Statistical Foundations of Data Science

Jianqing Fan

Runze Li

Cun-Hui Zhang

Hui Zou


Copyright: Jianqing Fan
Draft—Don't circulate
DEI
SUB NUMINE
VIGET

Preface

Big data are ubiquitous. They come in varying volume, velocity, and variety. They have a deep impact on systems such as storages, communications and computing architectures and analysis such as statistics, computation, optimization, and privacy. Engulfed by a multitude of applications, data science aims to address the large-scale challenges of data analysis, turning big data into smart data for decision making and knowledge discoveries. Data science integrates theories and methods from statistics, optimization, mathematical science, computer science, and information science to extract knowledge, make decisions, discover new insights, and reveal new phenomena from data. The concept of data science has appeared in the literature for several decades and has been interpreted differently by different researchers. It has nowadays become a multi-disciplinary field that distills knowledge in various disciplines to develop new methods, processes, algorithms and systems for knowledge discovery from various kinds of data, which can be either low or high dimensional, and either structured, unstructured or semi-structured. Statistical modeling plays critical roles in the analysis of complex and heterogeneous data and quantifies uncertainties of scientific hypotheses and statistical results.

This book introduces commonly-used statistical models, contemporary statistical machine learning techniques and algorithms, along with their mathematical insights and statistical theories. It aims to serve as a graduate-level textbook on the statistical foundations of data science as well as a research monograph on sparsity, covariance learning, machine learning and statistical inference. For a one-semester graduate level course, it may cover Chapters 2, 3, 9, 10, 12, 13 and some topics selected from the remaining chapters. This gives a comprehensive view on statistical machine learning models, theories and methods. Alternatively, one-semester graduate course may cover Chapters 2, 3, 5, 7, 8 and selected topics from the remaining chapters. This track focuses more on high-dimensional statistics, model selection and inferences but both paths emphasize a great deal on sparsity and variable selections.

Frontiers of scientific research rely on the collection and processing of massive complex data. Information and technology allow us to collect big data of unprecedented size and complexity. Accompanying big data is the rise of dimensionality and high dimensionality characterizes many contemporary statistical problems, from sciences and engineering to social science and humanities. Many traditional statistical procedures for finite or low-dimensional data are still useful in data science, but they become infeasible or ineffective for dealing with high-dimensional data. Hence, new statistical methods are indispensable. The authors have worked on high-dimensional statistics for two decades, and started to write the book on the topics of high-dimensional data analysis over a decade ago. Over the last decide, there have been surges in interest and exciting developments in high-dimensional and big data. This led us to concentrate mainly on statistical aspects of data science.

We aim to introduce commonly-used statistical models, methods and pro-

cedures in data science and provide readers with sufficient and sound theoretical justifications. It has been a challenge for us to balance statistical theories and methods and to choose the topics and works to cover since the amount of publications in this emerging area is enormous. Thus, we focus on the foundational aspects that are related to sparsity, covariance learning, machine learning, and statistical inference.

Sparsity is a common assumption in the analysis of high-dimensional data. By sparsity, we mean that only a handful of features embedded in a huge pool suffice for certain scientific questions or predictions. This book introduces various regularization methods to deal with sparsity, including how to determine penalties and how to choose tuning parameters in regularization methods and numerical optimization algorithms for various statistical models. They can be found in Chapters 3–6 and 8.

High-dimensional measurements are frequently dependent, since these variables often measure similar things, such as aspects of economics or personal health. Many of these variables have heavy tails due to big number of collected variables. To model the dependence, factor models are frequently employed, which exhibit low-rank plus sparse structures in data matrices and can be solved by robust principal component analysis from high-dimensional covariance. Robust covariance learning, principal component analysis, as well as their applications to community detection, topic modeling, recommender systems, ect. are also a feature of this book. They can be found in Chapters 9–11. Note that factor learning or more generally latent structure learning can also be regarded as unsupervised statistical machine learning.

Machine learning is critical in analyzing high-dimensional and complex data. This book also provides readers with a comprehensive account on statistical machine learning methods and algorithms in data science. We introduce statistical procedures for supervised learning in which the response variable (often categorical) is available and the goal is to predict the response based on input variables. This book also provides readers with statistical procedures for unsupervised learning, in which the responsible variable is missing and the goal concentrates on learning the association and patterns among a set of input variables. Feature creations and sparsity learning also arise in these problems. See Chapters 2, 12–14 for details.

Statistical inferences on high-dimensional data are another focus of this book. Statistical inferences require one to characterize the uncertainty, estimate the standard errors of the estimated parameters of primary interest and derive the asymptotic distributions of the resulting estimates. This is very challenging under the high-dimensional regime. See Chapter 7.

Fueled by the surging demands on processing high-dimensional and big data, there have been rapid and vast developments in high-dimensional statistics and machine learning over the last decade, contributed by data scientists from various fields such as statistics, computer science, information theory, applied and computational mathematics, among others. Even though we have narrowed the scope of the book to the statistical aspects of data science, the

field is still too broad for us to cover. Many important contributions that do not fit our presentation have been omitted. Conscientious effort was made in the composition of the reference list and bibliographical notes, but they merely reflect our immediate interests. Omissions and discrepancies are inevitable. We apologize for their occurrence.

Although we all contribute to various chapters and share the responsibility for the whole book, Jianqing Fan was the lead author for Chapters 1, 3 and 9–11, 14 and some sections in other chapters, Runze Li for Chapters 5, and 8 and part of Chapters 6–7, Cun-Hui Zhang for Chapters 4 and 7, and Hui Zou for Chapters 2, 6, 11 and 12 and part of Chapter 5. In addition, Jianqing Fan and Runze Li oversaw the whole book project.

Many people have contributed importantly to the completion of this book. In particular, we would like to thank the editor, John Kimmel, who has been extremely helpful and patient with us for over 10 years! We greatly appreciate a set of around 10 anonymous reviewers for valuable comments that lead to the improvement of the book. We are particularly grateful to Cong Ma and Yiqiao Zhong for preparing a draft of Chapter 14, to Zhao Chen for helping us with putting our unsorted and non-uniform references into the present form, to Tracy Ke, Bryan Kelly, Dacheng Xiu and Jia Wang for helping us with constructing Figure 1.3, and to Boxiang Wang, Yi Yang for helping produce some figures in Chapter 12. Various people have carefully proof-read certain chapters of the book and made useful suggestions. They include Krishna Balasubramanian, Pierre Bayle, Elynn Chen, Wenyan Gong, Yongyi Guo, Cong Ma, Igor Silin, Qiang Sun, Francesca Tang, Bingyan Wang, Kaizheng Wang, Weichen Wang, Yuling Yan, Zhuoran Yang, Mengxin Yu, Wenxin Zhou, Yifeng Zhou, and Ziwei Zhu. We owe them many thanks.

In the spring semester of 2019, we used a draft of this book as a textbook for a first-year graduate course at Princeton University and a senior graduate topic course at the Pennsylvania State University. We would like to thank the graduate students in the classes for their careful readings. In particular, we are indebted to Cong Ma, Kaizheng Wang and Zongjun Tan for assisting in preparing the homework problems at Princeton, most of which are now a part of our exercise at the end of each chapter. At Princeton, we covered chapters 2-3, 5, 8.1, 8.3, 9–14.

We are very grateful for grant supports from National Science Foundation and National Institutes of Health on our research. Finally, we would like to thank our families and our parents for their love and support.

Jianqing Fan
Runze Li
Cun-Hui Zhang
Hui Zou

January 2020.


Copyright: Jianqing Fan
Draft—Don't circulate
DEI
SUB NUMINE
VIGET

Contents


Copyright: Jianqing Fan
Draft—Don't circulate
DEI
SUB NUMINE
VIGET


Copyright: Jianqing Fan
Draft—Don't circulate
DEI
SUB NUMINE
VIGET

Chapter 1

Introduction

The first two decades of this century has witnessed the exposition of the data collection at a blossoming age of information and technology. The recent technological revolution has made information acquisition easy and inexpensive through automated data collection processes. The frontiers of scientific research and technological developments have collected huge amounts of data that are widely available to statisticians and data scientists via internet dissemination. Modern computing power and massive storage allow us to process this data of unprecedented size and complexity. This provides mathematical sciences great opportunities with significant challenges. Innovative reasoning and processing of massive data are now required; novel statistical and computational methods are needed; insightful statistical modeling and theoretical understandings of the methods are essential.

1.1 Rise of Big Data and Dimensionality

Information and technology have revolutionized data collection. Millions of surveillance video cameras, billions of internet searches and social media chats and tweets produce massive data that contain vital information about security, public health, consumer preference, business sentiments, economic health, among others; billions of prescriptions, and enormous amount of genetics and genomics information provide critical data on health and precision medicine; numerous experiments and observations in astrophysics and geosciences give rise to big data in science.

Nowadays, *Big Data* are ubiquitous: from the internet, engineering, science, biology and medicine to government, business, economy, finance, legal, and digital humanities. "There were 5 exabytes of information created between the dawn of civilization through 2003, but that much information is now created every 2 days", according to Eric Schmidt, the CEO of Google, in 2010; "Data are becoming the new raw material of business", according to Craig Mundie, Senior Advisor to the CEO at Microsoft; "Big data is not about the data", according to Gary King of Harvard University. The first quote is on the volume, velocity, variety, and variability of big data nowadays, the second is about the value of big data and its impact to the society, and the third quote is on the importance of the smart analysis of big data.

Accompanying *Big Data* is rising of dimensionality. Frontiers of scientific research depend heavily on the collection and processing of massive complex data. Big data collection and high dimensionality characterize many contemporary statistical problems, from sciences and engineering to social science and humanities. For example, in disease classification using microarray or proteomics data, tens of thousands of expressions of molecules or proteins are potential predictors; in genome-wide association studies, hundreds of thousands of single-nucleotide polymorphisms (SNPs) are potential covariates; in machine learning, millions or even billions of features are extracted from documents, images and other objects; in spatial-temporal problems in economics and earth sciences, time series of hundreds or thousands of regions are collected. When interactions are considered, the dimensionality grows much more quickly. Yet, the interaction terms are needed for understanding the synergy of two genes, proteins or SNPs or the meanings of words. Other examples of massive data include high-resolution images, high-frequency financial data, e-commerce data, warehouse data, functional and longitudinal data, among others. See also Donoho (2000), Fan and Li (2006), Hastie, Tibshirani and Friedman (2009), Bühlmann and van de Geer (2011), Hastie, Tibshirani and Wainwright (2015), and Wainwright (2019) for other examples.

1.1.1 Biological Sciences

Bioimaging technology allows us to simultaneously monitor tens of thousands of genes or proteins as they are expressed differently in the tissues or cells under different experimental conditions. Microarray measures expression profiles of genes, typically in the order of tens of thousands, in a single hybridization experiment, depending on the microarray technology being used. For customized microarrays, the number of genes printed on the chip can be much smaller, giving more accurate measurements on the genes of focused interest. Figure 1.1 shows two microarrays using the Agilent microarray technology and cDNA micorarray technology. The intensity of each spot represents the level of expression of a particular gene. Depending on the nature of the studies, the sample sizes range from a couple to tens or hundreds. For cell lines, the individual variations are relatively small and the sample size can be very small, whereas for tissues from different human subjects, the individual variations are far larger and the sample sizes can be a few hundred.

RNA-seq (Nagalakshmi, et al., 2008), a methodology for RNA profiling based on next-generation sequencing (NGS, Shendure and Ji, 2008), has replaced microarrays for the study of gene expression. Next-generation sequencing is a term used to describe a number of different modern sequencing technologies that allow us to sequence DNA and RNA much more quickly and cheaply. RNA-seq technologies, based on assembling of short reads 30$\sim$400 base pairs, offer advantages such as a wider range of expression levels, less noise, higher throughput, in addition to more information to detect allele-specific expression, novel promoters, and isoforms. There are a number of pa-

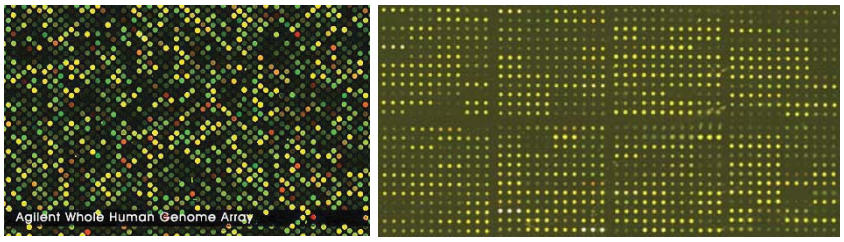

Figure 1.1: Gene expression profiles of microarrays. The intensity at each spot represents the gene expression profile (e.g. Agilent microarray, left panel) or relative profile (e.g. cDNA-microarray, right panel).

pers on statistical methods for detecting differentially expressed genes across treatments/conditions; see Kvam, Liu and Si (2012) for an overview.

After the gene/RNA expression measurements have been properly normalized through RNA-seq or microarray technology, one can then select genes with different expressions under different experimental conditions (e.g. treated with cytokines) or tissues (e.g. normal versus tumor) and genes that express differently over time after treatments (time course experiments). See Speed (2003). This results in a lot of various literature on statistical analysis of controlling the *false discovery rate* in large scale hypothesis testing. See, for example, Benjamini and Hochberg (1995), Storey (2002), Storey and Tibshirani (2003), Efron (2007, 2010b), Fan, Han and Gu (2012), Barber and Candés (2015), Candés, Fan, Janson and Lv (2018), Fan, Ke, Sun and Zhou (2018), among others. The monograph by Efron (2010a) contains a comprehensive account on the subject.

Other aspects of analysis of gene/RNA expression data include association of gene/RNA expression profiles with clinical outcomes such as disease stages or survival time. In this case, the gene expressions are taken as the covariates and the number of variables is usually large even after preprocessing and screening. This results in high-dimensional regression and classification (corresponding to categorical responses, such as tumor types). It is widely believed that only a small group of genes are responsible for a particular clinical outcome. In other words, most of the regression coefficients are zero. This results in high-dimensional sparse regression and classification problems.

There are many other high throughput measurements in biomedical studies. In proteomics, thousands of proteins expression profiles, which are directly related to biological functionality, are simultaneously measured. Similar to genomics studies, the interest is to associate the protein expressions with clinical outcomes and biological functionality. In genomewide association studies, many common genetic variants (typically single-nucleotide polymorphisms or *SNPs*) in different individuals are examined to study if any variant is associated with a trait (heights, weights, eye colors, yields, etc.) or a disease. These

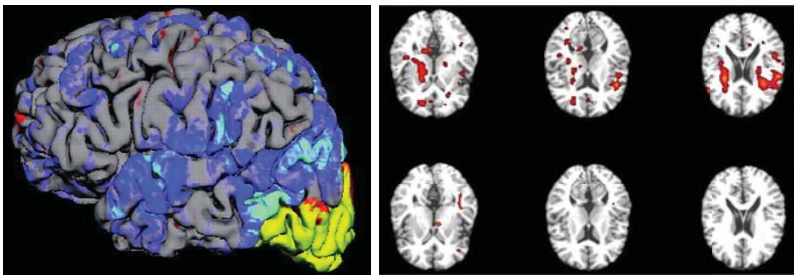

Figure 1.2: Schematic illustration of a brain response to a cognitive task and several slices of its associated fMRI measurements .

genetic variants are referred to as the *quantitative trait loci* (QTL) and hundreds of thousands or millions of SNPs are available for examination. The need for understanding pathophysiology has also led to investigating the so-called *eQTL* studies, the association between SNPs and the expressions of nearby genes. In this case, the gene expressions are regarded as the responses whereas the individual SNPs are taken as the covariates. This again results in high-dimensional regression problems.

High throughput measurements are also commonly used in neuroscience, astronomy, and agriculture and resource surveys using satellite and other imaging technology. In neuroscience, for example, *functional magnetic resonance imaging* (fMRI) technology is frequently applied to measure Blood Oxygenation Level-Dependent (*BOLD*) response to stimuli. This allows investigators to determine which areas of the brain are involved in a cognitive task, or more generally, the functionality of brains. Figure 1.2 gives a schematic illustration. fMRI data contain time-course measurements over tens or hundreds of thousand voxels, resulting in high-dimensional statistical problems.

1.1.2 Health Sciences

Health scientists employ many advanced bioinformatic tools to understand molecular mechanisms of disease initiation and progression, and the impact of genetic variations on clinical outcomes. Many health studies also collect a number of risk factors as well as clinical responses over a period of time: many covariates and responses of each subject are collected at different time points. These kinds of longitudinal studies can give rise to high-dimensional big data.

A famous example is the *Framingham Heart Study*, initiated in 1948 and sponsored by the National Heart, Lung and Blood Institute. Documentation of its first 55 years can be found at the website

http://www.framinghamheartstudy.org/.

More details on this study can be found from the website of the American Heart Association. Briefly, the study follows a representative sample of 5,209

adult residents and their offspring aged 28-62 years in Framingham, Massachusetts. These subjects have been tracked using standardized biennial cardiovascular examination, daily surveillance of hospital admissions, death information and information from physicians and other sources outside the clinic. In 1971, the study enrolled a second-generation group, consisting of 5,124 of the original participants' adult children and their spouses, to participate in similar examinations.

The aim of the Framingham Heart Study is to identify risk factors associated with heart disease, stroke and other diseases, and to understand the circumstances under which cardiovascular diseases arise, evolve and end fatally in the general population. In this study, there are more than 25,000 samples, each consisting of more than 100 variables. Because of the nature of this longitudinal study, some participants cannot be followed up due to their migrations. Thus, the collected data contain many missing values. During the study, cardiovascular diseases may develop for some participants, while other participants may never experience cardiovascular diseases. This implies that some data are censored because the event of particular interest never occurs. Furthermore, data between individuals may not be independent because data for individuals in a family are clustered and likely positively correlated. Missing, censoring and clustering are common features in health studies. These three issues make the data structure complicated and identification of important risk factors more challenging.

High-dimensionality is frequently seen in many other biomedical studies. It also arises in the studies of health costs, health care and health records.

1.1.3 Computer and Information Sciences

The development of information and technology itself collects massive amounts of data. For example, there are billions of web pages on the internet, and an internet search engine needs to statistically learn the most likely outcomes of a query and fast algorithms need to evolve with empirical data. The input dimensionality of queries can be huge. In Google, Facebook and other social networks, algorithms are designed to predict the potential interests of individuals on certain services or products. A familiar example of this kind is amazon.com in which related books are recommended online based on user inputs. This kind of recommendation system applies to other types of services such as music and movies. These are just a few examples of statistical learning in which the data sets are huge and highly complex, and the number of variables is ultrahigh.

Machine learning algorithms have been widely applied to pattern recognition, search engines, computer vision, document and image classification, bioinformatics, medical diagnosis, natural language processing, knowledge graphs, automatic driving machines, internet doctors, among others. The development of these algorithms are based on high-dimensional statistical regres-

Figure 1.3: Some illustrations of machine learning. Top panel: the word clouds of sentiments of a company (Left: Negative Words; Right: Positive Words). The plots were constructed by using data used in Ke, Kelly and Xiu (2019). Bottom left: It is challenging for computer to recognize the pavillion from the background in computer vision. Bottom right: Visualization of the friendship connections in Facebook.

sion and classification with a large number of predictors and a large amount of empirical data. For example, in text and document classification, the data of documents are summarized by word-document information matrices: the frequencies of the words and phrases x in document y are computed. This step of *feature extraction* is very important for the accuracy of classification. A specific example of document classification is E-mail spam in which there are only two classes of E-mails, junk or non-junk. Clearly, the number of features should be very large in order to find important features for accurate document classifications. This results in high-dimensional classification problems.

Similar problems arise for image or object classifications. Feature extractions play critical roles. One approach for such a feature extrapolation is the classical *vector quantization* technique, in which images represented by many small subimages or wavelet coefficients, which are further reduced by summary statistics. Again, this results in high-dimensional predictive variables. Figure 1.3 illustrates a few problems that arise in machine learning.

1.1.4 Economics and Finance

Thanks to the revolution of information and technology, high-frequency financial data have been collected for a host of financial assets, from stocks, bonds, and commodity prices to foreign exchange rates and financial derivatives. The asset correlations among 500 stocks in the S&P500 index already involve over a hundred thousand parameters. This poses challenges on accurately measuring the financial risks of the portfolios, systemic risks in the financial systems, bubble migrations, and risk contagions, in additional to the portfolio allocation and management (Fan, Zhang and Yu, 2012; Brownlees and Engle, 2017). For an overview of high-dimensional economics and finance, see, for example, Fan, Lv and Qi (2012).

To understand the dynamics of financial assets, large panels of financial time series are widely available within asset classes (e.g. components of Russell 3000 stocks) and across asset classes (e.g. stocks, bonds, options, commodities, and other financial derivatives). This is important for understanding the dynamics of price co-movements, time-dependent large volatility matrices of asset returns, systemic risks, and bubble migrations.

Large panel data also arise frequently in economic studies. To analyze the joint evolution of macroeconomic time series, hundreds of macroeconomic variables are compiled to better understand the impact of government policies and to gain better statistical accuracy via, for example, the vector autoregressive model (Sims, 1980). The number of parameters are very large since it grows quadratically with the number of predictors. To enrich the model information, Bernanke et al. (2005) propose to augment standard VAR models with estimated factors (FAVAR) to measure the effects of monetary policy. Factor analysis also plays an important role in prediction using large dimensional data sets (for reviews, see Stock, Watson (2006), Bai and Ng (2008)). A comprehensive collection 131 macroeconomics time series (McCracken and Ng, 2015) with monthly updates can be found in the website

https://research.stlouisfed.org/econ/mccracken/fred-databases/

Spatial-temporal data also give rise to big data in economics. Unemployment rates, housing price indices and sale data are frequently collected in many regions, detailed up to zip code level, over a period of time. The use of spatial correlation enables us to better model the joint dynamics of the data and forecast future outcomes. In addition, exploring homogeneity enables us to aggregate a number of homogeneous regions to reduce the dimensionality, and hence statistical uncertainties, and to better understand heterogeneity across spatial locations. An example of this in prediction of housing appreciation was illustrated in the paper by Fan, Lv, and Qi (2012). See Figure 1.4 and Section 3.9.

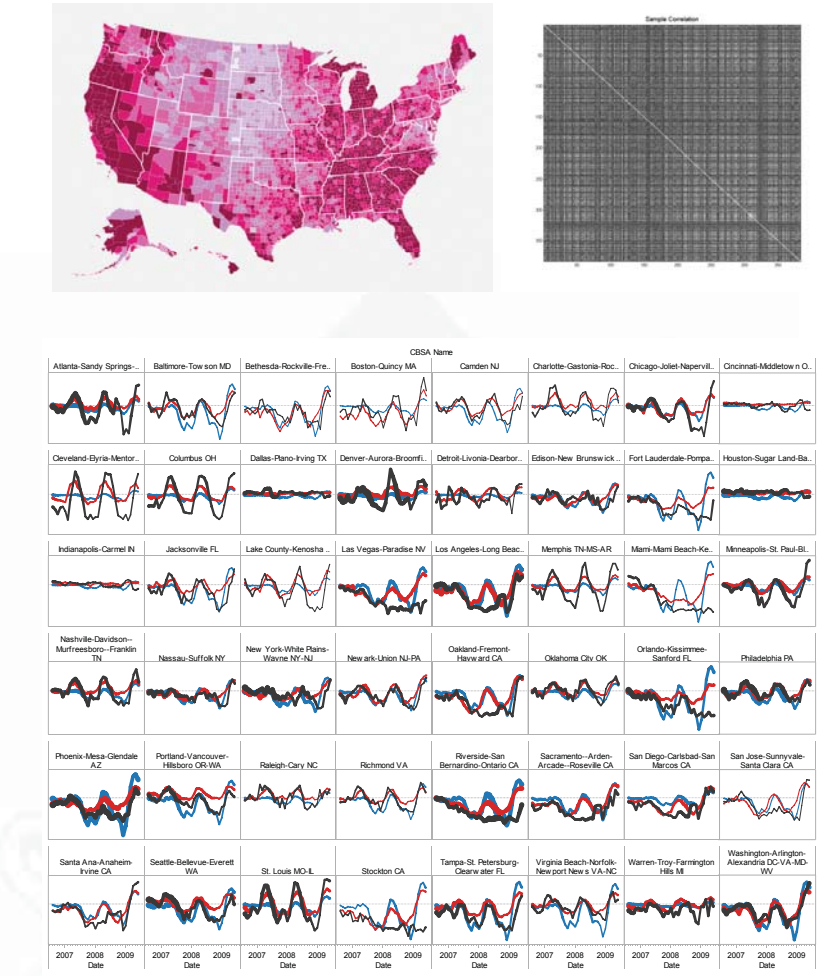

Figure 1.4: Prediction of monthly housing appreciation. Top panel-left: Choropleth map for the 2009 U.S. unemployment rate by county. Top panel-right: Spatial correlation of monthly housing price appreciation among 352 largest counties in the United States from January 2000 to December 2009 (from Fan, Lv, and Qi, 2012). Bottom panel: Prediction of monthly housing pricing appreciation in 48 regions from January 2006 to December 2009 using a large sparse econometrics model with 352 monthly time series from January 2000 to December 2005. Blue: OLS. Red: PLS. Black: Acutal. Thickness: Proportion to repeated sales. Adapted from Fan, Lv, and Qi (2012).

1.1.5 Business and Program Evaluation

Big data arises frequently in marketing and program evaluation. Multi-channel strategies are frequently used to market products, such as drugs and medical devices. Data from hundreds of thousands of doctors are collected with different marketing strategies over a period of time, resulting in big data. The design of marketing strategies and the evaluation of a program's effectiveness are important to corporate revenues and cost savings. This also applies to online advertisements and AB-tests.

Similarly, to evaluate government programs and policies, large numbers of confounders are collected, along with many individual responses to the treatment. This results in big and high-dimensional data.

1.1.6 Earth Sciences and Astronomy

Spatial-temporal data have been widely available in the earth sciences. In meteorology and climatology studies, measurements such as temperatures and precipitations are widely available across many regions over a long period of time. They are critical for understanding climate changes, local and global warming, and weather forecasts, and provide an important basis for energy storage and pricing weather based financial derivatives.

In astronomy, sky surveys collect a huge amount of high-resolution imaging data. They are fundamental to new astronomical discoveries and to understand the origin and dynamics of the universe.

1.2 Impact of Big Data

The arrival of *Big Data* has had deep impact on data system and analysis. It poses great challenges in terms of storage, communication and analysis. It has forever changed many aspects of computer science, statistics, and computational and applied mathematics: from hardware to software; from storage to super-computing; from data base to data security; from data communication to parallel computing; from data analysis to statistical inference and modeling; from scientific computing to optimization. The efforts to provide solutions to these challenges gave birth to a new disciplinary science, data science. Engulfed by the applications in various disciplines, *data science* consists of studies on data acquisition, storage and communication, data analysis and modeling, and scalable algorithms for data analysis and artificial intelligence. For an overview, see Fan, Han, and Liu (2014).

Big Data powers the success of statistical prediction and artificial intelligence. Deep *artificial neural network* models have been very successfully applied to many machine learning and prediction problems, resulting in a discipline called *deep learning* (LeCun, Bengio and Hinton, 2015; Goodfellow, Bengio and Courville, 2016). Deep learning uses a family of over parameterized models, defined through deep neural networks, that have small modeling biases. Such an over-parameterized family of models typically have large vari-

ances, too big to be useful. It is the big amount of data that reduces the variance to an acceptable level, achieving bias and variance trade-off in prediction. Similarly, such an over-parameterized family of models typically are too hard to find reasonable local minima and it is modern computing power and cheap GPUs that make the implementation possible. It is fair to say that today's success of deep learning is powered by the arrivals of big data and modern computing power. These successes will be further carried into the future, as we collect even bigger data and become even better computing architect.

As Big Data are typically collected by automated process and by different generations of technologies, the quality of data is low and measurement errors are inevitable. Since data are collected from various sources and populations, the problem of *heterogeneity* of big data arises. In addition, since the number of variables is typically large, many variables have high kurtosis (much higher than the normal distribution). Moreover, *endogeneity* occurs incidentally due to high-dimensionality that have huge impacts on model selection and statistical inference (Fan and Liao, 2014). These intrinsic features of Big Data have significant impacts on the future developments of big data analysis techniques: from heterogeneity and heavy tailedness to endogeneity and measurement errors. See Fan, Han, and Liu (2014).

Big data are often collected at multiple locations and owned by different parties. They are often too big and unsafe to be stored in one single machine. In addition, the processing power required to manipulate big data is not satisfied by standard computers. For these reasons, big data are often distributed in multiple locations. This creates the issues of communications, privacy and owner issues.

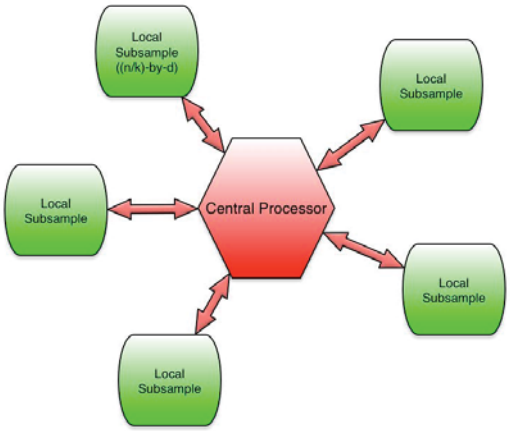

Figure 1.5: Schematic illustration of the distributed data analysis and computing architect.

A simple architect that tackles simultaneously the storage, communication, privacy and ownership issues is the *distributed data analysis* in Figure 1.5. Here, each node analyzes the local data and communicates only the results to the central machine. The central machine then aggregates the results and reports the final results (one-shot analysis) or communicates the results back to each node machine for further analysis (multi-shot analysis). For recent developments on this subject, see Shamir, Srebro and Zhang (2014), Zhang, Duchi and Wainwright (2015), Jordan, Lee and Yang (2018) for low-dimensional regression; Chen and Xie (2014), Lee, Liu, Sun and Taylor (2017), Battey, Fan, Liu, Lu and Zhu (2018) for high-dimensional sparse regression and inference, and El Karoui and d'Aspremont (2010), Liang, et al. (2014), Bertrand and Moonen (2014), Schizas and Aduroja (2015), Garber, Shamir and Srebro (2017), and Fan, Wang, Wang and Zhu (2019) for *principal component analysis*.

As mentioned before, big data are frequently accompanied by high-dimensionality. We now highlight the impacts of dimensionality on data analysis.

1.3 Impact of Dimensionality

What makes high-dimensional statistical inference different from traditional statistics? High-dimensionality has a significant impact on computation, spurious correlation, noise accumulation, and theoretical studies. We now briefly touch these topics.

1.3.1 Computation

Statistical inferences frequently involve numerical optimization. Optimizations in millions and billions dimensional spaces are not unheard of and arise easily when interactions are considered. High-dimensional optimization is not only expensive in computation, but also slow in convergence. It also creates numerical instability. Algorithms can easily get trapped at local minima. In addition, algorithms frequently use iteratively the inversions of large matrices, which causes many instability issues in addition to large computational costs and memory storages. Scalable and stable implementations of high-dimensional statistical procedures are very important to statistical learning.

Intensive computation comes also from the large number of observations, which can be in the order of millions or even billions as in marketing and machine learning studies. In these cases, computation of summary statistics such as correlations among all variables is expensive; yet statistical methods often involve repeated evaluations of summation of loss functions. In addition, when new cases are added, it is ideal to only update some of the summary statistics, rather than to use the entire updated data set to redo the computation. This also saves considerable data storage and computation. Therefore, scalability

of statistical techniques to both dimensionality and the number of cases are paramountly important.

The high dimensionality and the availability of big data have reshaped statistical thinking and data analysis. Dimensionality reduction and feature extraction play pivotal roles in all high-dimensional statistical problems. This helps reduce computation costs as well as improve statistical accuracy and scientific interpretability. The intensive computation inherent in these problems has altered the course of methodological developments. Simplified methods are developed to address the large-scale computational problems. Data scientists are willing to trade statistical efficiencies with computational expediency and robust implementations. Fast and stable implementations of optimization techniques are frequently used.

1.3.2 Noise Accumulation

High-dimensionality has significant impact on statistical inference in at least two important aspects: *noise accumulation* and *spurious correlation*. Noise accumulation refers to the fact that when a statistical rule depends on many parameters, each estimated with stochastic errors, the estimation errors in the rule can accumulate. For high-dimensional statistics, noise accumulation is more severe, and can even dominate the underlying signals. Consider, for example, a linear classification rule which classifies a new data point $\mathbf{x}$ to class 1 if $\mathbf{x}^T\boldsymbol{\beta} > 0$. This rule can have high discrimination power when $\boldsymbol{\beta}$ is known. However, when an estimator $\widehat{\boldsymbol{\beta}}$ is used instead, due to accumulation of errors in estimating the high-dimensional vector $\widehat{\boldsymbol{\beta}}$, the classification rule can be as bad as random guess.

To illustrate the above point, let us assume that we have random samples $\{\mathbf{X}_i\}_{i=1}^n$ and $\{\mathbf{Y}_i\}_{i=1}^n$ from class 0 and class 1 with the population distributions $N(\boldsymbol{\mu}_0, \mathbf{I}_p)$ and $N(\boldsymbol{\mu}_1, \mathbf{I}_p)$, respectively. To mimic the gene expression data, we take $p = 4500$, $\boldsymbol{\mu}_0 = 0$ without loss of generality, and $\boldsymbol{\mu}_1$ from a realization of $0.98\delta_0 + 0.02 * \text{DE}$, a mixture of point mass 0 with probability 0.98 and the standard double exponential distribution with probability 0.02. The realized $\boldsymbol{\mu}_1$ is shown in Figure 1.6, which should have about 90 non-vanishing components and is taken as true $\boldsymbol{\mu}_1$. The components that are considerably different from zero are numbered far less than 90, around 20 to 30 or so.

Unlike high-dimensional regression problems, high-dimensional classification does not have implementation issues if the Euclidian distance based classifier is used; see Figure 1.6. It classifies $\mathbf{x}$ to class 1 if

$$\|\mathbf{x} - \boldsymbol{\mu}_1\|^2 \leq \|\mathbf{x} - \boldsymbol{\mu}_0\|^2 \qquad \text{or} \qquad \boldsymbol{\beta}^T(\mathbf{x} - \boldsymbol{\mu}) \geq 0, \tag{1.1}$$

where $\boldsymbol{\beta} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ and $\boldsymbol{\mu} = (\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1)/2$. For the particular setting in the last paragraph, the distance-based classifier is the Fisher classifier and is the optimal Bayes classifier if prior probability of class 0 is 0.5. The misclassification probability for $\mathbf{x}$ from class 1 into class 0 is $\Phi(-\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|/2)$. This reveals the fact that components with large differences contribute more to differentiating

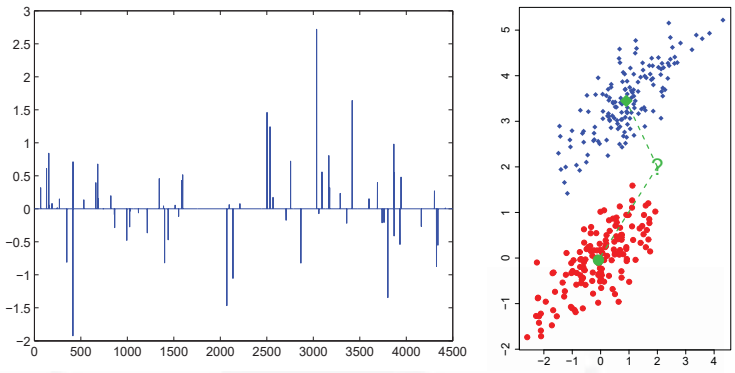

Figure 1.6: Illustration of Classification. Left panel: a realization of $\{\mu_j\}_{j=1}^{4500}$ from the mixture distribution $0.98\delta_0 + 0.02 * \text{DE}$, where DE standards the standard Double Exponential distribution. Right panel: Illustration of the Euclidian distance based classifier, which classifies the query to a class according to its distances to the centroids.

the two classes, and the more components the smaller the discrimination error. In other words, $\Delta_p = \|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|$ is a nondecreasing function of p. Let $\Delta_{(m)}$ be the distance computed based on the m largest components of the difference vector $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. For our particular specification in the last paragraph, the misclassification rate is around $\Phi(-\sqrt{2^2 + 2.5^2}/2) = 0.054$ when the two most powerful components are used ($m = 2$). In addition, $\Delta_{(m)}$ stops increasing noticeably when m reaches 30 and will be constant when $m \geq 100$.

The practical implementation requires estimates of the parameters such as $\widehat{\boldsymbol{\beta}}$. The actual performance of the classifiers can differ from our expectation due to the noise accumulation. To illustrate the noise accumulation phenomenon, let us assume that the rank of the importance of the p features is known to us. In this case, if we use only two features, the classification power is very high. This is shown in Figure 1.7(a). Since the dimensionality is low, the noise in estimated parameters is negligible. Now, if we take $m = 100$, the signal strength Δ_m increases. On the other hand, we need to estimate 100 coefficients $\boldsymbol{\beta}$, which accumulate stochastic noises in the classifier. To visualize this, we project the observed data onto the first two principal components of these 100-dimensional selected features. From Figure 1.7(b), it is clear that signal and noise effect cancel. We still have classification power to differentiate the two classes. When $m = 500$ and 4500, there is no further increase of signals and noise accumulation effect dominates. The performance is as bad as random guessing. Indeed, Fan and Fan (2008) show that almost all high-dimensional classifiers can perform as bad as random guessing unless the signal is excessively strong. See Figure 1.7(c) and (d).

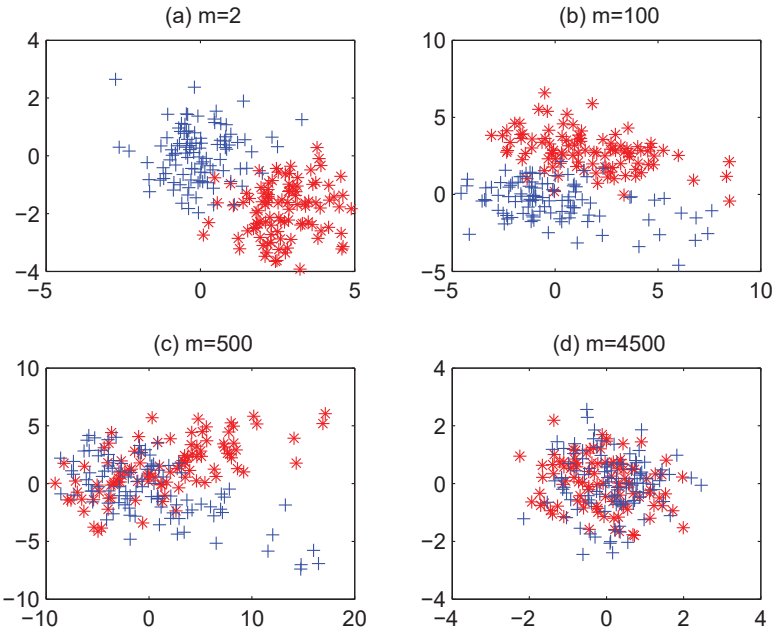

Figure 1.7: Illustration of noise accumulation. Left panel: Projection of observed data ($n = 100$ from each class) onto the first two principal components of m-dimensional selected feature space. The m most important features are extracted before applying the principal component analysis.

Fan and Fan (2008) quantify explicitly the price paid with use of more features. They demonstrate that the classification error rate depends on $\Delta_m/\sqrt{m}$. The numerator shows the benefit of the dimensionality through the increase of signals Δ_m, whereas the denominator represents the noise accumulation effect due to estimation of the unknown parameters. In particular, when $\Delta_p/\sqrt{p} \to \infty$ as $p \to \infty$, Hall, Pittelkow and Ghosh (2008) show that the problem is perfectly classifiable (error rate converges to zero).

The above illustration of the noise accumulation phenomenon reveals the pivotal role of feature selection in high dimensional statistical endeavors. Not only does it reduce the prediction error, but also improves the interpretability of the classification rule. In other words, the use of sparse $\boldsymbol{\beta}$ is preferable.

1.3.3 Spurious Correlation

Spurious correlation refers to the observation that two variables which have no population correlation have a high sample correlation. The analogy is that two persons look alike but have no genetic relation. In a small village,

spurious correlation rarely occurs. This explains why spurious correlation is not an issue in the traditional low-dimensional statistics. In a moderate sized city, however, spurious correlations start to occur. One can find two similar looking persons with no genetic relation. In a large city, one can easily find two persons with similar appearances who have no genetic relation. In the same vein, high dimensionality easily creates issues of spurious correlation.

To illustrate the above concept, let us generate a random sample of size $n = 50$ of $p+1$ independent standard normal random variables $Z_1, \cdots, Z_{p+1} \sim_{i.i.d.} N(0,1)$. Theoretically, the sample correlation between any of two random variables is small. When p is small, say $p = 10$, this is indeed the case and the issue of spurious correlation is not severe. However, when p is large, the spurious correlation starts to be noticeable. To illustrate this, let us compute

$$\widehat{r} = \max_{j \geq 2} \widehat{\text{cor}}(Z_1, Z_j) \tag{1.2}$$

where $\widehat{\text{cor}}(Z_1, Z_j)$ is the sample correlation between the variables Z_1 and Z_j. Similarly, let us compute

$$\widehat{R} = \max_{|\mathcal{S}|=5} \widehat{\text{cor}}(Z_1, \mathbf{Z}_{\mathcal{S}}) \tag{1.3}$$

where $\widehat{\text{cor}}(Z_1, \mathbf{Z}_{\mathcal{S}})$ is the multiple correlation between Z_1 and $\mathbf{Z}_{\mathcal{S}}$, namely, the correlation between Z_1 and its best linear predictor using $\mathbf{Z}_{\mathcal{S}}$. To avoid computing all $\binom{p}{5}$ multiple R^2 in (1.3), we use the forward selection algorithm to compute $\widehat{R}$. The actual value of $\widehat{R}$ is larger than what we present here. We repeat this experiment 200 times and present the distributions of $\widehat{r}$ and $\widehat{R}$ in Figure 1.8.

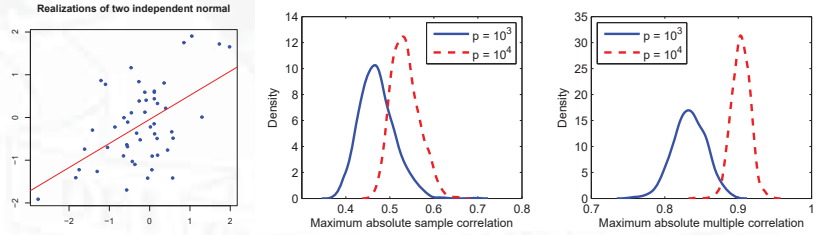

Figure 1.8: Illustration of spurious correlation. Left panel: a typical realization of Z_1 with its mostly spuriously correlated variable ($p = 1000$); middle and left panels: distributions of $\widehat{r}$ and $\widehat{R}$ for $p = 1,000$ and $p = 10,000$, respectively. The sample size is $n = 50$.

The maximum spurious correlation $\widehat{r}$ is around 0.45 for $p = 1000$ and 0.55 for $p = 10,000$. They become 0.85 and 0.91 respectively when multiple correlation $\widehat{R}$ in (1.3) is considered. Theoretical results on the order of these spurious correlations can be found in Cai and Jiang (2012) and Fan, Guo and

Hao (2012), and more comprehensively in Fan, Shao, and Zhou (2018) and Fan and Zhou (2016).

The impact of *spurious correlation* includes false scientific discoveries and false statistical inferences. Since the correlation between Z_1 and $\mathbf{Z}_{\widehat{S}}$ is around 0.9 for a set $\widehat{S}$ with $|\widehat{S}| = 5$ (Figure 1.8), Z_1 and $\mathbf{Z}_{\widehat{S}}$ are practically indistinguishable given $n = 50$. If Z_1 represents the gene expression of a gene that is responsible for a disease, we will also discover 5 genes $\widehat{S}$ that have a similar predictive power although they have no relation to the disease.

To further appreciate the concept of spurious correlation, let us consider the neuroblastoma data used in Oberthuer et al. (2006). The study consists of 251 patients, aged from 0 to 296 months at diagnosis with a median age of 15 months, of the German Neuroblastoma Trials NB90-NB2004, diagnosed between 1989 and 2004. Neuroblastoma is a common paediatric solid cancer, accounting for around 15% of paediatric cancers. 251 neuroblastoma specimens were analyzed using a customized oligonucleotide microarray with $p = 10,707$ gene expressions available after preprocessing. The clinical outcome is taken as the indicator of whether a neuroblastoma child has a 3 year event-free survival. 125 cases are taken at random as the training sample (with 25 positives) and the remaining data are taken as the testing sample. To illustrate the spurious correlation, we now replace the gene expressions by artificially simulated Gaussian data. Using only $p = 1000$ artificial variables along with the traditional forward selection, we can easily find 10 of those artificial variables that perfectly classify the clinical outcomes. Of course, these 10 artificial variables have no relation with the clinical outcomes. When the classification rule is applied to the test samples, the classification result is the same as random guessing.

