Patrick Breheny and Jian Huang

High-Dimensional Regression Modeling

Dedication goes
here.

Contents

Preface

The subject of the book is penalized regression modeling for high-dimensional data. Increasingly, the data collected in many fields is high-dimensional, in the sense that many characteristics, or features, are recorded for each observation. The collection of this kind of data is a relatively recent phenomenon, and it poses many challenges that traditional statistical methods have proven incapable of addressing. During the past two decades, penalized regression models have become a widespread and important tool for analyzing these kinds of data sets. Although there is a large literature on this topic, the field is still relatively new and there are few books available to individuals looking for an organized overview of this area.

Furthermore, there has been a fair amount of recent research developing inferential methods for high-dimensional data, including the construction of confidence intervals for penalized regression parameters and the estimation of false discovery rates and prediction error. As of this writing, few books are available summarizing these various approaches and illustrating their use in applied settings.

Our aim is to cover the concepts behind penalized regression, survey the variety of specific methods and models that have been proposed, present the relevant theoretical properties of penalized regression methods, provide an understanding of the algorithms used to fit these models, and discuss the practical aspects of using these methods to analyze real data. In particular, we have included numerous case studies of real data with reproducible `R` code to illustrate the use of various penalized regression packages.

Intended audience

We expect the book to be of interest to practicing statisticians and researchers who work with high-dimensional data or are interested in getting into the field, to students interested in high-dimensional modeling as a research area, and to instructors looking to develop a course on this topic.

This book is intended to be accessible to individuals who have completed at least one year of study in a statistics or biostatistics program.

Specifically, we assume knowledge of basic mathematical statistics at the level of *Statistical Inference* by Casella and Berger or higher, and a basic knowledge of linear models and matrix algebra. In addition, we will provide examples of data analysis using `R`, so some basic knowledge of how to use `R` will be important in order to follow along in those sections.

This book can serve as the primary textbook for a graduate-level elective on high-dimensional modeling in either a statistics or biostatistics department. Indeed, I have used this book as the text for the course "High-Dimensional Data Analysis" at the University of Iowa, which I have taught several times. For a one-semester course, I cover Parts I and II more or less in their entirety, and pick a handful of topics from Part IV, any of which can be taught after seeing Parts I and II. Part III, on other likelihood functions, is valuable material but in my experience difficult to cover in the interest of time and sometimes inaccessible (e.g., regularized Cox regression if some students have not taken a course in survival analysis).

Starred sections

Starred sections: these sections are more technical than the rest of the book. They include important proofs and mathematical results, but can be skipped if the reader is not interested.

`R` package

We include numerous sections of code in the book to demonstrate how to use available software for penalized regression modeling, including important options to be aware of, what they do, and when and why you might choose to modify them.

Furthermore, this book is intended to offer a "hands-on" experience, so that figures, simulations, analyses, etc. can be reproduced by the reader. To accomplish this, we provide an `R` package called `hdrm`, available at `https://github.com/pbreheny/hdrm`. To install the package, either download the latest release at the URL provided or open `R` and install `hdrm` using the `remotes` package:

```
remotes::install_github("pbreheny/hdrm")
```

Once installed, load the package with `library(hdrm)`. You can then reproduce, say, Figure 9.3, with

```
Fig9.3()
```

Likewise, you can reproduce Example 9.1 with `Ex9.1()` and Table 9.4 with `Tab9.4()`.

Running `Fig9.3()` will reproduce Figure 9.3 exactly as it appears in the book. However, options for changing various parameters are usually available. For example, to run a simulation with a different seed than the one used in the book,

```
Fig9.3(seed=12345)
```

Likewise, some of the simulations are quite time-consuming to run. To run them with a smaller number of replications (say, if the default was `N=1000`,

```
Fig9.3(N=100)
```

Finally, one might be interested in changing the parameters of a simulation. For example, perhaps the book illustrates the case where the correlation is 0.5; you might be interested in what the figure looks like with either greater or weaker correlation:

```
Fig9.3(rho=0.8)
Fig9.3(rho=0.2)
```

Check the documentation (`?Fig9.3`) for a list of available options.

Finally, note that some of the code to construct figures and tables is referred to in the text. For example, the text might say that "Figure 2.12 was produced with"

```
plot(fit)
```

which is mostly true in the sense that if you run `plot(fit)` you will obtain mostly the same plot. However, many of the figures in the book were modified for aesthetic purposes. So the actual code used to construct Figure 2.12 may have been

```
plot(fit, xvar="lambda", las=1, xlab=expression(lambda), xaxt="n",
     bty="n", xlim=xlim, col=pal(nv), lwd=2)
at <- seq(xlim[1], xlim[2], length=5)
axis(1, at=at, labels=round(exp(at), 2))
abline(v=log(cvfit$lambda.min), col="gray", lty=2, lwd=2)
abline(v=log(cvfit$lambda.1se), col="gray", lty=2, lwd=2)
```

This was done to avoid cluttering the text and the main idea of the code with a bunch of graphical options. If you want to reproduce the figure exactly, that is what `Fig9.3()` is for. You can also run `Fig9.3` without the parentheses to print the code and see exactly which options were specified.

Symbols

THIS IS JUST A DUMMY SYMBOL LIST FROM THE PUBLISHER. WE MAY OR MAY NOT WANT TO ACTUALLY INCLUDE A LIST LIKE THIS.

Symbol Description

α	To solve the generator maintenance scheduling, in the past, several mathematical techniques have been applied.
σ^2	These include integer programming, integer linear programming, dynamic programming, branch and bound etc.
$\sum$	Several heuristic search algorithms have also been developed. In recent years expert systems,
abc	fuzzy approaches, simulated annealing and genetic algorithms have also been tested.
$\theta\sqrt{abc}$	This paper presents a survey of the literature
ζ	over the past fifteen years in the generator
∂	maintenance scheduling. The objective is to
sdf	present a clear picture of the available recent literature
ewq	of the problem, the constraints and the other aspects of
bvcn	the generator maintenance schedule.

Part I

Foundations

1

Introduction

1.1 High-dimensional data

This book concerns the analysis of data in which we are attempting to predict an outcome Y using a number of explanatory factors $X_1, X_2, X_3, \ldots$, some of which may not be particularly useful. Although the methods we discuss here can be used solely for prediction (i.e., as a "black box"), we generally adopt the perspective that we are also interested in understanding the relationship between X and Y. That is, we would like the statistical methods to be interpretable and to explain something about the relationship between the features and the outcome.

Regression models are an attractive framework for approaching problems of this type, with a long history, solid statistical foundations, and a rich catalog of extensions that were developed over the course of the 20th century. These classical tools were intended for situations in which the number of explanatory factors was relatively small, and indeed, tend to work very well in those situations.

Modern computation, however, has changed the way science is conducted. Computers along with automated technologies have enabled researchers to easily collect, store, and access data for large numbers of features. This phenomenon occurs over a wide range of fields, technological developments, and orders of magnitude; for example:

- Advances in information technology such as REDCap have made it easy to manage online surveys and assemble databases containing dozens, or even hundreds, of variables.
- The adoption of electronic medical records have made it possible to link diverse sources of clinical data and patient care information (lab tests, medications administered, vital signs, personal and family histories, etc.) and integrate them into data sets containing hundreds, or even thousands of variables.
- Molecular biology technologies such as microarrays and RNA-Seq have made it possible to systematically measure gene expression

across the entire transcriptome, consisting of tens of thousands of measurements per sample.

- Dramatic advances in genotyping in the wake of the Human Genome Project has enabled researchers to conduct genome-wide genetic association studies in which hundreds of thousands, or even millions of genetic variants are measured for each individual in the study.

The list above concentrates on manifestations of this phenomenon in medicine and biology, which reflects the backgrounds of the authors. Throughout this book we will generally illustrate the application of various methods using examples from these areas because it is what we are most familiar with and can speak most knowledgeably about, but the same phenomenon is, without question, occurring throughout all scientific disciplines. From economists analyzing financial transaction data to astronomers looking at electromagnetic spectra to chemists predicting the chemical activity of a compound from its physiochemical and structural properties, the general pattern of collecting information about a large number of features and attempting to predict a quantity of interest in a data-driven manner is now a pervasive approach in all areas of modern science.

This type of data is known as *high dimensional data*. Specifically, let n denote the number of independent observations (e.g., for a biomedical study, the number of patients or samples) and p denote the number of features recorded for each independent unit. In high-dimensional data, p is large with respect to n. Often, this means that p is larger than n, possibly much larger than n – this would certainly be the case for the gene expression and genetic association studies described above. However, many of the general principles and specific methods we describe in this book also pertain to situations in which p is smaller than n. For example, if $n = 100$ and $p = 80$, one may still fit a classical regression model, but the estimates will be highly variable and the analytic approach far from optimal.

It is worth noting that high dimensional data is not a synonym for "big data". There are many situations arising in modern data analysis in which n is extremely large (e.g., the type of databases compiled by companies like Facebook and Google). These also represent interesting challenges, both statistically and computationally, but are not the focus of this book.

We will adopt the following general notation throughout the book, all of which is fairly common in the regression literature. Let $\mathbf{X}$ denote the $n \times p$ matrix containing the predictor variables, with element x_{ij} recording the value of the jth feature for the ith independent unit, and

let $\mathbf{y}$ denote the length-n vector of response values. For the sake of simplicity, we begin by treating Y as a continuous, normally distributed variable (Chapters 1-8), but consider several other types of distributions for Y in Part III.

1.2 Large-scale univariate testing

A simple, widely used approach to analyzing high-dimensional data is to split the problem up into a large number of low-dimensional problems. Specifically, rather than trying to regress $\mathbf{y}$ simultaneously on all the features, we can carry out p separate single-variable regressions, one for each feature:

$$\begin{aligned} y_i &= \alpha_j + \beta_j x_{ij} + \epsilon_{ij} \\ \epsilon_{ij} &\overset{\perp\!\!\!\perp}{\sim} \mathrm{N}(0, \sigma^2); \end{aligned} \tag{1.1}$$

this approach is known as *marginal regression*.

The appeal of this approach is that the well-developed tools of classical regression methods can be easily applied to the separate analyses to yield estimates $\{\widehat{\beta}_j\}_{j=1}^p$, confidence intervals, and test hypotheses to produce p-values $\{p_j\}_{j=1}^p$. The obvious complication, however, is that this approach involves a large number of separate analyses that must somehow be combined into a single set of results. Thus, while novel statistical methodology may not be required to carry out the initial analyses, there has been a great deal of innovation over the past 30 years in terms of how to assess the results of a single analysis within the context of a large number of other, comparable analyses.

From a classical standpoint, this situation may be viewed as a problem of simultaneous hypothesis testing, with a primary concern for controlling the overall Type I error rate, also known as the family-wise error rate (FWER). If we wish to limit the probability of falsely rejecting any true null hypothesis to be less than or equal to α, we can compare each p_j to α/p, rejecting only those hypothesis for which $p_j \leq \alpha/p$. This approach is known as the *Bonferroni correction*, and is easily shown by Boole's inequality to satisfy FWER $\leq \alpha$. Many other methods exist for controlling the FWER, but the Bonferroni correction is the simplest and most widely used approach, and illustrates the basic idea of FWER control.

When p is large, strict FWER control can be extremely conservative. For example, suppose that out of the hypotheses being tested, 100

were rejected, with only one of those hypotheses falsely rejected. This seems like a successful result to many people, yet it is much too liberal according to FWER, because a Type I error has been committed. The fraction of false rejections out of the total rejected hypotheses is known as the *false discovery rate* (FDR). In a seminal and highly influential paper, Benjamini and Hochberg (1995) proposed the following rule: for any fixed value q, letting $p_{(1)}, p_{(2)}$ denote the sorted p-values and $j_{\max}$ denote the largest index for which $p_{(j)} \leq jq/p$, reject any hypothesis H_j for which $p_j \leq p_{(j_{\max})}$. Benjamini & Hochberg proved that this approach controls the expected FDR at the level q under the assumption that the tests are independent. In the decades since its original proposal, the idea of FDR control has become a widely accepted approach to carrying out simultaneous inference in the large-scale setting, and many authors have subsequently extended the idea in various ways, as well as explored the control of FDR under various situations of dependence among the tests.

Benjamini & Hochberg's perspective on carrying out separate analyses was entirely frequentist in the sense of controlling long-run proportions; however, false discovery rates blur the lines somewhat between estimation and testing as well as between frequentist and Bayesian approaches. False discovery rates can also be motivated from Bayesian (Storey and Tibshirani, 2003) as well as Empirical Bayes (Efron et al., 2001) perspectives. There is a large literature on false discovery rates and related approaches to carrying out inference for large numbers of comparable analyses; this literature lies outside the scope of this book. Our focus is on carrying out a single joint analysis of the relationship between the features and the response, rather than on simultaneous inference for a large number of separate analyses. Nevertheless, the idea of false discovery rates is certainly relevant in the context of variable selection, and we discuss the estimation of FDR in the context of high-dimensional regression in Chapters 6 and 8. For readers looking to learn more about the subject of large-scale univariate testing, we recommend Brad Efron's excellent book, *Large Scale Inference* (Efron, 2010).

1.3 High-dimensional modeling

Although carrying out separate univariate regressions is very common and relatively straightforward, there are several drawbacks to the marginal approach described in the previous section:

- Marginal regression fails to account for correlation among the features. Thus, many features are likely to appear significant even

though they are simply correlated with other features related to the outcome.

- For the same reason, marginal regression provides no way to estimate the independent effect of a feature while other features remain unchanged.
- By failing to incorporate features with predictive accuracy, weak associations between features and the outcome may be masked. In other words, marginal regression has lower power to discover relationships between features and the outcome than a joint regression analysis.
- Marginal regression provides no way to combine the predictions of each separate regression into a single response prediction.
- Finally, marginal regression provides no way of assessing the overall proportion of the variability in the outcome that may be explained by the features. In some contexts, this proportion is of considerable scientific interest; for example, in genetics, the proportion of variation in the phenotype that can be explained by genetic variation is known as the heritability.

These issues can only be resolved by considering a joint model of the relationship between $\mathbf{y}$ and the full set of features $\{\mathbf{x}_j\}_{j=1}^p$:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \epsilon_i \tag{1.2}$$
$$\epsilon_i \overset{\perp\!\!\!\perp}{\sim} \mathrm{N}(0, \sigma^2).$$

Unlike equation (1.1), here we are fitting a single model in which we estimate a vector of regression coefficients $\boldsymbol{\beta}$.

The maximum likelihood approach involves solving for the value of $\boldsymbol{\beta}$, known as the maximum likelihood estimator (MLE), that minimizes the residual sum of squares $\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$. Here, $\|\mathbf{v}\| = \sqrt{\sum_i v_i^2}$ denotes the Euclidean norm. This constitutes a linear system of equations whose solution is given by

$$\mathbf{X}^\top \mathbf{X} \widehat{\boldsymbol{\beta}} = \mathbf{X}^\top \mathbf{y} \tag{1.3}$$
$$\implies \widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \quad \text{if } \mathbf{X}^\top \mathbf{X} \text{ is invertible} \tag{1.4}$$

Because the estimator $\widehat{\boldsymbol{\beta}}$ is found by minimizing a sum of squares, it is often referred to as the least squares or "ordinary least squares" (OLS) estimate. The multiple regression least squares estimate has well-recognized

benefits such as yielding best linear unbiased estimates of $\boldsymbol{\beta}$, and resolves the issues raised in the list at the beginning of this section, such as yielding integrated predictions using all features simultaneously.

However, there are many drawbacks to the use of maximum likelihood for estimating $\boldsymbol{\beta}$ when p is large. Most dramatically, when $p > n$ the matrix $\mathbf{X}^\top\mathbf{X}$ is not invertible and equation (1.3) has no unique solution. However, it is not only the maximum of the likelihood that is affected by dimensionality. Even if $\mathbf{X}^\top\mathbf{X}$ can be inverted and a unique maximum identified, as the dimension grows and $\mathbf{X}^\top\mathbf{X}$ approaches singularity, the likelihood surface becomes very flat. This means that a wide range of values of $\boldsymbol{\beta}$ are consistent with the data and wide confidence intervals required to achieve, say, 95% coverage. In particular, $\mathbb{V}(\widehat{\boldsymbol{\beta}}) = \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$, with the consequence that as $p \to n$, $\mathbb{V}(\widehat{\boldsymbol{\beta}})$ increases without bound.

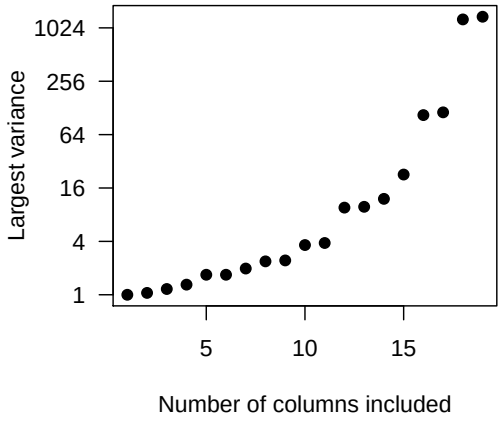

FIGURE 1.1
Largest variance of $\widehat{\boldsymbol{\beta}}$ as $p \to n$. The matrix has been standardized (see Section 1.6.2) and $\sigma^2 = n = 20$ to make the variance of the first estimate exactly 1.

To illustrate, consider a matrix $\mathbf{X}$ with $n = 20$ rows and $p = 19$ columns and whose elements consist of normally distributed random numbers. Figure 1.1 plots the largest variance of the $\widehat{\beta}_j$ estimates (i.e., the largest diagonal element of $\sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}$) as we include an increasing number of columns in $\mathbf{X}$. As the figure illustrates, the increase in variance is substantial as p approaches n, and infinite when $p \geq n$. Although we can still obtain unique solutions at $p = 15$, the variance of these solutions is over 20 times larger than when $p = 1$. In this example, the largest

variance is nondecreasing with respect to p; this holds for any matrix (Exercise 1.1).

Clearly, maximum likelihood cannot directly accommodate high dimensional data without running into serious problems with identifiability and inefficiency. However, when many features are unrelated to the outcome (in the sense that $\beta_j = 0$), it is possible to modify the maximum likelihood approach without abandoning it completely. Specifically, we could apply maximum likelihood only to the variables for which $\beta_j \neq 0$. By working with this smaller, identifiable subspace of $\mathbb{R}^p$, we can avoid the problems described above.

If we know in advance which elements of $\boldsymbol{\beta}$ are zero and which are not, then everything is straightforward: we simply fit model (1.2) using maximum likelihood, but including only the nonzero elements; this is known as the *oracle* model. Obviously, the oracle model is a theoretical gold standard, not a realistic approach to data analysis, as it would require consultation with an oracle that could tell you which features are related to the outcome and which are not. In the real world, we will have to use the data in order to make empirical decisions about which features are related to the outcome and which are not; this is known as *model selection*, and is discussed in the next section.

1.4 The model selection problem

The previous section pointed out the difficulties that arise in trying to use maximum likelihood estimation in high dimensions. Because these problems do not manifest themselves in low-dimensional maximum likelihood, a natural and very widespread strategy for dealing with high-dimensional data is to adopt a two-stage approach in which (1) we attempt to select the important parameters, then (2) apply maximum likelihood to the lower-dimensional space containing only these parameters. Unfortunately, using the same data for both purposes (model selection and inference) introduces substantial biases and invalidates the inferential properties that maximum likelihood typically possesses.

Example 1.1. To illustrate, consider the following simulation:

$$\begin{aligned} x_{ij} &\overset{\perp\!\!\!\perp}{\sim} \text{Unif}(0,1) \qquad \text{for } j \text{ in } 1,2,\ldots,100 \\ y_i &\overset{\perp\!\!\!\perp}{\sim} \text{N}(0,1) \end{aligned}$$

for i in $1,2,\ldots,25$. We will apply forward selection using BIC to identify the important variables up to a maximum of 5 selections, then fit a

maximum likelihood regression model using those variables, with the standard multiple linear regression result

$$\widehat{\boldsymbol{\beta}} \sim \mathrm{N}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}^\top\mathbf{X})^{-1}) \tag{1.5}$$

used to carry out hypothesis testing and construct confidence intervals. Finally, we repeat the entire process 100 times to get a sense of how well this works in general. □

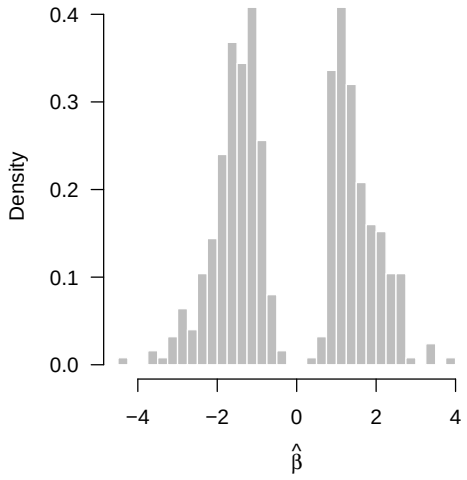

FIGURE 1.2
Sampling distribution of $\widehat{\beta}$ in the presence of bias arising from model selection. The true value is $\beta = 0$.

In general, this procedure performs very poorly. A histogram of the $\widehat{\beta}$ estimates resulting from this procedure is shown in Figure 1.2. As the figure illustrates, by using the data set for model selection as well as estimation and inference, we have distorted the actual sampling distribution of $\widehat{\boldsymbol{\beta}}$ far away from the sampling distribution (1.5) we use to carry out inference. This has dramatic consequences for the performance of this approach in terms of estimation, prediction, variable selection, and the validity of inference.

Estimation: As Figure 1.2 illustrates, the model selection process heavily biases the estimates of the regression coefficients away from zero. Most estimates are approximately ± 1.5 instead of being close to 0, the true value. In particular, the average value of $\widehat{\beta}_j^2$ is 2.7. In comparison, the expected value of $\widehat{\beta}_j^2$ for fitting a simple linear regression model to

this data in an unbiased manner (i.e, without the model selection step) is 0.52. Thus, the mean squared error of estimation is increased about 5-fold by the model selection process.

This is sometimes referred to as the phenomenon of the "winners' curse." Although linear regression is in general unbiased, model selection introduces selection bias. In particular, a coefficient is far more likely to be selected if its coefficient is overestimated; this is clearly seen in Figure 1.2. Thus, the true value of any estimate arising from a post-selection model is likely to be closer to zero than the MLE would indicate.

Variable selection: Here, we imposed an upper bound of 5 on the number of variables we allowed to be selected by the BIC-guided forward selection process. In all 100 of our replications, this upper bound was reached; in other words, the forward selection process would have continued to select additional variables if allowed to do so. Obviously, since the true model in this case is the null (intercept-only) model, the model selection process we have employed here results in systematic overfitting.

While it is true that in an asymptotic sense, using BIC for model selection will select the true model with probability tending to 1, that asymptotic argument relies on p remaining fixed while $n \to \infty$, or in other words, on $n \gg p$. Clearly, the asymptotic model-selection consistency of BIC is misleading in high-dimensional settings like this one, where BIC cannot be relied on for accurate variable selection. There have been efforts to modify BIC and correct this deficiency in high-dimensional settings (in particular the extended BIC, or EBIC (Chen and Chen, 2008)), but a fundamental challenge remains: in order to apply a criterion-based model selection approach, we must fit a large number of models and calculate the criterion for each model. In high dimensions, the number of possible models is 2^p and it is no longer possible to fit all models and rank them. Instead, we must use greedy, stepwise approaches such as the forward selection we have used here; such approaches have no guarantee that they will identify the best model.

Prediction: On average, the prediction error (defined in Chapter 1.5) of the selected model is 2.15. In comparison, the null model has a prediction error of $\mathbb{V}(Y) = 1$. Thus, by carrying out model selection, we have doubled its prediction error.

Inference: Finally, and perhaps most importantly, let us consider the validity of the inferences that we obtain from (1.5), ignoring the fact that we are using the data twice (once for selection and once for inference). From a hypothesis testing perspective, the p-values arising from testing $H_0 : \beta_j = 0$ for each of the selected coefficients range from 0.4 to 3×10^{-10}, with a median p-value of 0.0013. These results give the impression that rejecting H_0 and concluding that $\beta_j \neq 0$ is sound and

unlikely to produce many Type I errors. This impression, of course, is entirely misleading, as $\beta_j = 0$ for all j in this example.

We may also consider the validity of constructing 95% confidence intervals for the selected coefficients according to the usual linear regression procedure: $\widehat{\beta}_j \pm t_{.975,19}\text{SE}_j$, with $\text{SE} = \sqrt{\mathbb{V}(\widehat{\beta}_j)}$ and $\mathbb{V}(\widehat{\boldsymbol{\beta}})$ given by (1.5). The actual coverage achieved by this procedure with a nominal 95% coverage rate is less than 5%. As with hypothesis testing, ignoring selection effects when carrying out post-selection inference produces conclusions that are far too liberal, with actual errors accumulating at a much higher rate than the statistical inferential approaches would indicate.

This section paints a grim picture of the two-stage approach of selecting important variables and then applying maximum likelihood: estimates are biased away from zero, models are overfit, predictions are poor, p-values are far smaller than they should be, and confidence intervals are far narrower than they should be. In summary, this approach is wildly optimistic and overconfident. In most high-dimensional settings, there is considerable uncertainty in terms of the truly nonzero variables – ignoring this uncertainty, as in the two-stage approach, results in fundamentally invalid inferences.

These problems are, of course, widely recognized. They are also, unfortunately, widely ignored in practice. Post-selection confidence intervals and hypothesis tests are widely reported in the literature, and even regularly appear in introductory statistical textbooks on regression modeling.

A simple potential safeguard against these problems is to split the data into two components: one for model selection and one for inference. This avoids most, if not all, of the problems outlined above. However, this approach is often unsatisfactory for two reasons. First, such a split is arbitrary, with different choices potentially leading to substantially different results. This complicates the reproducibility of the analysis. Second, splitting the data involves reducing the sample size in half; this reduces the accuracy of both the model selection and the parameter estimation. Unless the sample size is very large, most analysts are unwilling to sacrifice half of their data for purposes of model validation.

The problem of developing statistical methods capable of simultaneous variable selection and inference has challenged statisticians for decades, from Scheffé (Scheffe, 1953) to the present; see Berk et al. (2013) and the citations therein for further references. One of the primary goals of this book is to demonstrate the extent to which recent developments in penalized regression address and alleviate the concerns about simultaneous selection and inference raised in this section.

1.5 Prediction

Before we introduce penalized regression, however, let us first discuss the general problem of how to choose between various models. With real data, using estimation accuracy is not possible, as estimation involves unknown population quantities. Prediction, however, depends on observable quantities and can be evaluated. Thus, the main idea of most approaches to model selection is that if model A predicts future observations better than model B, then we should prefer model A to model B. However, evaluating prediction error is not as straightforward as it may seem, and there are many competing approaches.

1.5.1 Prediction error

Consider the general regression model

$$y_i = f(\mathbf{x}_i) + \varepsilon_i,\ i = 1, \ldots, n,$$

where $\text{Var}(\varepsilon_i) = \sigma^2$. By fitting a model, we obtain $\hat{y}_i = \hat{f}(\mathbf{x}_i)$, the model's point prediction for y_i; for linear regression, $\hat{f}(\mathbf{x}_i) = \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}}$. Different models will of course produce different predictions. It is misleading, however, to evaluate predictive accuracy by comparing $\hat{y}_i$ to y_i: the observed value y_i has already been used to calculate $\hat{y}_i$, and is therefore not a genuine prediction. Indeed, for linear regression $\hat{y}_i$ is precisely calculated so that it minimizes the total squared difference between $\hat{y}_i$ and y_i, or the residual sum of squares:

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2.$$

Simply calculating RSS, then, will substantially overestimate the true predictive accuracy of the model (i.e., underestimate the prediction error). Instead, we must examine how well $\hat{f}(\mathbf{x})$ predicts *new* observations. However, there are two ways of defining what exactly we mean by "new" observations:

- PE_X: Fit using $(\mathbf{X}, \mathbf{y})$, predict $(\mathbf{X}, \mathbf{y}^{\text{new}})$
- PE: Fit using $(\mathbf{X}, \mathbf{y})$, predict $(\mathbf{X}^{\text{new}}, \mathbf{y}^{\text{new}})$

In the first scenario, new responses are obtained from the data generating mechanism for $y|\mathbf{x}$ but using the same feature values used to fit the model; this approach is therefore conditional on $\mathbf{X}$, hence the label

PE_X. In the second scenario, new features are obtained from the data generating mechanism, and then new responses from $y|\mathbf{x}^{\text{new}}$.

In principle, the first approach makes more sense if $\mathbf{X}$ represents a fixed design matrix, as in a controlled experiment. In this scenario, the concept of drawing new values of $\mathbf{x}$ from a distribution doesn't make sense. On the other hand, if the features are random (as they almost always are in high-dimensional settings), it doesn't make sense to fix $\mathbf{X}$ at the values used to fit the model – and even if it did, it's typically impossible to do so with real data if the features are random.

As we will discuss in the next section, information criteria approaches focus on PE_X, whereas cross-validation approaches (Chapter 2.5.2) focus on PE. Throughout this book, when presenting simulation studies involving prediction error, we present PE, as it seems to us more reasonable in the high-dimensional setting. For example, in Example 1.1, PE = 2.15 (as reported earlier) and PE_X = 1.74. Nevertheless, both approaches are reasonable ways to choose a model and both avoid the primary concern, which is to avoid bias due to overfitting.

1.5.2 Model selection criteria

Analytic model selection criteria typically focus on minimizing the expected prediction error

$$\mathbb{E}(\text{PE}_X) = \mathbb{E}\sum_{i=1}^{n}(y_i^{\text{new}} - \hat{y}_i)^2, \tag{1.6}$$

where the expectation is taken over both the original observations $\{y_i\}_{i=1}^n$ as well as the new observations $\{y_i^{\text{new}}\}_{i=1}^n$. It can be verified (Exercise 1.6) that

$$\mathbb{E}(\text{PE}_X) = \mathbb{E}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + 2\sum_{i=1}^{n}\text{Cov}(\hat{y}_i, y_i). \tag{1.7}$$

The expected prediction error consists of two terms. The first term is the within-sample fitting error; the second term is a bias correction factor that arises from the tendency of within-sample fitting error to underestimate out-of-sample prediction error, also known as the *optimism* of the model fit. The second term can also be considered a measure of model complexity. In fact, a general definition of the degrees of freedom of a model is

$$\text{df}(\lambda) = \sum_{i=1}^{n}\frac{\text{Cov}(\hat{y}_i, y_i)}{\sigma^2}. \tag{1.8}$$

This can also be written as

$$\mathrm{df} = \frac{\mathrm{tr}\{\mathrm{Cov}(\hat{\mathbf{y}}, \mathbf{y})\}}{\sigma^2},$$

where tr is the trace operator for a matrix, i.e., the sum of its diagonal elements.

Example 1.2. Consider the linear regression model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ with $\mathrm{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$. The least squares fitted value is given by

$$\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^-\mathbf{X}^\top\mathbf{y}.$$

Since $\mathrm{Cov}(\hat{\mathbf{y}}, \mathbf{y}) = \mathrm{Cov}(\mathbf{X}(\mathbf{X}^\top\mathbf{X})^-\mathbf{X}^\top\mathbf{y}, \mathbf{y}) = \sigma^2\mathbf{X}(\mathbf{X}^\top\mathbf{X})^-\mathbf{X}^\top$, we have

$$\mathrm{df} = \frac{\mathrm{tr}\{\mathrm{Cov}(\hat{\mathbf{y}}, \mathbf{y})\}}{\sigma^2} = \mathrm{tr}(\mathbf{X}(\mathbf{X}^\top\mathbf{X})^-\mathbf{X}^\top) = \mathrm{rank}(\mathbf{X}).$$

This agrees with our usual definition of degrees of freedom. □

We now give brief descriptions of various model selection criteria that can be used for model selection. A comprehensive description and the full derivations of these methods is beyond the scope of this book; our goal here is to introduce and motivate the various criteria.

To begin, let us turn our attention back to equation (1.7). We have just discussed the second term in this expression, df. For the first term, we can reasonably estimate $\mathbb{E}\sum_{i=1}^n(\hat{y}_i(\lambda) - y_i)^2$ by its observed value, the residual sum of squares. Doing so, and dividing by σ^2 to put both terms on the same scale, we obtain a criterion known as the C_p statistic:

$$C_p = \frac{\mathrm{RSS}(\lambda)}{\sigma^2} + 2\mathrm{df}(\lambda). \tag{1.9}$$

One downside of the C_p statistic is that it requires an estimate of σ^2. As we shall see in Section 2.6.1, estimation of σ^2 is not a trivial matter, particularly in high dimensional models.

The C_p criterion explicitly focuses on least squares as an objective. The *Akaike information criterion* (AIC) is a generalization of the C_p idea to maximum likelihood models. Rather than consider the expected value of $\{y_i^{\mathrm{new}} - \hat{y}_i(\hat{\theta})\}^2$, Akaike proposed estimating the expected value of $\log p(y_i^{\mathrm{new}}|\hat{\theta})$, where $\hat{\theta}$ denotes the estimated parameters of the model based on the original data $\{y_i\}_{i=1}^n$. Asymptotically, a relationship similar to equation (1.7) can be shown to hold for maximum likelihood estimation, yielding

$$\mathrm{AIC} = 2L(\hat{\theta}|\mathbf{X}, \mathbf{y}) + 2\mathrm{df}, \tag{1.10}$$

where, as with C_p, the expected value $\mathbb{E}\sum_{i=1}^{n}\log p(y_i|\hat{\theta})$ has been replaced by its observed value, the log-likelihood. For the normal distribution,

$$\text{AIC} = n\log\sigma^2 + \frac{\text{RSS}}{\sigma^2} + 2\text{df} + \text{constant}.$$

Thus, in the case of normally distributed errors with known variance σ^2, AIC and C_p are equivalent up to a constant.

A rather different approach is to consider model selection from a Bayesian perspective. Letting M denote a given model, we would be interested in calculating the posterior probability of M given the data, $\mathbb{P}(M|\mathbf{X},\mathbf{y})$. If we assume a uniform prior across all models, then $\mathbb{P}(M|\mathbf{X},\mathbf{y}) \propto \mathbb{P}(\mathbf{y}|\mathbf{X},M)$. In general, calculating this quantity involves numerical integration, but this integral can be approximated to yield

$$\log\mathbf{P}(\mathbf{y}|\mathbf{X},M) \approx -L(\hat{\theta}|\mathbf{X},\mathbf{y}) - \frac{1}{2}\text{df}\log(n).$$

The *Bayesian information criterion* (BIC) is defined as -2 times this quantity:

$$\text{BIC} = 2L(\hat{\theta}|\mathbf{X},\mathbf{y}) + \text{df}\log(n). \tag{1.11}$$

Thus, choosing the model with the smallest BIC is (approximately) equivalent to choosing the model with the highest posterior probability.

Note that, despite the very different derivations, the equations for AIC and BIC are surprisingly similar; the only difference is $\log(n)$ instead of 2 as the multiplicative factor for df. In practice, this means that BIC applies a heavier penalty to model complexity than does AIC (provided $n \geq 8$) and will therefore favor more parsimonious models.

1.6 Penalized regression

1.6.1 Penalized likelihood

The *likelihood* function $\ell(\theta|\text{Data})$ is defined as the probability distribution $p(\text{Data}|\theta)$, but considered as a function of the unknown parameter θ, conditional on the observed data, as opposed to the probability distribution, which describes the probability of observing various values of the data for a fixed θ. Here, we are using p as a generic function to denote

probability, probability density, or probability measure as appropriate for the situation.

Throughout this book, we will use the notation L to refer to the negative log-likelihood:

$$L(\theta|\text{Data}) = -\log \ell(\theta|\text{Data}) \tag{1.12}$$
$$= -\log p(\text{Data}|\theta). \tag{1.13}$$

Here, the function L is known as the *loss function* and we seek estimates with a reasonably low loss. This is equivalent to finding a value (or interval of values) with an acceptably high likelihood; the distinction between maximizing a likelihood and minimizing a loss is arbitrary, but we will use the loss function approach in agreement with the bulk of the current literature on penalized regression.

In the context of linear regression, the loss function is

$$\begin{aligned} L(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) &= \frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sum_i (y_i - \mathbf{x}_i^\top \boldsymbol{\beta})^2 \\ &= \frac{1}{2\sigma^2}\sum_i (y_i - \mathbf{x}_i^\top \boldsymbol{\beta})^2 + \text{constant}. \end{aligned}$$

For the purposes of likelihood-based inference, it is only the difference in loss functions between two values, $L(\boldsymbol{\beta}_1|\mathbf{X}, \mathbf{y}) - L(\boldsymbol{\beta}_2|\mathbf{X}, \mathbf{y})$, i.e., the likelihood ratio, that is relevant. Thus, we can ignore the leading term in the first equation above. For the purposes of finding the MLE, the $(2\sigma^2)^{-1}$ factor may also be ignored, although we must account for it when constructing likelihood-based intervals and answering other inferential questions.

As we have seen, working with the above likelihood directly is problematic in high dimensions. Reducing the dimensionality through model selection allows for some progress, but has several shortcomings. An alternative way of dealing with the problem is to introduce a penalty. Instead of the likelihood $L(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y})$, consider the function

$$Q(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) = L(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) + P_\lambda(\boldsymbol{\beta}), \tag{1.14}$$

where P is a *penalty function* that penalizes what one would consider less realistic values of the unknown parameters, and λ is a *regularization parameter* that controls the tradeoff between the two components. The combined function Q is known as the *objective function*.

The parameter λ controls the tradeoff between the penalty and the model fit, as measured by the likelihood. It is worth considering what happens to Q as we change λ. As $\lambda \to 0$, Q approaches L and we are back where we started in terms of finding the optimal values of a nearly

flat function. In other words, if λ is too small, we will tend to overfit the data and obtain estimates with high variance and wide confidence intervals. On the other hand, as $\lambda \to \infty$, the penalty dominates the objective function and all solutions will be close to zero. When λ is too large, we will tend to underfit the data and end up with estimates that are heavily biased towards zero. Thus, λ is directly responsible for balancing the bias-variance tradeoff; obviously, selection of λ is a very important practical aspect of fitting penalized regression models.

What exactly do we mean by "less realistic" values? The most common use of penalization is to impose an *a priori* belief that small regression coefficients are more likely than large ones; i.e., that we would not be surprised if β_j was 1.2 or 0.3, but would be very surprised if β_j was 9.7×10^4. Without penalization, all of these values are equally likely unless the data alone can rule them out, which, as we have seen, is difficult to accomplish in high dimensions. In Part IV of this book, we consider other uses for penalization to reflect beliefs that the true coefficients may be grouped into hierarchies, or display a spatial pattern such that β_j is likely to be close to β_{j+1}.

Some care is needed in the application of the idea that small regression coefficients are more likely than large ones. First of all, it typically does not make sense to apply this idea to the intercept, unless you happened to have some reason to think that the mean of Y should be zero. Hence, the intercept is not included in the penalty; if it were, coefficient estimates would not be invariant to changes of location.

Furthermore, the size of the regression coefficient depends on the scale with which the associated feature is measured; depending on the units $\mathbf{x}_j$ is measured in, $\beta_j = 9.7 \times 10^4$ might, in fact, be realistic. This is a particular problem if different features are measured on different scales, as the penalty would not have an equal effect on all coefficient estimates. To avoid this issue and ensure invariance to scale, features are usually *standardized* prior to model fitting to have mean zero and standard deviation 1:

$$\sum_{i=1}^{n} x_{ij} = 0$$

$$\sum_{i=1}^{n} x_{ij}^2 = n$$

for all j. This can be accomplished without any loss of generality, as any location shifts for $\mathbf{X}$ are absorbed into the intercept and scale changes

can be reversed after the model has been fit:

$$\begin{aligned} x_{ij}\beta_j &= \frac{x_{ij}}{a} a\beta_j \\ &= \tilde{x}_{ij}\tilde{\beta}_j; \end{aligned}$$

i.e., if we had to divide $\mathbf{x}_j$ by a to standardize it, we simply divide the transformed solution $\tilde{\beta}_j$ by a to obtain β_j on the original scale.

Centering and scaling the explanatory variables has added benefits in terms of computational savings and conceptual simplicity. The features are now orthogonal to the intercept term, meaning that in the standardized covariate space, $\widehat{\beta}_0 = \bar{y}$ regardless of what goes on in the rest of the model. In other words, if we center y by subtracting off its mean, we don't even need to estimate β_0. Also, standardization simplifies the solutions; to illustrate with simple linear regression,

$$\begin{aligned} \widehat{\beta}_0 &= \bar{y} - \widehat{\beta}\bar{x} \\ \widehat{\beta}_1 &= \frac{\sum(y_i - \bar{y})(x_i - \bar{x})}{\sum(x_i - \bar{x})^2} \end{aligned}$$

However, if we center and scale $\mathbf{x}$ and center $\mathbf{y}$, then we get the much simpler expression $\widehat{\beta}_0 = 0$, $\widehat{\beta}_1 = \mathbf{x}^\top \mathbf{y}/n$. As we will see throughout the book, the clarity afforded by these simpler expressions makes it considerably easier to see the effect of various penalties.

1.6.2 Ridge regression

As mentioned in the previous section, it is typically reasonable to assume that small regression coefficients are more likely than large ones. In other words, if two values of $\widehat{\boldsymbol{\beta}}$ provided equally satisfactory fits to the data, the estimate with the smaller values of $\widehat{\boldsymbol{\beta}}$ would be considered more realistic. To obtain penalized regression estimates according to this principle, we should choose a penalty that discourages large regression coefficients. A natural choice is to penalize the sum of squares of the regression coefficients:

$$P_\tau(\boldsymbol{\beta}) = \frac{1}{2\tau^2} \sum_{j=1}^{p} \beta_j^2. \tag{1.15}$$

Applying this penalty in the context of penalized regression is known as *ridge regression*, and has a long history in statistics, dating back to 1970 (Hoerl and Kennard, 1970).

For linear regression, the ridge penalty is particularly attractive to

work with because the maximum penalized likelihood estimator has a simple closed form solution. The ridge regression objective function is

$$Q(\boldsymbol{\beta}|\mathbf{X},\mathbf{y}) = \frac{1}{2\sigma^2}\sum_i (y_i - \mathbf{x}_i^\top\boldsymbol{\beta})^2 + \frac{1}{2\tau^2}\sum_{j=1}^p \beta_j^2. \tag{1.16}$$

For the purposes of minimizing this objective function, it is often convenient to multiply the above objective function by the constant σ^2/n; as we will see throughout this book, doing so in penalized regression problems often considerably simplifies the mathematical expressions involved. Rewriting the above equation, we have

$$Q(\boldsymbol{\beta}|\mathbf{X},\mathbf{y}) = \frac{1}{2n}\sum_i (y_i - \mathbf{x}_i^\top\boldsymbol{\beta})^2 + \frac{\lambda}{2}\sum_{j=1}^p \beta_j^2, \tag{1.17}$$

where $\lambda = \sigma^2/(n\tau^2)$. This objective function is differentiable, and it is straightforward to show (Exercise 1.3) that its minimum occurs at

$$\widehat{\boldsymbol{\beta}} = (n^{-1}\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}n^{-1}\mathbf{X}^\top\mathbf{y}, \tag{1.18}$$

The solution is similar to the least squares solution (1.4), but with the addition of a "ridge" down the diagonal of the matrix to be inverted. Note that, after standardizing $\mathbf{X}$ and $\mathbf{y}$ and writing the objective function in terms of λ, the ridge solution is a relatively simple function of the marginal regression solutions $n^{-1}\mathbf{X}^\top\mathbf{y}$ and the correlation matrix $n^{-1}\mathbf{X}^\top\mathbf{X}$.

As discussed in Section 1.3, the maximum likelihood estimator is not always unique. If $\mathbf{X}$ is not full rank, $\mathbf{X}^\top\mathbf{X}$ is not invertible and there is no unique value of $\boldsymbol{\beta}$ that maximizes the likelihood. This problem does not occur with ridge regression, however, as the following theorem demonstrates.

Theorem 1.1. *For any design matrix $\mathbf{X}$, the quantity $n^{-1}\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$ is always invertible provided that $\lambda > 0$; thus, there is always a unique solution $\widehat{\boldsymbol{\beta}}$.*

The proof of Theorem 1.1 is left as Exercise 1.5.

To understand the effect of the ridge penalty on the estimator $\widehat{\boldsymbol{\beta}}$, it helps to consider the special case of an orthonormal design matrix ($\mathbf{X}^\top\mathbf{X}/n = \mathbf{I}$). In this case, we can factor out a $(1+\lambda)$ term from (1.18) and see that

$$\widehat{\beta}_j = \frac{\widehat{\beta}_j^{\text{OLS}}}{1+\lambda}. \tag{1.19}$$

This illustrates the essential feature of ridge regression: *shrinkage*. The primary effect of applying ridge penalty (1.15) is to shrink the estimates toward zero. Doing so introduces bias but can considerably reduce the variance of the estimate.

Example 1.3. The benefits of ridge regression are most striking in the presence of multicollinearity. Consider the following very simple simulated example:

```
> x1 <- rnorm(20)
> x2 <- rnorm(20, mean=x1, sd=.01)
> y <- rnorm(20, mean=3+x1+x2)
> fit <- lm(y~x1+x2)
> coef(fit)
(Intercept)          x1          x2
   3.021159   21.121729  -19.089170
```

In this case, although there are only two covariates, the strong correlation between X_1 and X_2 causes a great deal of trouble for maximum likelihood. Although in truth, $\beta_1 = \beta_2 = 1$, all pairs of (β_1, β_2) values that add up to 2 yield virtually the same likelihood, including the maximum likelihood pair, $(40, -38)$. The likelihood surface is very flat here and there is a tremendous amount of uncertainty about β_1 and β_2.

When we introduce the added assumption that small coefficients are more likely than large ones by using a ridge penalty, however, this uncertainty is resolved. Using the `ridge` function provided by the `hdrm` package to fit the ridge regression model, we obtain a solution that is clearly much closer to the truth than the MLE:

```
> fit <- ridge(y~x1+x2)
> coef(fit, lambda=0.1)
(Intercept)          x1          x2
  3.0327231   0.9575176   0.9421784
```

Note that the syntax of `ridge` is similar to that of `lm`, although one must specify a value of λ in the call to `coef`. □

An obvious question is whether the ridge regression estimates are systematically closer to the truth than MLEs are, or whether the above example is a fluke. To address this question, let us first derive the bias and variance of ridge regression, The variance of the ridge regression estimate is

$$\mathrm{Var}(\widehat{\boldsymbol{\beta}}) = \tfrac{\sigma^2}{n}\mathbf{W}(\tfrac{1}{n}\mathbf{X}^\top\mathbf{X})\mathbf{W},$$

where $\mathbf{W} = (\frac{1}{n}\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}$. Meanwhile, the bias is

$$\mathrm{Bias}(\widehat{\boldsymbol{\beta}}) = -\lambda\mathbf{W}\boldsymbol{\beta}.$$

Both bias and variance contribute to overall accuracy, as measured by mean squared error (MSE):

$$\begin{aligned}\mathrm{MSE}(\widehat{\boldsymbol{\beta}}) &= \mathbb{E}\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2 \\ &= \sum_j \mathrm{Var}(\widehat{\boldsymbol{\beta}}_j) + \sum_j \mathrm{Bias}(\widehat{\boldsymbol{\beta}}_j)^2.\end{aligned}$$

Theorem 1.2 (Existence Theorem)**.** *There always exists a value λ such that*

$$MSE\left(\widehat{\boldsymbol{\beta}}_\lambda\right) < MSE\left(\widehat{\boldsymbol{\beta}}^{OLS}\right).$$

This is a rather surprising result with somewhat radical implications: despite the typically impressive theoretical properties of maximum likelihood and linear regression, and even if the model we fit is exactly correct and the outcome exactly follows the distribution we specify, we can *always* obtain a better estimator by shrinking the MLE towards zero.

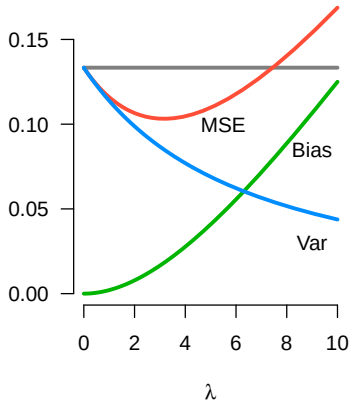

FIGURE 1.3
Variance, bias, and MSE of ridge regression. The horizontal gray line at 0.13 is the MSE of maximum likelihood (OLS).

The proof of Theorem 1.2 is left as Exercise 1.8, but the heuristic idea of the proof can be seen in Figure 1.3. As shown in the figure, the total variance ($\sum_j \mathrm{Var}(\widehat{\beta}_j)$) is a monotone decreasing sequence with respect to λ, while the total squared bias ($\sum_j \mathrm{Bias}^2(\widehat{\beta}_j)$) is a monotone increasing sequence with respect to λ. The MSE, on the other hand, is not monotone but decreases up to an optimal value of λ, then increases.

Because maximum likelihood is a special case of ridge regression with $\lambda = 0$, proving Theorem 1.2 amounts to showing that the derivative of the ridge MSE at 0 is negative, thus implying the existence of a region $(0, \lambda^*)$ over which ridge regression has a lower MSE than maximum likelihood.

Figure 1.3 was generated from a design matrix containing two features with a correlation of 0.5. The general contours of the figure are representative of ridge regressions in general, although specific quantities, such as the extent to which ridge outperforms OLS and the range over which this happens, depend on $\mathbf{X}$. For example, in the simulated, strongly correlated example from earlier in this section, ridge outperforms OLS (in terms of MSE) at $\lambda = 1$ by a factor of over 10^5 – certainly not a fluke.

1.6.3 Bayesian interpretation

Bayesian inference provides a more formal justification for the proposal of the penalty term. From a Bayesian perspective, one can think of the penalty as arising from a formal prior distribution on the parameters. Suppose that, given $\boldsymbol{\beta}$, $\mathbf{y}$ has conditional density $p(\mathbf{y}|\boldsymbol{\beta})$. Let $p(\boldsymbol{\beta})$ be the prior for $\boldsymbol{\beta}$. Then the posterior density is

$$p(\boldsymbol{\beta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\beta})p(\boldsymbol{\beta})}{p(\mathbf{y})} \propto p(\mathbf{y}|\boldsymbol{\beta})p(\boldsymbol{\beta}) \tag{1.20}$$

where the proportionality constant is independent of $\boldsymbol{\beta}$. Expressing (1.20) on the log scale,

$$\log p(\boldsymbol{\beta}|\mathbf{y}) = \log p(\mathbf{y}|\boldsymbol{\beta}) + \log p(\boldsymbol{\beta}) + C,$$

where C is a constant with respect to $\boldsymbol{\beta}$.

This is exactly the form of the objective function from (1.14). By optimizing this objective function, we are finding the mode of the posterior distribution of $\boldsymbol{\beta}$; this is known as the *maximum a posteriori*, or MAP, estimate.

Specifically, suppose that we assume the prior

$$\beta_j \overset{\perp\!\!\!\perp}{\sim} N(0, \tau^2).$$

The resulting log-posterior is exactly (1.16), up to a constant. Furthermore, because this prior is conjugate for linear regression (ignoring uncertainty about σ^2), the posterior distribution $p(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y})$ is also multivariate normal, and the ridge regression estimator (1.18) is the posterior mean in addition to being the posterior mode. Finally, the regularization parameter λ is the ratio of the prior precision $(1/\tau^2)$ to the information

(n/σ^2). This is entirely in agreement with our earlier remarks that λ quantifies the balance between fit (likelihood) and penalty (prior).

