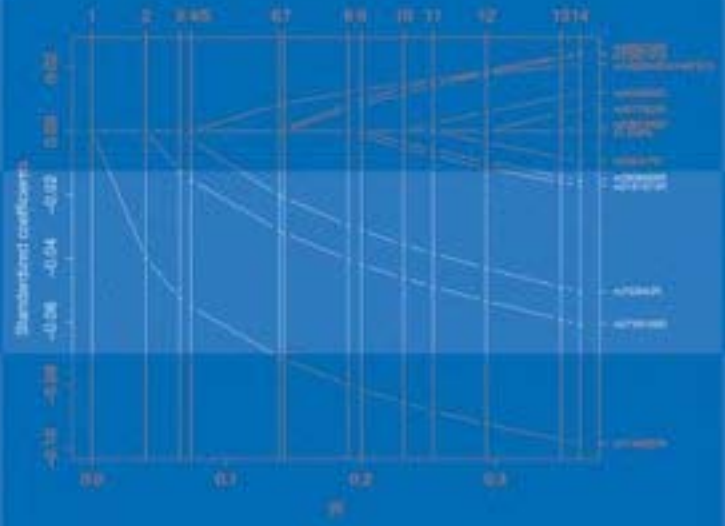
Applied Bioformatics and Biostatistices

Xiaochun Li
Ronghui Xu
Editors

High-Dimensional Data Analysis in Oncology



Springer

Applied Bioinformatics and Biostatistics in Cancer Research

Series Editors

Jeane Kowalski
John Hopkins University, Baltimore, MD, USA

Steven Piantadosi
Cedars Sinai Medical Center, Los Angeles, CA, USA

For other titles published in this series, go to
www.springer.com/series/7616

Xiaochun Li · Ronghui Xu
Editors

High-Dimensional Data Analysis in Cancer Research

Springer

Editors

Xiaochun Li
Harvard Medical School
Dana-Farber Cancer Institute
Dept. Biostatistics
375 Longwood St.
Boston MA 02115
USA
xiaochun@iupui.edu

Ronghui Xu
University of California
San Diego
Department of Family
and Preventive Medicine
and Department of Mathematics
9500 Gilman Dr.
La Jolla CA 92093-0112
USA
rxu@ucsd.edu

ISBN: 978-0-387-69763-5 e-ISBN: 978-0-387-69765-9
DOI: 10.1007/978-0-387-69765-9

Library of Congress Control Number: 2008940562

Printed on acid-free paper

springer.com

To our children, Anna, Sofia, and James

Preface

In an era with a plethora of high-throughput biological technologies, biomedical researchers are investigating more comprehensive aspects of cancer with ever-finer resolution. Not only does this result in large amount of data but also data with hundreds if not thousands of dimensions.

Multivariate analysis is a mainstay of statistical tools in the analysis of biomedical data. It concerns with associating data matrices of n rows by p columns, with rows representing samples or patients and columns attributes, to certain response or outcome variables. Classically, the sample size n is much larger than the number of attributes p. The theoretical properties of statistical models have mostly been discussed under the assumption of fixed p and infinite n. However, the advance of biological sciences and technologies has revolutionized the process of investigations in cancer. The biomedical data collection has become much more automatic and much more extensive. We are in the era of p as a large fraction of n, or even much larger than n, which poses challenges to the classical statistical paradigm. Take proteomics as an example. Although proteomic techniques have been researched and developed for many decades to identify proteins or peptides uniquely associated with a given disease state, until recently this has mostly been a laborious process, carried out one protein at a time. The advent of highthroughput proteome-wide technologies such as liquid chromatography-tandem mass spectroscopy make it possible to generate proteomic signatures that facilitate rapid development of new strategies for proteomics-based detection of disease. This poses new challenges and calls for scalable solutions to the analysis of such high-dimensional data.

In this volume, we present current analytical approaches as well as systematic strategies to the analysis of correlated and high-dimensional data.

The volume is intended as a reference book for researchers, statisticians, bioinformaticians, graduate students, and data analysts working in the field of cancer research. Our aim is to present methodological topics of important relevance to such analyses, and in a single volume such as this we do not attempt to exhaust all the analytical tools that have been developed so far.

This volume contains seven chapters. They do not necessarily cover all topics relevant to high-dimensional data analysis in cancer research. Instead, we have aimed

to choose those fields of research that are either relatively mature, but may not have been well read in applied statistics, such as risk estimation, or those fields that are fast developing and also have obtained substantial newer results that are reasonably well understood for practical use, such as variable selection. On the other hand, we have omitted such an important topic as multiple comparisons, which is currently undergoing much theoretical development (as reflected in the August 2007 issue of *Annals of Statistics*, for example), and we find it possibly difficult to provide an accurate stationary yet updated picture for the moment. Such topic, however, can be found in several other recently published books that contain its classical results ready for practical use. All the chapters included in this book contain practical examples to illustrate the analysis methods. In addition, they also reveal the types of research that are involved in developing these methods.

The opening chapter provides an overview of the various high-dimensional data sources, the challenges in analyzing such data, and in particular, strategies in the design phase, as well as possible future directions. Chapter 2 discusses methodologies and issues surrounding variable selection and model building, including postmodel selection inference. These have always been important topics in statistical research, and even more so in the analysis of high-dimensional data. Chapter 3 is devoted to the topic of multivariate nonparametric regression. Multivariate problems are common in oncological research, and often the relationship between the outcome of interest and its predictors is either nonlinear, or nonadditive, or both. This chapter focuses on the methods of regression trees and spline models. Chapter 4 discusses the more fundamental problem of risk estimation. This is the basis of many procedures and, in particular, model selection. It reviews the two major approaches to risk estimation, i.e., covariance penalty and resampling, and summarizes empirical evaluations of these approaches. Chapter 5 focuses on tree-based methods. After a brief review of classification and regression trees (CART), the chapter presents in more detail tree-based ensembles, including boosting and random forests. Chapter 6 is on support vector machines (SVMs), one of the methodologies stemming from the machine learning field that has gained popularity for classification of high dimensional data. The chapter discusses both two-class and multiclass classification problems, and linear and nonlinear SVM. For high-dimensional data, a particularly important aspect is sparse learning, that is, only a relatively small subset of the predictors are truly involved with the classification boundary. Variable selection is then again a critical step, and various approaches associated with SVM are described. The last, but by no means the least, chapter, presents Bayesian approaches to the analyses of microarray gene expression data. The emphasis is on nonparametric Bayesian methods, which allow flexible modeling of the data that might arise from underlying heterogeneous mechanisms. Computational algorithms are discussed.

It has been an exciting experience editing this volume. We thank all the authors for their excellent contributions.

Boston, MA *Xiaochun Li*
La Jolla, CA *Ronghui Xu*

Contents

Contributors

Adele Cutler Department of Mathematics and Statistics, Utah State University, 3900 Old Main Hill Logan, UT 84322-3900, USA, Adele.Cutler@usu.edu

Richard Cutler Department of Mathematics and Statistics, Utah State University, 3900 Old Main Hill Logan, UT 84322-3900, USA, Richard.Cutler@usu.edu

Anthony Gamst Division of Biostatistics and Bioinformatics, Department of Family and Preventive Medicine, University of California, San Diego, 9500 Gilman Drive, MC0717 La Jolla, CA 92093-0717, USA, agamst@ucsd.edu

Jaroslaw Harezlak Division of Biostatistics, Indiana University School of Medicine, 410 West 10th Street, Suite 3000 Indianapolis, IN 46202, USA, harezlak@iupui.edu

Sonia Jain Division of Biostatistics and Bioinformatics, University of California, San Diego, 9500 Gilman Drive, MC-0717, La Jolla, CA 92093-0717, USA, sojain@ucsd.edu

Charles Kooperberg Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, 1100 Fairview Ave N, M3-A410 Seattle, WA 98109-1024, USA, clk@fhcrc.org

Michael LeBlanc Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, 1100 Fairview Ave N, M3-C102 Seattle, WA 98109-1024, USA, mleblanc@fhcrc.org

Xiaochun Li Division of Biostatistics, Indiana University School of Medicine, 410 West 10th Street, Suite 3000 Indianapolis, IN 46202, USA, xiaochun@iupui.edu

Ross L. Prentice Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, 1100 Fairview Avenue North, Seattle, WA 98109, USA, rprentic@WHI.org

John R. Stevens Department of Mathematics and Statistics, Utah State University, 3900 Old Main Hill Logan, UT 84322-3900, USA, John.R.Stevens@usu.edu

Eric Tchetgen Department of Biostatistics, Harvard School of Public Health, 655 Huntington Avenue, Boston, MA 02115, USA, etchetge@hsph.harvard.edu

Ronghui Xu Division of Biostatistics and Bioinformatics, Department of Family and Preventive Medicine and Department of Mathematics, University of California, San Diego, 9500 Gilman Drive, MC 0112 La Jolla, CA 92093-0112, USA, rxu@ucsd.edu

Hao Helen Zhang Department of Statistics, North Carolina State University, 2501 Founders Drive Raleigh, NC 27613, USA, hzhang@stat.ncsu.edu

Chapter 1
On the Role and Potential of High-Dimensional Biologic Data in Cancer Research

Ross L. Prentice

1.1 Introduction

I am pleased to provide a brief introduction to this volume of "High-Dimensional Data Analysis in Cancer Research". The chapters to follow will focus on data analysis aspects, particularly related to regression model selection and estimation with high-dimensional data of various types. The methods described will have a major emphasis on statistical innovations that are afforded by high-dimensional predictor variables.

While many of the motivating applications and datasets for these analytic developments arise from gene expression data in therapeutic research contexts, there are also important applications, and potential applications, in risk assessment, early diagnosis and primary disease prevention research, as will be elaborated in Section 1.2. With this range of applications as background, some preliminary comments are made on related statistical challenges and opportunities (Section 1.3) and on needed future developments (Section 1.4).

1.2 Potential of High-Dimensional Data in Biomedical Research

1.2.1 Background

The unifying goal of the various types of high-dimensional data being generated in recent years is the understanding of biological processes, especially processes that relate to disease occurrence or management. These may involve, for example, characteristics such as single nucleotide polymorphisms (SNPs) across the genome to be

R.L. Prentice
Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, 1100 Fairview Avenue North, Seattle, WA, USA
email: rprentic@WHI.org

X. Li, R. Xu (eds.), *High-Dimensional Data Analysis in Cancer Research*, Applied Bioinformatics and Biostatistics in Cancer Research, DOI 10.1007/978-0-387-69765-9_1,

related to the risk of a disease; gene expression patterns in tumor tissue to be related to the risk of tumor recurrence; or protein expression patterns in blood to be related to the presence of an undetected cancer. Cutting across the biological processes related to carcinogenesis, or other chronic disease processes, are high-dimensional data related to treatment or intervention effects. These may include, for example, study of changes in the plasma proteome as a result of an agent having chronic disease prevention potential; or changes in gene expression in tumor tissue as a result of exposure to a therapeutic regimen, especially a molecularly targeted regimen. It is the confluence of novel biomarkers of disease development and treatment, with biomarker changes related to possible interventions that have great potential to enhance the identification of novel preventative and therapeutic interventions. Furthermore, biological markers that are useful for early disease detection open the door to reduced disease mortality, using current or novel therapeutic modalities. The technology available for these various purposes in human studies depends very much on the type of specimens available for study, with white blood cells and their DNA content, tumor tissue and its mRNA content, or blood serum or plasma and its proteomic and metabolomic (small molecule) content, as important examples. The next subsections will provide a brief overview of the technology for assessment of certain key types of high-dimensional biologic data.

1.2.2 High-Dimensional Study of the Genome

The study of genotype in relation to the risk of specific cancers or other chronic diseases has traditionally relied heavily on family studies. Such studies often involve families having a strong history of the study disease to increase the probability of harboring disease-related genes. A study may involve genotyping family members for a panel of genetic markers and assessing whether one or more markers co-segregate with disease among family members. This approach uses the fact that chromosomal segments are inherited intact, so that markers over some distance from a disease-related gene can be expected to associate with disease risk within families. Following the identification of a "linkage" signal with a genetic marker, some form of fine mapping is needed to close in on disease-related loci. There are many variations in ascertainment schemes and analysis procedures that may differ in efficiency and robustness (e.g., Ott, 1991; Thomas, 2004) with case–control family studies having a prominent role in recent years.

Markers that are sufficiently close on the genome tend to be correlated, depending somewhat on a person's evolutionary history (e.g., Felsenstein, 2007). The identification of several million SNPs across the human genome (e.g., Hinds et al., 2005) and the identification of tag SNP subsets (The International HapMap Consortium, 2003) that convey most genotype information as a result of such correlation (linkage disequilibrium) have opened the way not only to family-based studies that involve a very large number of genomic markers, but also to direct disease association studies

among unrelated individuals. For example, the latter type of study may simultaneously relate 100,000 or more tag SNPs to disease occurrence in a study cohort, typically using a nested case–control or case-cohort design.

However, for this type of association study to be practical, there needs to be reliable, high-throughput genotyping platforms having acceptable costs. Satisfying this need has been a major technology success story over the past few years, with commercially available platforms (Affymetrix, Illumina) having 500,000–1,000,000 well-selected tagging SNPs, and genotyping costs reduced to a few hundred dollars per specimen. These platforms, similar to the gene expression platforms that preceded them, rely on chemical coupling of DNA from target cells to labeled probes having a specified sequence affixed to microarrays, and use photolithographic methods to assess the intensity of the label following hybridization and washing. In addition to practical cost, these platforms can accommodate the testing of thousands of cases and controls in a research project in a matter of a few weeks or months.

The results of very high-dimensional SNP studies of this type have only recently begun to emerge, usually from large cohorts or cohort consortia, in view of the large sample sizes needed to rule out false positive associations. Novel genotype associations with disease risks have already been established for breast cancer (e.g., Easton et al., 2007; Hunter et al., 2007) and prostate cancer (Amundadottir et al., 2006; Freedman et al., 2006; Yeager et al., 2007), as well as for several other chronic diseases (e.g., Samani et al., 2007, for coronary heart disease). Although it is early to try to characterize findings, novel associations for complex common diseases tend to be weak, and mostly better suited to providing insight into disease processes and pathways, than to contributing usefully to risk assessment. The prostate cancer associations cited include well-established SNP associations that are not in proximity to any known gene, providing the impetus for further study of genomic structure and characteristics in relation to gene and protein expression.

1.2.3 High-Dimensional Studies of the Transcriptome

Studies of gene expression patterns in tumor tissue from cancer patients provided some of the earliest use of microarray technologies in biomedical research, and constituted the setting that motivated much of the statistical design and analysis developments to date for high-dimensional data studies. Gene expression can be assessed by the concentration of mRNA (transcripts) in cells, and many applications to date have focused on studies of tumors or other tissue, often in a therapeutic context. mRNA hybridizes with labeled probes on a microarray, with a photolithographic assessment of transcript abundance through label intensity. A microarray study may, for example, compare transcript abundance between two groups for 10,000 or more human genes.

Studies of the transcription pattern of specific tumors provide a major tool for assessing recurrence risk, and prognosis more generally, and for classifying patients

into relatively homogenous groups (e.g., Golub et al., 1999) according to tumor characteristics, for tailored molecularly targeted treatment. Examples of molecularly targeted cancer therapies, developed in recent years, include imatinib mesylate (Gleevec) for the treatment of chronic myeloid leukemia (e.g., Druker et al., 2006), and trastuzumab (Herceptin) for the treatment of HER-2 positive breast cancer (Piccart-Gebhart et al., 2005; Rouzier et al., 2005). This is an enterprise that can be expected to profoundly affect tumor classification and cancer patient management in the upcoming years. The related literature is already quite extensive and will not be summarized here. Other papers in this volume will bring out cancer therapy-related applications of gene expression data.

There is also a valuable emerging bioinformatics literature on inferring function and pathways from gene expression microarray data patterns (e.g., Khatri and Draghici, 2005).

1.2.4 High-Dimensional Studies of the Proteome

The technology for simultaneous study of the expression of a large number of proteins in tumor tissue, blood, or other body compartment is less developed than is the case for gene expression. However, technology developments are being vigorously pursued, and the potential for biomedical research is enormous. Proteins undergo a wide variety of modifications following transcription and translation, and post-translational modifications may greatly influence function, and potentially contribute to carcinogenesis and other disease processes. Of potential importance for early detection and risk assessment purposes, protein expression can be assessed using stored blood products (serum or plasma) in the form typically included in biological repositories associated with cohort studies or prevention clinical trials. One could also study the effects of, say, a dietary or physical activity intervention on a high-dimensional subset of the plasma proteome. When combined with data to describe the association of plasma protein concentrations with the risk of a range of clinical outcomes, such proteomic data have potential to invigorate the preventive intervention development enterprise, and perhaps avoid the late discovery of adverse effects that can eliminate the practical value of an otherwise promising intervention.

A particular challenge in plasma proteomics is the very broad range of concentrations of circulating proteins. For example, it may be that novel proteins in small malignant tumors that have yet to surface clinically are shed in minute quantities into blood. Identification of such proteins in plasma could be quite valuable for the early detection of latent cancers. To do so, however, may require the ability to identify and quantify proteins whose abundance is 10 orders of magnitude less than that of the most abundant proteins.

Accordingly, the type of liquid chromatography (LC) and mass spectrometry (MS) platforms that have been developed for this purpose mostly begin by immunodepletion of the few most abundant proteins, followed by varying degrees of fractionation to reduce the number of proteins in each fraction. The proteins

in each fraction are then trypsin-digested; and mass spectrometry is used to identify the peptides in a fraction. A second mass spectrometry step is then used to sequence and identify the peptides showing peaks in the first set of mass spectra, after which proteins present in the original sample are bioinformatically reconstructed from databases linking proteins to the peptides they contain. There are many variants on the LC-MS/MS approaches just alluded to, with differing degrees of initial immunodepletion, differing labeling strategies for relative protein quantitation (e.g., to quantitatively compare a diseased case to a matched control; or to compare a post-treatment to a pre-treatment plasma specimen from the same study subject), and differing fractionation schemes. In addition, some platforms digest proteins into peptides before fractionation.

To cite a specific example, the intact protein analysis system (e.g., Faca et al., 2006) of my colleague, Dr. Samir Hanash, immunodepletes the six most abundant proteins, applies either a light or a heavy acrylamide label to the members of a pair of samples to be compared, mixes the labeled samples, separates first by anion exchange chromatography, then by reverse phase chromatography, typically generating about 100 fractions, followed by trypsin digestion and tandem mass spectrometry in each fraction. This methodology may identify about 2,500 unique proteins from a sample pair, and provide relative abundance quantitation for a subset of 1,000 cysteine containing proteins (to which the acrylamide label binds).

Proteomic platforms involving antigen or antibody arrays are also under intensive development. For example, epitopes from a larger number of proteins may be spotted onto a microarray. The affinity and label luminescence of corresponding autoantibodies then reflect the presence and quantity of proteins included in the test sample. For example, Wang et al. (2005) report an autoantibody signature for prostate cancer from this type of approach. Arrays containing as many as 8,000 distinct epitopes have been developed and are becoming commercially available.

1.2.5 Other Sources of High-Dimensional Biologic Data

There are quite a number of additional sources of high-dimensional biologic data. For example, in addition to their use for studying SNPs, gene expression, and antigen-antibody profiles, microarrays are in use for studying DNA methylation patterns, DNA copy number changes (through comparative genomic hybridization), and microRNAs. This range of applications is likely to grow. Moreover, high-throughput DNA sequencing is rapidly developing, and data sets containing millions of short reads will soon become available. Methods for investigating the metabolome are also under rapid development, using the same type of LC-MS/MS technology sketched above, or other (e.g., Shurubor et al., 2005) methods. Important high-dimensional data also derive from various types of scans (e.g., PET scans or mammograms) with various applications in cancer research.

1.3 Statistical Challenges and Opportunities with High-Dimensional Data

1.3.1 Scope of this Presentation

Statistical aspects of the analysis and use of high-dimensional biologic data is the major focus of this volume. The attraction of such data, of course, is the potential to be rather thorough in studying the association between an outcome variable Y and a set of "predictor" variables $X_1, \ldots, X_p$ that may include a fairly comprehensive assessment of variables of interest within a targeted compartment. On the other hand, the high-dimensionality implies that many of the associations examined will meet conventional significance testing criteria by chance alone, so a major challenge is to rule out the "false positives" in an efficient and convincing manner, in a setting where there may be complex correlation patterns among predictor variables. Here only a brief introduction will be provided of some of the statistical concepts involved, with few citations. Subsequent chapters will provide a detailed account of statistical considerations and approaches to the analysis of certain types of high-dimensional data.

1.3.2 Two-Group Comparisons

Many "discovery-type" applications of high-dimensional data involve a binary Y, and a predictor variable array $X_1, \ldots, X_p$, with p very large. Logistic regression methods are frequently used to compare cases ($Y = 1$) to controls ($Y = 0$), or post-treatment ($Y = 1$) to pre-treatment ($Y = 0$) groups, with respect to the elements (one-at-a-time) of this array, usually with the control for other potentially confounding factors. Tests based on linear regression are also widely used, as are two-sample nonparametric tests. Under the assumption that at most a small fraction of predictor variables relate to Y, the array data may first be "normalized" by relocation and/or rescaling to reduce measurement variation between arrays.

While testing at a nominal level would generate many "associations" even under a global null hypothesis, the other extreme of ensuring an experiment-wise error rate below a certain α-level (e.g., using a Bonferroni correction) is typically far too stringent for discovery work. Hence, the false discovery rate (FDR) approach of controlling the expected fraction of discoveries that are false (Benjamini and Hochberg, 1995) provides a valuable alternative in many applications. Even then, however, the fact that many elements of an array may have limited plausibility for association with Y suggests that the power of discovery research may be enhanced by some external "filtering" of the elements of the array. For example, primary testing in humans could be based on the subset of an array for which there is evidence of association with a comparable response variable in the mouse or other model system, or high-dimensional human protein expression comparisons may benefit by restriction to proteins that derive from genes that express in a manner that is associated with Y.

An interesting analytic idea (Efron, 2004), again deriving from the assumption that only a small fraction of the elements of an array are plausibly associated with Y, involves using the empirical distribution of standardized test statistics to generate basis for judging statistical significance, rather than using the theoretical null distribution (e.g., normal or t-distribution). This approach has the potential to preserve desired error rates, for example, through the adaptability of the "empirical null" distribution, even in the presence of uncontrolled confounding of certain types. Permutation tests, formed by comparing a test statistic to the distribution of test statistics that arise by randomly reallocating response values ($Y = 1$ or $Y = 0$) also have utility in this type of setting, particularly if the sample size (e.g., number of arrays) is small and the asymptotic null distributional approximations may be inadequate. Classification error probabilities and ROC curves (with quantitative X variables), among other classification and predictiveness techniques, also have a useful role in the analysis of this type of data. There is also a useful place for testing and estimating methods that "borrow strength" across predictor variables (e.g., empirical Bayes' methods), especially if the study involves a small number of independent samples.

1.3.3 Study Design Considerations

A typical analysis of a high-dimensional data set with binary Y begins by scanning through the elements of the array, or a filtered subset thereof, one-at-a-time to identify "biomarkers" of interest for further evaluation. In some settings (e.g., SNP-disease association studies), the associations being sought are likely to be weak (e.g., odds ratios of 1.1 or 1.2 for presence of minor SNP allele) requiring enormous sample sizes for definitive testing. In other settings (e.g., plasma proteome disease associations), available technology may be low throughput and expensive. In either case, there may be important efficiencies and cost savings by using a multi-staged design, or using a specimen pooling strategy.

The NCI-sponsored Cancer Genetic Epidemiology Markers of Susceptibility (C-GEMS) program illustrates the multi-stage design approach. For example, an on-going C-GEMS breast cancer case–control study involved about 1,200 white cases and controls from the Nurses Health Study and Illumina's 550,000 SNP set at a first stage, and applied a likelihood ratio test to the minor allele counts ($X = 0, 1,$ or 2) at each SNP. In spite of the large number of SNPs tested, there was strong enough evidence for association with one particular SNP to report an association based primarily on first-stage data (Hunter et al., 2007). A second stage is currently underway using DNA from about 3,000 white cases and controls from the Women's Health Initiative cohort study. This second stage includes about 30,000 SNPs selected primarily based on empirical evidence for an association from the first stage. Testing the initial 550,000 SNPs in all first- and second-stage cases and controls would have doubled or tripled the genotyping costs for this project, with little corresponding gain in power. The study design calls for two or three subsequent stages,

each with some thousands of cases and controls to rule out false positives and thoroughly identify breast cancer-related SNPs.

Multi-stage SNP studies of breast cancer, coronary heart disease, and stroke are also nearing completion, each involving 2000–4000 cases and controls from the Women's Health Initiative. As a cost-saving device, the first stage in these multi-stage efforts involved pooled DNA, with pools formed from 125 cases or 125 matched controls (Prentice and Qi, 2006). DNA pooling reduced project genotyping costs by a factor of about 25. In theory, such cost reduction should be accommodated by fairly modest power reductions under additive or dominant genetic models for an SNP (but with major power reductions under a recessive genetic model), but available data to evaluate the role of pooling are limited, and genotyping costs continue to drop rapidly. Hence, while the issue cannot be said to be resolved, it is likely that large-scale SNP studies of the future will mostly use individual-level genotyping.

Multi-stage designs also have a fundamental role in proteomic discovery research, and specimen pooling for LC-MS/MS platforms has important practical and cost advantages in early detection, risk assessment, and preventive intervention development contexts. To cite a particular example, Women's Health Initiative investigators, in collaboration with Dr. Samir Hanash, have recently completed a comparison between the baseline and 1 year from randomization proteomes for 50 women assigned to estrogen plus progestin (Writing Group for the Women's Health Initiative Investigators, 2002) and 50 women assigned to estrogen alone (The Women's Health Initiative Steering Committee, 2004) using the Intact Protein Analysis System. In each case baseline and 1-year serum pools from sets of 10 women were analyzed and provided FDRs of interest for a sizeable number of the approximately 1,000 baseline to 1-year proteomic changes quantified. These data are being filtered by corresponding pooled data from breast cancer cases and controls to identify a small number of proteins that may be able to explain estrogen plus progestin effects on breast cancer risk, and differences in risk between estrogen plus progestin and estrogen alone. This small number of proteins will then be validated in an independent set of cases and controls using an individual-level (e.g., ELISA) test as a second stage. These collective data will also be used to ask whether hormone therapy clinical trial results on breast cancer, coronary heart disease, and stroke could have been anticipated by changes in the serum proteome, as a test case for the role of proteomic changes in prevention, intervention, development, and initial evaluation.

1.3.4 More Complex High-Dimensional Data Analysis Methods

The binary response ($Y = 0$ or 1) discussion above readily applies to case and control data from cohort studies with time to event responses, upon matching cases and controls on pertinent time variables. Other applications could involve other quantitative response variables, for which many of the same considerations would apply, perhaps using generalized linear models to derive test statistics.

Regardless of the nature of the response variable Y, additional important analytic topics include statistical methods for model building to relate multiple elements of an array $X = (X_1, \ldots, X_p)$ simultaneously to Y, including the study of interactions among elements of X (in relation to Y), and the study of interactions between the elements of an array and a set of "environmental" factors. With large p, the fitting of each of the potential (2^p) regression models of this type may well exceed modern computing power, and the ability to distinguish empirically among fitted models may be limited. Various approaches can be considered to these difficult and thorny modeling topics, with some form of cross-validation typically being crucial to the establishment of a specific regression model association. Additionally, specialized regression methods (e.g., Tibshirani, 1996; Ruczinski et al., 2003) have been proposed to enhance sensitivity to identifying combinations of specific types of predictor variables that may associate with an outcome. Often data analytic methods focus on X variables having evidence of association with Y marginally, to reduce the predictor variable space for gene–gene, or gene–environment interaction testing. These are topics that will be discussed in detail in the subsequent chapters.

1.4 Needed Future Research

It is still too early to ascertain and use many types of high-dimensional data. While the technology for high-dimensional SNP and gene expression studies has developed nicely, the sample sizes typically applied for the two types of studies differ to a surprising degree. While genome-wide association of SNP studies take a brute force approach with thousands of cases and controls to establish specific SNP-disease associations, gene expression comparisons (e.g., patients with or without relapse) have mostly relied on very small sample sizes (at most a few hundred) with focus on the entire expression profile or pattern. The underlying assumption seems to be, for example, that tumor recurrence includes a number of steps resulting in expression changes that are fairly common among patients, or that involve a small number of classes of patients with common expression profiles within classes. Additional research would seem to be needed to establish that biologic differences among patients do not dominate the expression pattern comparisons with such small sample sizes.

Proteomic technologies are still at an intermediate stage of development, with differences of opinion concerning the most promising platforms. The LC-MS/MS platforms, which seem to have the most potential at present, are decidedly low throughput, limiting their applications to date. The concept of plasma proteomics discovery for the early detection of cancer has yet to be firmly established. A related important research question concerns whether researchers can work in a sufficiently powerful manner from stored blood specimens alone for early detection discovery work, or whether it is necessary to work first with blood obtained at diagnosis, and with tumor tissue, where the proteins being sought may be comparatively abundant. Good quality prediagnostic blood specimens are well suited to serum or plasma proteomic discovery research, whereas it may be difficult to obtain blood having equivalent handling and storage from a suitable control group to compare

with blood specimens obtained at diagnosis for cancer patients and, of course, suitable control group tissue specimens may not often be available in human studies. It would considerably simplify proteomic early detection discovery research if prediagnostic stored blood specimens turn out to be sufficiently sensitive, but this issue is yet to be resolved.