To see the impact of spurious correlation on statistical inference, let us consider a linear model

$$Y = \mathbf{X}^T\boldsymbol{\beta} + \varepsilon, \qquad \sigma^2 = \mathrm{Var}(\varepsilon). \tag{1.4}$$

Let $\widehat{\mathcal{S}}$ be a selected subset and we compute the residual variances based on the selected variables $\widehat{\mathcal{S}}$:

$$\widehat{\sigma}^2 = \mathbf{Y}^T(I_n - \mathbf{P}_{\widehat{\mathcal{S}}})\mathbf{Y}/(n - |\widehat{\mathcal{S}}|), \qquad \mathbf{P}_{\widehat{\mathcal{S}}} = \mathbf{X}_{\widehat{\mathcal{S}}}(\mathbf{X}_{\widehat{\mathcal{S}}}^T\mathbf{X}_{\widehat{\mathcal{S}}})^{-1}\mathbf{X}_{\widehat{\mathcal{S}}}^T. \tag{1.5}$$

In particular, when $\boldsymbol{\beta} = 0$, all selected variables are spurious. In this case, $\mathbf{Y} = \boldsymbol{\varepsilon}$ and

$$\widehat{\sigma}^2 \approx (1 - \gamma_n^2)\|\boldsymbol{\varepsilon}\|^2/n \approx (1 - \gamma_n^2)\sigma^2, \tag{1.6}$$

when $|\widehat{\mathcal{S}}|/n \to 0$, where $\gamma_n^2 = \boldsymbol{\varepsilon}^T\mathbf{P}_{\widehat{\mathcal{S}}}\boldsymbol{\varepsilon}/\|\boldsymbol{\varepsilon}\|^2$. Therefore, σ^2 is underestimated by a factor of γ_n^2

Suppose that we select only one spurious variable, then that variable must be mostly correlated with $\mathbf{Y}$. Since the spurious correlation is high, the bias is large. The two left panels of Figure 1.9 depicts the distribution of γ_n along with the associated estimates of $\widehat{\sigma}^2$ for different choices of p. Clearly, the bias increases with the dimensionality p.

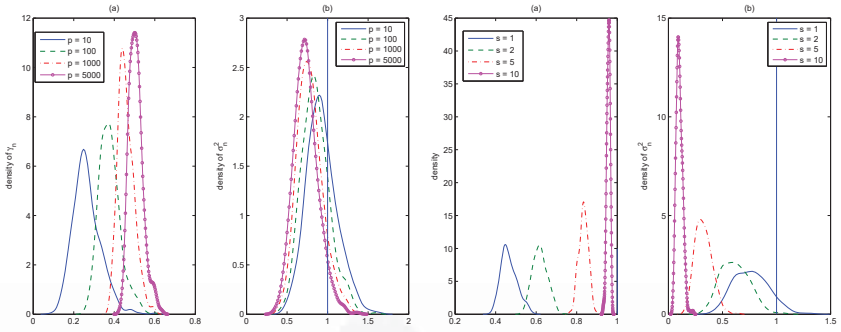

Figure 1.9: Distributions of spurious correlations. Left panel: Distributions of γ_n for the null model when $|\widehat{\mathcal{S}}| = 1$ and their associated estimates of $\sigma^2 = 1$ for various choices of p. Right panel: Distributions of γ_n for the model $Y = 2X_1 + 0.3X_2 + \varepsilon$ and their associated estimates of $\sigma^2 = 1$ for various choices of $|\widehat{\mathcal{S}}|$ but fixed $p = 1000$. The sample size $n = 50$. Adapted from Fan, Guo, and Hao (2012).

Spurious correlation gets larger when more than one spurious variables are selected, as seen in Figure 1.8. To see this, let us consider the linear model $Y = 2X_1 + 0.3X_2 + \varepsilon$ and use forward selection methods to recruit variables. Again, the spurious variables are selected mainly due to their spurious correlation with ε, the unobservable but realized random noises. As shown in the right panel of Figure 1.9, the spurious correlation is very large and $\widehat{\sigma}^2$ gets notably more biased when $|\widehat{\mathcal{S}}|$ gets larger.

Underestimate of residual variance leads to further wrong statistical inferences. More variables will be called statistically significant and that further leads to wrong scientific conclusions. There is active literature on the selective inference for dealing with such kind of issues, starting from Lockhart, Taylor, Tibshirani and Tibshirani (2014); see also Taylor and Tibshirani (2015) and Tibshirani, Taylor, Lockhart and Tibshirani (2016).

1.3.4 Statistical theory

High dimensionality has a strong impact on statistical theory. The traditional asymptotic theory assumes that sample size n tends to infinity while keeping p fixed. This does not reflect the reality of the high dimensionality and cannot explain the observed phenomena such as noise accumulation and spurious correlation. A more reasonable framework is to assume p grows with n and investigate how high the dimensionality p_n a given procedure can handle given the sample size n. This new paradigm is now popularly used in literature.

High dimensionality gives rise to new statistical theory. Many new insights have been unveiled and many new phenomena have been discovered. Subsequent chapters will unveil some of these.

1.4 Aim of High-dimensional Statistical Learning

As shown in Section 1.1, high-dimensional statistical learning arises from various different scientific contexts and has very different disciplinary goals. Nevertheless, its statistical endeavor can be abstracted as follows. The main goals of high dimensional inferences, according to Bickel (2008), are

(a) to construct a method as effective as possible to predict future observations and

(b) to gain insight into the relationship between features and responses for scientific purposes, as well as, hopefully, to construct an improved prediction method.

This view is also shared by Fan and Li (2006). The former appears in problems such as text and document classifications or portfolio optimizations, in which the performance of the procedure is more important than understanding the features that select spam E-mail or stocks that are chosen for portfolio construction. The latter appears naturally in many genomic studies and other scientific endeavors. In these cases, scientists would like to know which genes are responsible for diseases or other biological functions, to understand the molecular mechanisms and biological processes, and predict future outcomes. Clearly, the second goal of high dimensional inferences is more challenging.

The above two objectives are closely related. However, they are not necessarily the same and can be decisively different. A procedure that has a good mean squared error or, more generally risk properties, might not have model selection consistency. For example, if an important variable is missing in a model selection process, the method might find 10 other variables, whose linear combination acts like the missing important variable, to proxy it. As a result, the procedure can still have good prediction power. Yet, the absence of that important variable can lead to false scientific discoveries for objective (b).

As to be seen in Sec 3.3.2, Lasso (Tibshirani, 1996) has very good risk properties under mild conditions. Yet, its model selection consistency requires the restricted *irrepresentable condition* (Zhao and Yu, 2006; Zou, 2006; Meinshausen and Bühlmann, 2006). In other words, one can get optimal rates in mean squared errors, and yet the selected variables can still differ substantially from the underlying true model. In addition, the estimated coefficients are biased. In this view, Lasso aims more at objective (a). In an effort to resolve the problems caused by the L_1-penalty, a class of *folded-concave* penalized least-squares or likelihood procedures, including SCAD, was introduced by Fan and Li (2001), which aims more at objective (b).

1.5 What big data can do

Big Data hold great promise for the discovery of heterogeneity and search for personalized treatments and precision marketing. An important aim for big data analysis is to understand heterogeneity for personalized medicine or services from large pools of variables, factors, genes, environments and their interactions as well as latent factors. Such a kind of understanding is only possible when sample size is very large, particularly for rare diseases.

Another important aim of big data is to discover the commonality and weak patterns, such as the impact of drinking teas and wines on the health, in presence of large variations. Big data allow us to reduce large variances of complexity models such as deep neural network models, as discussed in Section 1.2. The successes of *deep learning* technologies rest to quite an extent on the variance reduction due to big data so that a stable model can be constructed.

1.6 Scope of the book

This book will provide a comprehensive and systematic account of theories and methods in high-dimensional data analysis. The statistical problems range from high-dimensional sparse regression, compressed sensing, sparse likelihood-based models, supervised and unsupervised learning, large covariance matrix estimation and graphical models, high-dimensional survival analysis, robust and quantile regression, among others. The modeling techniques can either be parametric, semi-parametric or nonparametric. In addition, variable selection via regularization methods and sure independent feature screening methods will be introduced.


Copyright: Jianqing Fan
Draft—Don't circulate
DEI
SUB NUMINE
VIGET

Chapter 2

Multiple and Nonparametric Regression

2.1 Introduction

In this chapter we discuss some popular linear methods for regression analysis with continuous response variable. We call them linear regression models in general, but our discussion is not limited to the classical multiple linear regression. They are extended to multivariate nonparametric regression via the kernel trick. We first give a brief introduction to multiple linear regression and least-squares, presenting the basic and important ideas such as inferential results, Box-Cox transformation and basis expansion. We then discuss linear methods based on regularized least-squares with ridge regression as the first example. We then touch on the topic of nonparametric regression in a reproducing kernel Hilbert space (RKHS) via the kernel trick and kernel ridge regression. Some basic elements of the RKHS theory are presented, including the famous representer theorem. Lastly, we discuss the leave-one-out analysis and generalized cross-validation for tuning parameter selection in regularized linear models.

2.2 Multiple Linear Regression

Consider a *multiple linear regression* model:

$$Y = \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon, \tag{2.1}$$

where Y represents the *response* or *dependent variable* and the X variables are often called *explanatory variables* or *covariates* or *independent variables*. The intercept term can be included in the model by including 1 as one of the covariates, say $X_1 = 1$. Note that the term "random error" ε in (2.1) is a generic name used in statistics. In general, the "random error" here corresponds the part of the response variable that cannot be explained or predicted by the covariates. It is often assumed that "random error" ε has zero mean, uncorrelated with covariates X, which is referred to as *exogenous* variables. Our goal is to estimate these β's, called *regression coefficients*, based on a random sample generated from model (2.1).

Suppose that $\{(X_{i1}, \cdots, X_{ip}, Y_i)\}, i = 1, \cdots, n$ is a random sample from

model (2.1). Then, we can write

$$Y_i = \sum_{j=1}^{p} X_{ij}\beta_j + \varepsilon_i. \tag{2.2}$$

The method of least-squares is a standard and popular technique for data fitting. It was advanced early in the nineteenth century by Gauss and Legendre. In (2.2) we have the residuals (r_i's)

$$r_i = Y_i - \sum_{j=1}^{p} X_{ij}\beta_j.$$

Assume that random errors ε_i's are *homoscedastic*, i.e., they are uncorrelated random variables with mean 0 and common variance σ^2. The *least-squares method* is to minimize the residual sum-of-squares (RSS):

$$\mathrm{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} (Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2. \tag{2.3}$$

with respect to $\boldsymbol{\beta}$. Since (2.3) is a nice quadratic function of $\boldsymbol{\beta}$, there is a closed-form solution. Denote by

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix}, \ \mathbf{X}_j = \begin{pmatrix} X_{1j} \\ \vdots \\ X_{nj} \end{pmatrix}, \ \mathbf{X} = \begin{pmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ldots & \vdots \\ X_{n1} & \cdots & X_{np} \end{pmatrix}, \ \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \ \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Then (2.2) can be written in the matrix form

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

The matrix $\mathbf{X}$ is known as the *design matrix* and is of crucial importance to the whole theory of linear regression analysis. The $\mathrm{RSS}(\boldsymbol{\beta})$ can be written as

$$\mathrm{RSS}(\boldsymbol{\beta}) = \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}).$$

Differentiating $\mathrm{RSS}(\boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}$ and setting the gradient vector to zero, we obtain the *normal equations*

$$\mathbf{X}^T\mathbf{Y} = \mathbf{X}^T\mathbf{X}\boldsymbol{\beta}.$$

Here we assume that $p < n$ and $\mathbf{X}$ has rank p. Hence $\mathbf{X}^T\mathbf{X}$ is invertible and the normal equations yield the least-squares estimator of $\boldsymbol{\beta}$

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{2.4}$$

In this chapter $\mathbf{X}^T\mathbf{X}$ is assumed to be invertible unless specifically mentioned otherwise.

The fitted Y value is

$$\widehat{\mathbf{Y}} = \mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y},$$

and the regression residual is

$$\widehat{\mathbf{r}} = \mathbf{Y} - \widehat{\mathbf{Y}} = (\mathbf{I} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{Y}.$$

Theorem 2.1 *Define* $\mathbf{P} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$. *Then we have*

$$\mathbf{P}\mathbf{X}_j = \mathbf{X}_j, \quad j = 1, 2, \cdots, p;$$

$$\mathbf{P}^2 = \mathbf{P} \quad \text{or} \quad \mathbf{P}(\mathbf{I}_n - \mathbf{P}) = \mathbf{0},$$

namely $\mathbf{P}$ *is a projection matrix onto the space spanned by the columns of* $\mathbf{X}$.

Proof. It follows from the direct calculation that

$$\mathbf{P}\mathbf{X} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X} = \mathbf{X}.$$

Taking the j column of the above equality, we obtain the first results. Similarly,

$$\mathbf{P}\mathbf{P} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T = \mathbf{P}.$$

This completes the proof. ∎

By Theorem 2.1 we can write

$$\widehat{\mathbf{Y}} = \mathbf{P}\mathbf{Y}, \quad \widehat{\mathbf{r}} = (\mathbf{I}_n - \mathbf{P})\mathbf{Y} \tag{2.5}$$

and we see two simple identities:

$$\mathbf{P}\widehat{\mathbf{Y}} = \widehat{\mathbf{Y}}, \quad \widehat{\mathbf{Y}}^T\widehat{\mathbf{r}} = 0.$$

This reveals an interesting geometric interpretation of the method of least-squares: the least-squares fit amounts to projecting the response vector onto the linear space spanned by the covariates. See Figure 2.1 for an illustration with two covariates.

2.2.1 The Gauss-Markov Theorem

We assume the linear regression model (2.1) with

- *exogeneity*: $\mathrm{E}(\varepsilon|X) = 0$;
- *homoscedasticity*: $\mathrm{Var}(\varepsilon|X) = \sigma^2$.

Theorem 2.2 *Under model (2.1) with exogenous and homoscedastic error, it follows that*

(i) (unbiasedness) $\mathrm{E}(\widehat{\boldsymbol{\beta}}|\mathbf{X}) = \boldsymbol{\beta}$.

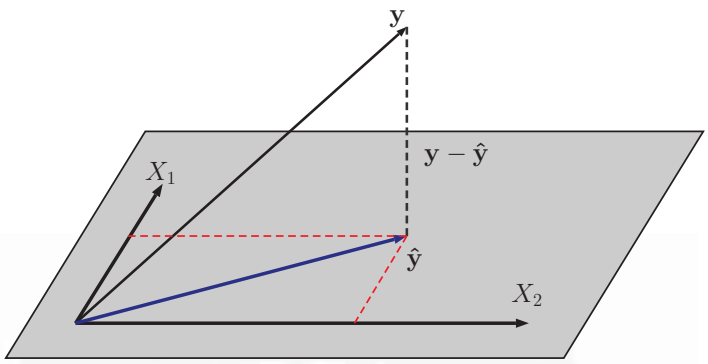

Figure 2.1: Geometric view of least-squares. The fitted value is the blue arrow, which is the projection of $\mathbf{Y}$ on the plane spanned by X_1 and X_2.

(ii) (conditional standard errors) $\text{Var}(\widehat{\boldsymbol{\beta}}|\mathbf{X}) = \sigma^2(\mathbf{X}^T\mathbf{X})^{-1}$.

(iii) (BLUE) *The least-squares estimator $\widehat{\boldsymbol{\beta}}$ is the best linear unbiased estimator (BLUE). That is, for any given vector $\mathbf{a}$, $\mathbf{a}^T\widehat{\boldsymbol{\beta}}$ is a linear unbiased estimator of the parameter $\theta = \mathbf{a}^T\boldsymbol{\beta}$. Further, for any linear unbiased estimator $\mathbf{b}^T\mathbf{Y}$ of θ, its variance is at least as large as that of $\mathbf{a}^T\widehat{\boldsymbol{\beta}}$.*

Proof. The first property follows directly from $\text{E}(\mathbf{Y}|\mathbf{X}) = \mathbf{X}\boldsymbol{\beta}$ and

$$\text{E}(\widehat{\boldsymbol{\beta}}|\mathbf{X}) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}) = \boldsymbol{\beta}.$$

To prove the second property, note that for any linear combination $\mathbf{AY}$, its variance-covariance matrix is given by

$$\text{Var}(\mathbf{AY}|\mathbf{X}) = \mathbf{A}\,\text{Var}(\mathbf{Y}|\mathbf{X})\mathbf{A}^T = \sigma^2\mathbf{A}\mathbf{A}^T. \tag{2.6}$$

Applying this formula to the least-squares estimator with $\mathbf{A} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$, we obtain the property (ii).

To prove property (iii), we first notice that $\mathbf{a}^T\widehat{\boldsymbol{\beta}}$ is an unbiased estimator of the parameter $\theta = \mathbf{a}^T\boldsymbol{\beta}$, with the variance

$$\text{Var}(\mathbf{a}^T\widehat{\boldsymbol{\beta}}|\mathbf{X}) = \mathbf{a}^T\,\text{Var}(\widehat{\boldsymbol{\beta}}|\mathbf{X})\mathbf{a} = \sigma^2\mathbf{a}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{a}.$$

Now, consider any linear unbiased estimator, $\mathbf{b}^T\mathbf{Y}$, of the parameter θ. The unbiasedness requires that

$$\mathbf{b}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{a}^T\boldsymbol{\beta},$$

namely $\mathbf{X}^T\mathbf{b} = \mathbf{a}$. The variance of this linear estimator is

$$\sigma^2\mathbf{b}^T\mathbf{b}.$$

To prove (iii) we need only to show that

$$\mathbf{a}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{a} \leq \mathbf{b}^T\mathbf{b}.$$

Note that

$$(\mathbf{X}^T\mathbf{X})^{-1/2}\mathbf{X}^T\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1/2}\mathbf{a}.$$

Hence, by computing their norms, we have

$$\mathbf{a}^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{a} = \mathbf{b}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{b} = \mathbf{b}^T\mathbf{P}\mathbf{b}.$$

Note that $\mathbf{P} = \mathbf{P}^2$ which means that the eigenvalues of $\mathbf{P}$ are either 1 or 0 and hence $\mathbf{I}_n - \mathbf{P}$ is semi-positive matrix. Hence,

$$\mathbf{b}^T(\mathbf{I}_n - \mathbf{P})\mathbf{b} \geq 0,$$

or equivalently $\mathbf{b}^T\mathbf{b} \geq \mathbf{b}^T\mathbf{P}\mathbf{b}$. ∎

Property (ii) of Theorem 2.2 gives the variance-covariance matrix of the least-squares estimate. In particular, the conditional standard error of $\widehat{\beta}_i$ is simply $\sigma a_{ii}^{1/2}$ and the covariance between $\widehat{\beta}_i$ and $\widehat{\beta}_j$ is $\sigma^2 a_{ij}$, where a_{ij} is the (i,j)-th element of matrix $(\mathbf{X}^T\mathbf{X})^{-1}$.

In many applications σ^2 is often an unknown parameter of the model in addition to the regression coefficient vector $\boldsymbol{\beta}$. In order to use the variance-covariance formula, we first need to find a good estimate of σ^2. Given the least-squares estimate of $\boldsymbol{\beta}$, RSS can be written as

$$\text{RSS} = \sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2 = (\mathbf{Y} - \widehat{\mathbf{Y}})^T(\mathbf{Y} - \widehat{\mathbf{Y}}). \tag{2.7}$$

Define

$$\widehat{\sigma}^2 = \text{RSS}\,/(n-p).$$

This can be shown in Theorem 2.3 that $\widehat{\sigma}^2$ is an unbiased estimator of σ^2.

Theorem 2.3 *Under the linear model (2.1) with homoscedastic error, it follows that*

$$\text{E}(\widehat{\sigma}^2|\mathbf{X}) = \sigma^2.$$

Proof. First by Theorem 2.1 we have

$$\text{RSS} = \|(\mathbf{I}_n - \mathbf{P})\mathbf{Y}\|^2 = \|(\mathbf{I}_n - \mathbf{P})(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\|^2 = \boldsymbol{\varepsilon}^T(\mathbf{I}_n - \mathbf{P})\boldsymbol{\varepsilon}.$$

Let $\text{tr}(\mathbf{A})$ be the trace of the matrix $\mathbf{A}$. Using the property that $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$, we have

$$\text{RSS} = \text{tr}\{(\mathbf{I}_n - \mathbf{P})\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\}.$$

Hence,

$$\mathrm{E}(\mathrm{RSS}\,|\mathbf{X}) = \sigma^2\,\mathrm{tr}(\mathbf{I}_n - \mathbf{P}).$$

Because the eigenvalues of $\mathbf{P}$ are either 1 or 0, its trace is equal to its rank which is p under the assumption that $\mathbf{X}^T\mathbf{X}$ is invertible. Thus,

$$\mathrm{E}(\widehat{\sigma}^2|\mathbf{X}) = \sigma^2(n-p)/(n-p) = \sigma^2.$$

This completes the proof. ∎

2.2.2 Statistical Tests

After fitting the regression model, we often need to perform some tests on the model parameters. For example, we may be interested in testing whether a particular regression coefficient should be zero, or whether several regression coefficients should be zero at the same time, which is equivalent to asking whether these variables are important in presence of other covariates. To facilitate the discussion, we focus on the fixed design case where $\mathbf{X}$ is fixed. This is essentially the same as the random design case but conditioning upon the given realization $\mathbf{X}$.

We assume a homoscedastic model (2.1) with normal error. That is, ε is a Gaussian random variable with zero mean and variance σ^2, written as $\varepsilon \sim N(0, \sigma^2)$. Note that

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta} + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\varepsilon}. \tag{2.8}$$

Then it is easy to see that

$$\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2). \tag{2.9}$$

If we look at each $\widehat{\beta}_j$ marginally, then $\widehat{\beta}_j \sim N(\beta_j, v_j\sigma^2)$ where v_j is the jth diagonal element of $(\mathbf{X}^T\mathbf{X})^{-1}$. In addition,

$$(n-p)\widehat{\sigma}^2 \sim \sigma^2\chi^2_{n-p} \tag{2.10}$$

and $\widehat{\sigma}^2$ is independent of $\widehat{\boldsymbol{\beta}}$. The latter can easily be shown as follow. By (2.7), $\widehat{\sigma}^2$ depends on $\mathbf{Y}$ through $\mathbf{Y} - \widehat{\mathbf{Y}} = (\mathbf{I}_n - \mathbf{P})\boldsymbol{\varepsilon}$ whereas $\widehat{\boldsymbol{\beta}}$ depends on $\mathbf{Y}$ through (2.8) or $\mathbf{X}^T\boldsymbol{\varepsilon}$. Note that both $(\mathbf{I}_n - \mathbf{P})\boldsymbol{\varepsilon}$ and $\mathbf{X}^T\boldsymbol{\varepsilon}$ are jointly normal because they are linear transforms of normally distributed random variables, and therefore their independence is equivalent to their uncorrelatedness. This can easily be checked by computing their covariance

$$\mathrm{E}(\mathbf{I}_n - \mathbf{P})\boldsymbol{\varepsilon}(\mathbf{X}^T\boldsymbol{\varepsilon})^T = \mathrm{E}(\mathbf{I}_n - \mathbf{P})\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T\mathbf{X} = \sigma^2(\mathbf{I}_n - \mathbf{P})\mathbf{X} = 0.$$

If we want to test the hypothesis that $\beta_j = 0$, we can use the following t test statistic

$$t_j = \frac{\widehat{\beta}_j}{\sqrt{v_j}\widehat{\sigma}} \tag{2.11}$$

which follows a t-distribution with $n-p$ degrees of freedom under the null hypothesis $H_0 : \beta_j = 0$. A level α test rejects the null hypothesis if $|t_j| > t_{n-p,1-\alpha/2}$, where $t_{n-p,1-\alpha/2}$ denotes the $100(1-\alpha/2)$ percentile of the t-distribution with $n-p$ degrees of freedom.

In many applications the null hypothesis is that a subset of the covariates have zero regression coefficients. That is, this subset of covariates can be deleted from the regression model: they are unrelated to the response variable given the remaining variables. Under such a null hypothesis, we can reduce the model to a smaller model. Suppose that the reduced model has p_0 many regression coefficients. Let RSS and RSS_0 be the residual sum-of-squares based on the least-squares fit of the full model and the reduced smaller model, respectively. If the null hypothesis is true, then these two quantities should be similar: The RSS reduction by using the full model is small, in relative term. This leads to the *F-statistic*:

$$F = \frac{(\text{RSS}_0 - \text{RSS})/(p-p_0)}{\text{RSS}\,/(n-p)}. \tag{2.12}$$

Under the null hypothesis that the reduced model is correct, $F \sim F_{p-p_0,n-p}$.

The normal error assumption can be relaxed if the sample size n is large. First, we know that $(\mathbf{X}^T\mathbf{X})^{\frac{1}{2}}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})/\sigma$ always has zero mean and an identity variance-covariance matrix. On the other hand, (2.8) gives us

$$(\mathbf{X}^T\mathbf{X})^{\frac{1}{2}}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta})/\sigma = (\mathbf{X}^T\mathbf{X})^{-\frac{1}{2}}\mathbf{X}^T\boldsymbol{\varepsilon}/\sigma.$$

Observe that $(\mathbf{X}^T\mathbf{X})^{-\frac{1}{2}}\mathbf{X}^T\boldsymbol{\varepsilon}/\sigma$ is a linear combination of n i.i.d. random variables $\{\varepsilon_i\}_{i=1}^n$ with zero mean and variance 1. Then the central limit theorem implies that under some regularity conditions,

$$\widehat{\boldsymbol{\beta}} \xrightarrow{D} N(\boldsymbol{\beta}, (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2). \tag{2.13}$$

Consequently, when n is large, the distribution of the t test statistic in (2.11) is approximately $N(0,1)$, and the distribution of the F test statistic in (2.12) is approximately $\chi^2_{p-p_0}/(p-p_0)$.

2.3 Weighted Least-Squares

The method of least-squares can be further generalized to handle the situations where errors are *heteroscedastic* or correlated. In the linear regression model (2.2), we would like to keep the assumption $\text{E}(\varepsilon|\mathbf{X}) = 0$ which means there is no structure information left in the error term. However, the constant variance assumption $\text{Var}(\varepsilon_i|\mathbf{X}_i) = \sigma^2$ may not likely hold in many applications. For example, if y_i is the average response value of the ith subject in a study in which k_i many repeated measurements have been taken, then it would be more reasonable to assume $\text{Var}(\varepsilon_i|\mathbf{X}_i) = \sigma^2/k_i$.

Let us consider a modification of model (2.1) as follows

$$Y_i = \sum_{j=1}^{p} X_{ij}\beta_j + \varepsilon_i; \quad \text{Var}(\varepsilon_i|\mathbf{X}_i) = \sigma^2 v_i \tag{2.14}$$

where v_is are known positive constants but σ^2 remains unknown. One can still use the ordinary least-squares (OLS) estimator $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$. It is easy to show that the OLS estimator is unbiased but no longer BLUE. In fact, the OLS estimator can be improved by using the *weighted least-squares* method.

Let $Y_i^* = v_i^{-1/2}Y_i$, $X_{ij}^* = v_i^{-1/2}X_{ij}$, $\varepsilon_i^* = v_i^{-1/2}\varepsilon_i$. Then the new model (2.14) can be written as

$$Y_i^* = \sum_{j=1}^{p} X_{ij}^*\beta_j + \varepsilon_i^* \tag{2.15}$$

with $\text{Var}(\varepsilon_i^*|\mathbf{X}_i^*) = \sigma^2$. Therefore, the working data $\{(X_{i1}^*, \cdots, X_{ip}^*, Y_i^*)\}_{i=1}^n$ obey the standard *homoscedastic* linear regression model. Applying the standard least-squares method to the working data, we have

$$\widehat{\boldsymbol{\beta}}^{wls} = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(Y_i^* - \sum_{j=1}^{p} X_{ij}^*\beta_j \right)^2 = \text{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} v_i^{-1} \left(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j \right)^2.$$

It follows easily from Theorem 2.2 that the weighted least-squares estimator is the BLUE for $\boldsymbol{\beta}$.

In model (2.14) the errors are assumed to be uncorrelated. In general, the method of least-squares can be extended to handle heteroscedastic and correlated errors.

Assume that

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

and the variance-covariance matrix of $\boldsymbol{\varepsilon}$ is given

$$\text{Var}(\boldsymbol{\varepsilon}|\mathbf{X}) = \sigma^2\mathbf{W}, \tag{2.16}$$

in which $\mathbf{W}$ is a known positive definite matrix. Let $\mathbf{W}^{-1/2}$ be the square root of $\mathbf{W}^{-1}$, i.e.,

$$(\mathbf{W}^{-1/2})^T\mathbf{W}^{-1/2} = \mathbf{W}^{-1}.$$

Then

$$\text{Var}(\mathbf{W}^{-1/2}\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I},$$

which are homoscedastic and uncorrelated.

Define the working data as follows:

$$\mathbf{Y}^* = \mathbf{W}^{-1/2}\mathbf{Y}, \quad \mathbf{X}^* = \mathbf{W}^{-1/2}\mathbf{X}, \quad \boldsymbol{\varepsilon}^* = \mathbf{W}^{-1/2}\boldsymbol{\varepsilon}.$$

Then we have

$$\mathbf{Y}^* = \mathbf{X}^*\boldsymbol{\beta} + \boldsymbol{\varepsilon}^*. \tag{2.17}$$

Thus, we can apply the standard least-squares to the working data. First, the residual sum-of-squares (RSS) is

$$\text{RSS}(\boldsymbol{\beta}) = ||\mathbf{Y}^* - \mathbf{X}^*\boldsymbol{\beta}||^2 = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T\mathbf{W}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}). \tag{2.18}$$

Then the *general least-squares* estimator is defined by

$$\begin{aligned} \widehat{\boldsymbol{\beta}} &= \text{argmin}_{\boldsymbol{\beta}} \, \text{RSS}(\boldsymbol{\beta}) \\ &= (\mathbf{X}^{*T}\mathbf{X}^{*})^{-1}\mathbf{X}^{*T}\mathbf{Y}^{*} \\ &= (\mathbf{X}^{T}\mathbf{W}^{-1}\mathbf{X})^{-1}\mathbf{X}^{T}\mathbf{W}^{-1}\mathbf{Y}. \end{aligned} \tag{2.19}$$

Again, $\widehat{\boldsymbol{\beta}}$ is the BLUE according to Theorem 2.2.

In practice, it is difficult to know precisely the $n \times n$ covariance matrix $\mathbf{W}$; the misspecification of $\mathbf{W}$ in the general least-squares seems hard to avoid. Let us examine the robustness of the general least-squares estimate. Assume that $\text{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{W}_0$, where $\mathbf{W}_0$ is unknown to us, but we employ the general least-squares method (2.19) with the wrong covariance matrix $\mathbf{W}$. We can see that the general least-square estimator is still unbiased:

$$\text{E}(\widehat{\boldsymbol{\beta}}|\mathbf{X}) = (\mathbf{X}^{T}\mathbf{W}^{-1}\mathbf{X})^{-1}\mathbf{X}^{T}\mathbf{W}^{-1}\mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta}.$$

Furthermore, the variance-covariance matrix is given by

$$\text{Var}(\widehat{\boldsymbol{\beta}}) = (\mathbf{X}^{T}\mathbf{W}^{-1}\mathbf{X})^{-1}(\mathbf{X}^{T}\mathbf{W}^{-1}\mathbf{W}_0\mathbf{W}^{-1}\mathbf{X})(\mathbf{X}^{T}\mathbf{W}^{-1}\mathbf{X})^{-1},$$

which is of order $O(n^{-1})$ under some mild conditions. In other words, using wrong covariance matrix would still give us a root-n consistent estimate. So even when errors are heteroscedastic and correlated, the ordinary least-squares estimate with $\mathbf{W} = \mathbf{I}$ and the weighted least-squares estimate with $\mathbf{W} = \text{diag}(\mathbf{W}_0)$ still give us an unbiased and $n^{-1/2}$ consistent estimator. Of course, we still prefer using a working $\mathbf{W}$ matrix that is identical or close to the true $\mathbf{W}_0$.

2.4 Box-Cox Transformation

In practice we often take a transformation of the response variable before fitting a linear regression model. The idea is that the transformed response variable can be modeled by the set of covariates via the classical multiple linear regression model. For example, in many engineering problems we expect $Y \propto X_1^{\beta_1}X_2^{\beta_2} \cdot X_p^{\beta_p}$ where all variables are positive. Then a linear model seems proper by taking logarithms: $\log(Y) = \sum_{j=1}^{p} \beta_j X_j + \varepsilon$. If we assume $\varepsilon \sim N(0, \sigma^2)$, then in the original scale the model is $Y = (\prod_j^p X_j^{\beta_j})\varepsilon^*$ where ε^* is a log-normal random variable: $\log \varepsilon^* \sim N(0, \sigma^2)$.

Box and Cox (1964) advocated the variable transformation idea in linear regression and also proposed a systematic way to estimate the transformation function from data. Their method is now known as *Box-Cox transform* in the literature. Box and Cox (1964) suggested a parametric family for the transformation function. Let $Y^{(\lambda)}$ denote the transformed response where λ parameterizes the transformation function:

$$Y^{(\lambda)} = \begin{cases} \frac{Y^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(Y) & \text{if } \lambda = 0 \end{cases}.$$

Box-Cox model assumes that

$$Y^{(\lambda)} = \sum_{j=1}^{p} X_j \beta_j + \varepsilon$$

where $\varepsilon \sim N(0, \sigma^2)$.

The likelihood function of the Box-Cox model is given by

$$L(\lambda, \boldsymbol{\beta}, \sigma^2) = (\frac{1}{\sqrt{2\pi}\sigma})^n e^{-\frac{1}{2\sigma^2}\|\mathbf{Y}^{(\lambda)} - \mathbf{X}\boldsymbol{\beta}\|^2} \cdot J(\lambda, \mathbf{Y})$$

where $J(\lambda, \mathbf{Y}) = \prod_{i=1}^n |\frac{dy_i^{(\lambda)}}{dy_i}| = (\prod_{i=1}^n |y_i|)^{\lambda - 1}$. Given λ, the maximum likelihood estimators (MLE) of $\boldsymbol{\beta}$ and σ^2 are obtained by the ordinary least-squares:

$$\widehat{\boldsymbol{\beta}}(\lambda) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}^{(\lambda)}, \quad \widehat{\sigma}^2(\lambda) = \frac{1}{n}\|\mathbf{Y}^{(\lambda)} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}^{(\lambda)}\|^2.$$

Plugging $\widehat{\boldsymbol{\beta}}(\lambda), \widehat{\sigma}^2(\lambda)$ into $L(\lambda, \boldsymbol{\beta}, \sigma^2)$ yields a likelihood function of λ

$$\log L(\lambda) = (\lambda - 1)\sum_{i=1}^{n} \log(|y_i|) - \frac{n}{2}\log \widehat{\sigma}^2(\lambda) - \frac{n}{2}.$$

Then the MLE of λ is

$$\widehat{\lambda}_{mle} = \text{argmax}_\lambda \log L(\lambda),$$

and the MLE of $\boldsymbol{\beta}$ and σ^2 are $\widehat{\boldsymbol{\beta}}(\widehat{\lambda}_{mle})$ and $\widehat{\sigma}^2(\widehat{\lambda}_{mle})$, respectively.

2.5 Model Building and Basis Expansions

Multiple linear regression can be used to produce nonlinear regression and other very complicated models. The key idea is to create new covariates from the original ones by adopting some transformations. We then fit a multiple linear regression model using augmented covariates.

For simplicity, we first illustrate some useful transformations in the case of $p = 1$, which is closely related to the curve fitting problem in *nonparametric regression*. In a nonparametric regression model

$$Y = f(X) + \varepsilon,$$

we do not assume a specific form of the regression function $f(x)$, but assume only some qualitative aspects of the regression function. Examples include that $f(\cdot)$ is continuous with a certain number of derivatives or that $f(\cdot)$ is convex. The aim is to estimate the function $f(x)$ and its derivatives, without a specific parametric form of $f(\cdot)$. See, for example Fan and Gijbels (1996), Li and Racine (2007), Hastie, Tibshirani and Friedman (2009), among others.

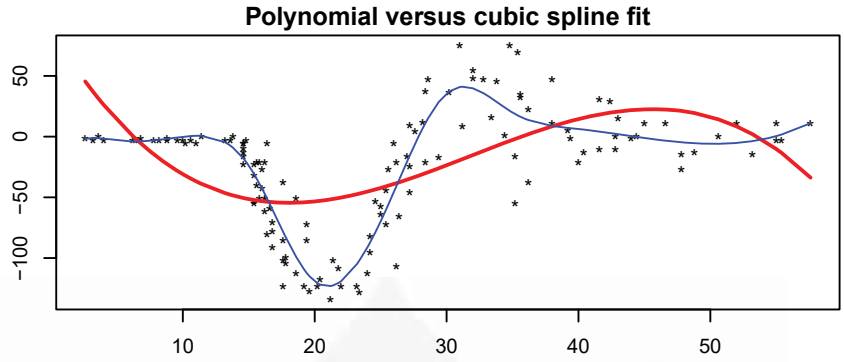

Figure 2.2: Scatter plot of time (in milliseconds) after a simulated impact on motorcycles against the head acceleration (in a) of a test object. Red = cubic polynomial fit, blue = cubic spline fit.

2.5.1 Polynomial Regression

Without loss of generality, assume X is bounded on $[0, 1]$ for simplicity. The Weierstrass approximation theorem states that any continuous $f(x)$ can be uniformly approximated by a polynomial function up to any precision factor. Let us approximate the model by

$$Y = \underbrace{\beta_0 + \beta_1 X + \cdots + \beta_d X^d}_{\approx f(X)} + \varepsilon$$

This *polynomial regression* is a multiple regression problem by setting $X_0 = 1, X_1 = X, \cdots, X_d = X^d$. The design matrix now becomes

$$\mathbf{B}_1 = \begin{pmatrix} 1 & x_1 & \cdots & x_1^d \\ \vdots & \vdots & \ldots & \vdots \\ 1 & x_n & \cdots & x_n^d \end{pmatrix}.$$

We estimate $f(x)$ by

$$\widehat{f}(x) = \widehat{\beta}_0 + \sum_{m=1}^{d} \widehat{\beta}_m x^m,$$

where $\widehat{\boldsymbol{\beta}} = (\mathbf{B}_1^T \mathbf{B}_1)^{-1} \mathbf{B}_1^T \mathbf{Y}$ is the least-squares estimate.

Polynomial functions have derivatives everywhere and are global functions. They are not very flexible in approximating functions with local features such as functions with various degrees of smoothness at different locations. Figure 2.2 shows the cubic polynomial fit to a motorcycle data. Clearly, it does not fit the data very well. Increasing the order of polynomial fits will help

reduce the bias issue, but will not solve the lack of fit issue. This is because that the underlying function cannot be economically approximated by a polynomial function. It requires high-order polynomials to reduce approximation biases, but this increases both variances and instability of the fits. This leads to the introduction of spline functions that allow for more flexibility in function approximation.

2.5.2 Spline Regression

Let $\tau_0 < \tau_1 < \cdots < \tau_{K+1}$. A *spline function* of degree d on $[\tau_0, \tau_{K+1}]$ is a piecewise polynomial function of degree d on intervals $[\tau_j, \tau_{j+1})$ $(j = 0, \cdots, K)$, with continuous first $d-1$ derivatives. The points where the spline function might not have continuous d^{th} derivatives are $\{\tau_j\}_{j=1}^{K}$, which are called *knots*. Thus, a cubic spline function is a piecewise polynomial function with continuous first two derivatives and the points where the third derivative might not exist are called knots of the cubic spline. An example of a cubic fit is given by Figure 2.2.

All spline functions of degree d form a linear space. Let us determine its basis functions.

Linear Splines: A continuous function on $[0, 1]$ can also be approximated by a piecewise constant or linear function. We wish to use a continuous function to approximate $f(x)$. Since a piecewise constant function is not continuous unless the function is a constant in the entire interval, we use a continuous piecewise linear function to fit $f(x)$. Suppose that we split the interval $[0, 1]$ into three regions: $[0, \tau_1], [\tau_1, \tau_2], [\tau_2, 1]$ with given knots τ_1, τ_2. Denote by $l(x)$ the continuous piecewise linear function. In the first interval $[0, \tau_1]$ we write

$$l(x) = \beta_0 + \beta_1 x, \ x \in [0, \tau_1],$$

as it is linear. Since $l(x)$ must be continuous at τ_1, the newly added linear function must have an intercept 0 at point τ_1. Thus, in $[\tau_1, \tau_2]$ we must have

$$l(x) = \beta_0 + \beta_1 x + \beta_2 (x - \tau_1)_+, \ x \in [\tau_1, \tau_2],$$

where z_+ equals z if $z > 0$ and zero otherwise. The function is linear in $[\tau_1, \tau_2]$ with slope $\beta_1 + \beta_2$. Likewise, in $[\tau_2, 1]$ we write

$$l(x) = \beta_0 + \beta_1 x + \beta_2 (x - \tau_1)_+ + \beta_3 (x - \tau_2)_+, \ x \in [\tau_2, 1].$$

The function is now clearly a piecewise linear function with possible different slopes on different intervals. Therefore, the basis functions are

$$B_0(x) = 1, B_1(x) = x, B_2(x) = (x - \tau_1)_+, B_3(x) = (x - \tau_2)_+; \qquad (2.20)$$

which are called a *linear spline* basis. We then approximate the nonparametric regression model as

$$Y = \underbrace{\beta_0 B_0(X) + \beta_1 B_1(X) + \beta_2 B_2(X) + \beta_3 B_3(X)}_{\approx f(X)} + \varepsilon.$$

This is again a multiple regression problem where we set $X_0 = B_0(X), X_1 = B_1(X), X_2 = B_2(X), X_3 = B_3(X)$. The corresponding design matrix becomes

$$\mathbf{B}_2 = \begin{pmatrix} 1 & x_1 & (x_1-\tau_1)_+ & (x_1-\tau_2)_+ \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & (x_n-\tau_1)_+ & (x_n-\tau_2)_+ \end{pmatrix},$$

and we estimate $f(x)$ by

$$\widehat{f}(x) = \widehat{\beta}_0 + \widehat{\beta}_1 x + \widehat{\beta}_2(x-\tau_1)_+ + \widehat{\beta}_3(x-\tau_2)_+,$$

where $\widehat{\boldsymbol{\beta}} = (\mathbf{B}_2^T\mathbf{B}_2)^{-1}\mathbf{B}_2^T\mathbf{Y}$. The above method applies more generally to a multiple knot setting for the data on any intervals.

Cubic Splines: We can further consider fitting piecewise polynomials whose derivatives are also continuous. A popular choice is the so-called cubic spline that is a piecewise cubic polynomial function with continuous first and second derivatives. Again, we consider two knots and three regions: $[0,\tau_1], [\tau_1,\tau_2], [\tau_2,1]$. Let $c(x)$ be a cubic spline. In $[0,\tau_1]$ we write

$$c(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3,\ x \le \tau_1.$$

And $c(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \delta(x)$ in $[\tau_1,\tau_2]$. By definition, $\delta(x)$ is a cubic function in $[\tau_1,\tau_2]$ and its first and second derivatives equal zero at $x = \tau_1$. Then we must have

$$\delta(x) = \beta_4(x-\tau_1)_+^3,\ x \in [\tau_1,\tau_2]$$

which means

$$c(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4(x-\tau_1)_+^3,\ x \in [\tau_1,\tau_2].$$

Likewise, in $[\tau_2,1]$ we must have

$$c(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4(x-\tau_1)_+^3 + \beta_5(x-\tau_2)_+^3,\ x > \tau_2.$$

Therefore, the basis functions are

$$B_0(x) = 1, B_1(x) = x, B_2(x) = x^2, B_3(x) = x^3$$
$$B_4(x) = (x-\tau_1)_+^3, B_5(x) = (x-\tau_2)_+^3.$$

The corresponding transformed design matrix becomes

$$\mathbf{B}_3 = \begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 & (x_1-\tau_1)_+^3 & (x_1-\tau_2)_+^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & (x_n-\tau_1)_+^3 & (x_n-\tau_2)_+^3 \end{pmatrix},$$

and we estimate $f(x)$ by

$$\widehat{f}(x) = \widehat{\beta}_0 + \widehat{\beta}_1 x + \widehat{\beta}_2 x^2 + \widehat{\beta}_3 x^3 + \widehat{\beta}_4 (x - \tau_1)_+^3 + \widehat{\beta}_5 (x - \tau_2)_+^3,$$

where $\widehat{\boldsymbol{\beta}} = (\mathbf{B}_3^T \mathbf{B}_3)^{-1} \mathbf{B}_3^T \mathbf{Y}$ is the least-squares estimate of the coefficients.