Thus, we arrive at the same estimator $\widehat{\boldsymbol{\beta}}$ whether we view it as a modified maximum likelihood estimator or a Bayes estimator. In other respects, however, the similarity between Bayesian and Frequentist breaks down. Two aspects, in particular, are worthy of mention. First is the inferential goal of constructing intervals for $\boldsymbol{\beta}$ and what properties such intervals should have. Frequentist confidence intervals are required to maintain a certain level of coverage for any fixed value of $\boldsymbol{\beta}$. Bayesian posterior intervals, on the other hand, may have much higher coverage at some values of $\boldsymbol{\beta}$ than others. For example, the Bayes coverage for a 95% posterior interval at $\beta_j \approx 0$ may be $> 99\%$, but only $\approx 20\%$ for larger values of β_j. The interval would nevertheless maintains 95% coverage across a collection of β_j values, integrated with respect to the prior. These are two rather different requirements, and, in the context of informative priors such as those in penalized regression, inherently incompatible.

The other aspect in which a clear divide emerges between Bayes and Frequentist perspectives is with regard to the specific value $\beta = 0$. From a Bayesian perspective, the posterior probability that $\beta = 0$ is 0 because its posterior distribution is continuous (unless one places a point mass at zero; this is an entirely different class of models that lies outside the scope of this book). From a Frequentist perspective, however, the notion of testing whether $\beta = 0$ is still meaningful and indeed, often of interest in an analysis.

The majority of research into penalized regression methods has focused on point estimation and its properties, so these inferential differences between Bayesian and Frequentist perspectives are relatively unexplored. Indeed, inferential methods of any kind for penalized regression have been largely lacking, although this is starting to change. We discuss various approaches to inference in Chapters 5-8. Throughout, the perspective of this book is generally aligned with maximum likelihood theory, although as we will see, the appearance of a penalty in the likelihood somewhat blurs the lines between Bayes and Frequentist ideas.

1.6.4 Selection of λ

In order to apply one of the model selection criteria introduced in Section 1.5.2, we must first work out the degrees of freedom for ridge regression. Fortunately, this is straightforward for any linear fitting method.

Example 1.4. A model fitting method is said to *linear* if we can write

$\hat{\mathbf{y}} = \mathbf{S}\mathbf{y}$, where $\mathbf{S}$ is an $n \times n$ matrix depending on the predictors and certain tuning parameters. Suppose $\mathrm{Var}(\mathbf{y}) = \sigma^2\mathbf{I}$. The effective number of parameters, or generalized degrees of freedom is

$$\mathrm{df} = \mathrm{tr}(\mathrm{Cov}(\hat{\mathbf{y}}, \mathbf{y}))/\sigma^2 = \mathrm{tr}(\mathbf{S}).$$

Note that ridge regression is a linear fitting method, with

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

Thus,

$$\mathrm{df}(\lambda) = \mathrm{tr}(\mathbf{X}(\mathbf{X}^\top\mathbf{X} + n\lambda\mathbf{I})^{-1}\mathbf{X}^\top) = \sum_{j=1}^{p} \frac{d_j}{d_j + \lambda}.$$

where $d_1, \ldots, d_p$ are the eigenvalues of $n^{-1}\mathbf{X}^\top\mathbf{X}$. □

Using this result, we can use AIC, BIC, or some other information criteria to choose λ. As Example 1.4 shows, the model selection involved in penalized regression is continuous. As we change λ, we gradually increase the complexity of the model, and small changes in λ result in small changes in estimation. This is in sharp contrast to best subsets model selection, where complexity is added by discrete jumps as we introduce parameters, and adding just a single parameter can introduce large changes in model estimates.

An alternative to information criteria is to leave observations out of the fitting process and save them to use for evaluating predictive accuracy. In general, this involves cross-validation, which will be discussed in greater detail in Section 2.5.2. However, for linear fitting methods, there is an elegant closed-form solution to the leave-one-out cross-validation error that does not require actually refitting the model. Let $\hat{f}_{(-i)}$ denote the fitted model with observation i left out. It can be shown (Exercise 1.10) that

$$\sum_i \left\{ y_i - \hat{f}_{(-i)}(x_i) \right\}^2 = \sum_i \left(\frac{y_i - \hat{f}(x_i)}{1 - s_{ii}} \right)^2, \tag{1.21}$$

where s_{ii} is the ith diagonal element of $\mathbf{S}$.

1.6.5 Case study: Air pollution data

To illustrate ridge regression in practice, we will now consider a study designed to estimate the relationship between pollution and mortality while adjusting for the potentially confounding effects of climate and socioeconomic conditions. To quantify pollution, "relative pollution potential" was measured for three pollutants – hydrocarbons (HC), nitrogen

oxides (NOX), and sulfur dioxide (SO_2) – in 60 Standard Metropolitan Statistical Areas in the United States between 1959-1961. The outcome of interest is total age-adjusted mortality from all causes, in deaths per 100,000 population. In total, there are $p = 15$ explanatory variables: the three pollution variables, 8 demographic/socioeconomic variables, and 4 climate variables. Although few would consider $p = 15$ "high-dimensional", the full maximum likelihood model nevertheless struggles with a sample size of just 60 and strong correlation among several variables, leaving it unable to provide a trustworthy answer to the primary question of the relationship between pollution and mortality.

For the sake of easily comparing the effects of the various explanatory variables, they have been standardized as described in Section 1.6.1. As a result, the interpretation of β is the same for all variables, quantifying the increase in deaths per 100,000 population that would be expected if the variable were to increase by 1 standard deviation. The following code uses the `hdrm` package to read in the data, perform this standardization, fit, and plot the model:

```
loadData("pollution")
XX <- std(X)
fit <- ridge(XX, y)
plot(fit, xaxis="both")
```

Figure 1.4 shows the fit of the ridge regression model as a function of λ. Such a plot was originally called the *ridge trace*; more recent authors refer to it as the *coefficient path*. The most salient trend is the role of λ as the regularization parameter: as $\lambda \to 0$, $\widehat{\boldsymbol{\beta}}^{\text{ridge}} \to \widehat{\boldsymbol{\beta}}^{\text{OLS}}$, while as $\lambda \to \infty$, $\widehat{\boldsymbol{\beta}}^{\text{ridge}} \to \mathbf{0}$.

It is particularly instructive to look at the coefficient paths of the three pollution parameters, all of which are fairly highly correlated with each other. At small λ values, the estimates indicate that NOX pollution has a very strong harmful effect, while HC pollution has a very strong protective effect. This result is surprising, and indeed rather difficult to believe – increasing the amount of HC pollution should *save* 60 lives per 100,000? We are witnessing an instance of the phenomenon discussed in Section 1.6.2. Judged purely by likelihood, it is plausible that one type of pollution is highly detrimental and the other is highly beneficial – the two types of pollution are highly correlated and as a result, the likelihood is very flat along the HC-NOX axis. However, as we increase the ridge penalty, we see that the estimated effects for these two types of pollution quite rapidly drop to near zero.

A parallel story is told by examining the SO_2 coefficient path. SO_2 is correlated with HC and NOX (although not as highly correlated as

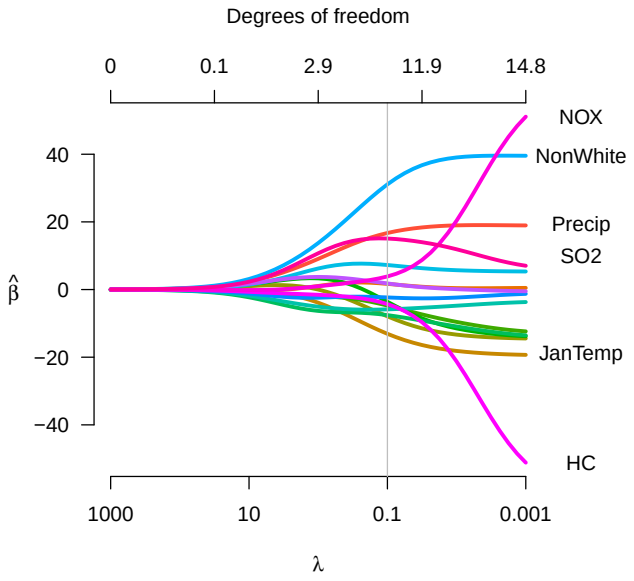

FIGURE 1.4
Ridge regression estimates as a function of λ, which is presented on a log scale. The SO_2 path is the one that is similar to the "Precip" path until $\lambda \approx 0.1$, then decreases back towards $\widehat{\beta} \approx 5$. A vertical line is drawn at the λ value that minimizes GCV.

HC and NOX are with each other), so its solution is affected by the estimated effects for the other two pollutants. In particular, while most of the other coefficient estimates increase monotonically as λ decreases from ∞ to 0, the estimated effect of SO_2 goes up, then decreases. As a result, depending on the value of λ one chooses, SO_2 pollution is either far more important, or far less important, than HC and NOX pollution.

Figure 1.5 illustrates GCV and RSS for the pollution data, as a function of λ; these values are calculated and returned by `ridge` as `fit$GCV` and `fit$RSS`, respectively. As we remarked earlier, RSS underestimates the true prediction error for all values of λ, but the problem is most severe for small λ values. Very large λ values are clearly not ideal, as they produce quite poor predictions. The estimated prediction error falls sharply until we reach $\lambda \approx 0.1$, then starts to increase, much as we saw in Figure 1.3. Note that we do not see this increase if we look at RSS, as its increasing tendency to underestimate the prediction error dominates

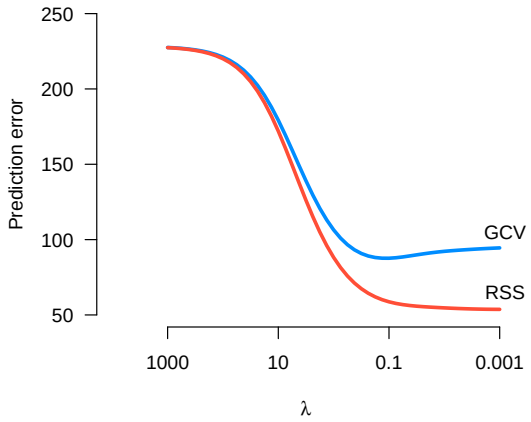

FIGURE 1.5
Fitting/prediction error, as estimated by GCV and underestimated by the observed RSS.

the actual decline in accuracy. The value that minimizes GCV occurs at $\lambda = 0.1$; this is illustrated by the vertical line in Figure 1.4.

TABLE 1.1
t-values for ridge regression ($\lambda = 0.1$) and OLS.

	Ridge	OLS
NonWhite	3.90	3.36
SO2	2.72	0.58
Precip	2.47	2.06
Density	1.41	0.91
NOX	0.37	1.33
Humidity	0.32	0.09
Poor	0.21	-0.05
WhiteCol	-0.38	-0.12
HC	-0.42	-1.37
House	-0.52	-1.53
Over65	-0.61	-1.07
Sound	-0.86	-0.37
Educ	-1.03	-1.44
JulyTemp	-1.11	-1.63
JanTemp	-1.77	-1.75

Finally, Table 1.1 presents the statistical significance of each term in the model, expressed as the ratio of the estimate to the standard error (equivalently here, ratio of the posterior mean to the posterior standard deviation), where we take the standard error to be the square root of the diagonal elements of

$$\nabla^2_{\boldsymbol{\beta}} Q^{-1} = \frac{\sigma^2}{n}(n^{-1}\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}.$$

Calculating this quantity requires an estimate of σ^2; a reasonable estimator is $\hat{\sigma}^2 = \text{RSS}/(n-\text{df})$, by analogy with OLS regression. We refer to this ratio as the t-value; one could obtain p-values by comparing against the t distribution with $n - \text{df}$ degrees of freedom, although we will be content with the observation that $|t| > 2$ indicates $\beta_j = 0$ is unlikely in light of the data. Standard errors and t-values can be obtained from a `ridge` fit via `summary(fit)`.

Table 1.1 presents the t-values for both ridge regression and OLS (maximum likelihood). As one might expect from Figure 1.4, SO_2 is much more significant under ridge regression than under maximum likelihood, with NOX and HC much less significant. Of particular note, however, is the fact that several variables, such as NonWhite (percentage of population that is nonwhite) and Precip (mean annual precipitation) are more significant at $\lambda = 0.1$ than at $\lambda = 0$. Thus, although the *estimates* have been shrunken towards zero, the significance (i.e., the evidence against $\beta = 0$) has actually increased.

1.7 Shrinkage and selection

The major limitation of ridge regression is the fact that all of its coefficients are nonzero. This poses two considerable problems for high-dimensional regression. The first is that the solutions become very difficult to interpret – it is difficult to understand a model with hundreds or thousands of parameters. The second reason is a computational one. In high dimensions, ridge regression can be rather slow. The reason is that, as we can see from equation (1.18), solving for $\widehat{\boldsymbol{\beta}}$ involves inverting (or at least factoring) a $p \times p$ matrix, which carries a heavy computational burden when p is large.

Another concern with ridge regression is the heavy shrinkage towards zero that it imposes. The squared term in the ridge penalty means that large values of β_j are judged to be extremely unlikely. In many situations, we expect a large number of β coefficients to be zero or very close to

zero, but the nonzero beta values to be somewhat large. In situations like this, the ridge regression estimates exhibit a heavy bias towards zero.

It is desirable, then, to have models which allow for both shrinkage *and selection*. In other words, we would like penalized regression methods that allow us to retain the benefits of ridge regression while at the same time selecting a subset of important variables. These are the sorts of models we will consider throughout the remainder of this book. However, it is worth remembering the lessons of ridge regression and the benefits of shrinkage as we proceed.

1.8 Exercises

1.1. *Increasing variance upon feature addition.* Let $\mathbf{X}$ denote a matrix with full column rank and $\mathbf{x}$ denote a new column that we are considering adding to $\mathbf{X}$ to form $\mathbf{X}^*$. Let $v(\mathbf{X})$ denote the largest variance (i.e., the largest diagonal element of the covariance matrix) of $\widehat{\boldsymbol{\beta}}$, the OLS estimate using $\mathbf{X}$ as a design matrix.

(a) Show that $v(\mathbf{X}^*) \geq v(\mathbf{X})$.

(b) Under what circumstance does $v(\mathbf{X}^*) = v(\mathbf{X})$?

Remark: This exercise examines the largest diagonal element of $\mathbb{V}(\widehat{\boldsymbol{\beta}})$. The same phenomenon occurs if we were to consider instead the largest eigenvalue of $\mathbb{V}(\widehat{\boldsymbol{\beta}})$, a consequence of the Cauchy interlacing eigenvalue theorem.

1.2. *Model selection using marginal regression.* Carry out a simulation study along the lines of Section 1.4, where y follows a normal distribution and the predictors follow independent uniform distributions. In this study, however, use marginal regression to select the 5 most significant variables to include in the OLS model. Use $n = 50$ and vary p along the set $\{5, 50, 500, 5000\}$; repeat the procedure $N = 1,000$ times (i.e., in the end, you will obtain $5,000$ coefficient estimates for each value of p). Construct plots illustrating how MSE, PE, and confidence interval coverage for the regression coefficients vary with p, and briefly comment on these relationships. For MSE and coverage, leave the intercept out of the calculation: the focus is on the coefficients to which a selection procedure has been applied.

1.3. *Ridge solution.* Show that $\widehat{\boldsymbol{\beta}} = (n^{-1}\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}n^{-1}\mathbf{X}^\top\mathbf{y}$ minimizes the objective function given in (1.17).

1.4. *Standardization and the intercept.* Suppose you have solved for the ridge regression solutions $\tilde{\boldsymbol{\beta}}$ on the standardized scale. The formula for recovering $\widehat{\beta}_1, \ldots, \widehat{\beta}_p$ on the original scale is given in the text. What is $\widehat{\beta}_0$ on the original scale?

1.5. *Uniqueness of ridge solution.* Prove Theorem 1.1: Show that for any design matrix $\mathbf{X}$, the quantity $n^{-1}\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}$ is invertible if $\lambda > 0$.

1.6. *Expected prediction error.* Show the identity for the mean squared prediction error (1.7).

1.7. *Bayesian and frequentist intervals for ridge regression.* Something along these lines? Orthonormal case? General case?

1.8. *Existence theorem.* This problem consists of proving Theorem 1.2 in two parts. Let $\mathbf{QDQ}^\top$ denote the spectral decomposition of $n^{-1}\mathbf{X}^\top\mathbf{X}$, where $\mathbf{D}$ is the diagonal matrix of eigenvalues $d_1, \ldots, d_p$.

(a) Show that

$$\lim_{\lambda \to 0^+} \frac{\partial}{\partial \lambda} \sum_j \mathrm{Var}(\widehat{\beta}_j) = -2\frac{\sigma^2}{n} \sum_{j:d_j>0} d_j^{-2}.$$

(b) Show that

$$\lim_{\lambda \to 0^+} \frac{\partial}{\partial \lambda} \mathrm{Bias}^2(\widehat{\boldsymbol{\beta}}) = 0;$$

you may wish to use the notation $\boldsymbol{\alpha} = \mathbf{Q}^\top\boldsymbol{\beta}$, as this quantity appears frequently in the derivation.

Thus, as we introduce a penalty to the OLS estimates ($\lambda = 0$), the MSE goes down, as it involves the sum of the terms in (a) and (b). Therefore, there exists a range of values for which the MSE of ridge regression is smaller than the MSE of OLS regression, regardless of $\mathbf{X}$.

1.9. *Pollution analysis without rescaling.* Re-analyze the pollution-mortality data in the original units (without rescaling). Show that the model is the same, in the sense that the predicted value of $\mathbf{y}$ is identical for a given value of λ, but that the actual $\widehat{\boldsymbol{\beta}}$ values are different.

1.10. *Leave-one-out cross-validation for linear fitting.* Let $\mathbf{Sy}$ and $\tilde{\mathbf{S}}\mathbf{y}$ denote the fitted values of $\mathbf{y}$ for the full model and the model where observation i has been left out, respectively. Suppose $\tilde{\mathbf{S}}$ has the property that $\tilde{s}_{ij} = s_{ij}(1 - s_{ii})^{-1}$ for $i \neq j$ and $\tilde{s}_{ii} = 0$ (many linear fitting

methods, including ridge regression, have this property). Show that for any linear fitting method with this property,

$$\sum_i \left\{ y_i - \hat{f}_{(-i)}(\mathbf{x}_i) \right\}^2 = \sum_i \left(\frac{y_i - \hat{f}(\mathbf{x}_i)}{1 - s_{ii}} \right)^2 .$$

1.11. *WHO pneumonia modeling.* The presentation of an acutely ill young infant presents health workers, especially those in developing countries, with a very difficult problem. Serious infections are the main cause of morbidity and mortality in infants under 3 months of age in these countries, and diagnosing the severity of the illness is rather difficult.

To study this problem, the World Health Organization (WHO) collected data on a number of readily accessible variables such as vital signs, family history, and clinical observations resulting from physical examination. The patients' disease status was later determined based on the course of the disease and various laboratory tests. The goal of the study was to develop an early prediction rule for grading the severity of the disease so that timely treatment could be delivered (and costly but unnecessary treatments avoided).

The WHO study looked at several acute respiratory illnesses in several countries; the data set `whoari` contains data on pneumonia from the country Ethiopia. The outcome, a pneumonia score abbreviated `pnsc`, was measured on the following scale:

1: No disease

2: Cold/cough

3: Pneumonia

4: Severe pneumonia

5: Life-threatening illness

The data, as well as descriptions of the variables, is available online. Fit a regular OLS model and a ridge regression model (with an appropriate choice of λ) to the data.

(a) Briefly, describe the variables that appear to be most important (in terms of affecting the pneumonia prediction). Do they make sense? Do OLS and ridge agree on which variables are most important?

(b) In your opinion, which variable is more important in predicting pneumonia, stridor or age? Why?

(c) With respect to statistical significance, are any variables considerably more significant in one analysis than the other?

(d) Comment on the significance of age in the two models. Why do you think age is more significant in the ridge regression analysis?

(e) Comment on the estimates you obtain from each model for the effect of sucking ability (`absu`) and drinking ability (`afe`); note that higher scores for these two variables mean more severe problems with sucking/feeding, not a greater ability to suck/feed. Which estimates do you consider to be more reasonable? Why?

2

The Lasso

2.1 ℓ_1-penalized regression

Section 1.6 introduced penalized regression, in which the usual least squares objective function was modified to include a penalty in order to stabilize the estimation and produce more reasonable solutions. In that section, the penalty took the form of a sum of squares of the regression coefficients, yielding an approach known as ridge regression. In this chapter, we continue to focus on the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{2.1}$$

where $\boldsymbol{\varepsilon} \sim \mathrm{N}(0, \sigma^2\mathbf{I}_n)$, but instead penalize the absolute values of the regression coefficients.

Consider the objective function

$$Q(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{2.2}$$

where $\|\boldsymbol{\beta}\|_1 = \sum_j |\beta_j|$ denotes the ℓ_1 norm of the regression coefficients. This is similar to ridge regression (1.17), but with an ℓ_1 norm $\|\boldsymbol{\beta}\|_1$ replacing the ℓ_2 norm $\|\boldsymbol{\beta}\|_2^2$ in the penalty term (Figure 2.1). Estimates of $\boldsymbol{\beta}$ are obtained by minimizing the above function for a given value of λ, yielding $\widehat{\boldsymbol{\beta}}(\lambda)$. In the context of regression analysis, this approach was originally proposed by Tibshirani, who called it the *least absolute shrinkage and selection operator*, or lasso. In the signal processing literature, the approach is known as *basis pursuit*.

Its name captures the essence of what the lasso penalty accomplishes: shrinkage and selection. Chapter 1 illustrated the way in which ridge regression produces estimates which are shrunken toward zero. This shrinkage property is shared by the lasso: both approaches penalize large values of the regression coefficients. In ridge regression, however, the estimates were *dense*: all predictors were present in the model with nonzero coefficient estimates. The lasso, on the other hand, produces *sparse* solutions: some coefficient estimates are exactly zero, effectively removing those predictors from the model.

Sparsity has two very attractive properties. First, as we will see in Section 2.4, algorithms which take advantage of sparsity can scale up very efficiently, offering considerable computational advantages in high dimensional regression. The second advantage is interpretability. For the example in Section 1.6.5, it was not difficult to display and consider all 15 predictors. This is no longer the case when the model contains hundreds or thousands of predictors. In these high-dimensional settings, sparsity offers a very helpful simplification of the model by allowing us to focus only on the predictors with nonzero coefficient estimates.

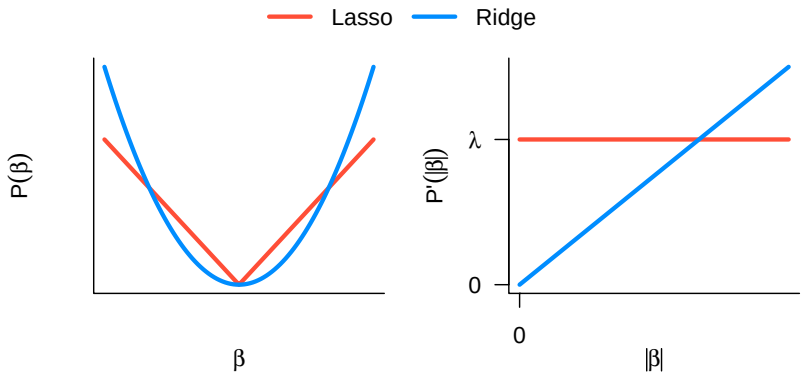

FIGURE 2.1
Penalty function (left) and its derivative (right) for the lasso and ridge penalties.

2.1.1 Karush-Kuhn-Tucker conditions for the lasso

How does such a seemingly simple change (ℓ_1 instead of ℓ_2 penalty) result in sparse solutions? We can shed some light on this question by considering the derivative of objective function (2.2). In classical statistical theory, the derivative of the log-likelihood function is called the *score function*, and maximum likelihood estimators are found by setting this derivative equal to zero, thus yielding the *likelihood equations* (or *score equations*):

$$0 = \frac{\partial}{\partial \theta} L(\theta), \tag{2.3}$$

where L, defined in (1.12), denotes the log-likelihood.

Extending this idea to penalized likelihoods involves taking the derivatives of objective functions like (1.14), yielding the *penalized score function*. For ridge regression, the penalized likelihood is everywhere differentiable, and the extension to penalized score equations is straightforward. For the lasso, and for the other penalties we will consider in this book, the penalized likelihood is not differentiable (specifically, not differentiable at zero). We can extend the idea of (2.3) to nondifferentiable functions with *subdifferentials*, which expand the notion of the derivative to include sets of tangent lines at the nondifferentiable points of a function; see Section 2.10 for a formal description of this concept. Letting $\partial Q(\theta)$ denote the subdifferential of the function Q, the *penalized likelihood equations* (or *penalized score equations*) are:

$$0 \in \partial Q(\theta). \tag{2.4}$$

In the optimization literature, these equations are known as the Karush-Kuhn-Tucker (KKT) conditions. For convex optimization problems such as the lasso, the KKT conditions are both necessary and sufficient to characterize the solution.

Evaluating $\partial Q(\boldsymbol{\beta})$ for the lasso, we find that $\widehat{\boldsymbol{\beta}}(\lambda)$ is a global minimizer of (2.2) if and only if it satisfies the KKT conditions

$$\begin{cases} \mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n = \lambda \text{sign}(\widehat{\beta}_j), & \widehat{\beta}_j \neq 0 \\ |\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n| \leq \lambda, & \widehat{\beta}_j = 0. \end{cases} \tag{2.5}$$

In other words, the correlation between a predictor and the residuals, $\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n$, must exceed a certain minimum threshold λ before it is included in the model. When this correlation is below λ, $\widehat{\beta}_j = 0$. It is instructive to compare these equations to Figure 2.1. There, we see that the rate of penalization $P'(|\beta|)$ is constant for the lasso, and in particular is λ at $\beta = 0$. For the ridge penalty, on the other hand, $P'(|\beta|)$ drops to zero as $\beta \to 0$: predictors enter the model no matter how small their correlation with the residual is. In addition, we can see from the plot that ridge regression will, compared with the lasso, shrink small regression coefficients less and large regression coefficients more.

The KKT conditions can also be expressed as a single equation:

$$-\frac{1}{n}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) + \lambda \hat{\mathbf{s}} = 0, \tag{2.6}$$

where $\hat{\mathbf{s}} = (\hat{s}_1, \ldots, \hat{s}_p)^T$, with $\hat{s}_j = 1$ if $\widehat{\beta}_j > 0$, $\hat{s}_j = -1$ if $\widehat{\beta}_j < 0$ and $\hat{s}_j \in [-1, 1]$ if $\widehat{\beta}_j = 0$. Here the term $\hat{\mathbf{s}}$ is the subdifferential of $\|\boldsymbol{\beta}\|_1$ evaluated at $\widehat{\boldsymbol{\beta}}$.

If we set

$$\lambda = \lambda_{\max} \equiv \max_{1 \le j \le p} |\mathbf{x}_j^T \mathbf{y}|/n, \tag{2.7}$$

then $\widehat{\boldsymbol{\beta}} = 0$ satisfies (2.5). That is, for this $\lambda_{\max}$, we have $\widehat{\boldsymbol{\beta}}(\lambda_{\max}) = 0$. Clearly, for any $\lambda > \lambda_{\max}$, we also have $\widehat{\boldsymbol{\beta}}(\lambda) = 0$. This means $\lambda_{\max}$ is the smallest value of λ that makes the solution zero. On the other hand, if we set $\lambda = 0$, then (2.5) becomes $\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = 0, j = 1, \ldots, p$. Putting these equations together in matrix notation yields the normal equation for least squares $\mathbf{X}^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) = 0$. As we have mentioned before, if $p > n$, there are infinitely many solutions to this equation. Thus, computing the solution path starts at $\lambda_{\max}$ and continues until $\lambda = 0$ if $\mathbf{X}$ is full rank. If $\mathbf{X}$ is not full rank, the lasso solution will fail to be unique for λ values below some point $\lambda_{\min}$; more details regarding this statement are given in Section 2.1.2.

2.1.2 *Uniqueness of lasso solution

The lasso criterion is convex, but not strictly convex if $\mathbf{X}^T\mathbf{X}$ does not have a full rank, which is always the case in $p \ge n$ settings. In such settings, the lasso solution will be unique for some, but not all, values of λ, as the following simple example illustrates.

Example 2.1. Suppose $n = 2$ and $p = 2$, and suppose the observations are $(y_1, x_{11}, x_{12}) = (1, 1, 1)$ and $(y_2, x_{21}, x_{22}) = (-1, -1, -1)$. The lasso criterion for these observations is

$$\frac{1}{2}(1 - \beta_1 - \beta_2)^2 + \lambda(|\beta_1| + |\beta_2|). \tag{2.8}$$

Then the solutions are

$$\begin{cases} (\widehat{\beta}_1, \widehat{\beta}_2) = (0, 0) \text{ if } \lambda \ge 1, \\ (\widehat{\beta}_1, \widehat{\beta}_2) \in \{\beta_1 + \beta_2 = 1 - \lambda, \beta_1 \ge 0, \beta_2 \ge 0\} \text{ if } 0 \le \lambda < 1. \end{cases}$$

The verification of this is left as Exercise 2.1. So in this example, for $\lambda \ge 1$, there is a single unique solution, while for $0 \le \lambda < 1$ there are infinitely many solutions, including two sparse solutions $(\widehat{\beta}_1, \widehat{\beta}_2) = (0, 1 - \lambda)$ or $(1 - \lambda, 0)$. □

When is a lasso solution unique? Let $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\beta}}^*$ be two solutions, and let $\mathbf{d} = \widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}^*$. Then

$$\mathbf{X}\mathbf{d} = \mathbf{0}. \tag{2.9}$$

Let $V(\widehat{\boldsymbol{\beta}})$ be the set of $\mathbf{v} \in \mathbb{R}^p$ of the form $v_j = 1$ if $\widehat{\beta}_j > 0$, $v_j = -1$ if $\widehat{\beta}_j < 0$ and v_j equals either -1 or 1 if $\widehat{\beta}_j = 0$. Suppose $\widehat{\boldsymbol{\beta}}$ has m zero entries, then $V(\widehat{\boldsymbol{\beta}})$ has $k = 2^m$ such vectors, so we can write $V(\widehat{\boldsymbol{\beta}}) = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$. The difference $\mathbf{d}$ must satisfy

$$\mathbf{d}^T\mathbf{v} \leq 0 \text{ for all } \mathbf{v} \in V(\widehat{\boldsymbol{\beta}}). \tag{2.10}$$

If this is not true, that is, $\mathbf{d}^T\mathbf{v} > 0$ for some $\mathbf{v} \in V(\widehat{\boldsymbol{\beta}})$, then for any $0 < \rho \leq 1$, $\|\widehat{\boldsymbol{\beta}} + \rho\mathbf{d}\|_1 \geq (\widehat{\boldsymbol{\beta}} + \rho\mathbf{d})^T\mathbf{v} = \|\widehat{\boldsymbol{\beta}}\| + \rho\mathbf{d}^T\mathbf{v} > \|\widehat{\boldsymbol{\beta}}\|$. But this is a contradiction since $\|\widehat{\boldsymbol{\beta}} + \rho\mathbf{d}\|_1 = \|\widehat{\boldsymbol{\beta}}\|_1$ for all $0 \leq \rho \leq 1$.

So (2.9) and (2.10) imply that $\widehat{\boldsymbol{\beta}}$ is unique if and only if there does not exist any non-zero vector $\mathbf{b} \in \mathbb{R}^p$ satisfying both $\mathbf{X}\mathbf{b} = \mathbf{0}$ and $\mathbf{b}^T\mathbf{v} \leq 0$ for every $\mathbf{v} \in V(\widehat{\boldsymbol{\beta}})$. This can be stated as $C(\widehat{\boldsymbol{\beta}}) \cap \mathcal{N}(\mathbf{X}) = \{\mathbf{0}\}$, where $C(\widehat{\boldsymbol{\beta}}) = \{\mathbf{b} : \mathbf{b}^T\mathbf{v} \leq 0, \text{for every } \mathbf{v} \in V(\widehat{\boldsymbol{\beta}}\}$.

A simpler way to characterize the uniqueness of $\widehat{\boldsymbol{\beta}}$ is as follows. Note that (2.9) implies $\mathbf{X}^T\mathbf{r}_1 = \mathbf{X}^T\mathbf{r}_2$. Let $\widehat{\beta} = \{k_1, \ldots, k_p\}$ be the set of indices for which $|(\mathbf{X}^T\mathbf{r})_{k_j}| = \|\mathbf{X}^T\mathbf{r}\|_\infty$ for $j = 1, \ldots, p$. If $\widehat{\boldsymbol{\beta}}$ is a solution then $\widehat{\beta}_j = 0$ for $j \notin \nu$, since every solution must satisfy (2.5). Therefore, if $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\beta}}^*$ are solutions, $\mathbf{d}_j = 0$ for all $j \notin \widehat{\beta}$.

Let $\mathbf{V}$ be the $k \times p$ matrix whose ith row is $\mathbf{v}_i^T$ in $V(\widehat{\boldsymbol{\beta}})$, let $\mathbf{X}_{\widehat{\beta}}$ be the $n \times p$ matrix whose jth column is the k_jth column of $\mathbf{X}, j = 1, \ldots, p$, and let $\mathbf{V}_{\widehat{\beta}}$ be the corresponding submatrix of $\mathbf{V}$. The above discussion shows that $\widehat{\boldsymbol{\beta}}$ is a unique solution if and only if there does not exist any $\mathbf{b} \neq \mathbf{0}$ satisfying

$$\mathbf{V}_{\widehat{\beta}}\mathbf{b} \leq 0 \text{ and } \mathbf{X}_{\widehat{\beta}}\mathbf{b} = \mathbf{0}.$$

Is a lasso solution *always* sparse? In general, the answer is no. For instance, in Example 2.1, there are infinitely many non-sparse solutions. But there exists an upper bound on the number of nonzero coefficients of a particular kind of solutions – a regular solution. A lasso solution is said to be regular if $\mathcal{N}(\mathbf{V}) \cap \mathcal{N}(\mathbf{X}) = \{\mathbf{0}\}$. If $p \geq n$ and $\widehat{\boldsymbol{\beta}}$ is a regular solution, then $\widehat{\boldsymbol{\beta}}$ has at most $n - 1$ nonzero coefficients.

2.2 Soft thresholding

Consider the case where the design matrix $\mathbf{X}$ is orthonormal: $\mathbf{X}^T\mathbf{X}/n = \mathbf{I}$. In this case, the KKT conditions are separable, and we can estimate β_j without having to consider the other covariates. Although this is clearly

a special case and rarely met in high-dimensional regression problems, it presents the distinct advantage of offering a closed-form solution. This solution will not only offer insights into how the lasso works, but we will also use this closed-form solution as the building block of the coordinate descent algorithm presented in Section 2.4.

The OLS estimator in the orthonormal case is given by $\widehat{\beta}_{\text{OLS}} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} = \mathbf{x}^T\mathbf{y}/n$, since $\mathbf{x}^T\mathbf{x} = n$; in this section we will drop the j subscript on β and $\mathbf{x}$ for the sake of simplicity. The lasso estimate is

$$\widehat{\beta}(\lambda) = \arg\min_{\beta\in\mathbb{R}} \frac{1}{2n}\|\mathbf{y} - \mathbf{x}\beta\|_2^2 + \lambda|\beta|. \tag{2.11}$$

Since $\|\mathbf{y} - \mathbf{x}\beta\|^2/n = (\widehat{\beta}_{\text{OLS}} - \beta)^2 + \mathbf{y}^T\mathbf{y}/n - \widehat{\beta}_{\text{OLS}}^2$, we have

$$\widehat{\beta}(\lambda) = \arg\min_{\beta\in\mathbb{R}} \frac{1}{2}(\widehat{\beta}_{\text{OLS}} - \beta)^2 + \lambda|\beta|. \tag{2.12}$$

Some calculation (Exercise 2.5) shows that

$$\widehat{\beta}(\lambda) = \begin{cases} \widehat{\beta}_{\text{OLS}} - \lambda, & \text{if } \widehat{\beta}_{OLS} > \lambda, \\ 0, & \text{if } |\widehat{\beta}_{\text{OLS}}| \le \lambda, \\ \widehat{\beta}_{\text{OLS}} + \lambda, & \text{if } \widehat{\beta}_{OLS} < -\lambda. \end{cases} \tag{2.13}$$

This can be written more compactly as

$$\widehat{\beta}(\lambda) = S(\widehat{\beta}_{\text{OLS}}|\lambda),$$

where

$$S(z|\lambda) = \text{sign}(z)(|z| - \lambda)_+, \tag{2.14}$$

where $x_+ = x$ for $x > 0$ and 0 otherwise. The $S(\cdot|\lambda)$ function is called the soft thresholding operator. This was originally proposed by Donoho and Johnstone (1994) for soft thresholding of wavelets coefficients in the context of nonparametric regression. By comparison, the "hard" thresholding operator is $H(z, \lambda) = zI\{|z| > \lambda\}$, where $I(S)$ is the indicator function for set S.

The hard and soft thresholding operators are plotted in Figure 2.2. As the figure shows, the hard thresholding operator is discontinuous, with $\widehat{\beta}(\lambda)$ jumping from 0 to $\widehat{\beta}_{OLS}$ as soon as $\widehat{\beta}_{OLS}$ cross the threshold. The soft thresholding operator, on the other hand, is a continuous function of both λ and z; hence their names.

From expression (2.13), we can see that the lasso has a positive probability of yielding an estimate of exactly 0 – in other words, of producing

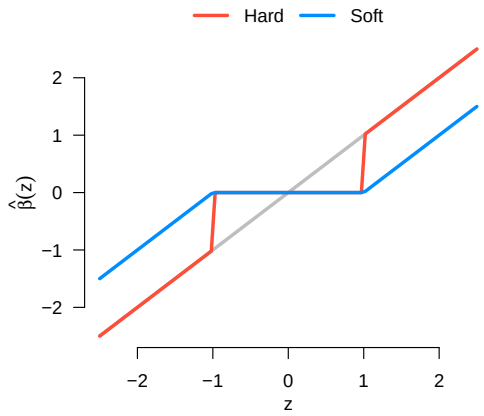

FIGURE 2.2
The soft and hard thresholding operators.

a sparse solution. Specifically, the probability of dropping $\mathbf{x}$ from the model is $\mathbf{P}(|\widehat{\beta}_{\text{OLS}}| \leq \lambda)$. Since the error terms satisfy $\epsilon_i \overset{\perp\!\!\!\perp}{\sim} \text{N}(0, \sigma^2)$, we have $\widehat{\beta}_{\text{OLS}} \sim \text{N}(\beta, \sigma^2/n)$. Thus,

$$\mathbf{P}(\widehat{\beta}(\lambda) = 0) = \Phi\Big(\frac{\lambda - \beta}{\sigma/\sqrt{n}}\Big) - \Phi\Big(\frac{-\lambda - \beta}{\sigma/\sqrt{n}}\Big),$$

where Φ is the cumulative distribution function for a standard normal random variable.

The sampling distribution of $\widehat{\beta}$ is plotted in Figure 2.3 for two cases, $\beta = 0$ and $\beta = 1$. There are several interesting observations to be drawn from the figure. The most notable is the fact that this sampling distribution is not regular: the distribution is mixed, with some portion continuously distributed and the rest concentrated at a point mass at zero. Furthermore, the continuous portion of the distribution is not normally distributed, although the distribution in the $\beta = 1$ is approximately normal. In the $\beta = 0$ case, there is a high probability of $\widehat{\beta} = 0$ and small probability that $\widehat{\beta}$ will be take on a value near zero. In the $\beta = 1$ case, there is a high probability that $\widehat{\beta}$ will be positive, and only a small probability that $\widehat{\beta} = 0$. It is worth noting, however, that the positive density of $\widehat{\beta}$ is not centered at $\beta = 1$, but instead at $\beta - \lambda = 0.5$ due to shrinkage. In all of these ways, the distribution of $\widehat{\beta}$ is substantially different from the distribution of $\widehat{\beta}_{OLS}$, which creates challenges for carrying out inference using the lasso; we will return to the problem of inference in Chapters 5-8.

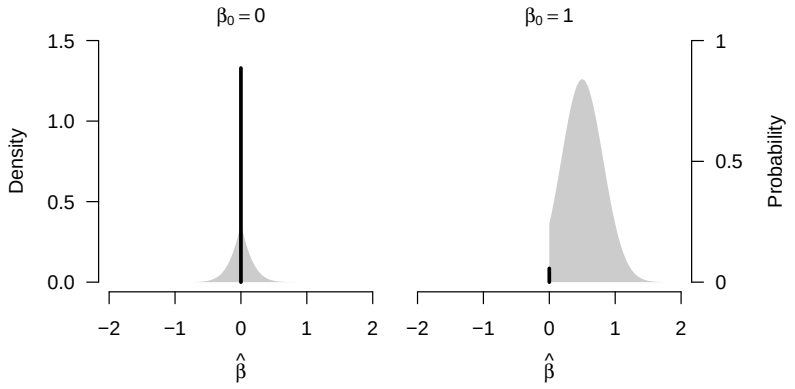

FIGURE 2.3
Sampling distribution of the lasso estimator $\widehat{\beta}$ in the orthonormal case. Here, $\sigma = 1$, $n = 10$, and $\lambda = 1/2$.

2.3 Lasso vs. forward selection

The lasso can be thought of as performing a multivariate version of soft thresholding. In the same sense, the multivariate version of hard thresholding is ℓ_0 penalization:

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_0, \tag{2.15}$$

where $\|\boldsymbol{\beta}\|_0 = \sum_j I(\beta_j \neq 0)$. For the orthonormal case $\mathbf{X}^\top \mathbf{X}/n = \mathbf{I}$, the solution to (2.15) is given by $\widehat{\beta}_j = H(\widehat{\beta}_j^{OLS}, \sqrt{2\lambda})$. Estimating $\boldsymbol{\beta}$ in this manner is equivalent to the subset selection of Section 1.4; many important model selection criteria, including AIC and BIC, can be considered special cases of this formulation taking different λ values.

Thus, the lasso can be thought of as a "soft" relaxation of ℓ_0 penalized regression. This relaxation has two important benefits. First, estimates are continuous with respect to both λ and the data. Second, the lasso objective function is convex, which offers considerable advantages in terms of optimization. Namely, one can solve for lasso estimates using gradient-based methods without having to be concerned with convergence to local minima. In contrast, the computation in ℓ_0 penalized problems is combinatorial and not feasible when p is large. In fact, solving (2.15) is an NP-hard computational problem (Natarajan, 1995). To get around this

difficulty, a common approach to solving (2.15) is to employ the greedy algorithm known as forward selection that we considered in Section 1.4. Like forward selection, the lasso will allow more variables to enter the model as λ is lowered, but as we will see, performs a continuous version of variable selection and is less greedy about allowing selected variables into the model.

Let us consider the regression paths of the lasso and forward selection (ℓ_1 and ℓ_0 penalized regression, respectively) as we lower λ, starting at $\lambda_{\max}$ where $\widehat{\boldsymbol{\beta}} = \mathbf{0}$. As λ is lowered below $\lambda_{\max}$, both approaches find the predictor most highly correlated with the response (let $\mathbf{x}_j$ denote this predictor), and set $\widehat{\beta}_j \neq 0$. With forward selection, the estimate jumps from $\widehat{\beta}_j = 0$ all the way to $\widehat{\beta}_j = \mathbf{x}_j^T\mathbf{y}/n$. The lasso solution $\widehat{\beta}_j = 0$ heads in this direction as well, but proceeds more cautiously, gradually advancing towards $\widehat{\beta}_j = \mathbf{x}_j^T\mathbf{y}/n$ as we lower λ.

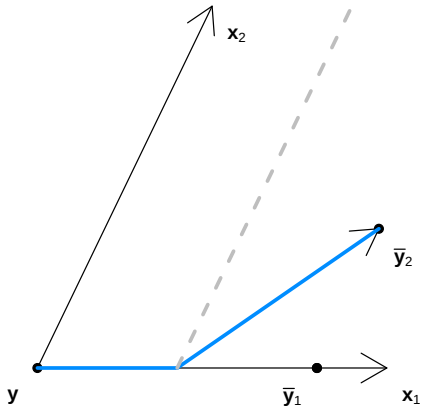

FIGURE 2.4
Geometry of the lasso path through the $\mathbf{x}_1$, $\mathbf{x}_2$ plane. Here, $\bar{\mathbf{y}}_1$ is the projection of $\mathbf{y}$ onto $\mathbf{x}_1$ and $\bar{\mathbf{y}}_2$ is the projection of $\mathbf{y}$ onto the column space of $(\mathbf{x}_1, \mathbf{x}_2)$. The lasso path is shown in blue.

The lasso solution proceeds in this manner until it reaches the point that a new predictor, $\mathbf{x}_k$, is equally correlated with the residual $\mathbf{r}(\lambda) = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}(\lambda)$. From this point, the lasso solution will contain both $\mathbf{x}_1$ and $\mathbf{x}_2$, and proceed in the direction that is equiangular between the two predictors. The geometry of the lasso path is depicted in Figure 2.4. In the figure, forward selection takes large, discontinuous jumps from $\bar{\mathbf{y}}_1$ to $\bar{\mathbf{y}}_2$, while the lasso takes a smooth, continuous path towards $\bar{\mathbf{y}}_2$. Furthermore, the lasso always proceeds in a direction such that every active

predictor (i.e., one with $\widehat{\beta}_j \neq 0$) is equally correlated with the residual $\mathbf{r}(\lambda)$. This property, that all active predictors are equally correlated with the residual, can also been seen from the KKT conditions (2.5).

The geometry of the lasso depicted in Figure 2.4 clearly illustrates the "greediness" of forward selection. By continuing along the path from $\mathbf{y}$ to $\bar{\mathbf{y}}_1$ past the point of equal correlation, forward selection continues to exclude $\mathbf{x}_2$ from the model even when $\mathbf{x}_2$ is more closely correlated with the residuals than $\mathbf{x}_1$. The lasso, meanwhile, allows the predictors most highly correlated with the residuals into the model, but only gradually, up to the point that the next predictor is equally useful in explaining the outcome.

This geometric approach to the lasso not only lends insight to the method and its relationship to forward selection, it can also be used as an algorithm. The approach, known as *least angle regression*, or the LARS algorithm, offers an elegant way to solve for $\widehat{\boldsymbol{\beta}}$ in lasso estimation. In the following section, we discuss a less beautiful but simpler and more flexible alternative approach known as coordinate descent algorithms for fitting lasso models.

2.4 The coordinate descent algorithm

A simple and effective algorithm for computing the lasso solutions is the coordinate descent algorithm. This algorithm optimizes a target function with respect to a single parameter at a time, iteratively cycling through all parameters until convergence is reached. It is particularly suitable for problems that have a simple closed form solution in a single dimension but lack one in higher dimensions.

The idea behind coordinate descent is to minimize Q with respect to β_j, while temporarily treating the other regression coefficients $\boldsymbol{\beta}_{-j}$ as fixed. Rewriting the objective function (2.2), we have

$$Q(\beta_j|\boldsymbol{\beta}_{-j}) = \frac{1}{2n}\sum_{i=1}^{n}(y_i - \sum_{k\neq j} x_{ik}\beta_k - x_{ij}\beta_j)^2 + \lambda|\beta_j| + \text{Constant}.$$

Let

$$\begin{aligned}\tilde{y}_{ij} &= \sum_{k \neq j} x_{ik} \widetilde{\beta}_k, \\ \tilde{r}_{ij} &= y_i - \tilde{y}_{ij}, \text{ and} \\ \tilde{z}_j &= n^{-1} \sum_{i=1}^{n} x_{ij} \tilde{r}_{ij},\end{aligned}$$

where $\{\tilde{r}_{ij}\}_{i=1}^n$ are the partial residuals with respect to the j^{th} predictor, and $\tilde{z}_j$ is the OLS estimator based on $\{\tilde{r}_{ij}, x_{ij}\}_{i=1}^n$. Some algebra shows that

$$Q(\beta_j | \widetilde{\boldsymbol{\beta}}_{-j}) = \frac{1}{2}(\beta_j - \tilde{z}_j)^2 + \lambda |\beta_j| + \text{Constant},$$

just as we obtained in (2.12). Thus, letting $\widetilde{\beta}_j$ denote the minimizer of $Q(\beta_j | \widetilde{\boldsymbol{\beta}}_{-j})$, (2.13) and (2.14) imply that

$$\widetilde{\beta}_j = S(\tilde{z}_j | \lambda). \tag{2.16}$$

Given the current value $\widetilde{\beta}^{(s)}$ in the sth iteration for $s = 0, 1 \ldots$, the algorithm for computing $\widehat{\boldsymbol{\beta}}$ is given in Algorithm 2.1. In the algorithm,

Algorithm 2.1 Coordinate descent algorithm for the lasso

repeat
  for $j = 1, 2, \ldots, p$
    $\tilde{z}_j = n^{-1} \sum_{i=1}^n x_{ij} r_i + \widetilde{\beta}_j^{(s)}$
    $\widetilde{\beta}_j^{(s+1)} \leftarrow S(\tilde{z}_j | \lambda)$
    $r_i \leftarrow r_i - (\widetilde{\beta}_j^{(s+1)} - \widetilde{\beta}_j^{(s)}) x_{ij}$ for all i.
until convergence

$\tilde{z}_j$ represents the unpenalized solution for $\widetilde{\beta}_j$ given $\widetilde{\beta}_{-j}$, and can be expressed in several equivalent ways:

$$\begin{aligned}\tilde{z}_j &= n^{-1} \sum_{i=1}^{n} x_{ij} \tilde{r}_{ij} \\ &= n^{-1} \sum_{i=1}^{n} x_{ij} \left(y_i - \tilde{y}_i + x_{ij} \widetilde{\beta}_j^{(s)} \right) \\ &= n^{-1} \sum_{i=1}^{n} x_{ij} r_i + \widetilde{\beta}_j^{(s)},\end{aligned}$$

where $\tilde{y}_i = \sum_{j=1}^{p} x_{ij}\widetilde{\beta}_j^{(s)}$ is the current fitted value for observation i and $r_i = y_i - \tilde{y}_i$ is the current residual. The last expression for $\tilde{z}_j$ is the most efficient computationally, as it allows one to calculate $\tilde{z}_j$ without calculating the $\{\tilde{r}_{ij}\}_{i=1}^{n}$ values. The last step in Algorithm 2.1 ensures that $\{r_i\}$ always hold the current values of the residuals, which is essential as $\{r_i\}$ will be used again in the first step of the next iteration.

The coordinate descent algorithm has the potential to be quite efficient, in that the three steps in Algorithm 2.1 require only $O(2n)$ operations, meaning that one full iteration can be completed at a computational cost of $O(2np)$ operations – linear in both n and p.