The assessment technology for high-dimensional metabolomic data is at a quite early stage of development, and may ultimately be needed for the development of a sufficiently comprehensive understanding of disease and intervention pathways and networks.

A more technical statistical need relates to estimation methods with high-dimensional data. Typically in high-dimensional SNP studies, for example, odds ratios will be presented only for a small number of SNPs meeting stringent statistical selection criteria. The uncorrected odds ratio estimates may be severely biased away from the null as a result of this extreme selection, and methods for odds ratio correction under single or multi-stage designs are needed for study reporting.

Finally, there is a growing need for methods to integrate data across multiple types of high-dimensional biologic data (e.g., SNPs, gene expression, protein expression, …) to obtain a more comprehensive picture of the regulatory processes and networks that may be pertinent to disease processes, or intervention effects, under study.

Statisticians and bioinformaticians have much to contribute to these burgeoning enterprises over upcoming years.

Acknowledgments This work was partially supported by grants CA53996, CA86368, and CA106320.

References

Amundadottir, L., Sulem, P., Gudmundsson, J., et al. (2006). A common variant associated with prostate cancer in european and african populations. *Nature Genetics*, 38(6):652–658.

Benjamini, Y. and Hochberg, Y. (1995). Controlling for false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57:289–300.

Druker, B. J., Guilhot, F., O'Brien, S. G., et al. (2006). Five-year follow-up of patients receiving imatinib for chronic myeloid leukemia. *New England Journal of Medicine*, 355(23):2408–2417.

Easton, D., Pooley, K., Dunning, A., et al. (2007). Genome-wide association study identifies novel breast cancer susceptibility loci. *Nature*, 447(7148):1087–1093.

Efron, B. (2004). Large-scale simultaneous hypothesis testing: The choice of a null hypothesis. *Journal of the American Statistical Association*, 99:96–104.

Faca, V., Coram, M., Phanstiel, D., et al. (2006). Quantitative analysis of acrylamide labeled serum proteins by lc-ms/ms. *Journal of Proteome Research*, 5(8):2009–2018.

Felsenstein, J. (2007). *Theoretical Evolutionary Genetics*. University of Washington/ASUW Publishing, Seattle, WA.

Freedman, M. L., Haiman, C. A., Patterson, N., et al. (2006). Admixture mapping identifies 8q24 as a prostate cancer risk locus in african-american men. *Proceedings of the National Academy of Sciences*, 103(38):14068–14073.

Golub, T. R., Slonim, D. K., Tamayo, P., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537.

Hinds, D. A., Stuve, L. L., Nilsen, G. B., et al. (2005). Whole-genome patterns of common DNA variation in three human populations. *Science*, 307(5712):1072–1079.

Hunter, D. J., Kraft, P., Jacobs, K. B., et al. (2007). A genome-wide association study identifies alleles in fgfr2 associated with risk of sporadic postmenopausal breast cancer. *Nature Genetics*, 39(6):870–874.

Khatri, P. and Draghici, S. (2005). Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, 18:3587–3595.

Ott, J. (1991). *Analysis of Human Genetic Linkage*. Johns Hopkins University Press, Baltimore, MD.

Piccart-Gebhart, M. J., Procter, M., Leyland-Jones, B., et al. (2005). Trastuzumab after adjuvant chemotherapy in her2-positive breast cancer. *New England Journal of Medicine*, 353(16): 1659–1672.

Prentice, R. and Qi, L. (2006). Aspects of the design and analysis of high-dimensional snp studies for disease risk estimation. *Biostatistics*, 7:339–354.

Rouzier, R., Perou, C. M., Symmans, W. F., et al. (2005). Breast cancer molecular subtypes respond differently to preoperative chemotherapy. *Clinical Cancer Research*, 11(16):5678–5685.

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2003). Logic regression. *Journal of Computational and Graphical Statististics*, 12:475–511.

Samani, N. J., Erdmann, J., Hall, A. S., et al. (2007). Genomewide association analysis of coronary artery disease. *New England Journal of Medicine*, 357(5):443–453.

Shurubor, Y., Matson, W., Martin, R., et al. (2005). Relative contribution of specific sources of systematic errors and analytic imprecision to metabolite analysis by hplc-ecd. *Metabolomics*, 1:159–168.

The International HapMap Consortium (2003). The international hapmap project. *Nature*, 426(6968):789–796.

The Women's Health Initiative Steering Committee (2004). Effects of conjugated equine estrogen in postmenopausal women with hysterectomy: The women's health initiative randomized controlled trial. *JAMA*, 291(14):1701–1712.

Thomas, D. (2004). *Statistical Methods in Genetic Epidemiology*. Oxford University Press, London.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288.

Wang, X., Yu, J., Sreekumar, A., et al. (2005). Autoantibody signatures in prostate cancer. *New England Journal of Medicine*, 353(12):1224–1235.

Writing Group for the Women's Health Initiative Investigators (2002). Risks and benefits of estrogen plus progestin in healthy postmenopausal women: Principal results from the women's health initiative randomized controlled trial. *Journal of the American Medical Association*, 288(3):321–333.

Yeager, M., N., O., Hayes, R. B., et al. (2007). Genome-wide association study of prostate cancer identifies a second risk locus at 8q24. *Nature Genetics*, 39(5):645–649.

Chapter 2
Variable Selection in Regression – Estimation, Prediction, Sparsity, Inference

Jaroslaw Harezlak, Eric Tchetgen, and Xiaochun Li

2.1 Overview of Model Selection Methods

Over the past 30 years, variable selection has become a topic of central importance to regression modeling. In recent years, its primary relevance to empirical methods for cancer research has further been underscored with the now routine collection of data from high-throughput technologies such as microarrays and mass spectrometry. In general, one is interested in the selection of a few explanatory variables in a regression problem with a large set of potential covariates. This chapter concerns variable selection in generalized linear models, a problem that is in fact common to "-omics" technologies used in cancer research (e.g., genomics and proteomics), where the number of possible explanatory variables p often surpasses the number of available observations n.

Let $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}'$ denote the outcome vector and $\mathbf{X} = [x_{ij}], i = 1, \ldots, n$; $j = 1, \ldots, p$ denote the matrix of possible explanatory variables with i indexing subjects and j indexing covariates. In general, the outcome may be quantitative (e.g., tumor size), or binary (e.g., "case" or "control"). To focus our exposition, the first six sections below consider only the linear regression setting; we briefly discuss extensions to generalized linear models in Section 2.7.

Consider the standard linear model

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \varepsilon_i, \tag{2.1}$$

where the errors ε_i have mean 0 and common variance σ^2. Without loss of generality, assume that $\sum_i x_{ij} = 0, \sum_i x_{ij}^2 = 1$, and $\sum_i y_i = 0$.

J. Harezlak
Division of Biostatistics, Indiana University School of Medicine, 410 West 10th street, Suite 3000 Indianapolis, IN 46202, USA
email: harezlak@iupui.edu

X. Li, R. Xu (eds.), *High-Dimensional Data Analysis in Cancer Research*, Applied Bioinformatics and Biostatistics in Cancer Research, DOI 10.1007/978-0-387-69765-9_2,

In the classical setting of ordinary least-squares regression (OLS) with the number of variables much less than the sample size, that is, $p \ll n$, $\boldsymbol{\beta}$ is typically estimated by minimizing the residual sum of squares (RSS). If the columns of $\mathbf{X}$ are linearly independent, the OLS estimate of $\boldsymbol{\beta}$ is $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$. When $p > n$, the design matrix $\mathbf{X}$ can no longer be full rank. As a result there is no unique OLS solution. If $\widehat{\boldsymbol{\beta}}_0$ is a solution, then $\widehat{\boldsymbol{\beta}}_1 = \widehat{\boldsymbol{\beta}}_0 + \boldsymbol{\alpha}$ is also a solution, where $\boldsymbol{\alpha}$ is in the null space of $\mathbf{X}$. Although $\widehat{\boldsymbol{\beta}}_0$ and $\widehat{\boldsymbol{\beta}}_1$ differ nontrivially in the parameter space, they give exactly the same prediction $\mathbf{X}\widehat{\boldsymbol{\beta}}_1 = \mathbf{X}\widehat{\boldsymbol{\beta}}_0$. When the purpose of the model selection is to obtain a good prediction of a future patient's outcome, the focus is on minimizing the risk $E(E(y|x) - x'\boldsymbol{\beta})^2$ by selecting a sparse regression model with minimal bias and variance. The bias here comes from the fact that the linear regression model need not hold. On the other hand, the estimation of $\boldsymbol{\beta}$ itself may be of interest, for example, if one is willing to assume that the posited regression model holds with only few nonzero coefficients, then identifying the corresponding relevant covariates and reporting confidence intervals centered at their estimated effects becomes the primary objective. A model selection procedure that achieves this goal has been referred to in the literature as *consistent* (e.g., see Knight and Fu, 2000, for the LASSO case). One should note that a model selection procedure leading to optimal prediction need not yield optimal estimates of $\boldsymbol{\beta}$ under the model and vice versa, see Leng et al. (2006) for a discussion of the case of LASSO.

In Section 2.2, we briefly review some classical subset selection methods. These methods typically attempt to find the "best" of the 2^p possible models using an automated search through a model space. Due to their intense combinatorial nature, best subset regression methods do not scale well with the size of the full model, and can become computationally intractable for even moderately sized regressions. In the $p \gg n$ setting, where no unique OLS solution exists and subset regression no longer applies, a remedy recently advocated in the literature on model selection is that of regularized estimation via a method of penalization. Below, we review this approach for a variety of penalties, though our interest is primarily in penalties that lead to sparse regularized solutions with few nonzero estimated coefficients. We do not attempt to discuss all recent proposals in this chapter. Interested readers can refer to the book by Miller (2002) or a more recent article by Fan and Li (2006). We provide details on four particular methods: LASSO (Tibshirani, 1996), SCAD (Fan and Li, 2001), LARS (Efron et al., 2004), and the Dantzig selector (Candes and Tao, 2007); we also mention certain interesting extensions of these initial proposals. Finally, we address the following two topics:

1. In Section 2.5, we discuss the distinction between optimal model selection gauged toward prediction versus optimality in terms of estimation; the former is related to model selection *persistence* (Greenshtein and Ritov, 2004), whereas as stated earlier, the latter has been referred to as model selection *consistency* (Knight and Fu, 2000).
2. In Section 2.6, we summarize some known limitations of post-model selection inference on $\boldsymbol{\beta}$ when using a procedure based on sparsity (Leeb and Pötscher, 2008b).

In the next section, we describe a data example used to illustrate various model selection methods.

2.1.1 Data Example

To illustrate the use of the variable selection methods, we use the data coming from a study by Stamey et al. (1989), which collected a number of clinical variables in men who were to receive a radical prostatectomy. The dataset contains the following variables, log(prostate specific antigen) (lpsa), log(cancer volume) (lcavol), log(prostate weight) (lwight), age, log(benign prostatic hyperplasia amount)(lbph), seminal vesicle invasion (svi), log(capsular penetration) (lcp), gleason score (gleason), and percentage Gleason scores 4 or 5 (pgg45). We fit a linear regression model to log(cancer volume) with the regressors consisting of polynomial terms of a maximum degree 2 and allowing for at most two-way interactions between the covariates.

2.1.2 Univariate Screening of Variables

In the simple case of an orthonormal design matrix $\mathbf{X}$, one can show that several optimal variable selectors essentially base the inclusion of a given covariate into the model, solely on the marginal association between the outcome and the candidate covariate (Tibshirani, 1996) with no consideration given to the other variables. Unfortunately, in the "-omics" studies that interest us, we can never hope to even approximate this idealized orthonormal design setting, as covariates in $\mathbf{X}$ are typically highly correlated, sometimes with a few pairwise correlations exceeding 0.9 for instance for gene expressions on the same pathway. The following example demonstrates that in such a setting, a procedure that ignores the dependence among the covariates can lead to a suboptimal variable selector. Our example concerns the undesirable features of univariate screening of variables as reported in the paper by Paul et al. (2008) on pre-selection of variables using univariate tests.

Define (Y, X_1, X_2, X_3) to follow a multivariate normal distribution with mean $\boldsymbol{\mu} = \mathbf{0}$ and the variance–covariance matrix

$$\Sigma = \begin{pmatrix} 1 & -0.5 & -0.5 & 0 \\ -0.5 & 1 & 0.5 & -0.5 \\ -0.5 & 0.5 & 1 & -0.5 \\ 0 & -0.5 & -0.5 & 1 \end{pmatrix}.$$

Also, let $(X_4, \ldots, X_{300})$ be additional predictors normally distributed with mean zero and identity covariance matrix. The true regression coefficients are $\boldsymbol{\beta} = (-1, -1, -1, 0, 0, \ldots, 0)$ and the marginal correlations of each predictor with the response are given by $\boldsymbol{\rho} = (-0.5, -0.5, 0, 0, 0, \ldots, 0)$. In this case, the covariate X_3 is uncorrelated with Y, but it has nonzero partial correlation with Y when conditioning on other covariates. Thus, by univariate screening of the covariates, we would falsely exclude X_3 from further analysis.

In the following section, we improve on univariate screening methods with the use of subset selection.

2.1.3 Subset Selection

Traditional strategies for the selection of important covariates include best subset selection, forward, backward, and stepwise selection procedures, which are all combinatorial in nature. Given $k \in \{1,2,\ldots,p\}$, best subset selection finds a subset of variables of size k that gives the smallest empirical loss (e.g. R^2). Furnival and Wilson (1974) developed an efficient algorithm, "leaps and bounds procedure," for best subset selection. Though subset selection procedures produce interpretable and possibly sparse models, they tend to be very discrete and discontinuous in the data, leading to highly unstable solutions. Furthermore, proposed algorithms can typically not handle over 40 potential covariates.

Backward stepwise selection can only be used when $p < n$, since the first model that is considered contains all possible covariates. Variables are excluded from the model according to an "exit" criterion, e.g., an exclusion threshold for p-values associated with individual coefficients. In contrast, both forward and stepwise selection start with a model consisting of a single covariate, and the covariates are subsequently added to the model according to an "entry" criterion of similar nature to that used in backward selection. In stepwise selection, a backward step is typically performed after a variable is entered into the model. The immense computational burden of these methods severely limits their applicability to large data sets encountered in cancer research; in view of this limitation, we turn next to model selection via penalization, an approach which has recently gained wide appeal among statistical analysts facing the multiple modeling challenges of high-dimensional data.

2.2 Multivariable Modeling: Penalties/Shrinkage

2.2.1 Penalization

The general form of penalized methods for standard linear model (2.1) is given by

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \sum_i \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \right\} + P_\lambda(\boldsymbol{\beta}), \tag{2.2}$$

where $P_\lambda(\boldsymbol{\beta})$ is the penalty and $\lambda \geq 0$. Popular penalties include $\lambda \sum_j |\beta_j|^m$, where the choice $m = 0$ yields the so-called L_0 penalty $\lambda \sum_j I(|\beta_j| \neq 0)$ common to classical methods such as AIC and BIC.

Fan and Li (2006) argue that the penalized criterion (2.2) encompasses the L_0 penalty with the choice of tuning parameter λ corresponding asymptotically to the classical maximization of the adjusted R^2 given by

$$1 - \frac{\text{RSS}_k/(n-k)}{\text{RSS}_1/(n-1)},$$

where RSS_k is the residual sum of squares for a model with k covariates and the tuning parameter $\lambda = \sigma^2/(2n)$. Generalized cross-validation (GCV)

$$\text{GCV}(k) = \frac{n\text{RSS}_k}{(n-k)^2}$$

offers another example of (2.2) with corresponding tuning parameter $\lambda = \sigma^2/n$.

In general, for powers $0 < m \leq 1$, the penalized criterion automatically performs variable selection due to the penalty's singularity at zero. Powers $m \geq 1$ result in so called "bridge regression models" (Fu, 1998) which include LASSO for $m = 1$ and ridge regression for $m = 2$.

2.2.2 Ridge Regression and Nonnegative Garrote

We begin with ridge regression which was introduced by Hoerl and Kennard (1970) and corresponds to using the penalty $P_\lambda(\boldsymbol{\beta}) = \lambda \sum_j \beta_j^2$. One of its attractive features is that unlike best subset selection, it is continuous in the data. The ridge estimator takes the form:

$$\widehat{\boldsymbol{\beta}}^{\text{ridge}} = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}'\mathbf{Y},$$

where $\mathbf{I}$ is an identity matrix. Ridge regression has been shown to give more accurate predictions than the best subset regression, unless the true model is sparse. Moreover, it is easy to show that the ridge estimator generally has smaller variance than the OLS estimator achieved by shrinking the OLS estimates toward zero. Note however, that none of the coefficients are estimated as exactly zero which makes the interpretation of the model difficult. Building on shrinkage estimators via ridge regression, Breiman (1995) proposed the nonnegative garrote defined as

$$\widehat{\boldsymbol{\beta}}^{\text{NG}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \left\{ \sum_i \left(y_i - \sum_{j=1}^{p} c_j \widehat{\beta}_j^o x_{ij} \right)^2 \right\}$$
$$\text{subject to} \quad c_j \geq 0 \text{ and } \sum_j c_j \leq t,$$

where $\widehat{\beta}_j^o$ are the OLS estimates, which are scaled by nonnegative c_j whose sum is constrained by t. The solution to the minimization problem can be written as: $\widehat{\beta}_j =$

$\widehat{c}_j\widehat{\beta}_j^o$, where $\widehat{c}_j$ is only available in closed form in the simple case of an orthogonal design matrix:

$$\widehat{c}_j = (1-\lambda/(\widehat{\beta}_j^o)^2)_+, \quad j=1,\ldots,p,$$

where $(x)_+ = x$ for $x > 0$ and is 0 otherwise. Breiman (1995) showed that this estimator has lower prediction error than best subset selection procedures and performs as well as ridge regression; although unlike the latter, the garrote can produce estimates of exactly zero. Unfortunately, the original garrote is vulnerable to instabilities in the preliminary least squares estimates, particularly in the presence of highly correlated covariates where OLS is known to break down. Moreover, computing the solution path of the garrote using standard quadratic programming techniques as originally proposed can be computationally demanding and impractical. Finally, the garrote is limited in the number of covariates ($p \leq n$) that may be used in fitting the procedure. To remedy the first limitation, Yuan and Lin (2007) recently proposed to substitute the ridge estimator for the OLS plug-in, and show that the modified estimator is far more stable. In the same manuscript, the authors offer a computational algorithm that obtains the entire solution path of the nonnegative garrote with computational load comparable to that of OLS regression, thereby resolving the second major limitation of the original garrote. In addition, they prove that the nonnegative garrote can be a consistent model selection procedure under fairly general conditions; model selection consistency of a procedure states that it estimates zero coefficients exactly with probability tending to one as sample size grows to infinity.

2.2.3 LASSO: Definition, Properties and Some Extensions

The least absolute shrinkage and selection operator (LASSO), introduced by Tibshirani (1996), was motivated by the aforementioned Breiman's nonnegative garrote estimator. The procedure shrinks some coefficients toward zero and sets some to be exactly zero; thereby combining the favorable features of best subset selection and ridge regression. The LASSO penalty term is given by $P_\lambda(\boldsymbol{\beta}) = \lambda\sum_j|\beta_j|$. In the simple case of an orthogonal design matrix $\mathbf{X}$, one gets an explicit solution to the minimization problem (2.2):

$$\widehat{\beta}_j^{\text{LASSO}} = \text{sign}(\widehat{\beta}_j^o)(|\widehat{\beta}_j^o| - \lambda/2)_+, \quad j=1,\ldots,p,$$

where $\widehat{\beta}_j^{\text{o}}$ is the OLS solution. For nonorthogonal designs, there exist efficient quadratic programming algorithms to find LASSO solutions for a given value of λ. More generally, the least angle regression (LARS) algorithm can find the whole LASSO solution path (all $\lambda > 0$) with the same computational load as the OLS solution (see Section 2.3). It is important to note that the original LASSO cannot make use of more than n covariates and it is highly sensitive to high correlations among covariates.

A major challenge not yet addressed in this chapter is the selection of an optimal tuning parameter λ. Among others, Leng et al. (2006) address this issue, and show that when the prediction accuracy is used as a criterion to choose the tuning parameter in penalized procedures, these procedures will generally fail to be consistent. In fact, in the simple orthogonal design case, they show that the probability of correctly identifying the model via LASSO is bounded above by a constant $c < 1$, uniformly in n.

For a fixed number of parameters p, Zou (2006) shows that LASSO is in general not consistent. In particular, he shows that the probability that zero coefficients are estimated as zero is generally less than 1. In Zhao and Yu (2006), "irrepresentability conditions" on the design matrix which we discuss below, provide an almost necessary and sufficient condition for LASSO to be consistent. However, the tuning parameter λ required for selection consistency can excessively shrink the nonzero coefficients leading to asymptotically biased estimates.

Irrepresentability conditions on the design matrix can be summarized as follows: let $C = X^{'}X$, and divide C into blocks corresponding to the s covariates with nonzero coefficients and $d = p - s$ covariates with zero coefficients resulting in:

$$C = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

Assume that C_{11} is invertible, then for a positive constant vector η

$$|C_{21}(C_{11})^{-1}\text{sign}(\boldsymbol{\beta}_{(s)})| \leq \mathbf{1} - \eta,$$

where $\boldsymbol{\beta}_{(s)} = (\beta_1, \ldots, \beta_s)$ and $\beta_j \neq 0$ for $j = 1, \ldots, s$ is the strong irrepresentability condition. There is also a weak irrepresentability condition which changes the inequality to $< \mathbf{1}$.

Examples of the conditions on the design matrix $\mathbf{X}$ implying strong irrepresentability include:

1. Constant positive correlation: $0 < \rho_{ij} \leq 1/(1+cs)$, where $c > 0$,
2. Bounded correlation: $|\rho_{ij}| \leq c/(2s-1)$ for a constant $0 \leq c < 1$ (also called *coherence in* Donoho et al., 2006),
3. Power decay correlation: $\rho_{ij} = \rho^{|i-j|}$.

A fascinating property of LASSO-like procedures is that of persistence (Greenshtein and Ritov, 2004); which entails the asymptotic equivalence of the L_2 risk evaluated at the LASSO estimated parameters and the smallest achievable L_2 risk on a particular restricted parameter space (e.g., regression coefficients with bounded L_1 norm) in a model of very high dimension. We shall return to persistence in Section 2.5 where more details are provided.

To alleviate the major limitations of LASSO, there have been proposals to extend L_1 penalization. A prominent proposal made by Zou (2006) is the "Adaptive LASSO" which makes use of data dependent weights. The penalty takes the form

$$P_\lambda(\boldsymbol{\beta}) = \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\widehat{\beta}_j^I|^\gamma},$$

where $\widehat{\beta}_j^I$ is an initial estimator of β_j and $\gamma > 0$, hence large coefficients are shrunk less and small coefficients are shrunk more than in the LASSO solution. For fixed p, adaptive LASSO yields selection consistency of nonzero estimates of coefficients with asymptotic distribution equal to that obtained in the model with prior knowledge of the location of zero parameters. This latter property has been described as a desirable "oracle" property (Fan and Li, 2001). In fact, Huang et al. (2007) showed that the adaptive LASSO maintains the oracle property for $p_n \to \infty$ as $n \to \infty$ under general conditions on the design matrices which we do not reproduce here.

Other extensions to LASSO include

- "Elastic net penalty" (Zou and Hastie, 2005) with penalty taking the form

$$P_\lambda(\boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} |\beta_j|^2,$$

  where λ_1 and λ_2 are tuning parameters. Elastic net alleviates the limitation on the maximum number of parameters chosen by LASSO. Zou and Hastie (2005) show in simulations that highly correlated variables are grouped, and are either selected or removed from the model as a group.
- "Fused LASSO" (Tibshirani et al., 2005)-a generalization to LASSO exploiting the ordering of the regression coefficients. The penalty takes the form

$$P_\lambda(\boldsymbol{\beta}) = \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|.$$

  Its solutions are sparse in the original coefficients, that is, nonzero estimates of coefficients are few, as well as in the differences between the coefficients, promoting the equality of the neighboring coefficients.

The penalized methods thus far discussed share the nice feature of being convex optimization problems. In the following section, we discuss a generalized penalized method with some additional appealing statistical properties.

2.2.4 Smoothly Clipped Absolute Deviation (SCAD)

Fan and Li (2001) proposed three desirable properties that penalized methods should fulfil:

a. *Sparsity.* An estimator should accomplish variable selection by automatically setting small coefficients to zero.
b. *Unbiasedness.* An estimator should have low bias, especially for large true coefficients β_j.
c. *Continuity.* An estimator should be continuous in data to avoid instability in model prediction.

The penalty methods given in Sections 2.2.1, 2.2.2, and 2.2.3 fail to simultaneously satisfy properties a, b, and c. For example the L_0 penalty does not satisfy the continuity condition, and the L_1 penalty violates the unbiasedness condition. Fan and Li (2001) construct a penalized estimator called the smoothly clipped absolute deviation (SCAD) to fulfill all three properties. The SCAD penalty is given by

$$P_\lambda(\boldsymbol{\beta}) = \lambda \left\{ I(|\beta_j| \leq \lambda) + \frac{(a\lambda - \beta_j)_+}{(a-1)\lambda} I(|\beta_j| > \lambda) \right\},$$

for some $a > 2$ (the original paper suggested using $a = 3.7$) and λ, a tuning parameter. The SCAD solution in the simple orthogonal design case corresponds to

$$\widehat{\boldsymbol{\beta}}^{\text{SCAD}} = \begin{cases} \text{sign}(x)(|x| - \lambda)_+, & \text{for } |x| \leq 2\lambda; \\ \{(a-1)x - \text{sign}(x)a\lambda\}/(a-2), & \text{for } 2\lambda < |x| \leq a\lambda; \\ x, & \text{for } |x| > a\lambda. \end{cases}$$

SCAD has two major advantages over best subset selection: it offers a lower computational cost and provides continuous solutions. SCAD is similar in spirit to LASSO, though it generally yields smaller bias. Moreover, as proved by Fan and Li, SCAD enjoys the particular oracle property described in the previous section, whereby, the estimator performs *as well as* an optimal estimator in a model where all zero coefficients are known to the analyst. In Section 2.6, we give a detailed discussion on known limitations of this property, particularly, as it pertains to post-model selection inference.

2.3 Least Angle Regression

The LARS algorithm recently proposed by Efron et al. (2004) is a model selection and fitting algorithm for model (2.2) with three important properties: (i) a simple modification of the LARS algorithm implements the LASSO estimator, (ii) a different modification efficiently implements Forward Stagewise linear regression (yet another model selection procedure), and (iii) a simple approximation for the degrees of freedom of a LARS estimate is available, allowing for the straightforward derivation of a C_p estimate of prediction error. The current section gives a general description of the LARS procedure, and solely emphasizes property (i), while (ii) and (iii) are discussed in detail in the original manuscript by Efron et al. (2004).

We begin with an informal description of the steps that constitute the LARS solution. Starting with $\widehat{\mu}_0 = 0$, find the index $j_1 = \arg\max_j \langle Y, \mathbf{x}_j \rangle$ of the covariate with maximal correlation with the outcome. Next, take the largest possible step in the $\mathbf{x}_{j_1}$ direction until some other predictor, say $\mathbf{x}_{j_2}$ has the same amount of correlation with the current residual; that is until $\max_{j \neq j_1} \left| \langle Y - \widehat{\gamma}_1 \mathbf{x}_{j_1}, \mathbf{x}_j \rangle \right| = \left| \langle Y - \widehat{\gamma}_1 \mathbf{x}_{j_1}, \mathbf{x}_{j_s} \rangle \right|, s = 1, 2.$ Write $\widehat{\mu}_1 = \widehat{\mu}_0 + \widehat{\gamma}_1 \mathbf{x}_{j_1}$. Next, follow the direction equiangular between $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$, which we denote $\mathbf{u}_2$ (defined below), until a third covariate $\mathbf{x}_{j_3}$ enters the most

correlated set so that $\widehat{\mu}_2 = \widehat{\mu}_1 + \widehat{\gamma}_2 \mathbf{u}_2$, $\max_{j \neq j_1, j_2} \left| \left\langle Y - \widehat{\mu}_2, \mathbf{x}_j \right\rangle \right| = \left| \left\langle Y - \widehat{\mu}_2, \mathbf{x}_{j_s} \right\rangle \right|, s = 1,2,3$; then continue along the equiangular direction $\mathbf{u}_3$ between $\mathbf{x}_{j_1}$,$\mathbf{x}_{j_2}$, and $\mathbf{x}_{j_3}$, and so on for $s = 4, \ldots, m \leq p$.

The appealing feature of LARS is that it is fairly easy to calculate the step sizes $\widehat{\gamma}_1, \widehat{\gamma}_2, \ldots$. In the case of linearly independent columns of $\mathbf{X}$, an explicit formula for $\widehat{\boldsymbol{\gamma}}$ was derived in Efron et al. (2004).