In general, if there are K knots $\{\tau_1, \cdots, \tau_K\}$, then the *basis functions* of *cubic splines* are

$$B_0(x) = 1, B_1(x) = x, B_2(x) = x^2, B_3(x) = x^3$$
$$B_4(x) = (x - \tau_1)_+^3, \cdots, B_{K+3}(x) = (x - \tau_K)_+^3.$$

By approximating the nonparametric function $f(X)$ by the spline function with knots $\{\tau_j\}_{j=1}^K$, we have

$$Y = \underbrace{\beta_0 B_0(X) + \beta_1 B_1(X) + \cdots + \beta_{K+3} B_{K+3}(X)}_{\approx f(X)} + \varepsilon \tag{2.21}$$

This *spline regression* is again a multiple regression problem.

Natural Cubic Splines: Extrapolation is always a serious issue in regression. It is not wise to fit a cubic function to a region where the observations are scarce. If we must, extrapolation with a linear function is preferred. A *natural cubic spline* is a special cubic spline with additional constraints: the cubic spline must be linear beyond two end knots. Consider a natural cubic spline, $\mathrm{NC}(x)$, with knots at $\{\tau_1, \cdots, \tau_K\}$. By its cubic spline representation, we can write

$$\mathrm{NC}(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \sum_{j=1}^K \beta_{3+j} (x - \tau_j)_+^3.$$

First, $\mathrm{NC}(x)$ is linear for $x < \tau_1$, which implies that

$$\beta_2 = \beta_3 = 0.$$

Second, $\mathrm{NC}(x)$ is linear for $x > \tau_K$, which means that

$$\sum_{j=1}^K \beta_{3+j} = 0, \qquad \sum_{j=1}^K \tau_j \beta_{3+j} = 0,$$

corresponding to the coefficients for the cubic and quadratic term of the polynomial $\sum_{j=1}^K \beta_{3+j} (x - \tau_j)^3$ for $x > \tau_K$. We solve for β_{K+2}, β_{K+3} from the above equations and then write $\mathrm{NC}(x)$ as

$$\mathrm{NC}(x) \quad = \quad \sum_{j=0}^{K-1} \beta_j B_j(x),$$

where the *natural cubic spline* basis functions are given by

$$B_0(x) = 1, B_1(x) = x,$$
$$B_{j+1}(x) = \frac{(x-\tau_j)_+^3 - (x-\tau_K)_+^3}{\tau_j - \tau_K} - \frac{(x-\tau_{K-1})_+^3 - (x-\tau_K)_+^3}{\tau_{K-1} - \tau_K}$$
for $j = 1, \cdots, K-2$.

Again, by approximating the nonparametric function with the natural cubic spline, we have

$$Y = \sum_{j=0}^{K-1} \beta_j B_j(X) + \varepsilon. \tag{2.22}$$

which can be solved by using multiple regression techniques.

2.5.3 Multiple Covariates

The concept of polynomial regression extends to multivariate covariates. The simplest example is the bivariate regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_1 X_2 + \beta_5 X_2^2 + \varepsilon.$$

The term X_1X_2 is called the *interaction*, which quantifies how X_1 and X_2 work together to contribute to the response. Often, one introduces interactions without using the quadratic term, leading to a slightly simplified model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon.$$

More generally, the multivariate quadratic regression is of the form

$$Y = \sum_{j=1}^{p} \beta_j X_j + \sum_{j \le k} \beta_{jk} X_j X_k + \varepsilon \tag{2.23}$$

and the multivariate regression with main effects (the linear terms) and interactions is of the form

$$Y = \sum_{j=1}^{p} \beta_j X_j + \sum_{j < k} \beta_{jk} X_j X_k + \varepsilon. \tag{2.24}$$

This concept can also be extended to the multivariate spline case. The basis function can be the tensor of univariate spline basis function for not only unstructured $f(\mathbf{x})$, but also other basis functions for structured $f(\mathbf{x})$. Unstructured nonparametric functions are not very useful: If each variable uses 100 basis functions, then there are 100^p basis functions in the tensor products, which is prohibitively large for say, $p = 10$. Such an issue is termed the "curse-of-dimensionality" in literature. See Hastie and Tibshirani (1990) and Fan and Gijbels (1996). On the other hand, for the structured multivariate

model, such as the following additive model (Stone, 1985, 1994; Hastie and Tibshirani, 1990),

$$Y = f_1(X_1) + \cdots + f_p(X_p) + \varepsilon \tag{2.25}$$

the basis functions are simply the collection of all univariate basis functions for approximating $f_1, \cdots, f_p$. The total number grows only linearly with p.

In general, let $B_m(\mathbf{x})$ be the basis functions $m = 1, \cdots, M$. Then, we approximate multivariate nonparametric regression model $Y = f(\mathbf{X}) + \varepsilon$ by

$$Y = \sum_{m=1}^{M} \beta_j B_j(\mathbf{X}) + \varepsilon. \tag{2.26}$$

This can be fit using a multiple regression technique. The new design matrix is

$$\mathbf{B} = \begin{pmatrix} B_1(\mathbf{X}_1) & \cdots & B_M(\mathbf{X}_1) \\ \vdots & \cdots & \vdots \\ B_1(\mathbf{X}_n) & \cdots & B_M(\mathbf{X}_n) \end{pmatrix}$$

and the least-squares estimate is given by

$$\widehat{f}(\mathbf{x}) = \sum_{m=1}^{M} \widehat{\beta}_m B_m(\mathbf{x}),$$

where

$$\widehat{\boldsymbol{\beta}} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{Y}.$$

The above fitting implicitly assumes that $M \ll n$. This condition in fact can easily be violated in unstructured multivariate nonparametric regression. For the additive model (2.25), in which we assume $f(\mathbf{x}) = \sum_{j=1}^{p} f_j(x_j)$ where each $f_j(x_j)$ is a smooth univariate function of x_j, the univariate basis expansion ideas can be readily applied to approximation of each $f_j(x_j)$:

$$f_j(x_j) \approx \sum_{m=1}^{M_j} B_{jm}(x_j)\beta_{jm}$$

which implies that the fitted regression function is

$$f(\mathbf{x}) \approx \sum_{j=1}^{p} \sum_{m=1}^{M_j} B_{jm}(x_j)\beta_{jm}.$$

In Section 2.6.5 and Section 2.7 we introduce a fully nonparametric multiple regression technique which can be regarded as a basis expansion method where the basis functions are given by kernel functions.

2.6 Ridge Regression

2.6.1 Bias-Variance Tradeoff

Recall that the ordinary least squares estimate is defined by $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ when $\mathbf{X}$ is of full rank. In practice, we often encounter highly correlated covariates, which is known as the *collinearity* issue. As a result, although $\mathbf{X}^T\mathbf{X}$ is still invertible, its smallest eigenvalue can be very small. Under the homoscedastic error model, the variance-covariance matrix of the OLS estimate is $\mathrm{Var}(\widehat{\boldsymbol{\beta}}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$. Thus, the collinearity issue makes $\mathrm{Var}(\widehat{\boldsymbol{\beta}})$ large.

Hoerl and Kennard (1970) introduced the *ridge regression* estimator as follows:

$$\widehat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}, \tag{2.27}$$

where $\lambda > 0$ is a regularization parameter. In the usual case ($\mathbf{X}^T\mathbf{X}$ is invertible), ridge regression reduces to OLS by setting $\lambda = 0$. However, ridge regression is always well defined even when $\mathbf{X}$ is not full rank.

Under the assumption $\mathrm{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}$, it is easy to show that

$$\mathrm{Var}(\widehat{\boldsymbol{\beta}}_\lambda) = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\sigma^2. \tag{2.28}$$

We always have $\mathrm{Var}(\widehat{\boldsymbol{\beta}}_\lambda) < (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2$. Ridge regression estimator reduces the estimation variance by paying a price in estimation bias:

$$\mathrm{E}(\widehat{\boldsymbol{\beta}}_\lambda) - \boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\beta} = -\lambda(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\boldsymbol{\beta}. \tag{2.29}$$

The overall estimation accuracy is gauged by the mean squared error (MSE). For $\widehat{\boldsymbol{\beta}}_\lambda$ its MSE is given by

$$\mathrm{MSE}(\widehat{\boldsymbol{\beta}}_\lambda) = \mathrm{E}(\|\widehat{\boldsymbol{\beta}}_\lambda - \boldsymbol{\beta}\|^2). \tag{2.30}$$

By (2.28) and (2.29) we have

$$\begin{aligned} \mathrm{MSE}(\widehat{\boldsymbol{\beta}}_\lambda) &= \mathrm{tr}\left((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\sigma^2\right) \\ &\quad +\lambda^2\boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-2}\boldsymbol{\beta} \\ &= \mathrm{tr}\left((\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-2}[\lambda^2\boldsymbol{\beta}\boldsymbol{\beta}^T + \sigma^2\mathbf{X}^T\mathbf{X}]\right). \end{aligned} \tag{2.31}$$

It can be shown that $\frac{d\mathrm{MSE}(\widehat{\boldsymbol{\beta}}_\lambda)}{d\lambda}|_{\lambda=0} < 0$, which implies that there are some proper λ values by which ridge regression improves OLS.

2.6.2 ℓ_2 Penalized Least Squares

Define a penalized residual sum-of-squares (PRSS) as follows:

$$\mathrm{PRSS}(\boldsymbol{\beta}|\lambda) = \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2. \tag{2.32}$$

Then let

$$\widehat{\boldsymbol{\beta}}_\lambda = \text{argmin}_{\boldsymbol{\beta}} \, \text{PRSS}(\boldsymbol{\beta}|\lambda). \tag{2.33}$$

Note that we can write it in a matrix form

$$\text{PRSS}(\boldsymbol{\beta}|\lambda) = \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|^2.$$

The term $\lambda\|\boldsymbol{\beta}\|^2$ is called the ℓ_2-penalty of $\boldsymbol{\beta}$. Taking derivatives with respect to $\boldsymbol{\beta}$ and setting it to zero, we solve the root of the following equation

$$-\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta} = 0,$$

which yields

$$\widehat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}.$$

The above discussion shows that ridge regression is equivalent to the ℓ_2 penalized least-squares.

We have seen that ridge regression can achieve a smaller MSE than OLS. In other words, the ℓ_2 penalty term helps regularize (reduce) estimation variance and produces a better estimator when the reduction in variance exceeds the induced extra bias. From this perspective, one can also consider a more general ℓ_q penalized least-squares estimate

$$\min_{\boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\sum_{j=1}^{p}|\beta_j|^q \tag{2.34}$$

where q is a positive constant. This is referred to as the Bridge estimator (Frank and Friedman, 1993). The ℓ_q penalty is strictly concave when $0 < q < 1$, and strictly convex when $q > 1$. For $q = 1$, the resulting ℓ_1 penalized least-squares is also known as the Lasso (Tibshirani, 1996). Chapter 3 covers the Lasso in great detail. Among all Bridge estimators only the ridge regression has a nice closed-form solution with a general design matrix.

2.6.3 Bayesian Interpretation

Ridge regression has a neat Bayesian interpretation in the sense that it can be a formal Bayes estimator. We begin with the homoscedastic Gaussian error model:

$$Y_i = \sum_{j=1}^{p} X_{ij}\beta_j + \varepsilon_i$$

and $\varepsilon_i|\mathbf{X}_i \sim N(0, \sigma^2)$. Now suppose that β_j's are also independent $N(0, \tau^2)$ variables, which represent our knowledge about the regression coefficients before seeing the data. In Bayesian statistics, $N(0, \tau^2)$ is called the prior distribution of β_j. The model and the prior together give us the posterior distribution of $\boldsymbol{\beta}$ given the data (the conditional distribution of $\boldsymbol{\beta}$ given $\mathbf{Y}, \mathbf{X}$). Straightforward calculations yield

$$P(\boldsymbol{\beta}|\mathbf{Y}, \mathbf{X}) \propto \exp(-\frac{1}{2\sigma^2}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2)\exp(-\frac{1}{2\tau^2}\|\boldsymbol{\beta}\|^2). \tag{2.35}$$

A maximum posteriori probability (MAP) estimate is defined as

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^{\text{MAP}} &= \text{argmax}_{\boldsymbol{\beta}} P(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X}) \\ &= \text{argmax}_{\boldsymbol{\beta}} \left\{ -\frac{1}{2\sigma^2}\|\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\|^2 - \frac{1}{2\tau^2}\|\boldsymbol{\beta}\|^2 \right\}. \end{aligned} \tag{2.36}$$

It is easy to see that $\widehat{\boldsymbol{\beta}}^{\text{MAP}}$ is ridge regression with $\lambda = \frac{\sigma^2}{\tau^2}$. Another popular Bayesian estimate is the posterior mean. In this model, the posterior mean and posterior mode are the same.

From the Bayesian perspective, it is easy to construct a generalized ridge regression estimator. Suppose that the prior distribution for the entire $\boldsymbol{\beta}$ vector is $N(0,\boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a general positive definite matrix. Then the posterior distribution is computed as

$$P(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X}) \propto \exp(-\frac{1}{2\sigma^2}\|\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\|^2)\exp(-\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}). \tag{2.37}$$

The corresponding MAP estimate is

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^{\text{MAP}} &= \text{argmax}_{\boldsymbol{\beta}} P(\boldsymbol{\beta}|\mathbf{Y},\mathbf{X}) \\ &= \text{argmax}_{\boldsymbol{\beta}} \left\{ -\frac{1}{2\sigma^2}\|\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}\|^2 - \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta} \right\}. \end{aligned} \tag{2.38}$$

It is easy to see that

$$\widehat{\boldsymbol{\beta}}^{\text{MAP}} = (\mathbf{X}^T\mathbf{X} + \sigma^2\boldsymbol{\Sigma}^{-1})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{2.39}$$

This generalized ridge regression can take into account different scales of covariates, by an appropriate choice of $\boldsymbol{\Sigma}$.

2.6.4 Ridge Regression Solution Path

The performance of ridge regression heavily depends on the choice of λ. In practice we only need to compute ridge regression estimates at a fine grid of λ values and then select the best from these candidate solutions. Although ridge regression is easy to compute for a λ owing to its nice closed-form solution expression, the total cost could be high if the process is repeated many times. Through a more careful analysis, one can see that the solutions of ridge regression at a fine grid of λ values can be computed very efficiently via singular value decomposition.

Assume $n > p$ and $\mathbf{X}$ is full rank. The singular value decomposition (SVD) of $\mathbf{X}$ is given by

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

where $\mathbf{U}$ is a $n \times p$ orthogonal matrix, $\mathbf{V}$ is a $p \times p$ orthogonal matrix and

$\mathbf{D}$ is a $p \times p$ diagonal matrix whose diagonal elements are the ordered (from large to small) singular values of $\mathbf{X}$. Then

$$\mathbf{X}^T\mathbf{X} = \mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{V}\mathbf{D}^2\mathbf{V}^T,$$

$$\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I} = \mathbf{V}\mathbf{D}^2\mathbf{V}^T + \lambda\mathbf{I} = \mathbf{V}(\mathbf{D}^2 + \lambda\mathbf{I})\mathbf{V}^T,$$

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1} = \mathbf{V}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{V}^T.$$

The ridge regression estimator $\widehat{\boldsymbol{\beta}}_\lambda$ can now be written as

$$\begin{aligned} \widehat{\boldsymbol{\beta}}_\lambda &= (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y} \\ &= \mathbf{V}(\mathbf{D}^2 + \lambda\mathbf{I})^{-1}\mathbf{D}\mathbf{U}^T\mathbf{Y} \\ &= \sum_{j=1}^{p} \frac{d_j}{d_j^2 + \lambda}\langle \mathbf{U}_j, \mathbf{Y}\rangle \mathbf{V}_j, \end{aligned} \tag{2.40}$$

where d_j is the j^{th} diagonal element of $\mathbf{D}$ and $\langle \mathbf{U}_j, \mathbf{Y}\rangle$ is the inner product between $\mathbf{U}_j$ and $\mathbf{Y}$ and $\mathbf{U}_j$ ($\mathbf{V}_j$ are respectively the j^{th} column of $\mathbf{U}$ and $\mathbf{V}$). In particular, when $\lambda = 0$, ridge regression reduces to OLS and we have

$$\widehat{\boldsymbol{\beta}}^{\text{OLS}} = \mathbf{V}D^{-1}\mathbf{U}^T\mathbf{Y} = \sum_{j=1}^{p} \frac{1}{d_j}\langle \mathbf{U}_j, \mathbf{Y}\rangle \mathbf{V}_j. \tag{2.41}$$

Based on (2.40) we suggest the following procedure to compute ridge regression at a fine grid $\lambda_1, \cdots, \lambda_M$:

1. Compute the SVD of $\mathbf{X}$ and save $\mathbf{U}, \mathbf{D}, \mathbf{V}$.
2. Compute $\mathbf{w}_j = \frac{1}{d_j}\langle \mathbf{U}_j \cdot \mathbf{Y}\rangle \mathbf{V}_j$ for $j = 1, \cdots, p$ and save $\mathbf{w}_j$s.
3. For $m = 1, 2 \cdots, M$,
   (i). compute $\gamma_j = \frac{d_j^2}{d_j^2 + \lambda_m}$
   (ii). compute $\widehat{\boldsymbol{\beta}}_{\lambda_m} = \sum_{j=1}^{p} \gamma_j \mathbf{w}_j$.

The essence of the above algorithm is to compute the common vectors $\{\mathbf{w}_j\}_{j=1}^p$ first and then utilize (2.40).

2.6.5 Kernel Ridge Regression

In this section we introduce a nonparametric generalization of ridge regression. Our discussion begins with the following theorem.

Theorem 2.4 *Ridge regression estimator is equal to*

$$\widehat{\boldsymbol{\beta}}_\lambda = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{Y} \tag{2.42}$$

and the fitted value of Y at $\mathbf{x}$ is

$$\widehat{y} = \mathbf{x}^T\widehat{\boldsymbol{\beta}}_\lambda = \mathbf{x}^T\mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I})^{-1}\mathbf{Y} \tag{2.43}$$

Proof. Observe the following identity

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{X}^T = \mathbf{X}^T\mathbf{X}\mathbf{X}^T + \lambda\mathbf{X}^T = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}).$$

Thus, we have

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

and

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

Then by using (2.27) we obtain (2.42) and hence (2.43). ∎

It is important to see that $\mathbf{X}\mathbf{X}^T$ not $\mathbf{X}^T\mathbf{X}$ appears in the expression for $\widehat{\boldsymbol{\beta}}_\lambda$. Note that $\mathbf{X}\mathbf{X}^T$ is a $n \times n$ matrix and its ij elements is $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Similarly, $\mathbf{x}^T\mathbf{X}^T$ is an n-dimensional vector with the ith element being $\langle \mathbf{x}, \mathbf{x}_i \rangle$ $i = 1, \cdots, n$. Therefore, the prediction by ridge regression boils down to computing the inner product between p-dimensional covariate vectors. This is the foundation of the so-called "*kernel trick*".

Suppose that we use another "inner product" to replace the usual inner product in Theorem 2.4 then we may end up with a new ridge regression estimator. To be more specific, let us replace $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ with $K(\mathbf{x}_i, \mathbf{x}_j)$ where $K(\cdot, \cdot)$ is a known function:

$$\mathbf{x}^T\mathbf{X}^T \to (K(\mathbf{x}, \mathbf{X}_1), \cdots, K(\mathbf{x}, \mathbf{X}_n)),$$

$$\mathbf{X}\mathbf{X}^T \to \mathbf{K} = (K(\mathbf{X}_i, \mathbf{X}_j))_{1 \le i,j \le n}.$$

By doing so, we turn (2.43) into

$$\widehat{y} = (K(\mathbf{x}, \mathbf{X}_1), \cdots, K(\mathbf{x}, \mathbf{X}_n))(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{Y} = \sum_{i=1}^{n} \widehat{\alpha}_i K(\mathbf{x}, \mathbf{X}_i), \tag{2.44}$$

where $\widehat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{Y}$. In particular, the fitted $\mathbf{Y}$ vector is

$$\widehat{\mathbf{Y}} = \mathbf{K}(\mathbf{K} + \lambda\mathbf{I})^{-1}\mathbf{Y}. \tag{2.45}$$

The above formula gives the so-called kernel ridge regression. Because $\mathbf{X}\mathbf{X}^T$ is at least positive semi-definite, it is required that $\mathbf{K}$ is also positive semi-definite. Some widely used kernel functions (Hastie, Tibshirani and Friedman, 2009) include

- *linear kernel*: $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$,
- *polynomial kernel*: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^d$, $d = 2, 3, \cdots$,
- *radial basis kernel*: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$, $\gamma > 0$, which is the *Gaussian kernel*, and $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|}$, $\gamma > 0$, which is the *Laplacian kernel*.

To show how we get (2.45) more formally, let us consider approximate the multivariate regression by using the kernel basis functions $\{K(\cdot, \mathbf{x}_j)\}_{j=1}^n$ so that our observed data are now modeled as

$$Y_i = \sum_{j=1}^{n} \alpha_j K(\mathbf{X}_i, \mathbf{X}_j) + \varepsilon_i$$

or in matrix form $\mathbf{Y} = \mathbf{K}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}$. If we apply the ridge regression

$$\|\mathbf{Y} - \mathbf{K}\boldsymbol{\alpha}\|^2 + \frac{\lambda}{2}\boldsymbol{\alpha}\mathbf{K}\boldsymbol{\alpha},$$

the minimizer of the above problem is

$$\widehat{\boldsymbol{\alpha}} = (\mathbf{K}^T\mathbf{K} + \lambda\mathbf{K})^{-1}\mathbf{K}^T\mathbf{Y} = \{\mathbf{K}(\mathbf{K} + \lambda\mathbf{I})\}^{-1}\mathbf{K}\mathbf{Y},$$

where we use the fact $\mathbf{K}$ is symmetric. Assuming $\mathbf{K}$ is invertible, we easily get (2.45).

So far we have only derived the kernel ridge regression based on heuristics and the kernel trick. In section 2.7 we show that the kernel ridge regression can be formally derived based on the theory of function estimation in a reproducing kernel Hilbert space.

2.7 Regression in Reproducing Kernel Hilbert Space

A *Hilbert space* is an abstract vector space endowed by the structure of an inner product. Let $\mathcal{X}$ be an arbitrary set and $\mathcal{H}$ be a Hilbert space of real-valued functions on $\mathcal{X}$, endowed by the inner product $\langle\cdot,\cdot\rangle_{\mathcal{H}}$. The evaluation functional over the Hilbert space of functions $\mathcal{H}$ is a linear functional that evaluates each function at a point x:

$$L_x : f \to f(x), \forall f \in \mathcal{H}.$$

A Hilbert space $\mathcal{H}$ is called a *reproducing kernel Hilbert space* (RKHS) if, for all $x \in \mathcal{X}$, the map L_x is continuous at any $f \in \mathcal{H}$, namely, there exists some $C > 0$ such that

$$|L_x(f)| = |f(x)| \le C\|f\|_{\mathcal{H}}, \qquad \forall f \in \mathcal{H}.$$

By the Riesz representation theorem, for all $x \in \mathcal{X}$, there exists a unique element $K_x \in \mathcal{H}$ with the reproducing property

$$f(x) = L_x(f) = \langle f, K_x\rangle_{\mathcal{H}}, \qquad \forall f \in \mathcal{H}.$$

Since K_x is itself a function in $\mathcal{H}$, it holds that for every $x' \in \mathcal{X}$, there exists a $K_{x'} \in \mathcal{H}$ such that

$$K_x(x') = \langle K_x,\ K_{x'}\rangle_{\mathcal{H}}.$$

This allows us to define the *reproducing kernel* $K(x, x') = \langle K_x,\ K_{x'}\rangle_{\mathcal{H}}$. From the definition, it is easy to see that the reproducing kernel K is a symmetric and semi-positive function:

$$\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) = \sum_{i,j=1}^{n} c_i c_j \langle K_{x_i},\ K_{x_j}\rangle_{\mathcal{H}} = \Big\| \sum_{i=1}^{n} c_i K_{x_i} \Big\|_{\mathcal{H}}^2 \geq 0,$$

for all $c's$ and $x's$. The reproducing Hilbert space is a class of *nonparametric functions*, satisfying the above properties.

Let $\mathcal{H}_K$ denote the reproducing kernel Hilbert space (RKHS) with kernel $K(\mathbf{x}, \mathbf{x}')$ (Wahba, 1990; Halmos, 2017). Then, the kernel $K(\mathbf{x}, \mathbf{x}')$ admits the eigen-decomposition

$$K(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{\infty} \gamma_j \psi_j(\mathbf{x}) \psi_j(\mathbf{x}'). \tag{2.46}$$

where $\gamma_j \geq 0$ are eigen-values and $\sum_{j=1}^{\infty} \gamma_j^2 < \infty$. Let g and g' be any two functions in $\mathcal{H}_K$ with expansions in terms of these eigen-functions

$$g(\mathbf{x}) = \sum_{j=1}^{\infty} \beta_j \psi_j(\mathbf{x}), \qquad g'(\mathbf{x}) = \sum_{j=1}^{\infty} \beta'_j \psi_j(\mathbf{x})$$

and their inner product is defined as

$$\langle g, g'\rangle_{\mathcal{H}_K} = \sum_{j=1}^{\infty} \frac{\beta_j \beta'_j}{\gamma_j}. \tag{2.47}$$

The functional ℓ_2 norm of $g(\mathbf{x})$ is equal to

$$\|g\|_{\mathcal{H}_K}^2 = \langle g, g\rangle_{\mathcal{H}_K} \quad = \quad \sum_{j=1}^{\infty} \frac{\beta_j^2}{\gamma_j}. \tag{2.48}$$

The first property shows the *reproducibility* of the kernel K.

Theorem 2.5 *Let g be a function in $\mathcal{H}_K$. The following identities hold:*

(i). $\langle K(\cdot, \mathbf{x}'), g\rangle_{\mathcal{H}_K} = g(\mathbf{x}')$,

(ii). $\langle K(\cdot, \mathbf{x}_1), K(\cdot, \mathbf{x}_2)\rangle_{\mathcal{H}_K} = K(\mathbf{x}_1, \mathbf{x}_2)$.

(iii). If $g(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$, *then* $\|g\|_{\mathcal{H}_K}^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)$.

Proof. Write $g(\mathbf{x}) = \sum_{j=1}^{\infty} \beta_j \psi_j(\mathbf{x})$, by (2.46) we have $K(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{\infty} (\gamma_j \psi_j(\mathbf{x}'))\psi_j(\mathbf{x})$. Thus

$$\langle K(\cdot, \mathbf{x}'), g\rangle_{\mathcal{H}_K} \quad = \quad \sum_{j=1}^{\infty} \frac{\beta_j \gamma_j \psi_j(\mathbf{x}')}{\gamma_j} = \sum_{j=1}^{\infty} \beta_j \psi_j(\mathbf{x}') = g(\mathbf{x}').$$

This proves part (i). Now we apply part (i) to get part (ii) by letting $g(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}_2)$.

For part (iii) we observe that

$$\begin{aligned} \|g\|^2_{\mathcal{H}_K} &= \langle \sum_{i=1}^n \alpha_i K(\mathbf{x}, \mathbf{x}_i), \sum_{j=1}^n \alpha_j K(\mathbf{x}, \mathbf{x}_j) \rangle_{\mathcal{H}_K} \\ &= \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j \langle K(\mathbf{x}, \mathbf{x}_i), K(\mathbf{x}, \mathbf{x}_j) \rangle_{\mathcal{H}_K} \\ &= \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j), \end{aligned}$$

where we have used part (ii) in the final step.

■

Consider a general regression model

$$Y = f(\mathbf{X}) + \varepsilon \tag{2.49}$$

where ε is independent of $\mathbf{X}$ and has zero mean and variance σ^2. Given a realization $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ from the above model, we wish to fit the regression function in $\mathcal{H}_K$ via the following penalized least-squares:

$$\widehat{f} = \text{argmin}_{f \in \mathcal{H}_K} \sum_{i=1}^n [Y_i - f(\mathbf{X}_i)]^2 + \lambda \|f\|^2_{\mathcal{H}_K}, \quad \lambda > 0. \tag{2.50}$$

Note that without $\|f\|^2_{\mathcal{H}_K}$ term there are infinite many functions in $\mathcal{H}_K$ that can fit the observations perfectly, i.e., $Y_i = f(\mathbf{X}_i)$ for $i = 1, \cdots, n$. By using the eigen-function expansion of f

$$f(\mathbf{x}) = \sum_{j=1}^\infty \beta_j \psi_j(\mathbf{x}), \tag{2.51}$$

an equivalent formulation of (2.50) is

$$\min_{\{\beta_j\}_{j=1}^\infty} \sum_{i=1}^n [Y_i - \sum_{j=1}^\infty \beta_j \psi_j(\mathbf{X}_i)]^2 + \lambda \sum_{j=1}^\infty \frac{1}{\gamma_j} \beta_j^2. \tag{2.52}$$

Define $\beta_j^* = \frac{\beta_j}{\sqrt{\gamma_j}}$ and $\psi_j^* = \sqrt{\gamma_j}\psi_j$ for $j = 1, 2, \cdots$. Then (2.52) can be rewritten as

$$\min_{\{\beta_j^*\}_{j=1}^\infty} \sum_{i=1}^n [Y_i - \sum_{j=1}^\infty \beta_j^* \psi_j^*(\mathbf{X}_i)]^2 + \lambda \sum_{j=1}^\infty (\beta_j^*)^2. \tag{2.53}$$

The above can be seen as a ridge regression estimate in an infinite dimensional

space. Symbolically, our covariate vector is now $(\psi_1^*(\mathbf{x}), \psi_2^*(\mathbf{x}), \cdots)$ and the enlarged design matrix is

$$\mathbf{\Psi} = \begin{pmatrix} \psi_1^*(\mathbf{X}_1) & \cdots & \psi_j^*(\mathbf{X}_1) & \cdots \\ \vdots & \cdots & \vdots & \cdots \\ \psi_1^*(\mathbf{X}_n) & \cdots & \psi_j^*(\mathbf{X}_n) & \cdots \end{pmatrix}.$$

Because Theorem 2.4 is valid for any finite dimensional covariate space, it is not unreasonable to extrapolate it to the above infinite dimensional setting. The key assumption is that we can compute the inner product in the enlarged space. This is indeed true because

$$\text{inner product} = \sum_{j=1}^{\infty} \psi_j^*(\mathbf{x}_i)\psi_j^*(\mathbf{x}_{i'}) = \sum_{j=1}^{\infty} \gamma_j \psi_j(\mathbf{x}_i)\psi_j(\mathbf{x}_{i'}) = K(\mathbf{x}_i, \mathbf{x}_j).$$

Now we can directly apply the kernel ridge regression formula from Section 2.6.5 to get

$$\widehat{f}(\mathbf{x}) = \sum_{i=1}^{n} \widehat{\alpha}_i K(\mathbf{x}, \mathbf{X}_i), \tag{2.54}$$

where $\mathbf{K} = (K(\mathbf{X}_i, \mathbf{X}_j))_{1 \le i,j \le n}$ and

$$\widehat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{Y}. \tag{2.55}$$

We have derived (2.54) by extrapolating Theorem 2.4 to an infinite dimensional space. Although the idea seems correct, we still need a rigorous proof. Moreover, Theorem 2.4 only concerns ridge regression, but it turns out that (2.54) can be made much more general.

Theorem 2.6 *Consider a general loss function $L(y, f(\mathbf{x}))$ and let*

$$\widehat{f} = \text{argmin}_{f \in \mathcal{H}_K} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + P_\lambda(\|f\|_{\mathcal{H}_K}), \quad \lambda > 0.$$

where $P_\lambda(t)$ is a strictly increasing function on $[0, \infty)$. Then we must have

$$\widehat{f}(\mathbf{x}) = \sum_{i=1}^{n} \widehat{\alpha}_i K(\mathbf{x}, \mathbf{X}_i) \tag{2.56}$$

where $\widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \cdots, \widehat{\alpha}_n)$ is the solution to the following problem

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} L\left(y_i, \sum_{j=1}^{n} \alpha_j K(\mathbf{x}, \mathbf{x}_j)\right) + P_\lambda(\sqrt{\boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}}). \tag{2.57}$$

Proof. Any function f in $\mathcal{H}_K$ can be decomposed as the sum of two functions: one is in the span $\{K(\cdot, \mathbf{X}_1), \cdots, K(\cdot, \mathbf{X}_n)\}$ and the other is in the orthogonal complement. In other words, we write

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{X}_i) + r(\mathbf{x})$$

where $\langle r(\mathbf{x}), K(\mathbf{x}, \mathbf{X}_i)\rangle_{\mathcal{H}_K} = 0$ for all $i = 1, 2, \cdots, n$. By part (i) of Theorem 2.5 we have

$$r(\mathbf{x}_i) = \langle r, K(\cdot, \mathbf{X}_i)\rangle_{\mathcal{H}_K} = 0, \quad 1 \le i \le n.$$

Denote by $g(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}, \mathbf{X}_i)$. Then we have $g(\mathbf{X}_i) = f(\mathbf{X}_i)$ for all i, which implies

$$\sum_{i=1}^{n} L(Y_i, f(\mathbf{X}_i)) = \sum_{i=1}^{n} L(Y_i, g(\mathbf{X}_i)). \tag{2.58}$$

Moreover, we notice

$$\begin{aligned} \|f\|^2_{\mathcal{H}_K} &= \langle g + r, g + r\rangle_{\mathcal{H}_K} \\ &= \langle g, g\rangle_{\mathcal{H}_K} + \langle r, r\rangle_{\mathcal{H}_K} + 2\langle g, r\rangle_{\mathcal{H}_K} \end{aligned}$$

and

$$\langle g, r\rangle_{\mathcal{H}_K} = \sum_{i=1}^{n} \alpha_i \langle K(\cdot, \mathbf{X}_i), r\rangle_{\mathcal{H}_K} = 0.$$

Thus $\|f\|^2_{\mathcal{H}_K} = \|g\|^2_{\mathcal{H}_K} + \|r\|^2_{\mathcal{H}_K}$. Because $P_\lambda(\cdot)$ is a strictly increasing function, we then have

$$P_\lambda(\|f\|_{\mathcal{H}_K}) \ge P_\lambda(\|g\|_{\mathcal{H}_K}) \tag{2.59}$$

and the equality holds if and only if $f = g$. Combining (2.58) and (2.59) we prove (2.56).

To prove (2.57), we use (2.56) and part (iii) of Theorem 2.5 to write

$$\|f\|^2_{\mathcal{H}_K} = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}. \tag{2.60}$$

Hence $P_\lambda(\|f\|_{\mathcal{H}_K}) = P_\lambda(\sqrt{\boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}})$ under (2.56) . ∎

Theorem 2.6 is known as the *representer theorem* (Wahba, 1990). It shows that for a wide class of statistical estimation problems in a RKHS, although the criterion is defined in an infinite dimensional space, the solution always has a finite dimensional representation based on the kernel functions. This provides a solid mathematical foundation for the kernel trick without resorting to any optimization/computational arguments.

Let the loss function in Theorem 2.6 be the squared error loss and $P_\lambda(t) = \lambda t^2$. Then Theorem 2.6 handles the problem defined in (2.50) and (2.57) reduces to

$$\min_{\boldsymbol{\alpha}} \|\mathbf{Y} - \mathbf{K}\boldsymbol{\alpha}\|^2 + \lambda \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}. \tag{2.61}$$

Table 2.1: A list of Commonly used kernels.

Linear kernel	$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$
Polynomial kernel	$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \langle \mathbf{x}_i, \mathbf{x}_j \rangle)^d$
Gaussian kernel	$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$
Laplacian kernel	$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|}$

It is easy to see the solution is

$$\widehat{\boldsymbol{\alpha}} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{Y}.$$

which is identical to (2.55). The fitted multivariate nonparametric regression function is given by (2.56). In practice, one takes a kernel function from the list of linear, polynomial, Gaussian or Laplacian kernels given in Table 2.1. It remains to show how to choose the regularization parameter λ (and γ for Gaussian and Laplacian kernels) to optimize the prediction performance. This can be done by cross-validation methods outlined in the next section.

2.8 Leave-one-out and Generalized Cross-validation

We have seen that both ridge regression and the kernel ridge regression use a *tuning parameter* λ. In practice, we would like to use the data to pick a data-driven λ in order to achieve the "best" estimation/prediction performance. This problem is often called tuning parameter selection and is ubiquitous in modern statistics and machine learning. A general solution is k-fold *cross-validation* (CV), such as 10-fold or 5-fold CV. k-fold CV estimates prediction errors as follows.

- Divide data randomly and evenly into k subsets.
- Use one of the subsets as the *testing set* and the remaining $k-1$ subsets of data as a *training set* to compute testing errors.
- Compute testing errors for each of k subsets of data and average these testing errors.

An interesting special case is the n-fold CV, which is also known as the *leave-one-out CV*.

In this section we focus on regression problems under the squared error loss. Following the above scheme, the leave-one-out CV error, using the quadratic loss, is defined as

$$\text{CV} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \widehat{f}^{(-i)}(\mathbf{X}_i))^2, \tag{2.62}$$

where $\widehat{f}^{(-i)}(\mathbf{X}_i)$ is the predicted value at $\mathbf{x}_i$ computed by using all the data except the ith observation. So in principle we need to repeat the same data

Table 2.2: A list of commonly used regression methods and their $\mathbf{S}$ matrices. d_js are the singular values of $\mathbf{X}$ and γ_is are the eigenvalues of $\mathbf{K}$.

Method	$\mathbf{S}$	$\operatorname{tr}\mathbf{S}$
Multiple Linear Regression	$\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$	p
Ridge Regression	$\mathbf{X}(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T$	$\sum_{j=1}^{p}\frac{d_j^2}{d_j^2+\lambda}$
Kernel Regression in RKHS	$\mathbf{K}(\mathbf{K}+\lambda\mathbf{I})^{-1}$	$\sum_{i=1}^{n}\frac{\gamma_i}{\gamma_i+\lambda}$

fitting process n times to compute the leave-one-out CV. Fortunately, we can avoid much computation for many popular regression methods.

A fitting method is called a *linear smoother* if we can write

$$\widehat{\mathbf{Y}} = \mathbf{S}\mathbf{Y} \tag{2.63}$$

for any dataset $\{(\mathbf{X}_i, Y_i\}_1^n$ where $\mathbf{S}$ is a $n \times n$ matrix that only depends on $\mathbf{X}$. Many regression methods are linear smoothers with different $\mathbf{S}$ matrices. See Table 2.2.

Assume that a linear smoother is fitted on $\{\mathbf{X}_i, Y_i\}_{i=1}^n$. Let $\mathbf{x}$ be a new covariate vector and $\widehat{f}(\mathbf{x})$ be its the predicted value by using the linear smoother. We then augment the dataset by including $(\mathbf{x}, \widehat{f}(\mathbf{x}))$ and refit the linear smoother on this augmented dataset. The linear smoother is said to be *self-stable* if the fit based on the augmented dataset is identical to the fit based on the original data regardless of $\mathbf{x}$.

It is easy to check that the three linear smoothers in Table 2.2 all have the self-stable property.

Theorem 2.7 *For a linear smoother $\widehat{\mathbf{Y}} = \mathbf{S}\mathbf{Y}$ with the self-stable property, we have*

$$Y_i - \widehat{f}^{(-i)}(\mathbf{X}_i) = \frac{Y_i - \widehat{Y}_i}{1 - S_{ii}}, \tag{2.64}$$

and its leave-one-out CV error is equal to $\frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i-\widehat{Y}_i}{1-S_{ii}}\right)^2$.

Proof. We first apply the linear smoother to all the data except the ith to compute $\widehat{f}^{(-i)}(\mathbf{X}_i)$. Write $\widetilde{y}_j = y_j$ for $j \neq i$ and $\widetilde{y}_i = \widehat{f}^{(-i)}(\mathbf{X}_i)$. Then we apply the linear smoother to the following working dataset:

$$\{(\mathbf{X}_j, Y_j), j \neq i, (\mathbf{X}_i, \widetilde{Y}_i)\}$$

The self-stable property implies that the fit stays the same. In particular,

$$\widetilde{Y}_i = \widehat{f}^{(-i)}(\mathbf{X}_i) = (\mathbf{S}\widetilde{\mathbf{Y}})_i = S_{ii}\widetilde{Y}_i + \sum_{j\neq i} S_{ij}Y_j \tag{2.65}$$

and

$$\widehat{Y}_i = (\mathbf{SY})_i = S_{ii}Y_i + \sum_{j\neq i} S_{ij}Y_j. \tag{2.66}$$

Combining (2.65) and (2.66) yields

$$\widetilde{Y}_i = \frac{\widehat{Y}_i - S_{ii}Y_i}{1 - S_{ii}}.$$

Thus,

$$Y_i - \widetilde{Y}_i = Y_i - \frac{\widehat{Y}_i - S_{ii}Y_i}{1 - S_{ii}} = \frac{Y_i - \widehat{Y}_i}{1 - S_{ii}}.$$

The proof is now complete. ∎

Theorem 2.7 shows a nice shortcut for computing the leave-one-out CV error of a self-stable linear smoother. For some smoothers $\mathrm{tr}\,\mathbf{S}$ can be computed more easily than its diagonal elements. To take advantage of this, *generalized cross-validation* (GCV) (Golub, Heath and Wahba, 1979) is a convenient computational approximation to the leave-one-out CV error. Suppose that we approximate each diagonal elements of $\mathbf{S}$ by their average which equals $\frac{\mathrm{tr}\,\mathbf{S}}{n}$, then we have

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{Y_i - \widehat{Y}_i}{1 - S_{ii}}\right)^2 \approx \frac{1}{n}\frac{\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2}{(1 - \frac{\mathrm{tr}\,\mathbf{S}}{n})^2} := \mathrm{GCV}.$$

In the literature $\mathrm{tr}\,\mathbf{S}$ is called the effective degrees of freedom of the linear smoother. Its rigorous justification is based on Stein's unbiased risk estimation theory (Stein, 1981; Efron, 1986). In Table 2.2 we list the degrees of freedom of three popular linear smoothers.

Now we are ready to handle the tuning parameter selection issue in the linear smoother. We write $\mathbf{S} = \mathbf{S}_\lambda$ and

$$\mathrm{GCV}(\lambda) = \frac{1}{n}\frac{\mathbf{Y}^T(\mathbf{I} - \mathbf{S}_\lambda)^2\mathbf{Y}}{(1 - \frac{\mathrm{tr}\,\mathbf{S}_\lambda}{n})^2}.$$

According to GCV, the best λ is given by

$$\lambda^{\mathrm{GCV}} = \mathrm{argmin}_\lambda \frac{1}{n}\frac{\mathbf{Y}^T(\mathbf{I} - \mathbf{S}_\lambda)^2\mathbf{Y}}{(1 - \frac{\mathrm{tr}\,\mathbf{S}_\lambda}{n})^2}.$$

2.9 Exercises

2.1 Suppose that $\{\mathbf{X}_i, Y_i\}$, $i = 1, \cdots, n$ is a random sample from linear regression model (2.1). Assume that the random error $\varepsilon \sim N(0, \sigma^2)$ and is independent of $\mathbf{X} = (X_1, \cdots, X_p)^T$.

(a) Show that the maximum likelihood estimate of $\boldsymbol{\beta}$ is the same as its least squares estimator, while the maximum likelihood estimate of σ^2 is RSS/n, where $\text{RSS} = \sum_{i=1}^{n}(Y_i - \mathbf{X}_i^T\widehat{\boldsymbol{\beta}})^2$ is the residual sum-of-squares and $\widehat{\boldsymbol{\beta}}$ is the least squares estimator.

(b) Assume that $\mathbf{X}$ is of full rank. Show that $\text{RSS} \sim \sigma^2\chi^2_{n-p}$.

(c) Prove that $1-\alpha$ CI for β_j is $\widehat{\beta}_j \pm t_{n-p}(1-\alpha/2)\sqrt{v_j\,\text{RSS}/(n-p)}$, where v_j is the j^{th} diagonal element of $(\mathbf{X}^T\mathbf{X})^{-1}$.