Numerical analysis of optimization problems of the form (1.14) has shown that coordinate descent algorithms converge to a stationary point provided that the loss function $L(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y})$ is differentiable and the penalty function $P_\lambda(\boldsymbol{\beta})$ is *separable*, meaning that it can be written as $P_\lambda(\boldsymbol{\beta}) = \sum_j P_\lambda(\beta_j)$. Lasso-penalized linear regression satisfies both of these criteria. Furthermore, because the lasso objective is a convex function, the sequence of the objective functions $\{Q(\widetilde{\boldsymbol{\beta}}^{(s)})\}$ converges to the global minimum. However, because the lasso objective is not strictly convex, there may be multiple solutions (Section 2.1.2). In such situations, there is no guarantee that the coordinate descent algorithm will converge to a regular solution.

2.4.1 Pathwise optimization

As we saw in Section 1.6, we are typically interested in determining $\widehat{\boldsymbol{\beta}}$ for a range of values of λ, thereby obtaining the coefficient path. In applying the coordinate descent algorithm to determine the lasso path, an efficient strategy is to compute solutions for decreasing values of λ, starting at $\lambda_{\max} = \max_{1 \le j \le p} |\mathbf{x}_j^T \mathbf{y}|/n$, the point at which all coefficients are 0. By continuing along a decreasing grid of λ values, we can use the solutions $\widehat{\boldsymbol{\beta}}(\lambda_k)$ as initial values when solving for $\widehat{\boldsymbol{\beta}}(\lambda_{k+1})$. Because the coefficient path is continuous, doing this automatically provides good initial values for the iterative optimization procedure. This strategy, known as employing "warm starts," substantially improves the efficiency of the algorithm, as the initial values are always fairly close to the final solution.

We proceed in this manner down to a minimum value $\lambda_{\min}$. If $p < n$ and the design matrix is full rank, $\lambda_{\min}$ can be 0. In other settings, the model may become excessively large or cease to be identifiable for small λ; in such cases, a value such as $\lambda_{\min} = 0.01\lambda_{\max}$ may be used. Because lasso solutions change more rapidly at low values of λ, the grid of λ values is typically chosen to be uniformly spaced on the log scale over the interval $[\lambda_{\max}, \lambda_{\min}]$.

Choosing $\lambda_{\min}$ is often a practical consideration in high-dimensional problems; because coordinate descent algorithms take advantage of sparsity, the vast majority of the computation time occurs when λ is small. This can be very inefficient if the interesting portion of the solution path lies between $\lambda_{\max}$ and $0.4\lambda_{\max}$, which it often does when $p \gg n$.

To illustrate the coefficient path of the lasso, we fit a lasso model to the pollution data from Section 1.6.5. The coordinate descent algorithm described in this section is implemented in the `R` package `glmnet`. The basic usage of `glmnet` is straightforward:

```
library(glmnet)
fit <- glmnet(X, y)
plot(fit)
```

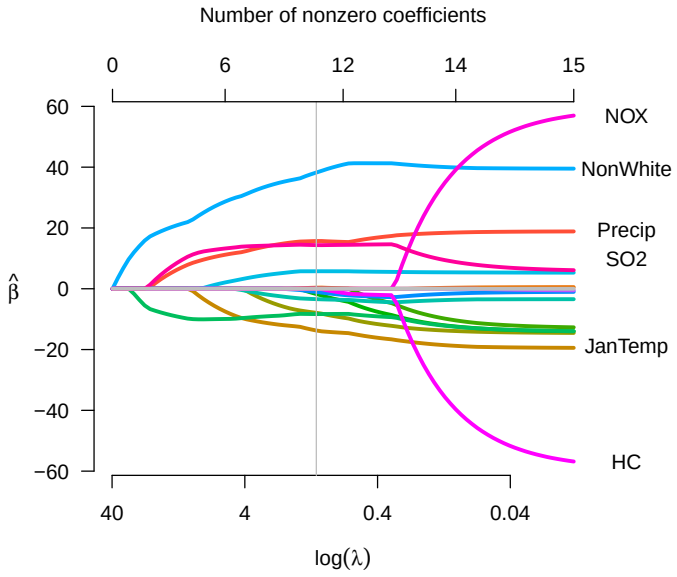

FIGURE 2.5
Coefficient path for the lasso as a function of λ, which is presented on a log scale. The SO_2 path is the one that is similar to the "Precip" path until $\lambda \approx 0.4$, then decreases back towards $\widehat{\beta} \approx 5$. A vertical line is drawn at the λ value that minimizes the cross-validation error.

The above code will solve for and then plot the solution path of the lasso. The (slightly reformatted) plot is presented in Figure 2.5. It

is highly instructive to compare this plot with the one in Figure 1.4. There are many similarities: in both plots, the estimates are $\widehat{\boldsymbol{\beta}} = \mathbf{0}$ on the left side and $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}_{OLS}$ on the right side; both display a rather striking pattern for the pollutants HC and NOX as λ changes; and both involve the coefficient for SO_2 increasing, then decreasing as λ decreases. However, there are many important differences as well. Most notably, the lasso solution path contains many exact zeros, with coefficients entering the model one by one as λ decreases. For example, at $\lambda = 1.84$, the value which minimizes the cross-validation error (Section 2.5.2), there are nine variables in the model. Notably, this does not include HC or NOX, the variables with the largest OLS regression coefficients. As with ridge regression, by incorporating an assumption that regression coefficients are likely to be small, the lasso avoids the questionable conclusion that increasing the amount of hydrocarbon pollution should save dozens of lives per year.

Another, more subtle difference between the lasso and ridge coefficient paths is that with the lasso, coefficients get larger faster than with ridge regression. For example, at $\lambda = 1.84$, $\widehat{\beta}_{\texttt{NonWhite}} = 35.6$ despite the fact that many other coefficients are either zero or very close to zero. This is a consequence of the fact that ridge regression applies heavier shrinkage than does the lasso.

2.5 Selection of λ

2.5.1 Information criteria

The use of information criteria for selecting λ was discussed in Section 1.5.2. The fitted values for models considered there were linear functions of the outcome variable: $\hat{\boldsymbol{\mu}} = \mathbf{S}\mathbf{y}$. For such models the degrees of freedom for the fit was shown to be $\text{tr}(\mathbf{S})$.

The fitted values for the lasso, however, are not linear functions of $\mathbf{y}$ and there is no exact, closed form solution to $\text{Cov}(\mathbf{y}, \hat{\boldsymbol{\mu}})$. A natural proposal would be to use $\text{df}(\lambda) = \|\widehat{\boldsymbol{\beta}}(\lambda)\|_0$, the number of nonzero coefficients. On the one hand, the nonzero coefficients were not prespecified – rather, they were selected from a larger pool of p coefficients because they exhibited the best correlation with the outcome. From this perspective, $\|\widehat{\boldsymbol{\beta}}(\lambda)\|_0$ would seem to underestimate the true degrees of freedom. On the other hand, shrinkage reduces the degrees of freedom in an estimator; for example, with ridge regression there are always p nonzero parameters, but there can be far less than p degrees of freedom, depend-

ing on λ. From this perspective, $\|\widehat{\boldsymbol{\beta}}(\lambda)\|_0$ might seem to overestimate the true degrees of freedom.

Surprisingly, as it turns out, these two factors exactly cancel and

$$\mathrm{df}(\lambda) = \|\widehat{\boldsymbol{\beta}}(\lambda)\|_0$$

can be shown to be an unbiased estimate of the lasso degrees of freedom. The derivation is this result is discussed in Exercise 2.9). With this estimate, we can use the information criteria presented in Section 1.5.2 for the purposes of selecting λ. For example,

$$\mathrm{BIC} = 2L(\widehat{\boldsymbol{\beta}}(\lambda)|\mathbf{X}, \mathbf{y}) + \mathrm{df}(\lambda)\log(n).$$

To illustrate the application of AIC and BIC to lasso models, we use the `ncvreg` package to fit the lasso path. The primary purpose of the `ncvreg` package is to provide penalties other than the lasso, and will be covered in more detail in Chapter 3. However, it also supplies a `penalty="lasso"` option and has some additional features that `glmnet` does not, such as providing a `logLik` method so that it can be used with `R`'s `AIC` and `BIC` functions. The syntax of `ncvreg` is very similar to its `glmnet` equivalent:

```
fit <- ncvreg(X, y, penalty="lasso")
ll <- log(fit$lambda)
IC <- cbind(AIC(fit), BIC(fit))
matplot(ll, IC, xlim=rev(range(ll)))
abline(v=ll[apply(IC, 2, which.min)])
```

A plot of AIC and BIC as a function of λ is shown in Figure 2.7 for the pollution data analyzed in Section 2.4.1. Here, $\hat{\lambda}_{\mathrm{AIC}} = 2.27$, while $\hat{\lambda}_{\mathrm{BIC}} = 4.90$. BIC applies a stronger penalty for overfitting, and as we would expect, chooses a smaller, more parsimonious model than does AIC.

The main advantage of AIC and BIC is that they are computationally convenient: they can be calculated using the fit of lasso model at very little computational cost. The primary disadvantage is that both AIC and BIC rely on a number of asymptotic approximations that can be quite inaccurate for high-dimensional data, especially when models are nearly saturated. Cross-validation, which we describe in the next section, is more reliable in general, although it comes at an added computation cost. In the low-dimensional pollution data example, both AIC and BIC give reasonable results and more or less agree with cross-validation, but as we will see in Section 2.7, this is not always the case.

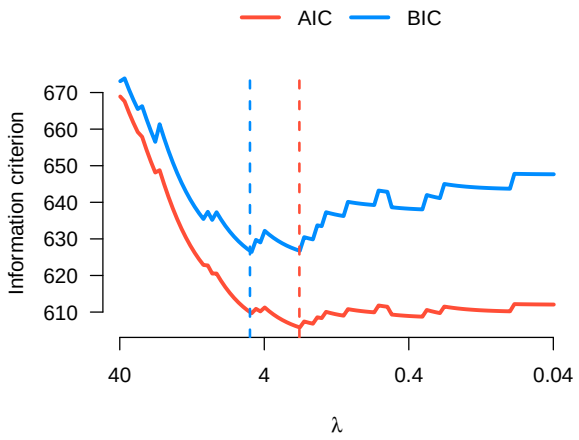

FIGURE 2.6
AIC, BIC for the lasso model fit to the pollution data. Dotted vertical lines are drawn at the values that minimize each information criterion.

2.5.2 Cross-validation

As discussed in Section 1.6.4, a reasonable approach to selecting λ in an objective manner is to choose the value of λ that yields a model with the greatest predictive power. The extent to which information criteria such as those presented in 2.5.1 accurately answer this question in high-dimensional settings is not always clear. For this reason, a more direct, empirical measurement of predictive accuracy is often preferable.

One idea is to split the data set into two fractions, a training set and test set, using one portion to estimate $\widehat{\boldsymbol{\beta}}$ (i.e., "train" the model) and the other to evaluate how well $\mathbf{X}\widehat{\boldsymbol{\beta}}$ predicts the observations in the second portion (i.e., "test" the model). The problem with this solution is that we rarely have so much data that we can freely part with half of it solely for the purpose of choosing λ. To finesse this problem, *cross-validation* splits the data into V folds, fits the data on $V-1$ of the folds, and evaluates prediction error on the fold that was left out. Specifically, the procedure works as follows for the lasso (or any other penalized regression method).

1. Specify a grid of regularization values $\Lambda = \{\lambda_1, \ldots, \lambda_V\}$.
2. Divide the data into V roughly equal parts $D_1, \ldots, D_V$. Common

choices for V are 5, 10, or n (also known as leave-one-out cross-validation).

3. For each $v = 1, \ldots, V$, compute the lasso solution path using the observations in $\{D_u, u \neq v\}$. Denote the resulting solutions $\widehat{\boldsymbol{\beta}}_{-v}(\lambda_k)$.

4. For each $\lambda \in \Lambda$, compute the mean squared prediction error

$$\mathrm{MSPE}_v(\lambda) = \frac{1}{n_v} \sum_{i \in D_v} (y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}_{-v}(\lambda))^2,$$

where n_v is the number of observations in D_v, as well as the overall cross-validation error

$$\mathrm{CV}(\lambda) = \frac{1}{V} \sum_{v=1}^{V} \mathrm{MSPE}_v(\lambda). \tag{2.17}$$

5. Choose $\hat{\lambda} = \arg\min_{\lambda \in \Lambda} \mathrm{CV}(\lambda)$.

Then $\widehat{\boldsymbol{\beta}} \equiv \widehat{\boldsymbol{\beta}}(\hat{\lambda})$ is taken as the estimate of the regression coefficients.

In the above procedure, $\mathrm{MSPE}_v(\lambda)$ is the mean squared prediction error for the model based on the training data $\{D_u, u \neq v\}$ in predicting the response variables in D_v, while $\mathrm{CV}(\lambda)$ is an estimate of the expected mean squared prediction error defined in (1.7).

Regardless of the number of cross-validation folds, each observation in the data appears exactly once in a test set. Let $\hat{\mu}_i(\lambda) = \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}_{u(i)}(\lambda)$ denote the predicted value of y_i based on the data set D_u not containing observation i (i.e., $\hat{\mu}_i(\lambda)$ is an out-of-sample prediction for y_i). The mean of $\{y_i - \hat{\mu}_i(\lambda)\}_{i=1}^n$ is equal to $\mathrm{CV}(\lambda)$. Its variability, however, is useful for estimating the accuracy with which $\mathbf{E}(\mathrm{MSPE}(\lambda))$ is estimated. Let $\mathrm{SD}_{\mathrm{CV}}(\lambda)$ denote the sample standard deviation of the $\{y_i - \hat{\mu}_i(\lambda)\}_{i=1}^n$ values. The standard error of the estimate (2.17) is therefore

$$\mathrm{SE}_{\mathrm{CV}}(\lambda) = \frac{\mathrm{SD}_{\mathrm{CV}}(\lambda)}{\sqrt{n}}.$$

The standard error, in turn, can be used to construct approximate confidence intervals for $\mathrm{CV}(\lambda)$:

$$[\mathrm{CV}(\lambda) - \mathrm{SE}_{\mathrm{CV}}(\lambda), \mathrm{CV}(\lambda) + \mathrm{SE}_{\mathrm{CV}}(\lambda)]. \tag{2.18}$$

Using the normal distribution to approximate the sampling distribution, this procedure will produce a 68% confidence interal for $\mathrm{CV}(\lambda)$. A more traditional 95% confidence interval could of course be created

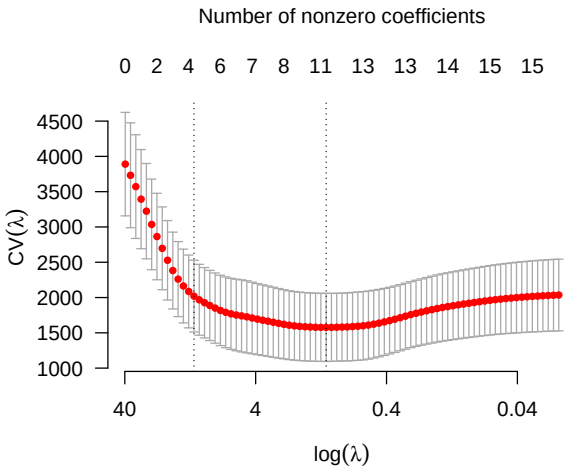

FIGURE 2.7
CV plot for lasso.

by adding/subtracting 1.96 standard errors instead, although there is no clear rationale for any particular confidence level in the context of selecting a regularization parameter.

A plot of the cross-validation error $\text{CV}(\lambda)$ as well as the confidence interval (2.18) is given in Figure 2.7 for the pollution data analyzed in Section 2.4.1. The cross-validation procedure described in this section, along with the estimates of $\text{CV}(\lambda)$ and its standard error, are implemented in `glmnet` and can be carried out using the following code, which produces Figure 2.7:

```
cvfit <- cv.glmnet(X, y)
plot(cvfit)
```

By default, `cv.glmnet` uses $V = 10$ folds, but this can be changed through the `nfolds` option.

For every $\lambda \in \Lambda$, the red dot represents $\text{CV}(\lambda)$, while the gray error bars depict interval (2.18). As mentioned in Section 2.4.1, the value $\lambda = 1.84$ minimizes the cross-validation error. However, as the confidence intervals show, there is substantial uncertainty about this minimum value. A fairly wide range of λ values ($\lambda \in [0.12, 9.83]$) yield $\text{CV}(\lambda)$ estimates falling within $\pm 1\text{SE}_{\text{CV}}$ of the minimum. As an alternative to selecting the value of λ that minimizes $\text{CV}(\lambda)$, some authors have suggested selecting the largest value of λ that falls within this interval (i.e.,

$\hat{\lambda} = 9.83$ in this example). The idea here is to err on the side of being more conservative, selecting the smallest/most parsimonious model whose prediction performance is not significantly worse than that of the apparent best model. The dashed vertical lines in Figure 2.7 illustrate these two choices: $\hat{\lambda} = 1.84$, minimizing $\text{CV}(\lambda)$, and $\hat{\lambda} = 9.83$, the more parsimonious choice (5 nonzero coefficients, as opposed to 9).

The general contours of Figure 2.7 are similar to those of Figure 1.5: very large values of λ shrink all the coefficients to zero and result in underfitting. Small values (under $\lambda = 1$), however, are also not ideal – as we decrease λ below 1, the model is overfit and prediction performance again suffers.

2.6 Estimation of σ^2

2.6.1 Plug-in and cross-validation estimators

In ordinary least squares regression, the standard estimator of σ^2 is

$$\hat{\sigma}^2_{\text{OLS}} = \frac{\text{RSS}}{n - \text{df}}, \tag{2.19}$$

where $\text{RSS} = \sum_{i=1}^n (y_i - \hat{\mu}_i)^2$ denotes the residual sum of squares and df is the rank of $\mathbf{X}$. For the lasso, an obvious plug-in alternative to (2.19) is

$$\hat{\sigma}^2_{\mathbf{P}} = \frac{\text{RSS}(\lambda)}{n - \text{df}(\lambda)}, \tag{2.20}$$

where $\text{RSS}(\lambda) = \sum_{i=1}^n \{y_i - \hat{\mu}_i(\lambda)\}^2$.

Estimator (2.20) is based on the observed fit of the model and makes no adjustment for overfitting. As a result, it tends to underestimate σ^2, particularly for low values of λ. An alternative approach is to use an estimate of the out-of-sample prediction error in place of the observed $\text{RSS}(\lambda)$. A natural choice here is the cross-validation estimate: we denote the cross-validation based variance estimate $\hat{\sigma}^2_{\text{CV}} = \text{CV}(\lambda)$.

Other, more computationally intensive methods have also been proposed involving sample splitting. Consider randomly partitioning the dataset into two sets $D_1 = (\mathbf{y}^{(1)}, \mathbf{X}^{(1)})$ and $D_2 = (\mathbf{y}^{(2)}, \mathbf{X}^{(2)})$ with sizes n_1 and n_2, respectively, where $n_1 + n_2 = n$. The idea behind the sample splitting is to use the lasso on D_1 for the purposes of variable selection, and then fit an OLS model to D_2 using the selected variables for the purposes of estimating σ^2. Let $\hat{s}_1$ be the set of predictors selected based

on applying the lasso to D_1 and $\text{RSS}(D_2, \hat{s}_1)$ denote the residual sum of squares from the OLS model fit to D_2 using only the variables in $\hat{s}_1$. Then

$$\hat{\sigma}_1^2 = \frac{\text{RSS}(D_2, \hat{s}_1)}{n_2 - |\hat{s}_1| + 1}$$

is the OLS estimate of the residual variance based on data D_2 and variables $\hat{s}_1$. The same procedure can be applied in the opposite direction to obtain $\hat{\sigma}_2^2$, the OLS estimate based on data D_1 and variables $\hat{s}_2$. These estimates can then be combined to form a refitted cross-validation estimate

$$\hat{\sigma}_{\text{RCV}}^2 = \frac{n_1}{n}\hat{\sigma}_1^2 + \frac{n_2}{n}\hat{\sigma}_2^2. \tag{2.21}$$

The above procedure can be repeated several times and averaged to obtain a more stable estimate.

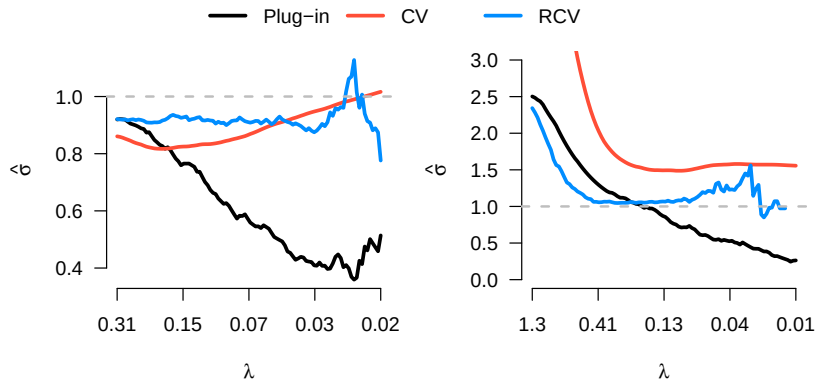

FIGURE 2.8
Estimates of σ. In both cases, $n = 100$, $p = 1,000$, and the true $\sigma = 1$. Left: $\boldsymbol{\beta} = \mathbf{0}$. Right: $\beta_j = 1$ for $j = 1, 2, \ldots, 5$; $\beta_j = 0$ for $j = 6, 7, \ldots, 1000$.

Figure 2.8 shows the plug-in, cross-validation, and refitted cross-validation estimates applied to data simulated from model (2.1). For the case plotted on the left-hand side, $\boldsymbol{\beta} = \mathbf{0}$, while for the right-hand case, five coefficients are equal to 1 and the rest equal to zero. In both cases, the true $\sigma = 1$. In the null case, we can see that the CV and RCV estimators give reasonable results. In the presence of a strong signal,

however, the CV estimator has a tendency to overestimate σ^2 somewhat, while the RCV estimator yields rather accurate estimates over a wide range of λ values. The plug-in estimator $\hat{\sigma}^2_{\mathbf{P}}$ is clearly not a reliable estimator of σ^2 across the entire range of λ values: it does not correct for overfitting, and clearly underestimates σ^2 when λ is small. However, it is worth noting that the plug-in estimator does perform well in both scenarios at the specific value of λ_{CV}. Although Figure 2.8 depicts only a case study of two simulated data sets, these general conclusions have been supported by more extensive simulation studies (Fan et al., 2012; Reid et al., 2016).

2.6.2 Estimating the coefficient of determination

One reason that estimating σ^2 is of considerable practical interest is that it enables us to estimate the proportion of variance in the outcome that can be explained by the model. This quantity, familiar from classical regression, is known as the *coefficient of determination* and denoted R^2.

The coefficient of determination is given by

$$R^2 = 1 - \frac{\text{Var}(Y|\mathbf{X})}{\text{Var}(Y)}. \tag{2.22}$$

The estimation of $\sigma^2 = \text{Var}(Y|\mathbf{X})$ was discussed in 2.6.1; estimation of $\text{Var}(Y)$ is straightforward.

Once cross-validation has been carried out, calculation of R^2 is straightforward. This can easily be carried out manually in `glmnet`:

```
cvfit <- cv.glmnet(X, y)
rsq <- 1-cvfit$cvm/var(y)
```

and is implemented as a default plot type in `ncvreg`:

```
cvfit <- cv.ncvreg(X, y, penalty="lasso")
plot(cvfit, type="rsq")
```

The resulting plot is shown in Figure 2.9. As the figure shows, at its maximum, the lasso-penalized linear regression model is capable of explaining 58% of the variability in mortality. Only a small amount of this comes from the pollution variables, however: the peak R^2, leaving out the three pollution variables, is 56%.

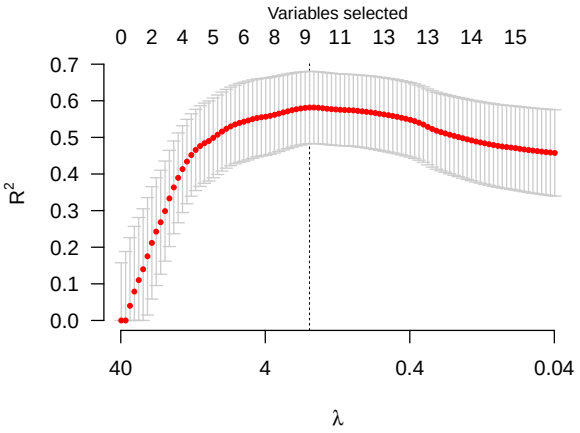

FIGURE 2.9
Estimates of R^2 for the pollution data.

2.7 Case study: Breast cancer gene expression study

As a case study in applying the lasso to high-dimensional data, we use breast cancer data from The Cancer Genome Atlas (TCGA) project. In this dataset, tumour samples were assayed on several platforms. Here we focus on the gene expression data obtained using Agilent mRNA expression microarrays. In this dataset, expression measurements of 17814 genes, including BRCA1, from 536 patients are available at `http://cancergenome.nih.gov/`. All expression measurements are recorded on the log scale.

BRCA1 is the first gene identified that increases the risk of early onset breast cancer. Because BRCA1 is likely to interact with many other genes, including tumor suppressors and regulators of the cell division cycle, it is of interest to find genes with expression levels related to that of BRCA1. These genes may be functionally related to BRCA1 and are useful candidates for further studies.

For this study, we excluded 491 genes with missing data, resulting in a design matrix with $p = 17,322$ predictors. We start by fitting, and then plotting, the lasso solution path together with 10-fold cross validation results:

```
cvfit <- cv.glmnet(X, y)
fit <- cvfit$glmnet.fit
xlim <- log(c(fit$lambda[1], cvfit$lambda.min))
plot(fit, xlim=xlim, xvar="lambda")
plot(cvfit)
```

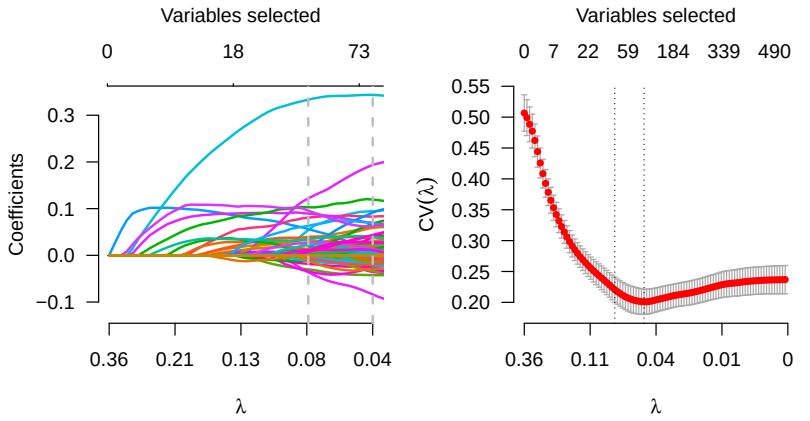

FIGURE 2.10
Lasso analysis of TCGA breast cancer data. Dotted lines denote $\hat{\lambda}$ and $\hat{\lambda}_{1\text{SE}}$.

The output of this code is shown in Figure 2.10. Here, to cut down on clutter in the coefficient path, we plot the path only up until the λ value that minimizes CV(λ). Note that the complete-data lasso path is included with the output of `cv.glmnet`; it is not necessary to call `glmnet` to obtain it.

The vertical lines are drawn at $\hat{\lambda}$ and $\hat{\lambda}_{1\text{SE}}$, the value that minimizes CV(λ) and the largest λ value within 1 SE of the minimum, respectively. By the sharp drop in CV error between $\lambda = 0.36$ and $\hat{\lambda} = 0.045$, we can see that the model successfully explains a substantial fraction of the variability in BRCA1 expression. Specifically, the maximum R^2 of the model is 0.60:

```
max(1-cvfit$cvm/var(y))
[1] 0.6041819
```

It is also fairly clear that lowering λ past 0.045 results in progressively worse predictions.

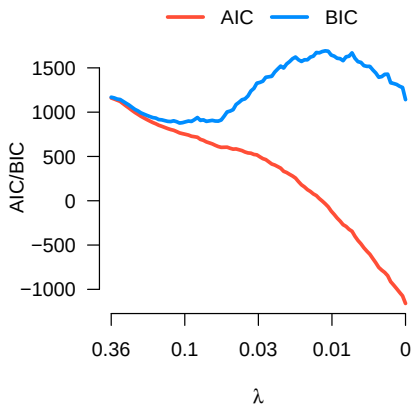

FIGURE 2.11
Application of AIC and BIC for the lasso analysis of TCGA breast cancer data.

Figure 2.11 shows the calculation of AIC and BIC for the breast cancer data. Unlike the low-dimensional pollution data example, in this high-dimensional problem AIC gives drastically different results from cross-validation. In particular, AIC offers no protection against overfitting, and is minimized at the (unidentifiable) unpenalized model. The cross-validation results indicate that this estimate of prediction error is almost certainly wrong. BIC is more reasonable, suggesting, like cross-validation, a regularization parameter somewhere in the range $0.05 < \lambda < 0.10$. However, note that the BIC criterion begins to decrease again as $\lambda \to 0$. Both the cross-validation results and common sense would indicate that this decrease is not meaningful, and merely the result of asymptotic approximations breaking down as the residual degrees of freedom approach zero. In this example, we stopped the path at $\lambda_{\min} = 0.005\lambda_{\max}$, but if we had continued, the decrease would as well, and a blind application of BIC would also lead to selecting $\lambda = 0$. In general, while BIC can be useful in selecting λ, (a) cross-validation is generally more reliable, and (b) it is always a good idea to inspect a plot like Figure 2.11 when using BIC in high dimensions.

Like many `R` modeling functions, `glmnet` offers `coef` and `predict` methods to interact with the fitted model. For example, from the coefficient path we can see that one gene stands out as being particularly significant. Obviously, it would be of interest to know the identity of that gene. Using `glmnet`'s `coef` operator along with `R`'s usual subsetting methods, we learn that this gene is named NBR2:

```
> b <- coef(cvfit)
> b[which(b > 0.15),,drop=FALSE]
NBR2 0.3334144
```

NBR2 is adjacent to BRCA1 on chromosome 17, and recent experimental evidence indicates that the two genes share a promoter, so its appearance in the plot makes perfect sense. It is worth noting that NBR2 was not the first gene to be included in the lasso path – i.e., it was not the gene with the highest marginal association with BRCA1. This illustrates the power of a regression-based approach over single gene association test to identify the most important biological factors from a large volume of noisy data.

By default, the `coef` method for `cv.glmnet` returns $\widehat{\boldsymbol{\beta}}(\hat{\lambda}_{1\mathrm{SE}})$ and the `coef` method for `glmnet` returns a matrix of $\widehat{\boldsymbol{\beta}}$ values for the entire grid, but one can obtain $\widehat{\boldsymbol{\beta}}(\lambda)$ for any λ value of interest. For example, we can see that at $\lambda = 0.2$, there are eight nonzero gene coefficients (plus the intercept):

```
> b <- coef(fit, s=0.2)
> sum(b != 0)
[1] 9
```

Finally, we illustrate the use of `predict` to obtain predictions of BRCA1 expression levels given expression levels for the other genes with nonzero coefficients in the model. For example, to obtain the predicted BRCA1 level for subject 85,

```
> predict(cvfit, X[85,,drop=FALSE])
[1,] -0.4495948
```

The range of BRCA1 expression in this study ranged from -3.9 to 0.5, so this actually represents a fairly high expected value.

2.8 Case study: Relative tumor size prediction

We present here a second case study involving a lasso analysis, both for the sake of variety and to illustrate the incorporation of unpenalized variables into an analysis. The data presented here come from a study by Koussounadis et al. of gene expression changes in ovarian cancer(Koussounadis et al., 2014).

The current standard treatment for ovarian cancer consists of surgery,

followed by either carboplatin and paclitaxel or carboplatin alone. This approach, however, is not effective for all patients. The goal of this study was to identify genes and pathways associated with drug response. To identify such genes, the investigators implanted ovarian cell lines into adult mice and allowed the tumors to grow for 2 months, at which point one of three treatments (carboplatin, carboplatin + paclitaxel, or control) was administered to each mouse. At various time points ranging from 0 to 14 days following the initiation of treatment, the mice were sacrificed, at which point the investigators measured the size of the tumor as well as gene expression in the cancerous tissue.

Our analysis here concentrates on relative tumor volume (RTV) as the outcome variable. We take a log base 2 transformation of RTV so that $y = 1$ means that the tumor has doubled in size since baseline. By definition, $y = 0$ for all samples taken at day 0. For this study, there were 34,694 features with expression data and a sample size of 101 mice.

It would appear critical to control for treatment group and time of collection in analyzing these data, both of which are highly significant in a marginal analysis. Our goal, however, is to assess the relationship between gene expression and tumor growth while accounting for the experimental design. The lasso model (2.2) is easily extended to allow for such an analysis. Up to this point, we have kept λ the same across all variables, but all of the derivations in this chapter can be easily modified to allow variable j to have its own regularization parameter, λ_j. In particular, it is trivial to modify the soft-thresholding step (2.16) of the coordinate descent algorithm so that the update is $S(\tilde{z}_j|\lambda_j)$.

This straightforward extension of the basic lasso model is implemented in both the `glmnet` and `ncvreg` packages, albeit with a slight reparameterization. Those software packages allow one to modify the penalty applied to individual covariates through the use of a weighting factor: $\lambda_j = \lambda w_j$, where w_j is the multiplicative factor applied to term j, thereby producing the objective function

$$Q_\lambda(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}, \mathbf{w}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \sum_j w_j \left|\beta_j\right|. \tag{2.23}$$

The idea here is that w_j scales the baseline regularization factor λ up or down for certain covariates.

In general, one could envision carefully choosing a unique w_j for each coefficient based on the likelihood that the feature will play a role in determining the outcome. For example, we might wish for genes that have been implicated in past cancer studies to receive less penalization than other genes.

Our goal here is more simple: by assigning $w_j = 0$ for the treatment group and time of collection variables, we can include them in the model

as unpenalized covariates. Unlike the gene expression variables, it is unlikely that treatment group and time of collection have a near-zero effect on relative tumor volume, so it does not make sense that they should receive the same penalization.

In the code below, we construct a 2 degree of freedom spline to represent the effect of day of collection and allow for an interaction between day of collection and treatment group. One advantage of penalized regression is that, by preserving the basic structure of regression, building relatively complex models such as this is as straightforward as it in ordinary linear modeling.

```
library(splines)
sDay <- ns(sampleData$Day, df=2)
X0 <- model.matrix(~ Treatment*sDay, sampleData)[,-1]
w <- rep(0:1, c(ncol(X0), ncol(X)))
XX <- cbind(X0, X)
```

Here, we have constructed a new design matrix, `XX`, by prepending the treatment group and day of collection covariates (without an intercept), `X0`, to the matrix of gene expression data, `X`. We can then carry out the analysis in `glmnet` or `ncvreg` as follows:

```
y <- log2(sampleData$RTV)
cvfit <- cv.glmnet(XX, y, penalty.factor=w)

## Or:

cvfit <- cv.ncvreg(XX, y, penalty.factor=w, penalty='lasso')
```

We thought it would be more interpretable here to plot the R^2 of the model; the plots in Figure 2.12 were produced with:

```
fit <- cvfit$fit         # cv.ncvreg output
plot(fit)                # Left side
plot(cvfit, type='rsq')  # Right side
```

In this example, R^2 is not zero even at $\lambda_{\max}$; treatment group and day of collection (which, along with their interaction, consists of 8 covariates) alone explain 36% of the variability in RTV. Nevertheless, the gene expression data seems to provide additional predictive benefit beyond that of treatment group and day of collection: by including the gene expression variables, we can increase the R^2 to 45%.

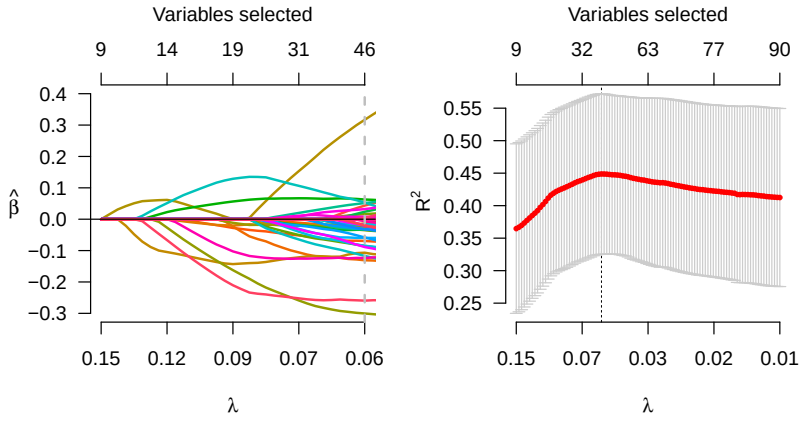

FIGURE 2.12
Lasso analysis of ovarian cancer chemotherapy data. Dotted line denotes the value of λ that minimizes the cross-validation error.

2.9 Bayesian interpretation

As with ridge regression (Section 1.6.3), the lasso objective function (2.2) can be seen to arise from a Bayesian formulation of the regression model. Here, the prior on the regression coefficients is a Laplace, or double-exponential, distribution as opposed to a normal distribution:

$$p(\boldsymbol{\beta}) = \prod_{j=1}^{p} \frac{\gamma}{2} \exp(-\gamma|\beta_j|) = \left(\frac{\gamma}{2}\right)^p \exp(-\gamma\|\boldsymbol{\beta}\|_1), \gamma > 0.$$

The MAP estimator of $\boldsymbol{\beta}$ is therefore

$$\widehat{\boldsymbol{\beta}}_{\text{MAP}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\{\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \gamma\|\boldsymbol{\beta}\|_1\}.$$

This can be written as

$$\widehat{\boldsymbol{\beta}}_{\text{MAP}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\{\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \frac{\gamma\sigma^2}{n}\|\boldsymbol{\beta}\|_1\}.$$

So the lasso solution $\widehat{\boldsymbol{\beta}}(\lambda)$ corresponds to the MAP estimator $\widehat{\boldsymbol{\beta}}_{\text{MAP}}$ with $\lambda = \gamma\sigma^2/n$.

An alternative Bayesian approach can also be adopted as follows, based on an interesting result that the Laplace prior can be represented as a scale mixture of normals. Namely, for $\gamma > 0$,

$$\int_0^\infty \frac{1}{\sqrt{2\pi s}} \exp(-\frac{t^2}{2s}) \frac{\gamma^2}{2} \exp(-\frac{\gamma^2 s}{2}) ds = \frac{\gamma}{2} \exp(-\gamma|t|). \tag{2.24}$$

Now suppose that β_j has a zero mean Gaussian prior with variance τ_j^2, $p(\beta_j|\tau_j^2) = \mathrm{N}(0, \tau_j^2)$, and that each τ_j^2 has an exponential (hyper) prior,

$$p(\tau_j^2) = \frac{\gamma^2}{2} \exp\left(-\frac{\gamma^2}{2}\tau_j^2\right), \quad \tau_j^2 > 0.$$

Then by (2.24),

$$p(\beta_j) = \int_0^\infty p(\beta_j|\tau_j^2) p(\tau_j^2) d\tau_j^2 = \frac{\gamma}{2} \exp(-\gamma|\beta_j|).$$

This gives the following hierarchical structure of the model:

$$\begin{aligned}
\mathbf{y}|\boldsymbol{\beta}, \sigma^2 &\sim \mathrm{N}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n), \\
\boldsymbol{\beta}|\sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim \mathrm{N}_p(\mathbf{0}_p, \sigma^2 \mathbf{D}_{\boldsymbol{\tau}}), \\
\mathbf{D}_{\boldsymbol{\tau}} &= \mathrm{diag}(\tau_1^2, \ldots, \tau_p^2), \\
\sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim p(\sigma^2) \left(\frac{\gamma^2}{2}\right)^p \prod_{j=1}^p \exp\left(-\frac{\gamma^2}{2}\tau_j^2\right).
\end{aligned}$$

Here it is convenient, for the sake of conjugacy, to take $p(\sigma^2) = 1/\sigma^2$.

Let $\mathbf{A} = \mathbf{X}^T\mathbf{X} + \mathbf{D}_{\boldsymbol{\tau}}^{-1}$. The full conditional distributions of the parameters are as follows.

$$\begin{aligned}
\boldsymbol{\beta}|\mathbf{y}, \sigma^2, \tau_1^2, \ldots, \tau_p^2 &\sim N(\mathbf{A}^{-1}\mathbf{X}^T\mathbf{y}, \sigma^2\mathbf{A}^{-1}), \\
\sigma^2|\mathbf{y}, \boldsymbol{\beta}, \tau_1^2, \ldots, \tau_p^2 &\sim \text{inverse-gamma } (\alpha, \beta)
\end{aligned}$$

with shape parameter $\alpha = (n - 1 + p)/2$ and scale parameter $\beta = \frac{1}{2}(\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \boldsymbol{\beta}^T\mathbf{D}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\beta})$, where the inverse-gamma density is

$$p(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} e^{-\beta/x}, x > 0.$$

In addition, $\tau_1^2, \ldots, \tau_p^2|\mathbf{y}, \boldsymbol{\beta}, \sigma^2$ are conditionally independent with $1/\tau_j^2$ conditionally distributed as inverse-Gaussian with parameters $a = \sqrt{\lambda^2\sigma^2/\beta_j^2}$ and $b = \lambda^2$, where the inverse-Gaussian density is given by

$$p(x; a, b) = \sqrt{\frac{b}{2\pi}} x^{-3/2} \exp\left\{-\frac{b(x-a)^2}{2a^2 x}\right\}, x > 0.$$

Based on these full conditional distributions, a Gibbs sampling approach can be used to sample from the posterior distribution that alternatively updates $\boldsymbol{\beta}, \sigma^2$ and $(\tau_1^2, \ldots, \tau_p^2)$. This requires either estimating the tuning parameter λ or giving it an appropriate hyperprior.

Sampling from the posterior distribution of $\boldsymbol{\beta}$ certainly provides richer information than simply finding the MAP estimate; for example, we can obtain posterior intervals for $\boldsymbol{\beta}$. However, unlike finding lasso estimates, this approach is unable to take advantage of sparsity. All point estimates are nonzero at all stages at all steps of the Gibbs sampler, which renders the approach computationally prohibitive for problems with large p.

2.10 *Subdifferential calculus and convex optimization

Let f be a convex function defined on a convex set $A \subseteq \mathbb{R}^p$. The subdifferential of f at a point $\mathbf{x} \in A$ is defined as

$$\partial f(\mathbf{x}) := \{\mathbf{w} \in \mathbb{R}^p : f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{w}^T(\mathbf{y} - \mathbf{x}), \forall \mathbf{y} \in A\}.$$

- A convex function f possesses a subdifferential at every interior point of its domain.
- If f is convex and differentiable, then

$$\partial f(\mathbf{x}) = \nabla f(\mathbf{x}), \tag{2.25}$$

  where $\nabla f(\mathbf{x})$ is the differential (or gradient) of f at $\mathbf{x}$.
- For two convex functions f and g defined on the same domain,

$$\partial(f + g)(\mathbf{x}) = \partial f(\mathbf{x}) + \partial g(\mathbf{x}). \tag{2.26}$$

Example 2.2. (**Subdifferential of $|x|$.**) The subdifferential of $f(x) = |x|, x \in \mathbb{R}$ is

$$\partial f(x) = \begin{cases} 1, & \text{if } x > 0, \\ [-1, 1], & \text{if } x = 0, \\ -1, & \text{if } x < 0. \end{cases} \tag{2.27}$$

To see this, verify that $|y| - |x| \geq \partial f(x)(y - x)$ for every $y \in \mathbb{R}$. □

For a convex function, its global minimum can be characterized by the KKT condition

$$\mathbf{z}^* \in \arg\min_{\mathbf{z}\in\mathbb{R}^p} \; f(\mathbf{z}) \text{ if and only if } \mathbf{0} \in \partial f(\mathbf{z}^*). \tag{2.28}$$

This is can be verified as follows. By that definition of subdifferential, $\mathbf{0} \in \partial f(\mathbf{z}^*)$ if and only $f(\mathbf{z}) - f(\mathbf{z}^*) \geq \mathbf{0}^T(\mathbf{z} - \mathbf{z}^*) = 0,$ for every $\mathbf{z}$. Thus $\mathbf{z}^* \in \arg\min f(\mathbf{z})$.

Note that (2.28) is a generalization of Fermat's rule for differentiable functions, that is, when f is differentiable, then

$$\mathbf{z}^* = \arg\min_{\mathbf{z}\in\mathbb{R}^p} \; f(\mathbf{z}) \text{ if and only if } f'(\mathbf{z}^*) = \mathbf{0}.$$

A useful generalization of (2.28) is

$$\mathbf{w} \in \partial f(\mathbf{z}) \text{ if and only if } \mathbf{z} = \text{Prox}_f(\mathbf{z} + \mathbf{w}), \tag{2.29}$$

where Prox_f is the proximity operator for f defined as

$$\text{Prox}_f(\mathbf{z}) := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + f(\mathbf{x}).$$

This can be shown as follows. By (2.28),

$$\begin{aligned} \mathbf{z} = \text{Prox}_f(\mathbf{z} + \mathbf{w}) \quad &= \quad \arg\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{x} - \mathbf{z} - \mathbf{w}\|_2^2 + f(\mathbf{x}) \\ &\Leftrightarrow \quad \mathbf{0} \in (\mathbf{z} - \mathbf{z} - \mathbf{w}) + \partial f(\mathbf{z}) \\ &\Leftrightarrow \quad \mathbf{0} \in -\mathbf{w} + \partial f(\mathbf{z}) \\ &\Leftrightarrow \quad \mathbf{w} \in \partial f(\mathbf{z}). \end{aligned}$$

The proximity operator of $\lambda\|\cdot\|_1$ is given in a closed form by the componentwise soft threshold operator, i.e.,

$$\text{Prox}_{\lambda\|\mathbf{x}\|_1}(\mathbf{z}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \lambda\|\mathbf{x}\|_1 = S_\lambda(\mathbf{z}), \tag{2.30}$$

where $S(\mathbf{z}|\lambda) := [S(z_1|\lambda), \ldots, S(z_p|\lambda)]'$ and $S(\cdot|\lambda)$ is the soft threshold operator given in (2.14), which can also be written as

$$S(z|\lambda) = z - \frac{|z + \lambda|}{2} + \frac{|z - \lambda|}{2}. \tag{2.31}$$

Recall the lasso criterion

$$Q(\boldsymbol{\beta}; \lambda) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{2.32}$$

According to Fermat's rule (2.28), $\widehat{\boldsymbol{\beta}}$ minimizes $Q(\boldsymbol{\beta};\lambda)$ if and only if

$$-\frac{1}{n}\mathbf{X}^T(\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}})+\lambda\partial(\|\widehat{\boldsymbol{\beta}}\|_1)=\mathbf{0},$$

In view of (2.27), this is exactly (2.6).

We now describe another characterization of the lasso solution based on the equations

$$\mathbf{d} = \frac{1}{n}\mathbf{X}^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}), \tag{2.33}$$

$$\boldsymbol{\beta} = S(\boldsymbol{\beta}+\mathbf{d}|\lambda), \tag{2.34}$$

where $S(\cdot|\lambda)$ is the soft threshold operator defined in (2.14).

Proposition 2.1. *Let $\widehat{\boldsymbol{\beta}}\in\mathbb{R}^p$ be a lasso solution. Then there exists a $\mathbf{d}\in\mathbb{R}^p$ such that* (2.33) - (2.34) *hold. Conversely, if there exist $\widehat{\boldsymbol{\beta}},\mathbf{d}\in\mathbb{R}^p$ satisfying* (2.33)-(2.34)*, then $\widehat{\boldsymbol{\beta}}$ is the a minimizer of L.*

Proof. Let $f_1(\boldsymbol{\beta})=\|\boldsymbol{\beta}\|_1$. We first assume that $\widehat{\boldsymbol{\beta}}\in\mathbb{R}^p$ is a minimizer of (2.32). Then, by (2.28), we have

$$\mathbf{0}\in\frac{1}{n}\mathbf{X}^T(\mathbf{X}\widehat{\boldsymbol{\beta}}-\mathbf{y})+\lambda\partial f_1(\widehat{\boldsymbol{\beta}}).$$

Therefore, there exists $\mathbf{d}\in\lambda\partial f_1(\widehat{\boldsymbol{\beta}})$ such that

$$\mathbf{0}=\frac{1}{n}\mathbf{X}^T(\mathbf{X}\widehat{\boldsymbol{\beta}}-\mathbf{y})+\mathbf{d},$$

that is, (2.33) holds.. Furthermore, by (2.29), the inclusion

$$\mathbf{d}\in\lambda\partial f_1(\widehat{\boldsymbol{\beta}})$$

is equivalent to

$$\widehat{\boldsymbol{\beta}}=\mathrm{Prox}_{\lambda\partial f_1(\widehat{\boldsymbol{\beta}})}(\widehat{\boldsymbol{\beta}}+\mathbf{d}).$$

Therefore, by (2.30), we have

$$\widehat{\boldsymbol{\beta}}=S(\widehat{\boldsymbol{\beta}}+\mathbf{d}|\lambda),$$

which proves (2.34).

Conversely, suppose (2.33) and (2.34) hold for some $\widehat{\boldsymbol{\beta}},\mathbf{d}\in\mathbb{R}^p$. By (2.29) and (2.30) again, we deduce $\mathbf{d}\in\lambda\partial f_1(\widehat{\boldsymbol{\beta}})$ from (2.34). Substituting this into (2.33), we have

$$\mathbf{0}\in\mathbf{X}^T(\mathbf{X}\widehat{\boldsymbol{\beta}}-\mathbf{y})+\lambda\partial f_1(\widehat{\boldsymbol{\beta}}),$$

which shows that $\widehat{\boldsymbol{\beta}}$ is a minimizer of (2.32) by Fermat's rule (2.28). □

Bibliographical notes

This section will include the bibliographical notes on the materials presented in this chapter.

Exercises

2.1. *Unique/non-unique lasso solutions.* Suppose $n = 2$ and $p = 2$, with $(y_1, x_{11}, x_{12}) = (1, 1, 1)$ and $(y_2, x_{21}, x_{22}) = (-1, -1, -1)$. Show that the lasso solution is the one given in Example 2.1.

2.2. *Convexity of loss functions.*

(a) Show that the lasso objective function (2.2) is convex, but not necessarily strictly convex. Under what condition(s) will the objective function be strictly convex?

(b) Let $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ denote the linear predictors (fitted values) for a regression model. Let

$$L(\boldsymbol{\eta}|\mathbf{y}) = \frac{1}{2n}\|\mathbf{y} - \boldsymbol{\eta}\|_2^2$$

denote the loss function for linear regression, written as a function of $\boldsymbol{\eta}$. Show that the loss is strictly convex as a function of $\boldsymbol{\eta}$.

2.3. *Uniqueness of lasso predictions.* This exercise concerns claim (2.9), that even though the lasso estimates $\widehat{\boldsymbol{\beta}}$ may not be unique, their fitted values are. Suppose $\widehat{\boldsymbol{\beta}}_1$ and $\widehat{\boldsymbol{\beta}}_2$ are two distinct lasso solutions; that is, both $\widehat{\boldsymbol{\beta}}_1$ and $\widehat{\boldsymbol{\beta}}_2$ minimize $Q(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y})$ at a given value of λ, with $\widehat{\boldsymbol{\beta}}_1 \neq \widehat{\boldsymbol{\beta}}_2$. Show that $\mathbf{X}\widehat{\boldsymbol{\beta}}_1 = \mathbf{X}\widehat{\boldsymbol{\beta}}_2$. Hint: Use the result from Exercise 2.2(b) and the definition of a convex function.