We now discuss property (i) of LARS. For $\mathcal{A} \subset \{1, \ldots, p\}$, let $\mathbf{X}_{\mathcal{A}} = (\ldots, s_j \mathbf{x}_j, \ldots)_{j \in \mathcal{A}}$, where $s_j = \pm 1, G_{\mathcal{A}} = \mathbf{X}'_{\mathcal{A}} \mathbf{X}_{\mathcal{A}}, A_{\mathcal{A}} = \left(\mathbf{1}'_{\mathcal{A}} G_{\mathcal{A}}^{-1} \mathbf{1}_{\mathcal{A}} \right)^{-1/2}$, where $\mathbf{1}_{\mathcal{A}} = (1, ..., 1)' \in \mathbb{R}^{|\mathcal{A}|}$ then the equiangular vector $\mathbf{u}_{\mathcal{A}} = \mathbf{X}_{\mathcal{A}} w_{\mathcal{A}}$, where $w_{\mathcal{A}} = A_{\mathcal{A}} G_{\mathcal{A}}^{-1} \mathbf{1}_{\mathcal{A}}$ is the unit vector making equal angles, less than 90°, with the columns of $\mathbf{X}_{\mathcal{A}}, \mathbf{X}'_{\mathcal{A}} \mathbf{u}_{\mathcal{A}} = A_{\mathcal{A}} \mathbf{1}_{\mathcal{A}}$ and $\|\mathbf{u}_{\mathcal{A}}\|^2 = 1$. The main idea is that the nonzero components $\widehat{\beta}_j$ of a LASSO solution $\widehat{\boldsymbol{\beta}}$ can be shown to satisfy the following:

$$\operatorname{sign}\left(\widehat{\beta}_j\right) = \operatorname{sign}\left(\widehat{c}_j\right) = s_j, \tag{2.3}$$

where $\widehat{c}_j$ is the jth component of a vector of correlations $\widehat{c} = \mathbf{X}'(Y - \widehat{\mu}_{\mathcal{A}})$ with $\widehat{\mu}_{\mathcal{A}}$ a LARS estimate based on $\mathcal{A}$ covariates, and $s_j = \operatorname{sign}(\widehat{c}_j)$. Note that this restriction is not imposed by LARS, however, (2.3) is easily incorporated into the LARS steps. Define $\mu(\gamma) = \widehat{\mu}_{\mathcal{A}} + \gamma \mathbf{u}_{\mathcal{A}}$, so that $\beta_j(\gamma) = \widehat{\beta}_j + \gamma s_j w_{\mathcal{A}j}$ for $j \in \mathcal{A}$, also showing that $\beta_j(\gamma)$ will switch sign at $\gamma_j = \widehat{\beta}_j (s_j w_{\mathcal{A}j})^{-1}$. Under the "one at a time" assumption that all increases and decreases of the active set involve at most one index j, the equivalence between LARS and LASSO solution paths is obtained by stopping an ongoing LARS step at $\gamma = \widetilde{\gamma} = \min_{j \in \mathcal{A}} (\gamma_j)$ if $\widetilde{\gamma} < \widehat{\gamma}$, and removing $\widetilde{j}$ from the calculation of the next equiangular direction: $\mu_{\mathcal{A}_+} = \widehat{\mu}_{\mathcal{A}} + \widetilde{\gamma} \mathbf{u}_{\mathcal{A}}$ and $\mathcal{A}^+ = \mathcal{A} - \left\{ \widetilde{j} \right\}$. An immediate consequence of this modification is that the solution path of LASSO will typically involve more steps than that of the original LARS procedure, as the active set of LARS grows monotonically whereas the LASSO modification allows $\mathcal{A}$ to decrease. However, the modified LARS calculates all possible LASSO estimates for a given problem in computing time of the same order of magnitude as that of least squares in the full model.

2.4 Dantzig Selector

The Dantzig selector (henceforth DS) of Candes and Tao (2007) is a novel approach recently proposed for performing variable selection and model fitting in precisely those scientific settings where $p \gg n$, but where only a few of the regression coefficients are expected to be nonzero. In this section, we give an overview of the DS, and some of its important statistical properties. We discuss conditions under which the DS is known to be optimal in a sense defined hereafter.

The Dantzig selector is defined as the solution to the convex problem:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 \quad \text{subject to} \quad \left\|\mathbf{X}'(Y-\mathbf{X}\boldsymbol{\beta})\right\|_\infty := \sup_{1\le j\le p}\left|\left[\mathbf{X}'(Y-\mathbf{X}\boldsymbol{\beta})\right]_j\right| \le \lambda\sigma, \quad (2.4)$$

where $\lambda > 0$ is the tuning parameter. In a fashion similar to the LASSO, the L_1 minimization (2.4) yields some coefficient estimates of exactly zero and thus makes the DS very useful for variable selection. A more general comparison to the LASSO solution (Tibshirani, 1996) is possible if we write (2.4) in its equivalent form given by Efron et al. (2007):

$$\min_{\boldsymbol{\beta}} \left\|\mathbf{X}'(Y-\mathbf{X}\boldsymbol{\beta})\right\|_\infty \quad \text{subject to} \quad \|\boldsymbol{\beta}\|_1 < s \quad (2.5)$$

for some $s > 0$. In this latter form, with a bound on the L_1 norm of $\boldsymbol{\beta}$, the DS minimizes the maximum component of the gradient of the squared error function, while the LASSO solution (Tibshirani, 1996) directly minimizes the squared error. The nature of the constraint in (2.4) is essential to guarantee that the residuals are within the noise level while ensuring that the estimation procedure remains invariant with respect to orthonormal transformations applied to the data. In their manuscript, Candes and Tao (2007) describe the minimization (2.4) as looking for the vector $\widehat{\boldsymbol{\beta}}$ with minimum complexity measured by the L_1-norm among all coefficient vectors consistent with the data.

An astonishing theoretical result on the DS is that even when p is much larger than the sample size n, the L_2 error in the estimated coefficients can remain within a $\log(p)$ of the one that could be achieved had the locations of the sparse nonzero coefficients been known. This oracle property realizes the analyst's ultimate desire to use the data at hand, to adapt to an existing parsimonious model. The DS achieves this goal without incurring too much of a cost in the L_2 estimation error, reflected by the $\log(p)$ factor, which increases very slowly in p. To make a more formal statement on the accuracy of the DS, suppose that $\boldsymbol{\beta}$ is $s-$sparse; that is at most s of the regression coefficients are nonzero. Furthermore, assume the error ε follows a normal distribution and that the design matrix $\mathbf{X}$ obeys "Uniform Uncertainty Principle," which is defined formally in Candes and Tao (2007) and summarized below. Then, by using $\lambda = \sqrt{(1+a)\log p}$ in solving(2.4) with $a \ge 0$, they proved that the following nonasymptotic bound on the L_2 error of the resulting DS $\widehat{\boldsymbol{\beta}}$ holds:

$$\left\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right\|_2^2 \le C(1+a)\, s\sigma^2 \log(p) \quad (2.6)$$

with probability exceeding $1 - \left(p^a\sqrt{\pi \log p}\right)^{-1}$ and $C > 0$ a constant that depends on $\mathbf{X}$.

"Uniform Uncertainty Principle" is defined in terms of properties of the design matrix $\mathbf{X}$ and its submatrices. The first property, the "$s-$restricted isometry hypothesis," quantified as δ_s means that every set of columns of $\mathbf{X}$ with cardinality

less that s behaves approximately as an orthonormal system. The second property, "restricted orthogonality," expressed as $\theta_{s,2s}$ puts a restriction on the disjoint subsets of covariates resulting in the subsets spanning nearly orthogonal subspaces.

Candes and Tao (2007) showed that it is sufficient to require

$$\delta_s + \theta_{s,2s} \leq 1, \tag{2.7}$$

which in effect restricts the result to design matrices endowed with certain attributes that make them sufficiently close to orthonormal matrices as reflected by a small constant $\delta_s + \theta_{s,2s}$. Assuming that (2.7) is satisfied, we may ask what makes (2.6) either a reasonable or desirable bound. To answer this question, consider an "oracle" who happens to know the exact locations of the nonzero coefficients, say T_0. This clairvoyant analyst could use this information to construct a least-squares estimator

$$\boldsymbol{\beta}^*_{T_0} = \left(\mathbf{X}'_{T_0}\mathbf{X}_{T_0}\right)^{-1}\mathbf{X}'_{T_0}Y = \boldsymbol{\beta} + \left(\mathbf{X}'_{T_0}\mathbf{X}_{T_0}\right)^{-1}\mathbf{X}'_{T_0}\boldsymbol{\varepsilon}$$

with mean standard error (MSE) given by $E\left\|\boldsymbol{\beta}^*_{T_0} - \boldsymbol{\beta}\right\|_2^2 = \sigma^2 \text{trace}\left[\left(\mathbf{X}'_{T_0}\mathbf{X}_{T_0}\right)^{-1}\right] \geq \sigma^2/(1+\delta_s)$s. This confirms the claim that the DS achieves the oracle MSE up to the factor $\log(p)\,C\,(1+a)$.

Though a remarkable result, (2.6) may be in some instances a little misleading, as in the revealing case where $\beta_j \ll \sigma$ for all coordinates j of $\boldsymbol{\beta}$. Since, by setting $\boldsymbol{\beta} = 0$, the squared error loss simply becomes $\sum_j \beta_j^2$, which is potentially much smaller than the number of nonzero covariates times the variance. This simple example corresponds to a setting where the variance is clearly much greater than the squared bias. A less extreme and more common situation often encountered in data, is one where some components of $\boldsymbol{\beta}$ exceed the noise level but most do not. To reduce the squared error loss an optimal estimator must now carefully trade off squared bias with variance. This can be done, for instance by estimating small coefficients ($\beta_j < \sigma$) with exactly zero, which results in small squared bias β_j^2 with no associated variance, while unbiasedly estimating large coefficients ($\beta_j > \sigma$) and therefore incurring a variance of σ^2 with each estimated component. One can easily show that this strategy yields an estimator with risk $\sum_j \min\left(\beta_j^2, \sigma^2\right) = \min_{I\subset\{1,\ldots,p\}} \|\boldsymbol{\beta} - \boldsymbol{\beta}_I\|^2 + |I|\,\sigma^2$, which has a natural interpretation in terms of least squared bias and variance. Moreover, one can show that this risk also has an interesting relationship with that of an ideal but infeasible least squares estimator $\boldsymbol{\beta}_{I^*} = \left(\mathbf{X}'_{I^*}\mathbf{X}_{I^*}\right)^{-1}\mathbf{X}'_{I^*}Y$, where $I^* = \underset{I\subset\{1,\ldots,p\}}{\arg\min}\left\|\boldsymbol{\beta} - \left(\mathbf{X}'_I\mathbf{X}_I\right)^{-1}\mathbf{X}'_I Y\right\|^2$. More specifically Candes and Tao proved that $E\,\|\boldsymbol{\beta}_{I^*} - \boldsymbol{\beta}\| \geq 1/2\left[\sum_j \min\left(\beta_j^2, \sigma^2\right)\right]$, so that the "ideal" risk among least squares estimators is bounded below by the risk that in a sense minimizes both squared bias and variance. On the basis of this last

observation, it is possible to improve on (2.6) by showing that if $\boldsymbol{\beta}$ is $s-$sparse, and $\mathbf{X}$ satisfies $\delta_s + \theta_{s,2s} < 1-t$ for $t > 0$, then the DS with $\lambda := \sqrt{2\log p}$ obeys:

$$\left\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right\|_2^2 \leq C\log(p)\left(\sigma^2 + \sum_j \min\left(\beta_j^2, \sigma^2\right)\right), \tag{2.8}$$

where $C > 0$ depends on $\delta_s, \theta_{s,2s}$. In other words, the estimator nearly adapts to the ideal risk achieved by a least squares estimator that minimizes both squared bias and variance in model (2.1). Due to space limitations, we refer the reader to Candes and Tao (2007) for more on oracle inequalities as well as for an overview of efficient fitting procedures for the DS.

2.5 Prediction and Persistence

In preceding sections, we have focused on model consistency and estimation error as measures of performance of a given procedure. Another optimality paradigm widely used in statistics is that of model prediction. The classical example of prediction mean squared error has a long history in the model selection literature, and has recently been used to study sparse estimators. In fact, Greenshtein and Ritov (2004) and Greenshtein (2006) discuss linear models (2.1) from a prediction loss perspective that departs from the assumption of the existence of a "true" model; instead they consider predictor selection in a framework where the number of potential predictors increases with the sample size; it is in this setting that they introduce the concept of "persistence," which we now discuss. Let $\mathbf{z}_i = (y_i, x_{i1}, \ldots, x_{ip})$, $i = 1, \ldots, n$, be i.i.d. random vectors with $\mathbf{z}_i \sim F_n$. The goal is to predict $\mathbf{Y}$ by a linear combination of columns of $\mathbf{X}$, i.e., $\sum \beta_j \mathbf{x}_j$, where $(\beta_1, \ldots, \beta_p) \in B^n$, and B^n are restricted by the maximum of n nonzero coefficients or the L_1-norm of the coefficients. The setup allows for a number of predictors p much larger than sample size (i.e., $p = n^\alpha, \alpha > 1$). Denote

$$L_F(\boldsymbol{\beta}) = E_F\left(Y - \sum_{j=1}^{p} \beta_j \mathbf{x}_j\right)^2. \tag{2.9}$$

Let $\boldsymbol{\beta}_{F_n}^\star = \arg\min_{\boldsymbol{\beta} \in B_n} L_{F_n}(\boldsymbol{\beta})$. Given a sequence of sets of predictors B_n, the sequence of procedures $\widehat{\boldsymbol{\beta}}^n$ is called *persistent* if, for every sequence F_n,

$$L_{F_n}(\widehat{\boldsymbol{\beta}}^n) - L_{F_n}(\boldsymbol{\beta}_{F_n}^\star) \xrightarrow{P} 0.$$

Greenshtein and Ritov (2004) argue that by using the distance between the $L(\cdot)$ functions, and not between the regression coefficients, they are directly targeting the problem of prediction, and not of estimation. In fact, an immediate advantage of this approach is that the collinearity of the columns of X is of no relevance, since

only the linear combination of covariates is evaluated. In addition, this approach is of particular appeal in situations where $L_{F_n}(\boldsymbol{\beta}^{\star}_{F_n})$ actually does not converge to 0, which is not uncommon. Greenshtein and Ritov (2004) study two types of sets $B^n \subset \mathbb{R}^p$. The first type contains vectors $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$ with at most $k(n)$ nonzero entries ("model selection" type), and the second contains vectors $\boldsymbol{\beta}$ with an L_1-norm of $\boldsymbol{\beta}$ less than $b(n)$ ("LASSO" type). Their results for LASSO-type methods are loosely described as follows; under boundedness assumptions on certain second and third moments of the data, for any sequence $B^n \subset \mathbb{R}^p$, where B^n consists of all vectors with L_1 norm less than $b(n) = o((n/log(n)))^{1/4}$, they prove that there exist a persistent sequence of procedures. In fact, they show that $\widehat{\boldsymbol{\beta}}^n$, the minimizer of (2.9) on the subsets $B_n \subset \mathbb{R}^p$, is persistent.

Despite these remarkable findings, there is growing evidence that model consistency and optimal model prediction may not always be reconcilable, particularly if prediction performance is judged from a minimax perspective. The manuscript by Yang (2007) provides interesting results and a revealing discussion on this subject. In the next section, we tackle the related problem of post-model selection inference.

2.6 Difficulties with Post-Model Selection Inference

Hitherto, our exposition on model selection methods for (2.1) with potentially high-dimensional covariates has emphasized the pointwise behavior of sparse estimators that yield only few nonzero estimated coefficients, while saying little of these estimators' performance uniformly over the model. Specifically, oracle properties that suggest one can essentially adapt to the true sparsity of the regression model are statements about the error and other properties of sparse estimators evaluated in a pointwise fashion (i.e., at an unknown fixed parameter value $\boldsymbol{\beta}$ at a time), and fail to provide an assessment of the proposed methods uniformly over the model (maximal error over all allowed parameter values of $\boldsymbol{\beta}$). A uniform (over the posited model) assessment of an estimation procedure is fundamental to statistical inference and is broadly recognized as yielding a more realistic evaluation of the overall performance of a proposed method, particularly with some of the poignant questions the statistician must routinely face, for instance: "How large an n is sufficient for asymptotics to yield a good approximation to the finite sample behavior of $\widehat{\boldsymbol{\beta}}$ independently of the value of $\boldsymbol{\beta}$?" This last question is especially relevant when the analysis goal includes the construction of valid asymptotic confidence sets for a subset of nonzero estimates of regression coefficients (as confidence sets are defined uniformly over the model).

Next, we point out an important pitfall related to the concept of the oracle property as it pertains to model selection and argue that there are unbridgeable differences between model-selection via sparsity and sound statistical inference on selected parameters. We emphasize that the problem is neither due to the high dimensionality of the data nor to any particular method used to obtain a sparse solution, but rather to the use of sparsity as a vehicle for model selection.

To proceed, we denote by $m(\boldsymbol{\beta})$ the indicator vector with components $m_j(\boldsymbol{\beta}) = 1$ if $\beta_j \neq 0$ and $m_j(\boldsymbol{\beta}) = 0$ if $\beta_j = 0$ and restrict attention to estimators $\widehat{\boldsymbol{\beta}}$ known to satisfy the following sparsity condition; for all $\boldsymbol{\beta}$ in $\mathbb{R}^p$,

$$\Pr\left\{m\left(\widehat{\boldsymbol{\beta}}\right) \leq m(\boldsymbol{\beta})\right\} \underset{n\to\infty}{\to} 1.$$

Evidently, the previous display is satisfied by the so-called consistent model selection procedures that meet the stronger condition

$$\Pr\left\{m\left(\widehat{\boldsymbol{\beta}}\right) = m(\boldsymbol{\beta})\right\} \underset{n\to\infty}{\to} 1.$$

Let the scaled mean squared error of $\widehat{\boldsymbol{\beta}}$ be defined as

$$\mathrm{MSE}_n\left(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}\right) = E_{n,\boldsymbol{\beta}}\left[n\left(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)'\left(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\right)\right].$$

Leeb and Pötscher (2008a) show that

$$\sup_{\boldsymbol{\beta}\in\mathbb{R}^p} \mathrm{MSE}_n\left(\widehat{\boldsymbol{\beta}}, \boldsymbol{\beta}\right) \to \infty,$$

in short, the maximal scaled MSE of any sparse estimator diverges to infinity with increasing sample size. This result can be generalized to other loss functions (Leeb and Pötscher, 2008a) and remains true even under conditions where ordinary least squares has bounded maximal scaled MSE (well-conditioned covariate design matrix).

A regular estimator is roughly a locally uniform estimator of which the scaled mean squared error with respect to Pitman alternatives does not blow up. The lack of uniformity in the performance of sparse estimators, which are no longer regular, becomes quite apparent under "Pitman" parameter values local to zero, such as under the model sequence $\boldsymbol{\beta}_n = C/\sqrt{n}$ (where C is a constant). For large n, $\boldsymbol{\beta}_n$ lies in the neighborhood of zero and thus is likely to be estimated as exactly zero by a given sparse procedure. The resulting unbounded, scaled squared bias C^2 reflects an additional cost associated with the use of sparsity, typically not captured by oracle statements; so that we may conclude that any such statement must be interpreted carefully, as contrary to the popular belief, sparse estimators do not truly perform (not even nearly) as if the underlying parsimonious regression model was known ahead of time. It also follows that any confidence interval centered around a nonzero component of a sparse estimator will fail to cover the truth at its nominal level uniformly over the model, and would therefore not be valid.

Other important implications for post-model selection (via sparsity) can be found in a series of manuscripts by Leeb and Potscher (2005, 2006, 2008b); see also, Kabaila and Leeb (2006), and Yang (2005).

2.7 Penalized Likelihood for Generalized Linear Models

As mentioned in Section 2.1, the penalized least squares criterion can be extended to discrete response variables. Let $g(x)$ be a known inverse link function. We define a penalized log-likelihood function as

$$\mathcal{L}_P(\boldsymbol{\beta}) = \sum_{i=1}^{n} \log f(g(\mathbf{x}_i'\boldsymbol{\beta}, y_i)) - \sum_{j=1}^{p} P_\lambda(\boldsymbol{\beta}),$$

where $f(\cdot)$ is a density of y_i conditional on $\mathbf{x}_i$. The above penalized likelihood includes among others logistic regression and Poisson regression. Again, maximizing the above likelihood function with, for example, a LASSO penalty, results in sparse solutions for $\widehat{\boldsymbol{\beta}}$.

2.8 Simulation Study

In order to evaluate the performance of model selection procedures with a badly conditioned design matrix $\mathbf{X}$, we performed a Monte Carlo simulation study of the LASSO applied to the linear model (2.1) with s nonzero coefficients.

We follow the general setup of the simulation studies in Donoho and Stodden (2006) and summarize our simulation results in a *Phase Diagram* (Donoho and Stodden, 2006).

The simulation data are generated as follows.

1. Generate data from a model $\mathbf{Y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\beta} = [\boldsymbol{\beta}_s, \boldsymbol{\beta}_{p-s}]$, $s < p$ with $\boldsymbol{\beta}_s = [\beta_1, \ldots, \beta_s]$ and $\beta_j \neq 0$ for $j = 1, \ldots, s$, $\boldsymbol{\beta}_{p-s} = [\beta_{s+1}, \ldots, \beta_p]$ and $\beta_j = 0$ for $s < j \leq p$, $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{I}\sigma^2)$ with $\sigma = 2$.
2. Run a LASSO model selection procedure to estimate $\widehat{\boldsymbol{\beta}}$,
3. Evaluate the performance of the procedure by

$$L_2 = \frac{||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}||_2}{||\boldsymbol{\beta}||_2}. \tag{2.10}$$

We selected a compound symmetry correlation structure for the columns of $\mathbf{X}$, i.e., $\text{corr}(\mathbf{x}_i, \mathbf{x}_j) = \rho$ for $i, j \in \{1, \ldots, p\}$. In our simulations, we set $p = 100$, β_j was generated from $Unif(0, 100)$ for $j \in \{1, \ldots, s\}$, and ρ was chosen as one of $(0, 0.3, 0.6, 0.9)$.

Define the level of indeterminateness as n/p and the sparsity level as s/n. We evaluate the performance of variable selection procedures as a function of $\delta = n/p$ and $\tau = s/n$. Figure 2.1 displays the average normalized L_2 errors for 50 simulated data sets for each combination of indeterminateness(δ) and sparsity (τ). The areas of the Figure 2.1 in dark blue indicate the region where the LASSO procedure recovered the true $\boldsymbol{\beta}$'s with the error close to zero. Other colors above the diagonal show the area where the model selection procedure was unable to provide good estimates of the regression coefficients.

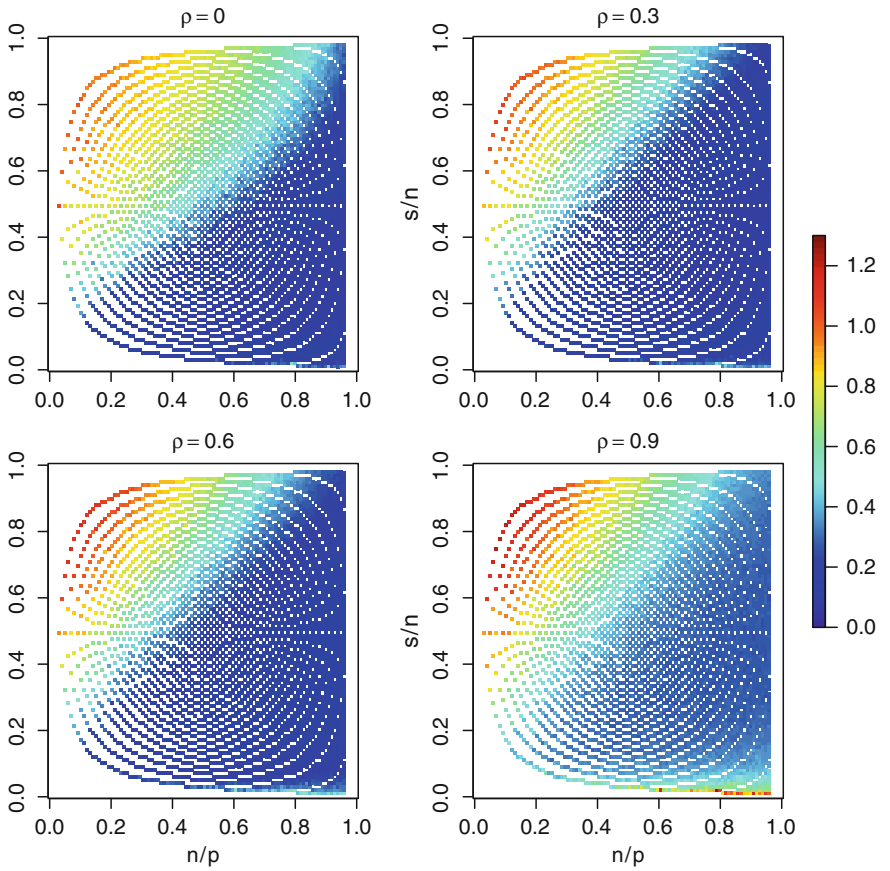


Fig. 2.1 Phase transition diagram with the sparse model recovered by LASSO with the tuning parameter selected by AIC. The number of variables is kept constant at $p = 100$. Columns of $\mathbf{X}$ exhibit compound symmetry correlation structure with $\rho = 0, .3, .6$, and .9.

Figure 2.2 displays the differences in the estimation errors between the models with correlated columns of $\mathbf{X}$ and the independent case (left panel) and between the models with correlated columns of $\mathbf{X}$ with increasing correlations (right panel). The performance of LASSO in the region of good signal recovery (approximately below the main diagonal) deteriorates as the correlation ρ increases. Surprisingly, the errors are lower for the $\mathbf{X}$ with correlated columns in the region immediately above the diagonal. The errors worsen with the increasing correlation between the columns of $\mathbf{X}$. This is especially visible in the lower right corner of Figure 2.2 for the transition from $\rho = 0.3$ to $\rho = 0.9$. Additionally, the transition of the model selection from penalized methods to combinatorial search happens at an earlier stage for the nonorthogonal $\mathbf{X}$.

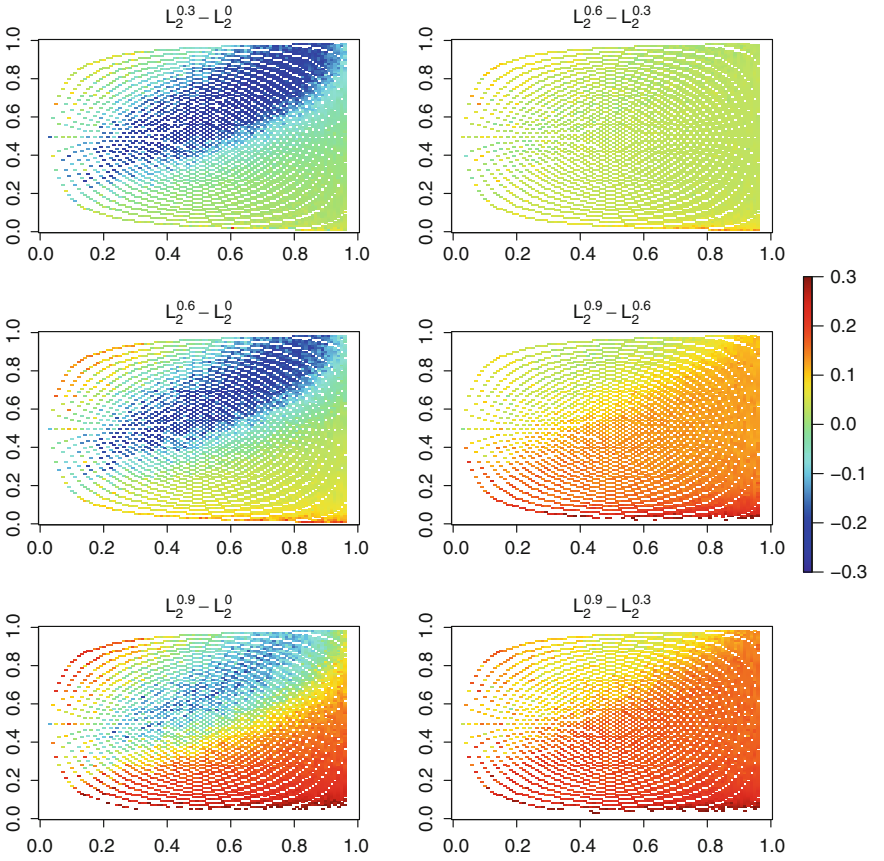


Fig. 2.2 Differences between the scaled estimation bias for the models estimated with LASSO, and the design matrix $\mathbf{X}$ exhibiting compound symmetry correlation structure with $\rho = 0, .3, .6,$ and .9. The notation L_2^{ρ} indicates the the scaled bias L_2 with the correlation ρ.

2.9 Application of the Methods to the Prostate Cancer Data Set

To illustrate the variable selection techniques we discussed in Sections 2.2.3-2.4, we consider a regression problem where we model the log(cancer volume) using other variables in the prostate cancer data set, including their power terms and interactions ($p = 43$).

First, a linear regression was performed with an L_1 constraint imposed on the coefficients (Tibshirani, 1996), using the `R` package `lars`. The L_1 constraint expressed as a proportion of the OLS solution $|\boldsymbol{\beta}|/\max|\boldsymbol{\beta}| = 0.0325$ was selected using the 20-fold CV. Next, we used an adaptive LASSO based on the algorithm of Zou (2006), with the weights set as either the inverses of the OLS coefficient

estimates, or the ridge regression estimates. Finally, we used the Dantzig selector using the R function dd developed by James and Radchenko (2008).

