Lemmas 4.2-4.8

Proof. Lemma 4.2

Let

$$\ell(\beta, \mathbf{y}) = \|\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j\|^2 + \lambda \sum_{j=1}^{p} |\beta_j| \,. \tag{4.65}$$

Given $\mathbf{y}$, $\hat{\beta}(\lambda)$ is the minimizer of $\ell(\beta, \mathbf{y})$. For those $j \in \mathcal{B}_m$ we must have $\frac{\partial \ell(\beta, \mathbf{y})}{\partial \beta_j} = 0$, i.e.,

$$-2\mathbf{x}_j^T \left(\mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \hat{\beta}(\lambda)_j \right) + \lambda \ \mathrm{Sgn}(\hat{\beta}(\lambda)_j) = 0, \quad \text{for } j \in \mathcal{B}_m. \tag{4.66}$$

Since $\hat{\beta}(\lambda)_i = 0$ for all $i \notin \mathcal{B}_m$, then $\sum_{j=1}^{p} \mathbf{x}_j \hat{\beta}(\lambda)_j = \sum_{j \in \mathcal{B}_\lambda} \mathbf{x}_j \hat{\beta}(\lambda)_j$. Thus equations in (4.66) become

$$-2\mathbf{X}_{\mathcal{B}_m}^T \left(\mathbf{y} - \mathbf{X}_{\mathcal{B}_m} \hat{\beta}(\lambda)_{\mathcal{B}_m} \right) + \lambda \ \mathrm{Sgn}_m = 0 \tag{4.67}$$

which gives (4.21). □

Proof. Lemma 4.3

We adopt the matrix notation used in S : $\mathbf{M}[i,]$ means the i-th row of $\mathbf{M}$. i_{add} joins $\mathcal{B}_m$ at λ_m, then

$$\hat{\beta}(\lambda_m)_{i_{add}} = 0. \tag{4.68}$$

Consider $\hat{\beta}(\lambda)$ for $\lambda \in (\lambda_{m+1}, \lambda_m)$. Lemma 4.2 gives

$$\hat{\beta}(\lambda)_{\mathcal{B}_m} = \left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m} \right)^{-1} \left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{y} - \frac{\lambda}{2} \ \mathrm{Sgn}_m \right). \tag{4.69}$$

By the continuity of $\hat{\beta}(\lambda)_{i_{add}}$, taking the limit of the i^*-th element of (4.69) as $\lambda \to \lambda_m - 0$, we have

$$2 \left\{ \left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m} \right)^{-1} [i^*,] \mathbf{X}_{\mathcal{B}_m}^T \right\} \mathbf{y} = \lambda_m \left\{ \left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m} \right)^{-1} [i^*,] \mathrm{Sgn}_m \right\}. \tag{4.70}$$

The second $\{\cdot\}$ is a non-zero scalar, otherwise $\hat{\beta}(\lambda)_{i_{add}} = 0$ for all $\lambda \in (\lambda_{m+1}, \lambda_m)$, which contradicts the assumption that i_{add} becomes a member of the active set $\mathcal{B}_m$. Thus we have

$$\lambda_m = \left\{ 2 \frac{\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m} \right)^{-1} [i^*,]}{\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m} \right)^{-1} [i^*,] \mathrm{Sgn}_m} \right\} \mathbf{X}_{\mathcal{B}_m}^T \mathbf{y} =: v(\mathcal{B}_m, i^*) \mathbf{X}_{\mathcal{B}_m}^T \mathbf{y}, \tag{4.71}$$

where $v(\mathcal{B}_m, i^*) = \left\{ 2 \frac{\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m} \right)^{-1} [i^*,]}{\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m} \right)^{-1} [i^*,] \mathrm{Sgn}_m} \right\}$. Rearranging (4.71), we get (4.22).