2.4. *Standardization and t-tests.* Let $\mathbf{x}_j$ and $\mathbf{y}$ be centered, but not scaled, with $\tilde{x}_j$ the scaled version of $\mathbf{x}_j$. Consider the orthogonal case, in which $\mathbf{x}_j^T\mathbf{x}_k = 0$ for all j, k. As we have seen, the standardized lasso will select a feature if $n^{-1}\left|\tilde{x}_j^T\mathbf{y}\right| > \lambda$.

(a) Show that for the unstandardized lasso, a feature is selected if

$$\tfrac{1}{n}\left|\tilde{x}_j^T\mathbf{y}\right| > \lambda\sqrt{\mathbf{x}_j^T\mathbf{x}_j/n}.$$

(b) Show that for OLS regression, the classical t-test declares a feature to be significant for a given critical value t^* if

$$\tfrac{1}{n}\left|\tilde{x}_j^T \mathbf{y}\right| > \lambda,$$

where $\lambda = t^* \hat{\sigma}/\sqrt{n}$.

In other words, there is a certain equivalence between the lasso and classical testing, but this equivalence is lost if the features are not standardized.

2.5. *Soft thresholding is the lasso solution in the orthonormal case.* Show that the soft thresholding operator defined in (2.13) minimizes (2.11).

2.6. *Programming the coordinate descent algorithm.* Write an R function called `lasso()` that implements the coordinate descent algorithm for the lasso assuming a standardized feature matrix. In other words, calling `lasso(X, y, lambda=0.1)` should return the correct coefficients assuming that `X` is standardized as one might get from `ncvreg::std(X)`, for example.

2.7. *Programming the coordinate descent algorithm, part 2.* Build upon the function from Exercise 2.6, now dropping the requirement that the feature matrix must be pre-standardized. The function should begin by standardizing the feature matrix, then using the coordinate descent algorithm as before, and then at the end reversing the standardization process so that the coefficients are returned on the original scale.

2.8. *Programming the coordinate descent algorithm, part 3.* Build upon the function from Exercise 2.7, only now fit the entire solution path. The function should calculate a grid of λ values equally spaced on the log scale, starting at $\lambda_{\max}$, then loop over these values. A matrix of coefficients, one for each feature and value of λ, should be returned.

2.9. *Degrees of freedom for the lasso (orthonormal case).* Let $f(y_i)$ denote the fitted value for observation i, considered as a function of the observed value y_i. For regression model (2.1), Stein's lemma (Stein, 1981) states the degrees of freedom for a fitting method is given by

$$\mathrm{df} = \sum_{i=1}^{n} f'(y_i),$$

provided that f is absolutely continuous and $\mathbb{E}|f'| < \infty$. Use this lemma to show that the degrees of freedom for the lasso is equal to the number of nonzero coefficients. For the sake of this problem, assume that the features are orthonormal: $\mathbf{x}_j^T \mathbf{x}_k = 0$ for all j and k, and $\frac{1}{n}\mathbf{x}_j^T \mathbf{x}_j = 1$ for all j. A proof in the general case is provided in Zou et al. (2007).

2.10. *WHO study of acute respiratory illness, revisited.* Re-analyze the pneumonia data from Exercise 1.11, this time using the lasso instead of ridge regression.

(a) How many coefficients are nonzero for the value of λ that minimizes cross-validation error? How many are nonzero for the largest value of λ within 1 SE of the minimum CV error? Is there convincing evidence that these models are better than the null model? Is there convincing evidence that they are better than the OLS model?

(b) Briefly, describe the variables that appear to be *most* important based on the lasso estimates.

(c) Comment on the number of nonzero coefficients in the model that minimizes CVE and how that number compares to the number of statistically significant coefficients in the OLS model. Which criterion is more liberal?

(d) For the model that minimizes CVE, describe how the lasso estimates of sucking ability (`absu`) and drinking ability (`afe`) compare to the ridge and OLS estimates. Which estimates do you consider to be most reasonable? Why?

2.11. *Simulation comparing ridge regression, forward selection, and the lasso.* For this simulation, generate the elements of the design matrix according to $X_{ij} \sim \mathrm{N}(0,1)$, with $y_i \sim \mathrm{N}(\mathbf{x}_i^T\boldsymbol{\beta}, 1)$. The simulation should compare three modeling approaches: (1) forward selection, (2) ridge regression, and (3) the lasso. To carry out forward selection, you can use the `step()` function:

```
step(fit0, scope=form, direction="forward", k=log(n), steps=n-5)
```

where `fit0` is the fit from the null model and `form` is a formula describing the full model. Here, `k=log(n)` specifies the use of BIC as a stopping rule.

Consider the following simulation settings, each with $n = 50$:

(I) Let $p = 16$ and set two $\beta = 5$, two $\beta = -5$, and the rest equal to zero.

(II) Let $p = 100$ and set two $\beta = 5$, two $\beta = -5$, and the rest equal to zero.

(III) Let $p = 50$ and set five $\beta = 1$, five $\beta = -1$, and the rest equal to zero.

(IV) Let $p = 100$ and set twenty-five $\beta = 0.5$, twenty-five $\beta = -0.5$, and the rest equal to zero.

For each simulation setting, calculate the squared error $\|\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\|_2^2$ for each method and present a box plot that compares this squared error loss across the methods. Based on your results, comment on the situations in which you would expect ridge, forward selection, or the lasso to yield the most accurate estimates.

3

Bias reduction

3.1 Adaptive lasso

Although the lasso has many excellent properties as discussed in Chapter 2, it is also a biased estimator. This can be seen from the penalized score equation (KKT condition)

$$\frac{1}{n}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) - \lambda \mathbf{s}(\boldsymbol{\beta}) = 0.$$

A fundamental property of classical maximum likelihood estimation is that the expected value of the score statistic is zero when evaluated at the true value of the unknown parameter. This is not true of the penalized score equation for the lasso:

$$\mathbf{E}[\frac{1}{n}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) - \lambda \mathbf{s}(\boldsymbol{\beta})] = -\lambda \mathbf{s}(\boldsymbol{\beta}).$$

In other words, the bias of the estimating equation is on the order of λ, since the range of s is $[-1, 1]$.

To see this another way, consider the simple scenario where $\mathbf{X}$ is orthogonal and the lasso has a closed form solution (2.13). Then

$$\begin{cases} \mathbf{E}|\widehat{\beta}_j - \beta_j| = 0 & \text{if } \beta_j = 0 \\ \mathbf{E}|\widehat{\beta}_j - \beta_j| \approx \beta_j & \text{if } |\beta_j| \in [0, \lambda] \\ \mathbf{E}|\widehat{\beta}_j - \beta_j| \approx \lambda & \text{if } |\beta_j| > \lambda \end{cases}$$

The last two results are only approximate because the bias depends on the probability that $|\widehat{\beta}_{\text{OLS}}| > \lambda$, which in turn depends on the sample size. However, at least for reasonably large samples, the bias of the lasso estimate is about λ for large regression coefficients.

Given that the bias of the estimate is determined by λ, one approach to reducing the bias of the lasso is to use the weighted penalty approach of (2.23). If one was able to choose the weights $\mathbf{w}$ such that the variables with large coefficients had smaller weights, then we could reduce the

estimation bias of the lasso while retaining its sparsity property. Indeed, by more accurately estimating $\boldsymbol{\beta}$, one would even be able to improve on the variable selection accuracy of the lasso.

All of this may seem circular in the sense that if we already knew which regression coefficients were large and which were small, we wouldn't need to be carrying out a regression analysis in the first place. However, it turns out that the choice of $\mathbf{w}$ does not need to be terribly precise in order to realize benefits from this approach. In practice, one can obtain reasonable values for $\mathbf{w}$ from an initial estimator of $\boldsymbol{\beta}$. Theoretical justifications for this approach require the initial estimator to be consistent. Thus, in low-dimensional models with $n \gg p$, the OLS estimator can be used as the initial estimator. For high dimensional models with $p > n$, it is more difficult to obtain a good initial estimator, although the lasso solution itself would seem a natural choice.

Let $\widetilde{\boldsymbol{\beta}}$ denote the initial estimate. The *adaptive lasso* estimate $\widehat{\boldsymbol{\beta}}$ is then defined as the argument minimizing the following objective function:

$$\begin{aligned} Q(\boldsymbol{\beta}|\mathbf{X},\mathbf{y},\mathbf{w}) &= \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\sum_j w_j|\beta_j| \\ w_j &= |\widetilde{\beta}_j|^{-1}. \end{aligned} \tag{3.1}$$

In line with our earlier discussion, note that this weighting scheme assigns smaller weights to larger regression coefficients, based off of the initial estimate $\widetilde{\boldsymbol{\beta}}$. Note as well that if the initial estimate $\widetilde{\beta}_j = 0$, as would not be uncommon if the lasso were used as an initial estimator, we have $w_j = \infty$, so $\widehat{\beta}_j = 0$ in order to minimize (3.1).

In the above approach, known as a *two-stage approach*, a single initial estimate $\widetilde{\boldsymbol{\beta}}$ is made, which in turn produces a single set of weights $\mathbf{w}$, which are held constant across all values of λ in (3.1). An alternative approach, known as a *pathwise approach* is to let the weights change with λ:

$$w_j(\lambda) = w(\widetilde{\beta}_j(\lambda)).$$

Here, the initial estimate is typically a lasso estimator, so that λ has the same meaning for the initial estimator as it does for the re-weighted, or adaptive, estimator.

3.1.1 Alternative weighting strategies

There are many possibilities besides $w_j = |\widetilde{\beta}_j|^{-1}$ for choosing weights based on initial estimates, although to accomplish the goal of assigning small weights to large coefficients, it is reasonable to require that the

function $w(\beta)$ is nonincreasing. In principle, any of these weighting functions could be used in either a two-stage or adaptive approach, although the resulting estimators may be quite different.

One straightforward extension is to allow powers other than -1:

$$w_j = |\widetilde{\beta}_j|^{-\gamma},$$

where γ influences how dependent on the initial value the final estimates are. In particular, as $\gamma \to 0$, we arrive back at the original lasso estimator, as all coefficients with nonzero initial estimates are given the same weight.

Another possibility is to apply a thresholding operator to the initial estimator to produce weights:

$$w_j = \begin{cases} 0 & \text{if } |\widetilde{\beta}_j| > \tau, \\ 1 & \text{if } |\widetilde{\beta}_j| \leq \tau. \end{cases} \tag{3.2}$$

In this approach, if the initial solution $\widetilde{\beta}_j$ is greater than τ, the corresponding coefficient is not penalized in the second stage. This approach also differs from the earlier weighting strategies in that no coefficients are assigned infinite weights. Thus it is possible for variables to be selected by the final model even though they were not selected by the initial estimator.

Finally, a more extreme weighting scheme is

$$w_j = \begin{cases} 0 & \text{if } \widetilde{\beta}_j \neq 0, \\ \infty & \text{if } \widetilde{\beta}_j = 0. \end{cases} \tag{3.3}$$

When applied in a two-stage fashion, this approach is known as the *lasso-OLS hybrid* estimator, and is equivalent to fitting an OLS model to only those variables selected by the lasso estimator. In other words, we use the lasso for variable selection, but OLS for estimation. When the approach is applied in a pathwise fashion, it is known as the *relaxed lasso*.

3.2 Concave penalties

The adaptive lasso consists of a two-stage approach involving an initial estimator to reduce bias for large regression coefficients. An alternative single-stage approach is to use a penalty that tapers off as β becomes larger in absolute value. Unlike the absolute value penalty employed

by the lasso, a tapering penalty cannot be convex. Rather, the penalty function $P(\beta|\lambda)$ will be concave with respect to $|\beta|$. Such functions are often referred to as *folded concave penalties*, to clarify that while $P(\cdot)$ itself is neither convex nor concave, it is concave on both the positive and negative halves of the real line, and also symmetric (or folded) due to its dependence on the absolute value.

In this section, we write the objective function as

$$Q(\boldsymbol{\beta}|\mathbf{X},\mathbf{y}) = \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} P(\beta_j|\lambda,\gamma), \tag{3.4}$$

where $P(\beta|\lambda,\gamma)$ is a folded concave penalty. Unlike the lasso, many concave penalties depend on λ in a non-multiplicative way, so that $P(\beta|\lambda) \neq \lambda P(\beta)$. Furthermore, they typically involve a tuning parameter γ that controls the concavity of the penalty (i.e., how rapidly the penalty tapers off). A variety of concave penalties have been proposed, including: (a) the smoothly clipped absolute deviations (SCAD) penalty; (b) the minimax concave penalty (MCP); and (c) the capped ℓ_1 penalty; and (d) the bridge penalty.

3.2.1 SCAD and MCP

One of the earliest and most influential folded concave penalties was SCAD, with the penalty function

$$P(x;\lambda,\gamma) = \begin{cases} \lambda|x| & \text{if } |x| \leq \lambda, \\ \frac{2\gamma\lambda|x|-x^2-\lambda^2}{2(\gamma-1)} & \text{if } \lambda < |x| < \gamma\lambda, \\ \frac{\lambda^2(\gamma+1)}{2} & \text{if } |x| \geq \gamma\lambda \end{cases}$$

for $\gamma > 2$. Note that for $x \geq 0$, the SCAD penalty coincides with the lasso penalty until $x = \lambda$, then smoothly transitions to a quadratic function until $x = \gamma\lambda$, after which it remains constant for all $x > \gamma\lambda$. The penalty may be written more compactly as

$$P(x;\lambda,\gamma) = \lambda \int_0^{|x|} \min\{1, (\gamma - t/\lambda)_+/(\gamma-1)\}dt,$$

where $x_+ \equiv x1_{\{x\geq 0\}}$ denotes the nonnegative part of x.

It is typically more instructive to consider a penalty's derivative than the penalty itself, as the derivative is the contribution made by the penalty to the penalized estimating equations (KKT conditions). The

derivative of the SCAD penalty is

$$\dot{P}(x;\lambda,\gamma) = \begin{cases} \lambda & \text{if } |x| \leq \lambda, \\ \frac{\gamma\lambda - |x|}{\gamma - 1} & \text{if } \lambda < |x| < \gamma\lambda, \\ 0 & \text{if } |x| \geq \gamma\lambda. \end{cases}$$

This lends some insight into the bias reduction properties of the SCAD penalty. As remarked earlier, the bias of the lasso is on the order of its rate of penalization, λ. The SCAD penalty retains that penalization rate for small coefficients, but continuously relaxes the rate of penalization as the absolute value of the coefficient increases. This relaxation of the rate of penalization is linear and takes place until the rate of penalization equals zero.

The idea behind MCP is very similar. The penalty takes the form

$$P_\gamma(x;\lambda) = \begin{cases} \lambda|x| - \frac{x^2}{2\gamma}, & \text{if } |x| \leq \gamma\lambda \\ \frac{1}{2}\gamma\lambda^2, & \text{if } |x| > \gamma\lambda \end{cases} \tag{3.5}$$

for $\gamma > 1$, or more compactly,

$$P_\gamma(x;\lambda) = \lambda \int_0^{|x|} (1 - t/(\gamma\lambda))_+ dt. \tag{3.6}$$

Its derivative is

$$\dot{P}_\gamma(x;\lambda) = \begin{cases} (\lambda - \frac{|x|}{\gamma})\text{sign}(x), & \text{if } |x| \leq \gamma\lambda, \\ 0, & \text{if } |x| > \gamma\lambda. \end{cases} \tag{3.7}$$

As with SCAD, the MCP starts out by applying the same rate of penalization as the lasso, then smoothly relaxes the rate down to zero as the absolute value of the coefficient increases. In comparison to SCAD, however, the MCP relaxes the penalization rate immediately while with SCAD the rate remains flat for a while before decreasing. The lasso, SCAD, and MCP penalties are depicted in Figure 3.1.

As the figure indicates, the penalty functions for lasso, SCAD, and MCP are all continuous, symmetric about zero, and produce sparse estimates (as the rightmost plot indicates, $\widehat{\beta} = 0$ whenever the unpenalized solution is less than 1 in absolute value). Of the three, only the lasso is convex, although as we have seen, this results in biased estimates.

Figure 3.1 also illustrates the sense in which the MCP is *minimax concave*. Out of all penalty functions continuously differentiable on $(0, \infty)$ that satisfy $\dot{P}(0+;\lambda) = \lambda$ and $\dot{P}(t;\lambda) = 0$ for all $t \geq \gamma\lambda$, the MCP minimizes the maximum concavity

$$\kappa = \sup_{0<t_1<t_2} \frac{\dot{P}(t_1;\lambda) - \dot{P}(t_2;\lambda)}{t_2 - t_1}. \tag{3.8}$$

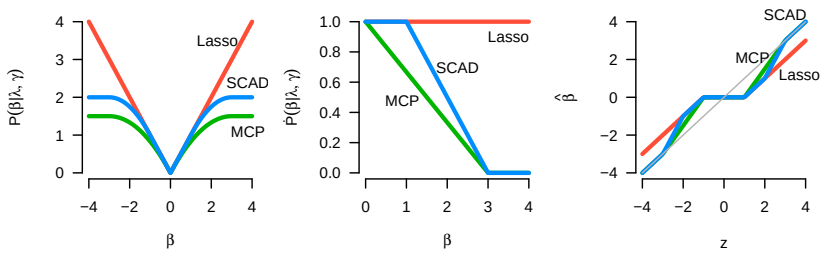

FIGURE 3.1
Shapes of the lasso, SCAD and MCP penalty functions. The panel on the left plots the penalties themselves; the panel in the middle plots the derivative of the penalty (note that none of the penalties are differentiable at 0); and the panel on the right plots the threshold operators corresponding to the penalties. On the right, z denotes the unpenalized OLS solution, as in Section 3.2.2. For the SCAD and MCP penalties, $\gamma = 3$.

As the figure shows, the derivatives of SCAD and MCP are equal at 0 and again at $\gamma\lambda$, but MCP has a constant concavity of $\kappa = 1/\gamma = 1/3$ over this region, while SCAD has a concavity of 0 from $t = 0$ to $t = \lambda$ and a concavity of $\kappa = 1/(\gamma - 1) = 1/2$ from $t = \lambda$ to $t = \gamma\lambda$. As we will discuss further in 3.6, as a penalty becomes more concave, optimization becomes more problematic as multiple local minima proliferate. Thus, MCP is typically more stable from an optimization standpoint than other folded concave penalties such as SCAD (for the same value of γ).

3.2.2 Solutions in the orthonormal case

As with the lasso, MCP and SCAD have closed-form solutions in the orthonormal case $n^{-1}\mathbf{X}^T\mathbf{X} = \mathbf{I}$ that provide insight into how the methods work. Here, we let $z = \mathbf{x}'\mathbf{y}/n$ denote the unpenalized (OLS) solution, as in Section 2.2.

For MCP, the univariate solution is known as the *firm thresholding operator*:

$$F(z|\lambda,\gamma) = \begin{cases} \frac{\gamma}{\gamma-1}S(z|\lambda) & \text{if } |z| \le \gamma\lambda, \\ z & \text{if } |z| > \gamma\lambda. \end{cases} \tag{3.9}$$

As $\gamma \to \infty$, the firm thresholding operator becomes equivalent to the soft thresholding operator: $F(z|\lambda, \gamma) \to S(z|\lambda)$. As $\gamma \to 1$, it becomes equivalent to hard thresholding. Thus, as γ changes, the solution bridges the gap between soft and hard thresholding; hence the name "firm thresholding".

The SCAD solution is similar, although somewhat more complicated. The SCAD thresholding operator is

$$T_{\text{SCAD}}(z|\lambda,\gamma) = \begin{cases} S(z|\lambda), & \text{if } |z| \leq 2\lambda, \\ \frac{\gamma-1}{\gamma-2} S(z|\frac{\gamma\lambda}{\gamma-1}), & \text{if } 2\lambda < |z| \leq \gamma\lambda, \\ z, & \text{if } |z| > \gamma\lambda. \end{cases} \tag{3.10}$$

As with MCP, $T_{\text{SCAD}}(\cdot|\lambda,\gamma) \to S(\cdot|\lambda)$ as $\gamma \to \infty$. However, as $\gamma \to 2$, $T_{\text{SCAD}}(\cdot|\lambda,\gamma)$ does not converge to hard thresholding; instead, it converges to

$$\begin{cases} S(z;\lambda), & \text{if } |z| \leq 2\lambda, \\ z, & \text{if } |z| > 2\lambda. \end{cases} \tag{3.11}$$

In other words, both T_{SCAD} and F converge to discontinuous functions as γ approaches its minimum value: for the firm thresholding operator F, the solution jumps from 0 to λ as z exceeds λ, while for the SCAD thresholding operator T_{SCAD}, the solution jumps from λ to 2λ as z exceeds 2λ.

3.2.3 Solution paths

To get a sense of how the MCP, SCAD, and adaptive lasso estimates compare to those of the regular lasso, we consider here the solution paths for the four penalties fit to the same data. We generate data from the linear regression model $y_i = \sum_{j=1}^{1000} x_{ij}\beta_j + \varepsilon_i, i = 1, \ldots, 200$, where $(\beta_1, \ldots, \beta_4) = (4, 2, -4, -2)$ and the remaining coefficients are zero. The `R` code to reproduce this data is given below:

```
set.seed(105)
n <- 200; p <- 1000
X <- matrix(rnorm(n*p), nrow=n, ncol=p)
z1 <- rnorm(n); z2 <- rnorm(n)
X[,1:4]   <- X[,1:4]+z1
X[,5]     <- X[,5]+2*z1
X[,6]     <- X[,6]+1.5*z1
X[,7:20]  <- X[,7:20]+0.5*z1
X[,21:40] <- X[,21:40]+0.5*z2
beta <- c(4, 2, -4,-2, rep(0, 996))
y <- rnorm(n, X%*%beta, sd=1.5)
```

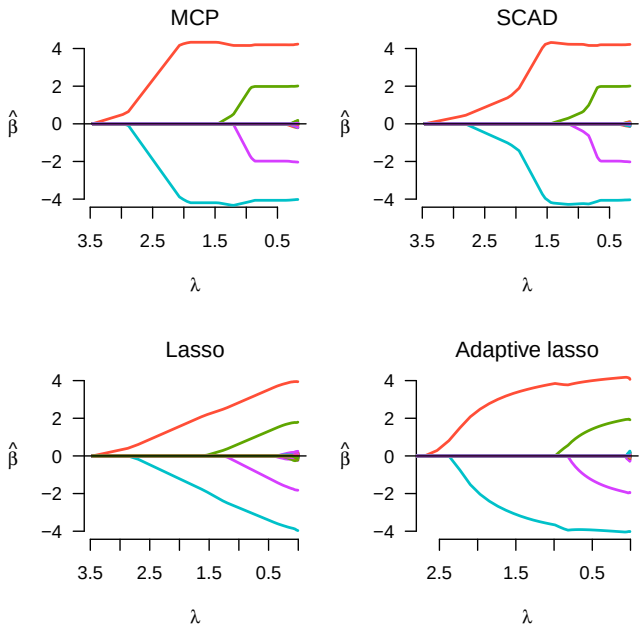

FIGURE 3.2
Solution paths for four different penalties fit to the same data.

Figure 3.2 depicts the solution paths for each of the four penalties for this data. For the adaptive lasso, we present the pathwise approach with weights $w_j(\lambda) = |\widetilde{\beta}_j(\lambda)|^{-1}$. For the MCP, we use $\gamma = 3$ while for SCAD we use $\gamma = 4$. As the paths show, the primary way in which the three penalties introduced in this chapter differ from the lasso is that they allow the estimated coefficients to reach large values more quickly than the lasso. In other words, although the methods all shrink most of the coefficients towards zero, MCP, SCAD, and the adaptive lasso apply less shrinkage to the nonzero coefficients; this is what we refer to in this chapter as *bias reduction*.

In this example, one can clearly see the piecewise components of MCP and SCAD. From $\lambda = 3$ down to $\lambda \approx 1.8$, the SCAD and lasso solutions are equivalent. For MCP, on the other hand, at $\lambda \approx 1.8$ the coefficient estimates for the two largest coefficients are very close to their true values, while the rest of the coefficients are still all zero. From $\lambda \approx 1.8$ to $\lambda \approx 1.4$, the SCAD estimates for the two largest coefficients make a fairly rapid transition from the lasso estimates to the unpenalized

estimates. A similar phenomenon happens for the other two coefficients. In particular, it is worth noting that both MCP and SCAD possess an interval of λ values over which all the estimates are flat – over this region, the estimates are the same as those of ordinary least squares regression, but with only the four variables with nonzero effects included. These estimates are referred to as the *oracle* estimates. They were first introduced in Section 1.3, and will be discussed further in Chapter 5.

In comparison, the adaptive lasso paths are not piecewise and change more smoothly as a function of λ. At $\lambda \approx 0.25$, the adaptive lasso solutions are quite close to the oracle estimates, although not exactly equal to them. Also, note that while $\lambda_{\max}$ is exactly the same for MCP, SCAD, and lasso, this is not true of the adaptive lasso.

3.2.4 The effect of γ

As discussed in the previous sections, the tuning parameter γ for the SCAD and MCP estimates controls how fast the penalization rate goes to zero. This, in turn, affects the bias of the estimates as well as the stability of the estimate (in the sense that as the penalty becomes more concave, there is a greater chance for multiple local minima to exist). As $\gamma \to \infty$, both the MCP and SCAD penalties converge to the ℓ_1 penalty. Here, bias is largest, but stability is greatest, as the optimization problem is once again convex. As γ approaches its minimum value, bias is minimized, but both estimates become unstable.

In the above paragraph, we used "stability" in the optimization sense that an objective function with a single, well-defined minimum is stable while optimization problems with multiple local minima tend to be unstable. However, the same remarks apply with respect to the statistical properties of the estimators, in the sense that a more highly variable estimator is less stable. For SCAD and MCP, lower values of γ also produce more highly variable (less stable) estimates. Thus, the γ parameter also plays a key role in controlling the bias-variance tradeoff.

To illustrate, Figure 3.3 plots the bias and variance of the MCP and SCAD estimates as a function of γ under orthonormal design conditions. In general, there is no ideal value of γ in terms of optimizing MSE. Depending on the signal-to-noise ratio, the optimal estimates could be produced by hard thresholding, soft thresholding, or something in between. In this example, MSE is relatively flat as a function of γ. However, γ has a large impact on the bias and variance, with estimates becoming increasingly variable as γ approaches its minimum.

Figure 3.4 revisits the example from Section 3.2.3 to show how the solution paths for MCP change depending on the value of γ. The figure depicts the MCP solution paths for $\gamma \in \{1.5, 2.7, 6\}$. The paths are

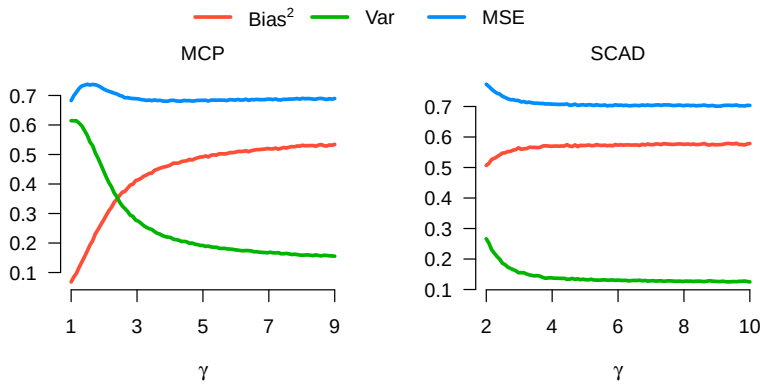

FIGURE 3.3
Bias and variance of MCP and SCAD as a function of γ, with λ held constant at 1. In this example, $\sigma^2 = 6$, $n = 10$, and there is a single feature with $\beta = 1$.

dramatically different from each other. For $\gamma = 1.5$, the MCP paths make a nearly discontinuous jump from zero all the way to their unpenalized solutions; the solution path here is essentially the same as hard thresholding/forward selection. As γ increases, the transition from 0 to unpenalized solution becomes more gradual. As $\gamma \to \infty$; the MCP path becomes equal to the lasso path from Figure 3.2.

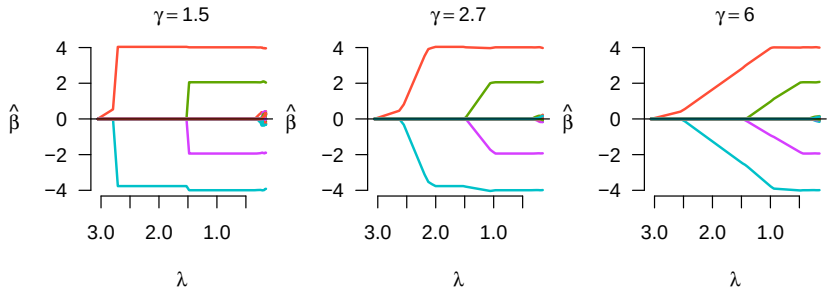

FIGURE 3.4
MCP coefficient paths for simulated example of Section 3.2.3, with $n = 200, p = 1000$ for three values of γ.

3.3 Other nonconvex penalties

Several other nonconvex penalties have been proposed. As with MCP, SCAD, and adaptive lasso, the primary motivation behind these penalties is to reduce the bias towards zero introduced by the lasso penalty. We briefly introduce some of these penalties here in less detail.

The MCP and SCAD penalties are continuously differentiable except at 0. A less smooth penalty is the capped ℓ_1 penalty

$$P(x;\lambda,\gamma) = \min(\gamma\lambda^2/2, \lambda|x|) = \begin{cases} \lambda|x|, & |x| \le \gamma\lambda/2, \\ \gamma\lambda^2/2, & |x| > \gamma\lambda/2, \end{cases} \tag{3.12}$$

where it is required that $\gamma > 1$.. This penalty also modifies the ℓ_1 penalty by truncating it at $x = \gamma\lambda/2$, and it is not differentiable at this point. Its derivative is

$$\dot{P}(x;\lambda,\gamma) = \begin{cases} \lambda, & 0 \le x \le \gamma\lambda/2, \\ 0 & x > \gamma\lambda/2. \end{cases} \tag{3.13}$$

In this expression, the derivative at $x = \gamma\lambda/2$ is the left derivative. Like MCP and SCAD, the γ tuning parameter controls the extent of bias reduction, with the penalty becoming equivalent to the lasso as $\gamma \to \infty$.

Another interesting concave penalty is the bridge penalty

$$P(x;\lambda,\gamma) = \lambda|x|^\gamma, 0 < \gamma < 1. \tag{3.14}$$

Here we require $0 < \gamma < 1$. Its derivative

$$\dot{P}(x;\lambda,\gamma) = \begin{cases} \lambda\gamma|x|^{\gamma-1}\text{sign}(x) & \text{if } x \neq 0, \\ \infty & \text{if } x = 0. \end{cases} \tag{3.15}$$

The bridge penalty has the oldest history of the nonconvex penalties mentioned in this chapter, having been originally proposed in 1993, before the lasso, nearly a decade before SCAD, and over a decade before MCP and the adaptive lasso. However, the derivative of the bridge penalty at 0 is singular, which introduces the unfortunate consequence that $\beta_j = 0$ is always a local minimum of the objective function. This makes the bridge penalized solutions much more difficult to obtain computationally, especially in high-dimensional models.

3.4 Bayesian connection

As discussed in Section 2.1, the Laplace prior corresponding to the ℓ_1 penalty can be constructed from scale mixtures of normal distributions and exponential distributions. For the MCP $P(x;\lambda,\gamma)$, because it is constant for $|x|>\gamma\lambda$, the corresponding prior $\exp(-P(x;\lambda,\gamma)$ is improper. However, it turns out that the MCP can also be constructed from scale mixtures of normal distributions, based on a representation of the MCP as the Moreau envelope of a simple quadratic function.

Let $\lambda>0$ and $\gamma>0$. Then,

$$\begin{cases}\lambda\int_0^{|x|}(1-\frac{t}{\gamma\lambda})_+dt=\min_{\tau\geq 0}\left\{\tau|x|+\frac{\gamma}{2}(\tau-\lambda)^2\right\}\\ \lambda(1-\frac{|x|}{\gamma\lambda})_+=\arg\min_{\tau\geq 0}\left\{\tau|x|+\frac{\gamma}{2}(\tau-\lambda)^2\right\}\end{cases}\tag{3.16}$$

Based on (3.16), we can formulated the MCP penalized solution as a MAP solution as follows. Suppose $z\sim \mathrm{N}(\theta,1)$. Consider the priors

$$p(\theta|\tau,\gamma,\lambda)\propto\tau\exp(-\tau|\theta|), p(\tau|\gamma,\lambda)\propto\tau^{-1}\exp\{-\gamma(\tau-\lambda)^2/2\}1_{\{\tau>0\}}.$$

The posterior

$$p(\theta,\tau|z;\gamma,\lambda)\propto\exp(-\frac{1}{2}(z-\theta)^2)\exp(-\tau|\theta|)\exp\{-\gamma(\tau-\lambda)^2/2\}1_{\{\tau>0\}}.$$

For any given $\gamma>0$, the resulting MAP estimator for (θ,τ) is

$$(\hat\theta,\hat\tau)=\underset{\theta\in\mathbb{R},\tau\in\mathbb{R}^+}{\arg\min}\ \frac{1}{2}|z-\theta|^2+\tau|\theta|+\frac{\gamma}{2}(\tau-\lambda)^2.$$

Therefore, by (3.16), the unique solution is the MCP threshold estimator, that is,

$$\begin{cases}\hat\theta=\arg\min_\theta\frac{1}{2}|z-\theta|^2+\lambda\int_0^{|\theta|}(1-\frac{t}{\gamma\lambda})_+dt,\\ \hat\tau=\lambda(1-|\hat\theta|/(\gamma\lambda))_+.\end{cases}$$

A direct consequence of (3.16) is

$$\begin{cases}\lambda\sum_{j=1}^p\int_0^{|x_j|}\left(1-\frac{t}{\gamma\lambda}\right)dt=\min_{\boldsymbol\tau\in\mathbb{R}^p_+}\sum_{j=1}^p\left(|x_j|\tau_j+\frac{\gamma}{2}(\tau_j-\lambda)^2\right)\\ \lambda(1-\frac{|x_j|}{\gamma\lambda})_+=\arg\min_{\tau\geq 0}\left\{\tau|x_j|+\frac{\gamma}{2}(\tau-\lambda)^2\right\}, j=1,\ldots,p.\end{cases}\tag{3.17}$$

Therefore, the MCP solution can be formulated as a MCP solution in the following way. Let

$$(\hat{\boldsymbol\theta},\hat{\boldsymbol\tau})=\underset{\mathbf{b}\in\mathbb{R}^p,\boldsymbol\tau\in\mathbb{R}^p_+}{\arg\min}\left\{\frac{1}{2n}\|\mathbf{y}-\mathbf{X}\mathbf{b}\|^2+\sum_{j=1}^p\{\tau_j|b_j|+\frac{\gamma}{2}(\tau_j-\lambda)^2\}\right\}.\tag{3.18}$$

Then

$$\begin{cases} \hat{\boldsymbol{\theta}} = \arg\min_{\mathbf{b}\in\mathbb{R}^p} \{ \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \sum_{j=1}^p P_{\mathrm{MCP}}(|b_j|; \lambda, \gamma), \\ \hat{\tau}_j = \lambda(1 - \frac{|\hat{\theta}_j|}{\gamma\lambda})_+, j = 1, \ldots, p. \end{cases}$$

So the MCP solution can be obtained by solving (3.18). It is interesting to note that the objective function in (3.18) is convex in $(\boldsymbol{\theta}, \boldsymbol{\tau})$.

The expression (3.18) naturally leads to an iterative algorithm for computing the MCP solutions. Let $\widehat{\boldsymbol{\beta}}^0(\lambda) = \widetilde{\boldsymbol{\beta}}(\lambda)$ be an initial estimator, for example, we can take $\widetilde{\boldsymbol{\beta}}(\lambda)$ to be the lasso solution. Then, for $s = 1, 2, \ldots$, the iteration proceeds as follows,

$$\begin{cases} \hat{\tau}^s(\lambda) = \lambda(1 - |\widehat{\beta}_j^{s-1}|/(\gamma\lambda))_+, \ j = 1, \ldots, p, \\ \widehat{\boldsymbol{\beta}}^s(\lambda) = \arg\min_{\mathbf{b}\in\mathbb{R}^p} \left\{ \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \sum_{j=1}^p \tau_j^s(\lambda)|b_j| \right\}. \end{cases} \tag{3.19}$$

Interestingly, this is the same as the algorithm based on the local linear approximation discussed below.

Since the objective function in (3.18) is convex, this expression also opens up the possibility of designing efficient algorithms for obtaining global solutions to MCP penalized regression.

3.5 Algorithms

3.5.1 Coordinate descent

The coordinate descent algorithm described for the lasso in Chapter 2.4 can be modified for use with objective functions containing nonconvex penalties. We describe these modifications here for the MCP and SCAD penalties, but the idea is broadly applicable.

Given the current value $\widetilde{\beta}^{(s)}$ in the sth iteration for $s = 0, 1 \ldots$, the algorithm for computing $\widehat{\boldsymbol{\beta}}$ is given in Algorithm 3.1.

The algorithm is identical to Algorithm 2.1 except for the step in which $\widetilde{\beta}_j$ is updated. Although the MCP and SCAD penalties are not convex functions, the objective function itself *is* convex with respect to any individual coordinate, as we state in the following lemma. As a result, the coordinate-wise updates are unique and always occur at the global minimum with respect to that coordinate.

Lemma 3.1. *Let $Q_j(\beta_j|\boldsymbol{\beta}_{-j})$ denote the objective function Q defined in* (3.4) *as a function of the single variable β_j, with the remaining elements of $\boldsymbol{\beta}$ fixed. For the SCAD penalty with $\gamma > 2$ and for the MCP with $\gamma > 1$, $Q_j(\beta_j|\boldsymbol{\beta}_{-j})$ is a convex function of β_j for all j.*

Algorithm 3.1 Coordinate descent algorithm for MCP/SCAD

repeat
  for $j = 1, 2, \ldots, p$
    $\tilde{z}_j = n^{-1} \sum_{i=1}^{n} x_{ij} r_i + \widetilde{\beta}_j^{(s)}$
    $\widetilde{\beta}_j^{(s+1)} \leftarrow \begin{cases} F(\tilde{z}_j | \lambda, \gamma) & \text{for MCP, or} \\ T_{\text{SCAD}}(\tilde{z}_j | \lambda, \gamma) & \text{for SCAD} \end{cases}$
    $r_i \leftarrow r_i - (\widetilde{\beta}_j^{(s+1)} - \widetilde{\beta}_j^{(s)}) x_{ij}$ for all i
until convergence

From this lemma, we can establish the following convergence properties of coordinate descent algorithms for SCAD and MCP. The technical details involved in proving these properties are given in Section 3.8.

Proposition 3.1. *Let* $\{\boldsymbol{\beta}^{(s)}\}$ *denote the sequence of coefficients produced at each iteration of the coordinate descent algorithms for SCAD and MCP. For all* $s = 0, 1, 2, \ldots$,

$$Q(\boldsymbol{\beta}^{(s+1)}) \leq Q(\boldsymbol{\beta}^{(s)}).$$

Furthermore, the sequence is guaranteed to converge to a local minimum of $Q(\boldsymbol{\beta})$.

While the objective functions for SCAD and MCP are convex in each coordinate dimension, they are not convex in general. Thus, multiple minima may exist, each satisfying the KKT conditions. The coordinate descent algorithm introduced here is not guaranteed to converge to the global minimum in such cases; neither are the local approximation algorithms of the next section. The issues of convexity and multiple solutions are discussed in greater detail, including a concrete example, in Section 3.6.

3.5.2 Local approximations

For MCP and SCAD, one can obtain closed-form coordinate-wise minima and use those solutions as updates. An alternative approach, which is particularly useful in penalties that do not yield tidy closed-form solutions, is to construct a local approximation of the penalty. One can construct quadratic approximations (the *local quadratic approximation*, or LQA, algorithm) so that the optimization problem resembles ridge regression and equation (1.18) can be used, or linear approximations (the *local linear approximation*, or LLA, algorithm) so that the optimization problem resembles the lasso and either the LARS algorithm or

coordinate descent methods from Chapter 2 can be used. In general, for sparsity-inducing penalties the LLA algorithm is more efficient, as LQA is unable to take advantage of sparsity.

The idea behind the LLA algorithm is to construct a linear approximation to the penalty in (3.4):

$$P(|x|) \approx P(|x_0|) + \dot{P}(|x_0|)(|x| - |x_0|).$$

Note that with this approximation, the penalty takes on the form of the lasso penalty (with $\dot{P}(|x_0|)$ playing the role of the regularization parameter) plus a constant. The approximation is applied in an iterative fashion. Thus, at the sth iteration, letting $\tilde{\lambda}_j = \dot{P}(|\beta_j^{(s-1)}|)$, the update is given by

$$\boldsymbol{\beta}^{(s)} = \underset{\mathbf{b}\in\mathbb{R}^p}{\arg\min} \left\{ \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \sum_{j=1}^{p} \tilde{\lambda}_j |b_j| \right\}. \tag{3.20}$$

It is worth noting the similarity between LLA and the adaptive lasso – equations (3.1) and (3.20) are nearly identical. However, the adaptive lasso weights are assigned in a more or less ad hoc fashion based on an initial estimator, while the LLA modifications to λ are explicitly determined by the penalty function P.

Like coordinate descent, LLA is guaranteed to drive the objective function downhill with every iteration and to converge to a local minimum of $Q(\boldsymbol{\beta})$. For MCP, the LLA leads to the same iteration algorithm given in (3.19).

Local quadratic approximation

To find the value of $\boldsymbol{\beta}$ that optimizes (1.14), the local quadratic approximation (LQA) algorithm approximates the penalty by a quadratic function. For $x \approx x_0$,

$$\dot{P}(|x|) = P'(|x|)\text{sign}(x) \approx \{\dot{P}(|x_0|)/|x_0|\}x.$$

Therefore,

$$P(|x|) \approx P(|x_0|) + \frac{1}{2}\{\dot{P}(x_0)/|x_0|\}(x^2 - x_0^2).$$

Let $\boldsymbol{\beta}^{(0)}$ be an initial solution. Then for $k = 1, 2, \ldots$, the iterative solution $\boldsymbol{\beta}^{(k)}$ is the argument minimizing

$$\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 + \frac{1}{2}\sum_{j=1}^{p} \frac{\dot{P}(\beta_j^{(k-1)}; \lambda)}{|\beta_j^{(k-1)}|} b_j^2$$

This minimization problem is an adaptively weighted ridge regression with the weights determined by the penalty function and the previous solution. Let $w(\beta;\lambda) = \dot{P}(\beta;\lambda)/|\beta|$, and let $\mathbf{W}^{(k)} = \text{diag}(w(\beta_j^{(k)};\lambda), j = 1,\ldots,p)$. Then

$$\boldsymbol{\beta}^{(k)} = \underset{\mathbf{b}\in\mathbb{R}^p}{\arg\min}\left\{\frac{1}{2n}\|\mathbf{y}-\mathbf{X}\mathbf{b}\|^2 + \frac{1}{2}\mathbf{b}^T\mathbf{W}^{(k-1)}\mathbf{b}\right\}, k = 1,2,\ldots.$$

The explicit solution is

$$\boldsymbol{\beta}^{(k)} = (n^{-1}\mathbf{X}^T\mathbf{X} + \mathbf{W}^{(k-1)})^{-1}\mathbf{X}^T\mathbf{y}/n, k = 1,2,\ldots.$$

The main computational task is the calculation of the inverse of the $p \times p$ matrix of the form $n^{-1}\mathbf{X}^T\mathbf{X} + \mathbf{W}$. In $p \gg n$ models, this can be calculated using

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

However, the LQA algorithm does not produce sparse solutions. It is necessary to threshold small coefficients to obtain sparse solutions.

3.6 Global and local convexity

As noted in Section 3.5, models with nonconvex penalties may possess multiple minima to their objective functions. This is problematic for two reasons. First, neither coordinate descent nor the LLA algorithm are guaranteed to converge to the global minimum; if multiple minima are present, we may obtain an inferior solution as our estimate. Second, estimation will no longer be continuous, in the sense that small changes to the data or to the regularization parameter can lead to large changes in the estimate, as our solution "jumps" from one minima to another. In this section, we discuss these issues in greater detail.

We begin by noting that it is possible for the objective function Q to be convex with respect to $\boldsymbol{\beta}$ even though the penalty component is nonconvex. Letting $c_{\min}$ denote the minimum eigenvalue of $\mathbf{X}^T\mathbf{X}/n$, the MCP objective function is strictly convex if $\gamma > 1/c_{\min}$, while the SCAD objective function is strictly convex if $\gamma > 1+1/c_{\min}$. In this case, the coordinate descent and LLA algorithms will converge to the unique global minimum of Q. Thus, at least for MCP and SCAD, provided that $\mathbf{X}$ is full rank it is always possible to choose γ such that the objective function is convex.

However, obtaining strict convexity is not always possible or desirable. For example, in high-dimensional settings where $p > n$, $c_{\min}$ will always be zero and a strictly convex objective function not possible unless the penalty is strictly convex. Nevertheless, it is not true in general that convex penalties outperform nonconvex ones in such scenarios, as the following example demonstrates.

Example 3.1. For this example, we will set $n = 50$ and $p = 100$. All features $\mathbf{x}_j$ will follow standard Gaussian distributions and be independent of each other. Note that in this construction, $\mathbf{X}$ cannot be full rank and the MCP and SCAD objective functions will not be convex. In the generating model, we set $\beta_1 = \beta_2 = \beta_3 = \cdots = \beta_6 \neq 0$ and $\beta_7 = \beta_8 = \cdots = \beta_{100} = 0$. The nonzero values of β_1 through β_6 were varied to produce a range of signal to noise ratios. For this problem,

$$\begin{aligned}\mathrm{Var}(Y) &= \mathrm{Var}(\mathrm{E}(Y|X)) + \mathrm{E}(\mathrm{Var}(Y|X)) \\ &= \boldsymbol{\beta}^T \mathrm{Var}(X)\boldsymbol{\beta} + \sigma^2 \\ &= \boldsymbol{\beta}^T\boldsymbol{\beta} + \sigma^2.\end{aligned}$$

The first term in the sum is known as the *signal* and the second term the *noise*. For each data set, an independent data set of equal size was generated for the purposes of selecting the regularization parameter. The estimates presented are those for the value of λ that minimized the prediction error on the validation data set.

Figure 3.5 presents the mean squared error (i.e, average value of $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2$) for this simulation across various SNR settings and for various γ values for MCP and SCAD.

As the figure indicates, for low signal-to-noise ratios there is indeed some benefit to increasing γ in an effort to bring the objective function closer to convexity. However, for larger SNR values, this strategy diminishes estimation accuracy, roughly doubling the MSE as γ increases from its minimum to maximum values over the given ranges. □

Example 3.1 demonstrates that it is possible for the solution to a nonconvex penalized regression model to outperform the solution from a convex model. One reason this happens is that the solutions are sparse: although $Q(\boldsymbol{\beta})$ may not be convex over the entire p-dimensional parameter space (i.e., *globally convex*), it is still convex on many lower-dimensional spaces. If these lower-dimensional spaces contain the solution of interest, then the existence of other local minima in much higher dimensions may not be relevant. We refer to this concept as *local convexity*.

Recall the conditions for global convexity: γ must be greater than $1/c_*$ for MCP and $1 + 1/c_*$ for SCAD, where c_* denoted the minimum eigenvalue of $\mathbf{X}^T\mathbf{X}/n$. A straightforward modification is to include only

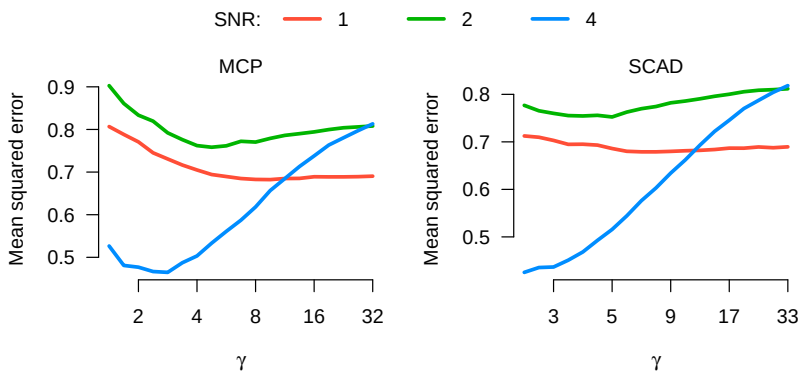

FIGURE 3.5
MSE for MCP and SCAD at various γ values across a range of SNR levels, as described in Example 3.1.

the covariates with nonzero coefficients (the covariates which are "active" in the model) in the calculation of c_*. Note that neither γ nor $\mathbf{X}$ change with λ. What does vary with λ is the set of active covariates; generally speaking, this will increase as λ decreases (with correlated/collinear data, however, exceptions are possible). Thus, local convexity of the objective function will not be an issue for large λ, but may cease to hold as λ is lowered past some critical value λ^*.

Specifically, let $\widehat{\boldsymbol{\beta}}(\lambda)$ denote the minimizer of (3.4) for a certain value of λ, $A(\lambda) = \{j : \widehat{\beta}_j(\lambda) \neq 0\}$ denote the active set of covariates, $U(\lambda) = A(\lambda) \cup A(\lambda^-)$ denote the set of covariates that are either currently active given a value λ or that will become active imminently upon the lowering of λ by an infinitesimal amount, and let $\mathbf{X}_{U(\lambda)}$ denote the design matrix formed from only those covariates for which $j \in U(\lambda)$, with $c_*(\lambda)$ denoting the minimum eigenvalue of $\mathbf{X}_{U(\lambda)}^T \mathbf{X}_{U(\lambda)}/n$ Now, let

$$\lambda^* = \inf\{\lambda : \gamma > 1/c_*(\lambda)\} \text{ for MCP}$$

and

$$\lambda^* = \inf\{\lambda : \gamma > 1 + 1/c_*(\lambda)\} \text{ for SCAD.}$$

We may then say that the objective function is locally convex over the interval $\lambda \in (\infty, \lambda^*)$. Because $c_*(\lambda)$ changes only when the composition of $U(\lambda)$ changes, λ^* must be a value of λ for which $A(\lambda) \neq A(\lambda^-)$.

Note that λ^* does not involve any unknown parameters, and therefore can be computed and used in practice as a useful diagnostic to

indicate which regions of the solution paths produce stable, well-defined estimates and which regions may suffer from multiple local minima and discontinuous paths.

The practical benefit of these diagnostics can be seen in Figure 3.6, which depicts an example of an MCP coefficient path from simulated data in which $n = 20$ and $p = 50$. As is readily apparent, solutions are smooth and well behaved in the unshaded, locally convex region, but suffer from an abrupt, discontinuous jump at λ^*. The region of the coefficient path that is not locally convex is shaded by default when the `plot.ncvreg` function is used with nonconvex penalties, but can be shut off using `shade=FALSE`.