The LASSO solution contained 18 nonzero coefficients, while the Adaptive LASSO solution had more coefficients shrunk to be zero, resulting in 8 (OLS) and 15 (ridge) nonzero coefficient estimates. Solution according to the Dantzig selector contained 14 nonzero coefficients.

Interestingly, only one covariate was chosen by all four considered fitting methods, lcp:lpsa (the interaction term of log(capsular penetration) and log(prostate specific antigen)). There was more agreement between the three methods based on LASSO where the same covariate was chosen in four instances. The summary of the results is presented in Table 2.1.

Table 2.1 Nonzero coefficients for the model relating the log(cancer volume) and explanatory covariates in the Prostate cancer data set.

Covariate	LASSO	ALas (OLS)	ALas (RR)	Dantzig
age	0.0166	0.0248	0.2583	0
lbph	0	−0.0827	−0.2062	0
svi	0	0	−0.0335	0
lcp	0.1395	0.5121	0.8905	0
gleason	0.0622	0.1385	0	0
lpsa	0.4272	0.5463	0	0.6483
lweight^2	−0.0080	0	0	−0.0132
lbph^2	0.0390	0	0	0.0546
lcp^2	0.0420	0	0	0.0530
lpsa^2	0	0	0.1066	0
lweight:age	0	−0.0013	−0.0380	0
lweight:lbph	0	0	0.0282	0
lweight:lcp	0.0982	0	0	0.1651
lweight:gleason	0	0	0.3618	0
age:lbph	−0.0009	0	0	0
age:gleason	0	0	−0.0161	0.0019
age:pgg45	0	−0.0001	−0.0001	0
age:lpsa	0	0	0	−0.0006
lbph:svi	0.0572	0	0	0.1069
lbph:pgg45	−0.0012	0	−0.0007	−0.0023
lbph:lpsa	−0.0072	0	0	−0.0238
svi:lcp	0	0	0.2987	0
svi:pgg45	−0.0024	0	−0.0084	0
svi:lpsa	0	0	0	−0.0191
lcp:pgg45	0.0024	0	0.0054	0.0026
lcp:lpsa	−0.0852	−0.0620	−0.2737	−0.1206
gleason:lpsa	0.0183	0	0	0
pgg45:lpsa	0	0	0	−0.0023

2.10 Conclusion

Model selection methods proposed in the past 15 years provide an exciting array of choices. However, the selection of an appropriate method and associated tuning parameters depends on the problem at hand. It is important to remember that the selection of a model for predictive accuracy versus sparsity are two very different and possibly irreconcilable goals. As an off-the-shelf method, LASSO provides a computationally convenient and reasonable method for choosing among models. Several extensions to the LASSO might be better suited to specific setups. For instance, SCAD is a promising method that achieves sparsity with lower bias in the nonzero estimated coefficients than LASSO; however, it remains computationally more challenging. The Dantzig selector, a more recent addition to the model selection literature has recently drawn much attention. Another important, but often neglected issue is that of post-model selection inference where a recent emergence of work points to unresolvable difficulties with making inferences through the use of sparsity-based procedures.

References

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, 37:373–384.

Candes, E. and Tao, T. (2007). The Dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2312–2351.

Donoho, D. and Stodden, V. (16-21 July 2006). Breakdown point of model selection when the number of variables exceeds the number of observations. In *IJCNN '06. International Joint Conference on Neural Networks, 2006.*, pages 1916–1921.

Donoho, D. L., Elad, M., and Temlyakov, V. N. (2006). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–499.

Efron, B., Hastie, T., and Tibshirani, R. (2007). Discussion of "the Dantzig selector: statistical estimation when p is much larger than n." *The Annals of Statistics*, 35(6):2358–2364.

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

Fan, J. and Li, R. (2006). Statistical challenges with high dimensionality: feature selection in knowledge discovery. In Sanz-Sole, M., Soria, J., Varona, J., and Verdera, J., editors, *Proceedings of the International Congress of Mathematicians*, volume III, pages 595–622. European Mathematical Society, Zurich.

Fu, W. J. (1998). Penalized regressions: The bridge versus the Lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416.

Furnival, G. M. and Wilson, R. W. J. (1974). Regressions by leaps and bounds. *Technometrics*, 16(4):499–511.

Greenshtein, E. (2006). Best subset selection, persistence in high-dimensional statistical learning and optimization under l_1-constraint. *The Annals of Statistics*, 34(5):2367–2386.

Greenshtein, E. and Ritov, Y. (2004). Persistence in high-dimensional predictor selection and the virtue of overparametrization. *Bernoulli*, 10:971–988.

Hoerl, A. and Kennard, R. (1970). Ridge regression: biased estimation for non-orthogonal problems. *Technometrics*, 12:55–68.

Huang, J., Ma, S., and Zhang, C.-H. (2007). Adaptive Lasso for sparse high-dimensional regression models. Technical report, The University of Iowa.

James, G. and Radchenko, P. (2008). A generalized Danztig selector with shrinkage tuning. http://www-rcf.usc.edu/∼gareth/GDS.pdf.

Kabaila, P. and Leeb, H. (2006). On the large-sample minimal coverage probability of confidence intervals after model selection. *Journal of the American Statistical Association*, 101:619–629.

Knight, K. and Fu, W. (2000). Asymptotics for Lasso-type estimators. *The Annals of Statistics*, 28:1356–1378.

Leeb, H. and Pötscher, B. (2005). Model selection and inference: Facts and fiction. *Econometric Theory*, 21:21–59.

Leeb, H. and Pötscher, B. (2006). Can one estimate the conditional distribution of post-model-selection estimators? *Annals of Statistics*, 34:2554–2591.

Leeb, H. and Pötscher, B. (2008a). Can one estimate the unconditional distribution of post-model-selection estimators? *Econometric Theory*, 24:338–376.

Leeb, H. and Pötscher, B. (2008b). Sparse estimators and the oracle property, or the return of Hodges' estimator. *Journal of Econometrics*, 142:201–211.

Leng, C., Lin, Y., and Wahba, G. (2006). A note on the Lasso and related procedures in model selection. *Statistica Sinica*, 16(4):1273–1284.

Miller, A. (2002). *Subset Selection in Regression*. Chapman & Hall/CRC, London.

Paul, D., Bair, E., Hastie, T., and Tibshirani, R. (2008). Pre-conditioning for feature selection and regression in high-dimensional problems. *The Annals of Statistics*, 36:1595–1618.

Stamey, T. A., Kabalin, J. N., McNeal, J. E., Johnstone, I. M., Freiha, F., Redwine, E. A., and Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients. *The Journal of Urology*, 141(5):1076–1083.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B, Methodological*, 58:267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused Lasso. *Journal of the Royal Statistical Society, Series B, Methodological*, 67:91–108.

Yang, Y. (2005). Can the strengths of AIC and BIC be shared? *Biometrika*, 92:937–950.

Yang, Y. (2007). Prediction/estimation with simple linear model: Is it really that simple? *Econometric Theory*, 23:1–36.

Yuan, M. and Lin, Y. (2007). On the non-negative garrote estimator. *Journal of the Royal Statistical Society: Series B*, 69(2):143–161.

Zhao, P. and Yu, B. (2006). On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2567.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 67(2):301–320.

Chapter 3
Multivariate Nonparametric Regression

Charles Kooperberg and Michael LeBlanc

As in many areas of biostatistics, oncological problems often have multivariate predictors. While assuming a linear additive model is convenient and straightforward, it is often not satisfactory when the relation between the outcome measure and the predictors is either nonlinear or nonadditive. In addition, when the number of predictors becomes (much) larger than the number of independent observations, as is the case for many new genomic technologies, it is impossible to fit standard linear models. In this chapter, we provide a brief overview of some multivariate nonparametric methods, such as regression trees and splines, and we describe how those methods are related to traditional linear models. Variable selection (discussed in Chapter 2) is a critical ingredient of the nonparametric regression methods discussed here; being able to compute accurate prediction errors (Chapter 4) is of critical importance in nonparametric regression; when the number of predictors increases substantially, approaches such as bagging and boosting (Chapter 5) are often essential. There are close connections between the methods discussed in Chapter 5 and some of the methods discussed in Section 3.8.2. In this chapter, we will briefly revisit those topics, but we refer to the respective chapters for more details. Support vector machines (Chapter 6), which are not discussed in this chapter, offer another approach to nonparametric regression.

We start this chapter by discussing an example that we will use throughout the chapter. In Section 3.2 we discuss linear and additive models. In Section 3.3 we generalize these models by allowing for interaction effects. In Section 3.4 we discuss basis function expansions, which is a form in which many nonparametric regression methods, such as regression trees (Section 3.5), splines (Section 3.6) and logic regression (Section 3.7) can be written. In Section 3.8 we discuss the situation in which the predictor space is high dimensional. We conclude the chapter with discussing some issues pertinent to survival data (Section 3.9) and a brief general discussion (Section 3.10).

C. Kooperberg
Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, 1100 Fairview Ave N, M3-A410 Seattle, WA 98109-1024, USA
email: clk@fhcrc.org

X. Li, R. Xu (eds.), *High-Dimensional Data Analysis in Cancer Research*, Applied Bioinformatics and Biostatistics in Cancer Research, DOI 10.1007/978-0-387-69765-9_3,

3.1 An Example

We illustrate the methods in this chapter using data from patients diagnosed with multiple myeloma, a cancer of the plasma cells found in the bone marrow. The data were obtained from three consecutive clinical trials evaluating aggressive chemotherapy regiments in conjunction with autologous transplantation conducted at the Myeloma Institute for Research and Therapy, University of Arkansas for Medical Sciences (Barlogie et al., 2006). The outcome for patients with myeloma is known to be variable and is associated with clinical and laboratory measures (Greipp et al., 2005). In this data set, potential predictors include several laboratory measures measured at the baseline of the trials, age, gender, and genomic features, including a summary of cytogenetic abnormalities and approximately 350 single nucleotide polymorphisms (SNPs) for candidate genes representing functionally relevant polymorphisms playing a role in normal and abnormal cellular functions, inflammation, and immunity, as well as for some genes thought to be associated with differential clinical outcome response to chemotherapy.

In most of our analysis we analyze the binary outcome whether there was disease progression after 2 years, using the laboratory measures, age, and gender as predictors. In Sections 3.7 and 3.8 we also analyze the SNP data; in Section 3.9 we analyze time to progression and survival using a survival analysis approach.

3.2 Linear and Additive Models

Let Y be a numerical response, and let $\mathbf{x} = (x_1, \ldots, x_p)'$ be a set of predictors spanning a covariate space $\mathcal{X}$. We assume that the regression model Y takes the form of a generalized linear model

$$g(E(Y|\mathbf{x})) = \eta(\mathbf{x}), \tag{3.1}$$

where $g(\cdot)$ is some appropriate link function and

$$\eta(\mathbf{x}) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i. \tag{3.2}$$

In this chapter, we mostly assume that Y is a continuous random variable and that $g(\cdot)$ is the identity function so that (3.1) is a linear regression model or that Y is a binary random variable and that $g(\cdot)$ is the logit function so that (3.1) is a logistic regression model, but most of the approaches that are discussed in this chapter are also applicable to other generalized linear models. In Section 3.9, we discuss some modifications that make these approaches applicable to survival data.

Estimation via the method of maximum likelihood (or least squares) is well established. Many nonparametric regression methods generalize the model in (3.2).

In particular, we can replace the linear functions x_i in (3.2) by smooth nonlinear functions $f_i(x_i)$. Now (3.2) becomes

$$\eta(\mathbf{x}) = \beta_0 + \sum_{i=1}^{k} f_i(x_i). \tag{3.3}$$

The functions $f_i(\cdot)$ are usually obtained by local linear regression (loess, e.g., Loader, 1999) or smoothing splines (e.g., Green and Silverman, 1994). The model (3.3) is known as a generalized additive model (Hastie and Tibshirani, 1990).

3.2.1 Example Revisited

Of the 778 subjects with complete covariate data in the multiple myeloma data, 171 subjects had progressed after 2 years while 570 subjects had not. Another 37 subjects were censored sufficiently early that we chose not to include them in our analysis to retain a binary regression strategy. These 37 subjects are included in the survival analysis (Section 3.9). We used nine predictors: age, gender, lactate dehydrogenase (ldh), C-reactive protein (crp), hemoglobin, albumin, serum β_2 microglobulin (b2m), creatnine, and anyca (an indicator of cytogenetic abnormality). The transformed values of ldh, crp, b2m, and creatnine on the logarithmic scale were used in the analysis. In a linear logistic regression model, anyca has a Z-statistic of 4.8 ($p = 10^{-6}$), log(b2m) has a Z-statistic of 2.7 ($p = 0.007$), and log(ldh), albumin, and gender are significant at levels between 0.02 and 0.04.

We then proceeded to fit a generalized additive model, using a smoothing spline to model each of the continuous predictors. We used the R-function `gam()`, which selects the smoothing parameter using generalized cross-validation, and provides approximate inference over the "significance" of the non-nonlinear components. Three predictors were deemed significantly nonlinear at $p = 0.05$: age, log(crp), and log(b2m), all at significance levels between 0.015 and 0.05. Note that these significance levels are approximate, and they should be treated with caution. The most interesting significant nonlinearity was probably in log(b2m). In Figure 3.1 we show the fitted component with a band of width twice the approximate standard errors. It appears that the effect of log(b2m) is only present when log(b2m) is above 1, which is approximately the median in our data set.

3.3 Interactions

Nonadditive regression models (models for $\eta(\mathbf{x})$ containing effects of interactions between predictors) occur frequently in oncology. Such models may be needed because additive models, as discussed above, may not provide an accurate fit to the data, but they may also be of interest to answer specific questions. For example, models containing interactions may be used to identify groups of patients that are

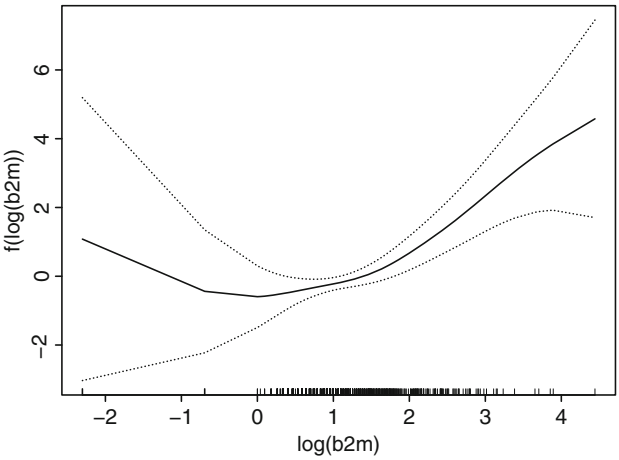


Fig. 3.1 The component of log(serum β_2 microglobulin) in the generalized additive model for progression after 2 years in the multiple myeloma data.

at especially high or low risk (e.g., LeBlanc et al., 2005), they may be of interest to identify subgroup effects in clinical trials (e.g., Singer, 2005), or to identify gene $\times$ environment interactions (e.g., Board on health sciences policy, 2002).

In the following several sections, we will discuss general models for interactions in a regression context. There are, however, special cases in which dedicated methods are more appropriate. For example, if the goal is to only identify patients at especially high risk, we may not feel a need to model the risk (regression function) for patients at low risk accurately (LeBlanc et al., 2006). When we know that some predictors are independent of each other, as is sometimes the case for gene $\times$ environment interactions or for nested case–control studies within clinical trials, more efficient estimation algorithms are possible (Dai et al., 2008). We will not discuss these situations in this chapter.

The most straightforward interaction model is to include all linear interactions up to a particular level in model (3.2); for example, a model with two- and three-level interactions is

$$\eta(\mathbf{x}) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{1 \le i < j \le k} \beta_{ij} x_i x_j + \sum_{1 \le i < j < l \le k} \beta_{ijl} x_i x_j x_l.$$

It is clear that with this approach the number of coefficients becomes very large quickly. The problems that this causes are even worse when we generalize the smooth model (3.3). This explosion of the size of the model is sometimes known as the "curse of dimensionality," and it can be formalized by establishing the convergence rates of parameters in such models under appropriate conditions (Stone, 1994). Instead we may want to include only those interactions that are really needed to accurately model the regression function $\eta(\mathbf{x})$. Often this is done using some form of stepwise regression. It turns out that approach can be generalized conveniently using a basis function approach.

3.4 Basis Function Expansions

The linear model (3.1) can also be used as the starting point for nonlinear, nonadditive, multivariate regression methods. Assume that the regression function $\eta(\mathbf{x})$ is in some p-dimensional linear space $\mathcal{B}(\mathcal{X})$, and let $B_1(\mathbf{x}),\ldots,B_p(\mathbf{x})$ be a basis for $\mathcal{B}(\mathcal{X})$. Then we can write

$$\eta(\mathbf{x}) = \sum_{i=1}^{p} \beta_i B_i(\mathbf{x}). \tag{3.4}$$

For a given set of basis functions $B_1(\cdot),\ldots,B_p(\cdot)$ estimation in (3.4) is a straightforward extension of (3.2).

Several nonparametric multivariate regression methodologies use a basis function approach, but rather than fixing the space $\mathcal{B}(\mathcal{X})$ these approaches select the space at the same time as the coefficients of the basis functions are estimated. Three of the methodologies that are discussed later in this chapter use this approach.

- Regression tree methods, such as classification and regression trees (CART, Breiman et al., 1984). The basis functions that are used for tree methods are indicator functions corresponding to rectangular regions of the predictor space. Tree methods are discussed in Section 3.5.
- Multivariate adaptive regression splines (MARS, Friedman, 1991) and related spline methods (e.g., Kooperberg et al., 1995; Stone et al., 1997). The basis functions that are used for MARS and related methods are piecewise polynomials (splines) and their tensor products. We discuss spline methods in Section 3.6.
- Logic regression (Ruczinski et al., 2003) is discussed in Section 3.7. The basis functions that are used for logic regression are Boolean combinations of binary predictors.

Stepwise regression methods provide useful tools for model selection using basis functions. As an example, suppose that we consider two linear spaces to model $\eta(\mathbf{x})$: a p-dimensional space $\mathcal{B}^p(\mathcal{X})$ that is a sub-space of a $(p+1)$-dimensional space $\mathcal{B}^{p+1}(\mathcal{X})$. After we fit model (3.4) using basis functions for the smaller space $\mathcal{B}^p(\mathcal{X})$ we can compute a score test (Rao statistic, Rao, 1973) to evaluate how much better $\eta(\mathbf{x})$ would be modeled if we would require that $\eta(\mathbf{x}) \in \mathcal{B}^{p+1}(\mathcal{X})$ instead. Similarly, after we fit model (3.4) using basis functions for the larger space $\mathcal{B}^{p+1}(\mathcal{X})$ we can compute a Wald statistic to evaluate how much worse $\eta(\mathbf{x})$ would be modeled if we would require that $\eta(\mathbf{x}) \in \mathcal{B}^p(\mathcal{X})$. If these would be prespecified spaces the score and Wald statistics could be compared with standard parametric distributions, similar to what is done in stepwise variable selection methods (see Chapter 2). The adaptivity of these approaches does typically require other approaches to obtain significant levels and prediction errors though (see Chapter 4).

We can generalize this stepwise procedure to an algorithm for stepwise model building, that is used both in tree and in spline methods.

1. Start with modeling $\eta(\mathbf{x}) \in \mathcal{B}_a^p$. A common situation is that $p = 1$ and $\mathcal{B}_a^1$ consists of only constant functions.
2. Stepwise addition: replace $\mathcal{B}_a^p$ by a $(p+1)$-dimensional space $\mathcal{B}_a^{p+1}$ of which $\mathcal{B}_a^p$ is a subspace by considering a (large) set of candidate spaces $\mathcal{B}_a^{p+1}$ that satisfy some method-dependent regularity conditions. Select the "best" $\mathcal{B}_a^{p+1}$ for example, by selecting the $\mathcal{B}_a^{p+1}$ corresponding to the largest score statistic.
3. Continue adding dimensions until either a prespecified dimension p^* is reached, or until the improvement in the fit between successive models becomes very small.
4. Set $\mathcal{B}_d^{p^*} = \mathcal{B}_a^{p^*}$.
5. Proceed with stepwise deletion: replace $\mathcal{B}_d^p$ by a $(p-1)$-dimensional subspace $\mathcal{B}_d^{p-1}$ that satisfies some method-dependent regularity conditions. Select the "best" $\mathcal{B}_d^{p-1}$, for example, by selecting the candidate corresponding to the smallest Wald statistic.
6. Continue until p reaches some minimum dimension (e.g., $p = 1$).
7. Out of all the linear spaces considered $\mathcal{B}_a^1, \ldots, \mathcal{B}_a^{p^*} = \mathcal{B}_d^{p^*}, \ldots, \mathcal{B}_d^1$, select one either using some penalized likelihood like the Akaike information criterion (Akaike, 1974) or the Bayesian information criterion (BIC, Schwarz, 1978), or an honest method to estimate the prediction error, such as cross-validation.

3.5 Regression Tree Models

3.5.1 Background

Regression and classification trees are primarily known for their easy-to-understand geometric representation. While a binary regression tree provides a simple description of groups of subjects, the model can also be cast in a regression spline form similar to the methods presented in Section 3.6. The CART algorithm (Breiman et al., 1984) is probably the best-known implementation of tree-based methods in the statistical literature and generally motivates the basics given in this section. There has also been extensive research of tree-structured methods in machine learning, for instance the C4.5 algorithm of Quinlan (1993). When extended to survival data, regression trees have found a significant following in medicine because the sequence of binary decisions leads to simple representation for prognostic groups of patients treated in a similar fashion. Most tree-based methods for survival data have adopted at least some aspects of the CART algorithm (Gordon and Olshen, 1985; Ciampi et al., 1986; Segal, 1988; LeBlanc and Crowley, 1993). Some recent examples in survival analysis using regression trees include Greipp et al. (2005), London et al. (2005), Farag et al. (2006), and Gimotty et al. (2007).

3.5.2 Model Building

3.5.2.1 Model Basis Set as Partition Function

A tree model can be represented as a binary tree T, where the set of terminal nodes $\widetilde{T}$ corresponds to the partition of the covariate space $\mathcal{X}$ into a number of $M(\widetilde{T})$ disjoint subsets. A tree model can also be expressed by a basis function representation

$$\eta(\mathbf{x}) = \sum_{h \in \widetilde{T}} \eta_h B_h(x),$$

(compare with (3.4)) where $B_h(x) = I\{x \in R_h\}$, R_h is the region corresponding to a terminal node h, and η_h is a vector of parameters (e.g., a mean, a clinical response probability, or a higher-dimensional object such as a survival function $S(t|\eta(\mathbf{x}))$ corresponding to a terminal region. For instance, the survival function could be of semiparametric form $S_0(t)^{\exp(\eta(\mathbf{x}))}$ as in the proportional hazards model. We outline important components of algorithms used to construct regression trees, including specifying the types of partitions that are permitted; rules to prune the tree back; and methods to choose model or tree size.

3.5.2.2 Splitting or Basis Selection

Trees represent a sequence of splits of the data or predictor space where each split is induced by a rule of the form "$x \in S$" where $S \subset \mathcal{X}$. Typically, splits are dependent on a single covariate, so we may have $S = \{\mathbf{x} : x_j \leq c\}$ for an ordered predictor, or S is a subset $S \subset B = \{v_1, v_2, ..., v_r\}$ of the r values of x_j for categorical variables.

The tree model is grown in a forward stepwise fashion, similar to the stepwise algorithm described in Section 3.4. Starting with the entire data set and predictor space, each variable and potential split point is evaluated. The split point and variable that leads to the "best" split (as described below) is chosen. The data and the predictor space are partitioned into two groups. The same algorithm is then recursively applied to each of the resulting groups. Therefore, at any point on the regression tree, a split at a node h yields two nodes which can also be represented with the pair of basis functions

$$b^+_{h(j)}(\mathbf{x}) = I\{x_{h(j)} \in S_{h(j)}\} \text{ and } b^-_{h(j)}(\mathbf{x}) = I\{x_{h(j)} \notin S_{h(j)}\}.$$

Each step in the growing process geometrically replaces a current node h with a left and right daughter nodes $l(h)$ and $r(h)$ or equivalently a current basis function $B_h(\mathbf{x})$ for node h with the basis functions

$$B_{l(h)}(\mathbf{x}) = B_h(\mathbf{x}) b^+_{h(j)}(\mathbf{x}) \text{ and } B_{r(h)}(\mathbf{x}) = B_h(\mathbf{x}) b^-_{h(j)}(\mathbf{x}).$$

Most tree algorithms use error, likelihood, or partial likelihood (or score tests such as the logrank test) to select split points (or knots). The improvement for a split at node h into left and right daughter nodes can be represented by

$$G(h) = D(h) - [D(l(h)) + D(r(h))],$$

where $D(h)$ is the residual error at a node. For uncensored continuous response problems, $D(h)$ is typically the mean residual sum of squares or mean absolute error or for binary data it is typically binomial deviance. For survival data, it would be reasonable to use the deviance corresponding to the assumed survival model. For instance, the exponential model deviance for node h is

$$D(h) = \sum 2 \left[\delta_i \log \left(\frac{\delta_i}{\widehat{\lambda}_h t_i} \right) - (\delta_i - \widehat{\lambda}_h t_i) \right],$$

where $\delta_i = 1$ if the ith observation was a failure, and $\delta_i = 0$ if the observation was censored, and $\widehat{\lambda}_h$ is the maximum likelihood estimate of the hazard rate in node h (Davis and Anderson, 1989). Alternatively $G(h)$ can be an appropriate score test statistic, for example the logrank test statistic.

Typically a large tree is grown to avoid missing structure and then pruned back using a method described below.

3.5.3 Backwards Selection (Pruning)

Many stepwise regression methods utilize variations of backwards selection to select more simple models (see Section 3.4). The local nature of the tree-based methods leads to a fast backwards method, called cost complexity pruning in the CART algorithm, for evaluating all possible submodels. The cost-complexity objective function is defined as a penalized measure of fit

$$D_\alpha(T) = \sum_{h \in \widetilde{T}} D(t) + \alpha M(\widetilde{T}),$$

where α is a nonnegative complexity parameter, $D(h)$ is the estimated cost or impurity of a node, and $M(\widetilde{T})$ is the number of terminal nodes or constant regression regions R_h. Therefore, the cost-complexity measure controls the trade-off between the size or complexity of the tree, and how well the tree fits the data. Then, for any value of α the goal is to find $T(\alpha)$: the tree that minimizes $D_\alpha(T)$ among all pruned subtrees of T. The algorithm finds the sequence of optimally pruned subtrees by repeatedly deleting branches of the tree for which the average reduction in residual error per split in the branch is small. The process yields a nested sequence of optimal subtrees $T(\alpha) = T(\alpha_l) = T_l$ for $\alpha_l \le \alpha < \alpha_{l+1}$. The removal of a branch can again be viewed in regression context as replacing each of the basis functions corresponding to the pruned branch with the sum of the basis functions

$$B_l(\mathbf{x}) = \sum_{h \in Q_l} B_h(\mathbf{x})$$

where Q_l represent the nodes in a branch rooted at node l. The final tree size is selected by resampling (often K-fold cross-validation is used), although some difficulties arise for semiparametric survival regression models.

3.5.4 Example Revisited

Using the example data set and variables described earlier, we constructed a regression tree to characterize the probability of death or progression within 2 years of registration. Figure 3.2 show a large tree constructed on the available predictors. Below each terminal node is an estimate of the probability of progression or death. Since the tree likely over-fits the data, a pruned tree is selected using cost-complexity pruning and ten fold cross-validation of binomial deviance. The resulting tree model is presented in Figure 3.3; it includes just two splits on variables serum β_2

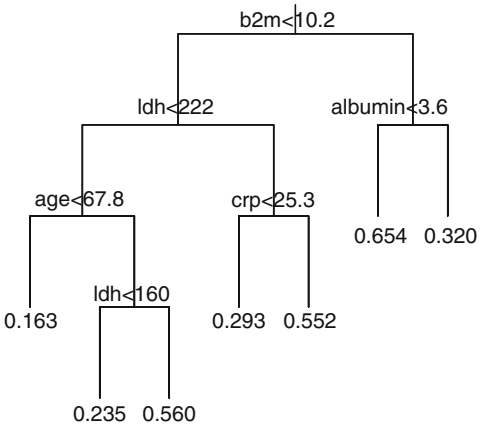


Fig. 3.2 An unpruned regression tree constructed to characterize 2-year progression probability for the multiple myeloma data.

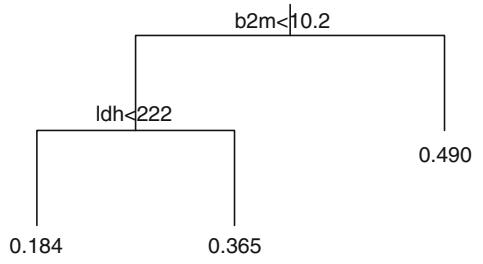


Fig. 3.3 A pruned regression tree constructed to characterize 2-year progression probability for the multiple myeloma data.

microglobulin and lactate dehydrogenase and identifies three outcome groups. Subjects with serum β_2 microglobulin ≥ 10.2 have the worst outcome with 49% having either progressed or died within 2 years.

3.5.5 Issues and Connections

An often cited limitation of regression trees is that they are piecewise constant functions when typically the underlying conditional distribution function of the outcome is a smooth function of the predictors. If interest is in studying groups of patients, this is not really a problem, other than the difficulty in specifying a specific fraction of patients to be indicated by the prognostic rule. However, for prediction applications the nonsmoothness does lead to reduced performance. Ensembles of trees, through boosting, bagging, and Random Forests (Freund and Schapire, 1996; Breiman, 1996; Friedman et al., 2000) have been used to circumvent this discreteness at the cost of losing the simple decision rules. Alternatively, spline methods such as HARE or MARS described in Section 3.6 can lead to substantially improved predictions.