Similarly, if j_{drop} is a dropped index at λ_{m+1}, we take the limit of the j^*-th element of

(4.69) as $\lambda \to \lambda_{m+1} + 0$ to conclude that

$$\lambda_{m+1} = \left\{ 2\frac{\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m}\right)^{-1}[j^*,]}{\left(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m}\right)^{-1}[j^*,]\mathrm{Sgn}_m} \right\} \mathbf{X}_{\mathcal{B}_m}^T \mathbf{y} =: v(\mathcal{B}_m, j^*)\mathbf{X}_{\mathcal{B}_m}^T \mathbf{y}, \tag{4.72}$$

where $v(\mathcal{B}_m, j^*) = \left\{ 2\frac{(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m})^{-1}[j^*,]}{(\mathbf{X}_{\mathcal{B}_m}^T \mathbf{X}_{\mathcal{B}_m})^{-1}[j^*,]\mathrm{Sgn}_m} \right\}$. Rearranging (4.72), we get (4.23). □

Proof. Lemma 4.4

Suppose for some $\mathbf{y}$ and m, $\lambda = \lambda(\mathbf{y})_m$. $\lambda > 0$ means m is not the last lasso step. By Lemma 4.3 we have

$$\lambda = \lambda_m = \{v(\mathcal{B}_m, i^*)\mathbf{X}_{\mathcal{B}_m}^T\}\mathbf{y} =: \alpha(\mathcal{B}_m, i^*)\mathbf{y}. \tag{4.73}$$

Obviously $\alpha(\mathcal{B}_m, i^*) = v(\mathcal{B}_m, i^*)\mathbf{X}_{\mathcal{B}_m}^T$ is a non-zero vector. Now let α_λ be the totality of $\alpha(\mathcal{B}_m, i^*)$ by considering all the possible combinations of $\mathcal{B}_m$, i^* and the sign vector Sgn_m. α_λ only depends on $\mathbf{X}$ and is a finite set, since at most p predictors are available. Thus $\forall \alpha \in \alpha_\lambda$, $\alpha\mathbf{y} = \lambda$ defines a hyperplane in $\mathbb{R}^n$. We define

$$\mathcal{N}_\lambda = \{\mathbf{y} : \alpha\mathbf{y} = \lambda \text{ for some } \alpha \in \alpha_\lambda\} \quad \text{and} \quad \mathcal{G}_\lambda = \mathbb{R}^n \setminus \mathcal{N}_\lambda.$$

Then on $\mathcal{G}_\lambda$ (4.73) is impossible.

□

Proof. Lemma 4.5

For writing convenience we omit the subscript λ. Let

$$\hat{\beta}(\mathbf{y})_{ols} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{4.74}$$

be the OLS estimates. Note that we always have the inequality

$$|\hat{\beta}(\mathbf{y})|_1 \le |\hat{\beta}(\mathbf{y})_{ols}|_1 \ . \tag{4.75}$$

Fix an arbitrary $\mathbf{y}_0$ and consider a sequence of $\{\mathbf{y}_n\}$ $(n = 1, 2, \ldots)$ such that $\mathbf{y}_n \to \mathbf{y}_0$. Since $\mathbf{y}_n \to \mathbf{y}_0$, we can find a Y such that $\|\mathbf{y}_n\| \leq Y$ for all $n = 0, 1, 2, \ldots$. Consequently $\|\hat{\beta}(\mathbf{y}_n)_{ols}\| \leq B$ for some upper bound B (B is determined by $\mathbf{X}$ and Y). By Cauchy's inequality and (4.75), we have

$$|\hat{\beta}(\mathbf{y}_n)|_1 \leq \sqrt{pB} \quad \text{for all } n = 0, 1, 2, \ldots \tag{4.76}$$

(4.76) implies that to show $\hat{\beta}(\mathbf{y}_n) \to \hat{\beta}(\mathbf{y}_0)$, it is equivalent to show for every converging subsequence of $\{\hat{\beta}(\mathbf{y}_n)\}$, say $\{\hat{\beta}(\mathbf{y}_{n_k})\}$, the subsequence converge to $\hat{\beta}(\mathbf{y})$.

Now assume $\hat{\beta}(\mathbf{y}_{n_k})$ converges to $\hat{\beta}_\infty$ as $n_k \to \infty$. We show $\hat{\beta}_\infty = \hat{\beta}(\mathbf{y}_0)$. The lasso criterion $\ell(\beta, \mathbf{y})$ is written in (4.65). Let

$$\Delta\ell(\beta, \mathbf{y}, \mathbf{y}') = \ell(\beta, \mathbf{y}) - \ell(\beta, \mathbf{y}'). \tag{4.77}$$

By the definition of $\hat{\beta}_{n_k}$, we must have

$$\ell(\hat{\beta}(\mathbf{y}_0), \mathbf{y}_{n_k}) \geq \ell(\hat{\beta}(\mathbf{y}_{n_k}), \mathbf{y}_{n_k}). \tag{4.78}$$