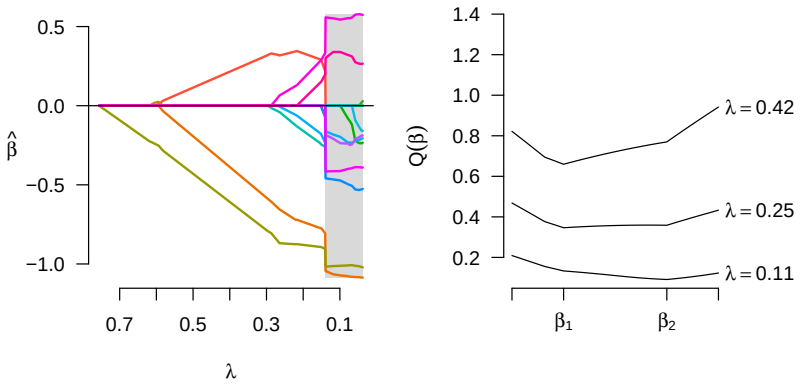

FIGURE 3.6
Left: An example MCP coefficient path for simulated data where $p > n$. The shaded region is the region in which the objective function is not locally convex. Right: Values of the objective function along the line segment joining the solutions on either side of λ^*.

The right side of Figure 3.6 provides a more detailed look at the discontinuous transition occurring at λ^*. It plots the objective function along the line segment joining $\boldsymbol{\beta}_1$, the solution just to the left of $\lambda^* *$ and $\boldsymbol{\beta}_2$, the solution just to the right of λ^*. When $\lambda = 0.42$, $\boldsymbol{\beta}_1$ clearly minimizes the objective function and when $\lambda = 0.11$, $\boldsymbol{\beta}_2$ clearly minimizes the objective function, but for $\lambda \approx 0.25$, the objective function is very broad and flat, indicating substantial uncertainty about which solution is preferable.

In conclusion, the convexity of the objective function and the possibility of multiple local minima is certainly an important issue to be

aware of and to monitor in practice with local convexity diagnostics. However, although multiple minima are possible, they are not always a relevant concern for models with nonconvex penalties. In particular, as Figure 3.6 illustrates, there is typically a range of λ values over which optimization is well-behaved. Furthermore, nonconvex penalties often outperform convex penalties, especially when the signal-to-noise ratio is high. In Chapter 4, we return to this and discuss another method for improving the stability/convexity of a model besides adjusting γ.

3.7 Case study: Breast cancer gene expression study (revisited)

In this section, we revisit the BRCA1 gene expression study originally discussed in Section 2.7. First, we consider an adaptive lasso model. Recall that the adaptive lasso requires an initial estimator. Here, we use the lasso estimates with λ chosen according to BIC:

```
fit <- ncvreg(X, y, penalty='lasso')
b <- coef(fit, which=which.min(BIC(fit)))[-1]
```

In the above, we used `ncvreg` for fitting due to its compatibility with `R`'s BIC built-in function. Selecting λ according to cross-validation would be another reasonable way of choosing an initial estimator. Once we have the initial estimator, we can fit an adaptive lasso model as follows:

```
w <- abs(b)^(-1)    # Calculate weights
w <- pmin(w, 1e10)  # cv.glmnet does not allow infinite weights
cvfit <- cv.glmnet(X, y, penalty.factor=w)
```

Figure 3.7 illustrates the output of `plot(cvfit)` for the adaptive lasso. The cross-validation procedure indicates that a model containing 20 genes (out of the initial 17,322) is optimal. It should be noted that the above application of cross-validation, while reasonable for the selection of λ, does not estimate the cross-validation error in an unbiased manner. The reason is that the left-out fold is not truly external to the fitting procedure, as it was used to obtain an initial estimator. As a result, the estimate of out-of-sample prediction error is biased.

For example, Figure 3.7 indicates that the adaptive lasso attains a minimum CV error of 0.18; in Section 2.7, we saw that the lasso had a minimum CV error of 0.20 (Figure 2.10). However, it does not follow from these results that the adaptive lasso is necessarily outperforming

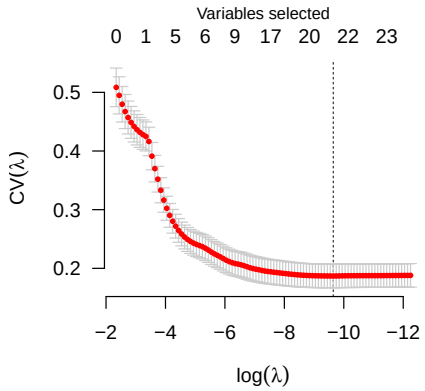

FIGURE 3.7
Cross-validation for the adaptive lasso, applied to the BRCA1 gene expression study, with (BIC) lasso as the initial estimator. As discussed in the text, the estimates of cross-validation error are slightly over-optimistic for the adaptive lasso unless the initial estimate is cross-validated as well.

the lasso here, as the decrease could be due to bias. To obtain an (approximately) unbiased estimate of CV error, one must cross-validate the entire procedure, including the initial estimate (the `hdrm` package provides a function, `cv.adaptive_lasso` for this). In doing so, the CV error for adaptive lasso increases to 0.23, indicating that, if anything, it performs slightly worse than the lasso in this example in terms of prediction accuracy. Unfortunately, while existing software packages can be used to fit adaptive lasso models, there are not currently any comprehensive software packages for the adaptive lasso (that we are aware of) that carry out full cross-validation.

The other methods discussed in this Chapter, MCP and SCAD, achieve the adaptive lasso's goal of reducing the bias associated with the lasso, but do so in a single procedure – as opposed to the adaptive lasso's two-step procedure – and thus prove more amenable to carrying out inference concerning predictive accuracy using cross-validation. Figure 3.8 presents two MCP-penalized regression models fit to the BRCA1 data, one with $\gamma = 3$ and the other with $\gamma = 7$. The models were fit with `ncvreg` using the following commands:

```
cvfit3 <- cv.ncvreg(X, y) # gamma=3 is default
cvfit7 <- cv.ncvreg(X, y, gamma=7)
```

and the plots made with

```
plot(cvfit3$fit, log.l=TRUE)
plot(cvfit3) # And so on for cvfit7
```

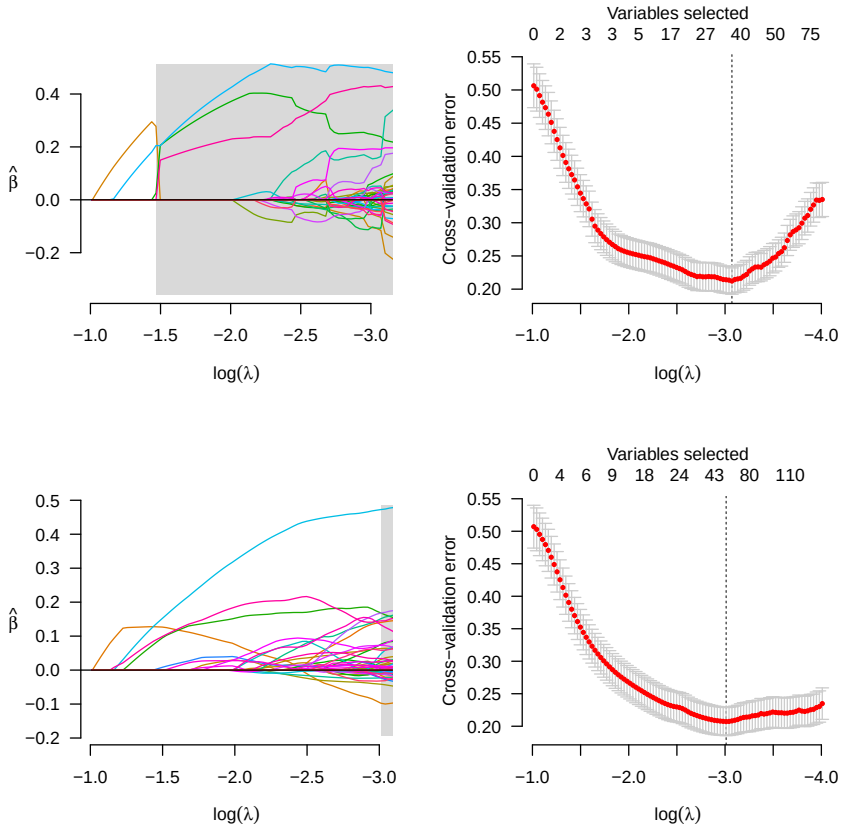

FIGURE 3.8
MCP models fit to the BRCA1 gene expression study. Top: $\gamma = 3$. Bottom: $\gamma = 7$.

As the figure indicates, the cross-validation error is minimized for both models at $\lambda \approx e^{-3}$. Also, for both models, the minimum error is $CV = 0.21$; very close to, although slightly larger than the $CV = 0.20$ achieved by the lasso. However, the two models select very different numbers of variables, both compared to each other and compared to the lasso, which selected 96 nonzero coefficients. This basic summary information can be displayed using the `summary` function:

```
> summary(cvfit3)
MCP-penalized linear regression with n=536, p=17322
At minimum cross-validation error (lambda=0.0464):
-------------------------------------------------
  Nonzero coefficients: 38
  Cross-validation error (deviance): 0.21
  R-squared: 0.58
  Signal-to-noise ratio: 1.39
  Scale estimate (sigma): 0.461
> summary(cvfit7)
MCP-penalized linear regression with n=536, p=17322
At minimum cross-validation error (lambda=0.0492):
-------------------------------------------------
  Nonzero coefficients: 52
  Cross-validation error (deviance): 0.21
  R-squared: 0.59
  Signal-to-noise ratio: 1.45
  Scale estimate (sigma): 0.455
```

The most striking difference between the two solution paths is that for MCP with $\gamma = 3$, the the optimal solution occurs in the region that is not locally convex. Indeed, when $\gamma = 3$, the objective function encounters problems with nonconvexity quite early in the solution path, making a discontinuous transition between local minima at $\lambda \approx e^{-1.5}$. On the other hand, while the objective function for MCP with $\gamma = 7$ eventually becomes locally non-convex, this is not until after the optimal solution has been reached. As this is real data, there is no gold standard for determining which of these two solutions is superior. In this example the authors prefer the $\gamma = 7$ solution, as we are somewhat concerned about the numerical stability of the $\gamma = 3$ solution; we would not necessarily expect everyone to agree with us on this point, however.

Finally, let us fit a SCAD-penalized regression model to this data using the following code; similar to the MCP case, we set $\gamma = 8$ here to increase the stability of the solution path:

```
> cvfit <- cv.ncvreg(X, y, gamma=8, penalty='SCAD')
> summary(cvfit)
SCAD-penalized linear regression with n=536, p=17322
At minimum cross-validation error (lambda=0.0478):
-------------------------------------------------
  Nonzero coefficients: 79
  Cross-validation error (deviance): 0.20
  R-squared: 0.61
  Signal-to-noise ratio: 1.53
  Scale estimate (sigma): 0.447
```

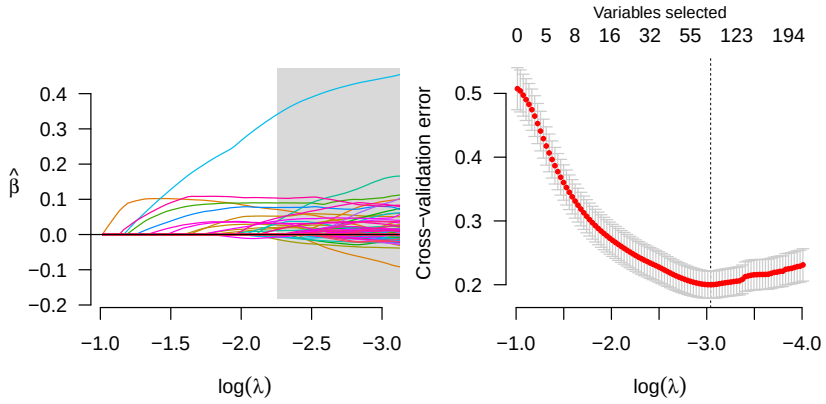

FIGURE 3.9
SCAD-penalized model fit to the BRCA1 gene expression study with $\gamma = 8$.

The SCAD solution path is more lasso-like than that of the MCP models, as one would expect from the fact that the SCAD and lasso penalties are more similar. Not only are the solution paths in Figure 3.9 and Figure 2.10 visually similar, but they achieve the same CV error ($CV = 0.20$) and the sparsity of the SCAD model (79 nonzero coefficients) is closer to lasso (96 nonzero coefficients) than it is to MCP (52 nonzero coefficients).

The results seen here are fairly representative, in our experience, of what one sees when applying lasso, MCP, and SCAD models to real data: prediction performance (as estimated by cross-validation) is typically similar, but there can be substantial differences in terms of the estimates themselves. Thus, if one cares solely about predictive accuracy, nonconvex methods do not typically offer significant advantages. However, if one is concerned with finding a highly sparse model that still offers optimal or nearly optimal predictive accuracy, or if estimation bias for the selected coefficients is an overriding concern, MCP and SCAD are attractive alternatives to the lasso.

3.8 *Convergence of coordinate descent algorithms

Consider the minimization problem

$$\min\left\{f(\mathbf{x}_1,\ldots,\mathbf{x}_p)=f_0(\mathbf{x}_1,\ldots,\mathbf{x}_p)+\sum_{j=1}^{p}f_j(\mathbf{x}_j)\right\} \tag{3.21}$$

for some $f_0:\mathbb{R}^{m_1+\cdots m_p}\mapsto\mathbb{R}\cup\{\infty\}$ and $f_j:\mathbb{R}^{m_j}\mapsto\mathbb{R}\cup\{\infty\}, j=1,\ldots,p$. Here $m_j\geq 1$ are integers. Let $m=m_1+\cdots+m_p$. Clearly, the lasso and concave penalized criterions for linear regression discussed above are special cases of (3.21) with f_0 being the least squares loss function, f_j the penalty function, and $m_1=\cdots m_p=1$. The formulation (3.21) also includes the group selection methods described in later chapters.

For any function h that maps $\mathbb{R}^m$ into $\mathbb{R}\cup\{\infty\}$, denote its effective domain by

$$\mathrm{dom}h=\{\mathbf{x}:h(\mathbf{x})<\infty\}.$$

A general *blockwise coordinate descent* algorithm for finding a local solution to (3.21) is as follows.

- *Initialization:* Choose any $\mathbf{x}^0=(\mathbf{x}_1^0,\ldots,\mathbf{x}_p^0)\in\mathrm{dom}f$.
- *Iteration step $s+1$:* Given $\mathbf{x}^s=(\mathbf{x}_1^s,\ldots,\mathbf{x}_p^s)\in\mathrm{dom}f$, choose an index j and compute a new iterate

  $$\mathbf{x}^{s+1}=(\mathbf{x}_1^{s+1},\ldots,\mathbf{x}_p^{s+1})$$

  with

  $$\mathbf{x}_j^{s+1}\in\arg\min_{\mathbf{x}_j}f(\mathbf{x}_1^s,\ldots,\mathbf{x}_{j-1}^s,\mathbf{x}_j,\mathbf{x}_{j+1}^s,\ldots,\mathbf{x}_p^s),$$

  $$\mathbf{x}_k^{s+1}=\mathbf{x}_j^s, k\neq j.$$

To ensure convergence, each coordinate block needs to be visited sufficiently often. A general rule is the essential cyclic rule.

Essential cyclic rule. There exists a constant $T\geq p$ such that every index $j\in\{1,\ldots,p\}$ is chosen at least once between the sth iteration and the $(s+T-1)$th iteration, for all s.

An important special case is the cyclic rule.

Cyclic rule. $T=p$. Choose $j=k$ at iterations $k,k+p,k+2p,\ldots$, for $k=1,\ldots,p$. This is what is implemented in the `R` packages `glmnet` and `ncvreg`.

A set of sufficient conditions are given below that guarantee that

the sequence of solutions from (blockwise) coordinate descent algorithm converges to a coordinatewise minimum point or a local minimum point.

A point $\mathbf{z} \in \mathbb{R}^m$ is a coordinatewise minimum point of $h : \mathbb{R}^m \mapsto \mathbb{R}$ if $\mathbf{z} \in \text{dom}h$ and

$$h(\mathbf{z} + (0, \ldots, \mathbf{d}_j, \ldots, 0)) \geq h(\mathbf{z}), \text{ for every } \mathbf{d}_j \in \mathbb{R}^{m_j}, j = 1, \ldots, p.$$

It is a local minimum (stationary point) of h if $\mathbf{z} \in \text{dom}h$ and

$$h'(\mathbf{z}; \mathbf{d}) \geq 0, \text{ for every } \mathbf{d},$$

where $h'(\mathbf{z}; \mathbf{d})$ is the lower directional derivative of h at $\mathbf{z}$ in the direction $\mathbf{d}$ defined as

$$h'(\mathbf{x}; \mathbf{d}) = \liminf_{t \downarrow 0} \frac{h(\mathbf{x} + t\mathbf{d}) - h(\mathbf{x})}{t}.$$

The following definitions will be useful in stating the convergence property of the blockwise coordinate descent algorithm.

- h is lower semi-continuous (lsc) at $\mathbf{x}_0$ if

  $$\liminf_{\mathbf{x} \to \mathbf{x}_0} h(\mathbf{x}) \geq h(\mathbf{x}_0).$$

  For example, if A is an open set, then the indicator function $1\{\mathbf{x} \in A\}$ is lsc.

- A function h is hemivariate if h is not constant on any line segment in dom$h$. The SCAD penalty and MCP are not hemivariate, since they are constant at the tails.

- h is quasiconvex if

  $$h(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) \leq \max\{h(\mathbf{x}), h(\mathbf{y})\}.$$

  Clearly, any convex function is quasiconvex. More generally, a univariate function $f(x)$ for which there exists a $x_0 \in \mathbb{R} \cup \{\infty\}$ such that f decreases on $(-\infty, x_0]$ and increases on $[x_0, \infty)$ is quasiconvex. Important examples of such functions include SCAD penalty and MCP.

Theorem 3.1. *Consider the minimization problem* (3.21). *Suppose*

(B1) f_0 is continuous on domf_0;

(B2) The function $\mathbf{x}_j \mapsto f(\cdot, \mathbf{x}_j, \cdots)$ is quasiconvex and hemivariate for $j = 1, \ldots, p$;

(B3) $f_1, \ldots, f_p$ are lsc;

(B4) $\{\mathbf{x} : f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$ *is bounded, where* $\mathbf{x}_0$ *is the initial value that starts the BCD algorithm.*

Also assume one of the following conditions on $domf_0$*:*

(C1) $domf_0$ *is open and* f_0 *tends to* ∞ *at boundary of* $domf_0$*;*

(C2) $domf_0 = Y_1 \times \cdots \times Y_p$ *for some* $Y_j \subset \mathbb{R}^{p_j}, j = 1, \ldots, p$.

Then, the sequence $\{\mathbf{x}^s\}$ *generated by the BCD method using the essentially cyclic rule is defined, bounded, and every cluster point is a coordinatewise minimum point of* f.

A function h is regular at $\mathbf{z} \in \text{dom}h$ if for any $\mathbf{d} = (\mathbf{d}_1, \ldots, \mathbf{d}_p)$ such that

$$h'(\mathbf{z}; (0, \ldots, \mathbf{d}_j, \ldots, 0)) \geq 0, j = 1, \ldots, p,$$

it holds that

$$h'(\mathbf{z}; \mathbf{d}) \geq 0.$$

It can be seen that any differentiable function h is regular since

$$h'(\mathbf{x}; \mathbf{d}) = \nabla h(\mathbf{x})^T \mathbf{d} = \sum_{j=1}^{p} \frac{\partial h}{\partial \mathbf{x}_j} \mathbf{d}_j = \sum_{j=1}^{p} h'(\mathbf{x}; (0, \ldots, \mathbf{d}_j, \ldots, 0)).$$

For a regular function, a coordinate-wise minimum point is also a local minimum point.

Theorem 3.2. *Suppose that* $domf_0$ *is open and* f_0 *is differentiable on* $domf_0$*. Then* f *in* (3.21) *is regular at every* $\mathbf{x} \in domf$*. Therefore, if the conditions of Theorem 3.1 also hold, then a coordinate-wise minimum point is also a local minimum point.*

Proof. If $\mathbf{x} \in \text{dom}f$, then $\mathbf{z} \in \text{dom}f_0$. For any $\mathbf{d} = (\mathbf{d}_1, \ldots, \mathbf{d}_p)$, suppose

$$f'(\mathbf{x}; (0, \ldots, \mathbf{d}_j, \ldots, 0)) \geq 0.$$

Then

$$\begin{aligned}
f'(\mathbf{z}; \mathbf{d}) &= \nabla f_0(\mathbf{x})^T \mathbf{d} + \liminf_{t \downarrow 0} \sum_{j=1}^{p} [f_j(\mathbf{x}_j + t\mathbf{d}_j) - f_j(\mathbf{x}_j)]/t \\
&= \nabla f_0(\mathbf{x})^T \mathbf{d} + \sum_{j=1}^{p} \liminf_{t \downarrow 0} [f_j(\mathbf{x}_j + t\mathbf{d}_j) - f_j(\mathbf{x}_j)]/t \\
&= \nabla f_0(\mathbf{x})^T \mathbf{d} + \sum_{j=1}^{p} f_j'(\mathbf{x}_j; \mathbf{d}_j) \\
&= \sum_{j=1}^{p} f'(\mathbf{x}; (0, \ldots, \mathbf{d}_j, \ldots, 0)) \\
&\geq 0.
\end{aligned}$$

Thus f is regular. The second conclusion now follows from Theorem 3.1. □

Bibliographical notes

This section will include the bibliographical notes on the materials presented in this chapter.

Exercises

3.1. *Two-stage adaptive lasso.* In Figure 3.2, solution paths for a simulated example were given for the adaptive lasso using the pathwise approach to selecting weights. Construct the corresponding figure for the two-stage adaptive lasso and comment on how it differs from both the pathwise adaptive lasso and the original lasso paths. Describe your approach and your initial estimator.

3.2. *Thresholding for MCP and SCAD.*

(a) Show that in the orthonormal case, the MCP estimates are given by (3.9).

(b) Show that in the orthonormal case, the SCAD estimates are given by (3.10).

3.3. *Degrees of freedom for MCP (orthonormal case).*

(a) Using Stein's lemma (see Exercise 2.9), derive the degrees of freedom for MCP in the setting where the features are orthonormal.

(b) Suppose $\gamma = 3$; comment on the degrees of freedom for MCP compared to the lasso.

3.4. *Sparsity and the bridge penalty.* Consider the objective function for the bridge penalty in the orthonormal/univariate setting:

$$Q(\beta) = \frac{1}{2}|z - \beta|^2 + \lambda|\beta|^\gamma, 0 < \gamma < 1.$$

You may wish to plot this objective function in order to better understand this problem.

(a) Show that for all values of λ, $\beta = 0$ is a local minimum.

(b) Show that Q is minimized at $\beta = 0$ if and only if

$$|z| < \frac{2-\gamma}{(2-2\gamma)^{(1-\gamma)/(2-\gamma)}}\lambda^{1/(2-\gamma)}.$$

3.5. *MCP as scale mixture of normal distributions.* Prove (3.16).

3.6. *Convexity of MCP and SCAD objective functions*

(a) Show that the objective function for MCP is strictly convex if $\gamma > 1/c_{\min}$.

(b) Show that the objective function for SCAD is strictly convex if $\gamma > 1 + 1/c_{\min}$.

3.7. *Simulation comparing ridge regression, forward selection, the lasso, MCP, and SCAD.* For this simulation, revisit the simulation described in Exercise 2.11, but now add MCP and SCAD as methods to be compared. Based on your results, comment on the situations in which you would expect each method to be the best approach (keeping in mind that some methods may never be the best approach).

3.8. *Signal to noise ratios (SNRs) for various simulation settings.*

(a) In Exercise 3.7, what is the SNR for each of the four simulation settings?

(b) Suppose, instead of being normally distributed, $X_{ij} \sim \text{Unif}(0, 1)$. Now what is the SNR?

(c) Suppose that $X_{ij} \sim \text{N}(0, 1)$, but that the features are not independent. Specifically, suppose a compound symmetric correlation structure with $\text{Cor}(\mathbf{x}_j, \mathbf{x}_k) = 0.5$ for all j and k. How does this affect the SNR?

3.9. *Exponential penalty.* Consider the following function, defined on $[0, \infty)$:

$$p(\theta|\lambda, \tau) = \frac{\lambda^2}{\tau}\left\{1 - \exp\left(-\frac{\tau\theta}{\lambda}\right)\right\}.$$

This function forms the basis of the *exponential penalty*, with tuning parameters $\lambda > 0$ and $0 < \tau < 1$. For the plots in (a) and (b), use $\lambda = 1$ and $\tau = 0.5$.

(a) Plot the penalty function $p(|\theta|)$ over the range $(-\lambda/\tau, \lambda/\tau)$.

(b) Derive $p'(|\theta|)$ and plot it over the range $(0, \lambda/\tau)$.

(c) On the basis of the plots in (a) and (b), comment on what you think estimates based on this penalty will look like (i.e., will they be sparse, will they resemble MCP, lasso, or ridge, etc.).

(d) Write an `R` function implementing a coordinate descent algorithm for the exponential penalty using the LLA approximation. For the sake of simplicity, you may assume that $\mathbf{X}$ has been standardized in advance. (Note: We recommend doing Exercise 2.6 prior to attempting part (d) of this problem.)

3.10. *Exponential penalty, continued.* Part (d) of Exercise 3.9 asks you to implement a coordinate descent algorithm for the exponential penalty for a single value of λ and assuming a standardized feature matrix. Extend your algorithm so that it (1) fits the entire coefficient path and (2) internally standardizes $\mathbf{X}$ and returns the coefficients on the original scale.

3.11. *Analysis of the carbotax data using MCP.* Analyze the carbotax data from Section 2.8 using MCP ($\gamma = 3$). Adjust for the clinical variables `Day` and `Treatment` as was done in Section 2.8. When comparing lasso and MCP models, use the same fold assignments for each model to obtain λ_{CV}.

(a) Make a table comparing the models selected by lasso and MCP. For each model, what is the maximum R^2 achieved, and how many genes are selected?

(b) Identify the gene that enters the lasso/MCP model first. Comment on the coefficient for that gene and how its coefficient path differs between the lasso and MCP models.

4

Stability and ridge-type penalties

In Chapter 3, we discussed methods for reducing the bias of lasso estimates. In this chapter, we discuss methods for doing the opposite: introducing ridge penalties in order to reduce the variance of lasso estimates at the cost of further increasing their bias. As we saw in Chapter 1, there is typically some degree of shrinkage that may be introduced for which the gains of variance reduction outweigh the cost of increased bias to produce more accurate estimates and predictions.

SHOULD INCLUDE SOMETHING ABOUT DEGREES OF FREEDOM FOR ELASTIC NET

4.1 Elastic Net

As discussed in Section 2.1.2, lasso solutions are not always unique. Example 2.1 presented a situation where two covariates were perfectly correlated, and we saw that any solution such that $\widehat{\beta}_1 + \widehat{\beta}_2 = 1 - \lambda$ and both $\widehat{\beta}_1$ and $\widehat{\beta}_2$ were positive was a solution in terms of minimizing the objective function. This happens because the absolute value penalty, while convex, is not strictly convex. In practice, the consequence is that if one solves for $\widehat{\boldsymbol{\beta}}$ using a coordinate descent algorithm (Section 2.4), one of β_1 or β_2 will be arbitrarily chosen: whichever one happens to be updated first. This is clearly unsatisfactory.

In contrast, the ridge penalty *is* strictly convex, and always produces a unique solution (Theorem 1.1). Consider, then, the following penalty, known as the *elastic net* penalty:

$$P_\lambda(\boldsymbol{\beta}) = \lambda_1 \|\boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2. \tag{4.1}$$

Here, the penalty consists of two terms, a lasso term plus a ridge term. Because the ridge penalty is strictly convex, the solution $\widehat{\boldsymbol{\beta}}$ is unique.

Example 4.1. To see how this works, let us revisit Example 2.1, which consisted of two observations: $(y_1, x_{11}, x_{12}) = (1, 1, 1)$ and

$(y_2, x_{21}, x_{22}) = (-1, -1, -1)$. Whereas the lasso penalty admitted infinitely many solutions, the elastic net penalty produces a single solution for any given value of λ:

$$\begin{cases} \widehat{\beta}_1 = \widehat{\beta}_2 = 0 & \text{if } \lambda_1 \geq 1, \\ \widehat{\beta}_1 = \widehat{\beta}_2 = \frac{1-\lambda_1}{2+\lambda_2} & \text{if } \lambda_1 < 1. \end{cases}$$

The fact that this minimizes the elastic net objective function can be verified by checking against the KKT conditions (4.3). □

Of particular note in the above example is that, regardless of λ_1 and λ_2, $\widehat{\beta}_1$ is always equal to $\widehat{\beta}_2$. This is reasonable, given that $\mathbf{x}_1 = \mathbf{x}_2$. Indeed, this is a general property of the elastic net: whenever $\mathbf{x}_j = \mathbf{x}_k$, $\widehat{\beta}_j = \widehat{\beta}_k$ (Exercise 4.1).

Example 4.1 also illustrates that the elastic net retains properties of both the lasso and ridge regression methods. From the lasso, it inherits sparsity – in particular, $\boldsymbol{\beta} = \mathbf{0}$ if $\lambda_1 > 1$. From ridge regression, the elastic net inherits the ability to always produce a unique solution as well as ridge regression's property of proportional shrinkage. In the lasso example (2.1), one possible solution was $\widehat{\beta}_1 = \widehat{\beta}_2 = (1 - \lambda_1)/2$. For the elastic net, the 2 in the denominator is replaced by $2 + \lambda_2$ in a manner analogous to equation (1.19).

A common reparameterization of the elastic net is to express the regularization parameters in terms of λ, which controls the overall degree of regularization, and α, which controls the balance between the lasso and ridge penalties:

$$\begin{aligned} \lambda_1 &= \alpha\lambda \\ \lambda_2 &= (1-\alpha)\lambda. \end{aligned} \tag{4.2}$$

This reparameterization is useful in practice, as it allows one to fix α and then select a single tuning parameter λ, which is considerably more straightforward than attempting to select λ_1 and λ_2 separately.

4.1.1 Orthonormal solutions

As with many other penalties we have considered, the elastic net has a closed form solution in the orthonormal case. Considering this special case lends considerable insight into the nature of the penalized regression method and in addition, proves useful for optimization via the coordinate descent algorithm.

The KKT conditions, or penalized likelihood equations, are slightly

modified from the equations for the lasso (2.5):

$$\begin{cases} \mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n - \lambda_2\widehat{\beta}_j = \lambda_1 \text{sign}(\widehat{\beta}_j), & \widehat{\beta}_j \neq 0 \\ |\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/n| \leq \lambda_1, & \widehat{\beta}_j = 0. \end{cases} \tag{4.3}$$

Simplifying these conditions for the orthonormal case yields

$$\begin{cases} z_j - \widehat{\beta}_j - \lambda_2\widehat{\beta}_j = \lambda_1 \text{sign}(\widehat{\beta}_j), & \widehat{\beta}_j \neq 0 \\ |z_j| \leq \lambda_1, & \widehat{\beta}_j = 0, \end{cases} \tag{4.4}$$

where $z_j = \mathbf{x}_j^T\mathbf{y}/n$. These equations can be further simplified by writing them in terms of the soft-thresholding operator (2.14):

$$\widehat{\beta}_j = \frac{S(z_j|\lambda_1)}{1 + \lambda_2}. \tag{4.5}$$

In the orthonormal case, then, the elastic net solutions are simply the lasso solutions divided by $1 + \lambda_2$. In other words, the additional ridge penalty has the same effect on the lasso as the ridge penalty itself has on ordinary least squares regression: it provides shrinkage.

As with ridge regression itself, shrinking the coefficients towards zero increases bias, but reduces variance. Since this involves drawbacks as well as advantages, adding a ridge penalty is not always universally beneficial, as the bias can dominate the variance. Still, as with ridge regression itself, it is typically the case that a profitable compromise can be reached by incorporating some (possibly small) ridge term into the penalty.

4.1.2 Grouping effect

Example 4.1 is an extreme example of a property possessed by the elastic net known as the *grouping effect*. The property states that highly correlated features will have similar estimated coefficients, which seems intuitively reasonable. The property can be described formally, in the sense that an upper bound involving the correlation between $\mathbf{x}_j$ and $\mathbf{x}_k$ can be placed on $|\widehat{\beta}_j - \widehat{\beta}_k|$ (Exercise 4.2), with the bound going to zero as the correlation $\rho_{jk} \to 1$.

Example 4.1 is a toy example involving identical features to illustrate the basic properties of the elastic net. However, even if a data set does not contain identical variables, data sets – particularly high dimensional ones – often contain highly correlated predictors. The shrinkage and grouping effects produced by the elastic net are an effective way of dealing with these correlated predictors, as we will see in the next example.

Example 4.2. For this example, we will set $n = 50$ and $p = 100$. All

features $\mathbf{x}_j$ will follow standard Gaussian distributions in the marginal sense, but we introduce correlation between the features in one of two ways:

- Compound symmetric: All features have the same pairwise correlation ρ.
- Block diagonal: The 100 features are partitioned into blocks of 5 features each, with a pairwise correlation of ρ between features in a block, but features from separate blocks are independent.

In the generating model, we will set $\beta_1 = \beta_2 = \cdots = \beta_5 = 0.5$ and $\beta_6 = \beta_7 = \cdots = \beta_{100} = 0$. Note that in the block diagonal case, this introduces a grouping property: correlated features have identical coefficients. In the compound symmetric case, on the other hand, correlation between features does not tell us anything about their corresponding coefficients.

For the elastic net penalty, for the sake of simplicity we set $\lambda_1 = \lambda_2 = \lambda$. For each independent replication of this simulation experiment, we select λ for lasso and elastic net by generating an independent validation set also of size n and select λ for that replication as the value which minimizes the prediction error on the validation set. Figure 4.1 shows the results of this simulation in terms of the MSE of lasso and elastic net. □

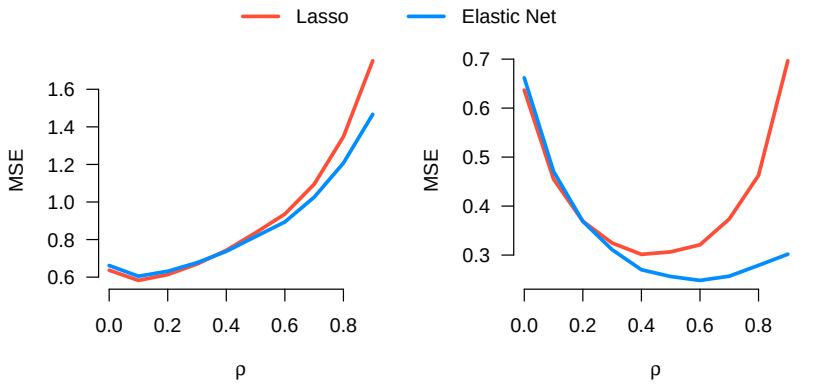

FIGURE 4.1
MSE of lasso and elastic net for the simulation study of Example 4.2. Left: Compound symmetic case. Right: Block diagonal case.

Figure 4.1 demonstrates that when the correlation between features

is not large, there is often little difference between the lasso and elastic net estimators in terms of their estimation accuracy. Indeed, when correlation is near zero, the lasso is typically more accurate, although as we will see in Section 4.2, this depends on $\boldsymbol{\beta}$ as well. When the correlation between features is large, however, the elastic net has an advantage over the lasso. The advantage is much more pronounced in the block diagonal case (right side), where the coefficients have a grouping property.

In practice, the grouping effect is often one of the strongest motivations for applying an elastic net penalty. For example, in gene expression studies, genes that have similar functions, or that work together in a pathway to accomplish a certain function, are often correlated. It is often reasonable to assume, then, that if the function is relevant to the response we are analyzing, the coefficients will be similar across the correlated group.

It is worth pointing out, however, that grouping does not always hold. For example, in a genetic association study, it is certainly quite possible for two nearby variants to be highly correlated in their inheritance patterns, but for one variant to be harmless and the other to be highly deleterious. Nevertheless, in such a case, it is often quite difficult to determine which of two highly correlated features is the causative feature, and the elastic net, which splits the estimated signal between the correlated features, offers a reasonable compromise.

4.2 Combining ridge and nonconvex penalties

The motivation for adding a ridge penalty to the lasso penalty presented in Section 4.1 also applies to the nonconvex MCP and SCAD penalties from Chapter 3. In fact, the motivation is perhaps even stronger in this case. As we saw in Chapter 3, the objective functions for MCP and SCAD may fail to be convex and present multiple local minima, which leads to difficulty in optimization and decreased numerical stability. Adding a strictly convex ridge penalty can often substantially stabilize the problem by making the objective function more convex.

The addition of a ridge penalty has a similar shrinkage effect on MCP and SCAD as it does on lasso-penalized models. In particular, for MCP

in the orthonormal case,

$$\widehat{\beta}_j = \begin{cases} \dfrac{z_j}{1+\lambda_2} & |z_j| > \gamma\lambda_1(1+\lambda_2) \\ \dfrac{S(z_j|\lambda_1)}{1-\frac{1}{\gamma}+\lambda_2} & |z_j| \leq \gamma\lambda_1(1+\lambda_2). \end{cases} \tag{4.6}$$

From this solution, we can see that the shrinkage role played by λ_2 is, in a sense, the opposite of the bias reduction role played by γ. While dividing by $1-\gamma^{-1}$ inflates the value of $S(z_j|\lambda_1)$, dividing by $1+\lambda_2$ shrinks it. When both are present in the model, the orthonormal solution is the soft-thresholding solution divided by $1-\gamma^{-1}+\lambda_2$, which could either shrink or inflate $S(z_j|\lambda_1)$ depending on the balance between γ and λ_2. It should be noted, however, that the terms are not entirely redundant; while they cancel each other out in the denominator of the bottom line of (4.6), they do not cancel out elsewhere, and in particular, they can have rather different effects in the presence of correlation among the features.

A similar phenomenon happens for SCAD, although its orthonormal solutions are somewhat more complex:

$$\widehat{\beta}_j = \begin{cases} \dfrac{z_j}{1+\lambda_2} & z_j > \gamma\lambda_1(1+\lambda_2) \\ \dfrac{S(z_j|\gamma\lambda_1/(\gamma-1))}{1-\frac{1}{\gamma-1}+\lambda_2} & \lambda_1(2+\lambda_2) < z_j \leq \gamma\lambda_1(1+\lambda_2), \\ \dfrac{S(z_j|\lambda_1)}{1+\lambda_2} & z_j \leq \lambda_1(2+\lambda_2). \end{cases} \tag{4.7}$$

Like the elastic net, the regularization parameters for the ridge-stabilized versions of MCP and SCAD are often expressed in terms of λ and α, as in (4.2).

We close this section with two examples comparing the estimation accuracy of lasso, MCP, the elastic net, and what we will abbreviate "MNet", the MCP version of the elastic net (i.e., a penalty that consists of MCP + Ridge).

Example 4.3. Suppose all covariates $\{x_j\}$ follow independent standard Gaussian distributions, and that the outcome y consists of $\mathbf{X}\boldsymbol{\beta}$ plus an error drawn from the standard Gaussian distribution. We will investigate the estimation accuracy for this situation via simulation study. For each independently generated set of data set, let $n = 100$ and $p = 500$, with 12 nonzero coefficients equal to s and the remaining 488 coefficients equal to zero. We will consider varying the signal strength s between 0.1 and 1.1. For all methods, tuning parameters are selected on the basis of mean-squared prediction error on an independent validation data set also of

size $n = 100$. For lasso and MCP, only one tuning parameter (λ) was selected (for MCP, $\gamma = 3$ was fixed); for the Enet and Mnet estimators, both λ and α were selected by external validation.

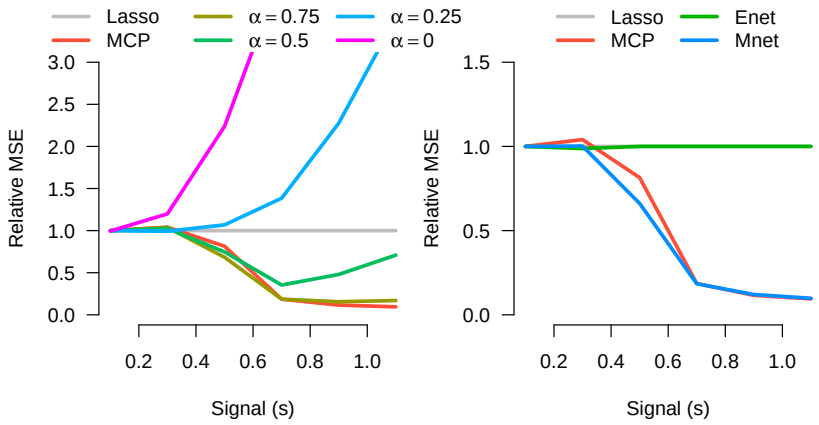

FIGURE 4.2
Left: Estimation mean squared error, relative to the lasso, for MCP and various fixed-α versions of the Mnet estimator for the simulation described in Example 4.3. Right: Estimation mean squared error, relative to the lasso, for MCP, Enet, and Mnet, with α selected by external validation for both Enet and Mnet.

Figure 4.2 presents the results of the simulation, as measured by mean squared error (MSE) of each method relative to that of the lasso. The left side presents results for various fixed-α versions of the Mnet estimator. All methods behave rather similarly when s is small, as all models end up with estimates of $\widehat{\boldsymbol{\beta}} \approx \mathbf{0}$ in these settings. As one might expect, a modest ridge penalty is beneficial in the medium-signal settings, with $\alpha = 0.5$ achieving the highest accuracy when $s = 0.5$. As signal increases, however, the downward bias of ridge and lasso play a larger role, and MCP becomes the most accurate estimator along with the $\alpha = 0.9$ Mnet estimator, which is similar to MCP.

The right side of the figure compares lasso and MCP with elastic net and Mnet, where the latter two have used external validation to select both λ and α. Here, there is little difference between the lasso and elastic net estimators, as we might have expected based on the results of Example 4.2. In particular, when s is large the two are virtually identical due, in part, to the fact that α is typically selected to be ≈ 1 for Enet when s is large. MCP and Mnet are similar to lasso and Enet when s is

small, but substantially outperform the lasso and elastic net when the signal is increased.

In summary, (a) MCP is typically no better than the lasso, and may in fact be slightly worse, when the amount of signal is small, and (b) one can improve estimation accuracy by adding a ridge penalty, but the gains are not particularly dramatic when the features are independent – in particular, there is little to no gain in adding a ridge penalty to the lasso in the absence of correlation. In Example 4.4, we examine how these conclusions are affected by the presence of correlation between the features. □

Example 4.4. This example is essentially identical to Example 4.3, except that the features are correlated. In particular, all covariates $\{x_j\}$ still follow a standard Gaussian distribution marginally, but are now correlated with a common (compound symmetric) correlation $\rho = 0.7$ between any two covariates. This is a somewhat extreme amount of correlation between features, but clearly illustrates the effect of correlation on the relative performance of the methods.

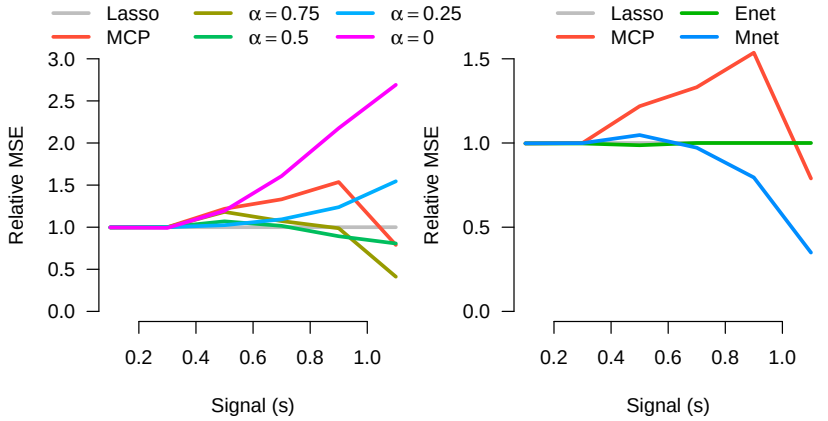

FIGURE 4.3
Left: Estimation mean squared error, relative to the lasso, for MCP and various fixed-α versions of the Mnet estimator for the simulation described in Example 4.4. Right: Estimation mean squared error, relative to the lasso, for MCP, Enet, and Mnet, with α selected by external validation for both Enet and Mnet.

Figure 4.3 differs from Figure 4.2 in several interesting aspects. One is that the benefits of shrinkage are much more pronounced in the presence of correlation. For example, while MCP was never far from the best

estimator in the uncorrelated case, it is one of the worst methods for most signal strengths in the correlated case, only outperforming the lasso when $s = 1.1$.

The other salient difference is that there is a much larger difference between Mnet and MCP in the correlated case than in the uncorrelated case. Whereas the two approaches were generally similar in Example 4.3, here Mnet outperforms MCP rather dramatically. In particular, at $s \approx 1$, Mnet is approximately twice as efficient (its MSE is half as large) as MCP, lasso, and the elastic net, which all perform similarly in terms of MSE at that setting. □

Because the relative performance of the Mnet (MCP + ridge) estimator depends considerably on the strength of signal and degree of correlation between features, it is difficult to rely on any particular value of α. In practice, it is advisable to try out several values of α and use cross-validation to guide its selection.

Furthermore, because MCP can suffer from high variance, the addition of a ridge penalty can often greatly stabilize the estimate. This is true in the sense of reducing variance and also in the numerical sense of stable solutions to the optimization problem, as we will see in the following section. Adjusting α to stabilize MCP in this way is often a more fruitful approach than adjusting γ.

Finally, although we focused on MCP + ridge in these two examples, similar statements would apply to the ridge-stabilized SCAD estimator, although because SCAD is more similar to the lasso, the effect of ridge stabilization is not as extreme as for MCP.

4.3 Coordinate descent algorithm

The coordinate descent algorithms for all of the elastic net-type methods described in this chapter (Lasso + ridge, SCAD + ridge, MCP + ridge) are very similar to the coordinate descent algorithms previously described in Chapters 2 and 3. For all of these coordinate descent algorithms, the only step that differs is the updating of bt_j. For the methods of this chapter, that updating step is given by (4.5) for the elastic net, (4.6) for MCP + ridge ("MNet"), and (4.7) for SCAD + ridge ("SNet").

Before moving on, however, it is worth revisiting the convexity considerations of Section 3.6 for the ridge-stabilized versions of MCP and SCAD. In the orthogonal case, the objective function is strictly convex

if

$$\text{MCP:} \qquad \gamma > \frac{1}{1+\lambda_2}$$
$$\text{SCAD:} \qquad \gamma > 1 + \frac{1}{1+\lambda_2}.$$

Thus, as we increase the ridge penalty regularization parameter λ_2, the objective function becomes increasingly convex. Or, to put it differently, by increasing λ_2 we increase the range of γ values over which the objective function remains convex. In Section 3.6, we discussed increasing γ to maintain the stability of the objective function and prevent discontinuous jumps between local minima along the solution path. Here, we see that another way to accomplish that same goal is by introducing a ridge component. The case study of 4.4 will further illustrate this point using the breast cancer data.

The corresponding equations for convexity in the general (non-orthogonal) case are:

$$\text{MCP:} \qquad \gamma > \frac{1}{c_{\min}+\lambda_2}$$
$$\text{SCAD:} \qquad \gamma > 1 + \frac{1}{c_{\min}+\lambda_2}.$$

The derivation of these convexity equations is left as Exercise 4.4.

4.4 Case study: Breast cancer gene expression study (revisited)

To illustrate the performance of ridge-stabilizing penalties in practice, as well as how to fit them using available software, we begin by revisiting our running example involving breast cancer gene expression data. In both `glmnet` and `ncvreg`, there is an `alpha` option that can be used to control the balance between lasso and ridge penalties, as in (4.2). In what follows, we will compare the following models in terms of their predictive accuracy and number of features selected:

```
# Elastic net
cvfit1 <- cv.glmnet(X, y)
cvfit2 <- cv.glmnet(X, y, alpha=0.75)
cvfit3 <- cv.glmnet(X, y, alpha=0.5)
cvfit4 <- cv.glmnet(X, y, alpha=0.25)
```

```
# Mnet
cvfit5 <- cv.ncvreg(X, y)
cvfit6 <- cv.ncvreg(X, y, alpha=0.75)
cvfit7 <- cv.ncvreg(X, y, alpha=0.5)
cvfit8 <- cv.ncvreg(X, y, alpha=0.25)
```

Table 4.1 contains the results of these models in terms of cross-validated prediction accuracy as summarized by $\hat{R}^2$ (2.22) and the number of variables selected by each procedure. In this example, the overall predictive accuracy for each approach is virtually identical across all the values of α considered here. The solutions themselves, however, are quite different. By increasing the proportion of the penalty allocated to the ridge component, the number of variables selected goes up: the number of nonzero coefficients for the elastic net increased by 67% as we dropped α from 1 to 0.25. A similar trend holds for Mnet, although not as pronounced.

TABLE 4.1
Predictive accuracy ($\hat{R}^2$) and number of variables selected for the breast cancer data

	$\hat{R}^2$	Variables selected
Elastic Net		
$\alpha = 1$	0.60	49
$\alpha = 0.75$	0.60	57
$\alpha = 0.5$	0.60	63
$\alpha = 0.25$	0.60	82
Mnet		
$\alpha = 1$	0.55	28
$\alpha = 0.75$	0.56	27
$\alpha = 0.5$	0.57	37
$\alpha = 0.25$	0.58	35

These results are essentially consistent with Example 4.2, in which the overall estimation accuracy of the lasso and elastic net were seen to be similar in the absence of strong correlation. For the breast cancer data, 99% of the pairwise correlations between genes were less than 0.4 in absolute value.

Nevertheless, it is worth noting that, as pointed out in Section 4.1, the ridge component stabilizes the Mnet solutions in terms of reducing concerns about local minima. As we first encountered in Section 3.7, cross-validation selects a value of λ that lies within the locally convex

portion of the MCP solution path. In that section, we addressed this concern by increasing γ. In Figure 4.4, we see that a similar effect can be achieved through the ridge penalty: as we decrease α, the locally nonconvex region shrinks and eventually λ is selected to be within the locally convex region.

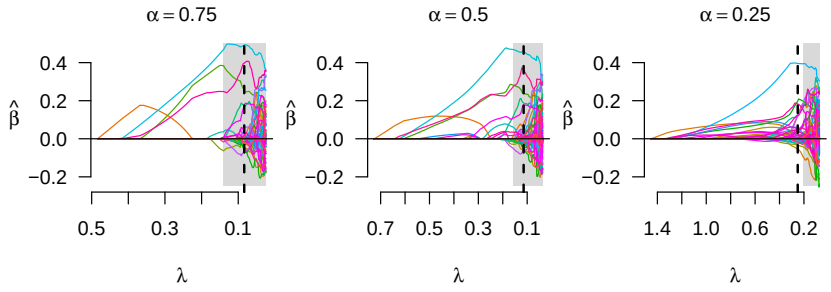

FIGURE 4.4
Mnet solution paths for the breast cancer data with different α values. The vertical dotted line represents the value of λ chosen by cross-validation.

4.5 Case study: Rat eye data

The breast cancer data from the previous section was not particularly highly correlated, nor did it suggest highly sparse solutions ($\approx$ 50 or more selected coefficients). As a contrast, we will also apply the methods of this section to gene expression data gathered from the eye tissue of 120 twelve-week-old male rats. The goal of the study was to detect genes whose expression patterns are related to that of the gene TRIM32, a gene known to be linked to a genetic disorder called Bardet-Biedl Syndrome (which, among other symptoms, leads to a number of problems with vision and proper formation of the retina). In the study, attention was restricted to the 5,000 genes with the largest variances in expression (on the log scale). Thus, this data set has $n = 120$ and $p = 5,000$. We applied the same 8 models from the previous section to this data; the results are presented in Table 4.2.