In part because of their nonsmoothness and the stepwise selection method, trees are subject to considerable variability. An important parameter to control variability is the minimum number of observations in a node (or uncensored observations in the case of censored survival data). This issue connects to the importance of avoiding placing knots in regression splines too close to the edge of the covariate distribution. Again, ensembles of trees have been used to reduce variability (sometimes dramatically) but again at the loss of the simple decision rule properties. Retaining decision rule but somewhat smoother methods have been proposed, such as rule induction via the PRIM method (Friedman and Fisher, 1999).

3.6 Spline Models

3.6.1 One Dimensional

Spline models are primarily used for the approximation of smooth univariate and multivariate functions. In univariate problems, splines are piecewise polynomial functions, that satisfy some regularity conditions. In particular, let $t_0 < t_1 < \cdots < t_K$ be a set of K knots. A function $f(x)$ is a cubic spline if in each of the intervals (t_{k-1}, t_k), $k = 1, \ldots, K$, the function $f(x)$ is a cubic polynomial, and it is twice differentiable everywhere. Different spline models may have boundary restrictions for $f(x)$ on the intervals $(-\infty, t_0]$ and $[t_K, \infty)$, but when there are no boundary conditions it is easy to see that these cubic spline functions form a linear space, with basis

$$1, x, x^2, x^3, (x - t_k)_+^3, \quad k = 0, \ldots, K, \tag{3.5}$$

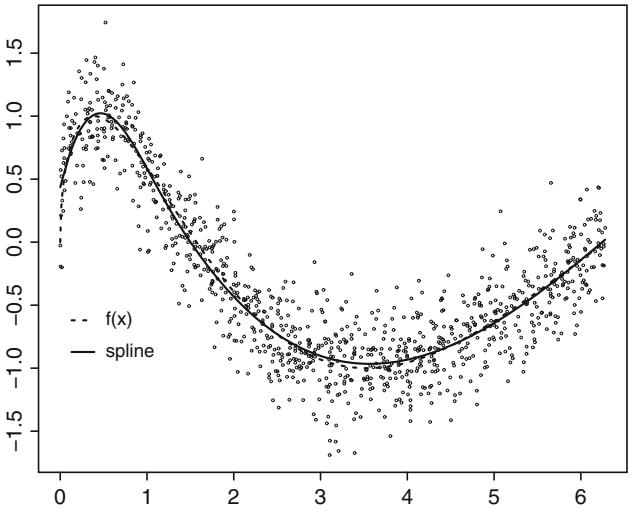


Fig. 3.4 Spline fit. The data were 1,000 i.i.d. samples generated from $Y = f(x) + \varepsilon$, $f(x) = \sin(\sqrt{2\pi x})$ where $x \sim \text{Unif}(0, 2\pi)$ and $\varepsilon \sim N(0, 0.25^2)$. The spline approximation only has a single knot at 1.13.

where $x_+ = x$ if $x > 0$ and 0 otherwise. Cubic spline functions can approximate functions very well, often with a small number of knots (Figure 3.4).

Similarly to cubic splines, a function $f(x)$ is a linear spline if it is continuous and linear on each of the intervals (t_{k-1}, t_k). A basis for linear spline functions is

$$1, x, (x - t_k)_+, \quad k = 0, \ldots, K. \tag{3.6}$$

Regression tree functions in one dimension can be seen as piecewise constant splines. Linear and piecewise constant splines are not as good as cubic splines in approximating smooth curves, but they are often easier to deal with algorithmically. As splines form a linear space, the spline model can be written in the form (3.4). Note that in most situations (3.5) and (3.6) are not the bases used for computations, as they are numerically very instable; instead usually a B-spline basis is used (de Boor, 1978).

Spline models naturally arise as solutions for some penalized regression problems. For example, based on regression data (Y_i, x_i), $i = 1, \ldots, n$, the solution of the minimization problem

$$\arg\min_{f(x)} \sum_i (Y_i - f(x_i))^2 + \lambda \int \left(\frac{\mathrm{d}^2 f(x)}{\mathrm{d}x^2} \right)^2 \mathrm{d}x \tag{3.7}$$

is a (natural) cubic spline with knots at every unique data point x_i (Green and Silverman, 1994). In practice, having a model with so many knots causes problems in many nonlinear and high-dimensional problems. Instead, several other approaches use spline methods with fewer knots.

- Instead of using n knots, express $f(x)$ as a spline function with a fairly large number of knots, that is still much smaller than n, and then use a penalized optimization like (3.7) (O'Sullivan, 1988; Eilers and Marx, 1996). This approach works fairly well in more complicated one-dimensional problems, as well as for generalized additive models, in particular with automatic rules to select smoothing parameters.
- Use a much smaller number of pre-specified knots, and carry out estimation without penalty terms. The advantage is that the resulting problem is fully parametric, and that inference is thus well established. Estimation problems are often small (and easy). See Quantin et al. (1999) for an application in oncology. The disadvantage is that selection of the location of the knots can be arbitrary, and generalizations to nonadditive models are not immediate.
- A third alternative is to use a stepwise algorithm like the one described in Section 3.4 using knots and basis functions from (3.5). This approach was first used in univariate regression by Smith (1982) and is behind algorithms like MARS (Friedman, 1991) for linear regression, HARE (Kooperberg et al., 1995) for survival data, and Polyclass (Kooperberg et al., 1997) for logistic regression and classification. We will discuss those in more detail for multivariate models below. See Polesel et al. (2005) for an application in oncology.

3.6.2 Higher-Dimensional Models

The common approach to using regression splines in higher dimensions is to use basis functions that are tensor products of basis functions in one dimension. For example, if $B_1(\mathbf{x}) = g_1(x_k)$ and $B_2(\mathbf{x}) = g_2(x_l)$ are two basis functions that depend on a single predictor, then $B_3(\mathbf{x}) = g_1(x_k)g_2(x_l)$ is a tensor product basis function that depends on two predictors. For high-dimensional problems, it is common to consider only a few selected lower-order interactions. This has a variety of advantages: (1) lower-dimensional components are typically easier to interpret, interactions in models that do not contain the corresponding main effects are particularly difficult to interpret; (2) using all (higher order) tensor products of lower-order basis functions would yield an extremely large number of basis functions and may cause numerical instability, and (3) from a theoretical perspective, it has been established that spline functions have faster convergence rates if the largest order of interactions in models is small (Stone, 1994). The exact restrictions on when tensor product basis functions are allowed in spline models differs from one methodology to the other: for example, MARS (Friedman, 1991) has fewer restrictions than HARE (Kooperberg et al., 1995), Polyclass, and Polymars (Kooperberg et al., 1997). Here we will describe the Polyclass algorithm for logistic regression as an example.

Assume that we have an i.i.d. sample of size n with a binary response variable Y and a p-dimensional vector of predictors $\mathbf{x} = (x_1, \ldots x_p)'$. Polyclass uses linear splines, and uses interactions involving at most two predictors (although the generalization to higher-dimensional interactions is immediate). An allowable

linear space $\mathcal{B}(\S)$ can have basis functions 1, x_i, $(x_i - t_{k_i})_+$, $x_i x_j$, $(x_i - t_{k_i})_+ x_j$, and $(x_i - t_{k_i})_+ (x_j - t_{k_j})_+$, with $i \neq j \in \{1, \ldots, p\}$, where the t_{k_i} are knots in the range of x_i, with the additional conditions that

- $B(\mathbf{x}) = x_i x_j$ can only be in $\mathcal{B}(\S)$ if $B(\mathbf{x}) = x_i$ and $B(\mathbf{x}) = x_j$ are in $\mathcal{B}(\S)$;
- $B(\mathbf{x}) = (x_i - t_{k_i})_+$ can only be in $\mathcal{B}(\S)$ if $B(\mathbf{x}) = x_i$ is in $\mathcal{B}(\S)$;
- $B(\mathbf{x}) = (x_i - t_{k_i})_+ x_j$ can only be in $\mathcal{B}(\S)$ if $B(\mathbf{x}) = x_i x_j$ and $B(\mathbf{x}) = (x_i - t_{k_i})_+$ are in $\mathcal{B}(\S)$; and
- $B(\mathbf{x}) = (x_i - t_{k_i})_+ (x_j - t_{k_j})_+$ can only be in $\mathcal{B}(\S)$ if $B(\mathbf{x}) = x_i (x_j - t_{k_j})_+$ and $B(\mathbf{x}) = (x_i - t_{k_i})_+ x_j$ are in $\mathcal{B}(\S)$.

The algorithm then proceeds with the stepwise algorithm in Section 3.4. The final model is selected as the one that minimizes

$$\mathrm{AIC}_\alpha = -\widehat{\ell}(\mathcal{B}(\S); Y_i, \mathbf{x}_i, i = 1, \ldots, n) + \alpha p, \tag{3.8}$$

where $\widehat{\ell}(\mathcal{B}(\S); Y_i, \mathbf{x}_i, i = 1, \ldots, n)$ is the fitted log-likelihood for one of the models (of dimension p) that was considered, and α is a penalty parameter, or that maximizes the cross-validated likelihood.

3.6.3 Example Revisited

We applied the Polyclass methodology to the multiple myeloma data. The polyclass model with the default penalty parameter of $\alpha = \log n \approx 6.66$ (3.8) only involved the two predictors log(b2m) and anyca in a linear fashion:

$$\mathrm{logit}(P(\text{progression})) = 2.19 - 0.99 \log(\text{b2m}) - 0.50 \text{anyca}.$$

The model with $\alpha = 4$, while likely overfitting the data somewhat, is more interesting, as it also involves age, gender, log(ldh), log(creatinine), a knot in age, log(ldh), and log(b2m), and an interaction between age and gender. In Figure 3.5 we show a contour plot for the fitted 2-year progression probabilities as a function of creatinine and age, separately for men and women, when the other predictors are held at their median values. The figure indicates that while older ages lead to quite similar progression proportions, younger females tend to have higher risk than younger males.

3.7 Logic Regression

Logic regression is a generalized regression methodology that is particularly suited for situations in which (most) predictors are binary. Clearly this is the case when predictors are single nucleotide polymorphisms (SNPs), as is the case for the multiple myeloma data and many other oncological problems. The logic regression model is

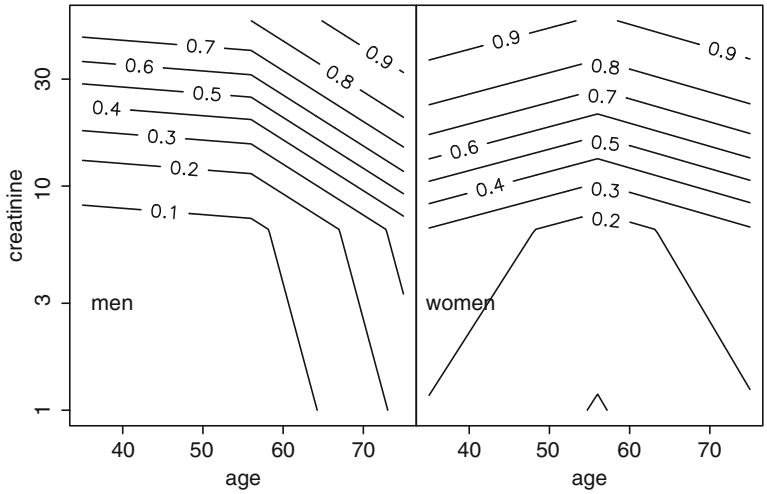


Fig. 3.5 Fitted 2-year progression probabilities for a Polyclass model selected with penalty $\alpha = 4$ as a function of creatinine and age, separately for men and women, when the other predictors are held at their median values.

$$\eta(\mathbf{x}) = \beta_0 + \sum_{i=1}^{m} \beta_i \mathrm{L}_i(\mathbf{x}). \tag{3.9}$$

Each of the L_i is a Boolean combination of binary predictors x_j, $j = 1, \ldots, J$ such as

$$\mathrm{L}_i = [(x_7 \text{ and } x_{13}^c) \text{ or } x_5],$$

where "1" equals "true," "0" equals "false," and c refers to the complement. Additional predictors Z or components to correct for population stratification can be included additively in model (3.9).

Logic regression is an adaptive algorithm which selects those logic terms L_i that minimize the residual sum of squares or maximize the log-likelihood corresponding to the model (3.9). Typically in logic regression the number of logic terms m is small (between 1 and 3), and the logic terms can be interpreted as "risk factors." The optimization of the logic regression model is carried out using a greedy stepwise algorithm or a stochastic simulated annealing algorithm.

For this simulated annealing algorithm it turns out to be very convenient to represent a logic expression $\mathrm{L}_i(\mathbf{x})$ in a logic tree form (Figure 3.6). During the simulated annealing algorithm, at each step one of the logic trees is replaced by another logic tree using one of the operations displayed in Figure 3.7. Based on the new tree the likelihood of $\eta(\mathbf{x})$ is evaluated. If the new model is an improvement over the existing model the new model is retained; if the old model was better the new model is retained with a probability that depends on the difference between the old and new log-likelihood and the stage of the algorithm: early on almost all new models are accepted, while toward the end of the algorithm only improved models are accepted.

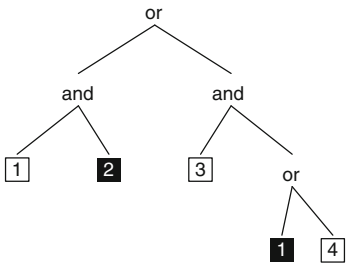


Fig. 3.6 A logic tree representation of the Boolean expression $(x_1 \wedge x_2^c) \vee (x_3 \wedge (x_1^c \vee x_4))$. Logic trees are evaluated from the bottom up; white numbers on a black background denote the complement.

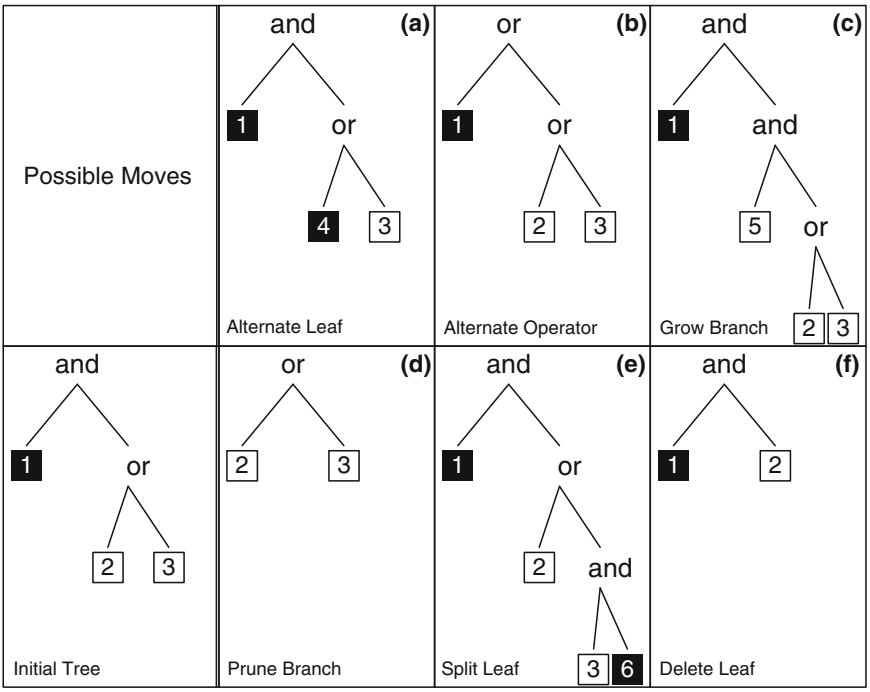


Fig. 3.7 Changes in logic regression trees considered during the simulated annealing algorithm.

3.7.1 Example Revisited

We applied logic regression to the 348 SNPs of the multiple myeloma data, using again 2-year progression as the outcome. Each of the 348 SNPs was recoded as two binary predictors corresponding to a dominant and a recessive effect. In Figure 3.8

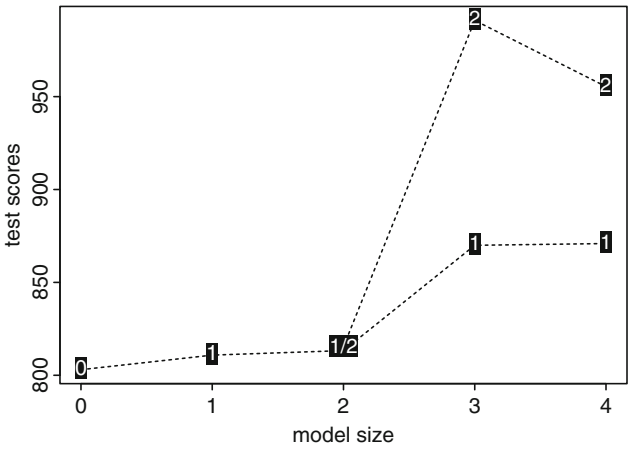


Fig. 3.8 Cross-validation (test set) deviance for the logic regression analysis of the multiple myeloma data. The white numbers in the black squares refer to the number of logic terms in the logic regression model, the model size refers to the total number of leaves in these models combines.

we show the test set deviance from tenfold cross-validation of the logic regression analysis of this data. We note from this figure that based purely on deviance, none of the models is better than the null-model. The model with two SNPs, however, has a deviance that is not much worse than the null-model, and may thus be of interest for further investigation. This model includes a logic regression term

$$\text{rs4148737D} \vee \text{rs1143627R}^c,$$

(rs1922242D was identical to rs4148737D) on this data. We will see the same SNPs appear in the analysis in the next section.

3.8 High-Dimensional Data

With the development of new genomic technologies, very high-dimensional data sets are now generated for oncological data. Data sets using gene expression data may have data on tens of thousands of genes (e.g., Rosenwald et al., 2002), data sets for whole genome association studies may have data on hundreds of thousands of SNPs (e.g., Easton et al., 2007; Yeager et al., 2007). The traditional statistical paradigm, where the number of cases n is much larger than the number of predictors p no longer holds in this situation. Typical statistical methods for this type of data involve substantial amounts of model selection, as well as shrinkage of the parameter estimates.

3.8.1 Variable Selection and Shrinkage

In moderate to high-dimensional predictor settings it is desirable to have parsimonious or sparse representations of prediction models. In the previous sections we have discussed stepwise basis function selection strategies. Alternatively, one can investigate smoother model selection methods.

3.8.2 LASSO and LARS

Consider the linear regression setting, where there are n independent observations $(y_i, x_{i1}, \ldots, x_{ik})$ of the response and k predictor variables. A technique proposed by Tibshirani (1996) introduces an L^1-penalty on the regression coefficients which leads to both shrinkage and variable selection called least absolute shrinkage and selection operator (LASSO). This is in contrast to ridge regression (Hoerl and Kennard, 1970) which minimizes the residual error subject to an L^2-penalty which does not lead to variable selection. The LASSO estimate $\widetilde{\beta} = (\tilde{\beta}_1, \ldots, \widetilde{\beta}_m)'$ is defined as the minimizer of

$$g(\beta) = \sum_{i=1}^{n} (y_i - \sum_k \beta_k x_{ik})^2 + \lambda_1 \sum |\beta_k|^1,$$

where λ_1 is a nonnegative penalty parameter. Often the response and predictors are standardized so that $\sum_i y_i = 0$ and $\sum_i x_{ik} = 0$ and $\sum_i x_{ik}^2 = 1$. This estimator has the attractive property that as λ_1 increases minimizing $g(\beta)$ with respect to β leads to some of the β_k set to zero and hence variable selection. For fixed λ_1, for optimization quadratic programming techniques or alternatives more efficient methods by Osborne et al. (2004) can be used. A related and highly efficient algorithm, the least angle regression algorithm (LARS, Efron et al., 2004), leads to efficient estimation and links forward stage-wise methods and LASSO. LASSO and LARS are discussed in more detail in Chapter 2.

LARS gives answers that are often close to LASSO; they are identical if the predictors are orthogonal. However, the estimation algorithm aligns closely with the forward stepwise model building strategies described in earlier sections. An outline of the algorithm is given below:

1. Start with $r = y$, $\widehat{\beta}_j = 0$, $j = 1, \ldots, p$. Assume that the x_j are standardized.
2. Find the predictor x_k that is most correlated with r.
3. Increase $\widehat{\beta}_k$ in the direction of $\text{sign}(\text{cor}(r, x_k))$ until another predictor x_j has equal correlation to r as it does with x_k. Put j in set of active predictors, S.
4. Move $(\widehat{\beta}_k : k \in S)$ in the joint least squares direction for $(x_k : k \in S)$ until yet another predictor has equal correlation with the current residual.
5. Repeat Step 4 until $\text{cor}(r, x_k) = 0$ for all k.

Note that the model can include at most $\min(p,n)$ variables. One strategy to alleviate this potential problem is the "elastic net" proposed by Zou and Hastie (2005). The elastic net can be expressed as an optimization problem with the objective function with both squared and absolute penalty terms

$$g(\beta) = \sum_{i=1}^{N}(y_i - \sum_k \beta_k x_{ik})^2 + \lambda_1 \sum |\beta_k|^1 + \lambda_2 \sum |\beta_k|^2.$$

Their simulations show that the elastic net method leads to grouping of variables where strongly correlated variables are either in or removed from the model as the penalty parameters λ_1 and λ_2 are increased.

Note, that in this section we have described these methods in terms of the original predictors x_k; we could generalize to sets of regression spline or regression tree basis functions, $B_j(\mathbf{x}), j = 1,\ldots,p$ as described in the previous section.

3.8.3 Dedicated Methods

While the methods described above directly lead to dimension reduction, there are a large number of other methods which can be viewed as two-stage procedures that at the first stage reduce the set of original variables x_i to a small number of combinations z_i and then at the second phase uses those combinations in further regression modeling. Many of the techniques can be viewed as generalizations or parallels to either principal components regression, which uses only the joint distribution of the x_i at the first stage, or partial least squares which constructs linear combinations of the predictors but also guides the selection by also using the outcome Y.

For instance, many gene expression modeling applications in oncology have used clustering of genes to derive predictor variables for association modeling. Jointly using outcome and expression was used by Hastie et al. (2001) and Dettling and Bühlmann (2002) and others. An important consideration when using both the joint distribution of outcome and predictors at the first stage is that appropriate assessment of prediction error and model fit is incorporated (for instance by cross-validation) and included in the modeling building.

3.8.4 Example Revisited

We applied a generalization of the LARS regression method appropriate for binary data (Park and Hastie, 2006) to the 2-year progression-free survival outcome, and the multiple myeloma SNP data. Each of the SNPs was coded in dominant and recessive form. In the Figure 3.9, we show the first few steps of the coefficient path. Three SNPs appear to enter the model early, "rs1143627R," "rs2756109D," and "rs703842R." Note that the SNPs that were selected by logic regression entered

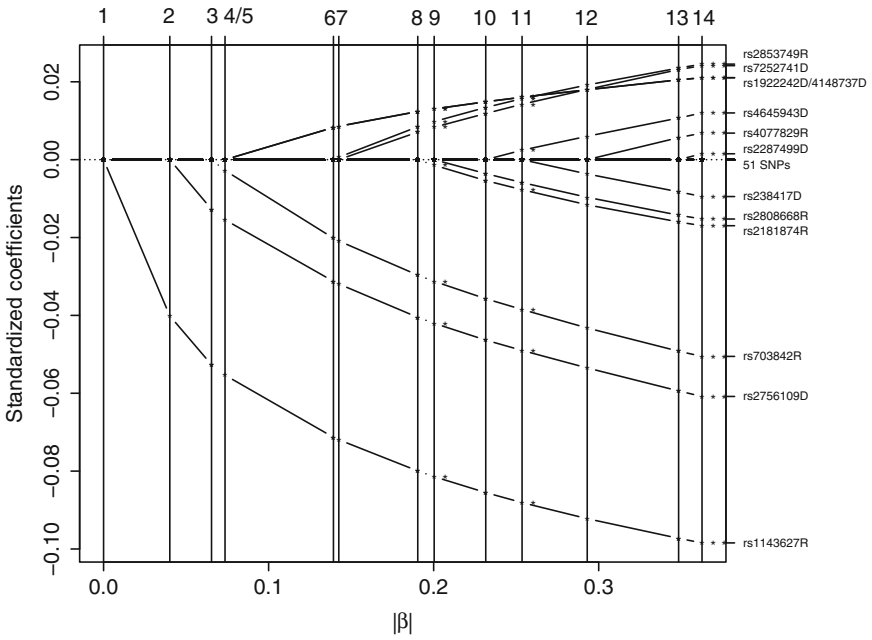


Fig. 3.9 Coefficient path for myeloma SNP data.

the model as the first, fourth, and fifth SNP. Cross-validating the model building process leads to the conclusion that the cross-validated estimates of deviance are relatively flat with respect to model complexity and then start to increase for models with larger numbers of predictors. Therefore, there is not strong evidence that the combination of SNPs are significantly associated with disease progression.

Often there is interest in assessing if genomic information adds to prediction beyond traditional laboratory measures. This can be easily incorporated by adjusting for known myeloma clinical variables then fitting SNP data using the Park and Hastie algorithm. This was done for the above example and while not unexpected given the earlier analysis, it suggested no additional impact with SNP data on prediction over the laboratory variables previously described.

3.9 Survival Data

An important goal in survival regression analysis is to determine how the distribution of survival times depends on the predictors. A complication in analyzing survival data in the context of oncology trials is that typically not all patients have died (or progressed) by the time the analysis for the study is completed. Those patients alive at the time of analysis are called censored.

We denote the true survival time as a positive random variable T, whose distribution may depend on a set of predictors $\mathbf{x} = (x_1, \ldots, x_k)'$. Often it can be assumed that the censoring mechanism is independent which facilitates likelihood construction and inference. Let the observed data be denoted by $(T_i, \delta_i, \mathbf{x}_i),\ i = 1, \ldots, n$.

While one can express the conditional survival distribution using an accelerated failure time specification which links the $\log(T)$ to a linear model of the predictors, $\log(T) = a + b'X + e$, hazard function modeling is most often used. The conditional hazard function is defined as

$$\lambda(t|\mathbf{x}) = \lim_{\Delta \to 0} \frac{P(t \le T < t + \Delta | t \le T; \mathbf{x})}{\Delta}.$$

Here, we limit discussion to predictors which are real values measured at baseline; in some survival settings they may represent time-dependent functions, $x_1(t), \ldots, x_k(t)$, as well. For instance, they may be measures of health status of the patient evolving over time. The (conditional) hazard function can be interpreted as the probability that someone dies in the next time interval of infinitesimal length Δ, given that he is alive at time t. It is convenient to specify models on the logarithm scale so we denote the logarithm of the hazard function as

$$\alpha(t|\mathbf{x}) = \log \lambda(t|\mathbf{x}).$$

If one assumes an additive model on the log scale,

$$\alpha(t|\mathbf{x}) = f(t) + \eta(\mathbf{x})$$

implies a proportional hazards assumption which is a focus of the model of Cox (1972), which also assumes the baseline hazard function to be an unspecified nonparametric function. Estimation in that case utilizes the partial likelihood. Note that $\eta(\mathbf{x})$ can represent a simple linear model or more flexible models depending on a regression spline basis described in earlier sections. For instance, let $B_1(\mathbf{x}), \ldots, B_p(\mathbf{x})$ be a basis for $\mathcal{B}(\mathcal{X})$. Then we can write

$$\eta(\mathbf{x}) = \sum_{i=1}^{p} \beta_i B_i(\mathbf{x}). \tag{3.10}$$

Within the proportional hazards class, tree-based or logic regression models can be used to characterize the basis section used in the above expression.

However, regression models can be more general. For instance, the HARE model (Kooperberg et al., 1995), can link both time and the predictors using a model specified as

$$\alpha(t|\mathbf{x}) = \sum_{i} \beta_i B_i(t|\mathbf{x}). \tag{3.11}$$

The basis functions $B_i(t|\mathbf{x})$ in HARE can depend solely on time or a predictor or both on time and a predictor which allows specification on nonproportional hazards

models. The basis functions are selected with an algorithm similar to the Polyclass algorithm in Section 3.6.2.

Modeling the full survival distribution is slightly more general than modeling within the proportional hazards framework. But there are also disadvantages: coefficients in model (3.10) are interpretable as log-relative risk estimates, while the nonproportionality in (3.11) removes this interpretation. Computationally the partial likelihood computations for (3.10) are much easier than the full likelihood computations for (3.11), as these later require integrating the conditional survival function for every unique set of covariates $\mathbf{x}$ which, except for piecewise linear splines, becomes very demanding.

We end this section by noting that a simple transformation of the survival times may facilitate modeling. Suppose that T is a continuous random variable having distribution function F. Then $U = F(T)$ has a standard uniform distribution and $\log(U) = \Lambda(T)$, where Λ represents the cumulative hazard function, has a standard exponential distribution and thus a constant hazard function. In the context of hazard function modeling with HARE, the regression model applied to survival times transformed by the marginal cumulative hazard function tends to require fewer knots applied to the time variable allowing more focus on the impact of predictors on the (transformed) outcome. The overall transformation applied to the data can be semiparametric, for instance using a regression spline model for the hazard function (the HEFT method of Kooperberg et al., 1995) or non-parametric using the empirical cumulative hazard function estimate. This transformation can facilitate the use of other flexible regression procedures utilizing exponential model likelihood, which typically allows for much faster computation than partial likelihood. For instance, after transforming the survival times by the cumulative hazard transformation, the survival times may be sufficiently well approximated by an exponential distribution, so that a regression tree program based on the exponential likelihood may perform well.