Then (4.78) gives

$$\begin{aligned} \ell(\hat{\beta}(\mathbf{y}_0), \mathbf{y}_0) &= \ell(\hat{\beta}(\mathbf{y}_0), \mathbf{y}_{n_k}) + \Delta\ell(\hat{\beta}(\mathbf{y}_0), \mathbf{y}_0, \mathbf{y}_{n_k}) \\ &\geq \ell(\hat{\beta}(\mathbf{y}_{n_k}), \mathbf{y}_{n_k}) + \Delta\ell(\hat{\beta}(\mathbf{y}_0), \mathbf{y}_0, \mathbf{y}_{n_k}) \\ &= \ell(\hat{\beta}(\mathbf{y}_{n_k}), \mathbf{y}_0) + \Delta\ell(\hat{\beta}(\mathbf{y}_{n_k}), \mathbf{y}_{n_k}, \mathbf{y}_0) + \Delta\ell(\hat{\beta}(\mathbf{y}_0), \mathbf{y}_0, \mathbf{y}_{n_k}). \end{aligned} \tag{4.79}$$

We observe

$$\Delta\ell(\hat{\beta}(\mathbf{y}_{n_k}), \mathbf{y}_{n_k}, \mathbf{y}_0) + \Delta\ell(\hat{\beta}(\mathbf{y}_0), \mathbf{y}_0, \mathbf{y}_{n_k}) = 2(\mathbf{y}_0 - \mathbf{y}_{n_k})\mathbf{X}^T(\hat{\beta}(\mathbf{y}_{n_k}) - \hat{\beta}(\mathbf{y}_0)). \tag{4.80}$$

Let $n_k \to \infty$, the right hand side of (4.80) goes to zero. Moreover, $\ell(\hat{\beta}(\mathbf{y}_{n_k}), \mathbf{y}_0) \to$

$\ell(\hat{\beta}_\infty, \mathbf{y}_0)$. Therefore (4.79) reduces to

$$\ell(\hat{\beta}(\mathbf{y}_0), \mathbf{y}_0) \geq \ell(\hat{\beta}_\infty, \mathbf{y}_0).$$

However, $\hat{\beta}(\mathbf{y}_0)$ is the unique minimizer of $\ell(\beta, \mathbf{y}_0)$, thus $\hat{\beta}_\infty = \hat{\beta}(\mathbf{y}_0)$.

□

Proof. Lemma 4.6

Fix an arbitrary $\mathbf{y}_0 \in \mathcal{G}_\lambda$. Denote Ball$(\mathbf{y}, r)$ the n-dimensional ball with center $\mathbf{y}$ and radius r. Note that $\mathcal{G}_\lambda$ is an open set, so we can choose a small enough ϵ such that Ball$(\mathbf{y}_0, \epsilon) \subset \mathcal{G}_\lambda$. Fix ϵ. Suppose $\mathbf{y}_n \to \mathbf{y}$ as $n \to \infty$, then without loss of generality we can assume $\mathbf{y}_n \in$ Ball$(\mathbf{y}_0, \epsilon)$ for all n. So λ is not a transition point for any $\mathbf{y}_n$.

By definition $\hat{\beta}(\mathbf{y}_0)_j \neq 0$ for all $j \in \mathcal{B}(\mathbf{y}_0)$. Then Lemma 4.5 says that $\exists$ a N, as long as $n > N_1$, we have $\hat{\beta}(\mathbf{y}_n)_j \neq 0$ and $\text{Sgn}(\hat{\beta}(\mathbf{y}_n)) = \text{Sgn}(\hat{\beta}(\mathbf{y}_n))$, for all $j \in \mathcal{B}(\mathbf{y}_0)$. Thus $\mathcal{B}(\mathbf{y}_0) \subseteq \mathcal{B}(\mathbf{y}_n)$ $\forall n > N_1$.