This data differs from the breast cancer data in two important ways. First, the variables are considerably more highly correlated with each

TABLE 4.2
Predictive accuracy ($\hat{R}^2$) and number of variables selected for the rat eye data

	$\hat{R}^2$	Variables selected
Elastic Net		
$\alpha = 1$	0.58	14
$\alpha = 0.75$	0.57	18
$\alpha = 0.5$	0.56	28
$\alpha = 0.25$	0.56	46
Mnet		
$\alpha = 1$	0.46	9
$\alpha = 0.75$	0.47	12
$\alpha = 0.5$	0.50	13
$\alpha = 0.25$	0.61	15

other: only 77% of the pairwise correlations are below 0.4 in absolute value, and 8% of the correlations are above 0.6. Second, we are able to identify accurate predictive models that include only a rather small number of features – perhaps as few as 9.

As a consequence, the incorporation of a ridge penalty has a larger impact in this setting than it did in the previous one, at least for MCP. Although MCP had inferior predictive accuracy than the lasso ($\hat{R}^2 = 0.46$, compared to $\hat{R}^2 = 0.58$), lowering α substantially increased the predictive accuracy to $\hat{R}^2 = 0.61$. The incorporation of a ridge penalty did not seem to benefit the lasso, although as usual it does affect the estimates and produce a more dense (less sparse) model. The Mnet estimator with $\alpha = 0.25$ is particularly attractive here, as it achieves the best prediction accuracy out of all models considered, and does so using only 15 features (out of 5,000).

Exercises

4.1. *Uniqueness of elastic net solutions.* Show that if $\mathbf{x}_j = \mathbf{x}_k$, then $\widehat{\beta}_j = \widehat{\beta}_k$ for the elastic net estimator, provided that $\lambda_2 > 0$. Hint: Use the fact that the penalty function is strictly convex.

4.2. *Grouping property for the elastic net.* Let $\widehat{\boldsymbol{\beta}}$ denote the elastic net

solution; for the sake of simplicity, you may assume throughout that $\widehat{\beta}_j$ and $\widehat{\beta}_k$ are both greater than 0.

(a) Show that

$$\widehat{\beta}_j - \widehat{\beta}_k = \frac{1}{n\lambda_2}(\mathbf{x}_j - \mathbf{x}_k)^T\mathbf{r},$$

where $\mathbf{r} = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$ denotes the vector of residuals.

(b) Show that

$$|\widehat{\beta}_j - \widehat{\beta}_k| \leq \frac{\|\mathbf{y}\|}{\lambda_2\sqrt{n}}\sqrt{2(1-\rho_{jk})},$$

where ρ_{jk} denotes the sample correlation between $\mathbf{x}_j$ and $\mathbf{x}_k$. Hint: Apply the Cauchy-Schwarz inequality to the result from part (a).

4.3. *Elastic net as reparameterized lasso.* Show that the elastic net objective function

$$Q(\boldsymbol{\beta}|\mathbf{X},\mathbf{y}) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1 + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2$$

can be rewritten in the form of the lasso objective function, with

$$\mathbf{X}^* = \begin{pmatrix} \mathbf{X} \\ \sqrt{n\lambda_2}\mathbf{I} \end{pmatrix} \qquad \text{and} \qquad \mathbf{y}^* = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix}.$$

4.4. *Convexity of objective functions for ridge + nonconvex penalties.* Let $c_{\min}$ denote the minimum eigenvalue of $\mathbf{X}^T\mathbf{X}/n$.

(a) Show that, for the MNet estimator (MCP + ridge) in the orthonormal case, the objective function is strictly convex if $\gamma > 1/(1+\lambda_2)$.

(b) Show that, for the SNet estimator (SCAD + ridge) in the orthonormal case, the objective function is strictly convex if $\gamma > 1 + 1/(1+\lambda_2)$.

(c) Show that the objective function for the MNet estimator is strictly convex if $\gamma > 1/(c_{\min} + \lambda_2)$.

(d) Show that the objective function for the SNet estimator is strictly convex if $\gamma > 1 + 1/(c_{\min} + \lambda_2)$.

4.5. *Analysis of Bardet-Biedl Syndrome data.* In Section 4.5, we analyzed gene expression data from the mammalian eye as it related to TRIM32, a gene linked to Bardet-Biedl Syndrome. In that Section, we filtered the features on the basis of variance. In this problem, we apply two different strategies: using an appropriate model (elastic net or Mnet for some α) of your choice.

- Analyze the unfiltered data (i.e., all 18,975 features).
- Instead of using variance, filter instead on the basis of chromosome (this information is included in `fData` for the mapped probes): TRIM32 is on chromosome 5, so restrict your analysis to features located on chromosome 5.

For all three approaches (unfiltered, filtered on the basis of variance, and filtered on the basis of chromosome), use 30-fold cross-validation to select λ and keep the fold assignments consistent across all the models. Use elastic net or MNet, although the values of α and γ can be different for each model, as you see fit.

(a) Describe how you chose α and γ for each approach.

(b) Provide a table that summarizes the results of each approach. For each approach, list the value of α and γ chosen, the number of features selected, and the estimated predictive accuracy of the model.

(c) On the basis of these three analyses, comment on what you see as the potential advantages and disadvantages of filtering out certain features prior to analysis.

5

Theoretical results

There is a rather large body of literature concerning theoretical results for high-dimensional penalized regression, far more than can be adequately summarized in a single chapter. Our goal for this chapter is to provide an introduction to these results, starting with the simplest case of orthonormal predictors and finishing with the $p > n$ case, as well as to convey an overview of the most important theoretical results. Although it is possible to apply penalized regression methods without understanding these results, a basic appreciation of the theoretical properties of the methods we have discussed in Chapters 1 - 4 is often quite helpful in understanding why these estimators behave they way they do. Readers interested only in the applied aspects of penalized regression may wish to skip the proofs, but we would recommend that everyone at least read through the main results, as some results are used to derive inferential approaches in Chapters 6 - 9.

Throughout this chapter, we will let $\widehat{\boldsymbol{\beta}}$ denote the estimator in question and $\boldsymbol{\beta}^*$ denote the (unknown) true value of $\boldsymbol{\beta}$. We will let $\mathcal{S} = \{j : \boldsymbol{\beta}_j^* \neq 0\}$ denote the set of nonzero coefficients (i.e., the *sparse set*), with $\boldsymbol{\beta}_{\mathcal{S}}$ and $\mathbf{X}_{\mathcal{S}}$ the corresponding subvector and submatrix. Similarly, we will let $\mathcal{N} = \{j : \boldsymbol{\beta}_j^* = 0\}$ denote the set of "null" coefficients equal to zero.

5.1 Introduction

There are three main categories of theoretical results, concerning three desirable qualities we would like our estimator to possess:

- *Prediction* The model produces accurate predictions. Specifically, the mean squared prediction error,

$$\frac{1}{n}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|^2,$$

  is small.

- *Estimation* The model produces accurate estimates. Specifically, the mean squared (estimation) error,

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|^2,$$

  is small.

- *Selection* The method accurately identifies the important features. Specifically, there is a high probability that

$$\text{sign}(\widehat{\beta}_j) = \text{sign}(\beta_j^*)$$

  for all j.

In general, we would like the methods we use to have all three qualities. However, some of these properties are more difficult to achieve than others. For example, suppose that features j and k are perfectly correlated, with $\beta_j^* > 0$ and $\beta_k^* = 0$. In this scenario, it is impossible to achieve consistent estimation or variable selection: even with an infinite amount of data, we cannot tell which of the two is the null feature (i.e., the model is unidentifiable). This does not prevent us from obtaining accurate predictions, however. Thus, as we will see, theoretical results concerning estimation and selection consistency require additional regularity conditions to exclude such scenarios, while prediction consistency can be achieved under weaker conditions.

As often in statistics, closed-form results for finite sample sizes are typically difficult to obtain, so we focus on asymptotic results as $n \to \infty$. Classically, we would treat $\boldsymbol{\beta}^*$ as fixed and consider the behavior of $\widehat{\boldsymbol{\beta}}$ as n grows. This offers a number of interesting insights, and is the setup we will initially focus on, in Sections 5.2 - 5.3.

However, these results also have the potential to be misleading, in that if n increases while $\boldsymbol{\beta}$ remains fixed, in the limit we are always looking at a situation in which $n \gg p$. Is this really a relevant justification for using the method with data where $p \gg n$? For this reason, it is also worth considering the high-dimensional asymptotic case where p is allowed to increase with n. Typically, this involves assuming that the size of the sparse set, $|\mathcal{S}|$, stays fixed, and it is only the size of the null set that increases, so that $|\mathcal{S}| \ll n$ and $|\mathcal{N}| \gg n$.

The setup we have been describing is sometimes referred to as "hard sparsity", and it is the setup we will focus on in this chapter. Specifically, hard sparsity means that $\boldsymbol{\beta}^*$ has a fixed, finite number of nonzero entries. However, many of the results we will describe also apply to other settings in which most elements of $\boldsymbol{\beta}^*$ are small, but not necessarily exactly zero. Such settings are sometimes called "weakly sparse". For example, we might allow the elements of $\mathcal{N}$ to be nonzero, provided that they are

under a certain size: $|\beta_j^*| < m$ for all $j \in \mathcal{N}$. Yet another theoretical setup is to assume that $\boldsymbol{\beta}^*$ is merely limited in size in the sense that $\sum_j |\beta_j^*| \leq R$. For many purposes, if this is true, then $\boldsymbol{\beta}^*$ can be approximated well by a sparse vector and penalized regression methods can still achieve good performance with respect to prediction and estimation.

5.2 Orthonormal case

We will begin our examination of the theoretical properties of penalized regression methods by considering the special case of an orthonormal design:

$$\begin{aligned} \tfrac{1}{n}\mathbf{X}^T\mathbf{X} &= \mathbf{I} \\ \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ \varepsilon_i &\overset{\perp\!\!\!\perp}{\sim} \mathrm{N}(0, \sigma^2). \end{aligned} \tag{O1}$$

Here, the matrix $\mathbf{X}$ is changing with the sample size, but satisfies the above condition for all values of n. For the sake of brevity, I'll refer to these assumptions in what follows as (O1). This might seem like an overly simplistic case, but it is a good place to start. Indeed, many of the important theoretical results concerning the relationship between the various methods carry over to the general design case provided some additional regularity conditions are met. We will start by showing the basic results for the lasso, then extend them to MCP, SCAD, and the elastic net.

5.2.1 Selection

Let us begin by considering the question: how large must λ be in order to ensure that all the coefficients in $\mathcal{N}$ are eliminated? The answer is given by the following theorem and its corollary.

Theorem 5.1. *Under* (O1)*,*

$$\mathbb{P}(\exists j \in \mathcal{N} : \widehat{\beta}_j \neq 0) \leq 2 \exp\left\{-\frac{n\lambda^2}{2\sigma^2} + \log p\right\}.$$

Proof. Under (O1), we have

$$\tfrac{1}{n}\mathbf{x}_j^T\mathbf{y} \sim \mathrm{N}(0, \tfrac{\sigma^2}{n})$$

and, by Exercise 5.1,

$$\mathbb{P}\{\tfrac{1}{n}\left|\mathbf{x}_j^T\mathbf{y}\right| \geq \lambda\} \leq 2\exp\{-\tfrac{n\lambda^2}{2\sigma^2}\}.$$

for all $j \in \mathcal{N}$. Therefore, the probability that $\widehat{\beta}_j \neq 0$ for at least one $j \in \mathcal{N}$ is, by the union bound,

$$\begin{aligned}\mathbb{P}\Big\{\bigcup_{j\in\mathcal{N}} \tfrac{1}{n}\left|\mathbf{x}_j^T\mathbf{y}\right| \geq \lambda\Big\} &\leq \sum_{j\in\mathcal{N}} 2\exp\{-\frac{n\lambda^2}{2\sigma^2}\} \\ &\leq 2p\exp\{-\frac{n\lambda^2}{2\sigma^2}\} \\ &= 2\exp\{-\frac{n\lambda^2}{2\sigma^2}+\log p\}. \qquad \square\end{aligned}$$

Theorem 5.1 provides an upper bound on the probability of incorrectly selecting a single null feature. We can therefore guarantee that we have eliminated all the null features by choosing a sufficiently large sequence of λ values as n increases:

Corollary 5.1. *Under* (O1)*, if* $\sqrt{n}\lambda \to \infty$*, then*

$$\mathbb{P}(\widehat{\beta}_j = 0\,\forall j \in \mathcal{N}) \to 1.$$

Note that if instead $\sqrt{n}\lambda \to c$, where c is a constant, then $\mathbb{P}(\widehat{\beta}_j = 0\,\forall j \in \mathcal{N}) \to 1-\epsilon$, where $\epsilon > 0$. In other words, for all n there remains the possibility that the lasso will select some variables from the null set $\mathcal{N}$.

Nevertheless, if $\lambda = O(\sigma\sqrt{n^{-1}\log p})$, then there is at least a chance of completely eliminating all variables in $\mathcal{N}$. Setting λ to something of this order comes up often in extending theoretical results to the case where p is allowed to grow with n (Section 5.4). Indeed, this gives us a glimpse of how it is possible to carry out statistical analyses in this setting: unless p is growing exponentially fast with n, the ratio $\log(p)/n$ can still go to zero even if $p > n$.

The above theorem considered eliminating all of the variables in $\mathcal{N}$. The natural question to consider next is: what is required in order for the lasso to select all of the variables in $\mathcal{S}$?

Theorem 5.2. *Under* (O1)*, if* $\lambda \to 0$ *as* $n \to \infty$*, then*

$$\mathbb{P}\{\mathit{sign}(\widehat{\beta}_j) = \mathit{sign}(\beta_j^*)\,\forall j \in \mathcal{S}\} \to 1.$$

Proof. Under (O1), we have

$$\tfrac{1}{n}\mathbf{x}_j^T\mathbf{y} \sim \mathrm{N}(\beta_j^*, \tfrac{\sigma^2}{n})$$

for all $j \in \mathcal{S}$. Let us assume without loss of generality that $\beta_j^* > 0$. In this case, the probability that β_j^* and $\widehat{\beta}_j$ have the same sign is

$$\mathbb{P}\{\tfrac{1}{n}\mathbf{x}_j^T\mathbf{y} \geq \lambda\} = 1 - \Phi\left(\frac{\lambda - \beta_j^*}{\sigma/\sqrt{n}}\right),$$

which converges to 1 if $\lambda \to 0$. Since $|\mathcal{S}|$ is finite, the result stated in the theorem follows. □

Putting together Theorem 5.1 and 5.2, we obtain the asymptotic conditions necessary for selection consistency as $n \to \infty$. Namely, for the lasso to be selection consistent (select the correct model with probability tending to 1), we require $\lambda \to 0$ and $\sqrt{n}\lambda \to \infty$. Note that it is possible to choose a sequence of λ values such that both conditions are satisfied simultaneously.

5.2.2 Estimation

Let us now consider estimation consistency. It is trivial to show that under (O1), $\widehat{\boldsymbol{\beta}}$ is a consistent estimator of $\boldsymbol{\beta}^*$ if $\lambda \to 0$: if $\lambda \to 0$, $\widehat{\boldsymbol{\beta}}$ converges to the OLS, which is consistent. A more interesting condition is $\sqrt{n}$-consistency.

Theorem 5.3. *Under* (O1), $\widehat{\boldsymbol{\beta}}$ *is a* $\sqrt{n}$*-consistent estimator of* $\boldsymbol{\beta}^*$ *if* $\sqrt{n}\lambda \to c$*, with* $c < \infty$*.*

Proof. Let us begin by noting that the lasso estimate is always within λ of the estimate. Thus,

$$\sqrt{n}(\widehat{\beta}_j - \beta_j^*) = \sqrt{n}(\widehat{\beta}_j^{\text{OLS}} - \beta_j^*) + O(\sqrt{n}\lambda).$$

The first term is $O_p(1)$ by the $\sqrt{n}$-consistency of the OLS estimator, while the second term is $O(1)$ by the conditions given in the theorem. □

As is clear from the above proof, the above result essentially holds if *and only if* $\sqrt{n}\lambda \to c < \infty$. Provided that $\boldsymbol{\beta}^*$ has at least one nonzero element, $\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)$ will contain a bias term on the order of $\sqrt{n}\lambda$, which will blow up if λ does not go to zero fast enough. Only in the special case of $\boldsymbol{\beta}^* = \mathbf{0}$ is bias not an issue.

Thus, in summary, it is possible for the lasso to be $\sqrt{n}$-consistent. Earlier, we saw that it was possible for the lasso to be selection consistent. However, it is not possible for the lasso to achieve both goals *at the same time*. Specifically, we require $\sqrt{n}\lambda \to \infty$ to correctly select the model with probability 1, but we require $\sqrt{n}\lambda \to c < \infty$ for $\sqrt{n}$-consistency. As we will see in Section 5.2.4, this is one of the main theoretical shortcomings of the lasso that methods such as MCP and SCAD aim to correct.

5.2.3 Prediction

In the orthonormal case, note that

$$\frac{1}{n}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 = \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2^2.$$

Thus, since $\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) = O_p(1)$ by Theorem 5.3, we have the immediate corollary that if $\sqrt{n}\lambda \to c$, the prediction error is $O_p(n^{-1})$. In the sections that follow (5.3 and 5.4), the connection between estimation and prediction consistency will not be so direct – as mentioned in Section 5.1, outside of the orthonormal case, we can encounter situations in which a model may be prediction consistent yet fail to be estimation consistent.

Nevertheless, the above result demonstrates the connection between prediction and estimation, and suggests that if we use a prediction-based criterion such as cross-validation to choose λ, we emphasize estimation accuracy over selection accuracy. Given the conflict in the requirements for λ between estimation/prediction and selection described at the end of Section 5.2.2, these results imply that cross-validation will tend to choose small values of λ that, while appropriate for estimation/prediction accuracy, have a high probability of allowing null coefficients into the model.

This result is most easily seen for the orthonormal case, but the remark is by no means specific to this situation. In general, lasso estimates based on cross-validation tend to recover the true variables with high probability, but also include a number of superfluous variables. This means that the lasso is not ideal if one desires a low false positive rate among the features selected by a model. However, the lasso can be very useful for purposes of a screening tool to recover the important variables as the first step in an analysis such as the adaptive lasso.

5.2.4 Other penalties

In the previous section, we saw that the lasso cannot simultaneously achieve both $\sqrt{n}$-consistency and selection consistency. MCP and SCAD, however, *can* accomplish this. We begin by noting that since lasso, MCP, and SCAD all have the same conditions for selecting a variable ($\frac{1}{n}\left|\mathbf{x}_j^T\mathbf{y}\right| > \lambda \implies \widehat{\beta}_j \neq 0$), the results of Theorems 5.1 and 5.2 apply to MCP and SCAD as well.

With respect to estimation, however, MCP and SCAD are able to achieve $\sqrt{n}$-consistency under weaker conditions than the lasso. Unlike Theorem 5.3 for the lasso, MCP and SCAD do not have a bias term that goes to ∞ if $\sqrt{n}\lambda \to \infty$; for MCP and SCAD the bias term goes to zero provided that $\lambda \to 0$.

In fact, we can prove an even stronger result for MCP and SCAD: given an appropriate choice of λ, these two estimators will equal the oracle estimator with probability tending to 1. The oracle estimator was mentioned in Chapter 1 and Chapter 3, but we give an explicit definition here. Letting $\widehat{\boldsymbol{\beta}}^*$ denote the oracle estimator, $\widehat{\boldsymbol{\beta}}^*$ satisfies $\widehat{\boldsymbol{\beta}}^*_{\mathcal{N}} = \mathbf{0}$ and $\widehat{\boldsymbol{\beta}}^*_{\mathcal{S}}$ minimizes $\|\mathbf{y} - \mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}_{\mathcal{S}}\|_2^2$.

Theorem 5.4. *Under* (O1)*, suppose* $\lambda \to 0$ *and* $\sqrt{n}\lambda \to \infty$*. Then* $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}^*$ *with probability tending to 1, where* $\widehat{\boldsymbol{\beta}}$ *is either the MCP or SCAD estimate.*

Proof. Corollary 5.1 establishes that $\widehat{\boldsymbol{\beta}}_{\mathcal{N}} = \mathbf{0}$ if $\sqrt{n}\lambda \to \infty$. For $j \in \mathcal{S}$, we have $\widehat{\beta}_j = \widehat{\beta}_j^{\text{OLS}}$ if $|\widehat{\beta}_j^{\text{OLS}}| > \gamma\lambda$. This happens with probability tending to 1 if $\lambda \to 0$. □

Thus, one can choose a sequence of λ values such that the MCP or SCAD estimator is both $\sqrt{n}$-consistent and selects the correct model with probability tending to 1. This result, combining estimation and selection consistency, is how the *oracle property* is usually stated. It means that the estimator is equivalent, at least in an asymptotic sense, to what we would obtain if an all-knowing oracle were to inform us in advance which coefficients were zero and which were nonzero, and we proceeded to fit an ordinary least squares model using only the nonzero features. As we will see in the coming sections, given suitable regularity conditions the oracle result holds for MCP/SCAD estimates in non-orthonormal settings as well.

Oracle results can be shown for the many of the methods introduced in Chapter 3, a reward for their efforts at bias-reduction. For example, the adaptive lasso possesses the oracle property: with a consistent initial estimator, the bias term goes to zero. Although it would never be exactly equal to the oracle estimator $\widehat{\boldsymbol{\beta}}^*$, it still obtains the optimal $\sqrt{n}$ rate of convergence while eliminating all the features in $\mathcal{N}$.

5.3 $p < n$ case

The results of Section 5.2 can be extended to the case of a general design matrix, although as mentioned in Section 5.3, we will need to require certain conditions on the design matrix $\mathbf{X}$ in order to be able to estimate $\boldsymbol{\beta}^*$ accurately. Specifically, consider the following set of assumptions,

which we will refer to as (G1):

$$\begin{aligned} \tfrac{1}{n}\mathbf{X}^T\mathbf{X} &= \boldsymbol{\Sigma}_n \\ \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ \varepsilon_i &\overset{\perp\!\!\!\perp}{\sim} \mathrm{N}(0, \sigma^2). \end{aligned} \tag{G1}$$

In addition, we assume that $\boldsymbol{\Sigma}_n \to \boldsymbol{\Sigma}$, with maximum eigenvalue ξ^* and minimum eigenvalue ξ_*.

5.3.1 Estimation

Theorem 5.5. *Under* (G1)*, the lasso estimator* $\widehat{\boldsymbol{\beta}}$ *is a* $\sqrt{n}$*-consistent estimator of* $\boldsymbol{\beta}^*$ *if (i)* $\sqrt{n}\lambda \to c$*, with* $c < \infty$ *and (ii)* $\xi_* > 0$*.*

Proof. To prove $\sqrt{n}$-consistency, we must show that for any ϵ and any $\boldsymbol{\beta}^*$, there exist n_0 and R such that

$$\mathbb{P}\{\widehat{\boldsymbol{\beta}} \in B_{R/\sqrt{n}}(\boldsymbol{\beta}^*)\} > 1 - \epsilon$$

for all $n > n_0$, where $B_{R/\sqrt{n}}(\boldsymbol{\beta}^*)$ is the p-dimensional ball centered at $\boldsymbol{\beta}^*$ with radius $R/\sqrt{n}$. In other words, $\widehat{\boldsymbol{\beta}}$ is guaranteed to be within an ever-shrinking ball centered at $\boldsymbol{\beta}^*$. We will show that, for sufficiently large n, the objective function at the surface of the ball is larger than the objective function at the center, and therefore, the minimizer $\widehat{\boldsymbol{\beta}}$ must lie somewhere within the ball. Letting $D(\mathbf{u}) \equiv nQ(\boldsymbol{\beta}^* + \frac{\mathbf{u}}{\sqrt{n}}) - nQ(\boldsymbol{\beta}^*)$ denote the difference in (rescaled) objective function between the surface and center of the ball, where $\mathbf{u}$ is a vector with $\|\mathbf{u}\| = R$, we have

$$\begin{aligned} D(\mathbf{u}) &\geq \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}(\boldsymbol{\beta}^* + \tfrac{\mathbf{u}}{\sqrt{n}})\|_2^2 - \tfrac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 \\ &\quad + n\lambda \sum_{j \in \mathcal{S}} \left\{ \left|\beta_j^* + \tfrac{u_j}{\sqrt{n}}\right| - \left|\beta_j^*\right| \right\} \\ &= \tfrac{1}{\sqrt{n}}\mathbf{u}^T\mathbf{X}^T\boldsymbol{\varepsilon} + \tfrac{1}{2n}\mathbf{u}^T\mathbf{X}^T\mathbf{X}\mathbf{u} + n\lambda \sum_{j \in \mathcal{S}} \left\{ \left|\beta_j^* + \tfrac{u_j}{\sqrt{n}}\right| - \left|\beta_j^*\right| \right\} \end{aligned}$$

As $n \to \infty$, the above converges (in distribution) to

$$\sigma\sqrt{\mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u}}Z + \frac{1}{2}\mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u} + c\sum_{j \in \mathcal{S}} \mathrm{sign}(\beta_j^*)u_j,$$

where Z is a standard normal random variable. Now, the first and third terms may be negative, but the middle term must be positive under the requirement that $\boldsymbol{\Sigma}$ is positive definite. What remains is to show that we can always choose an R such that the middle term dominates the

other two. Letting the first and third terms be as large as possible and the middle as small as possible, we arrive at

$$\lim_{n\to\infty} D(\mathbf{u}) \succeq R\sigma\sqrt{\xi^*}Z + \tfrac{R^2}{2}\xi_* - cR\left|\mathcal{S}\right|,$$

where $\succeq$ denotes stochastic ordering. Since the middle term is of order R^2 and the others of order R, we can always choose R such that $\sup_{\|\mathbf{u}\|=R} D(\mathbf{u}) > 0$ with probability at least $1-\epsilon$. □

In the above, note that if $\sqrt{n}\lambda \to \infty$, the above result would no longer hold.

5.3.2 Prediction

Given $\sqrt{n}$-consistency, it is straightforward to show that the prediction error is $O_p(n^{-1})$: demonstration here.

However, do we actually need $\xi_* > 0$ for prediction consistency? The answer, as it turns out, is no. Alternative proof without eigenvalue assumption here.

5.3.3 Selection

Fan and Li Lemma 1

Remark on convergence in distribution

Remark on oracle property

5.4 $p > n$ case

5.4.1 Eigenvalue conditions

Restricted eigenvalues

Sparse Riesz condition

Irrepresentable condition

5.4.2 Prediction

5.4.3 Estimation

5.4.4 The $p > n$ case

Let $\psi(\boldsymbol{\beta}) = \|\mathbf{X}\boldsymbol{\beta}\|^2/(2n)$ and $\mathbf{z} = \mathbf{X}^T\mathbf{y}/n$. Define

$$\ell(\boldsymbol{\beta}) = \psi(\boldsymbol{\beta}) - \mathbf{z}'\boldsymbol{\beta}. \tag{5.1}$$

We have

$$\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 = \ell(\boldsymbol{\beta}) + \frac{1}{2n}\mathbf{y}^T\mathbf{y}.$$

Since the term $\mathbf{y}^T\mathbf{y}/(2n)$ does not involve $\boldsymbol{\beta}$, the lasso can be equivalently defined as

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \Big\{ \ell(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1. \Big\}$$

The score function of ℓ is the gradient

$$\dot{\ell}(\boldsymbol{\beta}) = \frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \frac{\partial \psi(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} - \mathbf{z} = \dot{\psi}(\boldsymbol{\beta}) - \mathbf{z}.$$

Define

$$D(\boldsymbol{\beta}, \boldsymbol{\beta}^*) = \ell(\boldsymbol{\beta}) - \ell(\boldsymbol{\beta}^*) - \dot{\ell}(\boldsymbol{\beta}^*)^T(\boldsymbol{\beta} - \boldsymbol{\beta}^*).$$

This is the Bregman divergence associated with ℓ. Its symmetrized version is

$$\Delta(\boldsymbol{\beta}, \boldsymbol{\beta}^*) = D(\boldsymbol{\beta}, \boldsymbol{\beta}^*) + D(\boldsymbol{\beta}^*, \boldsymbol{\beta}) = (\boldsymbol{\beta} - \boldsymbol{\beta}^*)^T(\dot{\psi}(\boldsymbol{\beta}) - \dot{\psi}(\boldsymbol{\beta}^*)). \tag{5.2}$$

For the linear regression model (2.1), $\dot{\psi}(\boldsymbol{\beta}) - \dot{\psi}(\boldsymbol{\beta}^*) = \mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - \mathbf{X}^T\mathbf{X}\boldsymbol{\beta}^*$, thus

$$\Delta(\boldsymbol{\beta}, \boldsymbol{\beta}^*) = \|\mathbf{X}\boldsymbol{\beta} - \mathbf{X}\boldsymbol{\beta}^*\|^2/n.$$

which is simply the model error.

Define $z_0^* = \|\{\mathbf{z} - \dot{\psi}(\boldsymbol{\beta}^*)\}_S\|_\infty, z_1^* = \|\{\mathbf{z} - \dot{\psi}(\boldsymbol{\beta}^*)\}_{S^c}\|_\infty$, and $z^* = \|\mathbf{z} - \dot{\psi}(\boldsymbol{\beta}^*)\|_\infty$. Obviously, $z^* \geq \max\{z_0^*, z_1^*\}$. For the linear regression model (2.1),

$$\mathbf{z} - \dot{\psi}(\boldsymbol{\beta}^*) = (\mathbf{X}^T\mathbf{y} - \mathbf{X}^T\mathbf{X}\boldsymbol{\beta}^*)/n = X^T(\mathbf{y} - X\boldsymbol{\beta}^*)/n.$$

In particular, for $\boldsymbol{\beta}^* = \boldsymbol{\beta}_0$, $\mathbf{z} - \dot{\psi}(\boldsymbol{\beta}_0) = \mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}_0)/n = X^T\boldsymbol{\varepsilon}/n$.

Theorem 5.6. *Let $\boldsymbol{\beta}^*$ be a target vector. In the event $\{z^* \le \lambda\}$,*

$$\Delta(\boldsymbol{\beta}, \boldsymbol{\beta}^*) \le 2\lambda \|\boldsymbol{\beta}^*\|_1. \tag{5.3}$$

The upper bound in Theorem (5.6) gives the so called "slow rate" of convergence for the Bregman divergence. This theorem makes no assumption on the model. In particular, it does not assume any sparsity condition on $\boldsymbol{\beta}_0$, so $\boldsymbol{\beta}_0$ can be a dense vector. To obtain a faster rate of convergence, we will need to make use the sparsity condition.

Lemma 5.1. *Let $\boldsymbol{\delta} = \widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*$. For any target vector $\boldsymbol{\beta}^*$ and $S \supseteq \{j : \beta_j^* \neq 0\}$,*

$$\Delta(\boldsymbol{\beta}^* + \boldsymbol{\delta}, \boldsymbol{\beta}^*) + (\lambda - z_1^*)\|\boldsymbol{\delta}_{S^c}\|_1 \le (\lambda + z_0^*)\|\boldsymbol{\delta}_S\|_1.$$

Consequently, for any $\xi > 1$, in the event $\{z_0^* + \xi z_1^* \le (\xi - 1)\lambda\}$, $\boldsymbol{\delta}$ belongs to the set

$$R(\xi, S) = \{\mathbf{b} \in \mathbb{R}^p : \|\mathbf{b}_{S^c}\|_1 \le \xi \|\mathbf{b}_S\|_1\} \tag{5.4}$$

By Lemma 5.1, it suffices to study the analytical properties of the lasso in the restricted region (5.4) and show that the event $\{z_0^* + \xi z_1^* \le (\xi - 1)\lambda\}$ has large probability. The choices of the target vector $\boldsymbol{\beta}^*$ and the sparse set $S = \{j : \beta_j^* \neq 0\}$ are quite flexible. The main requirement is that $\{S, z_0^*, z_1^*\}$ should be small. In the linear regression model, we can consider $\boldsymbol{\beta}^*$ as the vector of true regression coefficients, that is, $\boldsymbol{\beta}^* = \boldsymbol{\beta}_0$. However, $\boldsymbol{\beta}^*$ can also be a sparse version of a true $\boldsymbol{\beta}_0$, for example, $\boldsymbol{\beta}_j^* = \beta_{0j} 1\{|\beta_{0j}| \ge \tau\}$ for a small τ.

Recall the Gram matrix $G = \mathbf{X}^T\mathbf{X}/n$. Define

$$\mathrm{RE}_2(\xi, S) = \inf_{\mathbf{b} \in R(\xi, S)} \frac{\|\mathbf{X}\mathbf{b}\|_2}{\sqrt{n}\|\mathbf{b}\|_2} = \inf_{\mathbf{b} \in R(\xi, S)} \frac{(\mathbf{b}^T G \mathbf{b})^{1/2}}{\|\mathbf{b}\|_2}. \tag{5.5}$$

Let $\phi(\mathbf{b})$ defined in $\mathbb{R}^p$ be a quasi-star shaped function, meaning $\phi(\mathbf{b})$ is continuous and nondecreasing in $t \in [0, \infty)$ for all $\mathbf{b} \in \mathbb{R}^p$ and $\lim_{\mathbf{b} \to 0} \phi(\mathbf{b}) = 0$. Important special cases of a quasi-star shaped functions include the ℓ_q norms $\|\mathbf{b}\|_q, q > 0$. Define

$$F_0(\xi, S; \phi) = \inf_{\mathbf{b} \in R(\xi, S)} \frac{\mathbf{b}^T G \mathbf{b}}{\|\mathbf{b}_S\|_1 \phi(\mathbf{b})}. \tag{5.6}$$

For $\phi(\mathbf{b}) = \|\mathbf{b}_S\|_1/|S|$, $F_0^{1/2}(\xi, S; \phi_{1,S})$ is the compatibility factor

$$\kappa(\xi, S) = \inf_{\mathbf{b} \in R(\xi, S)} \Big(\frac{\mathbf{b}^T G \mathbf{b}}{\|\mathbf{b}_S\|_1^2/|S|} \Big)^{1/2}.$$

Since $\|\mathbf{b}_S\|_1^2 \le \|\mathbf{b}\|_2^2 |S|$, $\kappa(\xi, S) \ge \mathrm{RE}_2(\xi, S)$.

Theorem 5.7. *In the event* $\{z_0^*+\xi z_1^* \le (\xi-1)\lambda\}$, *we have, with* $\phi_q(\mathbf{b}) = \|\mathbf{b}\|_q/|S|^{1/q}$,

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_q \le \frac{(\lambda + z_1^*)|S|^{1/q}}{F_0(\xi, S; \phi_q)}, \text{ for every } q > 0, \tag{5.7}$$

and with $\phi_{1,S}(\mathbf{b}) = \|\mathbf{b}_S\|_1/|S|$,

$$\Delta(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}_0) + (\lambda - z_1^*)\|\boldsymbol{\delta}_{S^c}\|_1 \le \frac{(\lambda + z_0^*)|S|}{F_0(\xi, S; \phi_{1,S})}. \tag{5.8}$$

Since $z^* \ge \max\{z_0^*, z_1^*\}$, the event $\{z_0^*+\xi z_1^* \le (\xi-1)\lambda\}$ is included in $\{z^* \le (\xi-1)\lambda/(\xi+1)\}$. It can be verified that with $\lambda_1 = \sigma\sqrt{2\log(p)/n}$,

$$\mathbf{P}(z^* > \lambda_1) \le \frac{2}{\sqrt{2\pi p}}. \tag{5.9}$$

Corollary 5.2. *Suppose* $\lambda = \sigma\sqrt{2\log(p)/n}$.

(i) With probability at least $1 - (2/\sqrt{2\pi p})$,

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_q \le \frac{2|S|^{1/q}}{(1+\xi)F_0(\xi, S; \phi_q)}\sigma\sqrt{\frac{2\log p}{n}}, \text{ for every } q > 0.$$

(ii) With the same probability as in (i), and with $\phi_{1,S}(\mathbf{b}) = \|\mathbf{b}_S\|_1/|S|$,

$$\Delta(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}_0) + (\lambda - z_1^*)\|\boldsymbol{\delta}_{S^c}\|_1 \le \frac{2|S|}{(1+\xi)F_0(\xi, S; \phi_{1,S})}\sigma\sqrt{\frac{2\log p}{n}}. \tag{5.10}$$

5.4.5 Selection

The above results give upper bounds for estimation and prediction errors. The following theorem provides sufficient conditions under which the lasso is selection consistent.

Theorem 5.8. *Suppose there exist constants* $0 < \kappa_0 < 1$ *and* $0 < \kappa_1 < \infty$ *such that*

$$|\mathbf{X}_{S^c}^T\mathbf{X}_S(\mathbf{X}_S^T\mathbf{X}_S)^{-1} sign(\boldsymbol{\beta}_S)|_\infty \le \kappa_0 \tag{5.11}$$

and

$$\|\mathbf{X}_{S^c}^T\mathbf{X}_S(\mathbf{X}_S^T\mathbf{X}_S)^{-1}\|_\infty \le \kappa_1. \tag{5.12}$$

Then $sign(\widehat{\boldsymbol{\beta}}) = sign(\boldsymbol{\beta}^*)$ *in the event that*

$$\|(\mathbf{X}_S^T\mathbf{X}_S)^{-1}\|_\infty(\lambda + z_0^*) < \min_{j\in S}|\beta_j^*|.$$

Therefore, $\mathbf{P}(sign(\widehat{\boldsymbol{\beta}}) = sign(\boldsymbol{\beta}^*)) \to 1$ *if* $\mathbf{P}\{\|(\mathbf{X}_S^T\mathbf{X}_S)^{-1}\|_\infty(\lambda + z_0^*) < \min_{j\in S}|\beta_j^*|\} \to 1$.

5.4.6 Proofs of the results in Section 5.4.4

The proofs will be based on Huang and Zhang (2012)[Estimation and selection via absolute penalized convex minimization and its multistage adaptive applications. *Journal of Machine Learning Research*, 13, 1839-1864], in which the theoretical properties of the general weighted lasso for a class of convex losses are studied.

For any $A \subseteq \{1, \ldots, J\}$, denote

$$\mathbf{X}_A = (X_j, j \in A), \quad \mathbf{\Sigma}_A = \mathbf{X}_A' \mathbf{X}_A / n.$$

Let the true value of the regression coefficients be $\boldsymbol{\beta}^o = (\beta_1^o, \ldots, \beta_p^o)'$. Let $S = \{j : \beta_j^o \neq 0, 1 \leq j \leq p\}$, which is the set of indices of the nonzero coefficients in the underlying model. Let $\beta_*^o = \min\{|\beta_j^o| : j \in S\}$ and set $\beta_*^o = \infty$ if S is empty. Define

$$\widehat{\boldsymbol{\beta}}^o = \arg\min_{\boldsymbol{\beta} \mathbb{R}^p} \{\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 : \beta_j = 0 \ \forall j \notin S\}. \tag{5.13}$$

This is the oracle least squares estimator. Of course, it is not a real estimator, since the oracle set is unknown.

Let $c_{\min}$ be the smallest eigenvalue of $\mathbf{\Sigma}$, and let c_1 and c_2 be the smallest and largest eigenvalues of $\mathbf{\Sigma}_S$, respectively.

We first consider the case where the MCP objective function is convex. This necessarily requires $c_{\min} > 0$.

Let

$$h(t, k) = \exp\big(-k(\sqrt{2t-1} - 1)^2/4\big), \ t > 1, k = 1, 2 \ldots. \tag{5.14}$$

This function arises from an upper bound for the tail probabilities of chi-square distributions given in Lemma 1 in the Appendix.

$$\eta_{1n}(\lambda) = (p - |S|)h\big(\lambda^2 n/\sigma^2, 1\big) \tag{5.15}$$

and

$$\eta_{2n}(\lambda) = |S|h\big(c_1 n(\beta_*^o - \gamma\lambda)^2/\sigma^2, 1\big). \tag{5.16}$$

Theorem 5.9. *Suppose $\varepsilon_1, \ldots, \varepsilon_n$ are independent and identically distributed as $N(0, \sigma^2)$. Then for any (λ, γ) satisfying $\gamma > 1/c_{\min}$, $\beta_*^o > \gamma\lambda$ and $n\lambda^2 > \sigma^2$, we have*

$$\mathbf{P}\big(\widehat{\boldsymbol{\beta}}(\lambda, \gamma) \neq \widehat{\boldsymbol{\beta}}^o\big) \leq \eta_{1n}(\lambda) + \eta_{2n}(\lambda).$$

This theorem provides an upper bound on the probability that $\widehat{\boldsymbol{\beta}}(\lambda, \gamma)$ is not equal to the oracle least squares estimator. The condition $\gamma > 1/c_{\min}$ ensures that the 2-norm group MCP criterion is strictly

convex. This implies $\widehat{\boldsymbol{\beta}}(\lambda,\gamma)$ is uniquely characterized by the Karush-Kuhn-Tucker conditions. The condition $n\lambda^2 > \sigma^2$ requires that λ cannot be too small.

Let

$$\begin{aligned}\lambda_n &= \sigma\sqrt{2\log(p)/(n)} \text{ and} \\ \tau_n &= \sigma\sqrt{2\log(\max\{|S|,1\})/(nc_1)}.\end{aligned} \tag{5.17}$$

The following corollary is an immediate consequence of Theorem 5.9.

Corollary 5.3. *Suppose that the conditions of Theorem 5.9 are satisfied. Also suppose that $\beta^o_* \geq \gamma\lambda + a_n\tau_n$ for $a_n \to \infty$ as $n \to \infty$. If $\lambda \geq a_n\lambda_n$, then*

$$\mathbf{P}(\widehat{\boldsymbol{\beta}}(\lambda,\gamma) \neq \widehat{\boldsymbol{\beta}}^{o}) \to 0 \text{ as } n \to \infty.$$

By Corollary 5.3, the MCP estimator behaves like the oracle least squares estimator with high probability. This of course implies it is selection consistent. For the standard LASSO estimator, a sufficient condition for its sign consistency is the strong irrepresentable condition (Zhao and Yu 2006). Here a similar condition holds automatically due to the form of the MCP. Specifically, let $\boldsymbol{\beta}^o_S = (\beta_j^{o'} : j \in S)'$. Then an extension of the irrepresentable condition to the present setting is, for some $0 < \delta < 1$,

$$\max_{j\notin S} \|\mathbf{X}_j'\mathbf{X}_S(\mathbf{X}_S'\mathbf{X}_S)^{-1}\dot{P}(\boldsymbol{\beta}^o_S;\lambda,\gamma)/\lambda\|_2 \leq 1-\delta, \tag{5.18}$$

where $\dot{P}(\boldsymbol{\beta}^o_S;\lambda,\gamma) = (\dot{P}(|\beta^o_j|;\lambda,\gamma)\text{sign}(\beta^o_j) : j \in S)'$ with

$$\dot{P}(|\boldsymbol{\beta}^o_j|;\lambda,\gamma) = \lambda\big(1 - |\beta^o_j|/\gamma\lambda\big)_+.$$

Since it is assumed that $\min_{j\in S}|\beta^o_j| > \gamma\lambda$, we have $\dot{P}(|\beta^o_j|;\lambda,\gamma) = 0$ for all $j \in S$. Therefore, (5.18) holds automatically.

We now consider the high-dimensional case where $J > n$. We require the sparse Riesz condition, or SRC (Zhang and Huang 2008), which is a form of sparse eigenvalue condition. We say that $\mathbf{X}$ satisfies the SRC with rank d^* and spectrum bounds $\{c_*, c^*\}$ if

$$0 < c_* \leq \|\mathbf{X}_A\mathbf{u}\|_2^2/n \leq c^* < \infty, \ \forall A \text{ with } |A| \leq d^*, \|\mathbf{u}\|_2 = 1. \tag{5.19}$$

We refer to this condition as SRC(d^*, c_*, c^*).

Let $K_* = (c^*/c_*) - (1/2)$, $m_* = K_*|S|$ and $\xi = 1/(4c^{)}$. Define

$$\eta_{3n}(\lambda) = (p-|S|)^{m_*}\frac{e^{m_*}}{m_*^{m_*}}h(\xi n\lambda^2\sigma^{-2}/d_{\max}, m_*). \tag{5.20}$$

Let η_{1n} and η_{2n} be as in (5.15) and (5.16).

Theorem 5.10. *Suppose $\varepsilon_1, \ldots, \varepsilon_n$ are independent and identically distributed as $N(0, \sigma^2)$ and $\mathbf{X}$ satisfies the SRC(d^*, c_*, c^*) in (5.19) with $d^* \geq (K_*+1)|S|$. Then for any (λ, γ) satisfying $\beta^o_* > \gamma\lambda$, $n\lambda^2\xi > \sigma^2 d_{\max}$ and $\gamma \geq c_*^{-1}\sqrt{4 + (c_*/c^*)}$, we have*

$$\mathbf{P}\big(\widehat{\boldsymbol{\beta}}(\lambda, \gamma) \neq \widehat{\boldsymbol{\beta}}^o\big) \leq \eta_{1n}(\lambda) + \eta_{2n}(\lambda) + \eta_{3n}(\lambda).$$

Let

$$\lambda_n^* = 2\sigma\sqrt{2c^* \log(p - |S|)/n}$$

and τ_n be as in (5.17). Theorem 5.10 has the following corollary.

Corollary 5.4. *Suppose the conditions of Theorem 5.10 are satisfied. Also suppose $\beta^o_* \geq \gamma\lambda + a_n\tau_n$ for $a_n \to \infty$ as $n \to \infty$. Then if $\lambda \geq a_n\lambda_n^*$,*

$$\mathbf{P}\big(\widehat{\boldsymbol{\beta}}(\lambda, \gamma) \neq \widehat{\boldsymbol{\beta}}^o\big) \to 0 \quad \text{as } n \to \infty.$$

Theorem 5.10 and Corollary 5.4 provide sufficient conditions for the selection consistency of the global MCP estimator in the $p \gg n$ situations. For example, we can have $p = \exp\{o(n/c^*)\}$. The condition $n\lambda^2\xi > \sigma^2$ is stronger than the corresponding condition $n\lambda^2 > \sigma^2$ in Theorem 5.9. The condition $\gamma \geq c_*^{-1}\sqrt{4 + (c_*/c^*)}$ ensures that the MCP criterion is convex in any d^*-dimensional subspace. It is stronger than the minimal sufficient condition $\gamma > 1/c_*$ for convexity in d^*-dimensional subspaces. These reflect the difficulty and extra efforts needed in reducing a p-dimensional problem to a d^*-dimensional problem. The SRC in (5.19) guarantees that the model is identifiable in a lower d^*-dimensional space.

The results presented above are concerned with the global solutions. The properties of the local solutions, such as those produced by the coordinate descent algorithm, to concave penalties remain largely unknown in models with $p \gg n$. An interesting question is under what conditions the local solutions are equal to or sufficiently close to the global solutions so that they are still selection consistent.

5.4.7 Proof of oracle property

The proof will be based on Zhang (2010). (Zhang (2010). Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.* **38** 894-942.)

5.4.8 Technical details

5.5 Oracle ridge estimators

A section will be added on the theoretical properties of the Mnet approach based on Huang et al. (2015). (Huang, Breheny, Lee, Ma and Zhang (2015). The Mnet method for variable selection. Accepted for publication by *Statistica Sinica*.)

5.6 Sandbox

THIS IS JUST A TEMPORARY HOLDING PLACE FOR MATERIAL THAT MAY OR MAY NOT APPEAR.

Theorem 5.11. *Suppose that as $n \to \infty$, we have $\mathbf{X}^T\mathbf{X}/n \to G_0$ and $\max_{1\le i\le n} \mathbf{x}_i^T\mathbf{x}_i/n \to 0$. If $\sqrt{n}\lambda \to \lambda_0$, then we have*

$$\sqrt{n}(\widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}^*) \to_D \arg\min_{\mathbf{t}} V(\mathbf{t}), \tag{5.21}$$

where

$$V(\mathbf{t}) = -\sigma\mathbf{t}'\boldsymbol{\Sigma}^{1/2}\mathbf{z} + \frac{1}{2}\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t} + \lambda_0 \sum_{j=1}^{p}\{t_j sign(\beta_{0j})1(\beta_{0j} \neq 0) + |t_j|1(\beta_{0j} = 0)\}.$$

Unlike in regular situations where we have an asymptotically normal distribution for our estimators such as an MLE, the lasso has a complicated asymptotic distribution that depends on the unknown and is difficult to evalute.

When $\lambda_0 = 0$, i.e., $\sqrt{n}\lambda \to 0$,

$$V(\mathbf{t}) = -\sigma\mathbf{t}'G_0^{1/2}Z + \frac{1}{2}\mathbf{t}'G_0\mathbf{t},$$

which is minimized at $\sigma G_0^{1/2}Z \sim N(0, \sigma^2 G_0)$. This is the limit distribution of the OLS estimator. However, this is an uninteresting case, for the Lasso essentially behaves like the OLS estimator, which does not do variable selection. The right order of growth for λ is when $\sqrt{n}\lambda \to \lambda_0$ for $\lambda_0 > 0$.

Consider the case where $p = 2$. Suppose $\beta_1 > 0$ and $\beta_2 = 0$, and suppose the off-diagonal element of $\mathbf{G}_0$ is ρ. Its diagonal elements are 1 since the predictors are standardized. Then

$$V(\mathbf{t}) = -\mathbf{t}'\mathbf{G}_0^{1/2}\mathbf{z} + \frac{1}{2}\mathbf{t}'\mathbf{G}_0\mathbf{t} + \lambda_0 t_1 + \lambda_0|t_2|.$$

If $V(\mathbf{t})$ is minimized at $t_2 = 0$, then

$$t_1 - z_1 = \lambda_0 \text{ and } \lambda_0 \leq \rho t_1 - z_2 \leq \lambda_0.$$

Solving for t_1 in the first equation, we get $t_1 = z_1 - \lambda_0$. Thus $t_2 = 0$ if

$$-\lambda_0 \leq \rho(z_1 - \lambda_0) - z_2 \leq \lambda_0.$$

So the probability of $t_2 = 0$ is $\mathbf{P}(|\rho(z_1 - \lambda_0) - z_2| \leq \lambda_0)$. After some calculation, this equals

$$\Phi\Big(\lambda_0\sqrt{\frac{1+\rho}{1-\rho}}\Big) - \Phi\Big(-\lambda_0\sqrt{\frac{1-\rho}{1+\rho}}\Big).$$

If we wish this probability big, we need a big λ_0, but a big λ_0 will cause big bias in t_1.