3.9.1 Example Revisited

The multiple myeloma data set included both overall survival and progression-free survival endpoint data. In this analysis, we consider all 778 subjects with complete covariate data. The HARE analysis of the time to progression is very similar to the Polyclass analysis presented in Section 3.6.3. The analysis of the survival time turned out more interesting, as it depended on serum β_2 microglobulin, anyca, log(ldh), and age, and included a nonproportionality component for log(ldh). In Figure 3.10, we show the fitted hazard function for a person of age 56, with a log(b2m) of 1, no anyca, and log(ldh) values of 4, 5, and 6, which roughly correspond to the 25th, 50th, and 75th percentile of the log(ldh).

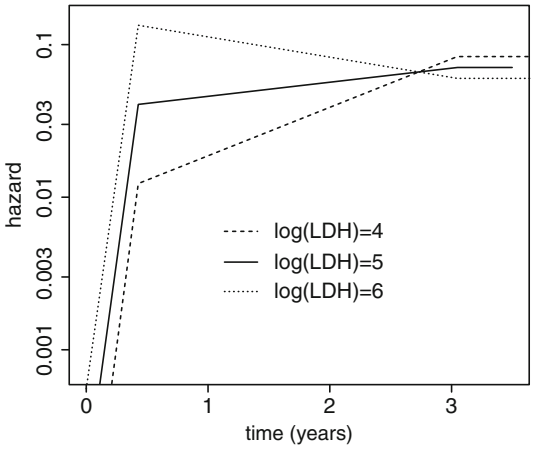


Fig. 3.10 Fitted hazard functions for the HARE analysis of the multiple myeloma data.

3.10 Discussion

Many choices exist for flexible regression modeling of patient data from oncology studies. Selection of appropriate methods, of course, depends on the goals in the particular analysis. For instance, it could be best to characterize the risk of progression as a smooth function of a single important prognostic variable or to develop a more general risk models using multiple predictors and variable selection. Adaptive regression spline methods such as HARE are well suited to such problems. Alternatively, one may want to characterize groups of patients or subjects, or identify interactions of binary predictor variables. Tree-based methods or logic regression are two tools useful for such problems.

A common aspect of cancer data is that the strength of associations between predictors and patient outcome is quite weak as demonstrated with the myeloma data. While sometimes it is useful to slightly overfit the data to suggest models that may be worth investigating further, in general we should prevent selecting regression models that are not supported by the data. Therefore, using methods to obtain honest prediction error to help avoid over-fitting is critical.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723.

Barlogie, B., Tricot, G., Rasmussen, E., Anaissie, E., van Rhee, F., Zangari, M., Fassas, A., Hollmig, K., Pineda-Roman, M., Shaughnessy, J., Epstein, J., and Crowley, J. (2006). Total therapy 2 without thalidomide in comparison with total therapy 1: role of intensified induction and posttransplantation consolidation therapies. *Blood*, 107:2633–2638.

Board on health sciences policy. (2002). *Cancer and the Environment: Gene–Environment Interaction.* Institute of Medicine, National Academy Press, Washington D. C.

de Boor, C. (1978). *A Practical Guide to Splines*. Springer-Verlag, New York.
Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.
Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Pacific Grove, California.
Ciampi, A., Thiffault, J., Nakache, J. P., and Asselain, B. (1986). Stratification by stepwise regression, correspondence analysis and recursive partition. *Computational Statistics and Data Analysis*, 4:185–204.
Cox, D. R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society B*, 34:187–220.
Dai, J. Y., LeBlanc, M., and Kooperberg, C. (2008). Semiparametric estimation exploiting covariate independence in two-phase randomized trials. *Biometrics*, May 12. [Epub ahead of print].
Davis, R. B. and Anderson, J. R. (1989). Exponential survival trees. *Statistics in Medicine*, 8:947–961.
Dettling, M. and Bühlmann, P. (2002). Supervised clustering of genes. *Genome Biology*, 3:69.1–69.15.
Easton, D. F., Pooley, K. A., Dunning, A. M., Pharoah, P. D. P., Thompson, D., Ballinger, D. G., Struewing, J. P., Morrison, J., Field, H., Luben, R., et al. (2007). Genome-wide association study identifies novel breast cancer susceptibility loci. *Nature*, 447:1087–1093.
Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32:407–499.
Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties (with discussion). *Statistical Science*, 11:89–121.
Farag, S., Archer, K. K., Mrózek, K., Ruppert, A. S., Carroll, A. J., Vardiman, J. W., Pettenati, J., Baer, M. R., Qumsiyeh, M. B., Koduru, P. R., et al. (2006). Pretreatment cytogenetics add to other prognostic factors predicting complete remission and long-term outcome in patients 60 years of age or older with acute myeloid leukemia: results from Cancer and Leukemia Group B 8461. *Blood*, Jul. 1; 108(1):63–73.
Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Tirteenth International Conference*, pp. 148–156. Morgan Kauffman, San Francisco.
Friedman, J. H. (1991). Multivariate adaptive regression splines (with discussion). *The Annals of Statistics*, 19:1–141.
Friedman, J. H. and Fisher, N. I. (1999). Bump-hunting for high dimensional data. *Statistics and Computation*, 9:123–143.
Friedman, J. H., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting (with discussion). *The Annals of Statistics*, 38:337–407.
Gimotty, P. A., Elder, D. E., Fraker, D. L., Botbyl, J., Sellers, K., Elenitsas, R., Ming, M. E., Schuchter, L., Spitz, F. R., Czerniecki, B. J., and Guerry, D. (2007). Identification of high-risk patients among those diagnosed with thin cutaneous melanomas. *Journal of Clinical Oncology*, 25:1129–1134.
Gordon, L. and Olshen, R. A. (1985). Tree-structured survival analysis. *Cancer Treatment Reports*, 69:1065–1069.
Green, P. J. and Silverman, B. W. (1994). *Nonparametric Regression and Generalized Linear Models: a Roughness Penalty Approach*. Chapman and Hall, London.
Greipp, P. R., San Miguel, J., Durie, B. G., Crowley, J. J., Barlogie, B., Bladé, J., Boccadoro, M., Child, J. A., Avet-Loiseau, H., Kyle, R. A., et al. (2005). International staging system for multiple myeloma. *Journal of Clinical Oncology*, 23:3412–3420.
Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized Additive Models*. Chapman and Hall, London.
Hastie, T., Tibshirani, R., Botstein, D., and Brown, P. (2001). Supervised harvesting of regression trees. *Genome Biology*, 2:3.1–3.12.
Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67.

Kooperberg, C., Stone, C. J., and Truong, Y. K. (1995). Hazard regression. *Journal of the American Statistical Association*, 90:78–94.

Kooperberg, C., Bose, S., and Stone, C. J. (1997). Polychotomous regression. *Journal of the American Statistical Association*, 92:117–127.

LeBlanc, M. and Crowley, J. (1993). Survival trees by goodness of split. *Journal of the American Statistical Association*, 88:457–467.

LeBlanc, M., Moon, J., and Crowley, J. (2005). Adaptive risk group refinement. *Biometrics*, 61:370–378.

LeBlanc, M., Moon, J., and Kooperberg, C. (2006). Extreme regression. *Biostatistics*, 13:106–122.

Loader, C. (1999). *Local Regression and Likelihood*. Springer-Verlag, New York.

London, W. B., Castleberry, R. P., Matthay, K. K., Look, A. T., Seeger, R. C., Shimada, H., Thorner, P., Broderu, G., Maris, J. M., Reynolds, C. P., and Cohn, S. L. (2005). Evidence for an age cutoff greater than 365 days for neuroblastoma risk group stratification in the Children's Oncology Group. *Journal of Clinical Oncology*, 23:6459–6465.

Osborne, M. R., Presnell, B., and Turlach, B. A. (2004). On the LASSO and its dual. *Journal of Computational and Graphical Statistics*, 9:319–337.

O'Sullivan, F. (1988). Fast computation of fully automated log-density and log-hazard estimators. *SIAM Journal on Scientific and Statistical Computing*, 9:363–379.

Park, M. Y. and Hastie, T. (2006). L_1 regularization path models for generalized linear models. *Journal of the Royal Statistical Society B*, page in press.

Polesel, J., Dal Maso, L., Bagnardi, V., Zucchetto, A., Zambon, A., Levi, F., La Vecchia, C., and Franeschi, S. (2005). Estimating dose-response relationship between ethanol and risk of cancer using regression spline models. *International Journal of Cancer*, 114:836–841.

Quantin, C., Abrahamowicz, M., Moreau, T., Bartlett, G., MacKenzie, T., Tazi, M. A., Lalonde, L., and Faivre, J. (1999). Variation over time of the effects of prognostic factors in a population-based study of colon cancer: Comparison of statistical models. *American Journal of Epidemiology*, 150:1188–1200.

Quinlan, J. R. (1993). *C4.5 Programs for Machine Learning*. Morgan Kaufman, San Francisco, CA.

Rao, C. R. (1973). *Linear Statistical Inference and Its Applications*. John Wiley, New York.

Rosenwald, A., Wright, G., Chan, W., Connors, J., Campo, D., Fisher, R., Gascoyne, R., Muller-Hermelink, H., Smeland, E., Staudt, L., et al. (2002). Molecular diagnosis and clinical outcome prediction in diffuse large B-cell lymphoma. *New England Journal of Medicine*, 346:1937–1947.

Ruczinski, I., Kooperberg, C., and LeBlanc, M. (2003). Logic regression. *Journal of Computational and Graphical Statistics*, 12:475–511.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6:461–464.

Segal, M. R. (1988). Regression trees for censored data. *Biometrics*, 44:35–47.

Singer, E. (2005). Personalized medicine prompts push to redesign clinical trials. *Nature*, 452:462.

Smith, P. L. (1982). Curve fitting and modeling with splines using statistical variable selection methods. Technical report, NASA, Langley Research Center, Hampla, Virginia.

Stone, C. J. (1994). The use of polynomial splines and their tensor products in multivariate function estimation (with discussion). *The Annals of Statistics*, 22:118–184.

Stone, C. J., Hansen, M. H., Kooperberg, C., and Truong, Y. K. (1997). Polynomial splines and their tensor products in extended linear modeling (with discussion). *The Annals of Statistics*, 25:1371–1470.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288.

Yeager, M., Orr, N., Hayes, R. B., Jacobs, K. B., Kraft, P., Wacholder, S., Minichiello, M. J., Fearnhead, P., Yu, K., Chatterjee, N., et al. (2007). Genome-wide association study of prostate cancer identifies a second risk locus at 8q24. *Nature Genetics*, 39:645–649.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67:301–320.

Chapter 4
Risk Estimation

Ronghui Xu and Anthony Gamst

In this chapter we discuss the concept and various aspects of loss and risk, and why they are important and interesting. We start with the perhaps well-known fact that naive estimates of risk tend to be biased, and that improvements are possible. There are a variety of loss functions one might use, and it is important to understand the differences and the fact that they may well imply different optimal estimates, in finite samples as well as asymptotically. On the other hand, risk estimates for a large class of loss functions, the q-class, are all based on a similar construction. For any loss function in the q-class, the bias correction for the risk estimate is a covariance term. Depending on the loss, this covariance estimate may be computed in a simple way, such as C_p, *AIC*, or Stein's estimator. These also have a close connection with the concept of degrees of freedom. Regardless of the loss function (and fitting procedure), other techniques such as parametric bootstrap or cross-validation can also be used to estimate the risk, some to better effect than others. We summarize empirical studies of risk estimation in the literature, and show some applications, both theoretical, such as adaptive model selection and Stein estimation, and practical, such as gene ranking.

4.1 Risk

Data analysis is about fitting models to data, usually, with the goal of summarization, prediction, or inference. In the analysis of oncology data, for example, we often analyze databases with the goal of using demographic, clinical, and biological markers to build prognostic models. Another typical analysis uses data generated from a high-throughput technology to classify tumors or subpopulations of patients. In all

R. Xu
Division of Biostatistics and Bioinformatics, Department of Family and Preventive Medicine and Department of Mathematics, University of California, San Diego. 9500 Gilman Drive, MC 0112, La Jolla, CA 92093-0112, USA
email: rxu@ucsd.edu

X. Li, R. Xu (eds.), *High-Dimensional Data Analysis in Cancer Research*, Applied Bioinformatics and Biostatistics in Cancer Research, DOI 10.1007/978-0-387-69765-9_4,

such practices, it is important that the model we build has as small a prediction error as possible. This error itself, of course, is unknown to us and has to be estimated from the data at hand. Although it is possible (and encouraged) to collect future data and evaluate the prediction error or validate the model in an independent study, here we concern ourselves with the estimation of prediction error as an integral part of the model building and model selection process.

Prediction error is closely related to *risk*. Let $Y = (y_1, ..., y_n)'$ denote the observed data. Let $\hat{\mu} = \hat{\mu}(Y) = (\hat{\mu}_1, ..., \hat{\mu}_n)'$ be an estimate or prediction for a "future" Y^0, independent of, but from the same distribution as, Y. Depending on the goal, a nonnegative *loss function* $L(Y^0, \hat{\mu})$ can be, for example,

$$\begin{cases} \|Y^0 - \hat{\mu}\|_2^2, \\ \|Y^0 - \hat{\mu}\|_1, \\ -2\{\log f(Y^0; \hat{\mu}) - \log f(Y^0; \mu)\}; \end{cases}$$

these are, respectively, squared error loss, L_1 loss, and deviance loss. A loss function measures the distance (or divergence) between the true data generating mechanism and the fitted model in terms of goals at hand. The *risk function* is then the expected loss:

$$R(Y^0, \hat{\mu}) = E_0\{L(Y^0, \hat{\mu})\}, \tag{4.1}$$

where E_0 is taken with respect to Y^0. It is clear that a good choice of $\hat{\mu}$ should minimize the risk.

4.2 Covariance Penalty

Notice that $E_0\{L(Y^0, \hat{\mu})\}$ is a random quantity, involving the original data Y. To approximate $E_0\{L(Y^0, \hat{\mu})\}$, or, to estimate $EE_0\{L(Y^0, \hat{\mu})\}$ where $E(\cdot)$ is taken with respect to Y, we start with the "apparent" loss $L(Y, \hat{\mu}) = \sum_{i=1}^n L(y_i, \hat{\mu}_i)$. It is known that $L(Y, \hat{\mu})$ generally underestimates the risk (Efron, 1983, 1986). This can be seen from the derivation of the unbiased estimate of the risk below, as well as the empirical studies mentioned later.

The difference between the actual risk and apparent loss, $E_0\{L(Y^0, \hat{\mu})\} - L(Y, \hat{\mu})$, is termed the *optimism* of the apparent loss. The bias of the apparent loss in estimating $EE_0\{L(Y^0, \hat{\mu})\}$ is then the *expected optimism*. To correct for this bias or optimism, various methods have been proposed in the literature and adopted in practice. These include a well-known example in the case of deviance loss: the Akaike (1973) information criterion (AIC).

4.2.1 Continuous Outcomes

Denote $\mu = (\mu_1, ..., \mu_n)' = E(Y) = E_0(Y^0)$. If the quadratic loss is used, then

$$
\begin{aligned}
EE_0\{L(Y^0,\hat{\mu})\} &= \sum_{i=1}^{n} EE_0(y_i^0-\hat{\mu}_i)^2 \\
&= \sum_{i=1}^{n} EE_0\{(y_i^0-\mu_i)^2+2(y_i^0-\mu_i)(\mu_i-\hat{\mu}_i)+(\mu_i-\hat{\mu}_i)^2\} \\
&= \sum_{i=1}^{n} E\{(y_i-\mu_i)^2+(\mu_i-\hat{\mu}_i)^2\} \\
&= \sum_{i=1}^{n} E\{(y_i-\hat{\mu}_i)^2+2(\hat{\mu}_i-\mu_i)(y_i-\mu_i)\} \\
&= E\{L(Y,\hat{\mu})+2\sum_{i=1}^{n}\mathrm{Cov}(y_i,\hat{\mu}_i)\} \qquad (4.2)
\end{aligned}
$$

The last line above gives the *covariance penalty* (Stein, 1981; Efron, 2004). In the linear case this relates directly to the trace of the "hat" matrix; that is, if $\hat{\mu}=HY$ is linear in Y, the covariance term is equal to $\sigma^2\mathrm{trace}(H)$, where $\sigma^2=\mathrm{Var}(y_i)$. Assuming that σ^2 is known, $L(Y,\hat{\mu})+2\sigma^2\mathrm{trace}(H)$ provides an unbiased estimate of the risk, and is known as Mallows' C_p (Mallows, 1973). When σ^2 is unknown, a modified estimate can be obtained. The trace of the "hat" matrix is also used to define the *degrees of freedom* in generalized linear models (Hastie and Tibshirani, 1990), hierarchical, and other richly parameterized models (Hodges and Sargent, 2001). It is clear that if we know the degrees of freedom, which might involve the true but unknown parameters, and know how to estimate the unknown parameters, then we can estimate the corresponding risk.

The covariance $\mathrm{Cov}(y_i;\hat{\mu}_i)$ is generally unknown. In the Gaussian homoscedastic case where $Y\sim N(\mu,\sigma^2 I)$, and assuming a differentiability condition on the mapping $\mathcal{M}: Y\to\hat{\mu}$, Stein (1981) using integration by parts, and the fact that $\partial\phi(z)/\partial z=-z\phi(z)$ for the standard normal density $\phi(\cdot)$ showed that

$$
\mathrm{Cov}(Y_i;\hat{\mu}_i)=\sigma^2 E\left\{\frac{\partial\hat{\mu}_i}{\partial y_i}\right\}. \qquad (4.3)
$$

This was used in Ye (1998) to define general degrees of freedom for the mapping $\mathcal{M}$, by the sum of sensitivity of each $\hat{\mu}_i$ to perturbation in y_i. Note that $\partial\hat{\mu}_i/\partial y_i$ is computable, and this leads to Stein's unbiased risk estimate (SURE)

$$
L(Y,\hat{\mu})+2\sigma^2\sum_{i=1}^{n}\frac{\partial\hat{\mu}_i}{\partial y_i}. \qquad (4.4)
$$

An application of SURE is discussed in Section 4.4.

4.2.2 Binary Outcomes

Many loss functions can be approximated, at least locally, by some version of weighted least-squares, and a covariance penalty similar to the one derived earlier

for squared loss can be applied in many cases (Efron, 2004). Consider, for example, loss functions defined by Bregman divergence (Bregman, 1967). A divergence $D(y,\mu)$ is similar to a metric, in that $D(y,\mu) \geq 0$ and $D(y,\mu) = 0$ if and only if $y = \mu$, but D need not be symmetric, so that $D(y,\mu) \neq D(\mu,y)$, and may not satisfy the triangle inequality. The q-class (Efron, 1986) of Bregman divergences is generated by strictly concave functions q as defined below.

Suppose that y is a random variable taking on values 0 or 1. In logistic regression

$$P(y=1) = \pi = \frac{1}{1+\exp(\beta' x)}, \tag{4.5}$$

where x is a vector of p covariates, and β is the vector of regression coefficients. Given data $Y = (y_1,...,y_n)'$, using the maximum likelihood estimate $\hat{\beta}$ of β, we obtain the estimated probabilities $\hat{\pi}_i = \hat{P}(y_i = 1)$, $i = 1,...,n$. The prediction $\hat{\mu}_i$ of y_i is usually made by $\hat{\mu}_i = 1$ if $\hat{\pi}_i > C$ for some constant C, and 0 otherwise. The default choice of C is $1/2$, for example.

Let $L(y,\hat{\mu})$ be a measure of prediction error, i.e. the loss function. In addition to the examples of loss functions given earlier, for binary outcomes we may also use counting error: $L(y_i,\hat{\mu}_i) = 1$ if $y_i \neq \hat{\mu}_i$, and 0 otherwise. To derive the expected loss as previously done for continuous outcomes, Efron (1986) defined the *q class* of loss functions. Let $q(\mu)$ be a concave function for $\mu \in [0,1]$, with $q(0) = q(1) = 1$. Then define

$$L(y,\hat{\mu}) = q(\hat{\mu}) + \dot{q}(\hat{\mu})(y-\hat{\mu}), \tag{4.6}$$

where $\dot{q}$ is the derivative of q (or the left limit of the derivative where it is discontinuous). For the counting error above, $q(\mu) = \min(\mu, 1-\mu)$. For the squared loss, i.e. $L(y,\hat{\mu}) = (y-\hat{\mu})^2$, we have $q(\mu) = \mu(1-\mu)$. Finally, for the deviance loss function, $L(y,\hat{\mu}) = -2\log\{\hat{\mu}^y(1-\hat{\mu})^{1-y}\}$ and $q(\mu) = -2\{\mu\log(\mu) + (1-\mu)\log(1-\mu)\}$.

Let Y^0 again be a future vector of outcomes, independent but from the same distribution as Y. Let

$$\hat{\zeta}_i = -\dot{q}(\hat{\mu}_i). \tag{4.7}$$

Using (4.6) Efron (1986) showed the expected optimism, i.e. the expected difference between the risk and the apparent loss

$$\begin{aligned} & EE_0\{L(Y^0,\hat{\mu})\} - E\left\{\sum_{i=1}^n L(y_i,\hat{\mu}_i)\right\} \\ &= E\left\{\sum_{i=1}^n \hat{\zeta}_i(y_i - \pi_i)\right\} \\ &= \sum_{i=1}^n \mathrm{Cov}(y_i,\hat{\zeta}_i). \end{aligned} \tag{4.8}$$

For counting error, $\hat{\zeta}_i = \text{sign}(2\hat{\mu}_i - 1)$; for squared error $\hat{\zeta}_i = 2\hat{\mu}_i - 1$; and for the deviance loss $\hat{\zeta}_i = 2\log\{\hat{\mu}_i/(1-\hat{\mu}_i)\}$. So we see that for the squared error, (4.8) is the same as (4.2).

Efron (1986) extended the above result to generalized linear models and exponential families. Briefly, suppose that y_i comes from a one-parameter exponential family

$$f_\mu(y) = \exp\{\lambda y - b(\lambda)\},$$

where $\mu = E(y) = g(\lambda) = \mathrm{d}b(\lambda)/\mathrm{d}\lambda$ is the expectation parameter, λ is the natural or canonical parameter, b is the normalizing function (which ensures that the density integrates to 1), and g is the (inverse) link function. Then, taking

$$q(\mu) = 2\{b[\lambda(\mu)] - y\lambda(\mu)\}$$

makes $L(y, \hat{\mu})$ the deviance loss function (Efron, 1986; Zhang, 2004).

4.2.3 A Connection with AIC

When deviance loss and the maximum likelihood estimate are used in curved exponential families, the covariance penalty derivation gives rise to the well-known AIC (Akaike, 1973; Efron, 1986, 2004). That is, the estimated risk is $-2\log f(Y;\hat{\mu}) + 2p$ plus a constant, so that the criterion picks the model with the largest maximized likelihood penalized by the number of parameters p. In this simple case the covariance correction is a constant that does not depend on the parameters of the model. However, for more complex models, such as those with mixed effects, the correction term in AIC could involve parameters of the model as well as the design matrix. See the next subsection on correlated outcomes.

4.2.4 Correlated Outcomes

The covariance penalty above holds when the y_is are correlated, since the derivation does not require independence. However, for estimation of the covariance penalty, most work, including resampling methods, has made use of an independence assumption (Zhang, 2004). Here we would like to briefly consider the case of clustered outcome data that often arise in cancer and other areas of biomedical research. Two types of approaches are commonly used: marginal and conditional. In the marginal approach, the parameters of the marginal distributions are of primary interest and are often estimated using generalized estimating equations, while the correlation structures are typically modeled as working assumptions (Diggle et al., 2002). In the conditional or random effects model approach, both the correlation and the cluster-specific parameters are of interest.

The AIC has been extended to both approaches mentioned here. Pan (2001) proposed an AIC for estimating equations in the marginal approach, the theory of which is based on an asymptotically quadratic approximation to the quasi-likelihood function. The equivalent of the number of degrees of freedom is computed as the trace of a matrix related to the sandwich estimate of the variances. For random effects models, Vaida and Blanchard (2005) proposed the concept of conditional Akaike information, which is defined as

$$\text{cAI} = -2E E_{Y^0|b} \log f(Y^0|\hat{b};\hat{\theta}), \tag{4.9}$$

where the future data Y^0 is sampled with the same (unobserved) random effects b as the original data Y, and θ denotes the population parameters in the model. The relevant likelihood here is the conditional likelihood of the observed data given the random effects. The emphasis here is on inference for both the fixed and the random effects, so that the corresponding conditional AIC (cAIC) can be used to compare models that treat the cluster effects as fixed effects with those that treat the cluster effects as random effects. The cAIC, the estimator of cAI, so far has been derived one model at a time. For the linear mixed-effects model, Vaida and Blanchard (2005) showed that the estimated optimism equals to the effective degrees of freedom of Hodges and Sargent (2001).

The covariance penalty argument appears in the derivation of cAIC, but it does not seem straightforward to use it for computation in practice.

4.2.5 Nuisance Parameters

Here we briefly discuss the types of data that are often modeled semiparametrically, such as survival outcomes. We use the more general framework of models with nuisance parameters, which includes parametric models as well.

Defining and estimating risks aimed at parameters of interest, and model selection based on that, was only developed rather recently. Hjort and Claeskens (2003) and Claeskens and Hjort (2003) considered parametric models and defined the limiting risk as mean squared error of the estimated parameter of interest under sequences of locally misspecified models (at the root-n rate). Claeskens and Carroll (2007) extended the result to semiparametric models, where the maximum likelihood estimators for parametric models are replaced by semiparametrically efficient profile estimators.

In the following we describe a risk function that is defined directly using the profile likelihood of the parameters of interest. Consider a family of models $\mathcal{M}$ parameterized by $\theta = (\phi, \lambda)$, where $\phi \in \Phi$ is the parameter of interest, and $\lambda \in \boldsymbol{\Lambda}$ is the nuisance parameter, possibly of infinite dimension. As in the deviance loss, the classical "distance" from the true distribution f to a member $g_\theta = g(\cdot|\phi, \lambda)$ of $\mathcal{M}$ is given by the Kullback-Leibler information (KL), $\text{KL}(f, g_\theta) = E\{\log f(Y) - \log g_\theta(Y)\}$. When the focus is on ϕ alone, the relevant distance is that between f and the

subfamily of models $\{g_{\phi,\lambda} : \lambda \in \boldsymbol{\Lambda}\}$: $\min_{\lambda\in\boldsymbol{\Lambda}} \mathrm{KL}(f, g_{\phi,\lambda})$. Suppose that the minimum is attained at some $\lambda = \tilde{\lambda}(\phi)$ for each ϕ, $\tilde{\lambda}(\phi)$ is in fact a least favorable curve under smoothness conditions (Severini and Wong, 1992; Fan and Wong, 2000), and $g_\phi = g(\cdot|\phi, \tilde{\lambda}(\phi))$ is the theoretical equivalent of the profile likelihood $\mathrm{pl}(Y;\phi)$ where the nuisance parameter λ is profiled out. Xu et al. (2008) shows that the minimum *KL* leads to the profile Akaike information

$$\mathrm{pAI} = -2EE_0\{\mathrm{pl}(Y^0; \hat{\phi}(Y))\}, \tag{4.10}$$

which is estimated by a profile Akaike information criterion that uses as penalty the dimension of ϕ under suitable conditions. Note that $\mathrm{pl}(Y^0; \hat{\phi}(Y))$ in (4.10) is different from the log-likelihood function computed at the maximum likelihood estimate $(\hat{\phi}, \hat{\lambda})$, since it allows maximizing the likelihood over λ based on the new data Y^0.

4.3 Resampling Methods

Using the covariance penalty, the risks above can be estimated whenever the covariance penalty can be estimated. Another approach to risk estimation, different from covariance penalty techniques, involves the use of (nonparametric) resampling methods to directly estimate the prediction error. In the following we give a brief description of a few resampling methods commonly used in estimating prediction error, and then discuss the parametric bootstrap for estimating the covariance.

Cross-validation

Although most readers of this book are probably familiar with the cross-validation procedure, for completeness we give a very brief description here. A K-fold cross-validation (randomly) divides the original sample into K approximately equal-sized parts, and takes turns using each of these parts as a test sample, with the rest as the training sample. The K-fold cross-validation estimate of the risk is

$$L^{CV} = \sum_{i=1}^{n} L(y_i, \hat{\mu}_i(Y^{-k(i)})), \tag{4.11}$$

where $Y^{-k(i)}$ denotes the training sample with the part containing y_i removed. The special case of $K = n$ is also called leave-one-out cross-validation.

Leave-one-out cross-validation provides an approximately unbiased estimate of risk; however, it has been shown to have large variance because the training samples are largely the same, one to another. On the other hand, K-fold cross-validation with smaller K tends to have smaller variance, at the expense of larger training-set-size bias; that is, the risk depends on the sample size of the training set. An illustration of such dependence can be found in Figure 7.8 of Hastie et al. (2001).

Nonparametric bootstrap

Another commonly used resampling method is bootstrap. For the deviance loss, for example, bootstrap methods have been shown to be asymptotically equivalent to the covariance penalty estimates for AIC (Shibata, 1997), but possibly with less bias in small samples (Cavanaugh and Shumway, 1997; Pan, 1999; Shang and Cavanaugh, 2008).