(d) Dropping the normality assumption, if $\{\mathbf{X}_i\}$ are independent and identically distributed from a population with $\text{E}\,\mathbf{X}_1\mathbf{X}_1^T = \boldsymbol{\Sigma}$ and independent of $\{\varepsilon_i\}_{i=1}^n$, which are independent and identically distributed from a population with $\text{E}\,\varepsilon = 0$ and $\text{Var}(\varepsilon) = \sigma^2$, show that

$$\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} N(0, \sigma^2\boldsymbol{\Sigma}^{-1}).$$

2.2 Suppose that a random sample of size n from linear regression model (2.1), where the random error $\varepsilon \sim N(0,\sigma^2)$ and is independent of $(X_1, \cdots, X_p)$. Consider a general linear hypothesis $H_0 : \mathbf{C}\boldsymbol{\beta} = \mathbf{h}$ versus $H_0 : \mathbf{C}\boldsymbol{\beta} \neq \mathbf{h}$, where $\mathbf{C}$ is $q \times p$ constant matrix with rank q ($\leq p$), and $\mathbf{h}$ is a $q \times 1$ constant vector.

(a) Derive the least squares estimator of $\boldsymbol{\beta}$ under H_0, denoted by $\widehat{\boldsymbol{\beta}}_0$.

(b) Define $\text{RSS}_1 = \|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2$ and $\text{RSS}_0 = \|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_0\|^2$, the residual sum-of-squares under H_1 and H_0. Show that $\text{RSS}_1/\sigma^2 \sim \chi^2_{n-p}$. Further, under the null hypothesis H_0, $(\text{RSS}_0 - \text{RSS}_1)/\sigma^2 \sim \chi^2_q$ and is independent of RSS_1.

(c) Show that Under H_0, $F = \{(\text{RSS}_0 - \text{RSS}_1)/q\}/\{\text{RSS}_1/(n-p)\}$ follows an $F_{q,n-p}$ distribution.

(d) Show that the F-test for H_0 is equivalent to the likelihood ratio test for H_0.

2.3 Suppose that we have n independent data $Y_i \sim N(\mu, \sigma_i^2)$, where $\sigma_i = \sigma^2 v_i$ with known v_i. Use the weighted least-squares method to find an estimator of μ. Show that it is the best linear unbiased estimator. Compare the variance of the sample mean $\bar{y}$ with that of the weighted least-squares estimator $v_i^2 = \log(i+1)$ when $n = 20$.

2.4 Consider the linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim N(0, \boldsymbol{\Sigma})$, and $\mathbf{X}$ is of full rank.

(a) Show that the general least-squares estimator, which minimizes $(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T\boldsymbol{\Sigma}^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$, is the best linear unbiased estimator. More precisely, for any vector $\mathbf{c} \neq 0$, $\mathbf{c}^T\widehat{\boldsymbol{\beta}}$ is the best linear estimator of β. Do we need the normality assumption?

(b) Deduce from Part (a) that the weighted least-squares estimator is the best linear unbiased estimator, when the error distribution is uncorrelated.

(c) If $\mathbf{\Sigma}$ is the equi-correlation matrix with unknown correlation ρ, what is the solution to Part (a)?

2.5 Suppose that $Y_1, \cdots, Y_n$ are random variables with common mean μ and covariance matrix $\sigma^2\mathbf{V}$, where $\mathbf{V}$ is of the form $v_{ii} = 1$ and $v_{ij} = \rho$ $(0 < \rho < 1)$ for $i \neq j$.

(a) Find the generalized least squares estimate of μ.

(b) Show that it is the same as the ordinary least squares estimate.

2.6 Suppose that data $\{X_{i1}, \cdots, X_{ip}, Y_i\}, i = 1, \cdots, n$, are an independent and identically distributed sample from the model

$$Y = f(X_1\beta_1 + \cdots + X_p\beta_p + \varepsilon),$$

where $\varepsilon \sim N(0, \sigma^2)$ with unknown σ^2, and $f(\cdot)$ is a known, differentiable, strictly increasing, non-linear function.

(a) Consider transform $Y_i^* = h(Y_i)$, where $h(\cdot)$ is a differentiable function yet to be determined. Show that $\text{Var}(Y_i^*)$ =constant for all i leads to the equation: $[h'\{f(u)\}]^2\{f'(u)\}^2$ =constant for all u.

(b) Let $f(x) = x^p$ $(p > 1)$. Find the correspondent $h(\cdot)$ using the equation in (a).

(c) Let $f(x) = \exp(x)$. Find the corresponding h transform.

2.7 The data set 'hkepd.txt' consist of daily measurements of levels of air pollutants and the number of total hospital admissions for circulatory and respiratory problems from January 1, 1994 to December 31, 1995 in Hong Kong. This data set can be downloaded from this book website. Of interest is to investigate the association between the number of total hospital admissions and the levels of air pollutants.

We set the Y variable to be the number of total hospital admissions and the X variables the levels of air pollutants. Define

X_1 = the level of sulfur dioxide $(\mu g/m^3)$;
X_2 = the level of nitrogen dioxide $(\mu g/m^3)$;
X_3 = the level of dust $(\mu g/m^3)$.

(a) Fit the data to the following linear regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon, \tag{2.67}$$

and test whether the level of each air pollutant has significant impact on the number of total hospital admissions.

(b) Construct residual plots and examine whether the random error approximately follows a normal distribution.

(c) Take $Z = \log(Y)$ and fit the data to the following linear regression model

$$Z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \varepsilon, \tag{2.68}$$

and test whether the level of each air pollutant has significant impact on the logarithm of the number of total hospital admissions.

(d) Construct residual plots based on model (2.68) and compare this residual plot with the one obtained in Part (b).

(e) Since the observations in this data set were collected over time, it is of interest to explore the potential seasonal trends on the total number of hospital admissions. For simplicity, define t to be the day at which data were collected. This corresponds to the first column of the data set. Consider time-varying effect model

$$Z = \beta_0(t) + \beta_1(t)X_1 + \beta_2(t)X_2 + \beta_3(t)X_3 + \varepsilon, \tag{2.69}$$

which allows the effects of predictors varying over time. Model (2.69) indeed is a nonparametric regression model. Fit model (2.69) to the data by using nonparametric regression techniques introduced in Section 2.5.

(f) For model (2.69), construct an F-type test for $H_0 : \beta_3(\cdot) \equiv 0$ (i.e., the level of dust not significant) by comparing their residual sum of squares under H_0 and H_1.

2.8 The data set 'macroecno.txt' consist of 129 macroecnomic time series and can be downloaded from this book website. Let the response $Y_t = \log(\text{PCE}_t)$ be the personal consumption expenditure. Define covariates as follows.

$$X_{t,1} = \log(\text{PCE}_{t-1}), \quad X_{t,2} = \text{Unrate}_{t-1}, \; X_{t,3} = \Delta\log(\text{IndPro}_t),$$
$$X_{t,4} = \Delta\log(\text{M2Real}_t), \; X_{t,5} = \Delta\log(\text{CPI}_t), \quad X_{t,6} = \Delta\log(\text{SPY}_t),$$
$$X_{t,7} = \text{HouSta}_t, \quad X_{t,8} = \text{FedFund}_t$$

Set the last 10 years data as testing data and remaining as training data. Conduct linear regression analysis and address the following questions.

(a) What are $\widehat{\sigma}^2$, adjusted R^2 and insignificant variables?

(b) Conduct the stepwise deletion, eliminating one least significant variable at a time (by looking at the small $|t|$-statistic) until all variables are statistically significant. Name this model as model $\widehat{\mathcal{M}}$. (The function `step` can do the job automatically)

(c) Using model $\widehat{\mathcal{M}}$, what are root mean-square prediction error and mean absolute deviation prediction error for the test sample?

(d) Compute the standardized residuals. Present the time series plot of the residuals, fitted values versus the standardized residuals, and QQ plot for the standardized residuals.

(e) Compare the result in part (c) with the nonparametric model using Gaussian kernel with $\gamma = 1/4$ (standardize predictors first) and λ chosen by 5-fold CV or GCV.

2.9 Zillow is an online real estate database company that was founded in 2006. The most important task for Zillow is to predict the house price. However, their accuracy has been criticized a lot. According to Fortune, "Zillow has Zestimated the value of 57 percent of U.S. housing stock, but only 65 percent of that could be considered 'accurate' by its definition, within 10 percent of the actual selling price. And even that accuracy isn't equally distributed". Therefore, Zillow needs your help to build a housing pricing model to improve their accuracy. Download the data from the book website, and read the data (traing data: 15129 cases, testing data: 6484 cases)

```
train.data <- read.csv('train.data.csv', header=TRUE)
test.data <- read.csv('test.data.csv', header=TRUE)
train.data$zipcode <- as.factor(train.data$zipcode)
test.data$zipcode <- as.factor(test.data$zipcode)
```

where the last two lines make sure that zip code is treated as factor. Let $\mathcal{T}$ be a test set, define out-of-sample R^2 as of a prediction method $\{\widehat{y}_i^{pred}\}$ as

$$R^2 = 1 - \frac{\sum_{i\in\mathcal{T}}(y_i - \widehat{y}_i^{pred})^2}{\sum_{i\in\mathcal{T}}(y_i - \bar{y}^{pred})^2},$$

where $\bar{y}^{pred} = \text{ave}(\{y_i\}_{i\in\mathcal{T}_0})$ and $\mathcal{T}_0$ is the training set.

(a) Calculate out-of-sample R^2 using variables "bedrooms", "bathrooms", "sqft_living", and "sqft_lot".

(b) Calculate out-of-sample R^2 using the 4 variables above along with interaction terms.

(c) Compare the result with the nonparametric model using Gaussian kernel with $\gamma = 0.3^{-2}/2$ and $\gamma = 0.1^{-2}/2$ (standardize predictors first) and λ chosen by 5-fold CV or GCV. **Hint**: To speed up computation, please divide data randomly into 10 pieces and get 10 predicted values based on 10 fitted kernel models. Use the median as these 10 predicted values as your final prediction.

(d) Add the factor zipcode to (b) and compute out-of-sample R^2.

(e) Add the following additional variables to (d): $X_{12} = I(view == 0)$, $X_{13} = L^2$, $X_{13+i} = (L - \tau_i)_+^2$, $i = 1, \cdots, 9$, where τ_i is $10 * i^{th}$ percentile and L is the size of living area ("sqft_living"). Compute out-of-sample R^2.

(f) Why do you see the increased out-of-sample R^2 with modeling complexity?


Copyright: Jianqing Fan
Draft—Don't circulate
DEI
SVB NVMINE
VIGET

Chapter 3

Introduction to Penalized Least-Squares

Variable selection is vital to high-dimensional statistical learning and inference, and is essential for scientific discoveries and engineering innovation. Multiple regression is one of the most classical and useful techniques in statistics. This chapter introduces *penalized least-squares* approaches to variable selection problems in multiple regression models. They provide fundamental insights and basis for *model selection* problems in other more sophisticated models.

3.1 Classical Variable Selection Criteria

In this chapter, we will follow the notation and model introduced in Chapter 2. To reduce noise accumulation and to enhance interpretability, variable selection techniques have been popularly used even in traditional statistics. When the number of predictors p is larger than the sample size n, the model parameters in the linear model (2.2) are not identifiable. What makes them estimable is the *sparsity* assumption on the regression coefficients $\{\beta_j\}_{j=1}^p$: many of them are too small to matter, so they are ideally regarded as zero. Throughout this chapter, we assume the linear model (2.1):

$$Y = \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon,$$

unless otherwise stated.

3.1.1 Subset selection

One of the most popular and intuitive variable selection techniques is the *best subset* selection. Among all models with m variables, pick the one with the smallest residual sum of squares (2.3), which is denoted by RSS_m. This is indeed very intuitive: among the models with the same complexity, a better fit is preferable. This creates a sequence of submodels $\{\mathcal{M}_m\}_{m=0}^p$ indexed by the model size m. The choice of the model size m will be further illuminated in Section 3.1.3.

Computation of the best subset method is expensive even when p is moderately large. At each step, we compare the goodness-of-fit among $\binom{p}{m}$ models of size m and there are 2^p submodels in total. Intuitive and greedy algorithms

have been introduced to produce a sequence of submodels with different numbers of variables. These include *forward selection* also called *stepwise addition*, *backward elimination* also named *stepwise deletion*, and *stepwise regression*. See for example, Weisberg (2005).

Forward selection recruits one additional regressor at a time to optimize the fit. Starting with $\mathcal{M}_0^a$ as the empty set, at step m one chooses a variable not in $\mathcal{M}_{m-1}^a$ along with the $m-1$ variables in $\mathcal{M}_{m-1}^a$ to minimize the RSS. At step m, only $p-m+1$ submodels instead of $\binom{p}{m}$ of size m are fitted and compared. The total number of regressions is only $p(p+1)/2$, which is considerably less than 2^p. Of course, the sequence of submodels $\{\mathcal{M}_m^a\}_{m=0}^p$ may not have been as good of a fit as the models produced by the best subset selection.

Backward elimination deletes one least statistically significant variable in each fitted model. Starting from the full model $\mathcal{M}_p$, denoted also by $\mathcal{M}_p^d$, one eliminates the least statistically significant variable in the full model, resulting in a model $\mathcal{M}_{p-1}^d$. We then use the variables in $\mathcal{M}_{p-1}^d$ to fit the data again and delete the least statistically significant variable to obtain model $\mathcal{M}_{p-2}^d$, continuing this process to yield a sequence of models $\{\mathcal{M}_m^d\}_{m=0}^p$.

In classical statistics, one does not produce the full sequence of the models in the forward selection and backward elimination methods. One often sets a very simple stopping criterion such as when all variables are statistically significant (e.g. P-values for each fitted coefficient is smaller than 0.05).

When p is larger than n, backward elimination cannot be applied since we cannot fit the full model. Yet, the forward selection can still be used to select a sequence of submodels. When $p < n$ or when p is relatively large compared to n, backward elimination cannot produce a stable selection process, but forward selection can as long as it is stopped early enough. These are the advantages of the forward selection algorithm.

Stepwise regression is a combination of backward elimination and forward selection procedures. We omit its details. Other greedy algorithms include *matching pursuit* (Mallot and Zhang, 1993), which picks the most correlated variable with the residuals from the previous step of fitting, also referred to as *partial residuals*, and runs the univariate regression to fit the partial residuals. See Section 3.5.11 for additional details.

3.1.2 Relation with penalized regression

Best subset selection can be regarded as *penalized least-squares* (PLS). Let $\|\boldsymbol{\beta}\|_0$ be the L_0-norm of the vector $\boldsymbol{\beta}$, which counts the number of non-vanishing components of $\boldsymbol{\beta}$. Consider the penalized least-squares with L_0 penalty:

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_0. \tag{3.1}$$

The procedure is also referred to as *complexity* or *entropy* based PLS. Clearly, given the model size $\|\boldsymbol{\beta}\|_0 = m$, the solution to the penalized least-squares (3.1) is the best subset selection. The computational complexity is NP-hard.

The stepwise algorithms in the last subsection can be regarded as greedy (approximation) algorithms for penalized least-squares (3.1).

Recall RSS_m is the smallest residual sum of squares among models with size m. With this definition, minimization of (3.1) can be written as

$$\text{RSS}_m + \lambda m. \tag{3.2}$$

The optimal model size is obtained by minimizing (3.2) with respect to m. Clearly, the regularization parameter λ dictates the size of the model. The larger the λ, the larger the penalty on the model complexity m, the smaller the selected model.

3.1.3 Selection of regularization parameters

The best subset technique does not tell us the choice of model size m. The criterion used to compare two models is usually the prediction error. For a completely new observation $(\mathbf{X}^{*T}, Y^*)$, the *prediction error* of using model $\mathcal{M}_m$ is

$$\text{PE}(\mathcal{M}_m) = \text{E}(Y^* - \widehat{\boldsymbol{\beta}}_m^T \mathbf{X}^*_{\mathcal{M}_m})^2,$$

where $\widehat{\boldsymbol{\beta}}_m$ is the fitted regression coefficient vector with m variables, $\mathbf{X}^*_{\mathcal{M}_m}$ is the subvector of $\mathbf{X}^*$ with the selected variables, and the expectation is taken only with respect to the new random variable $(\mathbf{X}^{*T}, Y^*)$.

An unbiased estimation of the prediction error $n\text{PE}(\mathcal{M}_m)$ was derived by Mallows (1973) (after ignoring a constant; see Section 3.6.1 for a derivation):

$$C_p(m) = \text{RSS}_m + 2\sigma^2 m. \tag{3.3}$$

This corresponds to taking $\lambda = 2\sigma^2$ in the penalized least-squares problem (3.1) or (3.2). The parameter m is chosen to minimize (3.3), which is often referred to as *Mallow's C_p criterion*.

Akaike (1973, 1974) derived an approximately unbiased estimate of the prediction error (in terms of the Kullback-Leibler divergence) in a general likelihood based model. His work is regarded as one of the important breakthroughs in statistics in the twentieth century. Translating his criterion into the least-squares setting, it becomes

$$\text{AIC}(m) = \log(\text{RSS}_m/n) + 2m/n,$$

which is called the *Akaike information criterion* (AIC). Note that when $\text{RSS}_m/n \approx \sigma^2$, which is correct when $\mathcal{M}_m$ contains the true model, by Taylor's expansion,

$$\begin{aligned} \log(\text{RSS}_m/n) &= \log\sigma^2 + \log(1 + \text{RSS}_m/(n\sigma^2) - 1) \\ &\approx \log\sigma^2 + (\text{RSS}_m/(n\sigma^2) - 1). \end{aligned}$$

Therefore,

$$\text{AIC}(m) \approx [\text{RSS}_m + 2\sigma^2 m]/(n\sigma^2) + \log\sigma^2 - 1,$$

which is approximately the same as the C_p criterion (3.3) after ignoring the affine transformation.

Many information criteria have been derived since the pioneering work of Akaike and Mallow. They correspond to different choices of λ in

$$\mathrm{IC}(m) = \log(\mathrm{RSS}_m/n) + \lambda m/n. \tag{3.4}$$

Examples include

- *Bayesian information criterion* (BIC, Schwarz, 1978): $\lambda = \log(n)\sigma^2$;
- *ϕ-criterion* (Hannan and Quinn, 1979; Shibata, 1984): $\lambda = c(\log\log n)\sigma^2$;
- Risk inflation criterion (RIC, Foster and George, 1994): $\lambda = 2\log(p)\sigma^2$.

Using the Taylor expansion above, the information criterion (3.4) is asymptotically equivalent to (3.2). An advantage of using these information criterions over criterion (3.2) is that they do not need to estimate σ^2. But this also creates the bias issue. In particular, when a submodel contains modeling biases, AIC is no longer an approximately unbiased estimator. The issue of model selection consistency has been thoroughly studied in Shao (1997).

In summary, the best subset method along with an information criterion corresponds to the L_0-penalized least-squares with penalty parameters λ being a multiple 2, $\log(n)$, $c\log\log n$, and $2\log p$ of σ^2, respectively for the AIC, BIC, ϕ-criterion, and RIC.

Cross-validation (Allen 1974; Stone, 1974) is a novel and widely applicable idea for estimating prediction error of a model. It is one of the most widely used and innovative techniques in statistics. It involves partitioning a sample of data into a *training set* used to estimate model parameters and a *testing set* reserved for validating the analysis of the fitted model. See Section 2.8.

In k-fold cross-validation, the original sample is randomly partitioned into k approximately equal-sized subsamples with index sets $\{\mathcal{S}_j\}_{j=1}^k$. Of the k subsamples, a single subsample $\mathcal{S}_k$ is retained as the validation set, and the remaining data $\{\mathcal{S}_j\}_{j\neq k}$ are used as a training set. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data. The prediction error of the k-fold cross-validation is computed as

$$\mathrm{CV}_k(m) = n^{-1}\sum_{j=1}^{k}\Big\{\sum_{i\in\mathcal{S}_j}(Y_i - \widehat{\boldsymbol{\beta}}_{m,-\mathcal{S}_j}^T\mathbf{X}_{i,\mathcal{M}_m})^2\Big\}, \tag{3.5}$$

where $\widehat{\boldsymbol{\beta}}_{m,-\mathcal{S}_j}$ is the fitted coefficients of the submodel $\mathcal{M}_m$ without using the data indexed in $\mathcal{S}_j$. The number of fittings is k, which is much smaller than n. In practice, the popular choice of k is 5 or 10. An interesting choice of k is n, which is called the leave-one-out cross-validation. The leave-one-out CV error of the submodel $\mathcal{M}_m$ is

$$\mathrm{CV}(m) = n^{-1}\sum_{i=1}^{n}(Y_i - \widehat{\boldsymbol{\beta}}_{m,-i}^T\mathbf{X}_{i,\mathcal{M}_m})^2. \tag{3.6}$$

In general, the leave-one-out CV error is expensive to compute. For multiple linear regression and other linear smoothers with *self-stable* property, there is a neat formula for computing CV(m) without fitting the model n times. See Theorem 2.7 in Section 2.8 of Chapter 2. Another simplification of CV(m) is to use generalized cross-validationGeneralized Cross-Validation (GCV, Craven and Wahba, 1979), defined by

$$\text{GCV}(m) = \frac{\text{RSS}_m}{n(1-m/n)^2}. \tag{3.7}$$

By using a simple Taylor expansion,

$$(1-m/n)^{-2} = 1 + 2m/n + o(m/n)$$

and $\text{RSS}_m/n \approx \sigma^2$, one can easily see that GCV(m) is approximately the same as $C_p(m)/n$.

A classical choice of m is to maximize the adjusted multiple R^2, defined by

$$R^2_{adj,m} = 1 - \frac{n-1}{n-m}\frac{\text{RSS}_m}{\text{RSS}_0}, \tag{3.8}$$

where RSS_0 is the sample standard deviation of the response variable $\{Y_i\}$. This is equivalent to minimizing $\text{RSS}_m/(n-m)$. Derived the same way as the GCV, it corresponds to approximately $\lambda = \sigma^2$ in (3.2).

3.2 Folded-concave Penalized Least Squares

The complexity based PLS (3.1) possesses many nice statistical properties, as documented in the paper by Barron, Birgé and Massart (1999). However, its minimization problem is impossible to carry out when the dimensionality is high. A natural relaxation is to replace the discontinuous L_0-penalty by more regular functions. This results in penalized least-squares

$$\begin{aligned} Q(\boldsymbol{\beta}) &= \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} p_\lambda(|\beta_j|) \\ &\equiv \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \|p_\lambda(|\boldsymbol{\beta}|)\|_1, \end{aligned} \tag{3.9}$$

where $p_\lambda(\cdot)$ is a penalty function in which the *regularization* or *penalization parameters* λ are the same for convenience of presentation.

A natural choice is $p_\lambda(\theta) = \lambda\theta^2/2$, whose solution is ridge regression

$$\widehat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + n\lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{Y}, \tag{3.10}$$

which is also called the *Tikhonov regularization* (Tikhonov, 1943). The estimator shrinks all components toward zero, but none of them are actually zero. It

does not have a model selection property and creates biases for large parameters. In order to reduce the bias, Frank and Friedman (1993) propose to use $p_\lambda(\theta) = \lambda|\theta|^q$ for $0 < q < 2$, called the *bridge regression*, which bridges the best subset selection (penalized L_0) and ridge regression (penalized L_2). Donoho and Johnstone (1994), Tibshirani (1996) and Chen, Donoho and Sanders (1998) observe that penalized L_1 regression leads to a sparse minimizer and hence possesses a variable selection property. The procedure is called *Lasso* by Tibshirani (1996), for 'least absolute shrinkage and selection operator'. Unlike the complexity penalty $p_\lambda(|\theta|) = \lambda I(|\theta| \neq 0)$, Lasso solves a convex optimization problem. This gives the Lasso huge computational advantages.

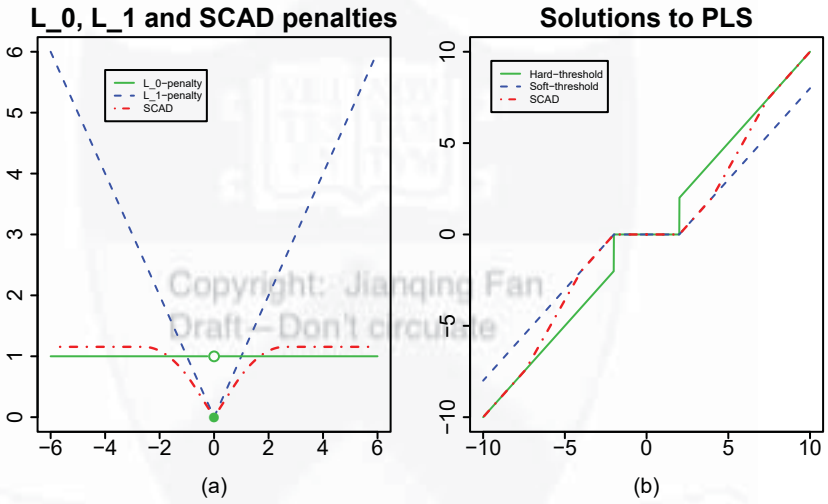

Figure 3.1: (a) The entropy or complexity penalty L_0 (green solid), L_1-penalty (blue dash), and the smoothly clipped L_1-penalty (SCAD, red dotdash). Clearly, SCAD inherits desired properties from L_0 and L_1 penalty at tails and the origin. (b) Solution to the penalized least-squares (3.12) (a=3.7, $\lambda = 2$ for SCAD)

As shown in Figure 3.1, L_1-penalty differs substantially from L_0-penalty. It penalizes the large parameters too much. To further reduce the bias in the estimation, Antoniadis and Fan (2001) and Fan and Li (2001) introduce folded concave penalized least-squares, in which $p_\lambda(\theta)$ is symmetric and concave on each side. In particular, the *smoothly clipped absolute deviation* (SCAD) penalty [see (3.14) below] is introduced to improve the bias property. As shown in Figure 3.1, SCAD behaves like the L_1-penalty at the origin in order to keep the variable selection property and acts like the L_0-penalty at the tails in order to improve the bias property of the L_1-penalty. The smoothness of the penalty function is introduced to ensure the continuity of the solution for *model stability*.

Let us examine what kind of penalty functions are desirable for variable selection. Significant insights can be gained by studying a specific case in which the design matrix is orthonormal.

3.2.1 Orthonormal designs

For an orthonormal design in which the design matrix multiplied by $n^{-1/2}$ is orthonormal (i.e., $\mathbf{X}^T\mathbf{X} = nI_p$, which implies $p \le n$), (3.9) reduces to

$$\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2 + \frac{1}{2}\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2 + \|p_\lambda(|\boldsymbol{\beta}|)\|_1, \tag{3.11}$$

where $\widehat{\boldsymbol{\beta}} = n^{-1}\mathbf{X}^T\mathbf{Y}$ is the ordinary least-squares estimate. Noticing that the first term is constant, minimizing (3.11) becomes minimizing

$$\sum_{j=1}^{p}\left\{\frac{1}{2}(\widehat{\beta}_j - \beta_j)^2 + p_\lambda(|\beta_j|)\right\},$$

which is a componentwise regression problem: each component consists of the univariate PLS problem of the form

$$\widehat{\theta}(z) = \arg\min_\theta \left\{\frac{1}{2}(z-\theta)^2 + p_\lambda(|\theta|)\right\}. \tag{3.12}$$

Fan and Li (2001) advocate penalty functions that give estimators with the following three properties:

1) *Sparsity*: The resulting estimator automatically sets small estimated coefficients to zero to accomplish variable selection and reduce model complexity.
2) *Unbiasedness*: The resulting estimator is nearly unbiased, especially when the true coefficient β_j is large, to reduce model bias.
3) *Continuity*: The resulting estimator is continuous in the data to reduce instability in model prediction (Breiman, 1996).

The third property is nice to have, but not necessarily required.

Let $p_\lambda(t)$ be nondecreasing and continuously differentiable on $[0, \infty)$. Assume that the function $-t - p'_\lambda(t)$ is strictly unimodal on $(0, \infty)$ with the convention $p'_\lambda(0) = p'_\lambda(0+)$. Antoniadis and Fan (2001) characterize the properties of $\widehat{\theta}(z)$ as follows:

(1) Sparsity if $\min_{t\ge 0}\{t + p'_\lambda(t)\} > 0$, which holds if $p'(0+) > 0$;

(2) Approximate unbiasedness if $p'_\lambda(t) = 0$ for large t;

(3) Continuity if and only if $\arg\min_{t\ge 0}\{t + p'_\lambda(t)\} = 0$.

Note that Properties 1) and 2) require the penalty functions to be folded-concave. Fan and Li (2001) advocate to use a family of folded-concave penalized likelihoods as a viable variable selection technique. They do not expect

the form of the penalty functions to play a particularly important role provided that it satisfies Properties 1) – 3). Antoniadis and Fan (2001) show further that $\widehat{\theta}(z)$ is an anti-symmetric shrinkage function:

$$|\widehat{\theta}(z)| \leq |z|, \quad \text{and} \quad \widehat{\theta}(-z) = -\widehat{\theta}(z). \tag{3.13}$$

The approximate unbiasedness requires $\theta(z)/z \to 1$, as $|z| \to \infty$.

3.2.2 Penalty functions

From the above discussion, singularity at the origin (i.e., $p'_\lambda(0+) > 0$) is sufficient for generating sparsity in variable selection and the concavity is needed to reduce the estimation bias. This leads to a family of folded concave penalty functions with singularity at the origin. The L_1 penalty can be regarded as both a concave and convex function. It falls on the boundary of the family of the folded-concave *penalty functions*.

The L_q penalty with $q > 1$ is convex. It does not satisfy the sparsity condition, whereas L_1 penalty does not satisfy the unbiasedness condition. The L_q penalty with $0 \leq q < 1$ is concave but does not satisfy the continuity condition. In other words, none of the L_q penalties possesses all three aforementioned properties simultaneously. For this reason, Fan (1997) introduces the smoothly clipped absolute deviation (SCAD), whose derivative is given by

$$p'_\lambda(t) = \lambda \left\{ I\left(t \leq \lambda\right) + \frac{(a\lambda - t)_+}{(a-1)\,\lambda} I\left(t > \lambda\right) \right\} \quad \text{for some } a > 2, \tag{3.14}$$

where $p_\lambda(0) = 0$ and often $a = 3.7$ is used (suggested by a Bayesian argument in Fan and Li, 2001). Now this satisfies the aforementioned three properties. Note that when $a = \infty$, SCAD reduces to the L_1-penalty.

In response to Fan (1997), Antoniadis (1997) proposes the penalty function

$$p_\lambda(t) = \frac{1}{2}\lambda^2 - \frac{1}{2}(\lambda - t)_+^2, \tag{3.15}$$

which results in the hard-thresholding estimator

$$\widehat{\theta}_H(z) = zI(|z| > \lambda). \tag{3.16}$$

Fan and Li (2001) refer to this penalty function as the hard thresholding penalty, whose derivative function is $p'_\lambda(t)/2 = (\lambda - t)_+$. An extension of this penalty function, derived by Zhang (2010) from a minimax point of view, is the *minimax concave penalty (MCP)*, whose derivative is given by

$$p'_\lambda(t) = (\lambda - t/a)_+ . \tag{3.17}$$

Note that the hard thresholding penalty corresponds to $a = 1$ and the MCP

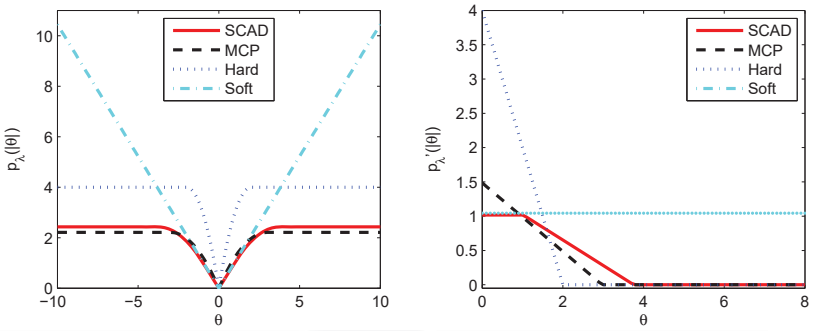

Figure 3.2: Some commonly used penalty functions (left panel) and their derivatives (right panel). They correspond to the risk functions shown in the right panel of Figure 3.3. More precisely, $\lambda = 2$ for hard thresholding penalty, $\lambda = 1.04$ for L_1-penalty, $\lambda = 1.02$ for SCAD with $a = 3.7$, and $\lambda = 1.49$ for MCP with $a = 2$. Taken from Fan and Lv (2010).

does not satisfy the continuity property. But this is not that important as noted before. Figure 3.2 depicts some of those commonly used penalty functions.

3.2.3 Thresholding by SCAD and MCP

We now look at the PLS estimator $\widehat{\theta}(z)$ in (3.12) for some penalties. The entropy penalty (L_0 penalty) and the hard thresholding penalty (3.15) yield the hard thresholding rule (3.16) (Donoho and Johnstone, 1994) and the L_1 penalty gives the soft thresholding rule (Bickel, 1983; Donoho and Johnstone, 1994):

$$\widehat{\theta}_{\text{soft}}(z) = \text{sgn}(z)(|z| - \lambda)_+. \tag{3.18}$$

The SCAD and MCP give rise to analytical solutions to (3.12), each of which is a linear spline in z. For the SCAD penalty, the solution is

$$\widehat{\theta}_{\text{SCAD}}(z) = \begin{cases} \text{sgn}(z)(|z| - \lambda)_+, & \text{when } |z| \le 2\lambda; \\ \text{sgn}(z)[(a-1)|z| - a\lambda]/(a-2), & \text{when } 2\lambda < |z| \le a\lambda; \\ z, & \text{when } |z| \ge a\lambda. \end{cases} \tag{3.19}$$

See Fan (1997) and Figure 3.1(b). Note that when $a = \infty$, the SCAD estimator becomes the *soft-thresholding* estimator (3.18).

For the MCP with $a \ge 1$, the solution is

$$\widehat{\theta}_{\text{MCP}}(z) = \begin{cases} \text{sgn}(z)(|z| - \lambda)_+/(1 - 1/a), & \text{when } |z| < a\lambda; \\ z, & \text{when } |z| \ge a\lambda. \end{cases} \tag{3.20}$$

It has discontinuity points at $|z| = \lambda$, which can create model instability. In

particular, when $a = 1$, the solution is the hard thresholding function $\widehat{\theta}_H(z)$ (3.16). When $a = \infty$, it also becomes a soft-thresholding estimator.

In summary, SCAD and MCP are folded concave functions. They are generalizations of the soft-thresholding and hard-thresholding estimators. The former is continuous whereas the latter is discontinuous.

3.2.4 Risk properties

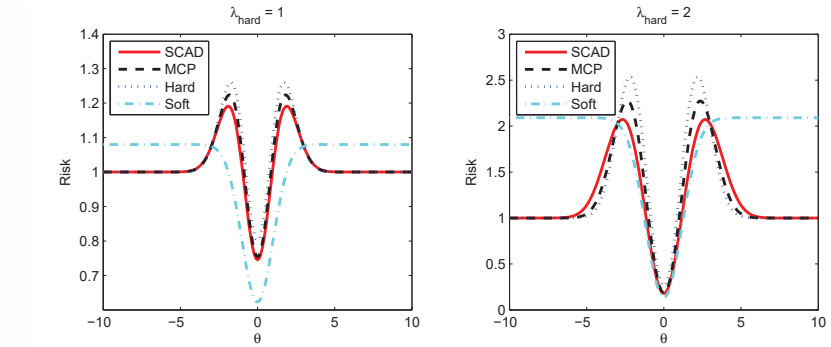

Figure 3.3: The risk functions for penalized least squares under the Gaussian model with the hard-thresholding penalty, L_1-penalty, SCAD ($a = 3.7$), and MCP ($a = 2$). The left panel corresponds to $\lambda = 1$ and the right panel corresponds to $\lambda = 2$ for the hard-thresholding estimator, and the rest of parameters are chosen so that their risks are the same at the point $\theta = 3$. Adapted from Fan and Lv (2010).

We now numerically compare the risk property of several commonly thresholded-shrinkage estimators under the fundamental model $Z \sim N(\theta, 1)$. Let

$$R(\theta) = \mathrm{E}(\widehat{\theta}(Z) - \theta)^2$$

be the risk function for the estimator $\widehat{\theta}(Z)$. Figure 3.3 depicts $R(\theta)$ for some commonly used penalty functions. To make them comparable, we chose $\lambda = 1$ and 2 for the hard thresholding penalty, and for other penalty functions the values of λ are selected to make their risks at $\theta = 3$ the same as that of the hard thresholding estimator $\widehat{\theta}_H(z)$.

Figure 3.3 shows that the PLS estimators improve the ordinary least squares estimator Z in the region where θ is near zero, and have the same risk as the ordinary least squares estimator when θ is far away from zero (e.g., 4 standard deviations away). An exception to this is the Lasso estimator. The Lasso estimator has a bias approximately of size λ for large θ, and this causes higher risk as shown in Figure 3.3. The better risk property at the origin is the payoff that we earn for exploring sparsity.

When $\lambda_{\text{hard}} = 2$, Lasso has higher risk than the SCAD estimator except in a small region. Lasso prefers smaller λ due to its bias. For $\lambda_{\text{hard}} = 1$, Lasso outperforms other methods near the origin. As a result, when λ is chosen automatically by data, Lasso has to choose a smaller λ in order to have a desired mean squared error (to reduce the modeling bias). Yet, a smaller value of λ yields a more complex model. This explains why Lasso tends to have many false positive variables in selected models.

3.2.5 Characterization of folded-concave PLS

Folded-concave penalized least-squares (3.9) is in general a non-convex function. It is challenging to characterize the global solution so let us first characterize its local minimizers.

From Lv and Fan (2009) and Zhang (2010), the local concavity of the penalty $p_\lambda(\cdot)$ at $\mathbf{v} = (v_1, \cdots, v_q)^T$ is defined as

$$\kappa(p_\lambda; \mathbf{v}) = \lim_{\epsilon \to 0+} \max_{1 \le j \le q} \sup_{t_1 < t_2 \in (|v_j|-\epsilon, |v_j|+\epsilon)} -\frac{p'_\lambda(t_2) - p'_\lambda(t_1)}{t_2 - t_1}. \tag{3.21}$$

By the concavity of p_λ on $[0, \infty)$, $\kappa(p_\lambda; \mathbf{v}) \ge 0$. It is easy to see by the mean-value theorem that $\kappa(p_\lambda; \mathbf{v}) = \max_{1 \le j \le q} -p''_\lambda(|v_j|)$ when the second derivative of $p_\lambda(\cdot)$ is continuous. For the L_1 penalty, $\kappa(p_\lambda; \mathbf{v}) = 0$ for any $\mathbf{v}$. For the SCAD penalty, $\kappa(p_\lambda; \mathbf{v}) = 0$ unless some component of $|\mathbf{v}|$ takes values in $[\lambda, a\lambda]$. In the latter case, $\kappa(p_\lambda; \mathbf{v}) = (a-1)^{-1}\lambda^{-1}$.

Let $\lambda_{\min}(\mathbf{A})$ be the minimum eigenvalue of a symmetric matrix $\mathbf{A}$ and $\|\mathbf{a}\|_\infty = \max_j |a_j|$. Lv and Fan (2009) prove the following result. The gap between the necessary condition for local minimizer and sufficient condition for strict local minimizer is tiny (non-strict versus strict inequalities).

Theorem 3.1 (Characterization of PLSE) *Assume that $p_\lambda(|\theta|)$ is folded concave. Then a necessary condition for $\widehat{\boldsymbol{\beta}} \in R^p$ being a local minimizer of $Q(\boldsymbol{\beta})$ defined by (3.9) is*

$$n^{-1}\mathbf{X}_1^T(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) - p'_\lambda(|\widehat{\boldsymbol{\beta}}_1|)\,\text{sgn}(\widehat{\boldsymbol{\beta}}_1) = \mathbf{0}, \tag{3.22}$$

$$\|n^{-1}\mathbf{X}_2^T(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})\|_\infty \le p'_\lambda(0+), \tag{3.23}$$

$$\lambda_{\min}(n^{-1}\mathbf{X}_1^T\mathbf{X}_1) \ge \kappa(p_\lambda; \widehat{\boldsymbol{\beta}}_1), \tag{3.24}$$

where $\mathbf{X}_1$ and $\mathbf{X}_2$ are respectively the submatrices of $\mathbf{X}$ formed by columns indexed by $\text{supp}(\widehat{\boldsymbol{\beta}})$ and its complement, and $\widehat{\boldsymbol{\beta}}_1$ is a vector of all non-vanishing components $\widehat{\boldsymbol{\beta}}$. On the other hand, if (3.22) – (3.24) hold with inequalities replaced by strict inequalities, then $\widehat{\boldsymbol{\beta}}$ is a strict local minimizer of $Q(\boldsymbol{\beta})$.

Conditions (3.22) – (3.24) can be regarded as the *Karush-Kuhn-Tucker conditions*. They can also be derived by using subgradient calculus. Conditions (3.22) and (3.24) are respectively the first and second order conditions

for $(\widehat{\boldsymbol{\beta}}_1, \mathbf{0})$ to be the local minimizer of $Q(\boldsymbol{\beta}_1, \mathbf{0})$, the local minimizer on the restricted coordinate subspace. Condition (3.23) guarantees the local minimizer on the restricted coordinate subspace is also the local minimizer of the whole space R^p.

When $Q(\boldsymbol{\beta})$ is strictly convex, there exists at most one local minimizer. In this case, the local minimizer is also the unique global minimizer. For a folded concave penalty function, let $\kappa(p_\lambda)$ be the maximum concavity of the penalty function p_λ defined by

$$\kappa(p_\lambda) = \sup_{t_1 < t_2 \in (0,\infty)} -\frac{p'_\lambda(t_2) - p'_\lambda(t_1)}{t_2 - t_1}. \tag{3.25}$$

For the L_1 penalty, SCAD and MCP, we have $\kappa(p_\lambda) = 0$, $(a-1)^{-1}$, and a^{-1}, respectively. Thus, the maximum concavity of SCAD and MCP is small when a is large. When

$$\lambda_{\min}(n^{-1}\mathbf{X}^T\mathbf{X}) > \kappa(p_\lambda), \tag{3.26}$$

the function $Q(\boldsymbol{\beta})$ is strictly convex, as the convexity of the quadratic loss dominates the maximum concavity of the penalty in (3.9). Hence, the global minimum is unique. Note that condition (3.26) requires $p \le n$.

In general, the global minimizer of the folded-concave penalized least-squares is hard to characterize. Fan and Lv (2011) are able to give conditions under which a solution is global optimal on the union of all m-dimensional coordinate subspaces:

$$\mathbb{S}_m = \{\boldsymbol{\beta} \in R^p : \|\boldsymbol{\beta}\|_0 \le m\}. \tag{3.27}$$

3.3 Lasso and L_1 Regularization

Lasso gains its popularity due to its convexity and computational expedience. The predecessor of Lasso is the negative garrote. The study of Lasso also leads to the Dantzig selector, the *adaptive Lasso* and the *elastic net*. This section touches on the basis of these estimators in which the L_1-norm regularization plays a central role.

3.3.1 Nonnegative garrote

The nonnegative garrote estimator, introduced by Breiman (1995), is the first modern statistical method that uses the L_1-norm regularization to do variable selection in multiple linear regression. Consider the usual setting with $p < n$ and let $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_1, \cdots, \widehat{\beta}_p)^T$ be the OLS estimator. When $p > n$, $\widehat{\boldsymbol{\beta}}$ can be the ridge regression estimator (Yuan and Lin, 2005), the main idea stays the same. Then, the fitted model becomes

$$\widehat{Y} = \widehat{\beta}_1 X_1 + \cdots + \widehat{\beta}_p X_p$$

The above model uses all variables. To do variable selection, we introduce the nonnegative shrinkage parameter $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_p)^T$ and regard $\widehat{\boldsymbol{\beta}}$ as fixed, a new fitted model becomes

$$\widehat{Y} = \theta_1 Z_1 + \cdots + \theta_p Z_p, \qquad Z_j = \widehat{\beta}_j X_j.$$

If θ_j is zero, then variable X_j is excluded from the fitted model. The *nonnegative garrote* estimates $\boldsymbol{\theta}$ via the following L_1 regularized least squares:

$$\min_{\boldsymbol{\theta} \geq 0} \frac{1}{2n} \|\mathbf{Y} - \mathbf{Z}\boldsymbol{\theta}\|^2 + \lambda \sum_j \theta_j. \tag{3.28}$$

Note that under the nonnegative constraints $\sum_{j=1}^p \theta_j = \|\boldsymbol{\theta}\|_1$.