On the other hand, we have the following *equiangular* conditions (Efron, Hastie, Johnstone & Tibshirani 2004)

$$\lambda = 2|\mathbf{x}_j^T(\mathbf{y}_0 - \mathbf{X}\hat{\beta}(\mathbf{y}_0))| \quad \forall\ j \in \mathcal{B}(\mathbf{y}_0), \tag{4.81}$$

$$\lambda > 2|\mathbf{x}_j^T(\mathbf{y}_0 - \mathbf{X}\hat{\beta}(\mathbf{y}_0))| \quad \forall\ j \notin \mathcal{B}(\mathbf{y}_0). \tag{4.82}$$

Using Lemma 4.5 again, we conclude that $\exists$ a $N > N_1$ such that $\forall\ j \notin \mathcal{B}(\mathbf{y}_0)$ the strict inequalities (4.82) hold for $\mathbf{y}_n$ provided $n > N$. Thus $\mathcal{B}^c(\mathbf{y}_0) \subseteq \mathcal{B}^c(\mathbf{y}_n)$ $\forall n > N$. Therefore we have $\mathcal{B}(\mathbf{y}_n) = \mathcal{B}(\mathbf{y}_0)$ $\forall n > N$. Then the local constancy of the sign vector follows the continuity of $\hat{\beta}(\mathbf{y})$. □

Proof. Lemma 4.7

Suppose at step m, $|\mathcal{W}_m(\mathbf{y})| \geq 2$. Let i_{add} and j_{add} be two of the predictors in $\mathcal{W}_m(\mathbf{y})$, and let i^*_{add} and j^*_{add} be their indices in the current active set $\mathcal{A}$. Note the current active

set $\mathcal{A}$ is $\mathcal{B}_m$ in Lemma 4.3. Hence we have

$$\lambda_m = v[\mathcal{A}, i^*]\mathbf{X}_{\mathcal{A}}^T\mathbf{y}, \tag{4.83}$$

$$\lambda_m = v[\mathcal{A}, j^*]\mathbf{X}_{\mathcal{A}}^T\mathbf{y}. \tag{4.84}$$

Therefore

$$0 = \left\{[v(\mathcal{A}, i^*_{add}) - v(\mathcal{A}, j^*_{add})]\mathbf{X}_{\mathcal{A}}^T\right\}\mathbf{y} =: \alpha_{add}\mathbf{y}. \tag{4.85}$$

We claim $\alpha_{add} = [v(\mathcal{A}, i^*_{add}) - v(\mathcal{A}, j^*_{add})]\mathbf{X}_{\mathcal{A}}^T$ is not a zero vector. Otherwise, since $\{\mathbf{X}_j\}$ are linearly independent, $\alpha_{add} = 0$ forces $v(\mathcal{A}, i^*_{add}) - v(\mathcal{A}, j^*_{add}) = 0$. Then we have

$$\frac{\left(\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}\right)^{-1}[i^*,]}{\left(\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}\right)^{-1}[i^*,]\mathrm{Sgn}_{\mathcal{A}}} = \frac{\left(\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}\right)^{-1}[j^*,]}{\left(\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}}\right)^{-1}[i^*,]\mathrm{Sgn}_{\mathcal{A}}}, \tag{4.86}$$

which contradicts the fact $(\mathbf{X}_{\mathcal{A}}^T\mathbf{X}_{\mathcal{A}})^{-1}$ is a full rank matrix.

Similarly, if i_{drop} and j_{drop} are dropped predictors, then

$$0 = \left\{[v(\mathcal{A}, i^*_{drop}) - v(\mathcal{A}, j^*_{drop})]\mathbf{X}_{\mathcal{A}}^T\right\}\mathbf{y} =: \alpha_{drop}\mathbf{y}, \tag{4.87}$$

and $\alpha_{drop} = [v(\mathcal{A}, i^*_{drop}) - v(\mathcal{A}, j^*_{drop})]\mathbf{X}_{\mathcal{A}}^T$ is a non-zero vector.

Let M_0 be the totality of α_{add} and α_{drop} by considering all the possible combinations of $\mathcal{A}$, (i_{add}, j_{add}), (i_{drop}, j_{drop}) and $\mathrm{Sgn}_{\mathcal{A}}$. Clearly M_0 is a finite set and only depends on $\mathbf{X}$. Let

$$\widetilde{\mathcal{N}}_0 = \Big\{y : \alpha y = 0 \ \text{ for some } \ \alpha \in M_0\Big\}. \tag{4.88}$$

Then on $\mathbb{R}^n \setminus \widetilde{\mathcal{N}}_0$, the conclusion holds. □