The nonparametric bootstrap estimate resamples with replacement and probability $1/n$ from the original n data points. The direct bootstrap estimate of the risk is

$$L^{\text{boot}} = E^*\{L(Y, \hat{\mu}(Y^*))\}, \tag{4.12}$$

where Y is the original data, Y^* is the bootstrapped data, and E^* is with respect to the bootstrap distribution, i.e., averaged over the bootstrap samples. On the other hand, a bias-corrected bootstrap estimate of the expected optimism for *i.i.d.* data is

$$\hat{O} = E^*\{L(Y, \hat{\mu}(Y^*)) - L(Y^*, \hat{\mu}(Y^*))\}, \tag{4.13}$$

so that the bias-corrected bootstrap estimate of the risk is $L(Y, \hat{\mu}) + \hat{O}$. Figure 4.1 shows the convergence of these two bootstrap estimates as functions of the number of bootstrap resamples. It is perhaps not surprising that the bias-corrected bootstrap takes longer to converge in comparison, since there is an extra term to be averaged in (4.13). We can also see what appears to be a bias correction effect in the plot.

Note that compared with the definition of loss, where Y^0 denotes future data independent of the original Y, the bootstrapped data Y^* are guaranteed to have some overlap with Y. In fact,

$$P(y_i \in \text{ a bootstrapped sample}) = 1 - (1 - \frac{1}{n})^n \approx 1 - e^{-1} = 0.632.$$

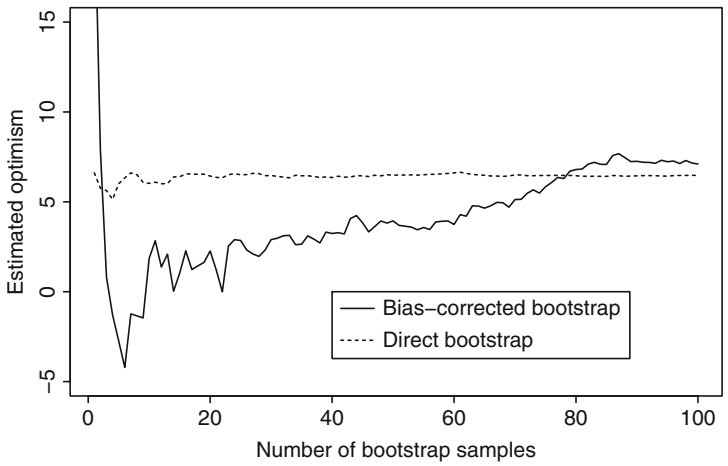


Fig. 4.1 Convergence of bootstrap estimates. *Solid line*: Bias-corrected; *dashed line*: Direct bootstrap.

This overlap can lead to underestimation of the risk. To improve on the bootstrap estimate (4.13), we can try to use only those data points that are not contained in a particular bootstrap sample to assess the error. The leave-one-out bootstrap estimate is given by

$$L^{(1)} = \sum_{i=1}^{n} \frac{1}{|\mathcal{B}^{-i}|} \sum_{b \in \mathcal{B}^{-i}} L(y_i, \hat{\mu}_i(Y^b)), \tag{4.14}$$

where $\mathcal{B}^{-i}$ is the set of bootstraps that do not contain observation i in the samples, and $|\cdot|$ denotes the size of a finite set. The leave-one-out bootstrap is analogous to cross-validation. Alternatively, Pan (1999) proposed a bootstrap-smoothed cross-validation (BCV) estimator:

$$L^{\mathrm{BCV}} = E^* \left\{ \frac{n}{|Y - Y^*|} \sum_{i \in Y - Y^*} L(y_i, \hat{\mu}_i(Y^*)) \right\}, \tag{4.15}$$

where $Y - Y^*$ denote the set of observations in Y that are not in the bootstrapped sample Y^*. Pan (1999) applied (4.15) to the deviance. One advantage of (4.15), as compared with (4.14), appears to be for loss functions that are not sums of i.i.d. terms, like the Cox (1975) partial likelihood.

Both of the above, however, suffer from the training-set-size bias mentioned above. Typically, this bias leads to overestimation of the risk. To correct for this bias, the *.632 estimator* (Efron, 1983) uses

$$0.368\, L(Y, \hat{\mu}) + 0.632\, L^{(1)}; \tag{4.16}$$

and similarly, the *.632 BCV estimator* is

$$0.368\, L(Y, \hat{\mu}) + 0.632\, L^{\mathrm{BCV}}. \tag{4.17}$$

Parametric bootstrap

Finally, as mentioned before, the parametric or model-based bootstrap can be used to estimate the covariance penalty. This type of bootstrap can also be useful for estimating certain risk functions directly, when the nonparametric bootstrap runs into difficulties due to small subsample sizes, for example (Donohue et al., 2007). An immediate question is what model to use to generate the bootstrap samples. Since we do not know or assume a true model in this case, the idea is to take a "moderately large" model. For Gaussian data, for example, we would want to generate bootstrap data from some $N(\tilde{\mu}, \tilde{\sigma}^2 I)$, where $\tilde{\mu}$ and $\tilde{\sigma}^2$ are obtained from fitting some large model to the data (Efron, 2004). A large model could be one that incorporates many predictors, or a smooth fit, etc. The "ultimate" large model is $N(Y, \tilde{\sigma}^2 I)$, which is in fact "model-free". This relates to the data perturbation (little bootstrap) techniques of Breiman (1992), Ye (1998), and Shen and Ye (2002), which generates data from

$N(Y, c\tilde{\sigma}^2 I)$ with $c < 1$. According to Efron (2004), the exact choice of the model is often unimportant. On the other hand, Shen et al. (2004) postulates that the resulting risk estimate and model selection will depend on the bootstrap model. So far we have not seen extensive studies in the literature on the effect of different model choices for generating parametric bootstrap data in the context of risk estimation, and future research on the topic remains an open area.

4.3.1 Empirical Studies

Various empirical studies of risk estimates can be found in the literature. One interesting, albeit perhaps unfortunate, phenomenon that has been repeatedly observed is that the actual optimism of the apparent loss tends to correlate negatively with its estimates. For example, Efron (1982), Table 7.2, illustrated this point in the ten trial runs of simulation, estimating the optimism using bootstrap, cross-validation, or jackknife. Efron (1983) observed the same phenomenon. Therefore the best one can hope for is to estimate the expected optimism (Efron, 2004).

With high-dimensional data in mind, a rather extensive simulation study can be found in Braga-Neto and Dougherty (2004). The context they considered was using microarray gene expression data for class prediction, and so the loss function was counting error. The authors compared three methods of estimating the risk: the apparent loss, cross-validation, and the bootstrap. For cross-validation the comparison included leave-one-out, fivefold (CV5), tenfold (CV10), and a repeated CV10, which averaged 10 runs of tenfold cross-validation with randomly picked folds. For the bootstrap they considered two nonparametric methods: the bias-corrected bootstrap of (4.13) and the .632 estimator.

Three classification methods were considered as examples: linear discriminant analysis (LDA), 3-nearest-neighbor (3NN), and classification trees (CART); more details on the methods can be found in their paper and relevant chapter(s) of this monograph. Similar to the Efron simulations mentioned earlier, the authors studied the difference between an error estimate and the true error rate of a classifier on a given (simulated) dataset. This difference was termed the *deviation*, and summary statistics and plots were given of the distribution of deviations. The true error rate was computed exactly for LDA, and by Monte-Carlo for 3NN and CART.

Their experiments included six different sample sizes: 20, 40, 60, 80, 100, and 120, and two different dimensionalities (number of genes): $p = 2$ and 5. For each case three different combinations of class separation and variance are used, corresponding to Bayes error rates of $\sim$0.1, 0.15, and 0.2, given equal prior probabilities of the two classes. Clear distinctions exist among the seven methods for estimating risk with the smaller sample sizes, especially with $n = 20$ and 40. For $n \geq 100$, the variances of the deviation distributions are essentially the same, with the .632 bootstrap having the smallest variance; their means are also essentially zero with the exception of the apparent loss.

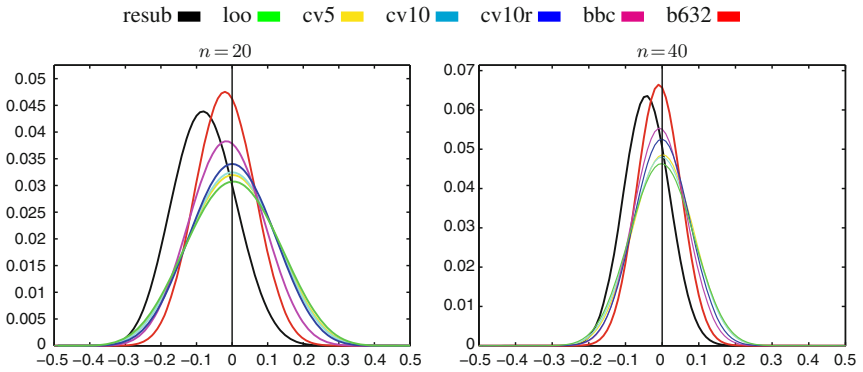


Fig. 4.2 Beta-fits of empirical deviation distribution. resub: "apparent" estimate; loo: leave-one-out; cv10r: repeated CV10; bbc: bias-corrected bootstrap; b632: .632 bootstrap.

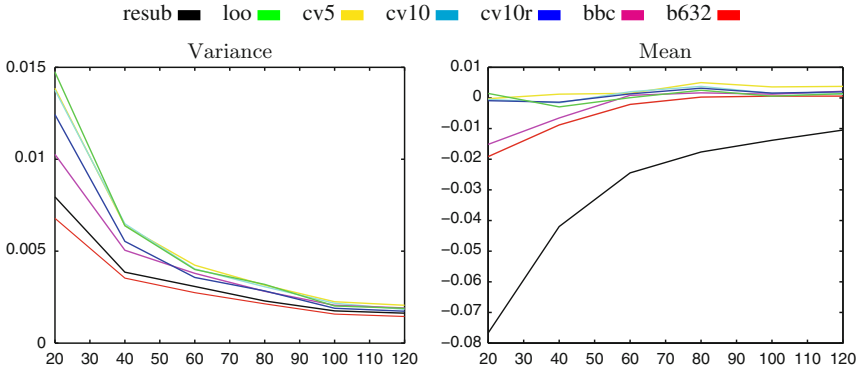


Fig. 4.3 Plots of the empirical deviation distribution.

For the smaller sample sizes, the cross-validation estimates are seen to have rather high variability, although all the cross-validation methods appear to perform similarly. Figures 4.2 and 4.3 showcase one such experiment ("Experiment 3" in Braga-Neto and Dougherty, 2004). The classifier here is LDA. Note that, in their published paper, the fitted curves using a Beta distribution for $n = 20$ were slightly in error. The figures here are kindly provided by the authors and are the same as the correct version, on the companion Web site for their paper. In the experiments, the cross-validation methods also tend to produce large outliers, which in practice could lead to wrong conclusions. The distinction between cross-validation estimates and the other estimates increases with the complexity of the classification, from LDA (simplest) to CART (most complex). On the other hand, the .632 bootstrap estimator has the best overall performance in the simulations. This, of course, comes with an increased computational cost, which could be substantial when the dimension of data (number of genes) is high. The computation time, along with other detailed summaries of all simulation scenarios, are also given on their paper's companion Web site.

4.4 Applications of Risk Estimation

In this section we discuss a few different applications of risk estimation. In addition to practical applications, we also present some theoretical ones, in the first two subsections. For these theoretical applications, instead of a technical in-depth treatment, we aim to give the reader a taste of the material.

4.4.1 SURE and Admissibility

Let $R(\theta,\delta) = E_\theta L[\theta,\delta(X)]$ be the risk in using δ, as an estimator of the parameter θ. An estimator δ is minimax, when $\sup_{\theta\in\Theta} R(\theta,\delta) = \inf_{\delta'} \sup_{\theta\in\Theta} R(\theta,\delta')$, where the inf is computed over all measurable δ'. An estimator δ' is said to be inadmissible, if there is an estimator δ, where $R(\theta,\delta) \le R(\theta,\delta')$, for all $\theta\in\Theta$, with strict inequality for at least one θ. An admissible estimator is any estimator that is not inadmissible. Stein (1973) showed that risk estimates could be used to prove the inadmissibility of standard estimators in a variety of multivariate estimation problems.

Suppose that we are interested in estimating a mean vector μ based on n-variate Gaussian data $y \sim N(\mu,\sigma^2 I_n)$, where I_n is the n-by-n identity matrix, using estimates of the form $\hat{\mu}(y) = y + \gamma(y)$. Stein's identity

$$E\{(y_i-\mu_i)\gamma_i(y)\} = \sigma^2 E\left\{\frac{\partial\gamma_i}{\partial y_i}\right\}$$

implies that the risk of the estimator is

$$\begin{aligned}\sum_{i=1}^n E(y_i-\mu_i+\gamma_i(y))^2 &= n\sigma^2 + 2\sum_{i=1}^n E\{(y_i-\mu_i)\gamma_i(y)\} + E\|\gamma(y)\|^2\\ &= n\sigma^2 + 2\sigma^2 E\{\operatorname{div}\gamma(y)\} + E\|\gamma(y)\|^2,\end{aligned}$$

where $\operatorname{div}\gamma(y) = \sum_{i=1}^n \partial\gamma/\partial y_i$, so that an unbiased estimate of the risk is $n\sigma^2 + 2\sigma^2\operatorname{div}\gamma(y) + \|\gamma(y)\|^2$.

It is known that if, for every $y\in\mathbf{R}^n$,

$$2\sigma^2\operatorname{div}\gamma(y) + \|\gamma(y)\|^2 \le 0,$$

then $\hat{\mu}$ is minimax, with risk smaller than y. Such γ's exist, as solutions to the differential inequality above, demonstrating that y is not an admissible estimate of μ in the Gaussian case (with $n\ge 3$; see below).

The function

$$\gamma(y) = \left(\frac{2-n}{\|y\|^2}\right) y$$

satisfies the differential inequality above (for $n\ge 3$) and leads to the James-Stein estimate

$$\hat{\mu}(y) = \left(1 - \frac{n-2}{\|y\|^2}\right) y.$$

Formal Bayes estimates of μ with prior π and posterior m are of the form $y + \nabla \log(m(y))$, where ∇f is the gradient of a function f. In this case, Stein's condition for minimaxity becomes $\Delta\sqrt{m(y)} \leq 0$, where Δf is the Laplacian of f, with $\Delta = \sum_{i=1}^{n} \partial^2/\partial x_i^2$. A function f for which $\Delta f \leq 0$ is called superharmonic.

The superharmonicity of $\sqrt{m}$, required above, can be difficult to verify. Although superharmonicity of m implies the superharmonicity of $\sqrt{m}$, if one uses a proper prior π, the induced marginal (posterior) cannot be superharmonic (Fourdrinier et al., 1998). In fact, Strawderman (1971) shows that proper minimax Bayes estimates of the multivariate normal mean do not exist for $n < 5$, but do exist whenever $n \geq 5$, in which case the Cauchy prior can be used.

The James-Stein estimate is dominated by its positive part version

$$\hat{\mu}(y) = \left(1 - \frac{n-2}{\|y\|^2}\right)_+ y,$$

which avoids sign changes. It turns out that neither the James-Stein estimate nor its positive part version is admissible, the former because it is dominated by the latter and the latter because it is nondifferentiable, but improving on the positive part version is quite difficult.

If the dimension $n \geq 3 + k$ for some integer $k > 0$, it is also possible to develop estimators that shrink toward a k-dimensional subspace. Thus, if $n \geq 4$, there is an estimator

$$\hat{\mu}(x) = \bar{x}\mathbf{1} + \left(1 - \frac{n-3}{\|x - \bar{x}\mathbf{1}\|^2}\right)(x - \bar{x}\mathbf{1})$$

that shrinks toward the grand mean. This has applications in the analysis of hierarchical linear models, where, for example, it can be shown that certain fixed effects models are inadmissible.

Stein's identity can be extended to other exponential families (Brown, 1986). Indeed, shrinkage is generally applicable in multidimensional estimation problems, and is also useful in nonparametric regression (Johnstone, 2002; Candès, 2006). Consider, for example, an orthogonal transformation of independent Gaussian data y into some other basis. By orthogonality, the resulting transformed random variable also consists of independent Gaussian components and we can apply SURE to shrink the coefficient estimates toward zero. Of course, if we choose a basis in which only a few of the components are likely to be nonzero – that is, a basis in which the unknown mean μ is sparse – then other techniques that take this knowledge into account, for example, using a double exponential prior on the coefficients, rather than the Gaussian prior implicitly used in the James-Stein estimate, lead to improvements. Silverman (1984) explains the connection between kernel, spline, and orthogonal series estimates.

4.4.2 Finite Sample Risk and Adaptive Regression Estimates

Risk estimates can also be used to derive nonparametric penalized empirical loss estimates, which adapt to the unknown smoothness of the function of interest. See Barron et al. (1999) for more details.

Suppose that we have *i.i.d.* pairs (x_i, y_i) with $y_i = g(x_i) + \varepsilon_i$, where the ε_i are mean 0, variance σ^2 and uniformly subgaussian or, $\sup_i P(|\varepsilon_i| > t) < C\exp(-t^2/2)$ for some $0 < C < \infty$ (van der Vaart and Wellner, 1996). Our goal is to find the best estimate of g from a class $\mathcal{M} = \mathcal{M}_n$ of (linear) models indexed by m, for example, splines or orthogonal series. More precisely, denote by $u_m = H_m u$ the projection of a vector u on the space defined by the (linear) model m. Let $\|u\|^2 = (u,u)$, with $(u,v) = n^{-1}\sum u_i v_i$, and $d_m = \text{trace}(H_m)$ be the dimension of the model m. Again let $Y = (y_1, ..., y_n)'$ and with a slight abuse of notation, let $g = (g(x_1), ..., g(x_n))$. Then we want to find $m^* \in \mathcal{M}$ such that

$$\begin{aligned} m^* &= \text{arginf}_{\mathcal{M}} E\|Y_m - g\|^2 \\ &= \text{arginf}_{\mathcal{M}} \left\{ \|g_m - g\|^2 + \sigma^2 \frac{d_m}{n} \right\}. \end{aligned}$$

This perspective of simply searching for the minimum risk model in some class of models has its advantages: There is no need to assume that any of the models is "correct"; that is, there may be no model m in the class $\mathcal{M}$ with g a fitted version of m. For example, $\mathcal{M}$ may be the class of piecewise constant regression estimates (regressograms) with g continuous and nonconstant. Even if there is a model $m \in \mathcal{M}$ with $g \in m$, it may be the case that $m^* \neq m$. For example, $\mathcal{M}$ may be the class of linear combinations of sinusoids with g a very high frequency sine-function; in small samples it may be impossible to resolve g and the risk-minimizing model m^* would be the sample mean.

We know from above that Mallows' C_p

$$\|Y - Y_m\|^2 + 2\sigma^2 \frac{d_m}{n} \tag{4.18}$$

is unbiased for the finite sample risk

$$E\|Y_m - g\|^2 + \sigma^2, \tag{4.19}$$

which is (essentially) what we want to minimize. This suggests that we look at penalized least squares estimates of g, with penalty terms that look something like $2\sigma^2 d_m/n$. This technique would work perfectly, if we could guarantee that (4.18) is uniformly close to (4.19). Unfortunately, this is not always the case. So, we aim for a more realistic goal: We look for penalties $\text{pen}(m) \geq 2\sigma^2 d_m/n$ such that

$$E\|\hat{Y} - g\|^2 \leq C \inf_{\mathcal{M}} \left\{ \|g_m - g\|^2 + \text{pen}(m) \right\} + O\left(\frac{1}{n}\right) \tag{4.20}$$

for some $C < \infty$, where

$$\hat{m} = \text{arginf}_{\mathcal{M}} \left\{ \|Y - Y_m\|^2 + \text{pen}(m) \right\}$$

and $\hat{Y} = Y_{\hat{m}}$. Note that an estimate satisfying such a bound would be optimal in a nonasymptotic sense and the class $\mathcal{M} = \mathcal{M}_n$ of models can change with n.

In an effort to be more precise about the meaning of models and classes of models, consider the class $\mathcal{M}$ of piecewise constant regression estimates

$$\hat{f}(x) = \sum_{j=1}^{k} \hat{f}_j \cdot 1\{x \in I_j\}$$

where I_j is the interval from t_{j-1} to t_j, and the $\hat{f}_j$s are parameter estimates. This is the class of regressograms (Tukey, 1961). Each model in this class is identified with a unique sequence of knots $\{t_j\}$ and estimation in that model corresponds to the construction of estimates for each of the regression parameters $\{f_j\}$. The class is formed by considering all possible sequences of knots of a certain size. Regular regressograms are those for which the bin width $h_j = t_j - t_{j-1}$ is constant in j. A locally adaptive regressogram has variable width bins. The dimension of a regressogram corresponds to the number of intervals I_j in the model. Regular regressograms have only one model for each dimension k; locally adaptive regressograms can have more than one. Our model selection problem for regressograms becomes: How many knots should there be? And, in the case of locally adaptive regressograms, where should they be?

Let $m^* = \inf_{\mathcal{M}} \left\{ \|g - g_m\|^2 + \text{pen}(m) \right\}$ with $\text{pen}(m)$ arbitrary at this point and take $g^* = g_{m^*}$. By definition, our minimum penalized least-squares estimator $\hat{Y}$ satisfies the basic inequality

$$\|Y - \hat{Y}\|^2 + \widehat{\text{pen}} \leq \|Y - g^*\|^2 + \text{pen}^* \tag{4.21}$$

where $\widehat{\text{pen}} = \text{pen}(\hat{m})$ and $\text{pen}^* = \text{pen}(m^*)$.

Expanding the above inequality about g implies

$$\|Y - g\|^2 \leq 2(\varepsilon, \hat{Y} - g^*) + \|g^* - g\|^2 + \text{pen}^* - \widehat{\text{pen}}, \tag{4.22}$$

and we want to bound the expected value of the left hand side of (4.22) by something like the right hand side of (4.20). That is, we want $(\varepsilon, \hat{Y} - g^*)$ to be uniformly close to something like

$$C \left[\left\{ \|\hat{Y} - g\|^2 + \|g^* - g\|^2 \right\} + \widehat{\text{pen}} \right].$$

To that end, define $p(m) = \sigma^2 \text{pen}(m)/n$ and for arbitrary $s > 0$,

$$\begin{aligned} w(u_m) &= \frac{1}{2} \left[\frac{1}{2} \left\{ \|u_m - g\|^2 + \|g^* - g\|^2 \right\} + \text{pen}(m) + \frac{\sigma^2}{n} s \right] \\ &\geq \frac{1}{2} \left[\frac{1}{2} \|u_m - g\|^2 + \text{pen}(m) + \frac{\sigma^2}{n} s \right] \\ &\geq \frac{\sigma}{\sqrt{2}} \|u_m - g\| \left[\frac{p(m) + s}{n} \right]^{\frac{1}{2}}. \end{aligned}$$

Take $Z(u_m) = (\varepsilon, u_m - g^*)/w(u_m)$. The uniform subgaussianity of the ε_i implies that

$$\begin{aligned} P\left(\sup_u Z(u_m) > \lambda\right) &\le P\left(\sup_u \frac{(\varepsilon, u_m - g^*)}{\sigma\|u_m - g^*\|} > \frac{\lambda}{\sqrt{2}}\left[\frac{p(m)+s}{n}\right]^{\frac{1}{2}}\right) \\ &\le P\left(\frac{\|\varepsilon\|^2}{\sigma^2} > \frac{\lambda^2}{2}\frac{p(m)+s}{n}\right) \\ &\le A\exp\left(-\frac{\lambda^2}{2}p(m)\right)\exp\left(-\frac{\lambda^2}{2}s\right), \end{aligned}$$

so that if we take $p(m)$ large enough that

$$\sum_{m\in\mathcal{M}} \exp\left(-\frac{p(m)}{2}\right) \le C, \tag{4.23}$$

and let $V(s)$ be the event that $\sup_m \sup_u Z(u_m) > 1$, then

$$\begin{aligned} PV(s) &\le \sum_{m\in\mathcal{M}} P\left(\sup_u Z(u_m) > 1\right) \\ &\le \sum_{m\in\mathcal{M}} A\exp\left(-\frac{p(m)}{2}\right)\exp\left(-\frac{s}{2}\right) \\ &\le AC\exp\left(-\frac{s}{2}\right) \end{aligned}$$

and, on $V(s)^c$, we have

$$2(\varepsilon, \hat{Y} - g) \le \frac{1}{2}\left\{\|\hat{Y} - g\|^2 + \|g^* - g\|^2\right\} + \widehat{\text{pen}} + \frac{\sigma^2}{n}s,$$

or

$$\|\hat{Y} - g\|^2 \le 3\left\{\|g^* - g\|^2 + \text{pen}^*\right\} + 2\frac{\sigma^2}{n}s. \tag{4.24}$$

Now all we need to do to prove the bound is to integrate over s. Indeed, if we take

$$R = \left(\|\hat{Y} - g\|^2 - 3\left\{\|g^* - g\|^2 + \text{pen}^*\right\}\right)_+ \ge 0$$

where $(x)_+ = \max(x, 0)$, then

$$P(R > r) \le C\exp\left(-\frac{nr}{4\sigma^2}\right)$$

with $r = 2\sigma^2 s/n$, and

$$E(R) = \int_0^\infty P(R > r)\,dr \le D\frac{\sigma^2}{n}.$$

This gives us the bound

$$E\|\hat{Y}-g\|^2 \leq 3\left\{\|g^*-g\|^2+\text{pen}^*\right\}+D\frac{\sigma^2}{n}. \tag{4.25}$$

So, what is the point of all this?

Suppose that we are interested in using regular regressograms to estimate the regression function g. If we knew that g satisfied a Hölder condition of the form $\|g(s)-g(t)\| \leq M|s-t|^\alpha$, then we could construct a regressogram estimate achieving the minimax rate of $O\left(n^{-\frac{2\alpha}{2\alpha+1}}\right)$ by carefully selecting the number of bins k in the estimate (Birman and Solomjak, 1967). The problem is that α is usually unknown. Of course, approximation theory tells us that

$$\|g_m-g\|^2 = O\left(k^{-2\alpha}\right)$$

and, using the argument above, if we take $\text{pen}(m) = 2\sigma^2 k/n$, then the constraint (4.23) is satisfied, and we have

$$E\|\hat{Y}-g\|^2 \leq 3\min_k\left\{C_0k^{-2\alpha}+C_1\frac{k}{n}\right\}+O\left(\frac{1}{n}\right) = O\left(n^{-\frac{2\alpha}{2\alpha+1}}\right). \tag{4.26}$$

This implies that we can use penalized least squares to obtain a regressogram with the minimax rate of convergence without having to know much about the smoothness of the underlying regression function g. That is, the penalized least-squares estimate adapts to the unknown smoothness in g.

If we consider a broader class of functions g and locally adaptive regressograms, we find that we need a stronger penalty and the corresponding penalized least squares estimates come within a log factor of the minimax rates. Indeed, if g is known to have finite α-variation (Mammen and van de Geer, 1997; Barron et al., 1999), then the minimax rate of convergence is $O\left(n^{-\frac{2\alpha}{2\alpha+1}}\right)$ as before, but it is known that a locally adaptive estimate is required to achieve this rate. Approximation theory (DeVore, 1998) tells us that

$$\min_{d_m=k}\|g_m-g\|^2 = O\left(k^{-2\alpha}\right),$$

but now there are $\Gamma(n)/\Gamma(k)\Gamma(n-k)$ models of size k. So, for the constraint (4.23) to be satisfied, we have to use a penalty such as $\text{pen}(m) = 2\sigma^2 k\ \log n/n$. This estimate achieves a

$$E\|\hat{Y}-g\|^2 = O\left(\left[\frac{n}{\log n}\right]^{-\frac{2\alpha}{2\alpha+1}}\right)$$

rate of convergence, which is within a log factor of the minimax rate. Unfortunately, the corresponding model selection problem is *NP*-hard (Natarajan, 1995), and we have to look at other techniques for (nearly) adaptive estimation of functions of finite α-variation. Regression tree approaches look promising in this regard.

Of course, there are many extensions of the ideas above. An easy extension is to (nearly) adaptive penalized quasi-likelihood estimators. The theory can also be

extended, with some difficulty, to continuously parameterized models, for example, kernel estimates.

The theory for general loss functions and nonlinear models is worked out in detail in Barron et al. (1999), where an example very similar to the problem above is worked out, as well.

4.4.3 Model Selection

The most obvious practical application of risk estimation is model selection. Accurate risk estimation aids in accurate model selection, although in some cases, for example, considering models of only a certain restricted size, even the apparent loss might work well. But more generally, a reliable risk estimation method is a prerequisite for reliable model selection. Some important methods of variable selection are described in Chapter 2.

An interesting aspect is the comparison of estimated risks from different models. Since the estimated risks themselves are subject to random variation, for a given set of models the ranking of the estimated risks may not reflect the ranking of the underlying true errors. From this point of view, it is useful to assess the significance of the estimated risk differences. This appears to be an underdeveloped field. For AIC, Burnham and Anderson (2002) recommended a difference of at least two as evidence in favor of the model with the smaller AIC; a more rigorous theory on the differences in AICs can be found in Vuong (1989).