By varying λ, the nonnegative garrote automatically achieves model selection. Many components of the minimizer of (3.28), $\widehat{\boldsymbol{\theta}}$, will be zero. This can be easily seen when $\mathbf{X}$ is scaled orthonormal $\mathbf{X}^T\mathbf{X} = n\mathbf{I}_p$, as in Section 3.2.1. In this case, $\{\mathbf{Z}_j\}$ are still orthogonal and $\mathbf{Z}^T\mathbf{Z}$ is diagonal. The ordinary least-squares estimator is given by

$$\widehat{\boldsymbol{\theta}}_0 = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{Y} = (\mathbf{Z}_1/\|\mathbf{Z}_1\|^2, \cdots, \mathbf{Z}_p/\|\mathbf{Z}_p\|^2)^T\mathbf{Y}.$$

Note that by the orthogonality of the least-squares fit to its residuals,

$$\begin{aligned} \|\mathbf{Y} - \mathbf{Z}\boldsymbol{\theta}\|^2 &= \|\mathbf{Y} - \mathbf{Z}\widehat{\boldsymbol{\theta}}_0\|^2 + \|\mathbf{Z}(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_0)\|^2 \\ &= \|\mathbf{Y} - \mathbf{Z}\widehat{\boldsymbol{\theta}}_0\|^2 + \sum_{j=1}^p \|\mathbf{Z}_j\|^2(\theta_j - \widehat{\theta}_{j0})^2, \end{aligned}$$

where $\widehat{\theta}_{j0} = \mathbf{Z}_j^T\mathbf{Y}/\|\mathbf{Z}_j\|^2$ is the j^{th} component of $\boldsymbol{\theta}_0$ and the last equality utilizes the orthogonality of $\mathbf{Z}_j$. Therefore, problem (3.28) becomes

$$\min_{\boldsymbol{\theta} \geq 0} \frac{1}{2} \sum_{j=1}^p \|\mathbf{Z}_j\|^2(\theta_j - \widehat{\theta}_{j0})^2 + \lambda \sum_{j=1}^p \theta_j.$$

This reduces to the componentwise minimization problem

$$\min_{\theta \geq 0} \frac{1}{2}(\theta - \theta_0)^2 + \lambda\theta,$$

whose minimizer is clearly $\widehat{\theta} = (\theta_0 - \lambda)_+$ by taking the first derivative and setting it to zero. Applying this to our scenario and noticing $\|\mathbf{Z}_j\|^2 = n\widehat{\beta}_j^2$ and $\widehat{\theta}_{j0} = n^{-1}\mathbf{X}_j^T\mathbf{Y}/\widehat{\beta}_j$ we have

$$\widehat{\theta}_j = \Big(\widehat{\theta}_{j0} - \frac{\lambda}{\|\mathbf{Z}_j\|^2}\Big)_+ = \Big(\frac{\mathbf{X}_j^T\mathbf{Y}}{n\widehat{\beta}_j} - \frac{\lambda}{n\widehat{\beta}_j^2}\Big)_+.$$

In particular, if $\widehat{\boldsymbol{\beta}} = n^{-1}\mathbf{X}^T\mathbf{Y}$ is the ordinary least-squares estimate, then

$$\widehat{\theta}_j = \Big(1 - \frac{\lambda}{n\widehat{\beta}_j^2}\Big)_+.$$

Model selection of the negative garrote now becomes clear. When $|\widehat{\beta}_j| \le \sqrt{\lambda/n}$, it is shrunk to zero. The larger the original estimate, the smaller the shrinkage. Furthermore, the shrinkage rule is continuous. This is in the same spirit as the folded concave PLS such as SCAD introduced in the last section.

3.3.2 Lasso

Lasso, the term coined by Tibshirani (1996), estimates the sparse regression coefficient vector $\boldsymbol{\beta}$ by minimizing

$$\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{3.29}$$

This corresponds to (3.9) by taking $p_\lambda(\theta) = \lambda\theta$ for $\theta \ge 0$. Comparing (3.29) and (3.28), we see that the Lasso does not need to use a preliminary estimator of $\boldsymbol{\beta}$, although both use the L_1-norm to achieve variable selection.

The KKT conditions (3.22)–(3.24) now become

$$n^{-1}\mathbf{X}_1^T(\mathbf{Y} - \mathbf{X}_1\widehat{\boldsymbol{\beta}}_1) - \lambda\,\mathrm{sgn}(\widehat{\boldsymbol{\beta}}_1) = \mathbf{0}, \tag{3.30}$$

and

$$\|(n\lambda)^{-1}\mathbf{X}_2^T(\mathbf{Y} - \mathbf{X}_1\widehat{\boldsymbol{\beta}}_1)\|_\infty \le 1, \tag{3.31}$$

since (3.24) is satisfied automatically. This first condition says that the signs of nonzero components of Lasso are the same as the correlations of the covariates with the current residual. The equations (3.30) and (3.31) imply that

$$\|n^{-1}\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})\|_\infty \le \lambda. \tag{3.32}$$

Note that condition (3.31) holds for $\widehat{\boldsymbol{\beta}} = 0$ when

$$\lambda > \|n^{-1}\mathbf{X}^T\mathbf{Y}\|_\infty. \tag{3.33}$$

Since the condition is imposed with a strict inequality, it is a sufficient condition (Theorem 3.1). In other words, when $\lambda > \|n^{-1}\mathbf{X}^T\mathbf{Y}\|_\infty$, $\widehat{\boldsymbol{\beta}} = \mathbf{0}$ is the unique solution and hence Lasso selects no variables. Therefore, we need only to consider λ in the interval $[0, \|n^{-1}\mathbf{X}^T\mathbf{Y}\|_\infty]$.

We now look at the model selection consistency of Lasso. Assuming the invertibility of $\mathbf{X}_1^T\mathbf{X}_1$, solving equation (3.30) gives

$$\widehat{\boldsymbol{\beta}}_1 = (\mathbf{X}_1^T\mathbf{X}_1)^{-1}(\mathbf{X}_1^T\mathbf{Y} - n\lambda\,\mathrm{sgn}(\widehat{\boldsymbol{\beta}}_1)), \tag{3.34}$$

and substituting this into equation (3.31) yields

$$\|(n\lambda)^{-1}\mathbf{X}_2^T(\mathbf{I}_n - \mathbf{P}_{\mathbf{X}_1})\mathbf{Y} + \mathbf{X}_2^T\mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\,\mathrm{sgn}(\widehat{\boldsymbol{\beta}}_1)\|_\infty \le 1, \tag{3.35}$$

where $\mathbf{P}_{\mathbf{X}_1} = \mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T$ is the projection matrix onto the linear space spanned by the columns of $\mathbf{X}_1$. For the true parameter, let $\text{supp}(\boldsymbol{\beta}_0) = \mathcal{S}_0$ so that

$$\mathbf{Y} = \mathbf{X}_{\mathcal{S}_0}\boldsymbol{\beta}_0 + \boldsymbol{\varepsilon}. \tag{3.36}$$

If $\text{supp}(\widehat{\boldsymbol{\beta}}) = \mathcal{S}_0$, i.e., *model selection consistency* holds, then $\mathbf{X}_{\mathcal{S}_0} = \mathbf{X}_1$ and $(\mathbf{I}_n - \mathbf{P}_{\mathbf{X}_1})\mathbf{X}_{\mathcal{S}_0} = 0$. By substituting (3.36) into (3.35), we have

$$\|(n\lambda)^{-1}\mathbf{X}_2^T(\mathbf{I}_n - \mathbf{P}_{\mathbf{X}_1})\boldsymbol{\varepsilon} + \mathbf{X}_2^T\mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\,\text{sgn}(\widehat{\boldsymbol{\beta}}_1)\|_\infty \le 1. \tag{3.37}$$

Note that this condition is also sufficient if the inequality is replaced with the strict one (see Theorem 3.1).

Typically, the first term in (3.37) is negligible. This will be formally shown in Chapter 4. A specific example is the case $\boldsymbol{\varepsilon} = 0$ as in the compressed sensing problem. In this case, condition (3.37) becomes

$$\|\mathbf{X}_2^T\mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\,\text{sgn}(\widehat{\boldsymbol{\beta}}_1)\|_\infty \le 1.$$

This condition involves $\text{sgn}(\widehat{\boldsymbol{\beta}}_1)$. If we require a stronger consistency $\text{sgn}(\widehat{\boldsymbol{\beta}}) = \text{sgn}(\boldsymbol{\beta}_0)$, called the *sign consistency*, then the condition becomes

$$\|\mathbf{X}_2^T\mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\,\text{sgn}(\boldsymbol{\beta}_{\mathcal{S}_0})\|_\infty \le 1. \tag{3.38}$$

The above condition does not depend on λ. It appeared in Zou (2006) and Zhao and Yu (2006) who coined the name *the irrepresentable condition*.

Note that $(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T\mathbf{X}_2$ in (3.38) is the matrix of the regression coefficients of each 'unimportant' variable X_j ($j \notin \mathcal{S}_0$) regressed on the important variables $\mathbf{X}_1 = \mathbf{X}_{\mathcal{S}_0}$. The irrepresentable condition is a condition on how strongly the important and unimportant variables can be correlated. Condition (3.38) states that the sum of the signed regression coefficients of each unimportant variable X_j for $j \notin \mathcal{S}_0$ on the important variables $\mathbf{X}_{\mathcal{S}_0}$ cannot exceed 1. The more the unimportant variables, the harder the condition is to meet. The irrepresentable condition is in general very restrictive. Using the regression intuition, one can easily construct an example when it fails. For example, if an unimportant variable is generated by

$$X_j = \rho s^{-1/2}\sum_{k\in\mathcal{S}_0}\text{sgn}(\beta_k)X_k + \sqrt{1-\rho^2}\varepsilon_k, \qquad s = |\mathcal{S}_0|,$$

for some given $|\rho| \le 1$ (all normalization is to make $\text{Var}(X_j) = 1$), where all other random variables are independent and standardized, then the L_1-norm of the signed regression coefficients of this variable is $|\rho|s^{1/2}$, which can easily exceed 1. The larger the 'important variable' set $\mathcal{S}_0$, the easier the irrepresentable condition fails. In addition, we need only one such unimportant predictor that has such a non-negligible correlation with important variables to make the condition fail. See also Corollary 1 in Zou (2006) for a counterexample.

The moral of the above story is that Lasso can have sign consistency, but this happens only in very specific cases. The irrepresentable condition (3.38) is independent of λ. When it fails, Lasso does not have sign consistency and this cannot be rescued by using a different value of λ.

We now look at the risk property of Lasso. It is easier to explain it under the constrained form:

$$\min_{\|\boldsymbol{\beta}\|_1 \le c} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 \tag{3.39}$$

for some constant c, as in Tibshirani (1996). Define the *theoretical risk* and *empirical risk* respectively as

$$R(\boldsymbol{\beta}) = \mathrm{E}(Y - \mathbf{X}^T\boldsymbol{\beta})^2 \quad \text{and} \quad R_n(\boldsymbol{\beta}) = n^{-1}\sum_{i=1}^n (Y_i - \mathbf{X}_i^T\boldsymbol{\beta})^2,$$

which are prediction errors using the parameter $\boldsymbol{\beta}$. The best prediction error is $R(\boldsymbol{\beta}_0)$. Note that

$$R(\boldsymbol{\beta}) = \boldsymbol{\gamma}^T\boldsymbol{\Sigma}^*\boldsymbol{\gamma} \quad \text{and} \quad R_n(\boldsymbol{\beta}) = \boldsymbol{\gamma}^T\mathbf{S}_n^*\boldsymbol{\gamma},$$

where $\boldsymbol{\gamma} = (-1, \boldsymbol{\beta}^T)^T$, $\boldsymbol{\Sigma}^* = \mathrm{Var}((Y, \mathbf{X}^T)^T)$, and $\mathbf{S}_n^*$ is the sample covariance matrix based on the data $\{(Y_i, \mathbf{X}_i^T)^T\}_{i=1}^n$. Thus, for any $\boldsymbol{\beta}$, we have the following risk approximation:

$$\begin{aligned} |R(\boldsymbol{\beta}) - R_n(\boldsymbol{\beta})| &= |\boldsymbol{\gamma}^T(\boldsymbol{\Sigma}^* - \mathbf{S}_n^*)\boldsymbol{\gamma}| \\ &\le \|\boldsymbol{\Sigma}^* - \mathbf{S}_n^*\|_\infty \|\boldsymbol{\gamma}\|_1^2 \\ &= (1 + \|\boldsymbol{\beta}\|_1)^2 \|\boldsymbol{\Sigma}^* - \mathbf{S}_n^*\|_\infty. \end{aligned} \tag{3.40}$$

On the other hand, if the true parameter $\boldsymbol{\beta}_0$ is in the feasible set, namely, $\|\boldsymbol{\beta}_0\|_1 \le c$, then $R_n(\widehat{\boldsymbol{\beta}}) - R_n(\boldsymbol{\beta}_0) \le 0$. Using this,

$$0 \le R(\widehat{\boldsymbol{\beta}}) - R(\boldsymbol{\beta}_0) \le \{R(\widehat{\boldsymbol{\beta}}) - R_n(\widehat{\boldsymbol{\beta}})\} + \{R_n(\boldsymbol{\beta}_0) - R(\boldsymbol{\beta}_0)\}.$$

By (3.40) along with $\|\widehat{\boldsymbol{\beta}}\|_1 \le c$ and $\|\boldsymbol{\beta}_0\|_1 \le c$, we conclude that

$$|R(\widehat{\boldsymbol{\beta}}) - R(\boldsymbol{\beta}_0)| \le 2(1 + c)^2 \|\boldsymbol{\Sigma}^* - \mathbf{S}_n^*\|_\infty. \tag{3.41}$$

When $\|\boldsymbol{\Sigma}^* - \mathbf{S}_n^*\|_\infty \to 0$, the risk converges. Such a property is called *persistency* by Greenshtein and Ritov (2004). Further details on the rates of convergence for estimating large covariance matrices can be found in Chapter 11. The rate is of order $O(\sqrt{(\log p)/n})$ for the data with Gaussian tails. The above discussion also reveals the relationship between covariance matrix estimation and sparse regression. A robust covariance matrix estimation can also reveal a robust sparse regression.

Persistency requires that the risk based on $\widehat{\boldsymbol{\beta}}$ is approximately the same as that of the optimal parameter $\boldsymbol{\beta}_0$, i.e.,

$$R(\widehat{\boldsymbol{\beta}}) - R(\boldsymbol{\beta}_0) = o_P(1).$$

By (3.41), this requires only $\boldsymbol{\beta}_0$ sparse in the sense that $\|\boldsymbol{\beta}_0\|_1$ does not grow too quickly (recalling $\|\boldsymbol{\beta}_0\|_1 \le c$) and the large covariance matrix $\boldsymbol{\Sigma}^*$ can be uniformly consistently estimated. For data with Gaussian tails, since $\|\boldsymbol{\Sigma}^* - \mathbf{S}_n^*\|_\infty = O_P(\sqrt{(\log p)/n})$ (see Chapter 11), we require

$$\|\boldsymbol{\beta}_0\|_1 \le c = o((n/\log p)^{1/4})$$

for Lasso to possess persistency. Furthermore, the result (3.41) does not require to have a true underlying linear model. As long as we define

$$\boldsymbol{\beta}_0 = \text{argmin}_{\|\boldsymbol{\beta}\|_1 \le c} R(\boldsymbol{\beta}),$$

the risk approximation inequality (3.41) holds by using the same argument above. In conclusion, Lasso has a good risk property when $\boldsymbol{\beta}_0$ is sufficiently sparse.

3.3.3 Adaptive Lasso

The irrepresentable condition indicates restrictions on the use of the Lasso as a model/variable selection method. Another drawback of the Lasso is its lack of unbiasedness for large coefficients, as explained in Fan and Li (2001). This can be seen from (3.34). Even when the signal is strong so that $\text{supp}(\widehat{\boldsymbol{\beta}}) = \mathcal{S}_0$, by substituting (3.36) into (3.34), we have

$$\widehat{\boldsymbol{\beta}}_1 = \boldsymbol{\beta}_0 + (\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T\boldsymbol{\varepsilon} - n\lambda(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\,\text{sgn}(\widehat{\boldsymbol{\beta}}_1).$$

The last term is the bias due to the L_1 penalty. Unless λ goes to 0 sufficiently fast, the bias term is not negligible. However, $\lambda \approx \frac{1}{\sqrt{n}}$ is needed in order to make the Lasso estimate root-n consistent under the fixed p large n setting. For $p \gg n$, the Lasso estimator uses $\lambda \approx \sqrt{\log(p)/n}$ to achieve the optimal rate $\sqrt{|\mathcal{S}_0|\log(p)/n}$. See Chapter 4 for more details. So now, it is clear that the optimal Lasso estimator has non-negligible biases.

Is there a nice fix to these two problems? Zou (2006) proposes to use the adaptively weighted L_1 penalty (a.k.a. *adaptive Lasso*) to replace the L_1 penalty in penalized linear regression and penalized generalized linear models. With the weighted L_1 penalty, (3.9) becomes

$$\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\sum_{j=1}^{p} w_j|\beta_j|. \tag{3.42}$$

To keep the convexity property of the Lasso, w_j should be nonnegative. It is important to note that if the weights are deterministic, then they cannot fix the aforementioned two problems of the Lasso. Suppose that some deterministic weights can make the Lasso gain sign consistency. Then no w_j should be zero, otherwise the variable X_j is always included, which will violate the sign consistency of the Lasso if the underlying model does not include X_j, i.e. X_j

is not an important variable. Hence, all w_js are positive. Then we redefine the regressors as $X_j^{\mathrm{w}} = X_j/w_j$ and $\theta_j = w_j\beta$, $1 \le j \le p$. The underlying regression model can be rewritten as

$$Y = \sum_{j=1}^{p} X_j^{\mathrm{w}}\theta_j + \epsilon$$

and (3.42) becomes

$$\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}^{\mathrm{w}}\boldsymbol{\theta}\|^2 + \lambda\sum_{j=1}^{p}|\theta_j|.$$

Its corresponding irrepresentable condition is

$$\|(\mathbf{X}_2^{\mathrm{w}})^T\mathbf{X}_1^{\mathrm{w}}[(\mathbf{X}_1^{\mathrm{w}})^T\mathbf{X}_1^{\mathrm{w}}]^{-1}\,\mathrm{sgn}(\boldsymbol{\beta}_{\mathcal{S}_0})\|_\infty \le 1.$$

We write $\mathbf{W} = (w_1, \ldots, w_p)^T = (\mathbf{W}_1, \mathbf{W}_2)^T$. and express the irrepresentable conditions using the original variables, we have

$$\|[\mathbf{X}_2^T\mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{W}_1 \circ \mathrm{sgn}(\boldsymbol{\beta}_{\mathcal{S}_0})] \circ \mathbf{W}_2^{-1}\| \le 1$$

Observe that if $\max \mathbf{W}_1/\inf \mathbf{W}_2 \to 0$, then this representable condition can be satisfied for general $\mathbf{X}_1, \mathbf{X}_2$ and $\mathrm{sgn}(\boldsymbol{\beta}_{\mathcal{S}_0})$. This condition can only be achieved using a data-driven scheme, as we do not know the set $\mathcal{S}_0$.

Zou (2006) proposes to use a preliminary estimate $\widehat{\beta}_j$ to construct w_j. For example, $w_j = |\widehat{\beta}_j|^{-\gamma}$ for some $\gamma > 0$, and $\gamma = 0.5$, 1 or 2. In the case of fixed p large n, the preliminary estimate can be the least-squares estimate. When $p \gg n$, the preliminary estimate can be the lasso estimate and $w_j = p'_\lambda(|\widehat{\beta}_j^{\mathrm{lasso}}|)/\lambda$ with a folded concave penalty $p_\lambda(\cdot)$.

As will be seen in Section 3.5.5, the adaptive lasso is connected to the penalized least-squares estimator (3.9) via the *local linear approximation* with $p'_\lambda(\theta) = \lambda\theta^{-\gamma}$ or $L_{1-\gamma}$ penalty. Since the derivative function is decreasing, the spirit of the adaptive Lasso is the same as the folded-concave PLS. Hence, the adaptive Lasso is able to fix the bias caused by the L_1 penalty. In particular, the adaptive lasso estimator for $\boldsymbol{\beta}_{\mathcal{S}_0}$ shares the asymptotical normality property of the oracle OLS estimator for $\boldsymbol{\beta}_{\mathcal{S}_0}$, i.e.. $\widehat{\boldsymbol{\beta}}_{\mathcal{S}_0}^{\mathrm{oracle}} = (\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T\mathbf{Y}$.

3.3.4 Elastic Net

In the early 2000s, the Lasso was applied to regression with micrarrays to do gene selection. The results were concerning because of high variability. This is mainly caused by the spurious correlation in high-dimensional data, as illustrated in Section 1.3.3 of Chapter 1. How to handle the strong (empirical) correlations among high-dimensional variables while keeping the continuous shrinkage and selection property of the Lasso? Zou and Hastie (2005) propose the *elastic net* regularization that uses a convex combination of L_1 and L_2

penalties. For the penalized least squares, the elastic net estimator is defined as

$$\arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda_2\|\boldsymbol{\beta}\|^2 + \lambda_1\|\boldsymbol{\beta}\|_1 \right\}, \tag{3.43}$$

where $p_{\lambda_1,\lambda_2}(t) = \lambda_1|t| + \lambda_2 t^2$ is called the elastic net penalty. Another form of the elastic net penalty is

$$p_{\lambda,\alpha}(t) = \lambda J(t) = \lambda[(1-\alpha)t^2 + \alpha|t|],$$

with $\lambda = \lambda_1 + \lambda_2$ and $\alpha = \frac{\lambda_1}{\lambda_1+\lambda_2}$. The elastic net is a pure ridge regression when $\alpha = 0$ and a pure Lasso when $\alpha = 1$. The advantage of using (λ, α) parametrization is that α has a natural range $[0, 1]$. In practice, we can use CV to choose α over a grid such as $0.1k$, $k = 1, \ldots, 10$. For the penalized least squares problem, using (λ_2, λ_1) parametrization is interesting because it can be shown that for a fixed λ_2 the solution path is piecewise linear with respect to λ_1. Zou and Hastie (2005) exploit this property to derive an efficient path-following algorithm named LARS-EN for computing the entire solution path of the Elastic Net penalized least squares (for each fixed λ_2).

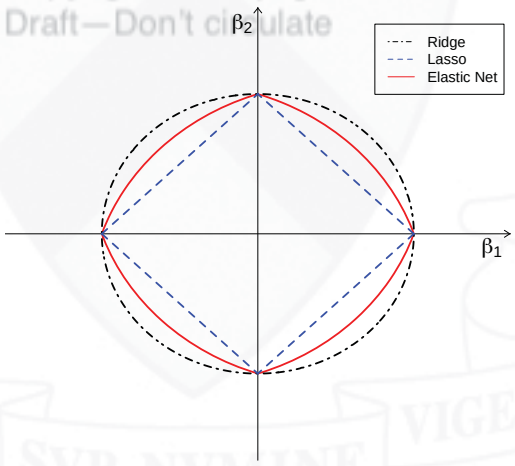

Figure 3.4: Geometric comparison of Lasso, Ridge and Elastic Net in two-dimensions. $\alpha = 0.5$ in the elastic net penalty. The Lasso and Elastic Net share four sharp vertices–sparsity. Similar to Ridge, Elastic Net has 'round' edges–strict convexity.

It is well known that L_2 regularization gracefully handles collinearity and achieves a good bias-variance tradeoff for prediction. The Elastic Net inherits the ability to handle collinearity from its L_2 component and keeps the

sparsity property of the Lasso through its L_1 component. Figure 3.4 shows a geometric comparison of the two penalty functions in two-dimensions. With high-dimensional data the Elastic Net often generates a more accurate predictive model than the Lasso. Zou and Hastie (2005) also reveal the group effect of the Elastic Net in the sense that highly correlated variables tend to enter or exit the model together, while the Lasso tends to randomly pick one variable and ignore the rest.

To help visualize the fundamental differences between Lasso and Elastic Net, let us consider a synthetic model as follows. Let Z_1 and Z_2 be two independent unif$(0, 20)$ variables. Response $\mathbf{Y}$ is generated by $\mathbf{Y} = Z_1 + 0.1 \cdot Z_2 + \epsilon$, with $\epsilon \sim N(0, 1)$ and the observed regressors are generated by

$$X_1 = Z_1 + \epsilon_1, \quad X_2 = -Z_1 + \epsilon_2, \quad X_3 = Z_1 + \epsilon_3,$$
$$X_4 = Z_2 + \epsilon_4, \quad X_5 = -Z_2 + \epsilon_5, \quad X_6 = Z_2 + \epsilon_6,$$

where ϵ_i are iid $N(0, \frac{1}{16})$. X_1, X_2, X_3 form a group whose underlying factor is Z_1, and X_4, X_5, X_6 form the other group whose underlying factor is Z_2. The within group correlations are almost 1 and the between group correlations are almost 0. Ideally, we would want to only identify the Z_1 group (X_1, X_2, X_3) as the important variables. We generated two independent datasets with sample size 100 from this model. Figure 3.5 displays the solution paths of Lasso and Elastic Net; see Section 3.5 for details. The two Lasso solution paths are very different, suggesting the high instability of the Lasso under strong correlations. On the other hand, the two Elastic Net solution paths are almost identical. Moreover, the Elastic Net identifies the corrected variables.

The Elastic Net relies on its L_1 component for sparsity and variable selection. Similar to the Lasso case, the Elastic Net also requires a restrictive condition on the design matrix for selection consistency (Jia and Yu, 2010). To bypass this restriction, Zou and Zhang (2009) follow the adaptive Lasso idea and introduce the adaptive Elastic Net penalty $p(|\beta_j|) = \lambda_1 w_j |\beta_j| + \lambda_2 |\beta_j|^2$ where $w_j = |\widehat{\beta}^{\text{enet}} + 1/n|^{-\gamma}$. The numeric studies therein shows the very competitive performance of the adaptive Elastic Net in terms of variable selection and model estimation.

3.3.5 Dantzig selector

The *Dantzig selector*, introduced by Candés and Tao (2007), is a novel idea of casting the regularization problem into a linear program. Recall that Lasso satisfies (3.32), but it might not have the smallest L_1 norm. One can find the estimator to minimize its L_1-norm:

$$\min_{\boldsymbol{\beta} \in R^p} \|\boldsymbol{\beta}\|_1, \qquad \text{subject to} \quad \|n^{-1}\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\|_\infty \le \lambda. \tag{3.44}$$

The target function and constraints in (3.44) are linear. The problem can be

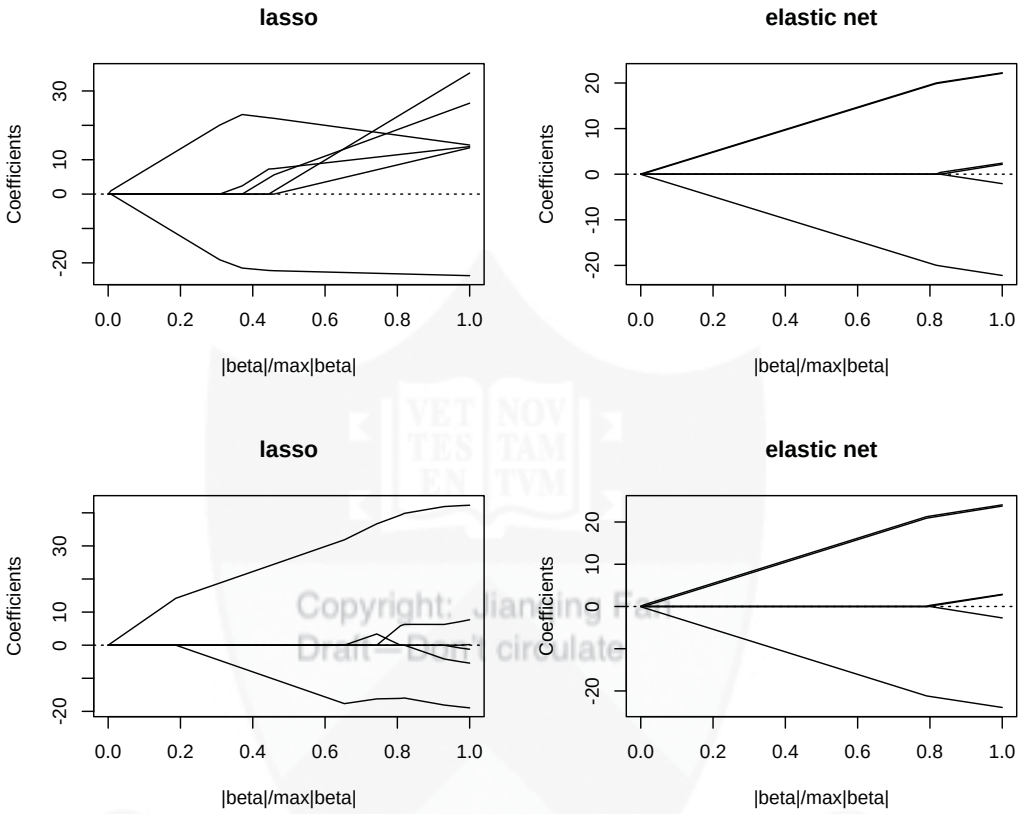

Figure 3.5: A toy example to illustrate the instability of the Lasso and how this is improved by the elastic net. Left two panels show the Lasso solution path on two independent dataset generated from the same model; right two panels are the elastic net solution path on the same two datasets. The Lasso paths are very different, but the Elastic Net paths are almost the same. The x-axis is the fraction of L_1 norm defined as $\|\widehat{\boldsymbol{\beta}}(\lambda)\|_1 / \max_{\lambda \le \lambda_{\max}} \|\widehat{\boldsymbol{\beta}}(\lambda)\|_1$.

formulated as a linear program by expressing it as

$$\min_{\mathbf{u}} \sum_{i=1}^{p} u_i, \qquad \mathbf{u} \ge 0, \quad -\mathbf{u} \le \boldsymbol{\beta} \le \mathbf{u}, \quad -\lambda \mathbf{1} \le n^{-1} \mathbf{X}^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \le \lambda \mathbf{1}.$$

The name "Dantzig selector" was coined by Emmanuel Candès and Terence Tao to pay tribute to George Dantzig, the father of linear programming who passed away while their manuscript was finalized.

Let $\widehat{\boldsymbol{\beta}}_{\mathrm{DZ}}$ be the solution. A necessary condition for $\widehat{\boldsymbol{\beta}}_{\mathrm{DZ}}$ to have model

selection consistency is that $\boldsymbol{\beta}_0$ is in the feasible set of (3.44), with probability tending to one. Using model (3.36), this implies that $\lambda \geq n^{-1}\|\mathbf{X}^T\boldsymbol{\varepsilon}\|_\infty$. For example, in the case when $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 I_n)$ and columns $\{\mathbf{X}_j\}_{j=1}^p$ of $\mathbf{X}$ are standardized so that $n^{-1}\|\mathbf{X}_j\|^2 = 1$, then $\mathbf{X}_j^T\boldsymbol{\varepsilon} \sim N(0, \sigma^2/n)$. Then, it can easily be shown (see Section 3.3.7) that it suffices to take λ as $\sigma\sqrt{2(1+\delta)n^{-1}\log p}$ for any $\delta > 0$, by using the union bound and the tail probability of normal distribution.

The Dantzig selector opens a new chapter for sparse regularization. Since the value λ is chosen so that the true parameter $\boldsymbol{\beta}_0$ falls in the constraint:

$$P\{\|n^{-1}\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}_0)\|_\infty \leq \lambda\} \to 1, \tag{3.45}$$

Fan (2014) interprets the set $\{\boldsymbol{\beta} : \|n^{-1}\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\|_\infty \leq \lambda\}$ as the high confidence set and this high confidence set summarizes the information on the parameter $\boldsymbol{\beta}_0$ provided by the data. He argues that this set is too big to be useful in high-dimensional spaces and that we need some additional prior about $\boldsymbol{\beta}_0$. If the prior is that $\boldsymbol{\beta}_0$ is sparse, one naturally combines these two pieces of information. This leads to finding the *sparsest solution in high-confidence set* as a natural solution to the sparse regulation. This idea applies to *quasi-likelihood* based models and includes the Dantzig selector as a specific case. See Fan (2014) for details. The idea is reproduced by Fan, Han, and Liu (2014).

To see how norm-minimization plays a role, let us assume that $\boldsymbol{\beta}_0$ is in the feasible set by taking a large enough value λ, i.e., $\lambda \geq n^{-1}\|\mathbf{X}^T\boldsymbol{\varepsilon}\|_\infty$ as noted above. This is usually achieved by a probabilistic statement. Let $\widehat{\boldsymbol{\Delta}} = \widehat{\boldsymbol{\beta}}_{\mathrm{DZ}} - \boldsymbol{\beta}_0$. From the norm minimization, we have

$$\|\boldsymbol{\beta}_0\|_1 \geq \|\widehat{\boldsymbol{\beta}}_{\mathrm{DZ}}\|_1 = \|\boldsymbol{\beta}_0 + \widehat{\boldsymbol{\Delta}}\|_1. \tag{3.46}$$

Noticing $\mathcal{S}_0 = \mathrm{supp}(\boldsymbol{\beta}_0)$, we have

$$\begin{aligned} \|\boldsymbol{\beta}_0 + \widehat{\boldsymbol{\Delta}}\|_1 &= \|(\boldsymbol{\beta}_0 + \widehat{\boldsymbol{\Delta}})_{\mathcal{S}_0}\|_1 + \|(\mathbf{0} + \widehat{\boldsymbol{\Delta}})_{\mathcal{S}_0^c}\|_1 \\ &\geq \|\boldsymbol{\beta}_0\|_1 - \|\widehat{\boldsymbol{\Delta}}_{\mathcal{S}_0}\|_1 + \|\widehat{\boldsymbol{\Delta}}_{\mathcal{S}_0^c}\|_1. \end{aligned} \tag{3.47}$$

This together with (3.46) entails that

$$\|\widehat{\boldsymbol{\Delta}}_{\mathcal{S}_0}\|_1 \geq \|\widehat{\boldsymbol{\Delta}}_{\mathcal{S}_0^c}\|_1, \tag{3.48}$$

or that $\widehat{\boldsymbol{\Delta}}$ is sparse (the L_1-norm of $\widehat{\boldsymbol{\Delta}}$ on a much bigger set is controlled by that on a much smaller set) or 'restricted'. For example, with $s = |\mathcal{S}_0|$,

$$\|\widehat{\boldsymbol{\Delta}}\|_2 \geq \|\widehat{\boldsymbol{\Delta}}_{\mathcal{S}_0}\|_2 \geq \|\widehat{\boldsymbol{\Delta}}_{\mathcal{S}_0}\|_1/\sqrt{s} \geq \|\widehat{\boldsymbol{\Delta}}\|_1/(2\sqrt{s}), \tag{3.49}$$

where the last inequality utilizes (3.48). At the same time, since $\widehat{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}_0$ are in the feasible set (3.44), we have $\|n^{-1}\mathbf{X}^T\mathbf{X}\widehat{\boldsymbol{\Delta}}\|_\infty \leq 2\lambda$, which implies further that

$$\|\mathbf{X}\widehat{\boldsymbol{\Delta}}\|_2^2 = \widehat{\boldsymbol{\Delta}}^T(\mathbf{X}^T\mathbf{X}\widehat{\boldsymbol{\Delta}}) \leq \|\mathbf{X}^T\mathbf{X}\widehat{\boldsymbol{\Delta}}\|_\infty\|\widehat{\boldsymbol{\Delta}}\|_1 \leq 2n\lambda\|\widehat{\boldsymbol{\Delta}}\|_1.$$

Using (3.49), we have

$$\|\mathbf{X}\widehat{\boldsymbol{\Delta}}\|_2^2 \le 4n\lambda\sqrt{s}\|\widehat{\boldsymbol{\Delta}}\|_2. \tag{3.50}$$

The regularity condition on $\mathbf{X}$ such as the *restricted eigenvalue condition* (Bickel, Ritov and Tsybakov, 2009)

$$\min_{\|\widehat{\boldsymbol{\Delta}}_{\mathcal{S}_0}\|_1 \ge \|\widehat{\boldsymbol{\Delta}}_{\mathcal{S}_0^c}\|_1} n^{-1}\|\mathbf{X}\widehat{\boldsymbol{\Delta}}\|_2^2/\|\widehat{\boldsymbol{\Delta}}\|_2^2 \ge a$$

implies a convergence in L_2. Indeed, from (3.50), we have

$$a\|\widehat{\boldsymbol{\Delta}}\|_2^2 \le 4\lambda\sqrt{s}\|\widehat{\boldsymbol{\Delta}}\|_2, \quad \text{or} \quad \|\widehat{\boldsymbol{\Delta}}\|_2^2 \le 16a^{-2}\lambda^2 s,$$

which is of order $O(sn^{-1}\log p)$ by choosing the smallest feasible $\lambda = O(\sqrt{2n^{-1}\log p})$ as noted above. Note that the squared error of each nonsparse term is $O(n^{-1})$ and we have to estimate at least s terms of nonsparse parameters. Therefore, $\|\widehat{\boldsymbol{\Delta}}\|_2^2$ should be at least of order $O(s/n)$. The price that we pay for searching the unknown locations of nonsparse elements is merely a factor of $\log p$. In addition, Bickel, Ritov and Tsybakov (2009) show that the Dantzig selector and Lasso are asymptotically equivalent. James, Radchenko and Lv (2009) develop the explicit condition under which the Dantzig selector and Lasso will give identical fits.

The restricted eigenvalue condition basically imposes that the *condition number* (the ratio of the largest to the smallest eigenvalue) is bounded for any matrix $n^{-1}\mathbf{X}_{\mathcal{S}}^T\mathbf{X}_{\mathcal{S}}$ with $|\mathcal{S}| = s$. This requires that the variables in $\mathbf{X}$ are weakly correlated. It does not allow covariates to share common factors and can be very restrictive. A method to weaken this requirement and to adjust for latent *factors* is given by Kneip and Sarda (2011) and Fan, Ke and Wang (2016).

3.3.6 SLOPE and Sorted Penalties

The Sorted L-One (ℓ_1) Penalized Estimation (*SLOPE*) iss introduced in Bogdan *et al.* (2015) to control the *false discovery rate* in variable selection. Given a sequence of penalty levels $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_p \ge 0$, it finds the solution to the sorted ℓ_1 penalized least squares problem

$$\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p}\lambda_j|\beta|_{(j)} \tag{3.51}$$

where $|\beta|_{(1)} \ge \cdots \ge |\beta|_{(p)}$ are the *order statistics* of $\{|\boldsymbol{\beta}|_j\}_{j=1}^p$, namely the decreasing sequence of $\{|\boldsymbol{\beta}|_j\}_{j=1}^p$.

For orthogonal design as in Section 3.2.1 with $\boldsymbol{\varepsilon} \sim N(0, \sigma^2\mathbf{I}_p)$, Bogdan *et al.* (2015) show that the *false discovery rate* for variable selection is controlled at level q if $\lambda_j = \Phi^{-1}(1 - jq/2p)\sigma/\sqrt{n}$, the rescaled critical values used by Benjamini and Hochberg (1995) for *multiple testing*. They also provide a fast computational algorithm. Su and Candés (2016) demonstrate

that it achieves adaptive minimaxity in prediction and coefficient estimation for high-dimensional linear regression. Note that using the tail property of the standard normal distribution (see (3.53)), it is not hard to see that $\lambda_j \approx \sigma\sqrt{(2/n)\log(p/j)}$.

From the bias reduction point of view, the SLOPE is not satisfactory as it is still ℓ_1-based penalty. This motivates Feng and Zhang (2017) to introduce *sorted folded concave penalties* that combine the strengths of concave and sorted penalties. Given a family of univariate penalty functions $p_\lambda(t)$ indexed by λ, the associated estimator is defined as

$$\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} p_{\lambda_j}\big(|\beta|_{(j)}\big).$$

This is a direct extension of (3.51) as it automatically reproduces it with $p_\lambda(t) = \lambda|t|$. The properties of SLOPE and its generalization will be thoroughly investigated in Section 4.5 under a unified framework.

3.3.7 Concentration inequalities and uniform convergence

The uniform convergence appears in a number of occasions for establishing consistency of regularized estimators. See, for example, (3.32), (3.37) and (3.45). It is fundamental to high-dimensional analysis. Let us illustrate the technique to prove (3.45), which is equivalent to showing

$$P\Big\{\|n^{-1}\mathbf{X}\boldsymbol{\varepsilon}\|_\infty \le \lambda\Big\} = P\Big\{\max_{1\le j\le p}|n^{-1}\sum_{i=1}^{n} X_{ij}\epsilon_i| \ge \lambda\Big\} \to 1. \tag{3.52}$$

If we assume $\varepsilon_i \sim N(0,\sigma^2)$, the conditional distribution of $n^{-1}\sum_{i=1}^n X_{ij}\epsilon_i \sim N(0,\sigma^2/n)$ under the standardization $n^{-1}\|\mathbf{X}_j\|^2 = 1$. Therefore, for any $t > 0$, we have

$$\begin{aligned} P\Big\{|n^{-1}\sum_{i=1}^{n} X_{ij}\epsilon_i| \ge t\sigma/\sqrt{n}\Big\} &= 2\int_t^\infty \frac{1}{\sqrt{2\pi}}\exp(-x^2/2)dx \\ &\le \frac{2}{\sqrt{2\pi}}\int_t^\infty \frac{x}{t}\exp(-x^2/2)dx \\ &= \frac{2}{\sqrt{2\pi}}\exp(-t^2/2)/t. \end{aligned} \tag{3.53}$$

In other words, the probability of the average of random variables at least t standard deviation from its mean converges to zero as t goes to ∞ exponentially fast. It is highly concentrated, and such a kind of inequality is called a *concentration inequality*.

Now, by the union bound, (3.52) and (3.53), we have

$$\begin{aligned} P\Big\{\|n^{-1}\mathbf{X}\boldsymbol{\varepsilon}\|_\infty > \frac{t\sigma}{\sqrt{n}}\Big\} &\le \sum_{j=1}^{p} P\Big\{|n^{-1}\sum_{i=1}^{n} X_{ij}\epsilon_i| > \frac{t\sigma}{\sqrt{n}}\Big\} \\ &\le p\frac{2}{\sqrt{2\pi}}\exp(-t^2/2)/t. \end{aligned}$$

Taking $t = \sqrt{2(1+\delta)\log p}$, the above probability is $o(p^{-\delta})$. In other words, with probability at least $1 - o(p^{-\delta})$,

$$\|n^{-1}\mathbf{X}\boldsymbol{\varepsilon}\|_\infty \le \sqrt{2(1+\delta)}\sigma\sqrt{\frac{\log p}{n}}. \tag{3.54}$$

The essence of the above proof relies on the concentration inequality (3.53) and the union bound. Note that the concentration inequalities in general hold for sum of independent random variables with *sub-Gaussian* tails or weaker conditions (see Lemma 4.2). They will appear in later chapters. See Boucheron, Lugosi and Massart (2013) and Tropp (2015) for general treatments. Below, we give a few of them so that readers can get an idea on these inequalities. These types of inequality began with Hoeffding's work in 1963.

Theorem 3.2 (Concentration inequalities) *Assume that $Y_1, \cdots, Y_n$ are independent random variables with mean zero (without loss of generality). Let $S_n = \sum_{i=1}^n Y_i$ be the sum of the random variables.*

a) Hoeffding's inequality: If $Y_i \in [a_i, b_i]$, then

$$P(|S_n| \ge t) \le 2\exp\Big(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\Big).$$

b) Berstein's inequality. If $E|Y_i|^m \le m! M^{m-2} v_i/2$ for every $m \ge 2$ and all i and some positive constants M and v_i, then

$$P(|S_n| \ge t) \le 2\exp\Big(-\frac{t^2}{2(v_1 + \cdots + v_n + Mt)}\Big).$$

See Lemma 2.2.11 of van der Vaart and Wellner (1996).

c) Sub-Gaussian case: If $E\exp(aY_i) \le exp(v_i a^2/2)$ for all $a > 0$ and some $v_i > 0$, then, for any $t > 0$,

$$P(|S_n| \ge t) \le 2\exp\Big(-\frac{t^2}{2(v_1 + \cdots + v_n)}\Big).$$

d) Bounded second moment–Truncated loss: Assume that Y_i are i.i.d. with mean μ and variance σ^2. Let

$$\widehat{\mu}_\tau = \text{argmin}\sum_{i=1}^{n} \rho_\tau(Y_i - \mu), \qquad \rho_\tau(x) = \begin{cases} x^2, & \text{if } |x| \le \tau \\ \tau(2|x| - \tau), & \text{if } |x| > \tau \end{cases}$$

be the adaptive Huber estimator. Then, for $\tau = \sqrt{n}c/t$ *with* $c \geq SD(Y)$ *(standard deviation of* Y*), we have (Fan, Li, and Wang, 2017)*

$$P(|\widehat{\mu}_\tau - \mu| \geq t\frac{c}{\sqrt{n}}) \quad \leq \quad 2\exp(-t^2/16), \quad \forall t \leq \sqrt{n/8},$$

e) Bounded second moment – Truncated data: Set $\widetilde{Y}_i = sgn(Y_i)\min(|Y_i|, \tau)$. *When* $\tau \asymp \sqrt{n}\sigma$, *then*

$$P\Big(\Big|\frac{1}{n}\sum_{i=1}^{n}\widetilde{Y}_i - \mu\Big| \geq t\frac{\sigma}{\sqrt{n}}) \leq 2\exp\Big(-ct^2\Big)$$

for some universal constant c. See Fan, Wang, and Zhu (2016).