Proof of (5.21). Let $\boldsymbol{\beta} = \boldsymbol{\beta}^* + n^{-1/2}\mathbf{t}$. Define

$$\begin{aligned} L(\mathbf{t}) &= \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1 \\ &= \frac{1}{2n}\|\mathbf{y} - \mathbf{X}(\boldsymbol{\beta}_0 + n^{-1/2}\mathbf{t})\|^2 + \lambda\|\boldsymbol{\beta}_0 + n^{-1/2}\mathbf{t}\|. \end{aligned}$$

Let $\hat{\boldsymbol{\theta}}_n = \arg\min_{\mathbf{t}} V_n(\mathbf{t})$, where

$$\begin{aligned} V_n(\mathbf{t}) &= n(L_n(t) - L_n(0)) \\ &= \frac{1}{2}(\|\boldsymbol{\varepsilon} - n^{-1/2}\mathbf{X}\mathbf{t}\|^2 - \|\boldsymbol{\varepsilon}\|^2) + n\lambda(\|\boldsymbol{\beta}_0 + n^{-1/2}\mathbf{t}\|_1 - \|\boldsymbol{\beta}_0\|_1). \end{aligned}$$

Then $\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) = \hat{\boldsymbol{\theta}}_n$. The idea is to show that $V_n(\mathbf{t}) \to_d V(\mathbf{t})$. By the argmin continuous mapping theorem (Kim and Pollard (1990)), if we can show

$$\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) = O_p(1), \tag{5.22}$$

and

$$V_n(t) \to_D V(t), \tag{5.23}$$

then

$$\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0) = \hat{\boldsymbol{\theta}}_n \to_d \arg\min_{\mathbf{t}} V(\mathbf{t}).$$

The assertion (5.22) can be easily proved. So we only need to show (5.23). The first term in the expression of V_n

$$\frac{1}{2}(\|\boldsymbol{\varepsilon} - n^{-1/2}X\mathbf{t}\|^2 - \|\boldsymbol{\varepsilon}\|^2) \to_D -\sigma\mathbf{t}'\boldsymbol{\Sigma}^{1/2}Z + \frac{1}{2}\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t},$$

where $Z \sim N(\mathbf{0}, I_p)$, The second term in the expression of V_n

$$n\lambda(\|\boldsymbol{\beta}_0 + n^{-1/2}\mathbf{t}\|_1 - \|\boldsymbol{\beta}_0\|_1) \to \\ \lambda_0 \sum_{j=1}^{p} \{t_j \text{sign}(\beta_{0j}) 1(\beta_{0j} \neq 0) + |t_j| 1(\beta_{0j} = 0)\}.$$

These imply $V_n(\mathbf{t}) \to_D V(\mathbf{t})$, where

$$V(\mathbf{t}) = -\sigma\mathbf{t}'\boldsymbol{\Sigma}^{1/2}\mathbf{z} + \frac{1}{2}\mathbf{t}'\boldsymbol{\Sigma}\mathbf{t} \\ + \lambda_0 \sum_{j=1}^{p} \{t_j \text{sign}(\beta_{0j}) 1(\beta_{0j} \neq 0) + |t_j| 1(\beta_{0j} = 0)\}.$$

Therefore, $\sqrt{n}(\widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_0) \to_D \arg\min_{\mathbf{t}} V(\mathbf{t})$. □

5.7 Bibliographical notes

This section will include the bibliographical notes on the materials presented in this chapter.

5.8 Exercises

5.1. *Gaussian tail bound.* The Chernoff bound for a random variable X says that for any $t > 0$, if $\mathbb{E}(e^{tX})$ exists, then

$$\mathbb{P}(X \geq \lambda) \leq \frac{\mathbb{E}(e^{tX})}{e^{t\lambda}}$$

Use the Chernoff bound to show that for $Z \sim \mathrm{N}(0, \sigma^2/n)$,

$$\mathbb{P}(|Z| \geq \lambda) \leq 2 \exp\left\{-\frac{n\lambda^2}{2\sigma^2}\right\}$$

for any $\lambda > 0$.

5.2. Verify the maximal inequality (5.9).

5.3. *Prediction bound under RE condition.* Suppose that the feature matrix $\mathbf{X}$ satisfies the restricted eigenvalue condition $\mathrm{RE}(\tau)$. Below, $\widehat{\boldsymbol{\beta}}$ is the lasso solution for a given value of λ.

(a) Show that if $\lambda \geq \frac{2}{n}\|\mathbf{X}^T\boldsymbol{\varepsilon}\|_\infty$,

$$\frac{1}{n}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 \leq \frac{9}{\tau}\lambda^2 |\mathcal{S}|.$$

(b) Show that if $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}$ with $\varepsilon_i \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$ and $\lambda = 2\sigma\sqrt{c\log(p)/n}$, then

$$\frac{1}{n}\|\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}^*\|_2^2 \leq 36c\frac{\sigma^2}{\tau}\frac{|\mathcal{S}|\log p}{n}$$

with probability $1 - 2\exp\{-\frac{1}{2}(c-2)\log p\}$.

Part II

Inference

6

False discovery rates

Up to this point in the book, the only inferential questions we have addressed have concerned the predictive ability of the model. Cross-validation is an extremely useful tool for inference about prediction, but we need to develop new approaches to address inference with respect to the parameters themselves. This is the subject of Part II of this book.

This chapter addresses the question: How reliable are the selections made by the model? Specifically, given that we have selected a number of features, we would like to know what fraction of those features are likely to be mere noise, only spuriously associated with the outcome. This is closely related to the idea of the false discovery rate. The first few sections of this chapter introduce the idea of the false discovery rate from the perspective of univariate testing, and the remaining sections apply this idea to penalized regression.

6.1 Introduction

Suppose we carry out a large number, h, of hypothesis tests (in Sections 6.1-6.3 we will use h to denote the number of features as opposed to the usual p used elsewhere in the book to avoid confusion with p-values). Suppose we arrange the outcomes of all these tests into a 2×2 table on the basis of our decision to reject the null hypothesis or not and whether the null hypothesis, in reality, is true or not. Note that the rejection decision is known, but random, while the true status of the null hypothesis is fixed, but unknown. This is represented in Table 6.1.

Classical frequentist statistics is entirely preoccupied with the "horizontal" proportions in Table 6.1: namely, the Type I error A/h_0 and the power B/h_1. In large-scale hypothesis testing, however, we can also focus on the "vertical" proportions A/R, which is known as the false discovery proportion. To prove anything about these proportions, we need to consider their expected values, or rates; thus, we define the *false discovery rate* (FDR) as $\mathbb{E}(A/R)$, and so on for the Type I error rate,

TABLE 6.1
Possible outcomes of hypothesis testing.

		Decision (random)		
		Null	"Discovery"	Total
Reality	Null true	$h_0 - A$	A	h_0
(fixed)	Null false	$h_1 - B$	B	h_1
	Total	$h - R$	R	h

etc. Note that this is a more complicated entity, being the ratio of two random quantities. In particular, some care is needed in the definition as it is possible for R to be zero; in that case the proportion is typically defined to be zero, although there are other possible ways of handing this situation.

The false discovery rate has a more direct interpretation than the Type I error rate, in that it explicitly tells us what fraction of claimed discoveries we can expect to be mere coincidences. This interpretation is unavailable from a frequentist perspective in the low-dimensional case. To calculate it requires specifying the prior probability of the hypothesis being false; in high dimensions, however, we can use the data to derive empirical estimates for the probability that a hypothesis will be false. This is closely related to the idea of an empirical Bayes analysis, as we will see in Section 6.3.

Example 6.1. To illustrate these ideas, we will use data from one of the earliest and most well-known high-dimensional studies: a gene expression study of leukemia patients Golub et al. (1999). In the study, the expression levels of $7,129$ genes were measured for for 72 patients. Of the 72 patients, 47 had acute lymphoblastic leukemia (ALL), while the other 25 had acute myeloid leukemia (AML). Of the two diseases, AML has a considerably worse prognosis: only 26% survive at least 5 years following diagnosis, compared to 68% for ALL.

One way to approach the analysis is to carry out 7,129 two-sample t-tests, obtaining the set of p-values $\{p_j\}_{j=1}^{7,129}$. A critical property of p-values is that for any value u,

$$\mathbb{P}_0\{P \leq u\} \leq u,$$

where P is the p-value and $\mathbb{P}_0$ denotes the probability under the null hypothesis; note that P is the random variable here in the sense that it depends on the data. Thus, for any continuous null distribution,

$$P \sim \text{Unif}(0, 1)$$

under the null hypothesis.

Sometimes, it is more useful to work with z-values than p-values:

$$z_j = \Phi^{-1}(p_j),$$

or, for two-sided tests,

$$z_j = -s_j \Phi^{-1}(p_j/2),$$

where s_j is the sign of the jth test and Φ^{-1} is the inverse of the standard normal CDF. Under H_0, $Z \sim \mathrm{N}(0,1)$. One advantage of z-values for two-tailed tests is that they retain the sign information; in the present context, the z-value tells us whether expression was higher in ALL or AML patients, while the p-value does not. A histogram of the z-values, along with the standard normal density as a reference, is given in Figure 6.1. □

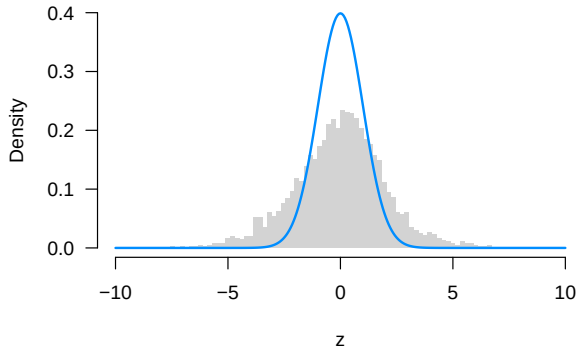

FIGURE 6.1
z-values for the Leukemia data. The blue curve is the density of the assumed null distribution, $\mathrm{N}(0,1)$.

As the figure shows, there are many more features with very large and small z-values than we would expect if the null distribution was true for every feature (and many fewer z-values near zero).

6.2 The Benjamini-Hochberg procedure

Having carried out these hypothesis tests, it would be desirable if we could apply a decision rule as in Table 6.1 that could offer some sort

of guarantee with respect to the resulting false discovery rate. In 1995, Yoav Benjamini and Yosef Hochberg published a paper describing a procedure capable of doing just this (Benjamini and Hochberg, 1995). The procedure was not necessarily new, nor was the term "false discovery rate", but they were the first to provide a rigorous proof that the procedure controlled the FDR, in the sense that $\mathbb{E}(A/R)$ was bounded above. The paper has gone on to become extraordinarily influential, with over 30,000 citations – one of the most highly cited papers in the history of statistics.

The Benjamini-Hochberg (BH) procedure is as follows:

(1) For a fixed value q, let $i_{\max}$ denote the largest index for which

$$p_{(i)} \leq \frac{i}{h} q \tag{6.1}$$

(2) Then reject all hypotheses $H_{0(i)}$ for $i = 1, 2, \ldots, i_{\max}$

The theorem proved by Benjamini and Hochberg is given below without proof. Benjamini and Hochberg's original proof is somewhat lengthy; a clever alternative proof based on martingale theory is given in Storey et al. (2004). The original theorem was proved only for the case of independent tests. Later efforts have extended the results to tests that are weakly dependent; as we will see in this chapter and the next two, correlation among tests/features is an important consideration in high-dimensional inference.

Theorem 6.1. *For independent test statistics and for any configuration of true and false null hypotheses, the BH procedure controls the FDR at q.*

The procedure as we have described it merely sorts tests into discoveries and non-discoveries without providing any relative indication of significance among the tests within a category. A useful FDR-based measure of the significance of a test is given by a quantity known as the *q value*, defined as

$$q_j = \inf\{q : H_{0j} \text{ rejected at FDR} \leq q\}.$$

Another appealing feature of the q value is that it allows us to easily find all the tests that can be rejected at an FDR control of 10% (namely, the tests with $q_j < .1$) or 5% or 20% without having to recalculate anything. In `R`, q values can be obtained from p values via

```
q <- p.adjust(p, method='BH')
```

although keep in mind that the interpretation of a false discovery rate is rather different from the interpretation of a p-value.

For the Leukemia data, the Benjamini-Hochberg procedure allows us to reject 1,635 hypotheses at an FDR of $q = 0.1$. This suggests that approximately 163 of these discoveries may in fact simply be due to chance. By comparison, 734 hypotheses may be rejected at an FDR of $q = 0.01$; here, we would expect only about 7 findings to result from chance alone.

FDR has become a widely accepted methodology, there is no conventional standard for FDR cutoffs the way there is for p-values. Part of the reason for this may be that FDR, being more directly interpretable, is in less need of a standard: an investigator can intuitively weigh the costs of failing to reproduce the findings in 20% of discoveries vs. 5%.

If you look into the details of the proof of Theorem 6.1, you will see that the procedure is conservative. Its actual FDR is

$$\mathbb{E}(A/R) = \frac{h_0}{h} q.$$

Letting $\pi_0 = h_0/h$ denote the fraction of hypotheses that are truly null, one potential improvement to the BH procedure is to estimate π_0. Given such an estimate, we can simply replace h with $\hat{h}_0 = h\hat{\pi}_0$ everywhere it appears in the BH procedure. Several authors have proposed approaches for estimating π_0 (Storey and Tibshirani, 2003; Efron, 2010). The idea is certainly relevant to inference for high-dimensional regression as well, although the details of the various estimation procedures fall outside the scope of this book.

6.3 Empirical Bayes interpretation

The outlook of Section 6.2 was purely frequentist: a procedure was proposed, and our justification was using it was to prove something about its long-run properties with respect to some error rate. The same result, however, can be motivated from several other perspectives, including an empirical Bayes perspective which sheds some additional light on the problem.

Suppose that the observed z-values come from a mixture of two groups: the null group with probability π_0 and density $f_0(z)$, and the non-null group with probability π_1 and density $f_1(z)$. Consider a region

of interest $\mathcal{Z}$ and let $F_0(\mathcal{Z})$ denote the probability, for a feature in the null group, of $z \in \mathcal{Z}$, with

$$F(\mathcal{Z}) = \pi_0 F_0(\mathcal{Z}) + \pi_1 F_1(\mathcal{Z})$$

denoting the marginal probability of $z \in \mathcal{Z}$. Suppose we observe $z \in \mathcal{Z}$ and wish to know the group it belongs to. By applying Bayes' rule, we have

$$\mathbb{P}(\text{Null}|z \in \mathcal{Z}) = \frac{\pi_0 F_0(\mathcal{Z})}{F(\mathcal{Z})}. \tag{6.2}$$

This expression involves three quantities: $F_0(\mathcal{Z}), \pi_0$, and $F(\mathcal{Z})$. Assuming we believe in the theoretical null, $F(\mathcal{Z}) = \Phi(\mathcal{Z})$. We could either estimate π_0, as mentioned in Section 6.2, or we could simply use 1 as an upper bound. Finally, since we observe a large number, h, of z-values, we can use their empirical distribution to estimate $F(\mathcal{Z})$:

$$\hat{F}(\mathcal{Z}) = \frac{\#\{z_j \in \mathcal{Z}\}}{h}.$$

Substituting these expressions into (6.2), we arrive at essentially the same result as Benjamini and Hochberg. To see this, suppose that $\mathcal{Z}$ is of the form $\mathcal{Z} = (-\infty, z_{(i)}]$, where $z_{(i)}$ is the ith ranked z-value. Then

$$\mathbb{P}(\text{Null}|z_{(i)} \in \mathcal{Z}) = \frac{p_{(i)}}{i/h}.$$

In other words, comparing $\mathbb{P}(\text{Null}|z_{(i)} \in \mathcal{Z})$ to an FDR cutoff q, we have the exact same inequality as in (6.1).

Note that the FDR has a nice interpretation here: whereas in frequentist statistics, a common misconception is that $p = 0.02$ means that $\mathbb{P}(H_0|\text{Data}) = 2\%$, here the FDR actually *does* mean that (at least, in the aggregate sense). From the empirical Bayes perspective, the FDR methodology is not a testing procedure with error rates to be controlled, but an estimation problem.

It is often more helpful to view FDR from an estimation perspective. For example, correlated tests pose a considerable challenge with respect to FDR contro. As an estimate, however,

$$\widehat{\text{FDR}} = \hat{\pi}_0 F_0(\mathcal{Z})/\hat{F}(\mathcal{Z}) \tag{6.3}$$

remains accurate even in the presence of correlated tests. Its accuracy depends primarily on the accuracy of $\hat{F}$. Correlation among the z-values introduces little or no bias to the empirical distribution function as an estimate of $F(\mathcal{Z})$. However, it can have a substantial impact on the variance. Thus, correlation among tests does not necessarily render an FDR estimate invalid, but it certainly diminishes our confidence in terms of how close it is to the true false discovery proportion A/R.

6.4 False discoveries in penalized regression under orthogonality

We now turn our attention to the problem of false discoveries in penalized regression. Our derivations will focus on the lasso, but apply to the MCP, SCAD, elastic net, and other penalties with straightforward modifications.

Recall the KKT conditions for the lasso:

$$\begin{aligned} \frac{1}{n}\mathbf{x}_j^T\mathbf{r} &= \lambda \operatorname{sign}(\widehat{\beta}_j) && \text{for all } \widehat{\beta}_j \neq 0 \\ \frac{1}{n}\left|\mathbf{x}_j^T\mathbf{r}\right| &\leq \lambda && \text{for all } \widehat{\beta}_j = 0 \end{aligned}$$

Let $\mathbf{X}_{-j}$ and $\boldsymbol{\beta}_{-j}$ denote the portions of the design matrix and coefficient vector that remain after removing the jth feature, and $\mathbf{r}_j = \mathbf{y} - \mathbf{X}_{-j}\widehat{\boldsymbol{\beta}}_{-j}$ denote the partial residuals with respect to feature j. The KKT conditions thus imply that

$$\begin{aligned} \frac{1}{n}\left|\mathbf{x}_j^T\mathbf{r}_j\right| &> \lambda && \text{for all } \widehat{\beta}_j \neq 0 \\ \frac{1}{n}\left|\mathbf{x}_j^T\mathbf{r}_j\right| &\leq \lambda && \text{for all } \widehat{\beta}_j = 0 \end{aligned} \tag{6.4}$$

and therefore that the probability that variable j is selected is

$$\mathbb{P}\left(\frac{1}{n}\left|\mathbf{x}_j^T\mathbf{r}_j\right| > \lambda\right)$$

This indicates that if we are able to characterize the distribution of $\frac{1}{n}\mathbf{x}_j^T\mathbf{r}_j$ under the null, we can estimate the number of false discoveries in the model. Indeed, this is straightforward in the case of orthonormal design ($\frac{1}{n}\mathbf{X}^T\mathbf{X} = \mathbf{I}$):

$$\frac{1}{n}\mathbf{x}_j^T\mathbf{r}_j \sim N(\beta_j, \sigma^2/n). \tag{6.5}$$

Thus, if $\beta_j = 0$, we have

$$\mathbb{P}\left(\frac{1}{n}\left|\mathbf{x}_j^T\mathbf{r}_j\right| > \lambda\right) = 2\Phi(-\lambda\sqrt{n}/\sigma).$$

These results are related to the expected number of false discoveries in the following theorem.

Theorem 6.2. *Suppose* $\frac{1}{n}\mathbf{X}^T\mathbf{X} = \mathbf{I}$. *Then for any value of* λ,

$$\mathbb{E}\,|\mathcal{S} \cap \mathcal{N}| = 2\,|\mathcal{N}|\,\Phi(-\lambda\sqrt{n}/\sigma),$$

where $\mathcal{S} = \{j : \widehat{\beta}_j \neq 0\}$ *is the set of selected variables and* $\mathcal{N} = \{j : \beta_j = 0\}$ *is the set of null variables.*

To use this as an estimate, the unknown quantities $|\mathcal{N}|$ and σ^2 must be estimated. First, $|\mathcal{N}|$, can be replaced by p, using the total number of variables as an upper bound for the null variables. The variance σ^2 can be estimated by

$$\hat{\sigma}^2 = \frac{\mathbf{r}^T\mathbf{r}}{n - |\mathcal{S}|};$$

this – dividing the residual sum of squares by the degrees of freedom of the lasso – is the simplest approach to estimating the residual variance, but other possibilities exist, as in Section 2.6.1. This implies the following estimate for the expected number of false discoveries:

$$\widehat{\mathrm{FD}} = 2p\Phi(-\sqrt{n}\lambda/\hat{\sigma}) \tag{6.6}$$

and, as an estimate of the false discovery rate:

$$\widehat{\mathrm{FDR}} = \frac{\widehat{\mathrm{FD}}}{S}. \tag{6.7}$$

6.5 False discoveries from a modeling perspective

The case of correlated variables, however, is considerably more complex. Consider the causal diagram presented in Figure 6.2. In this situation, variable A could never be considered a false discovery: it has a direct causal relationship with the outcome Y. Likewise, if variable C were selected, this would obviously count as a false discovery – C has no relationship, direct or indirect, to the outcome.

Variable B, however, occupies a gray area as far as false discoveries are concerned. From a marginal perspective, B is not a false discovery, since it is not independent of Y. However, from a fully conditional perspective, B is a false discovery because B and Y are conditionally independent given A. Finally, from a modeling perspective, we could also adopt the point of view that B is a false discovery only if A is already in the model.

To be more specific, here are the definitions of a false discovery under these three perspectives:

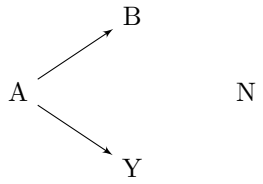

FIGURE 6.2
Causal diagram depicting three types of features and their relationship to the outcome.

- *Marginal* – A selected feature j is a false discovery if it is marginally independent of the outcome: $X_j \perp\!\!\!\perp Y$.
- *Fully conditional* – A selected feature j is a false discovery if it is independent of the outcome given all other features: $X_j \perp\!\!\!\perp Y|\{X_k\}_{k\neq j}$.
- *Partially conditional* – A selected feature j is a false discovery if it is independent of the outcome given the other features in the model: $X_j \perp\!\!\!\perp Y|\{X_k : k \in \mathcal{M}_{j-}\}$, where $\mathcal{M}_{j-}$ denotes the set of features with nonzero coefficients in the model, excluding feature j.

As we will see in the next few chapters, estimating the number of false selections from fully conditional and partially conditional perspectives tends to require fairly complex methods. In this chapter, we consider only the weaker, marginal definition of false discovery, and will see how simple approaches like the one derived in Section 6.4 may still be used to estimate the number of false selections arising from variables like N.

In this chapter, we define a *noise feature* to be a variable like N, that has no causal path (direct or indirect) between it and the outcome, and the *marginal false discovery rate* as the proportion of selected features that are noise variables. Again, this definition is consistent with how false discoveries are defined in univariate testing, but differs from conditional approaches that we will discuss in later chapters.

The marginal perspective has several advantages. First, when two variables (like A and B) are correlated, it is very difficult to distinguish between which of them is driving changes in Y and which is merely correlated with Y. This can lead approaches that define false discoveries according to $\beta_j = 0$ to be very conservative, especially in high dimensions.

Second, in many scientific applications, discovering variables like B

is not problematic. For example, two genetic variants in close proximity to each other on a chromosome will be highly correlated. Although it is obviously desirable to identify which of the two is the causal variant, locating a nearby variant is also an important scientific achievement, as it narrows the search to a small region of the genome for future follow-up studies.

The final advantage is clarity. From the marginal perspective, whether or not a variable is a false discovery depends only on the relationships between it and the outcome, not whether any other variables have been included in the model or not. For example, applying the partially conditional perspective to penalized regression means that feature j may be a false discovery for some values of λ, but not for others.

6.6 Marginal false discovery rates

The orthogonality of Section 6.4 clearly does not hold in general. Thankfully, the assumptions in that section can be relaxed in two important ways that make the results more widely applicable. First, the predictors do not have to be strictly orthogonal in order for the estimator to work; they can simply be uncorrelated. Second, this condition of being uncorrelated applies only to the noise features – i.e., the variables like N in Figure 6.2; variables like A and B can have any correlation structure.

To make these statements concrete, let $\mathcal{A}, \mathcal{N}$ partition $\{1, 2, \ldots, p\}$ such that $\beta_j = 0$ for all $j \in \mathcal{N}$ and the following condition holds:

$$\lim_{n\to\infty} \frac{1}{n}\mathbf{X}'\mathbf{X} = \begin{bmatrix} \Sigma_{\mathcal{A}} & \mathbf{0} \\ \mathbf{0} & \Sigma_{\mathcal{N}} \end{bmatrix}.$$

Under this definition, the opening remarks of this section can be stated precisely in the following theorem.

Theorem 6.3. *Suppose* $\frac{1}{n}\mathbf{X}_{\mathcal{N}}^T\mathbf{X}_{\mathcal{N}} \to \Sigma_{\mathcal{N}} = \mathbf{I}$. *Then for any* $j \in \mathcal{N}$ *and for any* λ,

$$\frac{1}{\sqrt{n}}\mathbf{x}_j'\mathbf{r}_j \xrightarrow{d} N(0, \sigma^2).$$

Theorem 6.3 shows that if the noise features are uncorrelated, $\frac{1}{n}\mathbf{x}_j'\mathbf{r}_j$ behaves precisely as it did (6.5) in Section 6.4. Thus, estimators (6.6) and (6.7) are just as valid here as they are in the orthonormal case.

Example 6.2. To illustrate the consequences of Theorem 6.3, let us

carry out the following simulation study, with both a "low-dimensional" ($n > p$) and "high-dimensional" ($n < p$) component. Motivated by Figure 6.2, three types of features will be included:

- Causative: Six variables with $\beta_j = 1$
- Correlated: Each causative feature is correlated ($\rho = 0.5$) with m other features; $m = 2$ for the low-dimensional case and 9 for the high-dimensional case
- Noise: Independent noise features are added to bring the total number of variables up to 60 in the low-dimensional case and 600 in the high-dimensional case

The causative, correlated, and noise features correspond to variables A, B, and N, respectively, in Figure 6.2. In each setting, the sample size was $n = 100$, while the total number of causative/correlated/noise features was 6/12/42 for the low-dimensional setting and 6/54/540 for the high-dimensional setting. □

The results of the simulation are shown in Figure 6.3.

As Theorem 6.3 implies, estimators (6.6) and (6.7) are quite accurate, on average, when the noise features are independent. The estimated number of marginal false discoveries and the marginal false discovery rate (mFDR) are both somewhat conservative, as we would expect from using p as an upper bound for the number of noise features (e.g., in the high-dimensional case, $p = 600$ but $|\mathcal{N}| = 540$). However, the effect is slight in this setting. For example, in the high-dimensional case at $\lambda = 0.55$, the actual mFDR was 5%, while the estimated rate was 6.5%.

Being able to estimate marginal false discovery rates means we can use them to select the regularization parameter λ. For example, we could choose λ to be the smallest value of λ such that $\widehat{\text{mFDR}}(\lambda) < 0.1$. Figure 6.4 compares this approach, "Lasso (mFDR)", with the method we have primarily relied upon thus far, cross-validation, as well as with univariate testing (i.e., marginal regression). For each method, the number of each type of feature the method selects on average is shown as a stacked bar chart. For Lasso (mFDR) and univariate testing, the nominal false discovery rates were set to 10%.

It is worth noting that both Lasso (mFDR and univariate testing limit the fraction of selections due to noise features (5% and 7%, respectively, in the $p = 60$ simulation) to the nominal rate, but claim nothing about the fraction arising from correlated features. This is obvious from the figure for univariate testing; for Lasso (mFDR), the fraction of selected features coming from either the Correlated or Noise groups (i.e., all features with $\beta_j = 0$) was 17% in the low-dimensional setting and

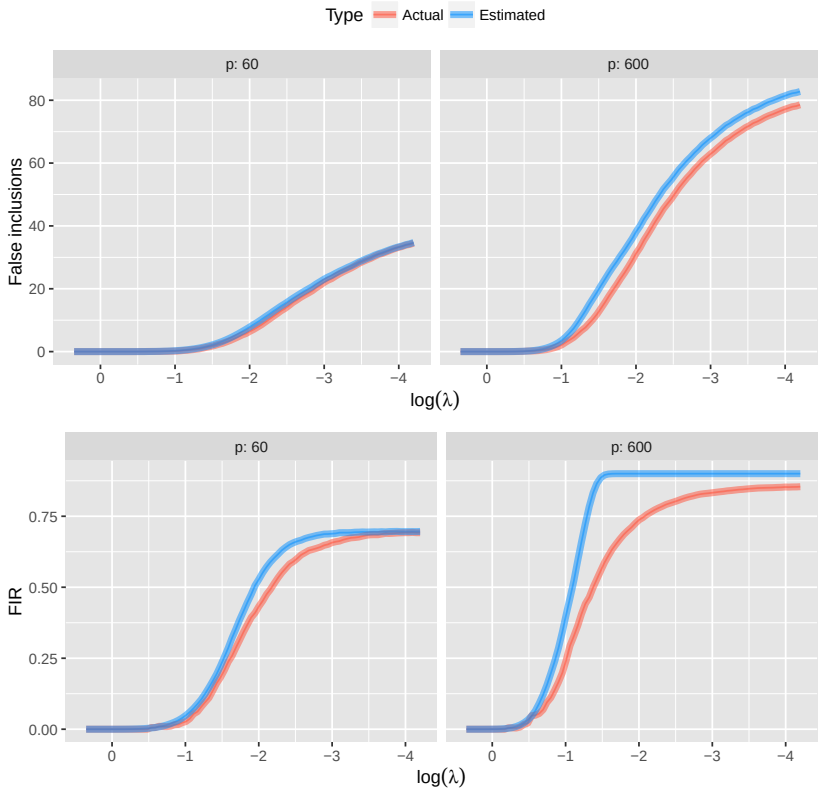

FIGURE 6.3
Accuracy of estimators (6.6) and (6.7) in the case of independent noise features.

23% in the high-dimensional setting. Nevertheless, compared to univariate testing, the Lasso (mFDR) approach has two distinct advantages. Figure 6.4 shows that using a penalized regression approach both diminishes the number of merely correlated features selected and improves power to detect the truly causative features.

Figure 6.4 also illustrates the lack of protection provided by cross-validation against the selection of noise features. In each setting, over half of the variables selected by the lasso with cross-validation were in fact mere noise. NOTE: CONNECT TO THEORY CHAPTER. The mFDR estimator provides an attractive way to assess this phenomenon for a specific data set; we will see its utility for real data in Section 6.7.

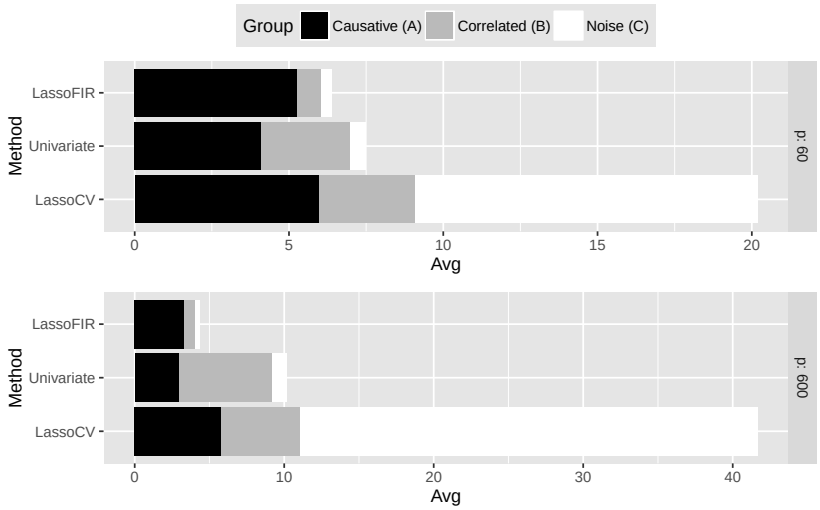

FIGURE 6.4
Average number of each type of feature selected by various methods for the simulation setup in Example 6.2.

The preceding results are something of a best case scenario for estimator (6.7), since the variables in $\mathcal{N}$ were independent and we know by Theorem 6.3 that the estimator is valid in this case. When noise features are correlated, the estimator becomes somewhat conservative. In most situations, however, the effect is relatively small.

Example 6.3. To illustrate, let us carry out the following simulation. The generating model contains 6 independent causative features and 494 correlated noise features, with a 1:1 signal-to-noise ratio ($n = 100$, $p = 500$, $R^2 = 0.5$). The noise features are given an autoregressive correlation structure with $\text{Cor}(\mathbf{x}_j, \mathbf{x}_k) = 0.8^{|j-k|}$. □

The results of the simulation are shown in Figure 6.5. Compared to Figure 6.3, the mFDR estimates are somewhat more conservative in this case, although still accurate enough to be useful in practice. For example, at $\lambda = 0.43$, the true mFDR was 14%, while the estimated rate according to (6.6) was 20%.

This illustrates that although its derivation was based on independent noise features, the mFDR estimator is reasonably robust to the presence of correlation. Furthermore, to the extent that it is inaccurate, it provides a conservative estimate of the mFDR, and thus offers some measure of control over the false discovery rate. To understand why the

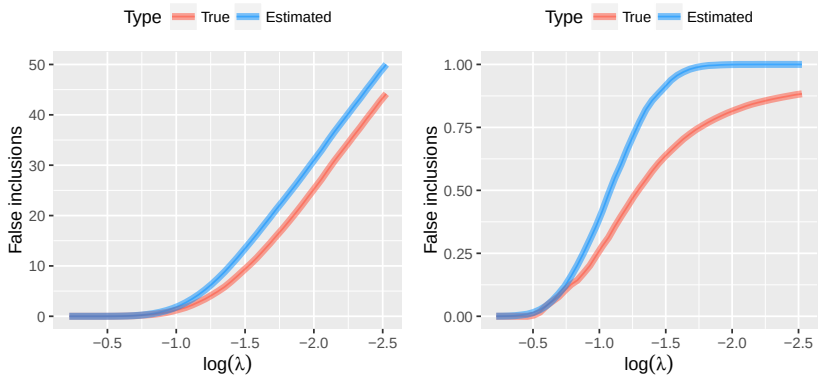

FIGURE 6.5
Accuracy of estimators (6.6) and (6.7) in the case of (autoregressive) correlated noise features.

estimator is conservative in the presence of correlation, note that the lasso (and most other penalized regression methods) will tend to select a single feature rather than both when the two are correlated. Thus, the uncorrelated case is not just mathematically convenient, it also represents a worst case scenario with respect to the number of noise features that we can expect to be falsely selected.

6.7 Case study: Breast cancer gene expression study

To see how this works with real data, let's take a look at the breast cancer TCGA data; recall that $n = 536$ and $p = 17,322$ in this example. We can fit a lasso model with

```
fit <- ncvreg(X, y, penalty="lasso")
```

and then calculate marginal false discovery rates for the fitted model object using the `fir` function (this function is available only in `ncvreg`, not `glmnet`):

```
obj <- mfdr(fit)
```

The resulting `mfdr` object is similar to a data frame, with elements `S`, the number of selected variables, `EF`, the expected number of noise

features, and `mFDR`, the estimated marginal false discovery rate. So, for example, to display the results for largest values of λ such that mFDR $<$ 10%, we can submit:

```
> tail(obj[obj$mFDR < .1,])
             EF  S       mFDR
0.0730 1.036495 48 0.02159364
0.0708 1.453916 50 0.02907831
0.0687 2.020152 52 0.03884907
0.0666 2.787655 52 0.05360874
0.0647 3.795806 54 0.07029270
0.0627 5.054797 55 0.09190540
```

The rows here are labeled with their corresponding λ value. From the output, we can see that for $\lambda = 0.0627$, we can select 55 features whereas only 5.05 would have been expected if all features were independent noise, an mFDR of 9%.

The resulting object can also be plotted with

```
plot(obj)             # Left side
plot(obj, type="EF")  # Right side
```

which produces the output shown in Figure 6.6 (although the figure uses the `log.l=TRUE` option to plot on the log scale). Both the plots and the tabular output from `R` show that many genes are predictive of BRCA1 expression – we can safely select 55 variables before the mFDR exceeds 10%, or 52 before the mFDR exceeds 5%. This make sense scientifically, as a large number of genes are known to affect BRCA1 expression through a variety of mechanisms, and the sample size here is sufficient that we should be able to identify many of them.

It is worth comparing these results to the selection of λ by cross-validation (CV). For the TCGA data, $\lambda = 0.042$ minimizes the CV error. The estimated mFDR at this value, however, is 77%, indicating that although this value of λ may be attractive from a prediction perspective, we cannot be confident that the variables selected by the model are truly related to the outcome. This is exactly what we would expect from THEORYCHAPTER: while the lasso has attractive variable selection and prediction properties, it cannot achieve both those aims simultaneously. In particular, if we select λ to minimize prediction error, we can expect to select a number of noise features. The mFDR estimates illustrate this concretely: $\lambda = 0.0436$ produces accurate predictions, but a larger value, $\lambda = 0.0627$, is required in order to have confidence that noise features have been eliminated from the set of selected variables.

Because the mFDR estimator follows directly from the KKT conditions, it is straightforward to extend to other penalties. In fact, the

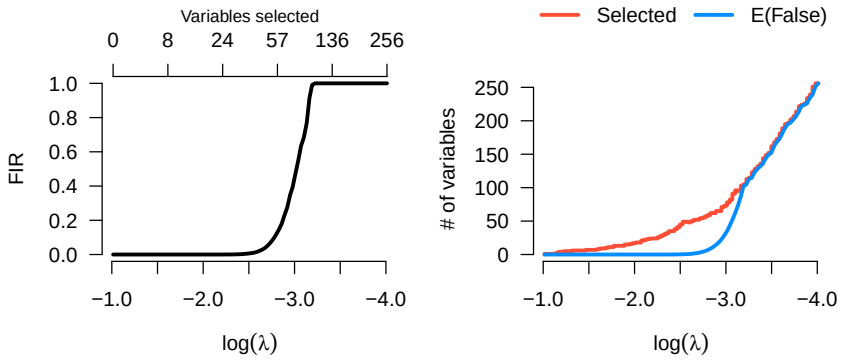

FIGURE 6.6
Marginal false discovery rate estimates for a lasso model applied to the breast cancer TCGA data.

KKT conditions for many penalties, such as MCP, SCAD, elastic net, and MCP/SCAD + ridge, lead to the same expression (6.4) and therefore the same mFDR estimator.

For the sake of comparison, let's also an MCP model to the TCGA data. As we have seen, for the value of λ minimizing CV error, the two methods have similar predictive accuracy, with lasso slightly higher (0.61 vs. 0.58) while the MCP model uses far fewer features (38 vs. 96). As we would expect from THEORYCHAPTER, the mFDR of the MCP model is much lower than that of the lasso for these values of λ: 5% compared to 77%. Obviously, we can restrict the lasso to an mFDR to 5% to make its selections comparable in reliability to those of the lasso, but in doing so, its predictive accuracy falls to $R^2 = 0.57$.

We have seen this pattern in previous chapters for simulated data sets and theoretical results, but mFDR estimates offer a way to observe and assess this tradeoff between prediction accuracy and variable selection accuracy in the analysis of real data. In the chapters to come, we will explore a variety of inferential approaches for penalized regression models that provide more comprehensive results, such as confidence intervals for all model parameters, but the mFDR is a simple, useful summary measure of feature selection reliability that is often useful to look at when assessing the fit of a model.

6.8 Bibliographical notes

This section will include the bibliographical notes on the materials presented in this chapter.

6.9 Exercises

6.1. *Conditional distribution for random knockoffs.* Suppose that the joint distribution $[X^\top \ \tilde{X}^\top]^\top \sim \mathrm{N}(\mathbf{0}, \mathbf{G})$ where

$$\mathbf{G} = \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma} - \mathbf{S} \\ \boldsymbol{\Sigma} - \mathbf{S} & \boldsymbol{\Sigma} \end{bmatrix};$$

here $\mathbf{S}$ is a diagonal matrix with entries $\{s_j\}$ satisfying the constraint that $\mathbf{G}$ is positive definite. Derive the conditional distribution of $\tilde{X}_i | X_i$, where X_i is the p-dimensional vector of features for observation i and $\tilde{X}_i$ the corresponding knockoffs.

6.2. *Selective inference in the p=2 case.* Suppose there are only two features and (at a fixed value of λ) lasso selects a model such that $\beta_1 > 0$ and $\beta_2 = 0$. Express the condition that this model was selected in the polyhedral form $\mathbf{A}\mathbf{y} \leq \mathbf{b}$, giving expressions for $\mathbf{A}$ and $\mathbf{b}$.

6.3. *HIV drug resistance study.* Although there are many drugs that have been approved for treating Human Immunodeficiency Virus (HIV) infection, one of the hallmarks of the virus is its ability to rapidly mutate and gain resistance to these drugs. In this study, isolates of HIV were extracted from infected individuals and sequenced. These isolates were also tested for their resistance to various drugs used in HIV therapy. The scientific goal of the project is to determine which mutations are associated with drug resistance, thereby helping to develop new antiretroviral drugs and to optimize the use of existing drugs.

The full study examined many drugs; the data set `Rhee2006` contains the results for one specific drug, Nelfinavir, a protease inhibitor, and the presence of mutations in the protease gene, which potentially confer resistance to the drug.

(a) Using a lasso-penalized linear regression model, choose three inferential approaches to determine mutations that confer resistance

to the drug Nelfinavir (e.g., marginal FDR, semi-penalized likelihood ratio test, stability selection, bootstrapping, sample splitting, knockoff filter, selective inference, covariance test). Write a "Methods" paragraph that describes how you implemented each approach (e.g., specific options you chose); roughly one or two sentences per method.

(b) Write a "Results" paragraph that compares the overall number of mutations found to be significant by each approach.

(c) Construct a figure that communicates the degree of resistance conferred by each mutation determined to be significant (by a method of your choice). The reader should be able to determine from the figure that, say, P13.V was the third-most important mutation in terms of conferring resistance.

(d) Find (at least) one mutation for which the methods do not agree: one method indicates that there is significant evidence for the mutation conferring resistance whereas another method indicates that the evidence is not significant. Provide some commentary on why you think the disagreement occurs. The commentary should be more in-depth than "method A is more powerful than method B"; why is this *specific* variable affected in the way that it is?

7

Inference for low-dimensional parameters

7.1 Inference for treatment effects in the presence of nuisance parameters

The penalized methods discussed in the previous chapters only yield point estimates of the parameters, but do not provide a way for making statistical inference.

In many important applications, the primary focus is on a low-dimensional parameter. For example, in the YSPORE data analysis, we are interested in the effect of BRAF inhibitors on prognosis; In the analysis of TCGA data, we are mostly interested in a CNV's effect on its corresponding gene expression. Under Aim 1, we will develop methods for the statistical inference of such low-dimensional effects in a class of important models including the linear and generalized linear models.

Suppose that we have observations $\{(y_i, \mathbf{x}_i, \mathbf{z}_i), i = 1, \ldots, n\}$, iid realizations of $(y, \mathbf{x}, \mathbf{z})$, where y is a response variable, $\mathbf{x}$ is a d-dimensional covariate of main interest, and $\mathbf{z}$ is a q-dimensional vector containing possibly confounding variables. An important special case is in clinical trials or observational studies where $\mathbf{x}$ is a binary covariate representing two treatments and $\mathbf{z}$ includes genomic measurements and other potential risk factors. The goal is to estimate the effect of $\mathbf{x}$, denoted by $\boldsymbol{\beta}$, while taking into account the effect of $\mathbf{z}$, denoted by $\boldsymbol{\eta}$.

We are interested in the case where q is large, possibly much larger than the sample size n.

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{Z}\boldsymbol{\xi} + \boldsymbol{\varepsilon},$$

where $\mathbf{X}$ is an $n \times d$ design matrix of covariates of main interest, $\mathbf{Z}$ is an $n \times q$ matrix of other covariates that are of less interest but may also be related to $\mathbf{y}$. To obtain valid inference about $\boldsymbol{\theta}$, it is necessary to take into account the effects of $\mathbf{Z}$ in the model. So the problem is to estimate $\boldsymbol{\theta}$ in the presence of the nuisance parameter $\boldsymbol{\xi}$.

We first consider the case where the combined design matrix $(\mathbf{X}, \mathbf{Z})$ is full column rank. The least squares estimator

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\xi}}) = \arg\min_{\boldsymbol{\theta}, \boldsymbol{\xi}} Q(\boldsymbol{\theta}, \boldsymbol{\xi}) = \frac{1}{2n} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta} - \mathbf{Z}\boldsymbol{\xi}\|^2. \tag{7.1}$$

The solution can be calculated using the general result for least squares by considering the combined parameter $(\boldsymbol{\theta}, \boldsymbol{\xi})$ and the combined design matrix $(\mathbf{X}, \mathbf{Z})$. However, it is instructive to consider the structure of the problem and focus on $\boldsymbol{\theta}$, the parameter of main interest. We describe three ways to solve this minimization problem and obtain the estimator of $\boldsymbol{\theta}$: (a) direct solution of the minimization problem; (b) profile least squares; (c) efficient score approach. These will provide a basis for the methods for estimating the low-dimensional parameter $\boldsymbol{\theta}$ in high-dimensional models.

(a) Direct solution. To directly solve (7.1), we calculate the partial derivatives of $Q(\boldsymbol{\theta}, \boldsymbol{\xi})$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ and set them to zero. This leads to the normal equations

$$\begin{cases} \mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta} - \mathbf{Z}\boldsymbol{\xi}) = \mathbf{0} \\ \mathbf{Z}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta} - \mathbf{Z}\boldsymbol{\xi}) = \mathbf{0} \end{cases} \tag{7.2}$$

Solving the second equation gives $\boldsymbol{\xi} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})$. Thus

$$\mathbf{Z}\boldsymbol{\xi} = \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}).$$

Let $\mathbf{P}_\mathbf{Z} = \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T$ be the projection matrix into the column space of $\mathbf{Z}$ and let $\mathbf{Q}_\mathbf{Z} = \mathbf{I}_n - \mathbf{P}_\mathbf{Z}$, where $\mathbf{I}_n$ is an $n \times n$ identity matrix.

With these notation, we can write

$$\mathbf{Z}\boldsymbol{\xi} = \mathbf{P}_\mathbf{Z}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}).$$

Substituting this into the first equation of (7.2) gives

$$X^T\mathbf{Q}_\mathbf{Z}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = 0. \tag{7.3}$$

It follows that

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{Q}_\mathbf{Z}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Q}_\mathbf{Z}\mathbf{y}, \tag{7.4}$$

(b) Profile least squares. The profile approach is a helpful way to solve a joint minimization problem

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d, \boldsymbol{\xi} \in \mathbb{R}^q} Q(\boldsymbol{\theta}, \boldsymbol{\xi}),$$

where Q can be a general statistical criterion, including the least squares criterion in (7.1) This approach is based on the idea that a joint minimization problem can be solved successively as follows

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d,\boldsymbol{\xi}\in\mathbb{R}^q} Q(\boldsymbol{\theta},\boldsymbol{\xi}) = \min_{\boldsymbol{\theta}\in\mathbb{R}^d}\{Q(\boldsymbol{\theta}) \equiv \min_{\boldsymbol{\xi}\in\mathbb{R}^q} Q(\boldsymbol{\theta},\boldsymbol{\xi})\}.$$

For a given $\boldsymbol{\theta}$, let

$$\boldsymbol{\xi}(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\xi}\in\mathbb{R}^q} Q(\boldsymbol{\theta},\boldsymbol{\xi}).$$

The profiled criterion for $\boldsymbol{\theta}$ is

$$Q(\boldsymbol{\theta}) = Q(\boldsymbol{\theta},\boldsymbol{\xi}(\boldsymbol{\theta})).$$

The profiled estimator of $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} Q(\boldsymbol{\theta}).$$

Then the estimator of $\boldsymbol{\xi}$ is $\hat{\boldsymbol{\xi}} = \boldsymbol{\xi}(\hat{\boldsymbol{\theta}})$.

For the Q given in (7.1), for a given $\boldsymbol{\theta}$, the value of $\boldsymbol{\xi}$ that minimizes $Q(\boldsymbol{\theta},\cdot)$ is simply the OLS estimator with $\mathbf{y}-\mathbf{X}\boldsymbol{\theta}$ as the response vector and $\mathbf{Z}$ as the design matrix, that is,

$$\hat{\boldsymbol{\xi}}(\boldsymbol{\theta}) = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T(\mathbf{y}-\mathbf{Z}\boldsymbol{\theta}).$$

Thus

$$\mathbf{y}-\mathbf{Z}\hat{\boldsymbol{\xi}}(\boldsymbol{\theta})-\mathbf{Z}\boldsymbol{\theta} = (\mathbf{I}_n - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T)(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}).$$

This can be written as

$$\mathbf{y}-\mathbf{Z}\hat{\boldsymbol{\xi}}(\boldsymbol{\theta})-\mathbf{X}\boldsymbol{\theta} = \mathbf{Q}_\mathbf{Z}(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}).$$

Substituting this into (7.1) to get the profile least squares criterion for $\boldsymbol{\theta}$,

$$Q(\boldsymbol{\theta}) \equiv Q(\boldsymbol{\theta},\hat{\boldsymbol{\xi}}(\boldsymbol{\theta}) = \frac{1}{2n}\|\mathbf{Q}_\mathbf{Z}(\mathbf{y}-\mathbf{X}\boldsymbol{\theta})\|^2.$$

Thus the corresponding normal equation is

$$(Q_\mathbf{Z}\mathbf{X})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}) = 0.$$

This is exactly the same as (7.3) and leads to the same estimator of $\boldsymbol{\theta}$ as (7.4) from the direct solution.

The roles of $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ can be reversed in the profiling. That is, first for a given $\boldsymbol{\xi}$ we obtain the profile least squares estimator $\hat{\boldsymbol{\theta}}(\boldsymbol{\xi})$. Then use the same calculation as above to obtain the profile least squares criterion for $\boldsymbol{\xi}$

$$Q(\boldsymbol{\xi}) \equiv Q(\boldsymbol{\theta}(\boldsymbol{\xi}),\boldsymbol{\xi}) = \frac{1}{2n}\|\mathbf{Q}_\mathbf{X}(\mathbf{y}-\mathbf{Z}\boldsymbol{\xi})\|^2. \tag{7.5}$$

Let

$$\hat{\boldsymbol{\xi}} = \arg\min_{\boldsymbol{\xi}\in\mathbb{R}^q} Q(\boldsymbol{\xi}).$$

Then the OLS estimator of $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\xi}}).$$

It can be verified that this expression is the same as that given in (7.4).

It turns out it is more convenient to modify this way of profiling for a high-dimensional nuisance parameter $\boldsymbol{\xi}$.