Proof. Lemma 4.8

First we can choose a sufficiently small ϵ^* such that $\forall\ \mathbf{y}' : \|\mathbf{y}' - \mathbf{y}\| < \epsilon^*$, $\mathbf{y}'$ is an interior point of $\mathbb{R}^n \setminus (\widetilde{\mathcal{N}}_0 \cup \widetilde{\mathcal{N}}_1)$. Suppose K is the last step of the lasso solution given $\mathbf{y}$. We show that for each $m \le K$, there is a $\epsilon_m < \epsilon^*$ such that $q_{m-1}\mathcal{V}_{m-1}$ and $\mathcal{W}_m$ are locally fixed in

the Ball$(\mathbf{y}, \epsilon_m)$; also λ_m and $\hat{\beta}_m$ are locally continuous in the Ball$(\mathbf{y}, \epsilon_m)$.

We proceed by induction. For $m = 0$ we only need to verify the local constancy of $\mathcal{W}_0$. Lemma 4.7 says $\mathcal{W}_0(\mathbf{y}) = \{j\}$. By the definition of $\mathcal{W}$, we have $|\mathbf{x}_j^T\mathbf{y}| > |\mathbf{x}_i^T\mathbf{y}|$ for all $i \neq j$. Thus the strict inequality holds if $\mathbf{y}'$ is sufficiently close to $\mathbf{y}$, which implies $\mathcal{W}_0(\mathbf{y}') = \{j\} = \mathcal{W}_0(\mathbf{y})$.

Assuming the conclusion holds for m, we consider points in the Ball$(\mathbf{y}, \epsilon_{m+1})$ with $\epsilon_{m+1} < \min_{\ell \leq m}\{\epsilon_\ell\}$. By the induction assumption, $A_m(\mathbf{y})$ is locally fixed since it only depends on $\{(q_\ell\mathcal{V}_\ell, \mathcal{W}_\ell), \ell \leq (m-1)\}$. $q_m\mathcal{V}_m = \varnothing$ is equivalent to $\widehat{\gamma}(\mathbf{y}) < \widetilde{\gamma}(\mathbf{y})$. Once A_m and $\mathcal{W}_m$ are fixed, both $\widehat{\gamma}(\mathbf{y})$ and $\widetilde{\gamma}(\mathbf{y})$ are continuous on $\mathbf{y}$. Thus if $\mathbf{y}'$ is sufficiently close to $\mathbf{y}$, the strict inequality still holds, which means $q_m(\mathbf{y}')\mathcal{V}_m(\mathbf{y}') = \varnothing$. If $q_m\mathcal{V}_m = \mathcal{V}_m$, then $\widehat{\gamma}(\mathbf{y}) > \widetilde{\gamma}(\mathbf{y})$ since the possibility of $\widehat{\gamma}(\mathbf{y}) = \widetilde{\gamma}(\mathbf{y})$ is ruled out. By Lemma 4.7, we let $\mathcal{V}_m(\mathbf{y}) = \{j\}$. By the definition of $\widetilde{\gamma}(\mathbf{y})$, we can see that if $\mathbf{y}'$ is sufficiently close to $\mathbf{y}$, $\mathcal{V}_m(\mathbf{y}') = \{j\}$, and $\widehat{\gamma}(\mathbf{y}') > \widetilde{\gamma}(\mathbf{y}')$ by continuity. So $q_m(\mathbf{y}')\mathcal{V}_m(\mathbf{y}') = \mathcal{V}_m(\mathbf{y}') = \mathcal{V}_m(\mathbf{y})$.

Then $\hat{\beta}_{m+1}$ and λ_{m+1} are locally continuous, because their updates are continuous on $\mathbf{y}$ once A_m,$\mathcal{W}_m$ and $q_m\mathcal{V}_m$ are fixed. Moreover, since $q_m\mathcal{V}_m$ is fixed, A_{m+1} is also locally fixed. Let $\mathcal{W}_{m+1}(\mathbf{y}) = \{j\}$ for some $j \in A_{m+1}^c$. Then we have

$$|\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}^T\hat{\beta}_{m+1}(\mathbf{y}))| > |\mathbf{x}_i^T(\mathbf{y} - \mathbf{X}^T\hat{\beta}_{m+1}(\mathbf{y})| \quad \forall\ i \neq j,\ i \in A_{m+1}^c$$