Proof. We give a proof of the sub-Gaussian case to illustrate the simple idea. By Makov's inequality, independence, sub-Gaussianity, we have for any $a > 0$

$$P(S_n \geq t) \leq \exp(-at)E\exp(aS_n) \leq \exp(-at)\prod_{i=1}^{n}\exp(v_i a^2/2).$$

By taking the optimal $a = t/(v_1 + \cdots + v_n)$, we obtain

$$P(S_n \geq t) \leq \exp\Big(-\frac{t^2}{2(v_1 + \cdots + v_n)}\Big).$$

This way of obtaining the inequality is called *Chernoff bound*. Now, applying the above inequality to $\{-Y_i\}$, we obtain that

$$P(S_n \leq -t) \leq \exp\Big(-\frac{t^2}{2(v_1 + \cdots + v_n)}\Big).$$

Combining the last two-inequalities, we obtain the result. ■

The common theme of the above results is that the probability of S_n deviating from its mean more than t times of its standard deviation converges to zero in the rate $\exp(-ct^2)$ for some positive constant c. Theorem 3.2(a) is for bounded random variables, whereas Cases b) and c) extend it to the case with sub-Gaussian moments or tails out. They all yield the same rate of convergence. Cases d) and e) extend the results further to the case only with bounded second moment. This line of work began with Catoni (2012). See also Devroye, Lerasle, Lugosi and Oliveira (2016).

3.3.8 A brief history of model selection

Figure 3.6 summarizes the important developments in model selection techniques. Particular emphasis is given to the development of the penalized

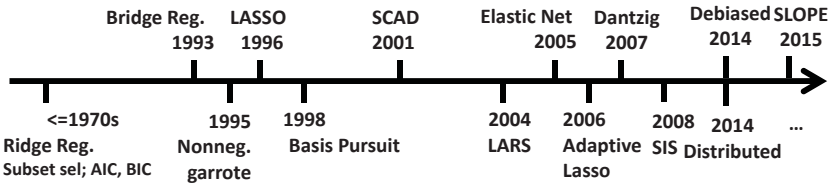

Figure 3.6: A snapshot of history on the major developments of the model selection techniques.

least-squares methods. The list is by far from complete. For example, Bayesian model selection is not even included. It intends only to give readers a snapshot on the some historical developments. For example, the SCAD penalty function was actually introduced by Fan (1997) but its systematic developments were given by Fan and Li (2001), who studied the properties and computation of the whole class of the folded-concave penalized least-squares, not just SCAD. As discussed in Section 3.1.2, AIC and BIC criteria can be regarded as penalized L_0 regression. The idea of the ridge regression and the subset selection appear long before the 1970s (see, e.g., Tikhonov, 1943; Hoerl, 1962).

Sure independence screening, which selects variables based on marginal utilities such as their marginal correlations with the response variable, is not a penalized method. It was introduced by Fan and Lv (2008) to reduce the dimensionality for high-dimensional problems with massive data. It will be systematically introduced in Chapter 8. Because of its importance in analysis of big data and that it can be combined with PLS, we include it here for completeness.

Debiased Lasso was proposed by Zhang and Zhang (2014), which is further extended by van de Geer, Bühlmann, Ritov, and Dezeure (2014) and improved by Javanmard and Montanari (2014). For *distributed estimation* of high-dimensional problem, see Chen and Xie (2014), Shamir, Srebro and Zhang (2014), Lee, Liu, Sun and Taylor (2017), Battey, Fan, Liu, Lu and Zhu (2018), Jordan, Lee and Yang (2018), among others.

3.4 Bayesian Variable Selection

3.4.1 Bayesian view of the PLS

Sparse penalized regression can be put in the Bayesian framework. One can regard the parameters $\{\beta_j\}_{j=1}^p$ as a realization from a prior distribution having a density $\pi(\cdot)$ with the mode at the origin. If the observed data $\mathbf{Y}$ has a density $p_Y(\mathbf{Y}|\mathbf{X}\boldsymbol{\beta})$ (conditioned on $\mathbf{X}$), then the joint density of the data and parameters is given by

$$f(\mathbf{Y};\boldsymbol{\beta}) = p_Y(\mathbf{Y}|\mathbf{X}\boldsymbol{\beta})\pi(\boldsymbol{\beta}).$$

The posterior distribution of $\boldsymbol{\beta}$ given $\mathbf{Y}$ is $f(\mathbf{Y};\boldsymbol{\beta})/g(\mathbf{Y})$, which is proportional to $f(\mathbf{Y};\boldsymbol{\beta})$ as a function of $\boldsymbol{\beta}$, where $g(\mathbf{Y})$ is the marginal distribution of $\mathbf{Y}$. Bayesian inference is based on the posterior distribution of $\boldsymbol{\beta}$ given $\mathbf{X}$ and $\mathbf{Y}$.

One possible estimator is to use the *posterior mean* $E(\boldsymbol{\beta}|\mathbf{X},\mathbf{Y})$ to estimate $\boldsymbol{\beta}$. Another is the *posterior mode*, which finds

$$\widehat{\boldsymbol{\beta}} = \text{argmax}_{\boldsymbol{\beta}\in R^p} \log p_Y(\mathbf{Y}|\mathbf{X}\boldsymbol{\beta}) + \log \pi(\boldsymbol{\beta}).$$

It is frequently taken as a *Bayesian estimator*. In particular, when $\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\beta}, \mathbf{I}_n)$ (the standard deviation is taken to be one for convenience), then

$$\log p_Y(\mathbf{Y}|\mathbf{X}\boldsymbol{\beta}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2.$$

Thus, finding the posterior mode reduces to minimizing

$$\frac{1}{2}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 - \log \pi(\boldsymbol{\beta}).$$

Typically, the prior distributions are taken to be independent: $\pi(\boldsymbol{\beta}) = \prod_{j=1}^p \pi_j(\boldsymbol{\beta}_j)$ where $\pi_j(\cdot)$ is the marginal prior for β_j, though this is not mandatory. In this case, the problem becomes the penalized least-squares

$$\frac{1}{2}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 - \sum_{j=1}^p \log \pi_j(\beta_j). \tag{3.55}$$

When $\beta_j \sim_{i.i.d.} \exp(-p_\lambda(|\beta_j|))$ where we hide the normalization constant, (3.55) becomes the penalized least-squares (3.9). In particular, when $\beta_j \sim_{i.i.d.} \lambda\exp(-\lambda|\beta_j|)/2$, the double exponential distribution with scale parameter λ, the above minimization problem becomes the Lasso problem (3.29). Note that when $p_\lambda(|\beta_j|)$ is flat (constant) at the tails, the function $\exp(-p_\lambda(|\beta_j|))$ cannot be scaled to be a density function as it is not integrable. Such a prior is called an *improper prior*, one with very heavy tails. SCAD penalty corresponds to an improper prior.

The prior $\pi_j(\theta)$ typically involves some parameters $\boldsymbol{\gamma}$, called *hyper parameters*. An example is the scale parameter λ in the double exponential distribution. One can regard them as the parameters generated from some other prior distributions. Such methods are called *hierarchical Bayes*. They can also be regarded as fixed parameters and are estimated through maximum likelihood, maximizing the marginal density $g(\mathbf{Y})$ of $\mathbf{Y}$ with respect to $\boldsymbol{\gamma}$. In other words, one estimates $\boldsymbol{\gamma}$ by the maximum likelihood and employs a Bayes rule to estimate parameters $\boldsymbol{\beta}$ for the given estimated $\boldsymbol{\gamma}$. This procedure is referred to as *empirical Bayes*. Park and Casella (2008) discuss the use of empirical Bayes in the *Bayesian Lasso* by exploiting a hierarchical representation of the double exponential distribution as a scale mixture of normals (Andrews and Mallows 1974):

$$\frac{a}{2}e^{-a|z|} = \int_0^\infty \frac{e^{-z^2/(2s)}}{\sqrt{2\pi s}} \frac{a^2}{2} e^{-a^2 s/2} ds, \ a > 0.$$

They develop a nice Gibbs sampler for sampling the posterior distribution. As demonstrated in Efron (2010), the empirical Bayes plays a very prominent role in large-scale statistical inference.

3.4.2 A Bayesian framework for selection

Bayesian inference of $\boldsymbol{\beta}$ and model selection are related but not identical problems. Bayesian model selection can be more complex. To understand this, let us denote $\{\mathcal{S}\}$ as all possible models, each model has a prior probability $p(\mathcal{S})$. For example, $\mathcal{S}$ is equally likely among models with the same size and assign the probability proportion to $|\mathcal{S}|^{-\gamma}$. We can also assign the prior probability p_j to models with size j such that models of size j are all equally likely. Within each model $\mathcal{S}$, there is a parameter vector $\boldsymbol{\beta}_{\mathcal{S}}$ with prior $\pi_{\mathcal{S}}(\cdot)$. In this case, the "joint density" of the models, the model parameters, and the data is

$$p(\mathcal{S})\pi_{\mathcal{S}}(\boldsymbol{\beta}_{\mathcal{S}})p_Y(\mathbf{Y}|\mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}_{\mathcal{S}}).$$

There is a large amount of literature on *Bayesian model selection*. The posterior modes are in general computed by using the *Markov Chain Monte Carlo*. See for example Andrieu, De Freitas, Doucet and Jordan (2003) and Liu (2008). Bayesian model selection techniques are very powerful in many applications, where disciplinary knowledge can be explicitly incorporated into the prior. See, for example, Raftery (1995).

A popular Bayesian idea for variable selection is to introduce p latent binary variables $Z = (z_1, \ldots, z_p)$ such that $z_j = 1$ means variable x_j should be included in the model and $z_j = 0$ means excluding x_j. Given $z_j = 1$, the distribution of β_j has a flat tail (slab), but the distribution of β_j given $z_j = 0$ is concentrated at zero (spike). The marginal distribution of β_j is a *spike and slab prior*. For example, assume that β_j is generated from a mixture of the point mass at 0 and a distribution $\pi_j(\beta)$ with probability α_j:

$$\beta_j \sim \alpha_j\delta_0 + (1-\alpha_j)\pi_j(\beta).$$

See Johnstone and Silverman (2005) for an interesting study of this in wavelet regularization. For computation considerations, the spike distribution is often chosen to be a normal distribution with mean zero and a small variance. The slab distribution is another normal distribution with mean zero and a much bigger variance. See, for example, George and McCulloch (1993), Ishwaran and Rao (2005) and Narisetty and He (2014), among others. A working Bayesian model selection model with the *Gaussian spike and slab prior* is given as follows:

$$\begin{aligned}
\mathbf{Y}|(\mathbf{X}, \boldsymbol{\beta}, \sigma^2) &\sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n),\\
\beta_j|(Z_j = 0, \sigma^2) &\sim N(0, \sigma^2 v_0),\\
\beta_j|(Z_j = 1, \sigma^2) &\sim N(0, \sigma^2 v_1),\\
P(Z_j = 1) &= q,\\
\sigma^2 &\sim IG(\alpha_1, \alpha_2).
\end{aligned}$$

where IG denotes the inverse Gamma distribution. The data generating process is bottom up in the above representation. The joint posterior distribution $P(\boldsymbol{\beta}, Z, \sigma^2|\mathbf{Y}, \mathbf{X})$ can be sampled by a neat Gibbs sampler. Model selection is based on the marginal posterior probabilities $P(Z_j = 1|\mathbf{Y}, \mathbf{X})$. According to Barbieri and Berger (2004), x_j is selected if $P(Z_j = 1|\mathbf{Y}, \mathbf{X}) \geq 0.5$. This selection method leads to the *median probability model* which is shown to be predictive optimal. For the high-dimension setting $p \gg n$, Narisetty and He (2014) establish the frequentist selection consistency of the Bayesian approach by using dimension-varying prior parameters: $v_0 = v_0(n, p) = o(n^{-1})$, $v_1 = v_1(n, p) = O(\frac{p^{2+\delta}}{n})$ and $q = q(n, p) \approx p^{-1}$.

3.5 Numerical Algorithms

This section introduces some early developed algorithms to compute the folded-concave penalized least-squares. We first present the algorithms for computing the Lasso as it is more specific. We then develop algorithms for more general folded concave PLS such as SCAD and MCP. In particular, the connections between the folded-concave PLS and iteratively reweighted adaptive Lasso are made. These algorithms provide us not only a way to implement PLS but also statistical insights on the procedures.

In many applications, we are interested in finding the solution to PLS (3.9) for a range of values of λ. The solutions $\widehat{\boldsymbol{\beta}}(\lambda)$ to the PLS as a function of λ are called *solution paths* or *coefficient paths* (Efron, Hastie, Johnstone and Tibshirani, 2004). This allows one to examine how the variables enter into the solution as λ decreases. Figure 3.7 gives an example of coefficient paths.

Each section below is independent, where some sections are harder than others, and can be skipped without significant impact on understanding the other sections.

3.5.1 Quadratic programs

There are several algorithms for computing Lasso: Quadratic programming, least-angle regression, and coordinate descent algorithm. The first two algorithms are introduced in this and next sections, and the last one will be introduced in Section 3.5.6.

First of all, as in Tibshirani (1996), a convenient alternative is to express the penalized L_1-regression (3.29) into its dual problem (3.39). Each λ determines a constant c and vice versa. The relationship depends on the data $(\mathbf{X}, \mathbf{Y})$.

The quadratic program, employed by Tibshirani (1996), is to regard the constraints $\|\boldsymbol{\beta}\|_1 \leq c$ as 2^p linear constraints $\mathbf{b}_j^T\boldsymbol{\beta} \leq c$ for all p-tuples $\mathbf{b}_j$ of form $(\pm 1, \pm 1, \cdots, \pm 1)$. A simple solution is to write (3.39) as

$$\begin{array}{ll} & \min_{\boldsymbol{\beta}^+, \boldsymbol{\beta}^-} \|\mathbf{Y} - \mathbf{X}(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-)\|^2 \\ \text{s.t.} & \sum_{i=1}^p \beta_i^+ + \sum_{i=1}^p \beta^- \leq c, \quad \beta_i^+ \geq 0, \quad \beta_i^- \geq 0. \end{array} \tag{3.56}$$

This is a $2p$-variable convex optimization problem and the constraints are lin-

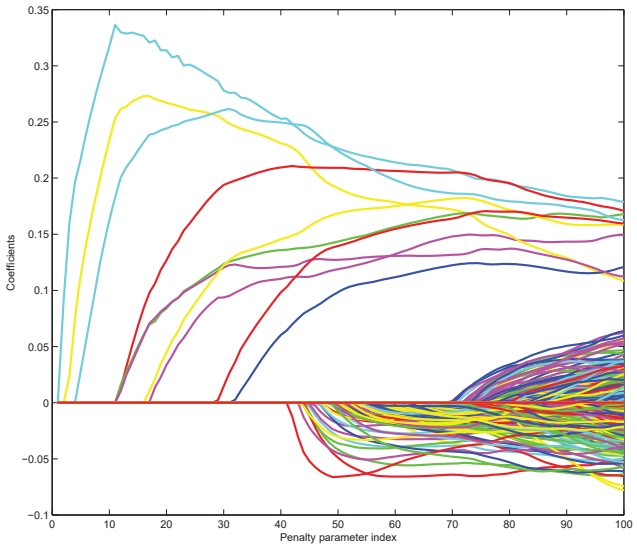

Figure 3.7: Solution paths $\widehat{\boldsymbol{\beta}}(\lambda)$ as a function of $1/\lambda$. For each given value of λ, only non-vanishing coefficients are presented. As λ decreases, we can examine how variables enter into the regression. Each curve shows $\widehat{\beta}_j(\lambda)$ as a function of λ for an important regressor X_j.

ear in those variables. Therefore, the standard *convex optimization* algorithms and solvers (Boyd and Vandenberghe, 2004) can be employed. An alternative expression to the optimization problem (3.39) is

$$\begin{aligned} &\min_{\boldsymbol{\beta},\boldsymbol{\gamma}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 \\ \text{s.t.} \quad &-\gamma_i \le \beta_i \le \gamma_i, \quad \textstyle\sum_{i=1}^p \gamma_i \le c, \quad \gamma_i \ge 0. \end{aligned} \tag{3.57}$$

This is again a $2p$-variable convex optimization problem with linear constraints.

To find the solution paths, one needs to repeatedly solve a quadratic programming problem for a grid values of c. This is very inefficient and does not offer statistical insights.

Osborne, Presnell and Turlach (2000) expressed the L_1 constraint as

$$\operatorname{sgn}(\boldsymbol{\beta})^T\boldsymbol{\beta} \le c.$$

They treated the problem as a quadratic program with $\operatorname{sgn}(\boldsymbol{\beta})$ taken from the previous step of the iteration and developed a "homotopy method" based on this linearized constraint. Their homotopy method is related to the solution-path algorithm of Efron *et al.* (2004).

3.5.2 Least angle regression*

Efron *et al.* (2004) introduce the Least-Angle Regression (LARS) to explain the striking but mysterious similarity between the lasso regression path and the ϵ-boosting linear regression path observed by Hastie, Friedman and Tibshirani in 2001. Efron *et al.* (2004) show that the lasso regression and ϵ-boosting linear regression are two variants of *LARS* with different small modifications, thus explaining their similarity and differences. LARS itself is also an interesting procedure for variable selection and model estimation.

LARS is a forward stepwise selection procedure but operates in a less greedy way than the standard *forward selection* does. Assuming that all variables have been standardized so that they have mean-zero and unit variance, we now describe the LARS algorithm for the constrained least-square problem (3.39). Let $\mathbf{z} = \mathbf{X}^T\mathbf{Y}/n$ and $\boldsymbol{\chi}_j = \mathbf{X}^T\mathbf{X}_j/n$, where $\mathbf{X}_j$ is the j^{th} column of $\mathbf{X}$. Then, necessary conditions for minimizing the Lasso problem (3.29) are [see (3.30) and (3.31)]

$$\begin{cases} \tau z_j - \boldsymbol{\chi}_j^T\mathbf{b} = \text{sgn}(b_j) & \text{if } b_j \neq 0, \\ |\tau z_j - \boldsymbol{\chi}_j^T\mathbf{b}| \leq 1 & \text{if } b_j = 0, \end{cases} \tag{3.58}$$

where $\tau = 1/\lambda$ and $\mathbf{b} = \tau\widehat{\boldsymbol{\beta}}$. The solution path is given by $\widehat{\mathbf{b}}(\tau)$ that solves (3.58). When $\tau \leq 1/\|n^{-1}\mathbf{X}^T\mathbf{Y}\|_\infty$, as noted in Section 3.3.2, the solution is $\widehat{\mathbf{b}}(\tau) = 0$. We now describe the LARS algorithm for the constrained least-square problem (3.39). First of all, when $c = 0$ in (3.57), no variables are selected. This corresponds to $\widehat{\boldsymbol{\beta}}(\lambda) = 0$ for $\lambda > \|n^{-1}\mathbf{X}^T\mathbf{Y}\|_\infty$, as noted in Section 3.3.2.

As soon as c moves slightly away from zero, one picks only one variable ($\mathbf{X}_1$, say) that has the maximum absolute correlation (least angle) with the response variable $\mathbf{Y}$. Then, $\widehat{\boldsymbol{\beta}}_c = (\text{sgn}(r_1)c, 0, \cdots, 0)^T$ is the solution to problem (3.39) for sufficiently small c, where r_1 is the correlation between $\mathbf{X}_1$ and $\mathbf{Y}$. Now, as c increases, the absolute correlation between the current residual

$$\mathbf{R}_c = \mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_c$$

and $\mathbf{X}_1$ decreases until a (smallest) value c_1 at which there exists a second variable $\mathbf{X}_2$, say, that has the same absolute correlation (equal angle) with $\mathbf{R}_{c_1}$:

$$|\text{cor}(\mathbf{X}_1, \mathbf{R}_{c_1})| = |\text{cor}(\mathbf{X}_2, \mathbf{R}_{c_1})|.$$

Then, $\widehat{\boldsymbol{\beta}}_c$ is the solution to problem (3.39) for $0 \leq c \leq c_1$ and the value c_1 can easily be determined, as in (3.62) below.

LARS then proceeds equiangularly between $\mathbf{X}_1$ and $\mathbf{X}_2$ until a third variable, $\mathbf{X}_3$ (say), joins the rank of "most correlated variables" with the current residuals. LARS then proceeds equiangularly between $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$ and so on. As we proceed down this path, the maximum of the absolute correlation of covariates with the current residual keeps decreasing until it becomes zero.

The equiangular direction of a set of variables $\mathbf{X}_{\mathcal{S}}$ is given by

$$\mathbf{u}_{\mathcal{S}} = \mathbf{X}_{\mathcal{S}}(\mathbf{X}_{\mathcal{S}}^T\mathbf{X}_{\mathcal{S}})^{-1}\mathbf{1}/w_{\mathcal{S}} \equiv \mathbf{X}_{\mathcal{S}}\widehat{\boldsymbol{\beta}}_{\mathcal{S}} \tag{3.59}$$

where $\mathbf{1}$ is a vector of $1's$ and $w_{\mathcal{S}}^2 = \mathbf{1}^T(\mathbf{X}_{\mathcal{S}}^T\mathbf{X}_{\mathcal{S}})^{-1}\mathbf{1}$ is a normalization constant. The equiangular property can easily be seen:

$$\mathbf{X}_{\mathcal{S}}^T\mathbf{u}_{\mathcal{S}} = \mathbf{1}/w_{\mathcal{S}}, \qquad \|\mathbf{u}_{\mathcal{S}}\| = 1.$$

We now furnish some details of LARS. Assume that $\mathbf{X}$ is of full rank. Start from $\boldsymbol{\mu}_0 = 0$, $\mathcal{S} = \varnothing$, the empty set, and $\boldsymbol{\beta}_{\mathcal{S}} = 0$. Let $\mathcal{S}$ be the current active set of variables and $\widehat{\boldsymbol{\mu}}_{\mathcal{S}} = \mathbf{X}_{\mathcal{S}}\widehat{\boldsymbol{\beta}}_{\mathcal{S}}$ be its current fitted value of $\mathbf{Y}$. Compute the marginal correlations of covariates $\mathbf{X}$ with the current residual (except a normalization constant)

$$\widehat{\mathbf{c}} = \mathbf{X}^T(\mathbf{Y} - \widehat{\boldsymbol{\mu}}_{\mathcal{S}}). \tag{3.60}$$

Define $s_j = \operatorname{sgn}(\widehat{c}_j)$ for $j \in \mathcal{S}$. Note that the absolute correlation does not change if the columns of $\mathbf{X}$ are multiplied by ± 1. Update the active set of variables by taking the most correlated set

$$\mathcal{S}_{\text{new}} = \{j : |\widehat{c}_j| = \|\widehat{\mathbf{c}}\|_\infty\} \tag{3.61}$$

and compute the equiangular direction $\mathbf{u}_{\mathcal{S}_{\text{new}}}$ by (3.59) using variables $\{\operatorname{sgn}(\widehat{c}_j)\mathbf{X}_j, j \in \mathcal{S}\}$. For the example in the beginning of this section, if $\mathcal{S} = \phi$, an empty set, then $\widehat{\boldsymbol{\mu}}_{\mathcal{S}} = 0$ and $\mathcal{S}_{\text{new}} = \{1\}$. If $\mathcal{S} = \{1\}$, then $\widehat{\boldsymbol{\mu}}_{\mathcal{S}} = \mathbf{X}\widehat{\boldsymbol{\beta}}_{c_1} = \operatorname{sgn}(r_1)c_1\mathbf{X}_1$ and $\mathcal{S}_{\text{new}} = \{1, 2\}$.

Now compute

$$\gamma_{\mathcal{S}} = \min_{j\in\mathcal{S}^c}{}^{+}\left\{\frac{\|\mathbf{c}\|_\infty - \widehat{c}_j}{w_{\mathcal{S}}^{-1} - a_j}, \frac{\|\mathbf{c}\|_\infty + \widehat{c}_j}{w_{\mathcal{S}}^{-1} + a_j}\right\}, \qquad \mathbf{a} = \mathbf{X}^T\mathbf{u}_{\mathcal{S}}. \tag{3.62}$$

where "$\min^+$" is the minimum taken only over positive components. It is not hard to show that this step size $\gamma_{\mathcal{S}}$ is the smallest positive constant γ such that some new indices will join the active set (see Efron *et al.*, 2004). For example, if $\mathcal{S} = \{1\}$, this $\gamma_{\mathcal{S}}$ is c_1 in the first step. Update the fitted value along the equiangular direction by

$$\widehat{\boldsymbol{\mu}}_{\mathcal{S}_{\text{new}}} = \widehat{\boldsymbol{\mu}}_{\mathcal{S}} + \gamma_{\mathcal{S}_{\text{new}}}\mathbf{u}_{\mathcal{S}_{\text{new}}}. \tag{3.63}$$

The solution path for $\gamma \in (0, \gamma_{\mathcal{S}_{\text{new}}})$ is

$$\widehat{\boldsymbol{\mu}}_{\mathcal{S}_{\text{new}},\gamma} = \widehat{\boldsymbol{\mu}}_{\mathcal{S}} + \gamma\mathbf{u}_{\mathcal{S}_{\text{new}}}.$$

Note that $\mathcal{S}_{\text{new}}$ is always a bigger set than $\mathcal{S}$. Write $\widehat{\boldsymbol{\mu}}_{\mathcal{S}} = \mathbf{X}\widehat{\boldsymbol{\beta}}_{\mathcal{S}}$, in which $\widehat{\boldsymbol{\beta}}_{\mathcal{S}}$ has support $\mathcal{S}$ so that $\widehat{\boldsymbol{\mu}}_{\mathcal{S}}$ is in the linear space spanned by columns of $\mathbf{X}_{\mathcal{S}}$. By (3.59), we have

$$\widehat{\boldsymbol{\mu}}_{\mathcal{S}_{\text{new}},\gamma} = \mathbf{X}(\widehat{\boldsymbol{\beta}}_{\mathcal{S}} + \gamma\boldsymbol{\beta}_{\mathcal{S}_{\text{new}}}).$$

Note that by (3.59), $\widehat{\boldsymbol{\beta}}_{\mathcal{S}} + \gamma \boldsymbol{\beta}_{\mathcal{S}_{\text{new}}}$ has a support $\mathcal{S}_{\text{new}}$. In terms of coefficients, we have updated the coefficients from $\widehat{\boldsymbol{\beta}}_{\mathcal{S}}$ for variables $\mathbf{X}_{\mathcal{S}}$ to

$$\widehat{\boldsymbol{\beta}}_{\mathcal{S}_{\text{new}},\gamma} = \widehat{\boldsymbol{\beta}}_{\mathcal{S}} + \gamma \boldsymbol{\beta}_{\mathcal{S}_{\text{new}}} \tag{3.64}$$

for variables $\mathbf{X}_{\mathcal{S}_{\text{new}}}$, expressed in R^p. Some modifications of the signs in the second term in (3.64) is needed since we use the variables $\{\text{sgn}(\widehat{c}_j)\mathbf{X}_j, j \in \mathcal{S}\}$ rather than $\mathbf{X}_{\mathcal{S}}$ to compute the equiangular direction $\mathbf{u}_{\mathcal{S}_{\text{new}}}$.

The LARS algorithm is summarized as follows.

- **Initialization**: Set $\mathcal{S} = \phi$, $\widehat{\boldsymbol{\mu}}_{\mathcal{S}} = 0$, $\widehat{\boldsymbol{\beta}}_{\mathcal{S}} = \mathbf{0}$.
- **Step 1**: Compute the current correlation vector $\widehat{\mathbf{c}}$ by (3.60), the new subset $\mathcal{S}_{\text{new}}$, the least angular covariates with the current residual $\mathbf{Y} - \widehat{\boldsymbol{\mu}}_{\mathcal{S}}$, by (3.61), and the stepsize $\gamma_{\mathcal{S}_{\text{new}}}$, the largest stepsize along the equiangular direction, by (3.62).
- **Step 2**: Update $\mathcal{S}$ with $\mathcal{S}_{\text{new}}$, $\widehat{\boldsymbol{\mu}}_{\mathcal{S}}$ with $\widehat{\boldsymbol{\mu}}_{\mathcal{S}_{\text{new}}}$ in (3.63), and $\widehat{\boldsymbol{\beta}}_{\mathcal{S}}$ with $\widehat{\boldsymbol{\beta}}_{\mathcal{S}_{\text{new}}}$ in (3.64) with $\gamma = \gamma_{\mathcal{S}_{\text{new}}}$.
- **Iterations**: Iterate between Steps 1 and 2 until all variables are included in the model and the solution reaches the OLS estimate.

The entire LARS solution path simply connects p-dimensional coefficients linearly at each discrete step above. However, it is not necessarily the solution to the lasso problem (3.39). The LARS model size is enlarged by one after each step, but the Lasso may also drop a variable from the current model as c increases. Technically speaking, (3.30) shows that Lasso and the current correlation must have the same sign, but the LARS solution path does not enforce this. Efron *et al.* (2004) show that this sign constraint can easily be enforced in the LARS algorithm: during the ongoing LARS update step, if the $\widetilde{j}$th variable in $\mathcal{S}$ has a sign change before the new variable enters $\mathcal{S}$, stop the ongoing LARS update, drop the $\widetilde{j}$th variable from the model and recalculate the new equiangular direction for doing the LARS update. Efron *et al.* (2004) prove that the modified LARS path is indeed the Lasso solution path under a "one at a time" condition, which assumes that at most one variable can enter or leave the model at any time.

Other modifications of the LARS algorithm are also possible. For example, by modifying LARS shrinkage, James and Radchenko (2008) introduce variable inclusion and shrinkage algorithms (VISA) that intend to attenuate the over-shrinkage problem of Lasso. James, Radchenko and Lv (2009) develop an algorithm called *Dasso* that allows one to fit the entire path of regression coefficients for different values of the Dantzig selector tuning parameter.

The key argument in the LARS algorithm is the piecewise linearity property of the Lasso solution path. This property is not unique to the Lasso PLS. Many statistical models can be formulated as $\min\{\text{Loss} + \lambda\text{Penalty}\}$. In Rosset and Zhu (2007) it is shown that if the loss function is almost quadratic and the penalty is L_1 (or piecewise linear), then the solution path is piecewise linear as a function of λ. Examples of such models include the L_1 penalized

Huber regression (Rosset and Zhu, 2007) and the L_1 penalized support vector machine (Zhu, Rosset, Hastie and Tibshirani, 2004). Interestingly, if the loss function is L_1 (or piecewise linear) and the penalty function is quadratic, we can switch their roles when computing and the solution path is piecewise linear as a function of $1/\lambda$. See, for example, the solution path algorithms for the support vector machine (Hastie, Rosset, Tibshirani and Zhu, 2003) and support vector regression (Gunter and Zhu, 2007).

3.5.3 Local quadratic approximations

Local quadratic approximation (LQA) was introduced by Fan and Li (2001) before LARS-Lasso or other effective methods that are available in statistics for computing Lasso. It allows statisticians to implement folded concave penalized likelihood and to compute the standard error of estimated nonzero components. Given an initial value β_0, approximate the function $p_\lambda(|\beta|)$ locally at this point by a quadratic function $q(\beta|\beta_0)$. This quadratic function is required to be symmetric around zero, satisfying

$$q(\beta_0|\beta_0) = p_\lambda(|\beta_0|) \quad \text{and} \quad q'(\beta_0|\beta_0) = p'_\lambda(|\beta_0|).$$

These three conditions determine uniquely the quadratic function

$$q(\beta|\beta_0) = p_\lambda(|\beta_0|) + \frac{1}{2}\frac{p'_\lambda(|\beta_0|)}{|\beta_0|}(\beta^2 - \beta_0^2). \tag{3.65}$$

See Figure 3.8.

Given the current estimate $\boldsymbol{\beta}_0$, by approximating each folded concave function in PLS (3.9) by its LQA, our target becomes minimizing

$$Q(\boldsymbol{\beta}|\boldsymbol{\beta}_0) = \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} q(\beta_j|\beta_{j0}). \tag{3.66}$$

Minimizing (3.66) is the same as minimizing

$$\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p}\frac{p'_\lambda(|\beta_{j0}|)}{2|\beta_{j0}|}\beta_j^2.$$

This is a ridge regression problem with solution computed analytically as

$$\widehat{\boldsymbol{\beta}}_{\text{new}} = (\mathbf{X}^T\mathbf{X} + n\,\text{diag}\{p'_\lambda(|\beta_{j0}|)/|\beta_{j0}|\})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{3.67}$$

The LQA is to iteratively use (3.67), starting from an initial value (e.g. univariate marginal regression coefficients). Fan and Li (2001) note that the approximation (3.65) is not good when $|\beta_{j0}| \le \varepsilon_0$, a tolerance level. When this happens, delete variables from the model before applying (3.67). This speeds

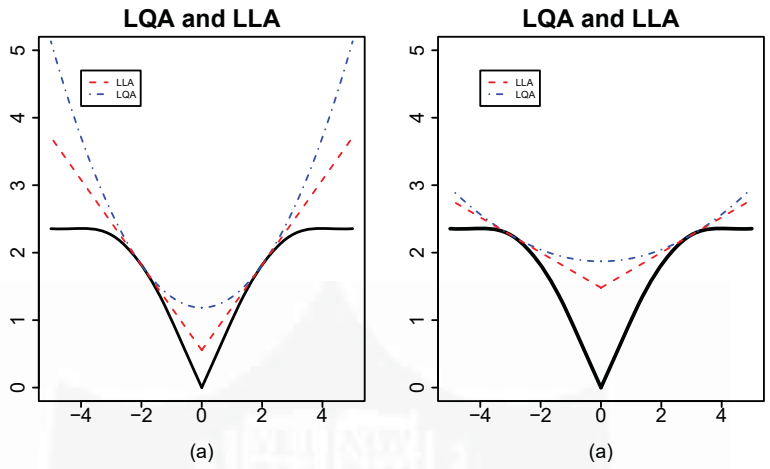

Figure 3.8: Local quadratic and local linear approximations to SCAD with $\lambda = 1$ and $a = 3.7$ at at point x. (a) $x = 2$ and (b) $x = 3$.

up the computation. Furthermore, they proposed to compute the standard error for surviving variables using (3.67), as if $p'_\lambda(\beta_{j0})/|\beta_{j0}|$ were non-stochastic. They validated the accuracy of the estimated standard error. See Fan and Peng (2004) for a theoretical proof.

Does the algorithm converge? and if so, in what sense? Hunter and Li (2005) realized that the local quadratic approximation is a specific case of the *majorization-minimization* (MM) algorithm (Hunter and Lange, 2000). First of all, as shown in Figure 3.8, thanks to the folded-concaveness,

$$q(\beta|\beta_0) \geq p_\lambda(\beta) \quad \text{and} \quad q(\beta_0|\beta_0) = p_\lambda(\beta_0),$$

namely $q(\beta|\beta_0)$ is a convex majorant of $p_\lambda(\cdot)$ with $q(\beta_0|\beta_0) = p_\lambda(|\beta_0|)$. This entails that

$$Q(\boldsymbol{\beta}|\boldsymbol{\beta}_0) \geq Q(\boldsymbol{\beta}) \quad \text{and} \quad Q(\boldsymbol{\beta}_0|\boldsymbol{\beta}_0) = Q(\boldsymbol{\beta}_0), \tag{3.68}$$

namely, $Q(\boldsymbol{\beta}|\boldsymbol{\beta}_0)$ is a convex majorization of the folded-concave PLS $Q(\boldsymbol{\beta})$ defined by (3.9). Let $\boldsymbol{\beta}_{\text{new}}$ minimize $Q(\boldsymbol{\beta}|\boldsymbol{\beta}_0)$. Then, it follows from (3.68) that

$$Q(\boldsymbol{\beta}_{\text{new}}) \leq Q(\boldsymbol{\beta}_{\text{new}}|\boldsymbol{\beta}_0) \leq Q(\boldsymbol{\beta}_0|\boldsymbol{\beta}_0) = Q(\boldsymbol{\beta}_0), \tag{3.69}$$

where the second inequality follows from the definition of the minimization. In other words, the target function decreases after each iteration and will converge.

3.5.4 Local linear algorithm

With LARS and other efficient algorithms for computing Lasso, the *local linear approximation* (LLA) approximates $p_\lambda(|\beta|)$ at β_0 by

$$l(\beta|\beta_0) = p_\lambda(|\beta_0|) + p'_\lambda(|\beta_0|)(|\beta| - |\beta_0|),$$

which is the first-order Taylor expansion of $p_\lambda(|\beta|)$ at the point β_0. Clearly, as shown in Figure 3.8, $l(\beta|\beta_0)$ is a better approximation than LQA $q(\beta|\beta_0)$. Indeed, it is the minimum convex majorant of $p_\lambda(|\beta|)$ with $l(\beta_0|\beta_0) = p_\lambda(|\beta_0|)$.

With the local linear approximation, given the current estimate $\boldsymbol{\beta}_0$, the folded concave penalized least-squares (3.9) now becomes

$$\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} p'_\lambda(|\beta_{j0}|)|\beta_j|, \tag{3.70}$$

after ignoring the constant term. This is now an adaptively weighted Lasso problem and can be solved by using algorithms in Sections 3.5.1 and 3.5.2. The algorithm was introduced by Zou and Li (2008). As it is also a specific MM algorithm, as shown in (3.69), the LLA algorithm also enjoys a decreasing target value property (3.69).

Unlike LQA, if one component hits zero at a certain step, it will not always be stuck at zero. For example, if $\boldsymbol{\beta}_0 = 0$, (3.70) reduces to Lasso. In this view, even when the initial estimator is very crude, LLA gives a good one-step estimator.

Through LLA approximation (3.70), the folded-concave PLS can be viewed as an iteratively reweighted penalized L_1 regression. The weights depend on where the current estimates are. The larger the magnitude, the smaller the weighted penalty. This reduces the biases for estimating large true coefficients. Lasso is the one-step estimator of the folded concave PLS with initial estimate $\widehat{\boldsymbol{\beta}} = \mathbf{0}$. Lasso puts a full stop yet the folded concave PLS iterates further to reduce the bias due to the Lasso shrinkage.

It is now clear that the adaptive Lasso is a specific example of the LLA implementation of the folded-concave PLS with $p'_\lambda(|\beta|) = |\beta|^{-\gamma}$. This function is explosive near 0 and is therefore inappropriate to use in the iterative application of (3.70): once a component β_j hits zero at certain iteration, its weights cannot be computed or the variable X_j is eliminated forever.

The LLA implementation of folded concave PLS has a very nice theoretical property. Fan, Xue and Zou (2014) show that with Lasso as the initial estimator, with high probability, the LLA implementation (3.70) produces the oracle estimator in one step. The result holds in a very general likelihood-based context under some mild conditions. This gives additional endorsement of the folded-concave PLS implemented by LLA (3.70).

The implementation of LLA is available in the R package called "SIS" (function: *scadglm*), contributed by Fan, Feng, Samworth, and Wu.

*3.5.5 Penalized linear unbiased selection**

The penalized linear unbiased selection (PLUS) algorithm, introduced by Zhang (2010), finds multiple local minimizers of folded-concave PLS in a branch of the graph (indexed by $\tau = \lambda^{-1}$) of critical points determined by (3.22) and (3.23). The PLUS algorithm deals with the folded concave-penalized functions $p_\lambda(t) = \lambda^2\rho(t/\lambda)$, in which $\rho(\cdot)$ is a quadratic spline. This includes L_1, SCAD (3.14), hard-threhsolding penalty (3.15) and MCP (3.17) as specific examples. Let $t_1 = 0, \cdots, t_m$ be the knots of the quadratic spline $\rho(\cdot)$. Then, the derivative of $\rho(\cdot)$ can be expressed as

$$\rho'(t) = \sum_{i=1}^{m}(u_i - v_i t)I(t_i < t \le t_{i+1}), \tag{3.71}$$

for some constants $\{v_i\}_{i=1}^m$, in which $u_1 = 1$ (normalization), $u_m = v_m = 0$ (flat tail), and $t_{m+1} = \infty$. For example, L_1 penalty corresponds to $m = 1$; MCP corresponds to $m = 2$ with $t_2 = a$, $v_1 = 1/a$; SCAD corresponds to $m = 3$ with $t_2 = 1$, $t_3 = a$, $v_1 = 0$, $u_2 = a/(a-1)$ and $v_2 = 1/(a-1)$.

Let $\mathbf{z} = \mathbf{X}^T\mathbf{Y}/n$ and $\boldsymbol{\chi}_j = \mathbf{X}^T\mathbf{X}_j/n$, where $\mathbf{X}_j$ is the j^{th} column of $\mathbf{X}$. Then, estimating equations (3.22) and (3.23) can be written as

$$\begin{cases} \tau z_j - \boldsymbol{\chi}_j^T\mathbf{b} = \text{sgn}(b_j)\rho'(|b_j|) & \text{if } b_j \neq 0, \\ |\tau z_j - \boldsymbol{\chi}_j^T\mathbf{b}| \le 1 & \text{if } b_j = 0, \end{cases} \tag{3.72}$$

where $\tau = 1/\lambda$ and $\mathbf{b} = \tau\boldsymbol{\beta}$. (3.72) can admit multiple solutions for each given λ. For example, $\mathbf{b} = \mathbf{0}$ is a local solution to (3.72), when $\lambda \ge \|\mathbf{X}^T\mathbf{Y}/n\|_\infty$. See also (3.33). Unlike Lasso, there can be other local solutions to (3.72). PLUS computes the main branch $\widehat{\boldsymbol{\beta}}(\tau)$ starting from $\widehat{\boldsymbol{\beta}}(\tau) = \mathbf{0}$, where $\tau = 1/\|\mathbf{X}^T\mathbf{Y}/n\|_\infty$.

Let us characterize the solution set of (3.72). The component b_j of a solution $\mathbf{b}$ falls in one of the intervals $\{(t_i, t_{i+1}]\}_{i=1}^m$, or 0, or in one of the intervals $\{[-t_{i+1}, -t_i)\}_{i=1}^m$. Let us use $i_j \in \{-m, \cdots, m\}$ to indicate such an interval and $\mathbf{i} \in \{-m, \cdots, m\}^p$ be the the vector of indicators. Then, by (3.71), (3.72) can be written as

$$\begin{cases} \tau z_j - \boldsymbol{\chi}_j^T\mathbf{b} = \text{sgn}(i_j)(u_{i_j} - b_j v_{i_j}), & \bar{t}_{i_j} \le b_j \le \bar{t}_{i_j+1}, & i_j \neq 0, \\ -1 \le \tau z_j - \boldsymbol{\chi}_j^T\mathbf{b} \le 1, & b_j = 0, & i_j = 0, \end{cases} \tag{3.73}$$

where $u_{-k} = u_k$, $v_{-k} = v_k$, and $\bar{t}_i = t_i$ for $0 < i \le m+1$ and $-t_{|i|+1}$ for $-m \le i \le 0$. Let $\mathcal{S}_\tau(\mathbf{i})$ be the set of $(\tau\mathbf{z}^T, \mathbf{b}^T)^T$ in R^{2p}, whose coordinates satisfy (3.73). Note that the solution $\mathbf{b}$ is piecewise linear τ.

Let $H = R^p$ represent the data $\mathbf{z}$ and its dual $H^* = R^p$ represent the solution $\mathbf{b}$, and $\mathbf{z} \oplus \mathbf{b}$ be members of $H \oplus H^* = R^{2p}$. The set $\mathcal{S}_\tau(\mathbf{i})$ in R^{2p} is more compactly expressed as

$$\mathcal{S}_\tau(\mathbf{i}) = \{\tau\mathbf{z} \oplus \mathbf{b} : \ \tau\mathbf{z} \text{ and } \mathbf{b} \text{ satisfy (3.73)}\}.$$

For each given τ and $\mathbf{i}$, the set $\mathcal{S}_\tau(\mathbf{i})$ is a parallelepiped in R^{2p} and $\mathcal{S}_\tau = \tau\mathcal{S}_1$.

The solution $\mathbf{b}$ is the projection of $\mathcal{S}_\tau(\mathbf{i})$ onto H^*, denoted by $\mathcal{S}_\tau(\mathbf{i}|\mathbf{z})$. Clearly, all solutions to (3.72) is a p-dimensional set given by

$$\mathcal{S}_\tau(\mathbf{z}) = \cup\{\mathcal{S}_\tau(\mathbf{i}|\mathbf{z}) : \mathbf{i} \in \{-m, -m+1, \cdots, m\}^p\}. \tag{3.74}$$

Like LARS, the PLUS algorithm computes a solution $\boldsymbol{\beta}(\tau)$ from $\mathcal{S}_\tau(\mathbf{z})$. Starting from $\tau_0 = 1/\|n^{-1}\mathbf{X}^T\mathbf{Y}\|_\infty$, $\widehat{\boldsymbol{\beta}}(\tau_0) = 0$ and $\mathbf{i} = \mathbf{0}$, PLUS updates the active set of variables as well as the branch $\mathbf{i}$, determines the step size τ and solution $\mathbf{b}$. The solutions between two turning points are connected by lines. We refer to Zhang (2010) for additional details. In addition, Zhang (2010) gives the conditions under which the solution becomes the oracle estimator, derives the risk and model selection properties of the PLUS estimators.