Efficient score approach. The score functions for $\boldsymbol{\theta}$ and $\boldsymbol{\xi}$ are

$$\dot{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta},\boldsymbol{\xi}) \equiv \frac{\partial}{\partial\boldsymbol{\theta}}Q(\boldsymbol{\theta},\boldsymbol{\xi}) = -\frac{1}{n}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta} - \mathbf{Z}\boldsymbol{\xi}) = -\frac{1}{n}\mathbf{X}^T\boldsymbol{\varepsilon}$$
$$\dot{Q}_{\boldsymbol{\xi}}(\boldsymbol{\theta},\boldsymbol{\xi}) \equiv \frac{\partial}{\partial\boldsymbol{\xi}}Q(\boldsymbol{\theta},\boldsymbol{\xi}) = -\frac{1}{n}\mathbf{Z}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\theta} - \mathbf{Z}\boldsymbol{\xi}) = -\frac{1}{n}\mathbf{Z}^T\boldsymbol{\varepsilon}$$

To calculate the efficient score function for $\boldsymbol{\theta}$, we need to find the projection of $\dot{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta},\boldsymbol{\xi})$ onto the linear space spanned by $\dot{Q}_{\boldsymbol{\xi}}(\boldsymbol{\theta},\boldsymbol{\xi})$. That is, we need to find a $q \times d$ matrix $\mathbf{A}$ that minimizes

$$\mathrm{E}\|\dot{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta},\boldsymbol{\xi}) - \mathbf{A}^T\dot{Q}_{\boldsymbol{\xi}}(\boldsymbol{\theta},\boldsymbol{\xi})\|^2.$$

Using the expressions given above, this amounts to finding $\mathbf{A}$ that minimizes

$$\mathrm{E}\|\mathbf{X}^T\boldsymbol{\varepsilon} - \mathbf{A}^T\mathbf{Z}^T\boldsymbol{\varepsilon}\|^2$$

Since $\mathrm{E}\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T = \sigma^2\mathbf{I}_n$, it is equivalent to finding $\mathbf{A}$ that minimizes

$$\sum_{i=1}^{n}\|\mathbf{x}_i - \mathbf{A}^T\mathbf{z}_i\|^2. \tag{7.6}$$

Thus $\mathbf{A}$ must satisfy

$$-2\sum_{i=1}^{n}(\mathbf{x}_i - A^T\mathbf{z}_i)\mathbf{z}_i^T = 0.$$

Note that $\sum_{i=1}^n \mathbf{x}_i\mathbf{z}_i^T = \mathbf{X}^T\mathbf{Z}$ and $\sum_{i=1}^n \mathbf{z}_i\mathbf{z}_i^T = \mathbf{Z}^T\mathbf{Z}$, this equation can be written as

$$\mathbf{X}^T\mathbf{Z} - \mathbf{A}^T\mathbf{Z}^T\mathbf{Z} = \mathbf{0}.$$

Therefore,

$$\mathbf{A}^T = \mathbf{X}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}.$$

So the efficient score function for $\boldsymbol{\theta}$ is

$$\dot{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta},\boldsymbol{\xi}) - \mathbf{A}^T\dot{Q}_{\boldsymbol{\xi}}(\boldsymbol{\theta},\boldsymbol{\xi}) = -\frac{1}{n}(\mathbf{X}^T - \mathbf{A}^T\mathbf{Z}^T)(\mathbf{y} - \mathbf{X}\boldsymbol{\theta} - \mathbf{Z}\boldsymbol{\xi}) \tag{7.7}$$

Since

$$\mathbf{X}^T - \mathbf{A}^T\mathbf{Z}^T = \mathbf{X}^T - \mathbf{X}^T\mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T = \mathbf{X}^T(\mathbf{I}_n - \mathbf{P}_{\mathbf{Z}}) = \mathbf{X}^T\mathbf{Q}_{\mathbf{Z}}.$$

The efficient score function can be written as

$$\dot{Q}_{\boldsymbol{\theta}}(\boldsymbol{\theta},\boldsymbol{\xi}) - \mathbf{A}^T\dot{Q}_{\boldsymbol{\xi}}(\boldsymbol{\theta},\boldsymbol{\xi}) = -\frac{1}{n}\mathbf{X}^T\mathbf{Q}_{\mathbf{Z}}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}).$$

The efficient score estimator is the solution to the (efficient score) equation

$$\mathbf{X}^T\mathbf{Q}_{\mathbf{Z}}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) = \mathbf{0}.$$

7.2 Semi-penalized estimator

Now suppose the $n \times d$ design matrix $\mathbf{X}$ is full column rank, but the $n \times q$ design matrix $\mathbf{Z}$ is not full column rank. This can happen when d is small relative to n, but q is large or even larger than n. This is the case where $\mathbf{X}$ consists of a few covariates of main interest, but Z contains a large number possibly confounding covaraites. In this case, the three approaches described above for estimating $\boldsymbol{\theta}$ no longer work. It is natural to apply penalized approach to dealing with the high-dimensionality of $\mathbf{Z}$. Specifically, we consider a semi-penalized least squares criterion

$$\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta} - \mathbf{Z}\boldsymbol{\xi}\|^2 + \sum_{j=1}^{q} P(\xi_j;\lambda). \tag{7.8}$$

An important feature of this criterion is that $\boldsymbol{\theta}$ is not penalized. Indeed, since the main interest is in making statistical inference about $\boldsymbol{\theta}$, there is no need to impose sparsity on it. Also, not penalizing $\boldsymbol{\theta}$ will reduce bias and lead to an estimator that is asymptotically normal. This will make inference about this parameter possible.

Using a similar argument as in deriving the second form of profile least squares estimator of $\boldsymbol{\theta}$, for a given $\boldsymbol{\xi}$, the OLS estimator of $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\xi}) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y} - \mathbf{Z}\boldsymbol{\xi}). \tag{7.9}$$

Substituting this expression into (7.8) yields the penalized profile least

squares criterion

$$\frac{1}{2n}\|\mathbf{Q}_{\mathbf{X}}(\mathbf{y}-\mathbf{Z}\boldsymbol{\xi})\|^2+\sum_{j=1}^{q}P(\xi_j;\lambda). \tag{7.10}$$

Indeed, we can also start from (7.5) directly by imposing a penalty function on $\boldsymbol{\xi}$ to get a regularized estimator of $\boldsymbol{\xi}$, then obtain an estimator of $\boldsymbol{\theta}$ through (7.9).

The criterion (7.10) can be written in a standard penalized least squares form by letting $\mathbf{y}_*=\mathbf{Q}_{\mathbf{X}}\mathbf{y}$ and $\mathbf{Z}_*=\mathbf{Q}_{\mathbf{X}}\mathbf{Z}$,

$$\frac{1}{2n}\|\mathbf{y}_*-\mathbf{Z}_*\boldsymbol{\xi})\|^2+\sum_{j=1}^{q}P(\xi_j;\lambda).$$

The `R` package `ncvreg` can be used to compute the solution path

$$\hat{\boldsymbol{\xi}}(\lambda)=\arg\min_{\boldsymbol{\xi}\in\mathbb{R}^q}\frac{1}{2n}\|\mathbf{y}_*-\mathbf{Z}_*\boldsymbol{\xi})\|^2+\sum_{j=1}^{q}P(\xi_j;\lambda).$$

Then for a given λ the estimator of $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}}(\lambda)=(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{y}-\mathbf{Z}\hat{\boldsymbol{\xi}}(\lambda)).$$

The main computational task in this procedure in computing the solution path $\hat{\boldsymbol{\xi}}(\lambda)$.

An alternative expression for $\hat{\boldsymbol{\theta}}$ that sheds some insights into the properties of this estimator is as follows. Let $\hat{S}(\lambda)=\{j:\hat{\xi}_j(\lambda)\neq 0\}$ be the number of nonzero elements in $\hat{\boldsymbol{\xi}}(\lambda)$. For simplicity, write $\hat{S}=\hat{S}(\lambda)$. For $A\subset\{1,\ldots,q\}$, let $\mathbf{Z}_A$ be the matrix consisting of the columns of $\mathbf{Z}$ whose indices are in A. Denote the projection matrix into the column space of $\mathbf{Z}_A$ by $\mathbf{P}_A=\mathbf{Z}_A(\mathbf{Z}_A^T\mathbf{Z}_A)^{-1}\mathbf{Z}_A^T$. Let $\mathbf{Q}_{\hat{S}}=I-\mathbf{P}_{\hat{S}}$, and let $\Sigma_{\hat{S}}=\mathbf{Z}_{\hat{S}}^T\mathbf{Z}_{\hat{S}}/n$. It is shown in Section 7.7 that $\hat{\boldsymbol{\theta}}$ can also be written as

$$\hat{\boldsymbol{\theta}}(\lambda)=(\mathbf{X}^T\mathbf{Q}_{\hat{S}}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Q}_{\hat{S}}\mathbf{y}-(\mathbf{X}^T\mathbf{Q}_{\hat{S}}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Z}_{\hat{S}}\Sigma_{\hat{S}}^{-1}\dot{P}(\hat{\boldsymbol{\xi}}_{\hat{S}};\lambda), \tag{7.11}$$

where the second term on the right hand side represents the bias introduced by penalization and correlation between $\mathbf{X}$ and $\mathbf{Z}_{\hat{S}}$.

If the nonzero coefficients of $\boldsymbol{\xi}$ are bigger than $\gamma\lambda$ and the estimator $\hat{\boldsymbol{\xi}}_{\hat{S}}$ is consistent so that $\hat{\xi}_j\geq\gamma\lambda$ for all $j\in\hat{S}$ with high probability, then since the derivative of MCP

$$\dot{P}(t;\lambda)=\lambda\{1-|t|/(\gamma\lambda)\}_+\text{sign}(t),$$

we have $\dot{P}(\boldsymbol{\beta}_{\hat{S}_j};\lambda)=0$ with high probability. In addition, under suitable

conditions the MCP estimator is selection consistent in the sense that $\hat{S}$ equals $S \equiv \{j : \xi_j \neq 0\}$ with high probability. Thus

$$\hat{\boldsymbol{\theta}} \approx (\mathbf{X}^T\mathbf{Q}_S\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Q}_S\mathbf{y} = \boldsymbol{\theta} + (\mathbf{X}^T\mathbf{Q}_S\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Q}_S\boldsymbol{\varepsilon}.$$

It follows that $\widehat{\boldsymbol{\beta}}$ is approximately distributed as $\mathrm{N}(\boldsymbol{\theta}, \sigma^2(\mathbf{X}^T\mathbf{Q}_S\mathbf{X})^{-1})$.

It is important to note that the above calculation also shows that the use of Lasso in the semi-penalized estimation will not lead to root-n consistent and asymptotically normal estimator of $\boldsymbol{\beta}$.

7.3 Regularized efficient score estimator

When $\mathbf{Z}$ has full column rank, the minimizer of (7.6) is well defined and unique. According to (7.7), the efficient score is

$$-\frac{1}{n}(\mathbf{X}^T - \mathbf{A}^T\mathbf{Z}^T)(\mathbf{y} - \mathbf{X}\boldsymbol{\theta} - \mathbf{Z}\boldsymbol{\xi})$$

where $\mathbf{A}$ minimizes

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

However, in $q > n$ models, the solution is not unique.

First consider the case where $d = 1$, so $\mathbf{x}_i$ is a scalar, and we write it as x_i. Then $\mathbf{A}$ is a $q \times 1$ column vector. We write this vector as $\mathbf{a}$.

An approach is to regularize this projection by considering

$$\frac{1}{2n}\sum_{i=1}^{n}(x_i - \mathbf{a}^T\mathbf{z}_i)^2 + \sum_{j=1}^{q} P(a_j; \lambda). \tag{7.12}$$

Let

$$\hat{\mathbf{a}} = \underset{\mathbf{a}\in\mathbb{R}^q}{\arg\min} \frac{1}{2n}\sum_{i=1}^{n}(x_i - \mathbf{a}^T\mathbf{z}_i)^2 + \sum_{j=1}^{q} P(a_j; \lambda). \tag{7.13}$$

The efficient score function is

$$\Psi(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i\theta - \mathbf{z}_i^T\boldsymbol{\xi})(x_i - \mathbf{a}^T\mathbf{z}_i).$$

The regularized version is

$$\tilde{\Psi}(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i\theta - \mathbf{z}_i^T\tilde{\boldsymbol{\xi}})(x_i - \hat{\mathbf{a}}^T\mathbf{z}_i).$$

The efficient score estimator is the solution to the equation

$$\tilde{\Psi}(\theta) = 0.$$

This gives

$$\hat{\theta} = \frac{\sum_{i=1}^{n}(y_i - z_i^T\tilde{\boldsymbol{\xi}})(x_i - \hat{\mathbf{a}}^T\mathbf{z}_i)}{\sum_{i=1}^{n}(x_i - \hat{\mathbf{a}}^T\mathbf{z}_i)\mathbf{x}_i}.$$

This can be rewritten as

$$\hat{\theta} = \tilde{\theta} + \frac{\sum_{i=1}^{n}(y_i - x_i\tilde{\theta} - \mathbf{z}_i^T\tilde{\boldsymbol{\xi}})(x_i - \hat{\mathbf{a}}^T\mathbf{z}_i)}{\sum_{i=1}^{n}(x_i - \hat{\mathbf{a}}^T\mathbf{z}_i)x_i}$$

7.4 Efficient score and Wald tests

For testing $H_0 : \theta = \theta_0$, we consider the regularized efficient score statistic

$$S = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i\theta_0 - \mathbf{z}_i^T\hat{\boldsymbol{\xi}}_0)(x_i - \mathbf{a}^T\mathbf{z}_i).$$

Here $\hat{\boldsymbol{\xi}}_0$ is the estimator obtained under H_0, that is,

$$\hat{\boldsymbol{\xi}}_0 = \arg\min_{\boldsymbol{\xi}} \frac{1}{2n}\|\mathbf{y} - \mathbf{x}\theta_0 - \mathbf{Z}\boldsymbol{\xi}\|^2 + \lambda\|\boldsymbol{\xi}\|_1,$$

and $\hat{\mathbf{a}}$ is given in (7.13).

7.5 Applications

7.5.1 Genetic factors of longevity study

7.5.2 Breast cancer gene expression study

7.6 Theoretical properties

7.6.1 Semi-penalized estimator

Define

$$(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\xi}}) = \underset{\boldsymbol{\theta}\in\mathbb{R}^d, \boldsymbol{\xi}\in\mathbb{R}^q}{\arg\min} \{\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta} - \mathbf{Z}\boldsymbol{\xi}\|^2, \boldsymbol{\xi}_{S^c} = \mathbf{0}\}$$

Then

$$\tilde{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{Q}_S\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Q}_S\mathbf{y} = \boldsymbol{\theta} + (\mathbf{X}^T\mathbf{Q}_S\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Q}_S\boldsymbol{\varepsilon}.$$

Thus $\tilde{\boldsymbol{\theta}}$ has a multivariate normal distribution with

$$\mathrm{E}\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}, \mathbb{V}(\tilde{\boldsymbol{\theta}}) = \sigma^2(\mathbf{X}^T\mathbf{Q}_S\mathbf{X})^{-1}.$$

We first state a result when the penalized criterion (7.8) is convex. This necessarily requires $p < n$, but allows $p \to \infty$ as $n \to \infty$. Let $c_{\min} = \min\{c_j : 1 \le j \le p\}$, where c_j is the smallest eigenvalue of $\mathbf{Z}^T\mathbf{Q}_\mathbf{X}\mathbf{Z}/n$. Let $w^o = \max\{w_k^o : k \in S\}$, where $(w_k^o, k \in S)$ are the diagonal elements of $(\mathbf{Z}^T\mathbf{Q}_\mathbf{X}\mathbf{Z}/n)^{-1}$. Denote the smallest nonzero coefficient by $\beta_* = \min\{|\beta_j^o| : \beta_j^o \neq 0, 1 \le j \le p\}$. Denote the cardinality of S by $|S|$.

Theorem 7.1. *Suppose that $\varepsilon_1, \ldots, \varepsilon_n$ are independent and identically distributed as $N(0, \sigma^2)$. Also, suppose that (a) $\gamma > 1/c_{\min}$; (b) for a small $\epsilon > 0$, $\beta_* > \gamma\lambda + \sigma\sqrt{(2/n)w^o \log(|S|/\epsilon)}$; and (c) $\lambda \ge \sigma\sqrt{4\log p}\max_{j\le p}\|\mathbf{x}_j\|/n$. Then,*

$$\mathbf{P}\{\cup_{j=1}^p(\hat{S}_j \neq S_j)\} \le 3\epsilon \text{ and } \mathbf{P}\{\cup_{j=1}^p(\widehat{\beta}_j(\lambda) \neq \widetilde{\beta}_j)\} \le 3\epsilon.$$

This theorem shows that in the convex case, the SPIDR estimator is asymptotically ideal, meaning that it equals the ideal estimator with high probability. As a consequence, it is asymptotically normal. The conditions are mild. The normality assumption on the errors is mainly used for bounding the tail probabilities of the error distribution. This assumption can be relaxed. Condition (a) guarantees that the SPIDR criterions in are strictly convex to ensure unique solution. Condition (b) requires that the nonzero coefficients not be too small so that it is possible to separate them from zero in the presence of random noise. Condition (c) requires the penalty to be proportionally greater than the noise level to prevent false selection of null variables. For standardized predictors with $\|\mathbf{x}_j\|^2 = n$, this condition simplifies to $\lambda \ge \sigma\sqrt{(4/n)\log p}$. Conditions (b) and (c) are related, a bigger λ requires a bigger β^*.

We now consider the high-dimensional cases where $p \gg n$ and the criterions are nonconvex. We require the sparse Riesz condition (SRC, Zhang and Huang (2008)) on the the matrices $Q_j X$. Specifically, we assume there exist constants $0 < c_* \le c^* < \infty$ and integer $d^* \ge |S|(K_* + 1)$ with $K_* = c^*/c_* - 1/2$ such that

$$0 < c_* \le \|Q_j X_{A_j}\mathbf{u}\|^2/n \le c^* < \infty, \|\mathbf{u}\|_2 = 1, \tag{7.14}$$

for every $A_j \subset \{1, \ldots, p\} \setminus \{j\}$ with $|A_j \cup S_j| \le d^*$, for all $1 \le j \le p$.

Theorem 7.2. *Suppose that $\varepsilon_1, \ldots, \varepsilon_n$ are independent and identically distributed as $N(0, \sigma^2)$. Also, suppose that (a) the SRC (7.14) holds with $\gamma \geq c_*^{-1}\sqrt{4 + c_*/c^*}$; (b) for a small $\epsilon > 0$, $\beta_* \geq \gamma 2\sqrt{c^*}\lambda + \sigma\sqrt{(2/n)w^o \log(p|S|/\epsilon)}$; (c) $\lambda \geq \sigma\sqrt{(4\log(p/\epsilon)}\max_{j\leq p}\|\mathbf{x}_j\|/n$. Then*

$$\mathbf{P}\{\cup_{j=1}^p(\hat{S}_j(\hat{\lambda}) \neq S_j)\} \leq 3\epsilon, \quad \textit{and} \quad \mathbf{P}\{\cup_{j=1}^p(\widehat{\beta}_j(\hat{\lambda}) \neq \widetilde{\beta}_j)\} \leq 3\epsilon.$$

Therefore, $\mathbf{P}\{\cup_{j=1}^p(\hat{S}_j(\hat{\lambda}) \neq S_j)\} \to 0$ *and* $\mathbf{P}\{\cup_{j=1}^p(\widehat{\beta}_j(\hat{\lambda}) \neq \widetilde{\beta}_j)\} \to 0$ *as* $\epsilon \to 0$.

The SRC (7.14) ensures that the model is identifiable in a lower-dimensional space that contains the underlying model. When $p > n$, the smallest eigenvalue of $X_j'Q_jX_j/n$ is always zero. But the requirement $c_* > 0$ only concerns $d^* \times d^*$ diagonal submatrices of $X_j'Q_jX_j/n$. By examining the conditions (b) and (c), for standardized predictors with $\|\mathbf{x}_j\| = \sqrt{n}$, we can have $\log(p|S|/\epsilon) = o(n)$ or $p = \epsilon\exp(o(n))/|S|$. Thus for sparse models with $|S|$ small relative to n, Theorem 7.2 shows that the asymptotic idealness property of the SPIDR estimators continues to hold in high-dimensional settings under the SRC and other suitable conditions.

Theorems 7.1 and 7.2 are stated for fixed predictors. For random predictors, the conditions involving the predictors such as the SRC (7.14) need to hold with high probability.

7.7 Technical details

Verification of (7.11)**.** The solution to (7.8) satisfies

$$\begin{cases} X'_{\hat{S}_j}(\mathbf{y} - \mathbf{X}_{\hat{S}}\hat{\boldsymbol{\theta}}_{\hat{S}} - \mathbf{Z}_{\hat{S}_j}\hat{\boldsymbol{\xi}}_{\hat{S}}) = n\dot{P}(\hat{\boldsymbol{\xi}}_{\hat{S}}; \lambda), \\ \mathbf{x}_j'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}} - \mathbf{Z}_{\hat{S}}\hat{\boldsymbol{\xi}}_{\hat{S}}) = 0. \end{cases}$$

The first equation gives

$$\hat{\boldsymbol{\theta}}_{\hat{S}} = (\mathbf{X}_{\hat{S}}^T\mathbf{X}_{\hat{S}})^{-1}\mathbf{X}_{\hat{S}}^T(\mathbf{y} - \mathbf{X}_{\hat{S}}\hat{\boldsymbol{\theta}}_{\hat{S}}) + n(\mathbf{X}_{\hat{S}}^T\mathbf{X}_{\hat{S}})^{-1}\dot{P}(\hat{\boldsymbol{\xi}}_{\hat{S}}; \lambda).$$

Thus

$$\mathbf{X}_{\hat{S}}\hat{\boldsymbol{\theta}}_{\hat{S}} = \mathbf{P}_{\hat{S}}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}}) + \mathbf{X}_{\hat{S}}\Sigma_{\hat{S}}^{-1}\dot{P}(\hat{\boldsymbol{\xi}}_{\hat{S}}; \lambda).$$

Substituting this expression into the second equation gives

$$\mathbf{X}^T\{\mathbf{Q}_{\hat{S}}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\theta}}) - \mathbf{X}_{\hat{S}}\Sigma_{\hat{S}}^{-1}\dot{P}(\hat{\boldsymbol{\xi}}_{\hat{S}}; \lambda)\} = 0.$$

It follows that

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T\mathbf{Q}_{\hat{S}}\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{Q}_{\hat{S}}\mathbf{y} - \mathbf{X}_{\hat{S}}\Sigma_{\hat{S}}^{-1}\dot{P}(\hat{\boldsymbol{\xi}}_{\hat{S}};\lambda).$$

This verifies (7.11).

7.8 Bibliographical notes

This section will include the bibliographical notes on the materials presented in this chapter.

Broadly speaking, the ideas in this chapter fall under the category of what is known as *debiasing* as an approach to inference in penalized likelihood problems. The basic idea behind debiasing is that frequentist inference tends to work well if $\widehat{\beta}_j \dot{\sim} \mathrm{N}(\beta_j, \mathrm{SE}^2)$. Penalized regression estimates obviously do not have this property (with the possible exception of MCP/SCAD), so debiasing approaches attempt to construct an estimate $\tilde{\beta}_j$, based on $\widehat{\boldsymbol{\beta}}$ in some way, for which approximate unbiased normality holds.

Semi-penalized inference is one way to accomplish this: simply set $\lambda_j = 0$ for β_j. Many other approaches along these lines have been proposed, instead using analytical means to develop a bias correction term:

- Zhang and Zhang (2014)
- Bühlmann (2013)
- van de Geer et al. (2013)
- Javanmard and Montanari (2014)

It is worth noting that these ideas are not exactly inferential approaches for penalized regression estimates, but rather ways of using penalized regression estimates as starting points for high-dimensional inference

7.9 Exercises

8

Variable selection with FDR control

9

Resampling approaches to inference

This chapter focuses on the related ideas of subsampling, resampling, and sample splitting as ways to carry out inference for high-dimensional models. These methods tend to be somewhat computationally intensive, as they can involve fitting a high-dimensional model hundreds or thousands of times, although with modern computing power, this is typically not prohibitive except in the case of very large data sets.

Example 9.1. To illustrate the various methods in this chapter, we'll apply them to a simulated data set with the same basic construction as that in Chapter 6. The basic dimensions are $n = 100$ and $p = 60$, with $\sigma^2 = 1$ and the coefficients as follows:

- Six variables with $\beta_j \neq 0$ (category "A"):
  - Two variables with $\beta_j = \pm 1$:
  - Four variables with $\beta_j = \pm 0.5$:
- Each of the six variables is correlated ($\rho = 0.5$) with two other variables (i.e., 12 variables fall into this category) for which $\beta_j = 0$ ("B")
- The remaining 42 variables are pure noise, $\beta_j = 0$ and independent of all other variables ("C")

□

9.1 Sample splitting

9.1.1 Single split

We begin with the simplest idea: sample splitting. We have already seen the basic idea of sample splitting when we discussed the "refitted cross-validation" approach to estimating σ^2 (Section 2.6.1). The approach involves two steps:

(1) Take half of the data and fit a penalized regression model; typically this involves cross-validation as well for the purposes of selecting λ.

(2) Use the remaining half to fit an ordinary least squares model using only the variables that were selected in step (1).

Let's split the data from Example 9.1 into two halves, D_1 and D_2, each with $n = 50$ observations. Fitting a lasso model to D_1 and using cross-validation to select λ, we select 29 variables:

- 5 from category A
- 5 from category B
- 19 from category C

Note that the lasso does a reasonably good job at recovering all of the features with $\beta_j \neq 0$, although it does fail to select one of them. Here, the lasso also selects 24 variables with $\beta_j = 0$. As we have seen in several chapters now, the selection of a feature by the lasso (using cross-validation to choose λ) is not very strong evidence that the feature is important.

Thus, in the second stage of the sample-splitting procedure, we fit an ordinary linear regression model to the selected variables: here, $n = 50$ and $p = 29$. When this is carried out, only two coefficients (the two with $\beta_j = 1$) are significant in the $p < 0.05$ sense. If we relax that to $p < 0.1$, an additional variable with $\beta_j = 0.5$ is found to be significant, as is a variable from category "B" (as you might expect, the variable it was correlated with was the one that was not selected by the original lasso). Note that since the inference here is based on a classical linear model, we have access to all of the usual inferential tools, including confidence intervals. However, we only obtain confidence intervals for coefficients selected in step (1).

The main advantage of the sample splitting approach is that it is clearly valid: all inference is derived from classical linear model theory, and by splitting the data into independent portions, we eliminate feature selection bias. A minor obstacle is that one can have increased type I errors if we fail to select all of the important variables at stage (1); we see a hint of this in the above results, obtaining a borderline significant result for a variable correlated with an unselected feature.

The main disadvantages of sample splitting are the lack of power due to splitting the sample size in half, and the fact that results can vary considerably depending on the split we choose. There is little that can be done to resolve the first issue; the second disadvantage, however, can be addressed by using multiple random splits.

9.1.2 Multiple splits

A simple extension to the approach introduced in Section 9.1.1 is to to apply the sample splitting procedure many times and "average", in some sense, over the splits. This will also help with the problem of failing to select important variables in stage (1); although this may happen in some splits, it is unlikely to happen consistently across a majority of the splits. The major challenge with this approach, however, is how exactly we average over results in which a covariate was not included in the model.

One conservative remedy is to simply assign $p_j = 1$ whenever $j \notin \mathcal{S}$, the set of selected variables from stage 1. With this substitution in place, we will have, for each variable, a vector of p-values $p_j^{(1)}, \ldots, p_j^{(B)}$, where B is the number of random splits, which we could aggregate in a variety of ways. In the results that follow, we use the median, although more complex summaries are also possible (Dezeure et al., 2015).

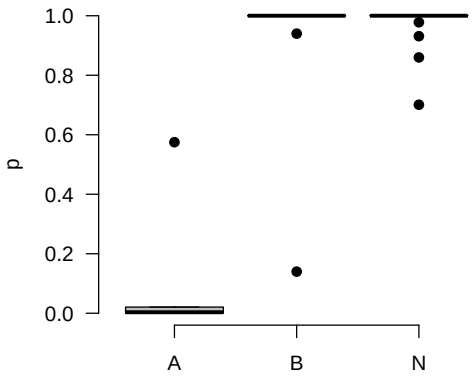

FIGURE 9.1
p-values from the multi-split procedure. Four variables from A have $p < 0.05$.

The resulting p-values from this procedure are plotted in Figure 9.1. Certainly, the results are much more stable if we average across sample splits, although we still suffer from a lack of power to detect all the variable in set A. It is possible to extend this idea to obtain confidence intervals as well by inverting the hypothesis tests; see Dezeure et al. (2015) for details.

Example 9.2. To get a feel for how conservative this approach is, let's apply it to the TCGA data ($n = 536$, $p = 17,322$). Using the multiple-splitting approach, only a single variable is significant with $p < 0.05$ (one

other variable has $p = 0.08$; all others are above 0.1). This is in sharp contrast to the results from Chapter 6, in which the false inclusion rate approach was able to identify 52 features at an FIR of 5%. □

9.2 Stability selection

One could argue that classical p-values are not the most relevant quantity to consider for high-dimensional modeling. Perhaps we should be focused instead on the consistency with which a penalized regression method selects a certain variable. This is main idea behind *stability selection*: we decide that a feature is significant if, in considering the sampling distribution of $\widehat{\boldsymbol{\beta}}$, the feature is selected a high proportion of the time.

Since in practice we have just a single data set, we will have to "perturb" the data in some manner in order to obtain a reasonable approximation to the sampling distribution. This can be done in a variety of ways, but the most familiar method is via resampling (i.e., bootstrapping). Furthermore, there are a variety of ways of carrying out bootstrapping, as we will see in Section 9.3. For simplicity, in this section we will stick to the basic nonparametric bootstrap.

Letting π_{thr} denote a specified threshold and $\hat{\pi}_j(\lambda)$ the fraction of times variable j is selected for a given value of λ, the set of *stable variables* is defined as

$$\{j : \hat{\pi}_j(\lambda) > \pi_{\text{thr}}\}.$$

Figure 9.2 depicts the application of stability selection to the data from Example 9.1 for $N = 100$ bootstrap replications, with the six variables for which $\beta \neq 0$ shown in red. For the sake of discussion, let's focus on $\lambda = 0.4$. The two variables with $|\beta| = 1$ clearly stand out, and are selected in 100% of the resampled data sets. Two of the variables with $|\beta| = 0.5$ stand out, and are selected over 70% of the time, but the other two are difficult to distinguish from noise, being selected just 7% and 18% of the time, compared with 26% for the variable with $\beta = 0$.

The results of stability selection are clear for any given λ, although deciding which λ to focus on is less straightforward. For example, if we had decided to focus on $\lambda = 0.15$, then all of the features with $beta \neq 0$ have higher selection proportions than the features with $\beta = 0$. Of course, whether $\lambda = 0.15$ would be attractive to us if we didn't already know which features were non-null is a good question. In practice, λ for stability selection is often chosen in a subjective manner through

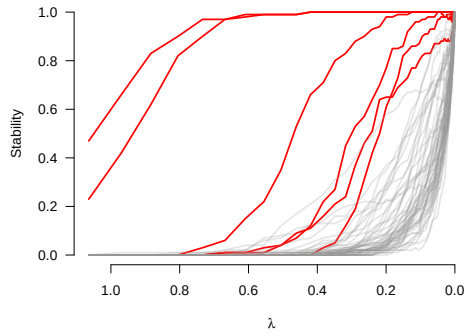

FIGURE 9.2
Stability selection applied to the data from Example 9.1. Variables with $\beta_j \neq 0$ are shown in red.

inspection of plots such as Figure 9.2, although it is possible to choose λ based on FDR considerations (Exercise 9.1).

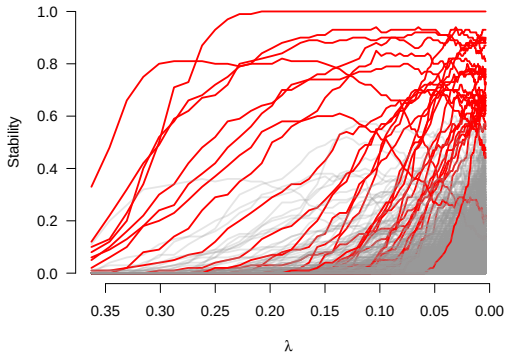

FIGURE 9.3
Stability selection applied to the TCGA data. Features that exceed $\pi_{\text{thr}} = 0.6$ for any λ in red.

In high dimensions, another possible approach is to focus on features that exceed π_{thr} for any value of λ. For example, a stability selection plot for the breast cancer TCGA data is shown in Figure 9.3. Here, features that exceed $\pi_{\text{thr}} = 0.6$ at any point along the regularization path are

shown in red. Some of these variables surpass π_{thr} at low values of λ, then fall below the cutoff, others are almost never selected at high values of λ and only pass π_{thr} towards the end of the regularization path, but all of the variables shown red are likely to be interesting – albeit perhaps for different reasons.

9.3 Bootstrapping

The bootstrap is a widely used statistical method for obtaining confidence intervals for estimates for which analytical, asymptotic formulas are either inaccurate or difficult to derive. Can it be used for the lasso?

First, let's define the bootstrap. There are two common approaches to bootstrapping in the regression setting:

I need to fix or clarify the notation here, since I use $\mathbf{x}_1$ to refer to the first column of $\mathbf{X}$ throughout the rest of the book.

- **Pairwise bootstrap:** Draw a bootstrap sample $\{(\mathbf{x}_1^*, y_1^*), \ldots, (\mathbf{x}_n^*, y_n^*)\}$ with replacement from the $\mathbf{x}, y$ pairs in the original data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$.
- **Residual bootstrap:** Given an estimator $\widehat{\boldsymbol{\beta}}$ and associated residuals $r_1 = y_1 - \mathbf{x}_1^T\widehat{\boldsymbol{\beta}}, \ldots, r_n = y_n - \mathbf{x}_n^T\widehat{\boldsymbol{\beta}}$, draw the bootstrap sample $\{(\mathbf{x}_1, y_1^*), \ldots, (\mathbf{x}_n, y_n^*)\}$, where

$$y_i^* = \mathbf{x}_i^T\widehat{\boldsymbol{\beta}} + \mathbf{r}_i^*,$$

  with r_i* sampled with replacement from $\{r_1, \ldots, r_n\}$. Could also be done in a parametric fashion. Note that $\mathbf{X}$ is essentially treated as fixed here.

In this section, we will focus on the pairwise bootstrap, although many of the same conclusions apply to the residual bootstrap as well.

Here we apply the pairwise bootstrap to the simulated data from Example 9.1; confidence intervals for the coefficients which were nonzero at λ_{CV} are shown in Figure 9.4. We can see immediately that the bootstrap confidence intervals are not achieving the nominal 95% coverage for the "A" variables with $\beta_j = \pm 1$. Due to the shrinkage of the lasso, the bootstrap intervals are also shrunk and systematically fall closer to zero than the true coefficient values. In this example, only two of the confidence intervals for features A1-6 include the true value of β.

However, for the "B" and "N" variables, the opposite phenomenon

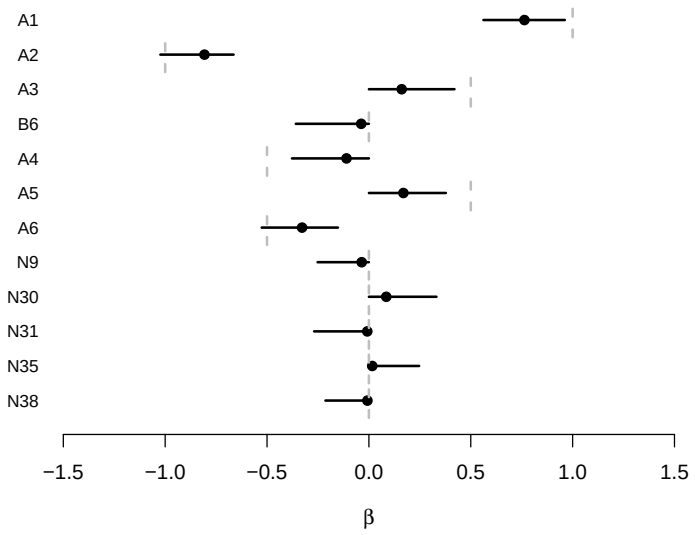

FIGURE 9.4
Pairwise bootstrap for the data from Example 9.1. Dots represent the lasso estimates at λ_{CV}, dotted lines represent the true values of the coefficients, solid lines represent 95% confidence intervals.

happens. All of these features have $\beta_j = 0$; since we are shrinking towards zero, this leads to coverage *higher* than the nominal rate. In this example, the true value of 0 was covered by the 95% confidence interval in all 54 cases involving these coefficients.

This is a consistent pattern. Table 9.1 shows the results of repeating this bootstrap approach to inference for $N = 500$ data sets with 10 "A" variables, 20 "B" variables (each A variable being correlated with two B variables) and 70 "N" variables (pure noise, although correlated with each other: $\text{Cor}(\mathbf{x}_i, \mathbf{x}_j) = 0.8^{|i-j|}$).

Bibliographical notes

Wasserman and Roeder (2009) studied sample splitting for a single split.

TABLE 9.1
Bootstrap simulation

	A	B	N	Overall
Coverage	0.650	0.995	0.999	0.963

Meinshausen et al. (2009) extended this approach by considering multiple random splits and combining the results. Dezeure et al. (2015) provides a comprehensive review of this approach and the details involved, along with procedures for limiting the overall false discovery rate through this form of testing and constructing confidence intervals.

Meinshausen and Buhlmann (2010) proposed stability selection. In that paper, they obtained perturbed data by randomly selecting $n/2$ indices from $\{1, \ldots, n\}$ without replacement. This is based on an argument from Freedman (1977) that sampling $n/2$ without replacement is fairly similar to resampling n with replacement.

Using the bootstrap to obtain confidence intervals for the lasso was first investigated by Knight and Fu (2000), and later by Chatterjee and Lahiri (2010, 2011). However, their work was primarily theoretical and concerned with convergence of the bootstrap distribution to the sampling distribution, rather than with coverage of confidence intervals, which was our focus here.

Exercises

9.1. *FDR bound for stability selection.* Meinshausen & Bühlmann also provide an upper bound for the expected number of false selections in the stable set (i.e., variables with $\beta_j = 0$ and $\hat{\pi}_j(\lambda) > \pi_{\text{thr}}$), which can be used to bound the FDR. In high dimensions, however, this bound tends to be very conservative in practice and not particularly useful: for example, in the TCGA data set, no variables can be stably selected under this rule.

Part III

Other likelihood functions

10

Logistic regression and generalized linear models

10.1 The logistic regression loss function

10.2 Algorithms

10.3 Semi-penalized inference and regularized efficient score estimation

10.4 Selection of λ using ROC curves

10.5 Prediction measures for logistic regression

10.6 Penalized logistic regression using `glmnet` and `ncvreg`

10.6.1 Prediction of origin tissue in metastatic tumor data

10.6.2 Case-control genetic association study of macular degeneration

10.7 Other generalized linear models

10.7.1 Analysis of count data

10.8 *Theoretical properties

In essence, the results derived in Chapter 5 carry over to more general likelihood functions such as the ones we describe in this chapter, although additional regularity conditions are required. We stop short of fully covering the theory for general likelihoods here, but it is worth pointing out the necessary changes in regularity conditions. Generally

speaking, the basic regularity conditions required to ensure asymptotic normality of the MLE (NEEDS TO BE MORE SPECIFIC):

- common support
- identifiability
- the Fisher information $\boldsymbol{\mathcal{F}}(\boldsymbol{\beta})$ is positive definite at $\boldsymbol{\beta}_0$
- all third derivatives of the log-likelihood are bounded

In the case where $p > n$, we require something called restricted strong convexity (RSC), since since $\boldsymbol{\mathcal{F}}(\boldsymbol{\beta})$ cannot be positive definite in that case.

10.9 Exercises

10.1. *Logistic regression: Score and Hessian.* For the logistic regression model

$$\log \frac{\pi_i}{1-\pi_i} = \mathbf{x}_i^T \boldsymbol{\beta},$$

let $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ and L denote the negative log-likelihood.

(a) Show that $-\partial L/\partial \boldsymbol{\eta} = \mathbf{y} - \boldsymbol{\pi}$.

(b) Show that $\partial^2 L/\partial \boldsymbol{\eta}^2 = \text{diag}\{\pi_i(1-\pi_i)\}$.

10.2. *Quadratic approximation to loss functions.* Let $L(\boldsymbol{\beta})$ denote a twice-differentiable loss function. Consider taking a second-order Taylor series expansion of L about $\tilde{\boldsymbol{\eta}}$, where $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ and $\tilde{\boldsymbol{\eta}} = \mathbf{X}\widetilde{\boldsymbol{\beta}}$ (L can be thought of equivalently as a function of $\boldsymbol{\beta}$ or a function of $\boldsymbol{\eta}$). Let $\mathbf{v}$ and $\mathbf{A}$ denote the first and second derivatives of L with respect to $\boldsymbol{\eta}$ (evaluated at $\tilde{\boldsymbol{\eta}}$), and let $\mathbf{z} = \tilde{\boldsymbol{\eta}} - \mathbf{A}^{-1}\mathbf{v}$. Show that, up to a constant,

$$L(\boldsymbol{\beta}) \approx \frac{1}{2}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta})^T \mathbf{A}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}).$$

10.3. *Standardization for multiclass logistic regression.* As mentioned in the chapter, the loss function $L(\boldsymbol{\beta})$ does not change if we shift all K coefficients for feature $\mathbf{x}_j$ by a constant amount c_j. However, this is not true for the penalty term. Show that the value of c that minimizes

$$\lambda \sum_{k=1}^{K} |\beta_{kj} - c_j|$$

is given by the sample median of $\{\beta_{1j}, \ldots, \beta_{Kj}\}$ (more precisely, "any sample median", since the median is not necessarily unique). Do not assume that there are no ties among the coefficients; with sparse regression, there are likely to be multiple zero coefficients.

10.4. *Classification of leukemia subtypes.* There are six known categories of acute lymphoblastic leukemia (ALL). Unfortunately, the accurate assignment of patients to these subtypes is a difficult and expensive process, requiring intensive laboratory studies and the collective expertise of a number of professionals (usually only available at major medical centers).

In this study (`Yeoh2002`), bone marrow samples were obtained from pediatric patients, and gene expression measurements were taken. The goal is to determine ALL subtype from the gene expression data alone – if this can be done accurately, it would make it possible to diagnose ALL subtype at rural hospitals, in developing countries, etc.

(a) Fit a penalized multinomial regression model to the data and select λ in an objective manner. How many nonzero coefficients are in the model? Give both an overall total and the number broken down by ALL subtype category. How many genes are included in the model (keep in mind that a single gene can potentially have multiple nonzero coefficients)?

(b) Summarize the accuracy of your selected model in terms of R^2 and misclassification accuracy. Make sure these quantities are not overestimated due to using the same data for both fitting and for prediction.

(c) The object `Yeoh2002$Xnew` contains an additional 100 samples of gene expression data. For each sample, predict the most likely subtype as well as the probability of that subtype. How many of these predictions do you expect will be correct?

11

Cox regression

Mention AFTs? Stute? Rank? Cai2009?

11.1 Partial likelihoods in the Cox proportional hazards model

11.2 Algorithms

11.3 Theoretical properties

11.4 Semi-penalized inference and regularized efficient score estimation

11.5 Prediction measures for Cox regression

11.6 Fitting penalized Cox regression models in R

11.6.1 Genetic association study of suicidal behaviors

11.6.2 Glioblastoma and exon inclusion and skipping counts

12

Robust regression

Part IV

Structured sparsity

13

Grouped variable selection

13.1 Group lasso, SCAD, and MCP

13.2 Standardization and orthonormalization

13.3 Algorithms

13.4 Theoretical properties

13.5 Fitting group penalized models with `grpreg`

13.5.1 Gene expression in Bardet-Biedl syndrome study

13.5.2 Case-control genetic association study of macular degeneration

13.5.3 Multi-task learning example?

13.6 Overlapping groups

13.6.1 Pathway analysis of gene expression data in olfactory neurons

13.7 Exercises

13.1. *Simulation comparing lasso and group lasso.* Carry out a simulation under three scenarios, each with $n = 100$, $X_{ij} \overset{\perp\!\!\!\perp}{\sim} \mathrm{N}(0,1)$, and $y_i \sim \mathrm{N}(\mathbf{x}_i^T\boldsymbol{\beta}, 1)$, where $\mathbf{X}$ consists of 100 groups and each group has three elements (i.e., the total number of features is 300). The regression coefficients $\boldsymbol{\beta}$ differ across the three scenarios:

(I) Four nonzero groups, each with three nonzero coefficients.

(II) Six nonzero groups, each with two nonzero coefficients.

(III) Twelve nonzero groups, each with one nonzero coefficient.

Note that in each scenario, there are twelve nonzero coefficients. Set each nonzero coefficient to $\pm\sqrt{1/12}$ so that the overall SNR is equal to 1 in all scenarios. Thus, in all three scenarios, $\boldsymbol{\beta}$ is identical in terms of size and sparsity, the only thing that changes is the configuration with respect to the groups. Note: the `genDataGrp()` function in the **hdrm** package offers a convenient way to simulate data in this manner.

For each scenario, fit both a regular lasso and group lasso model. Select λ either using cross-validation or by generating an independent tuning data set (the latter involves slightly more coding, but is much faster to run).

Create a table reporting the MSE for estimating $\boldsymbol{\beta}$ for each method in each scenario, based on repeating this process for 100 independent data sets. Comment briefly on the results and what they illustrate.

14

Bi-level selection

15

Fusion penalties

16

Additive and semiparametric models

16.1 Variable selection in nonparametric additive models

16.2 Structure estimation in partially linear models

16.3 Theoretical properties

16.4 Fitting additive and partially linear models using `grpreg`

16.4.1 Breast cancer gene expression data

16.5 Exercises

16.1. *Group normalization and SPAM.* The sparse additive modeling (SPAM) approach of Ravikumar et al. (2009) proposes the estimation of nonparametric additive regression models by finding the functions $\{f_j\}_{j=1}^p$ that minimize

$$\tfrac{1}{2}\mathbb{E}\left(Y - \sum_{j=1}^{p} f_j(X_j)\right)^2 + \lambda \sum_{j=1}^{p} \|f_j\|_2,$$

where $\|f_j\|_2^2 = \mathbb{E}f_j^2(X_j)$. For any specific random sample, these expected values are replaced by sample averages.

(a) Show that if the functions $\{f_j\}_{j=1}^p$ are modeled via the basis ex-

pansion $f_j(x_{ij}) = \sum_{k=1}^{K} h_{jk}(x_{ij})\beta_{jk}$, this is equivalent to a standard group lasso provided that the basis expansions are standardized such that $\frac{1}{n}\mathbf{H}_j^\top \mathbf{H}_j = \mathbf{I}$ prior to fitting, where the i, jth element of $\mathbf{H}_j$ is $h_{jk}(x_{ij})$.

(b) Suppose the basis matrices $\{\mathbf{H}_j\}_{j=1}^{p}$ are not standardized so that $\frac{1}{n}\mathbf{H}_j^\top \mathbf{H}_j = \mathbf{I}$ prior to fitting the model. What does this mean in terms of fitting sparse additive models?

16.2. *Group lasso analysis of leukemia data.* In this exercise, you will reanalyze the leukemia data set `Golub1999` originally described in Example 6.1 using logistic regression. This time, however, you will construct basis expansions for each feature to allow for nonlinear effects.

Using the function `ns` from the `splines` package, construct a three degree-of-freedom basis expansion for each of the original features. You should end up with a design matrix consisting of 7,129 groups, each with 3 members ($p = 21,387$). Fit a group lasso-penalized logistic regression model for leukemia type (ALL/AML) using this design matrix; select λ using leave-one-out cross-validation.

(a) How many genes are selected? How does this compare to the number of genes selected by the ordinary lasso?

(b) Does allowing for nonlinear effects seem to improve accuracy? In other words, in terms of deviance and misclassification error, does this group lasso approach outperform the ordinary lasso?

17

Multivariate outcomes

18

Variable selection for interactions

18.1 Hierarchical formulation

18.2 Algorithms

18.3 Fitting using `hierNet`

18.4 Fitting using `glinternet`

18.4.1 Gene-gene interactions in the longevity study

18.4.2 Gene-environment interactions in the longevity study

18.5 Exercises

18.1. *Spam detection.* Unsolicited commercial e-mail ("spam") greatly diminishes the value of electronic communication in general, and it is desirable to remove as much of it as possible automatically. The `spam` data set contains information on 3,000 e-mails, including whether or not it was spam as well as 57 numeric features extracted from the e-mail (see `?spam` for details).

(a) Fit a lasso-penalized logistic regression model to the data. Now, noting that many of the features are right-skewed, apply a transformation designed to reduce this skewness and refit the model with these new, transformed predictors. Does the transformation improve the prediction accuracy of the model? If so, continue to use this transformation for the remainder of the problem.

(b) Fit three models to the `spam` data: (1) a lasso with main effects only (2) a lasso with all main effects and pairwise interactions (3) a latent variable group lasso (`glinternet`) model with main effects and pairwise interactions. For each model, report the number of features selected (broken down by main effects / interactions) as well as the prediction accuracy (with respect to misclassification) of the model.

(c) For the best-performing model in part (b), how many spam e-mails in the test set are incorrectly classified as "not spam"? How many non-spam e-mails are incorrectly classified as "spam"? In general, the second kind of mistake (sending important e-mails to a spam folder) is much worse than the first. Suppose we considered the second sort of error as 10 times worse than the first, and reclassified as to minimize the total "weighted" loss. How do the numbers of incorrect classifications in the test set change?

Bibliography

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B*, **57** 289–300.

Berk, R., Brown, L., Buja, A., Zhang, K. and Zhao, L. (2013). Valid post-selection inference. *Annals of Statistics*, **41** 802–837.

Chatterjee, A. and Lahiri, S. (2010). Asymptotic properties of the residual bootstrap for lasso estimators. *Proceedings of the American Mathematical Society*, **138** 4497–4509.

Chatterjee, A. and Lahiri, S. N. (2011). Bootstrapping lasso estimators. *Journal of the American Statistical Association*, **106** 608–625.

Chen, J. and Chen, Z. (2008). Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, **95** 759.

Dezeure, R., Bühlmann, P., Meier, L. and Meinshausen, N. (2015). High-dimensional inference: confidence intervals, p-values and R-software `hdi`. *Statistical Science*, **30** 533–558.

Efron, B. (2010). *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Cambridge University Press.

Efron, B., Tibshirani, R., Storey, J. D. and Tusher, V. (2001). Empirical bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, **96** 1151–1161.

Fan, J., Guo, S. and Hao, N. (2012). Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society Series B*, **74** 37–65.

Freedman, D. (1977). A remark on the difference between sampling with and without replacement. *Journal of the American Statistical Association*, **72** 681–681.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing,

J. R., Caligiuri, M. A. et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286** 531–536.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: applications to nonorthogonal problems. *Technometrics*, **12** 69–82.

Knight, K. and Fu, W. (2000). Asymptotics for lasso-type estimators. *Annals of Statistics*, **28** 1356–1378.

Koussounadis, A., Langdon, S. P., Harrison, D. J. and Smith, V. A. (2014). Chemotherapy-induced dynamic gene expression changes in vivo are prognostic in ovarian cancer. *British Journal of Cancer*, **110** 2975–2984.

Meinshausen, N. and Buhlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society Series B*, **72** 417–473.

Meinshausen, N., Meier, L. and Bühlmann, P. (2009). p-values for high-dimensional regression. *Journal of the American Statistical Association*, **104** 1671–1681.

Ravikumar, P., Lafferty, J., Liu, H. and Wasserman, L. (2009). Sparse additive models. *Journal of the Royal Statistical Society Series B*, **71** 1009–1030.

Reid, S., Tibshirani, R. and Friedman, J. (2016). A study of error variance estimation in lasso regression. *Statistica Sinica* 35–67.

Scheffe, H. (1953). A method for judging all contrasts in the analysis of variance. *Biometrika*, **40** 87–110.

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

Storey, J. D., Taylor, J. E. and Siegmund, D. (2004). Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *Journal of the Royal Statistical Society Series B*, **66** 187–205.

Storey, J. D. and Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, **100** 9440–9445.

Wasserman, L. and Roeder, K. (2009). High dimensional variable selection. *Annals of Statistics*, **37** 2178–2201.

Zou, H., Hastie, T. and Tibshirani, R. (2007). On the "degrees of freedom" of the lasso. *Annals of Statistics*, **35** 2173–2192.