By the continuity argument, the above strict inequality holds for all $\mathbf{y}'$ provided $\|\mathbf{y}' - \mathbf{y}\| \leq \epsilon_{m+1}$ for a sufficiently small ϵ_{m+1}. So $\mathcal{W}_{m+1}(\mathbf{y}') = \{j\} = \mathcal{W}_{m+1}(\mathbf{y})$. In conclusion, we can choose a small enough ϵ_{m+1} to make sure that $q_m\mathcal{V}_m$ and $\mathcal{W}_{m+1}$ are locally fixed, and $\hat{\beta}_{m+1}$ and λ_{m+1} are locally continuous. □

Chapter 5

Summary of Thesis

In Chapter 2, we have proposed 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). We have considered an algorithm called LARS-EN for efficiently computing the entire elastic net regularization path. In the $p \gg n$ problem, it is not necessary to run the LARS-EN algorithm to the end, thus we suggest an early stopping strategy to save computations.

In Chapter 3 we have derived a sparse principal component algorithm using the elastic net, which is a principled approach to modify PCA based on a new sparse PCA criterion. We show that PCA can be formulated as a regression-type optimization problem, then sparse loadings are obtained by imposing an elastic net constraint on the regression coefficients. To minimize our SPCA criterion, we have proposed an efficient algorithm which takes advantage of SVD and the LARS-EN algorithm for solving the elastic net. SPCA allows flexible control of the sparse structure of the resulting loadings. Compared to other methods for deriving sparse PCs, SPCA appears to have advantages in computational efficiency, high explained variance, and ability of identifying important variables.

In Chapter 4, we have studied the degrees of freedom of the lasso. We prove that the number of non-zero coefficients is an unbiased estimate for the degrees of freedom of the lasso—a conclusion requires no special assumption on the predictors. Our analysis also indicates that k is a good estimate of $df(m_k)$, where m_k can be any LARS-lasso step containing exact k non-zero predictors. The estimates of degrees of freedom are used to construct various model selection criteria such as C_p, AIC and BIC, which provide a principled and efficient approach to obtain the optimal lasso fit with the computational effort of a single ordinary least-squares fit.

Bibliography

Akaike, H. (1973), 'Information theory and an extension of the maximum likelihood principle', *Second International Symposium on Information Theory* pp. 267–281.

Alter, O., Brown, P. & Botstein, D. (2000), 'Singular value decomposition for genome-wide expression data processing and modeling', *Proceedings of the National Academy of Sciences* **97**, 10101–10106.

Breiman, L. (1996), 'Heuristics of instability and stabilization in model selection', *The Annals of Statistics* **24**, 2350–2383.

Cadima, J. & Jolliffe, I. (1995), 'Loadings and correlations in the interpretation of principal components', *Journal of Applied Statistics* **22**, 203–214.

Chen, S., Donoho, D. & Saunders, M. (2001), 'Atomic decomposition by basis pursuit', *SIAM Review* **43(1)**, 129C59.

Dettling, M. & Bühlmann, P. (2004), 'Finding predictive gene groups from microarray data', *Journal of Multivariate Analysis* **90**, 106–131.

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

Donoho, D. (2004), For most large underdetermined systems of equations, the minimal ℓ^1-norm near-solution approximates the sparsest near-solution, Technical report, Department of Statistics, Stanford University.

Donoho, D. & Elad, M. (2002), 'Optimally sparse representation from overcomplete dictionaries via ℓ^1-norm minimization', *Proceedings of the National Academy of Sciences* **1005**, 2197–2002.

Donoho, D. & Huo, X. (2001), 'Uncertainty principles and ideal atomic decomposition', *IEEE Transactions on Information Theory* **47(7)**, 2845C62.

Donoho, D., Johnstone, I., Kerkyacharian, G. & Picard, D. (1995), 'Wavelet shrinkage: asymptopia? (with discussion)', *Journal of the Royal Statistical Society, B* **57**, 301–337.

Efron, B. (1986), 'How biased is the apparent error rate of a prediction rule?', *Journal of the American Statistical Association* **81**, 461–470.

Efron, B. (2004), 'The estimation of prediction error: covariance penalties and cross-validation', *Journal of the American Statistical Association* **99**, 619–632.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), 'Least angle regression', *The Annals of Statistics* **32**, 407–499.