3.5.6 Cyclic coordinate descent algorithms

Consider the sparse penalized least squares, the computation difficulty comes from the nonsmoothness of the penalty function. Observe that the penalty function part is the sum of p univariate nonsmooth functions. Then, we can employ *cyclic coordinate descent algorithms* (Tseng, 2001; Tseng and Yun, 2009) that successively optimize one coefficient (coordinate) at a time. Let

$$L(\beta_1, \ldots, \beta_p) = \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} p_\lambda(|\beta_j|)$$

and the cyclic coordinate descent (CCD) algorithm proceeds as follows:

1. choose an initial value of $\widehat{\boldsymbol{\beta}}$
2. for $j = 1, 2, \ldots, p, 1, 2, \ldots$, update $\widehat{\beta}_j$ by solving a univariate optimization problem of β_j:

$$\widehat{\beta}_j^{\text{update}} \leftarrow \text{argmin}_{\beta_j} L(\widehat{\beta}_1, \ldots, \widehat{\beta}_{j-1}, \beta_j, \widehat{\beta}_{j+1}, \widehat{\beta}_p). \tag{3.75}$$

3. Repeat (2) till convergence.

Let $\mathbf{R}_j = \mathbf{Y} - \mathbf{X}_{-j}\widehat{\boldsymbol{\beta}}_{-j}$ be the current residual, where $\mathbf{X}_{-j}$ and $\widehat{\boldsymbol{\beta}}_{-j}$ are respectively $\mathbf{X}$ and $\widehat{\boldsymbol{\beta}}$ with the j^{th} column and j^{th} component removed. Then, the target function in (3.75) becomes, after ignoring a constant,

$$Q_j(\beta_j) \equiv \frac{1}{2n}\|\mathbf{R}_j - \mathbf{X}_j\beta_j\|^2 + p_\lambda(|\beta_j|),$$

Recall that $\|\mathbf{X}_j\|^2 = n$ by standardization and $\widehat{c}_j = n^{-1}\mathbf{X}_j^T\mathbf{R}_j$ is the current covariance [c.f. (3.60)]. Then, after ignoring a constant,

$$Q_j(\beta_j) = \frac{1}{2}(\beta_j - \widehat{c}_j)^2 + p_\lambda(|\beta_j|). \tag{3.76}$$

This is the same problem as (3.12). For L_1, SCAD and MCP penalty, (3.76)

admits an explicit solution as in (3.18)–(3.20). In this case, the *CCD* algorithm is simply an iterative thresholding method.

The CCD algorithm for the Lasso regression is the same as the shooting algorithm introduced by Fu (1998). Friedman, Hastie, Höfling and Tibshirani (2007) implement the CCD algorithm by using several tricks such as warm start, active set update, etc. As a result, they were able to show that the coordinate descent algorithm is actually very effective in computing the Lasso solution path, proving to be even faster than the LARS algorithm. Fan and Lv (2011) extend the CCD algorithm to the penalized likelihood.

The user needs to be careful when applying the coordinate descent algorithm to solve the concave penalized problems because the algorithm converges to a local minima but this solution may not be the statistical optimal one. The choice of initial value becomes very important. In Fan, Xue and Zou (2014) there are simulation examples showing that the solution by CCD is suboptimal compared with the LLA solution in SCAD and MCP penalized regression and logistic regression. It is beneficial to try multiple initial values when using CCD to solve nonconvex problems.

3.5.7 Iterative shrinkage-thresholding algorithms

The iterative shrinkage-thresholding algorithm (ISTA, Daubechies *et al.*, 2004) is developed to optimize the functions of form $Q(\boldsymbol{\beta}) = f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$, in which f is smooth whereas $g(\boldsymbol{\beta})$ is non-smooth. Note that the *gradient descent algorithm*

$$\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} - s_k f'(\boldsymbol{\beta}_{k-1}),$$

for a suitable stepsize s_k is the minimizer to the *local isotropic quadratic* approximation of f at $\boldsymbol{\beta}_{k-1}$:

$$f_A(\boldsymbol{\beta}|\boldsymbol{\beta}_{k-1}, s_k) = f(\boldsymbol{\beta}_{k-1}) + f'(\boldsymbol{\beta}_{k-1})^T(\boldsymbol{\beta} - \boldsymbol{\beta}_{k-1}) + \frac{1}{2s_k}\|\boldsymbol{\beta} - \boldsymbol{\beta}_{k-1}\|^2. \quad (3.77)$$

The local isotropic approximation avoids computing the Hessian matrix, which is expensive and requires a lot of storage for high-dimensional optimization. Adapting this idea to minimizing $Q(\cdot)$ yields the algorithm

$$\boldsymbol{\beta}_k = \text{argmin}\{f_A(\boldsymbol{\beta}|\boldsymbol{\beta}_{k-1}, s_k) + g(\boldsymbol{\beta})\}.$$

In particular, when $g(\boldsymbol{\beta}) = \sum_{j=1}^p p_\lambda(|\beta_j|)$, the problem becomes a componentwise optimization after ignoring a constant

$$\boldsymbol{\beta}_k = \text{argmin}\left\{\frac{1}{2s_k}\|\boldsymbol{\beta} - (\boldsymbol{\beta}_{k-1} - s_k f'(\boldsymbol{\beta}_{k-1}))\|^2 + \sum_{j=1}^p p_\lambda(|\beta_j|)\right\},$$

for each component of the form (3.12). Let us denote

$$\theta_s(z) = \text{argmin}_\theta \left\{\frac{1}{2}(z-\theta)^2 + sp_\lambda(|\theta|)\right\}.$$

Then, ISTA is to iteratively apply

$$\boldsymbol{\beta}_k = \theta_{s_k}\big(\boldsymbol{\beta}_{k-1} - s_k f'(\boldsymbol{\beta}_{k-1})\big). \tag{3.78}$$

In particular, for the Lasso problem (3.29), the ISTA becomes

$$\boldsymbol{\beta}_k = \big(\boldsymbol{\beta}_{k-1} - s_k n^{-1}\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}_{k-1}) - s_k\lambda\big)_+. \tag{3.79}$$

Similar iterative formulas can be obtained for SCAD and MCP. This kind of algorithm is called a *proximal gradient method* in the optimization literature.

Note that when $\|f'(\boldsymbol{\beta}) - f'(\boldsymbol{\theta})\| \le \|\boldsymbol{\beta} - \boldsymbol{\theta}\|/s_k$ for all $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$, we have $f_A(\boldsymbol{\beta}|\boldsymbol{\beta}_{k-1}, s_k) \ge f(\boldsymbol{\beta})$. This holds when the largest eigenvalue of the Hessian matrix $f''(\boldsymbol{\beta})$ is bounded by $1/s_k$. Therefore, the ISTA algorithm is also a specific implementation of the MM algorithm, when the condition is met.

The above isotropic quadratic majorization requires strong conditions regarding to the function f. Inspecting the proof in (3.69) for the MM algorithm, we indeed do not require majorization but only the *local majorization* $Q(\boldsymbol{\beta}_{\text{new}}) \le Q(\boldsymbol{\beta}_{\text{new}}|\boldsymbol{\beta}_0)$. This can be achieved by using the *backtracking rule* to choose the step size s_k as follows. Take an initial step size $s_0 > 0$, $\delta < 1$, and the initial value $\boldsymbol{\beta}_0$. Find the smallest nonnegative integer i_k such that with $s = \delta^{i_k} s_{k-1}$,

$$Q(\boldsymbol{\beta}_{k,s}) \le Q_A(\boldsymbol{\beta}_{k,s}) \equiv f_A(\boldsymbol{\beta}_{k,s}|\boldsymbol{\beta}_{k-1}, s) + g(\boldsymbol{\beta}_{k,s}), \tag{3.80}$$

where $\boldsymbol{\beta}_{k,s} = \theta_s\big(\boldsymbol{\beta}_{k-1} - sf'(\boldsymbol{\beta}_{k-1})\big)$ is the same as the above with emphasis on its dependence on s. Set $s_k = \delta^{i_k} s_{k-1}$ and compute

$$\boldsymbol{\beta}_k = \theta_{s_k}\big(\boldsymbol{\beta}_{k-1} - s_k f'(\boldsymbol{\beta}_{k-1})\big).$$

Note that the requirement (3.80) is really the local majorization requirement. It can easily hold since $s_k \to 0$ exponentially fast as $i_k \to \infty$. According to (3.69), the sequence of objective values $\{Q(\boldsymbol{\beta}_k)\}$ is non-increasing. The above choice of the step size of s_k can be very small as k gets large. Another possible scheme is to use $s = \delta^{i_k} s_0$ rather than $s = \delta^{i_k} s_{k-1}$ in (3.80) in choosing s_k.

The fast iterative shrinkage-thresholding algorithm (FISTA, Beck and Teboulle, 2009) is proposed to improve the convergence rate of ISTA. It employs Nesterov acceleration idea (Nesterov, 1983). The algorithm runs as follows. Input the step size s such that s^{-1} is the upper bound of the Lipchitz constant of $f'(\cdot)$. Take $\mathbf{x}_1 = \boldsymbol{\beta}_0$ and $t_1 = 1$. Compute iteratively for $k \ge 1$

$$\begin{aligned} \boldsymbol{\beta}_k &= \theta_s(\mathbf{x}_k - sf'(\mathbf{x}_k)), \quad t_{k+1} = (1 + \sqrt{1 + 4t_k^2}\,)/2, \\ \mathbf{x}_{k+1} &= \boldsymbol{\beta}_k + \frac{t_k - 1}{t_{k+1}}(\boldsymbol{\beta}_k - \boldsymbol{\beta}_{k-1}). \end{aligned}$$

The algorithm utilizes a constant "stepsize" s. The backtracking rule can also

be employed to make the algorithm more practical. Beck and Teboulle (2009) show that the FISTA has a quadratic convergence rate whereas the ISTA has only linear convergence rate.

3.5.8 Projected proximal gradient method

Agarwal, Negahban and Wainwright (2012) propose a projected proximal gradient descent algorithm to solve the problem

$$\min_{R(\boldsymbol{\beta})\le c} \{f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})\}. \tag{3.81}$$

Given the current value $\boldsymbol{\beta}_{k-1}$, approximate the smooth function f by isotropic quadratic (3.77). The resulting unconstrained solution is given by (3.78). Now, project $\boldsymbol{\beta}_k$ onto the set $\{\boldsymbol{\beta} : R(\boldsymbol{\beta}) \le c\}$ and continue with the next iteration by taking the projected value as the initial value. When $\|R(\boldsymbol{\beta})\| = \|\boldsymbol{\beta}\|_1$, the projection admits an analytical solution. If $\|\boldsymbol{\beta}_k\|_1 \le c$, then the projection is just itself; otherwise, it is the soft-thresholding at level λ_n so that the constraint $\|\boldsymbol{\beta}_k\|_1 = c$. The threshold level λ_n can be computed as follows: (1) sort $\{|\beta_{k,j}|\}_{j=1}^p$ into $b_1 \ge b_2 \ge \ldots \ge b_p$; (2) find $J = \max\{1 \le j \le p : b_j - (\sum_{r=1}^j b_r - c)/j > 0\}$ and let $\lambda_n = (\sum_{r=1}^J b_j - c)/J$.

3.5.9 ADMM

The *alternating direction method of multipliers* (ADMM) (Douglas and Rachford (1956), Eckstein and Bertsekas (1992)) has a number of successful applications in modern statistical machine learning. Boyd *et al.* (2011) give a comprehensive review on *ADMM*. Solving the Lasso regression problem is a classical application of ADMM. Consider the Lasso penalized least square

$$\min_{\boldsymbol{\beta}} \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\boldsymbol{\beta}\|_1$$

which is equivalent to

$$\min_{\boldsymbol{\beta},\mathbf{z}} \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\mathbf{z}\|_1 \quad \text{subject to} \quad \mathbf{z} = \boldsymbol{\beta}.$$

The augmented Lagrangian is

$$\mathcal{L}_\eta(\boldsymbol{\beta}, \mathbf{z}, \boldsymbol{\theta}) = \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda\|\mathbf{z}\|_1 - \boldsymbol{\theta}^T(\mathbf{z} - \boldsymbol{\beta}) + \frac{\eta}{2}\|\mathbf{z} - \boldsymbol{\beta}\|_2^2,$$

where η can be a fixed positive constant set by the user, e.g. $\eta = 1$. The term $\boldsymbol{\theta}^T(\mathbf{z} - \boldsymbol{\beta})$ is the Lagrange multiplier and the term $\frac{\eta}{2}\|\mathbf{z} - \boldsymbol{\beta}\|_2^2$ is its augmentation. The choice of η can affect the convergence speed. ADMM is an iterative procedure. Let $(\boldsymbol{\beta}^k, \mathbf{z}^k, \boldsymbol{\theta}^k)$ denote the kth iteration of the ADMM algorithm for $k = 0, 1, 2, \ldots$. Then the algorithm proceeds as follows:

$$\begin{aligned}
\boldsymbol{\beta}^{k+1} &= \text{argmin}_{\boldsymbol{\beta}}\, \mathcal{L}_\eta(\boldsymbol{\beta}, \mathbf{z}^k, \boldsymbol{\theta}^k),\\
\mathbf{z}^{k+1} &= \text{argmin}_{\mathbf{z}}\, \mathcal{L}_\eta(\boldsymbol{\beta}^{k+1}, \mathbf{z}, \boldsymbol{\theta}^k),\\
\boldsymbol{\theta}^{k+1} &= \boldsymbol{\theta}^k - (\mathbf{z}^{k+1} - \boldsymbol{\beta}^{k+1}).
\end{aligned}$$

It is easy to see that $\boldsymbol{\beta}^{k+1}$ has a close form expression and $\mathbf{z}^{k+1}$ is obtained by solving p univariate L_1 penalized problems. More specifically, we have

$$\boldsymbol{\beta}^{k+1} = (\mathbf{X}^T\mathbf{X}/n + \eta\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{Y}/n + \eta\mathbf{z}^k - \eta\boldsymbol{\theta}^k),$$
$$z_j^{k+1} = \text{sgn}(\beta_j^{k+1} + \theta_j^k)(|\beta_j^{k+1} + \theta_j^k| - \lambda/\eta), j = 1, \ldots, p.$$

3.5.10 Iterative Local Adaptive Majorization and Minimization

Iterative local adaptive majorization and minimization is an algorithmic approach to solve the folded concave penalized least-squares problem (3.9) or more generally the *penalized quasi-likelihood* of the form:

$$f(\boldsymbol{\beta}) + \sum_{j=1}^{p} p_\lambda(|\beta_j|) \tag{3.82}$$

with both algorithmic and statistical guaranteed, proposed and studied by Fan, Liu, Sun, and Zhang (2018). It combines the local linear approximation (3.70) and the proximal gradient method (3.78) to solve the problem (3.82). More specifically, starting from the initial value $\boldsymbol{\beta}^{(0)} = 0$, we use LLA to case problem (3.82) into the sequence of problems:

$$\widehat{\boldsymbol{\beta}}^{(1)} = \arg\min\Big\{f(\boldsymbol{\beta}) + \sum_{j=1}^{d} \lambda_j^{(0)}|\beta_j|\Big\}, \quad \text{with } \lambda_j^{(0)} = p'_\lambda(|\widehat{\beta}_j^{(0)}|) \tag{3.83}$$

.....

$$\widehat{\boldsymbol{\beta}}^{(t)} = \arg\min\Big\{f(\boldsymbol{\beta}) + \sum_{j=1}^{d} \lambda_j^{(t-1)}|\beta_j|\Big\}, \text{ with } \lambda_j^{(t-1)} = p'_\lambda(|\widehat{\beta}_j^{(t-1)}|) \tag{3.84}$$

Within each problem (3.83) or (3.84) above, we apply proximal gradient method. More specifically, by (3.79), starting from the initial value $\widehat{\boldsymbol{\beta}}_{t,0} = \widehat{\boldsymbol{\beta}}_{t-1}$, the algorithm used to solve (3.84) utilizes the iterations

$$\widehat{\boldsymbol{\beta}}_{t,k} = \big(\widehat{\boldsymbol{\beta}}_{t,k-1} - s_{t,k}n^{-1}\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{t,k-1}) - s_{t,k}\lambda\big)_+, \tag{3.85}$$

for $k = 1, \cdots, k_t$, where the step size is computed by using $s = \delta^{i_k}s_0$ to check (3.80) in choosing s_k. The flowchat of the algorithm can be summarized in Figure 3.9. This algorithmic approach of statistical estimator is called *I-LAMM* by Fan *et al.* (2018).

Note that the problem (3.83) is convex but not strongly convex. It converges only at a sublinear rate. Hence, it takes longer to get to a consistent neighborhood. Once the estimate is in a consistent neighborhood, from step 2 and on, the solutions are sparse and therefore the function (3.84) is strongly convex in this restricted neighborhood and the algorithmic convergence is exponentially fast (at a linear rate). This leads Fan *et al.* (2018) to take

$$
\begin{array}{lllll}
\boldsymbol{\lambda}^{(0)}: & \boldsymbol{\beta}^{(1,0)}=\mathbf{0} \overset{\text{LAMM}}{\Longrightarrow} \boldsymbol{\beta}^{(1,1)} \overset{\text{LAMM}}{\Longrightarrow} \cdots \overset{\text{LAMM}}{\Longrightarrow} \boldsymbol{\beta}^{(1,k_1)}=\widetilde{\boldsymbol{\beta}}^{(1)}, & k_1 \lesssim \varepsilon_c^{-2}; \\
\boldsymbol{\lambda}^{(1)}: & \boldsymbol{\beta}^{(2,0)}=\widetilde{\boldsymbol{\beta}}^{(1)} \overset{\text{LAMM}}{\Longrightarrow} \boldsymbol{\beta}^{(2,1)} \overset{\text{LAMM}}{\Longrightarrow} \cdots \overset{\text{LAMM}}{\Longrightarrow} \boldsymbol{\beta}^{(2,k_2)}=\widetilde{\boldsymbol{\beta}}^{(2)}, & k_2 \lesssim \log(\varepsilon_t^{-1}); \\
\vdots & & \vdots & \vdots \\
\boldsymbol{\lambda}^{(T-1)}: & \boldsymbol{\beta}^{(T,0)}=\widetilde{\boldsymbol{\beta}}^{(T-1)} \overset{\text{LAMM}}{\Longrightarrow} \boldsymbol{\beta}^{(T,1)} \overset{\text{LAMM}}{\Longrightarrow} \cdots \overset{\text{LAMM}}{\Longrightarrow} \boldsymbol{\beta}^{(T,k_T)}=\widetilde{\boldsymbol{\beta}}^{(T)}, & k_T \lesssim \log(\varepsilon_t^{-1}).
\end{array}
$$

Figure 3.9: Flowchart of iterative local majorization and minorization algorithm. For Guassian noise, $\varepsilon_c \asymp \sqrt{n^{-1}\log p}$ and $\varepsilon_t \asymp \sqrt{n^{-1}}$. Taken from Fan, Liu, Sun, and Zhang (2018)

$k_1 \asymp n/\log p$ and $k_2 \asymp \log n$. In addition, they show that when the number of outer loop $T \asymp \log(\log(p))$, the estimator achieves statistical optimal rates and further iteration will not improve nor deteriorate the statistical errors.

3.5.11 Other Methods and Timeline

There are many other algorithms for computing penalized least-squares problem. For example, *matching pursuit*, introduced by Mallot and Zhang (1993), is similar to the forward selection algorithm for subset selection. As in the forward selection and LARS, the most correlated variable $\mathbf{X}_j$ (say) with the current residual $\mathbf{R}$ is selected and the univariate regression

$$\mathbf{R} = \beta_j \mathbf{X}_j + \boldsymbol{\varepsilon}$$

is fitted. This is an important deviation from the forward selection in high-dimensional regression as the matching pursuit does not compute multiple regression. It is similar but more greedy than the coordinate decent algorithm, as only the most correlated coordinate is chosen. With fitted univariate coefficient $\widehat{\beta}_j$, we update the current residual by $\mathbf{R} - \widehat{\beta}_j \mathbf{X}_j$. The variables selected as well as coefficients used to compute $\mathbf{R}$ can be recorded along the fit.

Iterated SIS(sure independence screening) introduced in Fan and Lv (2008) and extended by Fan, Samworth and Wu (2009) can be regarded as another greedy algorithm for computing folded concave PLS. The basic idea is to iteratively use large scale screening (e.g. marginal screening) and moderate scale selection by using the penalized least-squares. Details will be introduced in Chapter 8.

The *DC algorithm* (An and Tao, 1997) is a general algorithm for minimizing the difference of two convex functions. Suppose that $Q(\boldsymbol{\beta}) = Q_1(\boldsymbol{\beta}) - Q_2(\boldsymbol{\beta})$, where Q_1 and Q_2 are convex. Given the current value $\boldsymbol{\beta}_0$, linearize $Q_2(\boldsymbol{\beta})$ by

$$Q_{2,L}(\boldsymbol{\beta}) = Q_2(\boldsymbol{\beta}_0) + Q_2'(\boldsymbol{\beta}_0)^T(\boldsymbol{\beta} - \boldsymbol{\beta}_0).$$

Now update the minimizer by the convex optimization problem

$$\widehat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}}\{Q_1(\boldsymbol{\beta}) - Q_{2,L}(\boldsymbol{\beta})\}.$$

Note that for any convex function

$$Q_2(\boldsymbol{\beta}) \geq Q_{2,L}(\boldsymbol{\beta}) \quad \text{with} \quad Q_2(\boldsymbol{\beta}_0) = Q_{2,L}(\boldsymbol{\beta}_0).$$

Thus, the DC algorithm is a special case of the MM-algorithm. Hence, its target value should be non-increasing $Q(\boldsymbol{\beta}_{\text{new}}) \leq Q(\boldsymbol{\beta}_0)$ [c.f. (3.69)]. The algorithm has been implemented to support vector machine classifications by Liu, Shen and Doss (2005) and Wu and Liu (2007). It was used by Kim, Choi and Oh (2008) to compute SCAD in which the SCAD penalty function is decomposed as

$$p_\lambda(|\beta|) = \lambda|\beta| - [\lambda|\beta| - p_\lambda(|\beta|)].$$

Agarwal, Negahban and Wainwright (2012) propose the composite gradient descent algorithm. Liu, Yao and Li (2016) propose a mixed integer programming-based global optimization (MIPGO) to solve the class of folded concave penalized least-squares that find a provably global optimal solution. Fan, Liu, Sun and Zhang (2018) propose I-LAMM to simultaneously control of algorithmic complexity and statistical error.

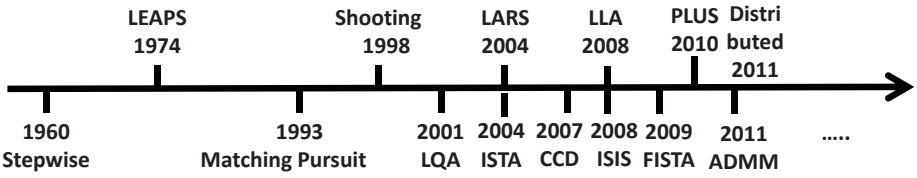

Figure 3.10: A snapshot of the history of the algorithms for computing penalized least-squares.

3.6 Regularization parameters for PLS

In applications of the folded concave PLS (3.9), one needs to determine the regularization parameter λ. The solution paths such as in Figure 3.7 can help us to choose a model. For example, it is not unreasonable to select a model with $1/\lambda$ somewhat larger than 40 in Figure 3.7. After that point, the model complexity increases substantially and there will be no more variables with large coefficients.

In many situations, one would also like to have data-driven choice of λ. The choice of λ for the L_0-penalty was addressed in Section 3.1.3. The basic idea of choosing regularization parameters to minimize the estimated prediction error continues to apply. For example, one can choose λ in the folded concave PLS by using cross-validation (3.5). However, other criteria such as AIC and BIC

utilize the model size m that is specific to L_0-penalty. We need to generalize this concept of model size, which will be called the degrees of freedom.

3.6.1 Degrees of freedom

To help motivate the definition of *degrees of freedom*, following Efron (1986) and Efron *et al.* (2004), we assume that given the covariates $\mathbf{X}$, $\mathbf{Y}$ has the conditional mean vector $\boldsymbol{\mu}(\mathbf{X})$ (also called regression function) that depends on $\mathbf{X}$ and homoscedastic variance σ^2. The conditional mean vector $\boldsymbol{\mu}$ (whose dependence on $\mathbf{X}$ is suppressed) is unknown and estimated by $\widehat{\boldsymbol{\mu}}$, a function of the data $(\mathbf{X}, \mathbf{Y})$. Note that

$$\|\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}}\|^2 = \|\mathbf{Y} - \widehat{\boldsymbol{\mu}}\|^2 - \|\mathbf{Y} - \boldsymbol{\mu}\|^2 + 2(\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu})^T(\mathbf{Y} - \boldsymbol{\mu}). \tag{3.86}$$

Thus, we have *Stein's identity*: the *mean squared error*

$$\mathrm{E}\,\|\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}}\|^2 = \mathrm{E}\left\{\|\mathbf{Y} - \widehat{\boldsymbol{\mu}}\|^2 - n\sigma^2\right\} + 2\sum_{i=1}^{n} \mathrm{cov}(\widehat{\mu}_i, Y_i). \tag{3.87}$$

and the *prediction error*

$$\mathrm{E}\,\|\mathbf{Y}^{\mathrm{new}} - \widehat{\boldsymbol{\mu}}\|^2 = n\sigma^2 + \mathrm{E}\,\|\boldsymbol{\mu} - \widehat{\boldsymbol{\mu}}\|^2 = \mathrm{E}\left\{\|\mathbf{Y} - \widehat{\boldsymbol{\mu}}\|^2\right\} + 2df_{\widehat{\boldsymbol{\mu}}}\sigma^2 \tag{3.88}$$

with

$$df_{\widehat{\boldsymbol{\mu}}} = \sigma^{-2}\sum_{i=1}^{n} \mathrm{cov}(\widehat{\mu}_i, Y_i) \tag{3.89}$$

as the *degrees of freedom*.

If $df_{\widehat{\boldsymbol{\mu}}}$ is known and σ^2 is given, a C_p-type of unbiased risk estimation is given by

$$C_p(\widehat{\boldsymbol{\mu}}) = \|\mathbf{Y} - \widehat{\boldsymbol{\mu}}\|^2 + 2\sigma^2\, df_{\widehat{\boldsymbol{\mu}}}. \tag{3.90}$$

The above formula shows that $df_{\widehat{\boldsymbol{\mu}}}$ plays the same role as the number of parameters in (3.3).

For many linear smoothers, their degrees of freedom are indeed known quantities. A linear estimator has the form $\widehat{\boldsymbol{\mu}} = \mathbf{S}\mathbf{Y}$ with $\mathbf{S}$ being a smoother matrix that only depends on $\mathbf{X}$. See examples given in Section 2.8 of Chapter 2. By independence among $\mathbf{Y}_i$s, $\mathrm{cov}(\widehat{\mu}_i, \mathbf{Y}_i) = \mathbf{S}_{ii}\sigma^2$. From (3.86), it follows that

$$df_{\widehat{\boldsymbol{\mu}}} = \frac{1}{\sigma^2}\sum_{i=1}^{n} \mathbf{S}_{ii}\sigma^2 = \mathrm{tr}(\mathbf{S}).$$

We mentioned $\mathrm{tr}(\mathbf{S})$ as the degrees of freedom of the linear smoother $\mathbf{S}$ in Chapter 2. Here, a formal justification is provided. In particular, when $\mathbf{S} = \mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T$, the projection matrix using m variables of the full model, we have

$$df_{\widehat{\boldsymbol{\mu}}} = \mathrm{tr}(\mathbf{X}_1(\mathbf{X}_1^T\mathbf{X}_1)^{-1}\mathbf{X}_1^T) = m.$$

Therefore, the degrees of freedom formula is an extension of the number of variables used in the classical linear model.

Degrees of freedom can be much more complex for nonlinear model fitting procedures. For example, let us consider the best subset selection. For a given subset size m, the final model always has m variables and one may naively think the degrees of freedom is m. This is in general wrong unless $m = 0$ or $m = p$. This is because the final subset is obtained by exclusively searching over $\binom{p}{m}$ many candidate models. We can not ignore the stochastic nature of the search unless $m = 0$ or $m = p$. A simulation study in Lucas, Fithian and Hastie (2015) shows that the degree of freedom is larger than m and can be even larger than p. Another interesting and counter-intuitive finding is that the degrees of freedom is not a monotonic increasing function of m, which again reflects the complexity due to the stochastic search over $\binom{p}{m}$ many submodels. The same phenomenon is also observed for the degrees of freedom of forward selection.

For least angle regression, Efron *et al.* (2004) show that under the orthogonal design assumption, the degree of freedom in the m^{th} step of the LARS algorithm is m. This matches our intuition, as at the m^{th} step of the LARS algorithm, m variables are effectively recruited. For a general design matrix, let $\widehat{\boldsymbol{\beta}}_\lambda^{\text{lasso}}$ be the Lasso penalized least square estimator with penalization parameter λ. Let $df_\lambda^{\text{lasso}}$ denote its degrees of freedom. Zou, Hastie and Tibshirani (2007) prove a surprising result:

$$df_\lambda^{\text{lasso}} = \mathrm{E}[\|\widehat{\boldsymbol{\beta}}_\lambda^{\text{lasso}}\|_0]. \tag{3.91}$$

Therefore, the number of nonzero estimated coefficients is an exact unbiased estimator of the degrees freedom of the Lasso. The estimation consistency is also established. In theory we view the L_1 PLS as a convex relaxation of L_0 PLS, but their degrees of freedom (model complexity) has very different properties. For the L_0 PLS, the number of nonzero estimated coefficients can severely underestimate the true degrees of freedom. The final model of L_1 PLS is also obtained via a stochastic search, but (3.91) implies that on average the complexity due to stochastic search is zero.

The unbiasedness result is good enough for constructing a C_p type statistic for the Lasso:

$$C_p^{\text{lasso}} = \|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_\lambda^{\text{lasso}}\|^2 + 2\sigma^2\|\widehat{\boldsymbol{\beta}}_\lambda^{\text{lasso}}\|_0, \tag{3.92}$$

which is an exact unbiased estimator of the prediction risk of the Lasso.

3.6.2 Extension of information criteria

Suppose that $\widehat{\boldsymbol{\mu}}(\lambda)$ is constructed by using a regularization parameter λ. An extension of the C_p criterion (3.3) and the information criterion (3.4) is

$$C_p(\lambda) = \|\mathbf{Y} - \widehat{\boldsymbol{\mu}}(\lambda)\|^2 + \gamma\sigma^2\, df_{\widehat{\boldsymbol{\mu}}(\lambda)}, \tag{3.93}$$

and

$$\mathrm{IC}(\lambda) = \log(\|\mathbf{Y} - \widehat{\boldsymbol{\mu}}(\lambda)\|^2/n) + \gamma\, df_{\widehat{\boldsymbol{\mu}}(\lambda)}/n. \tag{3.94}$$

As shown in (3.87), $C_p(\lambda)$ with $\gamma = 2$ is an unbiased estimation of the risk $E\|\widehat{\boldsymbol{\mu}}(\lambda) - \boldsymbol{\mu}\|^2$ except a constant term $-n\sigma^2$. When $\gamma = 2$, $\log(n)$ and $2\log(p)$, the criteria (3.93) will be called respectively the AIC, BIC, and RIC criterion. With an estimate of σ^2 (see Section 3.7), one can choose λ to minimize (3.93). Similarly, we can choose λ to minimize (3.94).

Similarly, one can extend the *generalized cross-validation* criterion (3.7) to this framework by

$$\mathrm{GCV}(\lambda) = \frac{\|\mathbf{Y} - \widehat{\boldsymbol{\mu}}(\lambda)\|^2}{(1 - df_{\widehat{\boldsymbol{\mu}}(\lambda)}/n)^2}. \tag{3.95}$$

In particular, when the linear estimator $\widehat{\boldsymbol{\mu}}(\lambda) = \mathbf{H}(\lambda)\mathbf{Y}$ is used, by (3.90), we have

$$C_p(\lambda) = \|\mathbf{Y} - \widehat{\boldsymbol{\mu}}(\lambda)\|^2 + \gamma\sigma^2\ \mathrm{tr}(\mathbf{H}(\lambda)), \tag{3.96}$$

$$\mathrm{IC}(\lambda) = \log(\|\mathbf{Y} - \widehat{\boldsymbol{\mu}}(\lambda)\|^2/n) + \gamma\,\mathrm{tr}(\mathbf{H}(\lambda))/n. \tag{3.97}$$

and

$$\mathrm{GCV}(\lambda) = \frac{\|\mathbf{Y} - \widehat{\boldsymbol{\mu}}(\lambda)\|^2}{[1 - \mathrm{tr}(\mathbf{H}(\lambda))/n]^2}. \tag{3.98}$$

As mentioned in Section 3.1.3, an advantage of the information criterion and GCV is that no estimation of σ^2 is needed, but this can lead to inaccurate estimation of prediction error.

3.6.3 Application to PLS estimators

For PLS estimator (3.9), $\widehat{\boldsymbol{\mu}}(\lambda)$ is not linear in $\mathbf{Y}$. Some approximations are needed. For example, using the LQA approximation, Fan and Li (2001) regard (3.67) as a linear smoother with (recalling $\boldsymbol{\mu}(\lambda) = \mathbf{X}\widehat{\boldsymbol{\beta}}(\lambda)$)

$$\mathbf{H}(\lambda) = \mathbf{X}(\mathbf{X}^T\mathbf{X} + n\,\mathrm{diag}\{p'_\lambda(\widehat{\beta}_j(\lambda))/|\widehat{\beta}_j(\lambda)|\})^{-1}\mathbf{X}^T,$$

and choose λ by GCV (3.98).

For the LARS-Lasso algorithm, as mentioned at the end of Section 3.6.1, Zou, Hastie and Tibshirani (2007) demonstrate that the degree of freedom is the same as the number of variables used in the LARS algorithm. This motivates Wang, Li and Tsai (2007) and Wang and Leng (2007) to use directly $\|\widehat{\boldsymbol{\beta}}\|_0$ as the degree of freedom. This leads to the definition of modified information criterion as

$$\mathrm{IC}^*(\lambda) = \log(\|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_\lambda\|^2/n) + \gamma\frac{\|\widehat{\boldsymbol{\beta}}_\lambda\|_0}{n}C_n \tag{3.99}$$

for a sequence of constants C_n. It has been shown by Wang, Li and Tsai (2007) that SCAD with AIC ($\gamma = 2$, $C_n = 1$) yields an inconsistent model (too many false positives) while BIC ($\gamma = \log n$, $C_n = 1$) yields a consistent estimation

of the model when p is fixed. See also Wang and Leng (2007) for similar model selection results. Wang, Li and Leng (2009) show that the modified BIC (3.99) with $\gamma = \log n$ and $C_n \to \infty$ produces consistent model selection when SCAD is used. For high-dimensional model selection, Chen and Chen (2008) propose an extended BIC, which adds a multiple of the logarithm of the prior probability of a submodel to BIC. Here, they successfully establish its model selection consistency.

3.7 Residual variance and refitted cross-validation

Estimation of noise variance σ^2 is fundamental in statistical inference. It is prominently featured in the statistical inference of regression coefficients. It is also important for variable selection using the C_p criterion (3.96). It provides a benchmark for forecasting error when an oracle actually knows the underlying regression function. It also arises from genomewise association studies (see Fan, Han and Gu, 2012). In the classical linear model as in Chapter 2, the noise variance is estimated by the residual sum of squares divided by $n - p$. This is not applicable to the high-dimensional situations in which $p > n$. In fact, as demonstrated in Section 1.3.3 (see Figure 1.9 there), the impact of spurious correlation on residual variance estimation can be very large. This leads us to introducing the refitted cross-validation.

In this section, we introduce methods for estimating σ^2 in the high-dimensional framework. Throughout this section, we assume the linear model (2.2) with homoscedastic variance σ^2.

3.7.1 Residual variance of Lasso

A natural estimator of σ^2 is the residual variance of penalized least-squares estimators. As demonstrated in Section 3.3.2, Lasso has a good risk property. We therefore examine when its residual variance gives a consistent estimator of σ^2.

Recall that the *theoretical risk* and *empirical risk* are defined by

$$R(\boldsymbol{\beta}) = \mathrm{E}(Y - \mathbf{X}^T\boldsymbol{\beta})^2 \quad \text{and} \quad R_n(\boldsymbol{\beta}) = n^{-1}\sum_{i=1}^{n}(Y_i - \mathbf{X}_i^T\boldsymbol{\beta})^2,$$

Let $\widehat{\boldsymbol{\beta}}$ be the solution to the Lasso problem (3.39) and c be sufficiently large so that $\|\boldsymbol{\beta}_0\|_1 \le c$. Then, $R_n(\boldsymbol{\beta}_0) \ge R_n(\widehat{\boldsymbol{\beta}})$. Using this, we have

$$\begin{aligned} R(\boldsymbol{\beta}_0) - R_n(\widehat{\boldsymbol{\beta}}) &= [R(\boldsymbol{\beta}_0) - R_n(\boldsymbol{\beta}_0)] + [R_n(\boldsymbol{\beta}_0) - R_n(\widehat{\boldsymbol{\beta}})] \\ &\ge R(\boldsymbol{\beta}_0) - R_n(\boldsymbol{\beta}_0) \\ &\ge -\sup_{\|\boldsymbol{\beta}\|_1 \le c} |R(\boldsymbol{\beta}) - R_n(\boldsymbol{\beta})|. \end{aligned}$$

On the other hand, by using $R(\boldsymbol{\beta}_0) \le R(\widehat{\boldsymbol{\beta}})$, we have

$$R(\boldsymbol{\beta}_0) - R_n(\widehat{\boldsymbol{\beta}}) \le R(\widehat{\boldsymbol{\beta}}) - R_n(\widehat{\boldsymbol{\beta}}) \le \sup_{\|\boldsymbol{\beta}\|_1 \le c} |R(\boldsymbol{\beta}) - R_n(\boldsymbol{\beta})|.$$

Therefore,

$$|R(\boldsymbol{\beta}_0) - R_n(\widehat{\boldsymbol{\beta}})| \le \sup_{\|\boldsymbol{\beta}\|_1 \le c} |R(\boldsymbol{\beta}) - R_n(\boldsymbol{\beta})|.$$

By (3.40), we conclude that

$$|R(\boldsymbol{\beta}_0) - R_n(\widehat{\boldsymbol{\beta}})| \le (1+c)^2 \|\boldsymbol{\Sigma}^* - \mathbf{S}_n^*\|_\infty, \tag{3.100}$$

provided that $\|\boldsymbol{\beta}_0\|_1 \le c$. In other words, the average residual sum of squares of Lasso

$$\widehat{\sigma}^2_{\text{Lasso}} = n^{-1}\|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2$$

provides a consistent estimation of σ^2, if the righthand side of (3.100) goes to zero and $\|\boldsymbol{\beta}_0\|_1 \le c$

$$R(\boldsymbol{\beta}_0) = \sigma^2 \qquad \text{and} \qquad R_n(\widehat{\boldsymbol{\beta}}) = \widehat{\sigma}^2_{\text{Lasso}}.$$

As shown in Chapter 11, $\|\boldsymbol{\Sigma} - \widehat{\mathbf{S}}_n\|_\infty = O_P(\sqrt{(\log p)/n})$ for the data with Gaussian tails. That means that $\widehat{\sigma}^2_{\text{Lasso}}$ is consistent when

$$\|\boldsymbol{\beta}_0\|_1 \le c = o\big((n/\log p)^{1/4}\big). \tag{3.101}$$

Condition (3.101) is actually very restrictive. It requires the number of significantly nonzero components to be an order of magnitude smaller than $(n/\log p)^{1/4}$. Even when that condition holds, $\widehat{\sigma}^2_{\text{Lasso}}$ is only a consistent estimator and can be biased or not optimal. This leads us to consider refitted cross-validation.

3.7.2 Refitted cross-validation

Refitted cross-validation (RCV) was introduced by Fan, Guo and Hao (2012) to deal with the spurious correlation induced by data-driven model selection. In high-dimensional regression models, model selection consistency is very hard to achieve. When some important variables are missed in the selected model, they create a non-negligible bias in estimating σ^2. When spurious variables are selected into the model, they are likely to predict the realized but unobserved noise $\boldsymbol{\varepsilon}$. Hence, the residual variance will seriously underestimate σ^2 as shown in Section 1.3.3.

Note that our observed data follow

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \text{ and } \text{Var}(\boldsymbol{\varepsilon}) = \sigma^2\mathbf{I}_n.$$

Even though we only observe $(\mathbf{X}, \mathbf{Y})$, $\boldsymbol{\varepsilon}$ is a realized vector in R^n. It can have a spurious correlation with a subgroup of variables $\mathbf{X}_{\mathcal{S}}$, namely, there exists a vector $\boldsymbol{\beta}_{\mathcal{S}}$ such that $\mathbf{X}_{\mathcal{S}}\boldsymbol{\beta}_{\mathcal{S}}$ and $\boldsymbol{\varepsilon}$ are highly correlated. This can occur easily when the number of predictors p is large as shown in Section 1.3.3. In this case, $\mathbf{X}_{\mathcal{S}}$ can be seen by a model selection technique as important variables. A way to validate the model is to collect new data to see whether the variables in the

set $\mathcal{S}$ still correlated highly with the newly observed Y. But this is infeasible in many studies and is often replaced by the data splitting technique.

RCV splits data evenly at random into two halves, where we use the first half of the data along with a model selection technique to get a submodel. Then, we fit this submodel to the second half of the data using the ordinary least-squares and get the residual variance. Next we switch the role of the first and the second half of the data and take the average of the two residual variance estimates. The idea differs importantly from cross-validation in that the refitting in the second stage reduces the influence of the spurious variables selected in the first stage.

We now describe the procedure in detail. Let datasets $(\mathbf{Y}^{(1)}, \mathbf{X}^{(1)})$ and $(\mathbf{Y}^{(2)}, \mathbf{X}^{(2)})$ be two randomly split data. Let $\widehat{\mathcal{S}}_1$ be the set of selected variables using data $(\mathbf{Y}^{(1)}, \mathbf{X}^{(1)})$. The variance σ^2 is then estimated by the residual variance of the least-squares estimate using the second dataset along with variables in $\widehat{\mathcal{S}}_1$ (only the selected model, not the data, from the first stage is carried to the fit in the second stage), namely,

$$\widehat{\sigma}_1^2 = \frac{(\mathbf{Y}^{(2)})^T(\mathbf{I}_{n/2} - \mathbf{P}^{(2)}_{\widehat{\mathcal{S}}_1})\mathbf{Y}^{(2)}}{n/2 - |\widehat{\mathcal{S}}_1|}, \qquad \mathbf{P}^{(2)}_{\widehat{\mathcal{S}}_1} = \mathbf{X}^{(2)}_{\widehat{\mathcal{S}}_1}(\mathbf{X}^{(2)T}_{\widehat{\mathcal{S}}_1}\mathbf{X}^{(2)}_{\widehat{\mathcal{S}}_1})^{-1}\mathbf{X}^{(2)T}_{\widehat{\mathcal{S}}_1}. \tag{3.102}$$

Compare residual variance estimation in (2.7). Switching the role of the first and second half, we get a second estimate

$$\widehat{\sigma}_2^2 = \frac{(\mathbf{Y}^{(1)})^T(\mathbf{I}_{n/2} - \mathbf{P}^{(1)}_{\widehat{\mathcal{S}}_2})\mathbf{Y}^{(1)}}{n/2 - |\widehat{\mathcal{S}}_2|}.$$

We define the final estimator as the simple average

$$\widehat{\sigma}^2_{\mathrm{RCV}} = (\widehat{\sigma}_1^2 + \widehat{\sigma}_2^2)/2,$$

or the weighted average defined by

$$\widehat{\sigma}^2_{\mathrm{wRCV}} = \frac{(\mathbf{Y}^{(2)})^T(\mathbf{I}_{n/2} - \mathbf{P}^{(2)}_{\widehat{\mathcal{S}}_1})\mathbf{Y}^{(2)} + (\mathbf{Y}^{(1)})^T(\mathbf{I}_{n/2} - \mathbf{P}^{(1)}_{\widehat{\mathcal{S}}_2})\mathbf{Y}^{(1)}}{n - |\widehat{\mathcal{S}}_1| - |\widehat{\mathcal{S}}_2|}. \tag{3.103}$$

The latter takes into account the degrees of freedom used in fitting the linear model in the second stage. We can now randomly divide the data multiple times and take the average of the resulting RCV estimates.

The point of refitting is that even though $\widehat{\mathcal{S}}_1$ may contain some unimportant variables that are highly correlated with $\boldsymbol{\varepsilon}^{(1)}$, they play minor roles in estimating σ^2 in the second stage since they are unrelated with the realized noise vector $\boldsymbol{\varepsilon}^{(2)}$ in the second half of data set. Furthermore, even when some important variables are missed in $\widehat{\mathcal{S}}_1$, they still have a good chance of being well approximated by the other variables in $\widehat{\mathcal{S}}_1$. Thanks to the refitting in the second stage, the best linear approximation of those selected variables is used to reduce the biases in (3.102).

Unlike cross-validation, the second half of data also plays an important role in fitting. Therefore, its size can not be too small. For example, it should be bigger than $|\widehat{\mathcal{S}}_1|$. Yet, larger set $\mathcal{S}_1$ gives a better chance of sure screening (no false negatives) and hence reduces the bias of estimator (3.102). RCV is applicable to any variable selection rule, including the marginal screening procedure in Chapter 8. Fan, Guo and Hao (2012) show that under some mild conditions, the method yields an asymptotic efficient estimator of $\widehat{\sigma}^2$. In particular, it can handle intrinsic model size $s = o(n)$, much higher than (3.101), when folded concave PLS is used. They verified the theoretical results by numerous simulations. See Section 8.7 for further developments.

3.8 Extensions to Nonparametric Modeling

The fundamental ideas of the penalized least squares can be easily extended to the more flexible *nonparametric models*. This section illustrates the versatility of high-dimensional linear techniques.

3.8.1 Structured nonparametric models

A popular modeling strategy is the *generalized additive model* (GAM):

$$Y = \mu + f_1(X_1) + \cdots + f_p(X_p) + \varepsilon, \tag{3.104}$$

This model was introduced by Stone (1985) to deal with the "*curse of dimensionality*" in multivariate nonparametric modeling and was thoroughly treated in the book by Hastie and Tibshirani (1990). A simple way to fit the additive model (3.104) is to expand the regression function $f_j(x)$ into a basis:

$$f_j(x) = \sum_{k=1}^{K_j} \beta_{jk} B_{jk}(x), \tag{3.105}$$

where $\{B_{jk}(x)\}_{k=1}^{K_j}$ are the basis functions (e.g. a B-spline basis with certain number of knots) for variable X_j. See Section 2.5.2. Substituting the expansion into (3.104) yields

$$\begin{aligned} Y &= \mu + \{\beta_{1,1}B_{1,1}(X_1) + \cdots + \beta_{1,K_1}B_{1,K_1}(X_1)\} + \cdots \\ &\quad + \{\beta_{p,1}B_{p,1}(X_p) + \cdots + \beta_{p,K_p}B_{p,K_p}(X_p)\} + \varepsilon. \end{aligned} \tag{3.106}$$

Treating the basis functions $\{B_{j,k}(X_j) : k = 1, \cdots, K_j, j = 1, \cdots, p\}$ as predictors, (3.106) is a high-dimensional linear model. By imposing a sparsity assumption, we assume that only a few f_j functions actually enter the model. So, many β coefficients are zero. Therefore, we can employ penalized folded concave PLS (3.9) to solve this problem. Another selection method is via the *group penalization*. See, for example, the PLASM algorithm in Baskin (1999) and the SpAM algorithm in Ravikumar, Liu, Lafferty and Wasserman (2007).

The *varying-coefficient model* is another widely-used nonparametric extension of the multiple linear regression model. Conditioning on an exposure variable U, the response and covariates follow a linear model. In other words,

$$Y = \beta_0(U) + \beta_1(U)X_1 + \cdots + \beta_p(U)X_p + \varepsilon. \tag{3.107}$$

The model allows regression coefficients to vary with the level of exposure U, which is an observed covariate variable such as age, time, or gene expression. For a survey and various applications, see Fan and Zhang (2008). Expanding the coefficient functions similar to (3.105), we can write

$$Y = \sum_{j=0}^{p}\{\sum_{k=1}^{K_j} \beta_{j,k}B_{j,k}(U)X_j\} + \varepsilon, \tag{3.108}$$

where $X_0 = 1$. By regarding variables $\{B_{j,k}(U)X_j, k = 1, \cdots, K_j, j = 0, \cdots, p\}$ as new predictors, model (3.108) is a high-dimensional linear model. The sparsity assumption says that only a few variables should be in the model (3.107) which implies many zero β coefficients in (3.108). Again, we can employ penalized folded concave PLS (3.9) to do variable selection or use the group selection method.

3.8.2 Group penalty

The penalized least-squares estimate to the nonparametric models in Section 3.8 results in term-by-term selection of the basis functions. In theory, when the folded concave penalty is employed, the selection should be fine. On the other hand, the term-by-term selection does not fully utilize the sparsity assumption of the functions. In both additive model and varying coefficient model, a zero function implies that the whole group of its associated coefficients in the basis expansion is zero. Therefore, model selection techniques should ideally keep or kill a group of coefficients at the same time.

Group penalty was proposed in Antoniadis and Fan (2001, page 966) to keep or kill a block of wavelets coefficients. It was employed by Lin and Zhang (2006) for component selection in smoothing spline regression models, including the additive model as a special case. Their COSSO algorithm iterates between a smoothing spline fit and a non-negative garrote shrinkage and selection. A special case of COSSO becomes a more familiar group lasso regression formulation considered in Yuan and Lin (2006) who named the group penalty *group-lasso*.

Let $\{\mathbf{x}_j\}_{j=1}^p$ be p groups of variables, each consisting of K_j variables. Consider a generic linear model

$$Y = \sum_{j=1}^{p} \mathbf{x}_j^T\boldsymbol{\beta}_j + \varepsilon. \tag{3.109}$$

Two examples of (3.109) are (3.106) and (3.108) in which $\mathbf{x}_j$ represents K_j

spline bases and $\boldsymbol{\beta}_j$ represents their associated coefficients. In matrix form, the observed data based on a sample of size n follow the model

$$\mathbf{Y} = \sum_{j=1}^{p} \mathbf{X}_j \boldsymbol{\beta}_j + \boldsymbol{\varepsilon}, \tag{3.110}$$

where $\mathbf{X}_j$ is $n \times K_j$ design matrix of variables $\mathbf{x}_j$.

The *group penalized least-squares* is to minimize

$$\frac{1}{2n}\|\mathbf{Y} - \sum_{j=1}^{p} \mathbf{X}_j \boldsymbol{\beta}_j\|^2 + \sum_{j=1}^{p} p_\lambda(\|\boldsymbol{\beta}_j\|_{W_j}) \tag{3.111}$$

where $p_\lambda(\cdot)$ is a penalty function and

$$\|\boldsymbol{\beta}_j\|_{W_j} = \sqrt{\boldsymbol{\beta}_j^T \mathbf{W}_j \boldsymbol{\beta}_j}$$

is a generalized norm with a semi-definite matrix $\mathbf{W}_j$. In many applications, one takes $\mathbf{W}_j = \mathbf{I}_{K_j}$, resulting in

$$\frac{1}{2n}\|\mathbf{Y} - \sum_{j=1}^{p} \mathbf{X}_j \boldsymbol{\beta}_j\|^2 + \sum_{j=1}^{p} p_\lambda(\|\boldsymbol{\beta}_j\|). \tag{3.112}$$

For example, the group lasso is defined as

$$\frac{1}{2n}\|\mathbf{Y} - \sum_{j=1}^{p} \mathbf{X}_j \boldsymbol{\beta}_j\|^2 + \lambda \sum_{j=1}^{p} K_j^{1/2} \|\boldsymbol{\beta}_j\|. \tag{3.113}$$

The extra factor $K_j^{1/2}$ is included to balance the impact of group size.

The group-lasso (3.113) was proposed by Baskin (1999) for variable selection in the additive model. Turlach, Venables and Wright (2005) also used the group-lasso for simultaneous variable selection in multiple responses linear regression, a example of multi-task learning.

Assuming a group-wise orthogonality condition, that is, $\mathbf{X}_j^T \mathbf{X}_j = n\mathbf{I}_{K_j}$ for all j, Yuan and Lin (2006) used a group descent algorithm to solve (3.112). Similar to coordinate descent, we update the estimate one group at a time. Consider the coefficients of group j while holding all other coefficients fixed. Then, by $\mathbf{X}_j^T \mathbf{X}_j = n\mathbf{I}_{K_j}$ (3.112) can be written as

$$\frac{1}{2n}\|\mathbf{Y}_{-j} - \mathbf{X}_j \widehat{\boldsymbol{\beta}}_{-j}\|^2 + \frac{1}{2}\|\widehat{\boldsymbol{\beta}}_{-j} - \boldsymbol{\beta}_j\|^2 + \sum_{k=1}^{p} p_\lambda(\|\boldsymbol{\beta}_k\|), \tag{3.114}$$

where $\mathbf{Y}_{-j} = \mathbf{Y} - \sum_{k \neq j} \mathbf{X}_k \boldsymbol{\beta}_k$ and $\widehat{\boldsymbol{\beta}}_{-j} = n^{-1}\mathbf{X}_j^T \mathbf{Y}_{-j}$. This problem was solved by Antoniadis and Fan (2001, page 966). They observed that

$$\min_{\boldsymbol{\beta}_j} \frac{1}{2}\|\widehat{\boldsymbol{\beta}}_{-j} - \boldsymbol{\beta}_j\|^2 + p_\lambda(\|\boldsymbol{\beta}_j\|) = \min_r \left\{ \frac{1}{2} \min_{\|\boldsymbol{\beta}_j\|=r} \|\widehat{\boldsymbol{\beta}}_{-j} - \boldsymbol{\beta}_j\|^2 + p_\lambda(r) \right\}. \tag{3.115}$$

The inner bracket is minimized at $\widehat{\boldsymbol{\beta}}_{j,r} = r\widehat{\boldsymbol{\beta}}_{-j}/\|\widehat{\boldsymbol{\beta}}_{-j}\|$. Substituting this into (3.115), the problem becomes

$$\min_r \left\{ \frac{1}{2}(\|\widehat{\boldsymbol{\beta}}_{-j}\| - r)^2 + p_\lambda(r) \right\}. \tag{3.116}$$

Problem (3.116) is identical to problem (3.12), whose solution is denoted by $\widehat{\theta}(\|\widehat{\boldsymbol{\beta}}_{-j}\|)$. For the L_1, SCAD and MCP, the explicit solutions are given respectively by (3.18)–(3.20). With this notation, we have

$$\widehat{\boldsymbol{\beta}}_j = \frac{\widehat{\theta}(\|\widehat{\boldsymbol{\beta}}_{-j}\|)}{\|\widehat{\boldsymbol{\beta}}_{-j}\|}\widehat{\boldsymbol{\beta}}_{-j}. \tag{3.117}$$

In particular, for the L_1-penalty,

$$\widehat{\boldsymbol{\beta}}_j = \left(1 - \frac{\lambda}{\|\widehat{\boldsymbol{\beta}}_{-j}\|}\right)_+ \widehat{\boldsymbol{\beta}}_{-j},$$

and for the hard-thresholding penalty

$$\widehat{\boldsymbol{\beta}}_j = I(\|\widehat{\boldsymbol{\beta}}_{-j}\| \geq \lambda)\widehat{\boldsymbol{\beta}}_{-j}.$$

These formulas were given by Antoniadis and Fan (2001, page 966). They clearly show that the strength of the group estimates is pulled together to decide whether or not to keep a group of coefficients.

The groupwise orthogonality condition is in fact not natural and necessary to consider. Suppose that the condition holds for the data $(\mathbf{Y}_i, \mathbf{X}_i), 1 \leq i \leq n$. If we bootstrap the data or do cross-validation to selection λ, the groupwise orthogonality condition easily fails on the perturbed dataset. For computational considerations, the groupwise orthogonality condition is not needed for using the group descent algorithm. Several algorithms for solving the Lasso regression, such as ISTA, FISTA and ADMM, can be readily used to solve the group-lasso regression with a general design matrix. We omit the details here.

3.9 Applications

We now illustrate high-dimensional statistical modeling using the monthly house price appreciations (HPA) for 352 counties in the United States. The housing price appreciation is computed based on monthly repeated sales. These 352 counties have the largest repeated sales and hence their measurements are more reliable. The spatial correlations of these 352 HPAs, based on the data in the period from January 2000 to December 2009, are presented in Figure 1.4.

To take advantage of the spatial correlation in their prediction, Fan, Lv and Qi (2011) utilize the following high-dimensional time-series regression. Let Y_t^i be the HPA in county i at time t and $\mathbf{X}_{i,t}$ be the observable factors

that drive the market. In the application below, $\mathbf{X}_{i,t}$ will be taken as the national HPA, the returns of the national house price index that drives the overall housing markets. They used the following s-period ahead county-level forecast model:

$$Y_{t+s}^i = \sum_{j=1}^{p} b_{ij} Y_t^j + \mathbf{X}_{i,t}^T \boldsymbol{\beta}_i + \varepsilon_{t+s}^i, \quad i = 1, \ldots, p, \tag{3.118}$$

where $p = 352$ and b_{ij} and $\boldsymbol{\beta}_i$ are regression coefficients. In this model, we allow neighboring HPAs to influence the future housing price, but we do not know which counties have such prediction power. This leads to the following PLS problem: For each given county i,

$$\min_{\{b_{ij}, j=1,\ldots,p, \boldsymbol{\beta}_i\}} \sum_{t=1}^{T-s} \Big(Y_{t+s}^i - \mathbf{X}_{i,t}^T \boldsymbol{\beta}_i - \sum_{j=1}^{p} b_{ij} Y_t^j \Big)^2 + \sum_{j=1}^{p} w_{ij} p_\lambda(|b_{ij}|),$$

where the weights w_{ij} are chosen according to the geographical distances between counties i and j. The weights are used to discourage HPAs from remote regions from being used in the prediction. The non-vanishing coefficients represent the selected neighbors that are useful for predicting HPA at county i.

Monthly HPA data from January 2000 to December 2009 were used to fit model (3.118) for each county with $s = 1$. The top panel of Figure 3.11 highlights the selected neighborhood HPAs used in the prediction. For each county i, only 3-4 neighboring counties are chosen on average, which is reasonable. Figure 3.11 (bottom left) presents the spatial correlations of the residuals using model (3.118). No pattern can be found, which indicates the spatial correlations have already been explained by the neighborhood HPAs. In contrast, if we ignore the neighborhood selection (namely, setting $b_{ij} = 0, \forall i \neq j$), which is a lower-dimensional problem and will be referred to as the OLS estimate, the spatial correlations of the residuals are visible (bottom right). This provides additional evidence on the effectiveness of the neighborhood selection by PLS.

We now compare the forecasting power of the PLS with OLS. Training sample covers the data for 2000.1-2005.12, and the test period is 2006.1-2009.12. Fan, Lv and Qi (2011) carried out prediction throughout next 3 years in the following manner. For the short-term prediction horizons s from 1 to 6 months, each month is predicted separately using model (3.118); for the time horizon of 7-36 months, only the average HPA over 6-month periods (e.g. months 7-12, 13-18, etc) is predicted. This increases the stability of the prediction. More precisely, for each of the 6 consecutive months (e.g. months 13-18), they obtained a forecast of average HPA during the 6 months using PLS with historical 6-month average HPAs as a training sample. They treated the (annualized) 6-month average as forecast of the middle month of the 6-month period (e.g. month 15.5) and linearly interpolated the months in

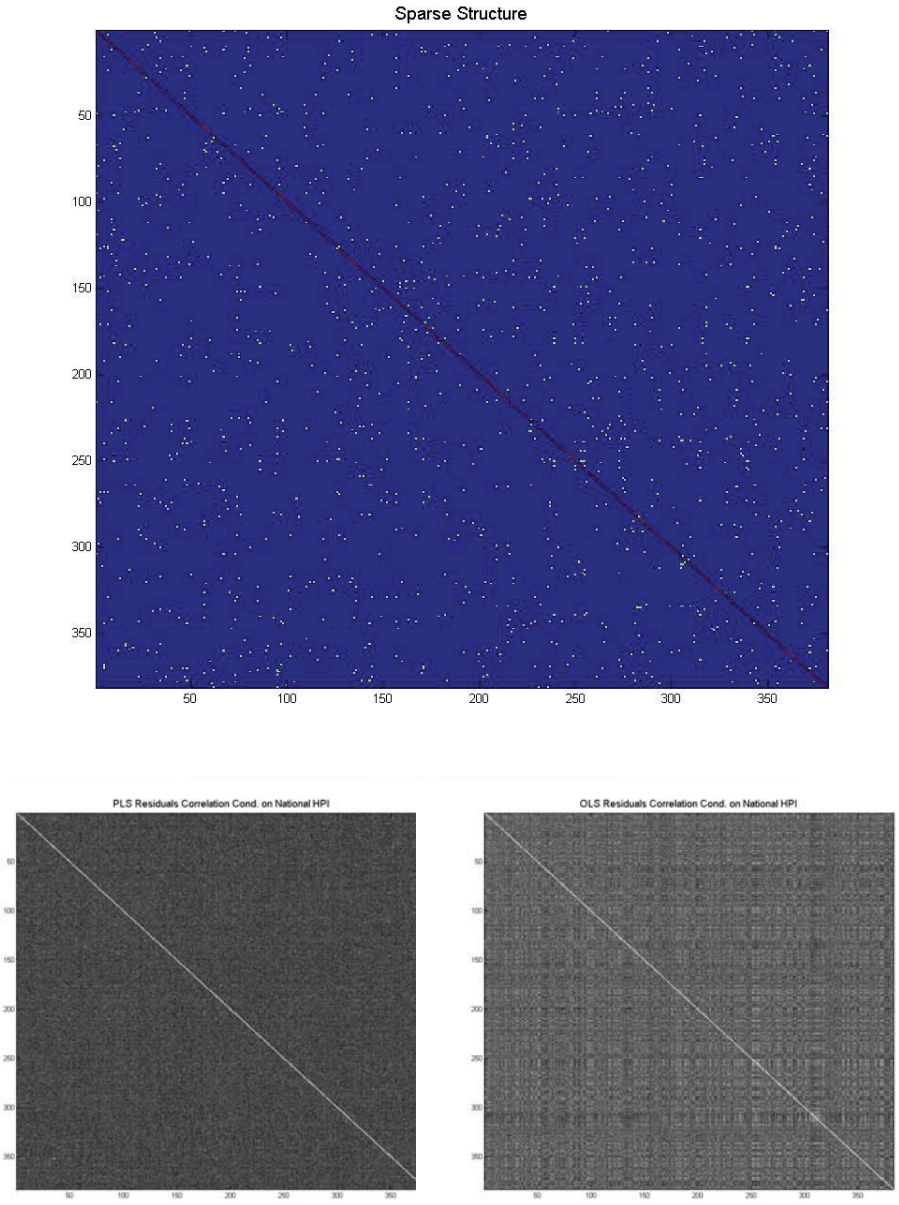

Figure 3.11: Top panel: Neighborhoods with non-zero regression coefficients: for each county i in the y-axis, each row, i.e. x-axis indicates the neighborhood that has impact on the HPA in county i. Bottom left: Spatial-correlation of residuals with national HPA and neighborhood selection. Bottom right: Spatial-correlation of residuals using only national HPA as the predictor. Adapted from Fan, Lv and Qi (2011).

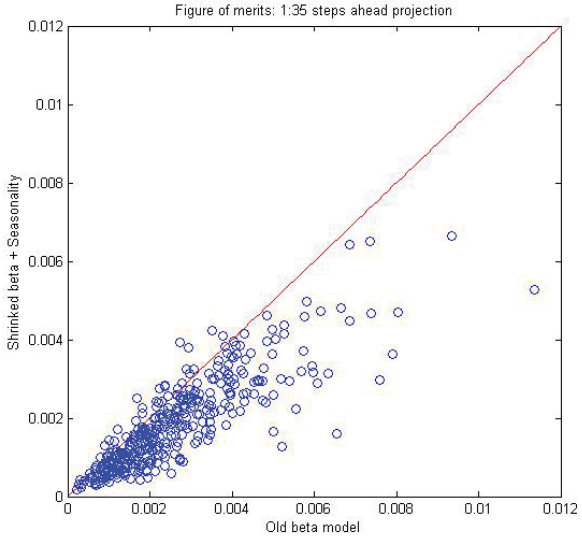

Figure 3.12: Aggregated forecast errors (3.119) in 36 months over 352 counties. For each dot, the x-axis represents prediction errors by OLS with only national factor, y-axis error by PLS with additional neighborhood information. The line indicates both methods having the same performance. From Fan, Lv and Qi (2011).

between. The discounted aggregated squared errors were used as a measure of overall performance of the prediction for county i:

$$\text{Forecast Error}_i = \sum_{s=1}^{\tau} \rho^s (\widehat{Y}^i_{T+s} - Y^i_{T+s})^2, \quad \rho = 0.95, \tag{3.119}$$

where τ is the time horizon to be predicted.

The results in Figure 3.12 show that over 352 counties, the sparse regression model (3.118) with neighborhood information performs on average 30% better in terms of prediction error than the model without using the neighborhood information. Figure 1.4 compares forecasts using OLS with only the national HPA (blue) and PLS with additional neighborhood information (red) for the largest counties with the historical HPAs (black).

How good is a prediction method? The residual standard deviation σ provides a benchmark measure when the ideal prediction rule is used. To illustrate this, we estimate σ for one-step forecast in San Francisco and Los Angeles, using the HPA data from January 1998 to December 2005 (96 months). The RCV estimates, as a function of the selected model size s, are shown in Figure 3.13. The naive estimates, which compute directly the residual variances, de-

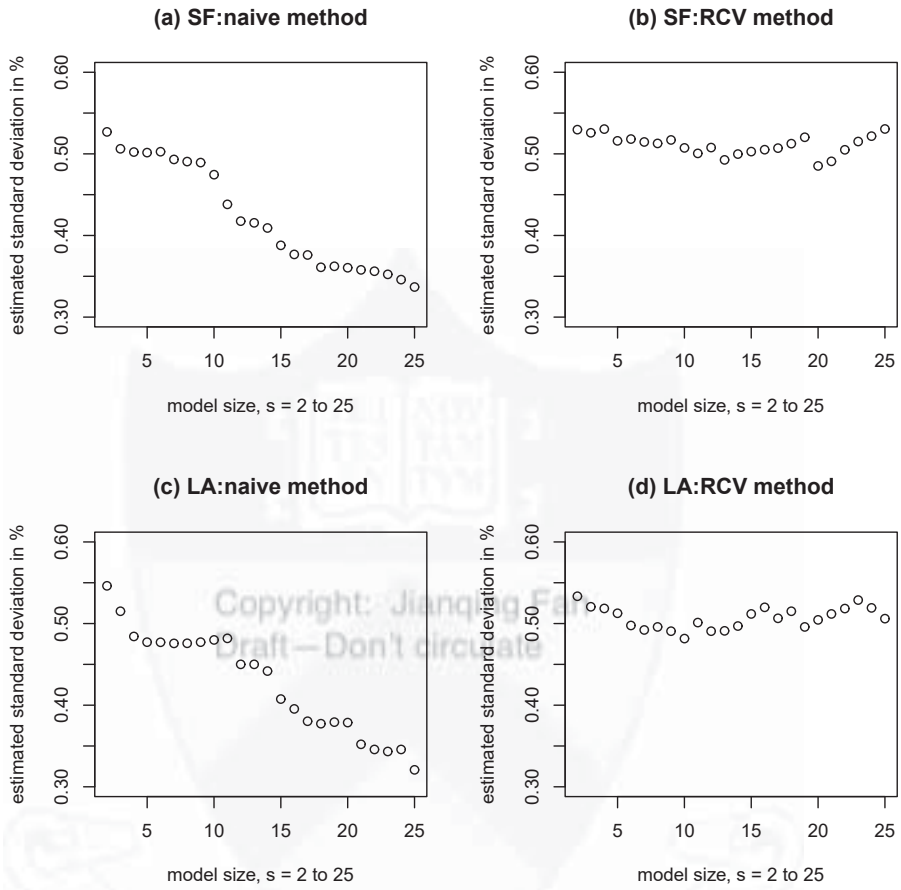

Figure 3.13: Estimated standard deviation σ for one-step ahead forecast as a function of selected model size s in both San Francisco (top panel) and Los Angeles (bottom panel) using both naive (left panel) and RCV (right panel) methods. Taken from Fan, Lv and Qi (2011).

crease with s due to spurious correlation. On the other hand, the RCV gives reasonably stable estimates for a range of selected models. The benchmarks of prediction errors for both San Francisco and Los Angeles regions are about .53%, comparing the standard deviations of month over month variations of HPAs 1.08% and 1.69%, respectively. In contrast, the rolling one-step prediction errors over 12 months in 2006 are .67% and .86% for San Francisco and Los Angeles areas, respectively. They are clearly larger than the benchmark as

expected, but considerably smaller than the standard deviations, which used no variables to forecast. They also show that some small room of improvements is the PLS are possible.

3.10 Bibliographical notes

There are many other exciting developments on variable selection. We have no intention to give a comprehensive survey here. Instead, we focus only on some important inventions on penalized least-squares that lead to the vast literature today.

The idea of L_2-regularization appears in the early work of Tikhonov regularization in the 1940's (Tikhonov, 1943). It was introduced to statistics as ridge regression by Hoerl (1962) and Hoerl and Kennard (1970). The concept of sparsity and L_1 penalty appeared in a series of the work by David Donoho and Iain Johnstone (see e.g. Donoho and Johnstone, 1994). Penalized L_1-regression was employed by David Donoho and Shaobing Chen, in a technical report on "basis pursuit" in 1994, to select a basis from an over complete dictionary. It was then used by Tibshirani (1996) to conduct variable selection. Fan and Li (2001) introduced folded concave penalized likelihood including least-squares to reduce the biases in the Lasso shrinkage and for better variable selection consistency. They introduced local quadratic approximation to cast the optimization problem into a sequence of a quadratic optimization problems and established the oracle property. LARS was introduced by Efron *et al.* (2004) to efficiently compute the Lasso path. An early work on the asymptotic study of penalized least-squares (indeed, penalized likelihood) with diverging dimensionality was given by Fan and Peng (2004). Zou and Zhang (2009) introduced the adaptive elastic net and studied its properties under diverging dimensions. Zhao and Yu (2006), Meinshausen and Bühlmann (2006) and Zou (2006) gave irrepresentable conditions for model selection consistency of Lasso. Candés and Tao (2007) proposed the Dantzig selector, which can be cast as a linear program. Zhang (2010) introduced the PLUS algorithm for computing a solution path of a specific class of folded concave PLS including MCP and SCAD and established a strong oracle property. A family of folded concave penalties that bridge the L_0 and L_1 penalties was studied by Lv and Fan (2009). A thorough investigation on the properties of folded concave PLS when dimensionality diverges was given by Lv and Fan (2010). Meinshausen and Bühlmann (2010) proposed stability selection based on subsampling.

Belloni, Chernozhukov, and Wang (2011) proposed square-root Lasso for sparse regression. Negahban, Ravikumar, Wainwright and Yu (2012) proposed a unified analysis of high-dimensional M-estimators with decomposable regularizers. Agarwal, Negahban and Wainwright (2012) proposed the composite gradient descent algorithm and developed the sampling properties by taking computational error into consideration. Belloni, Chen, Chernozhukov, and Hansen (2012) investigated optimal instruments selection. The focussed GMM was proposed by Fan and Liao (2014) to solve endogeneity problems pandemic

in high-dimensional sparse regression. Belloni and Chernozhukov (2013) investigated post-model selection estimators. Fan, Xue and Zou (2014) showed that the one-step LLA algorithm produces a strong oracle solution as long as the problem is localizable and regular. Loh and Wainwright (2014) developed statistical and algorithmic theory for local optima of regularized M-estimators with nonconvexity penalty. They showed surprisingly that all local optima will lie within statistical precision of the sparse true parameter vector. Support recovery without incoherence was investigated by Loh and Wainwright (2017) using nonconvex regularization.

There were many developments on robust regularization methods and quantile regression. For fixed dimensionality variable selection, see, for example, Wang, Li and Jiang (2007), Li and Zhu (2008), Zou and Yuan (2008), and Wu and Liu (2009). The penalized composite likelihood method was proposed in Bradic, Fan, and Wang (2011) for improvement of the efficiency of Lasso in high dimensions. Belloni and Chernozhukov (2011) showed that the L_1-penalized quantile regression admits the near-oracle rate and derived bounds on the size of the selected model, uniformly in a compact set of quantile indices. Bounds on the prediction error were derived in van de Geer and Müller (2012) for a large class of L_1 penalized estimators, including quantile regression. Wang, Wu and Li (2012) showed that the oracle estimate belongs to the set of local minima of the nonconvex penalized quantile regression. Fan, Fan and Barut (2014) proposed and studied the adaptive robust variable selection. Fan, Li, and Wang (2017) considered estimating high-dimensional mean regression using adaptive Huber loss, which assumes only the second moment condition on the error distribution. Sun, Zhou, and Fan (2017) weakened the second moment condition to $(1+\delta)$-moment and unveiled optimality and phase transition for the adaptive Lasso. Loh (2017) investigated theoretical properties of regularized robust M-estimators, applicable for data contaminated by heavy-tailed distributions and/or outliers in the noises and covariates.

3.11 Exercises

3.1 Let $g(\theta|z,\lambda) = \frac{1}{2}(z-\theta)^2 + p_\lambda(|\theta|)$ for a given λ and z, and denote $\widehat{\theta}(z|\lambda) = \text{argmin}_\theta\, g(\theta|z,\lambda)$.

(a) Let $p_\lambda(|\theta|) = \frac{\lambda^2}{2} I(|\theta| \neq 0)$, the L_0-penalty. Show that $\widehat{\theta}_H(z|\lambda) = zI(|z| \geq \lambda)$, the hard thresholding rule.

(b) Let $p_\lambda(|\theta|) = \frac{1}{2}\lambda^2 - \frac{1}{2}(\lambda - \theta)_+^2$, the hard-thresholding penalty defined in (3.15). Show that $\widehat{\theta}_H(z|\lambda) = zI(|z| \geq \lambda)$. This implies that different penalty functions may result in the same penalized least squares solution.

(c) Comment the advantages of the hard-thresholding penalty over the L_0-penalty.

(d) Let $p_\lambda(|\theta|) = \lambda|\theta|$, the L_1-penalty. Show that $\widehat{\theta}_S(z|\lambda) = \text{sgn}(z)(|z| - \lambda)_+$,

the soft-thresholding rule. Compare with L_0-penalty, the regularization parameter λs in different penalty functions may be in different scale.

(e) Let $p_\lambda(\theta) = \lambda\{(1-\alpha)\theta^2 + \alpha|\theta|\}$ with a fixed α. Derive the close-form solution of $\widehat{\theta}(z|\lambda)$, which is the elastic net thresholding rule.

3.2 Let $g(\theta|z,\lambda) = \frac{1}{2}(z-\theta)^2 + p_\lambda(|\theta|)$ for a given λ and z, and denote $\widehat{\theta}(z|\lambda) = \text{argmin}_\theta\, g(\theta|z,\lambda)$. Following the convention, let $p'_\lambda(0) = p'_\lambda(0+)$. Assume that $p_\lambda(\theta)$ is nondescreasing and continuously differentiable on $[0,\infty)$, and the function $-\theta - p'_\lambda(\theta)$ is strictly unimodal on $[0,\infty)$.

(a) Show that if $t_0 = \min_{\theta\geq 0}\{\theta + p'_\lambda(\theta)\} > 0$, then $\widehat{\theta}(z|\lambda) = 0$ when $z \leq t_0$. This leads the sparsity of $\widehat{\theta}(z|\lambda)$.

(b) Show that if $p'_\lambda(|\theta|) = 0$ for $|\theta| \geq t_1$, then $\widehat{\theta}(z|\lambda) = z$ for $|\theta| \geq t_1$ with large t_1. This leads the unbiasedness of $\widehat{\theta}(z|\lambda)$.

(c) Show that $\widehat{\theta}(z|\lambda)$ is continuous in z if and only if $\text{argmin}_{\theta\geq 0}\{\theta + p'_\lambda(\theta)\} = 0$. This leads to the continuity of $\widehat{\theta}(z|\lambda)$.

3.3 Let $g(\theta|z,\lambda,\sigma) = \frac{1}{2\sigma^2}(z-\theta)^2 + p_\lambda(|\theta|)$ for a given λ, z and σ, and denote $\widehat{\theta}(z|\lambda,\sigma) = \text{argmin}_\theta\, g(\theta|z,\lambda,\sigma)$.

(a) Take the penalty function to be the SCAD penalty whose derivative is given in (3.14). Derive the expressive form solution of $\widehat{\theta}(z|\lambda,\sigma)$.

(b) Take the penalty function to be the MCP whose derivative is given in (3.17). Derive the closed form solution $\widehat{\theta}(z|\lambda,\sigma)$.

(c) Comment how λ relates to σ so that the solutions in (a) and (b) still have sparsity, unbiasedness and continuity.

3.4 Suppose that $Z \sim N(\theta,\sigma^2)$. Derive the closed form of risk function $R(\theta) = E(\widehat{\theta}(Z) - \theta)^2$ for the hard-thresholding rule (3.16), the soft-thresholding rule (3.18), the SCAD thresholding rule given (3.19) and the MCP thresholding rule (3.20), respectively. Plot $R(\theta)$ against θ with $\sigma = 2$ and 3, and compare your plot with Figure 3.3.

3.5 Consider the lasso problem $\min_{\boldsymbol{\beta}} \frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1$, where $\lambda > 0$ is a tuning parameter.

(a) If $\widehat{\boldsymbol{\beta}}_1$ and $\widehat{\boldsymbol{\beta}}_2$ are both minimizers of the lasso problem, show that they have the same prediction, i.e., $\mathbf{X}\widehat{\boldsymbol{\beta}}_1 = \mathbf{X}\widehat{\boldsymbol{\beta}}_2$. **Hint:** Consider the vector $\alpha\widehat{\boldsymbol{\beta}}_1 + (1-\alpha)\widehat{\boldsymbol{\beta}}_2$ for $\alpha \in (0,1)$.

(b) Let $\widehat{\boldsymbol{\beta}}$ be a minimizer of the lasso problem with j^{th} componenet $\widehat{\beta}_j$. Denote $\mathbf{X}_j$ to be the j-th column of $\mathbf{X}$. Show that

$$\begin{cases} \lambda = n^{-1}\mathbf{X}_j^T(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) & \text{if } \widehat{\beta}_j > 0; \\ \lambda = -n^{-1}\mathbf{X}_j^T(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}) & \text{if } \widehat{\beta}_j < 0; \\ \lambda \geq |n^{-1}\mathbf{X}_j^T(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})| & \text{if } \widehat{\beta}_j = 0. \end{cases}$$

(c) If $\lambda > \|n^{-1}\mathbf{X}^T\mathbf{Y}\|_\infty$, prove that $\widehat{\boldsymbol{\beta}}_\lambda = \mathbf{0}$, where $\widehat{\boldsymbol{\beta}}_\lambda$ is the minimizer of the lasso problem with regularization parameter λ.

3.6 Verify the KKT conditions in (3.22), (3.23) and (3.24) for penalized least squares.

3.7 Consider the elastic-net loss $p(\theta) = \lambda_1|\theta| + \lambda_2\theta^2$. Let $\widehat{\boldsymbol{\beta}}$ be the minimizer of $\frac{1}{2n}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \sum_{j=1}^p p(|\beta_j|)$.

(a) For $\lambda_2 > 0$, show that $\widehat{\boldsymbol{\beta}}$ is unique.

(b) Give the necessary and sufficient conditions for $\widehat{\boldsymbol{\beta}}$ being the penalized least-squares solution.

(c) If $\lambda_1 > \|n^{-1}\mathbf{X}^T\mathbf{Y}\|_\infty$, show that $\widehat{\boldsymbol{\beta}} = \mathbf{0}$.

3.8 Concentration inequalities.

(a) The random vector $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is called σ-sub-Gaussian if $E\exp\left(\mathbf{a}^T\boldsymbol{\varepsilon}\right) \leq \exp\left(\|\mathbf{a}\|_2^2\sigma^2/2\right)$, $\forall \mathbf{a} \in \mathbb{R}^n$. Show that $E\boldsymbol{\varepsilon} = \mathbf{0}$ and $\mathrm{Var}(\boldsymbol{\varepsilon}) \leq \sigma^2\mathbf{I}_n$. **Hint:** Expand exponential functions as infinite series.

(b) For $\mathbf{X} \in \mathbb{R}^{n\times p}$ with the j-th column denoted by $\mathbf{X}_j \in \mathbb{R}^n$, suppose that $\|\mathbf{X}_j\|_2^2 = n$ for all j, and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is a σ-sub-Gaussian random vector. Show that there exists a constant $C > 0$ such that

$$P\Big(\|n^{-1}\mathbf{X}^T\boldsymbol{\varepsilon}\|_\infty > \sqrt{2(1+\delta)}\sigma\sqrt{\frac{\log p}{n}}\Big) \leq Cp^{-\delta}, \qquad \forall \delta > 0.$$

3.9 The goal is to show the concentration inequality for the median-of-means estimator when the random variable only has finite second moment. We divide the problem into three simple steps.

(a) Let X be a random variable with $EX = \mu < \infty$ and $\mathrm{Var}(X) = \sigma^2 < \infty$. Suppose that we have m i.i.d. random samples $\{X_i\}_{i=1}^m$ with the same distribution as X. Let $\widehat{\mu}_m = \frac{1}{m}\sum_{i=1}^m X_i$. Show that

$$P\big(|\widehat{\mu}_m - \mu| \geq \frac{2\sigma}{\sqrt{m}}\big) \leq \frac{1}{4}.$$

(b) Given k i.i.d. Bernoulli random variables $\{B_j\}_{j=1}^k$ with $EB_j = p < \frac{1}{2}$. Use the moment generating function of B_j, i.e., $E(\exp(tB_j))$, to show that

$$P\big(\frac{1}{k}\sum_{j=1}^k B_j \geq \frac{1}{2}\big) \leq (4p(1-p))^{\frac{k}{2}}.$$

(c) Suppose that we have n i.i.d. random samples $\{X_i\}_{i=1}^n$ from a population with mean $\mu < \infty$ and variance $\sigma^2 < \infty$. For any positive integer k, we randomly and uniformly divide all the samples into k subsamples, each having size $m = n/k$ (for simplicity, we assume n is always divisible by

k). Let $\widehat{\mu}_j$ be the sample average of the j^{th} subsample and $\widetilde{m}$ be the median of $\{\widehat{\mu}_j\}_{j=1}^k$. Apply the previous two results to show that

$$P\Big(|\widetilde{m}-\mu| \geq 2\sigma\sqrt{\frac{k}{n}}\Big) \leq \big(\frac{\sqrt{3}}{2}\big)^k.$$

Hint: Consider the Bernoulli random variable $B_j = \mathbb{1}\{|\widehat{\mu}_j - \mu| \geq 2\sigma\sqrt{\frac{k}{n}}\}$ for $j = 1, ..., k$.

3.10 This problem intends to show that the gradient decent method for a convex function $f(\cdot)$ is a member of majorization-minimization algorithms and has a sublinear rate of convergence in terms of function values. From now on, the function $f(\cdot)$ is convex and let $\mathbf{x}^* \in \text{argmin} f(\mathbf{x})$. Here we implicitly assume the minimum can be attained at some point $\mathbf{x}^\star \in \mathbb{R}^p$.

(a) Suppose that $f''(\mathbf{x}) \leq L\mathbf{I}_p$ and $\delta \leq 1/L$. Show that the quadratic function $g(\mathbf{x}) = f(\mathbf{x}_{i-1}) + f'(\mathbf{x}_{i-1})^T(\mathbf{x}-\mathbf{x}_{i-1}) + \frac{1}{2\delta}\|\mathbf{x}-\mathbf{x}_{i-1}\|^2$ is a majorization of $f(\mathbf{x})$ at point $\mathbf{x}_{i-1}$, i.e., $g(\mathbf{x}) \geq f(\mathbf{x})$ for all $\mathbf{x}$ and also $g(\mathbf{x}_{i-1}) = f(\mathbf{x}_{i-1})$.

(b) Show that gradient step $\mathbf{x}_i = \mathbf{x}_{i-1} - \delta f'(\mathbf{x}_{i-1})$ is the minimizer of the majorized quadratic function $g(\mathbf{x})$ and hence the gradient descend method can be regarded as a member of MM-algorithms.

(c) Use (a) and the convexity of $f(\cdot)$ to show that

$$f(\mathbf{x}_i) \leq f(\mathbf{x}^*) + \frac{1}{2\delta}(\|\mathbf{x}_{i-1}-\mathbf{x}^*\|^2 - \|\mathbf{x}^*-\mathbf{x}_i\|^2).$$

(d) Conclude using (c) that $f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \|\mathbf{x}_0 - \mathbf{x}^*\|^2/(2k\delta)$, namely gradient descent converges at a sublinear rate. (**Note**: The gradient descent method converges linearly if $f(\cdot)$ is strongly convex.)

3.11 Conduct a numerical comparison among the quadratic programming algorithm (3.57), the LARS algorithm, the cyclic coordinate descent algorithms and the ADMM algorithm for lasso.

3.12 Conduct a numerical comparison between the lasso and the one-step SCAD estimator using LLA with lasso initial values.

3.13 Show that when the dimension of $\mathbf{x}$ is finite and fixed, the SCAD with GCV-tuning parameter selector defined in (3.98) leads to an overfitted model. That is, the selected model contains all important variables, but with a positive probability, the selected model contains some unimportant variables. See, Wang, Li and Tsai (2007).

3.14 Show that when the dimension of $\mathbf{x}$ is finite and fixed, the SCAD with BIC-tuning parameter selector defined in (3.99) yields a consistent estimation of the model.

3.15 Show that $\widehat{\sigma}_1^2$ defined in (3.102) is a root-n consistent estimator of σ^2.

3.16 Extend the ADMM algorithm in Section 3.5.9 for group-lasso regression.

3.17 Extend the ISTA algorithm in Section 3.5.7 for penalized least squares with group penalty, and further apply the new algorithm for variable selection in varying coefficient models in (3.107).

3.18 Let us consider the 128 macroeconomic time series from Jan. 1959 to Dec. 2018, which can be downloaded from the book website. In this problem, we will explore what macroeconomic variables are associated with the unemployment rate contemporarily and which macroeconomic variables lead the unemployment rates.

(a) Extract the data from Jan. 1960 to Oct. 2018 (in total 706 months) and remove the feature named "sasdate". Then, remove the features with missing entries and report their names.

(b) The column with name "UNRATE" measures the difference in unemployment rate between the current month and the previous month. Take this column as the response and take the remaining variables as predictors. To conduct contemporary association studies, do the following steps for lasso (using R package `glmnet`) and SCAD (using R package `ncvreg`): Set a random seed by `set.seed(525)`; Plot the regularization paths as well as the mean squared errors estimated by 10-fold cross-validation; Choose a model based on cross-validation, report its in-sample R^2, and point out two most important macroeconomic variables that are correlated with the current change of unemployment rate and explain why you choose them.

(c) In this sub-problem, we are going to study which macroeconomic variables are leading indicators for the changes of future unemployment rate. To do so, we will pair each row of predictors with the next row of response. The last row of predictors and the first element in the response are hence discarded. After this change, do the same exercise as (b).

(d) Consider the setting of (c). Leave the last 120 months as testing data and use the rest as training data. Set a random seed by `set.seed(525)`. Run lasso and SCAD on the training data using `glmnet` and `ncvreg`, respectively, and choose a model based on 10-fold cross-validation. Compute the out-of-sample R^2's for predicting the changes of the future unemployment rates.

3.19 Consider the Zillow data analyzed in Exercise 2.9. We drop the first 3 columns ("(empty)", "id", "date") and treat "zipcode" as a factor variable. Now, consider the variables

(a) "bedrooms", "bathrooms", "sqft_living", and "sqft_lot" and their interactions and the remaining 15 variables in the data, including "zipcode".

(b) "bedrooms", "bathrooms", "sqft_living", "sqft_lot" and "zipcode", and their interactions and the remaining 14 variables in the data. (We can use *model.matrix* to expand factors into a set of dummy variables.)

(c) Add the following additional variables to (b): $X_{12} = I(view == 0)$, $X_{13} = L^2$, $X_{13+i} = (L - \tau_i)_+^2$, $i = 1, \cdots, 9$, where τ_i is $10 * i^{th}$ percentile and L is the size of living area ("sqft_living").

Compute and compare out-of-sample R^2 using ridge regression, lasso, SCAD with regularization parameter chosen by 10 fold cross-validation.