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

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

Friedman, J. (1989), 'Regularized discriminant analysis', *Journal of the American Statistical Association* **84**, 249–266.

Friedman, J., Hastie, T., Rosset, S., Tibshirani, R. & Zhu, J. (2004), 'Discussion of boosting papers', *The Annals of Statistics* **32**, 102–107.

Fu, W. (1998), 'Penalized regression: The bridge versus the lasso', *Journal of Computational and Graphical Statistics* **7**, 397–416.

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.

Hancock, P., Burton, A. & Bruce, V. (1996), 'Face processing: human perception and principal components analysis', *Memory and Cognition* **24**, 26–40.

Hastie, T. & Tibshirani, R. (1990), *Generalized Additive Models*, Chapman and Hall, London.

Hastie, T., Tibshirani, R., Botstein, D. & Brown, P. (2003), 'Supervised harvesting of expression trees', *Genome Biology* **2**, 0003.1–0003.12.

Hastie, T., Tibshirani, R., Eisen, M., Brown, P., Ross, D., Scherf, U., Weinstein, J., Alizadeh, A., Staudt, L. & Botstein, D. (2000), ''gene shaving' as a method for identifying distinct sets of genes with similar expression patterns', *Genome Biology* **1**, 1–21.

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

Hoerl, A. & Kennard, R. (1988), Ridge regression, *in* 'Encyclopedia of Statistical Sciences', Vol. 8, Wiley, New York, pp. 129–136.

Jeffers, J. (1967), 'Two case studies in the application of principal component', *Applied Statistics* **16**, 225–236.

Jolliffe, I. (1986), *Principal component analysis*, Springer Verlag, New York.

Jolliffe, I. (1995), 'Rotation of principal components: choice of normalization constraints', *Journal of Applied Statistics* **22**, 29–35.

Jolliffe, I. T., Trendafilov, N. T. & Uddin, M. (2003), 'A modified principal component technique based on the lasso', *Journal of Computational and Graphical Statistics* **12**, 531–547.

Mallows, C. (1973), 'Some comments on c_p', *Technometrics* **15**, 661–675.

Mardia, K., Kent, J. & Bibby, J. (1979), *Multivariate Analysis*, Academic Press.

McCabe, G. (1984), 'Principal variables', *Technometrics* **26**, 137–144.

Meyer, M. & Woodroofe, M. (2000), 'On the degrees of freedom in shape-restricted regression', *Annals of Statistcs* **28**, 1083–1104.

Osborne, M., Presnell, B. & Turlach, B. (2000), 'A new approach to variable selection in least squares problems', *IMA Journal of Numerical Analysis* **20(3)**, 389–403.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukheriee, S., Yeang, C., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J., Poggio, T., Gerald, W., Loda, M., Lander, E. & Golub, T. (2001), 'Multiclass cancer diagnosis using tumor gene expression signature', *Proceedings of the National Academy of Sciences* **98**, 15149–15154.

Schwartz, G. (1978), 'Estimating the dimension of a model', *Annals of Statistics* **6**, 461–464.

Segal, M., Dahlquist, K. & Conklin, B. (2003), 'Regression approach for microarray data analysis', *Journal of Computational Biology* **10**, 961–980.

Shao, J. (1997), 'An asymptotic theory for linear model selection (with discussion)', *Statistica Sinica* **7**, 221–242.

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', *Journal of Urology.* **16**, 1076–1083.

Stein, C. (1981), 'Estimation of the mean of a multivariate normal distribution', *Annals of Statistics* **9**, 1135–1151.

Tibshirani, R. (1996), 'Regression shrinkage and selection via the lasso', *Journal of the Royal Statistical Society, 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.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

Vines, S. (2000), 'Simple principal components', *Applied Statistics* **49**, 441–451.

Wahba, G. (1990), *Spline Models for Observational Data*, Vol. 59 of *Series in Applied Mathematics*, SIAM, Philadelphia.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Marks, J. & Nevins, J. (2001), 'Predicting the clinical status of human breast cencer using gene expression profiles', *PNAS* **98**, 11462–11467.

Yang, Y. (2005), 'Can the strengths of AIC and BIC be shared?-a conflict between model identification and regression estimation', *Biometrika* **In press**.

Zhu, J. & Hastie, T. (2004), 'Classification of gene microarrys by penalized logistic regression', *Biostatistics* **5(3)**, 427–444.