There are broader aspects of model fitting, selection, and validation. Hand (2006) cautions that there are important aspects of "real problems" to be accounted for in addition to the statistical estimation of prediction error. These include the fact that model-fitting itself is a sequential process of progressive refinement, the assumption that the current data are randomly drawn from the same distribution as future data, error or change in class labels, the choice of loss function, and limitations of empirical evaluation such as the particular data sets used and the experience of the researcher. In the context of biomarker discovery and validation, Feng et al. (2004) summarized research issues and possible strategies for genomic and proteomic studies.

4.4.4 Gene Ranking

There is a close relationship between gene ranking and model selection, as the former is a special case of the latter. Braga-Neto et al. (2004) considered the ranking of small gene feature sets, consisting of three or four genes, using different methods to estimate the prediction error. Xu and Li (2003) and Lu et al. (2007) considered ranking of genes one-by-one.

For ranking genes one-by-one, p-values are often used, computed either directly under parametric assumptions or based on permutation tests. The p-values under parametric models are equivalent, for example, to the deviance loss, if the likelihood ratio test is used. For permutation tests, there is also a model for the loss although it is more complicated, depending on the design and assumptions such as symmetry, homoscedasticity, etc. (Hajek et al., 1999; Romano, 1990). Once again, a question that needs to be addressed is the variability associated with the ranking. Zhang et al. (2006) addressed the problem in a hypothesis testing framework. Xu and Li (2003) described the concept of *rediscovery* as a way to summarize such variability. The rediscovery rate, for an integer $k > 0$, is defined as the probability that the k-th top ranked gene will have a rank of at least k in an independently replicated experiment. Xu and Li (2003) proposed a nonparametric bootstrap method to estimate the rediscovery rate. In their example of short time course leukemia cell line data, parametric (assuming normality of regression errors) and permutation F-statistics were used to rank the candidate genes. For $k \leq 100$, they found that the parametric method had a rediscovery rate of only about 10% (Figure 4.4b), while the permutation method had a higher rediscovery rate of 53%. Note that the width of the confidence intervals in Figure 4.4b depends only on the number of bootstrap samples, an increase of which could lead to substantial computational burden when the permutation tests are used.

Lu et al. (2007) further developed the concept of rediscovery. Their definition of the rediscovery rate was slightly different, as the probability of the top k genes from the original data being selected among top k again in an independently replicated experiment. They also proposed using the bootstrap to estimate the rediscovery rates,

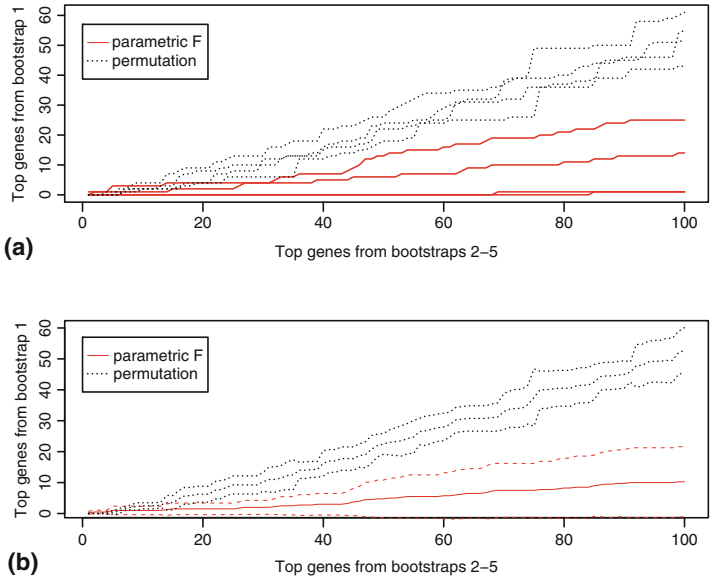


Fig. 4.4 Rediscovery rates estimated using bootstrap: **(a)** 5 individual runs; **(b)** estimated rediscovery rates and their 95% confidence intervals.

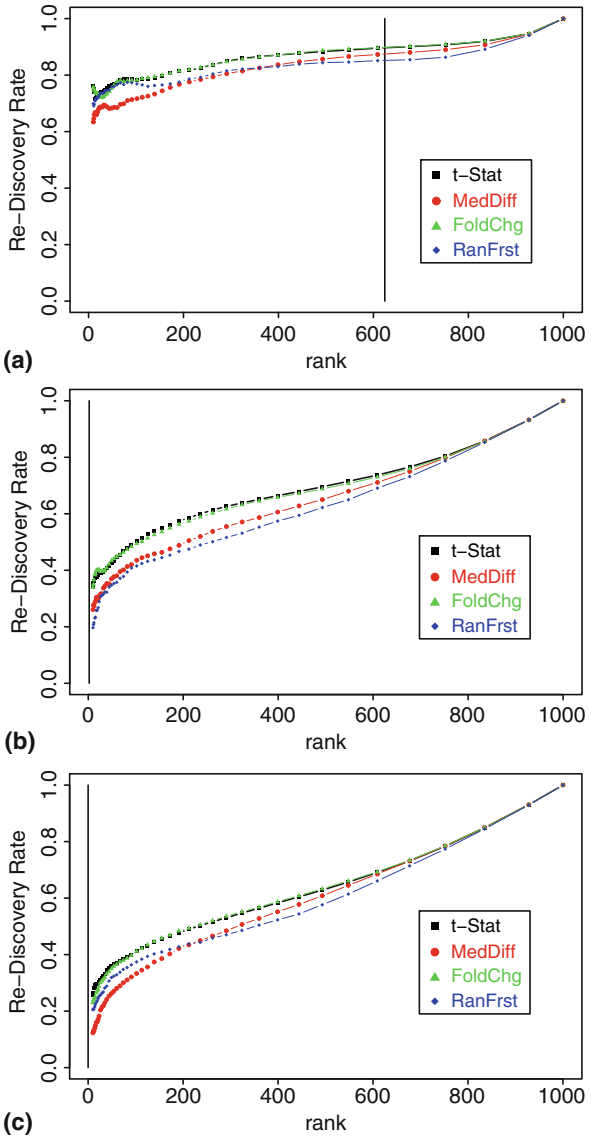


Fig. 4.5 RDCurve of gene selection associated with ER status and lymph node metastasis status, and a random data set. **(a)** RDCurve of gene selection for ER status; **(b)** RDCurve of gene selection for lymph metastasis status; **(c)** RDCurve of gene selection from noninformative data set. The vertical lines correspond to the number of genes selected with FDR < 0.05.

plotting them against k to obtain *rediscovery curves* (RDCurves). Figure 4.5 shows an example from a breast cancer data set. In the data set the genes are ranked according to their differential expressions under two conditions separately: estrogen receptor (ER) status and lymph node metastasis status. It is known that microarray

gene expressions are often informative of a patient's ER status. On the other hand, the lymph node metastasis is more of a "downstream" event, with much weaker molecular signals. Figure 4.5 shows that the rediscovery rates (using t-statistic, median difference, fold change, or random forest to rank the genes) are much higher for ranking according to differential expressions between ER positive and ER negative patients, but substantially lower for ranking according to differential expressions between the lymph node metastasis status. The latter is in fact similar to the RDCurves for a simulated noninformative data set. This example illustrates the RDCurves as a means of depicting the signal-to-noise ratio in a given data set.

Acknowledgment The authors thank Michael Donohue and Xin Lu for on-going joint work on risk estimation under the proportional hazards mixed-effects model and on gene discovery and rediscovery, which has motivated a number of thoughts in the process of writing this chapter. This work was partially supported by the National Institutes of Health grant R01DK075128 and a grant from Bayer Pharmaceuticals.

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Proceedings of the Second International Symposium on Information Theory*, pp. 267–281, B.N. Petrov and F. Caski, eds. Akademiai Kiado, Budapest.

Barron, A., Birgé, L., and Massart, P. (1999). Risk bounds for model selection by penalization. *Probability Theory and Related Fields*, 113:301–413.

Birman, M. S. and Solomjak, M. Z. (1967). Piecewise polynomial approximations of functions in the class w_p^α. *Matematicheskii Sbornik*, 73:331–355.

Braga-Neto, U., Hashimoto, R., Dougherty, E. R., and Carroll, R. J. (2004). Is cross-validation better than resubstitution for ranking genes? *Bioinformatics*, 20:253–258.

Braga-Neto, U. M. and Dougherty, E. R. (2004). Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20:374–380.

Bregman, L. M. (1967). The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217.

Breiman, L. (1992). The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error. *Journal of the American Statistical Association*, 87:735–754.

Brown, L. D. (1986). *Fundamentals of Statistical Exponential Families*. IMS Lecture Notes Monograph, Hayward.

Burnham, K. P. and Anderson, D. R. (2002). *Model Selection and Multimodel Inference: A Practical Information – Theoretic Approach*, 2nd edn. Springer, New York.

Candès, E. J. (2006). Modern statistical estimation via oracle inequalities. *Acta Numerica*, 15:257–325.

Cavanaugh, J. E. and Shumway, R. H. (1997). A bootstrap variant of aic for state-space model selection. *Statistica Sinica*, 7:473–496.

Claeskens, G. and Hjort, N. L. (2003). The focused information criterion. *Journal of the American Statistical Association*, 98:900–916.

Claeskens, G. and Carroll, R. J. (2007). An asymptotic theory for model selection inference in general semiparametric problems. *Biometrika*, 94:249–265.

Cox, D. (1975). Partial likelihood. *Biometrika*, 62:269–276.

DeVore, R. A. (1998). Nonlinear approximation. *Acta Numerica*, 7:51–150.

Diggle, P., Heagerty, P., Liang, K. Y., and Zeger, S. (2002). *Analysis of Longitudinal Data*. Oxford University Press, New York.

Donohue, M., Xu, R., Gamst, A., Vaida, F., and Harrington, D. P. (2007). Model selection under the proportional hazards mixed-effects model. *Proceedings of the 2007 Joint Statistical Meetings*, CD-ROM.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans*. SIAM.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78:316–331.

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

Efron, B. (2004). The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99:619–642.

Fan, J. and Wong, W. H. (2000). Discussion of 'On profile likelihood', in Murphy, S. A. and van der Vaart, A. W., eds. *Journal of the American Statistical Association*, 95:468–471.

Feng, Z., Prentice, R., and Srivastava, S. (2004). Research issues and strategies for genomic and proteomic biomarker discovery and validation: a statistical perspective. *Pharmacogenomics*, 5:709–719.

Fourdrinier, D., Strawderman, W. E., and Wells, M. T. (1998). On the construction of Bayes minimax estimators. *Annals of Statistics*, 26:660–671.

Hajek, J., Sidak, Z., and Sen, P. K. (1999). *The Theory of Rank Tests, 2nd edn.* Academic Press, New York.

Hand, D. J. (2006). Classifier technology and the illusion of progress. *Statistical Science*, 21:1–14.

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

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer, New York.

Hjort, N. L. and Claeskens, G. (2003). Frequentist model average estimators. *Journal of the American Statistical Association*, 98:879–899.

Hodges, J. S. and Sargent, D. J. (2001). Counting degrees of freedom in hierarchical and other richly parameterized models. *Biometrika*, 88(2):367–279.

Johnstone, I. M. (2002). Function estimation and gaussian sequence models. http://www-stat.stanford.edu/ĩmj/baseb.pdf.

Lu, X., Gamst, A., and Xu, R. (2007). On gene discovery and rediscovery. In *Proceedings of the 2007 Joint Statistical Meetings*. CD-ROM.

Mallows, C. L. (1973). Some comments on C_p. *Technometrics*, 15:661–675.

Mammen, E. and van de Geer, S. (1997). Local adaptive regression splines. *Annals of Statistics*, 25:387–413.

Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24:227–234.

Pan, W. (1999). Bootstrapping likelihood for model selection with small samples. *Journal of Computational and Graphical Statistics*, 8:687–698.

Pan, W. (2001). Model selection in estimating equations. *Biometrics*, 57:120–125.

Romano, J. P. (1990). On the behavior of randomization tests without a group invariance assumption. *Journal of the American Statistical Association*, 85:686–692.

Severini, T. A. and Wong, W. H. (1992). Profile likelihood and conditionally parametric models. *Annals of Statistics*, 20:1768–1802.

Shang, J. and Cavanaugh, J. E. (2008). Bootstrap variants of the Akaike information criterion for mixed model selection. *Computational Statistics and Data Analysis*, 52:2004–2021.

Shen, X. and Ye, J. (2002). Adaptive model selection. *Journal of the American Statistical Association*, 97:210–221.

Shen, X., Huang, H., and Ye, J. (2004). Comment to Efron (2004). *Journal of the American Statistical Association*, 99:634–637.

Shibata, R. (1997). Bootstrap estimate of Kullback-Leibler information for model selection. *Statistica Sinica*, 7:375–394.

Silverman, B. W. (1984). Spline smoothing: The equivalent kernel method. *Annals of Statistics*, 12:898–916.

Stein, C. (1973). Estimation of the mean of a multivariate normal distribution. *Proceedings of Prague Symposium on Asymptotic Statistics*, pp. 345–381.

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

Strawderman, W. E. (1971). Proper Bayes minimax estimators of the multivariate normal mean. *Annals of Mathematical Statistics*, 42:385–388.

Tukey, J. W. (1961). Curves as parameters and touch estimation. *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*.

Vaida, F. and Blanchard, S. (2005). Conditional Akaike information for mixed-effects models. *Biometrika*, 92:351–370.

van der Vaart, A. W. and Wellner, J. A. (1996). *Weak Convergence and Empirical Processes*. Springer, New York.

Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, 57:307–333.

Xu, R. and Li, X. (2003). A comparison of parametric versus permutation methods with applications to general and temporal microarray gene expression data. *Bioinformatics*, 19:1284–1289.

Xu, R., Vaida, F., and Harrington, D. (2008). Using profile likelihood for semiparametric model selection with application to proportional hazards mixed models. *Statistica Sinica*, in press.

Ye, J. (1998). On meausuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association*, 93:120–131.

Zhang, C. (2004). Comment to Efron (2004). *Journal of the American Statistical Association*, 99:637–640.

Zhang, C., Lu, X., and Zhang, X. (2006). Significance of gene ranking for classification of microarray samples. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3:312–320.

Chapter 5
Tree-Based Methods

Adele Cutler, D. Richard Cutler, and John R. Stevens

Data analysts in many disciplines are increasingly faced with high-dimensional data that would have been unthinkable just a few years ago, and nowhere is this more prevalent than oncology. Technological advances such as microarrays, mass spectrometry, and genome-wide single nucleotide polymorphism (SNP) analysis offer immense potential to investigate the genetic foundations of disease, to explore gene–gene and gene–environment interactions, and ultimately to improve diagnosis and treatment options. However, these technologies give rise to data for which the number of predictor variables (genes, peaks, SNPs) can far exceed the sample size. Better statistical tools are needed to deal with such data, and tree-based methods are among the most effective methods currently available. This chapter is an overview of tree-based methods including boosting and Random Forests.[1]

5.1 Chapter Outline

This chapter contains a brief introduction to classification and regression trees in Section 5.3 followed by an overview of tree-based ensembles including bagging, Random Forests, and boosting, in Section 5.4. Section 5.5 provides some practical advice for using Random Forests, including how to deal with unequal class sizes. Several tree-based ensemble methods are compared on some high-dimensional oncology data in Section 5.6 and the chapter concludes with a summary of recent research in this area.

A. Cutler
Department of Mathematics and Statistics, Utah State University, 3900 Old Main Hill, Logan, UT 84322-3900, USA
email: Adele.Cutler@usu.edu

[1]Random Forests is a trademark of Leo Breiman and Adele Cutler and is licensed exclusively to Salford Systems.

X. Li, R. Xu (eds.), *High-Dimensional Data Analysis in Cancer Research*, Applied Bioinformatics and Biostatistics in Cancer Research, DOI 10.1007/978-0-387-69765-9_5,

5.2 Background

Tree-based methods may be used for classification and regression problems in which a number of predictor variables, e.g., genes, peaks in spectra, SNPs, and a response variable are measured for each subject, e.g., person, tissue sample, organism, in the sample. Regression is used to model and predict a continuous response variable. Classification deals with categorical response variables. That is, each observation is known to come from one of a number of distinct groups or classes, and the goal is to use the predictor variables to classify unlabeled observations. In some situations, the analyst may also want to determine "variable importance," namely, which predictors are associated with the response, how they relate, and perhaps even which predictor variables interact with others in predicting the response.

5.2.1 Microarray Data

In the microarray context, the observations typically represent the microarrays themselves, or the patients from whom they are obtained, and the predictors are the genes, or more specifically the genes' expression levels. For example, the analyst may have microarrays for cancer patients and controls and may want to classify a new person into one of these two groups based on their microarray results. More importantly, the analyst may also want to determine which genes on the microarray are useful in distinguishing between patients and controls, with the idea of developing a more efficient diagnostic tool or giving useful information about the genetic foundation of the disease. In a more complicated situation, the microarray data may be accompanied by environmental predictors and it may be of interest to detect gene–environment interactions that help separate patients from controls. In this chapter, we assume that all microarray data have been appropriately normalized. The tree-based methods we discuss are invariant under monotone transformations of the predictor variables, so the data do not need to be log-transformed, although transformation may be advisable for numerical reasons.

5.2.2 Mass Spectrometry Data

In mass spectrometry analysis, the observations again represent people or tissue samples that have been evaluated using mass spectrometry, yielding a spectrum for each subject. A peak detection algorithm may be used to pre-process the spectra, but there may be a large number of peaks. The response variable might be disease state, in which case the problem is one of classification, or a continuous measure of outcome such as the level of an antigen, which is a regression problem.

5.2.3 Traditional Approaches to Classification and Regression

Traditional statistical methods for regression include linear regression and various nonlinear methods such as splines, wavelets, kernel methods, and generalized additive models, while those for classification include linear discriminant analysis and logistic regression (Hastie et al., 2001, for example). For high-dimensional data, these methods are not directly applicable because the number of predictor variables is too large.

5.2.4 Dimension Reduction

One way to deal with the high dimensionality is to use one or more preprocessing methods, followed by a regression or classification method suitable for low-dimensional data. For example, *t*-tests may be used to determine a small set of predictors that can then be used in a logistic regression. Similarly, variables can be screened for regression by only considering those that are individually correlated with the response. In practice, many of these methods are not properly cross-validated and can lead to overfitting and highly optimistic estimates of generalization error. Moreover, these forms of dimension reduction ignore the multivariate structure of the data and may sacrifice valuable predictive information. A multivariate approach would be to use principal components analysis or a singular value decomposition to reduce the dimensionality of the data before doing regression or classification. One problem with this is that principal components analysis concentrates on finding linear combinations of predictors with large variance, which may not have a great deal to do with the response variable (Cutler and Stevens, 2006).

Although the combination of a dimension-reduction step and a standard statistical regression or classification procedure may suffice for some problems, greater accuracy, and possibly greater insight, may be possible with some of the newer techniques such as the tree-based methods described in this chapter. These methods can be used for either regression or classification and do not require dimension-reduction preprocessing, although such may be desirable for computational feasibility. All the tree-based methods described in the chapter require choice of one or more tuning parameters, which may be chosen using cross-validation as described in Chapter 4 of this volume. Compared to other machine-learning methods such as neural nets, trees need relatively few tuning parameters and tend to be relatively insensitive to the choice of these parameters. In fact, it is not uncommon for tree-based methods to perform well with default values, with no need for additional tuning.

5.3 Classification and Regression Trees

Tree-based methods for classification and regression were introduced from a statistical perspective by Breiman et al. (1984). Regression trees are used when the response variable is continuous, while classification trees are used for a categorical response.

A tree is grown by first considering a "root" node containing all the observations. Observations in this node are sent to one of two descendant nodes, one left and one right, using a "split" on a single predictor variable. For a continuous predictor variable, a split is determined by a single split-point; observations for which the predictor is smaller than the split-point go to the left, the rest go to the right. For a categorical predictor variable, a split sends a subset of categories to the left and the rest to the right. The particular split a tree uses to partition a node into its two descendants is chosen by considering every possible split on every predictor variable. The predictor and split combination giving the "best" value according to some criterion is used to partition the node. Examples of splitting criteria are given in Breiman et al. (1984) and include sums of squared residuals for regression trees and measures of homogeneity, such as the Gini index and entropy, for classification trees. The same procedure is applied in turn to the descendant nodes, sometimes called "recursive partitioning." Usually, the trees are grown until a stopping criterion is met, for example, all nodes contain fewer than some fixed number of cases, then "pruned" back to prevent overfitting (Breiman et al., 1984). Once a tree has been grown and possibly pruned, it will have some nonpartitioned nodes called "terminal nodes." Predicted values are obtained from the observations in a terminal node by averaging the response for regression problems or computing either class membership proportions or the most frequent class for classification problems.

5.3.1 Example: Regression Tree for Prostate Cancer Data

The top panel of Figure 5.1 shows a regression tree for data from the prostate cancer study of Stamey et al. (1989), also studied in Hastie et al. (2001). The response variable is the level of prostate-specific antigen (lpsa). For illustrative purposes, only two predictors are included, namely log cancer volume (lcavol) and log prostate weight (lweight). At each node, cases that satisfy the inequality go to the left, while ones that do not satisfy the inequality go to the right. Each terminal node results in a single predicted value, namely the average value of the response for the observations falling into the node. At the bottom left, Figure 5.1 shows a perspective plot of the piecewise linear regression surface corresponding to the regression tree in the top panel. On the bottom right, Figure 5.1 shows the partitioning of the predictor space. For continuous predictors, the splits are parallel to the coordinate axes and the predictor space is divided into (hyper-) rectangles, each with a single predicted value.

5.3.2 Properties of Trees

Trees are popular for a wide range of problems, in part because trees can capture complex interactions. The rank-based nature of the splits makes trees robust to outliers and insensitive to monotone transformations of the predictor variables.

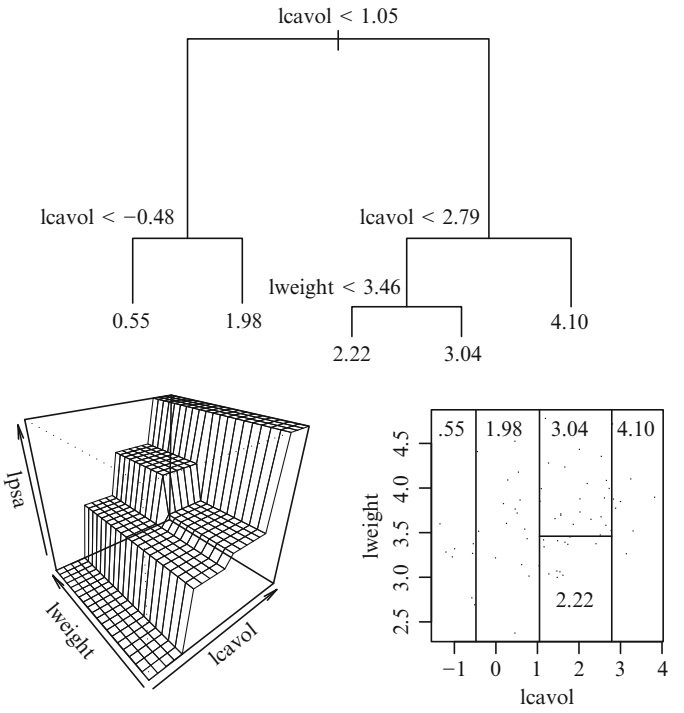


Fig. 5.1 Regression tree for two-dimensional prostate cancer data. The *top panel* shows the tree diagram, the *bottom left* contains a perspective plot of the fitted regression surface, the *bottom right* shows the partitioning of the predictor space.

A summary of the characteristics that make trees popular, even for low-dimensional problems, is (Hastie et al., 2001, Section 10.7) that trees:

- Can capture interactions
- Naturally handle both continuous and categorical predictor variables
- Handle missing values in the predictor variables
- Are robust to outliers in the predictor variables
- Are insensitive to monotone transformations of the predictor variables
- Scale well for large sample sizes
- Deal well with irrelevant predictor variables

Neither support vector machines nor neural networks rate highly on any of the above characteristics (Hastie et al., 2001, Section 10.7). On the downside, regression trees have sharp jumps in the predictions at the edges of the nodes, which may be overcome by Friedman's MARS algorithm (Friedman, 1991). Also, trees:

- Are not good at capturing linear combinations of predictor variables
- Are known to be unstable in the sense that if the data are perturbed slightly, the tree can change a lot
- Are not as accurate as some of the more recently developed methods

Trees enjoy a mixed reception when it comes to interpretability. Tree diagrams are easily understood, but interpretation can be difficult because adjacent or nearby rectangles can appear in quite distant parts of the tree. A less obvious problem occurs when two or more predictor variables are highly correlated within a node. Such variables are called surrogates, and lead to similar splits of the node. However, they make interpretation more difficult because *different* surrogates may be selected for splits at this and descendant nodes. If there are only a few predictor variables, good software can help keep track of surrogates, but in very high-dimensional examples the task becomes much more difficult and it may be impossible to extract a coherent story from the tree diagram.

Perhaps the single largest drawback of trees is that they are not as accurate as more recently developed methods. In particular, more accurate results can be obtained by combining a variety of suitably chosen trees, leading to methods known as tree-based ensembles.

5.4 Tree-Based Ensembles

Tree-based ensembles combine the predictions of many different trees to give an aggregated prediction. To obtain the different trees, some ensemble methods use randomness in the tree-fitting procedure, others fit nonrandom trees to different versions of the data set, and some employ both of these strategies (Dietterich, 2000). Methods also differ in how the predictions are aggregated. In regression, a simple aggregates prediction is the average of the predictions from the individual trees. A simple version for classification is to use the most frequently predicted class, in a procedure known as "voting" the trees.

Ensembles can give improved prediction accuracy over individual trees. An intuitive idea behind the improved accuracy of ensemble classifiers is that if the individual classifiers tend to make prediction errors in different regions of predictor space, then the incorrect predictors may be "outvoted" by the correct ones. So, for example, if 3 trees each have an error rate of 1/3 but the errors are on three disjoint sets of observations, the voted classifier will predict the correct class for every observation. Reality is not so simple because the "hard to classify" cases tend to be the same for all the trees. Moreover, it is possible to construct examples to show that voting bad classifiers can make them even worse (Hastie et al., 2001, Section 8.7.1).

Another intuitive way to see why ensembles may give more accurate predictions is to think of them as a way of "smoothing" individual trees, giving smoother regression predictions and smoother boundaries for classification. A less heuristic rationale in the regression context is provided by the concepts of bias and variance. If an ensemble is comprised of trees which have low bias and high variance, then aggregating their predictions can give an ensemble predictor with lower variance than the individual trees, while maintaining low bias (Hastie et al., 2001, Section 8.7.1).

5.4.1 Bagged Trees

Bagging (Breiman, 1996) stands for "bootstrap aggregating" and denotes ensembles built by fitting predictors to bootstrap samples from the original data. In this context, bootstrapping is done by randomly sampling cases with replacement, until the bootstrap sample has the same number of cases as the original data. Some of the original cases will not appear in the bootstrap sample. Others will appear once, twice, or even more often. Predictions are combined by averaging for regression and voting for classification.

5.4.1.1 Example: Bagged Regression Trees for Prostate Cancer Data

Figure 5.2 shows the regression surface for bagging 100 regression trees for the two-dimensional prostate cancer data example described in Section 5.3. The surface is noticeably smoother than the one in Figure 5.1. Using the same random split of the data as that used by Hastie et al. (2001) gives a training set of size 67 and a test set of size 30. Fitting to the training data and using the test set to estimate error rates gives an error rate of 0.435 for the single tree and 0.432 for the bagged tree. In fact, it is not clear that either method is significantly "better" than ordinary linear regression; the point of this example is simply to illustrate the nature of the regression surface for the two methods.

5.4.1.2 Properties of Bagged Trees

Bagged trees retain the positive characteristics of trees listed in Section 5.3, but they are computationally slower and more difficult to interpret.

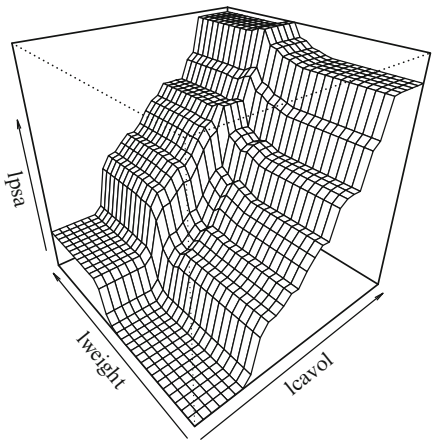


Fig. 5.2 Regression surface for bagging 100 regression trees, two-dimensional prostate cancer data.

Breiman (1996) suggests that bagging can substantially increase the predictive accuracy of an unstable predictor, namely, one for which small changes in the data set can result in large changes in the predictions. Trees are well known to be unstable, and empirical evidence suggests that bagged trees seldom do worse than an individual tree. Whereas, individual trees are frequently less accurate than the more sophisticated tree-based ensembles presented later in this chapter and for problems with thousands of predictors, such as those arising from microarrays, SNPs, and mass spectrometry, bagging can be prohibitively slow.

5.4.2 Random Forests

Random Forests (RF) (Breiman, 2001) are tree-based ensembles that use bootstrap samples and randomness in the tree-building procedure. In Random Forests, trees are fit to bootstrap samples using a random sample of m predictors on which to split each node. The value of m is a tuning parameter of the method; m is chosen to be much smaller than the total number of predictors. The m predictors are chosen independently for each node, and the best split for the selected predictors is used to split the node, where "best" is determined as for a single tree (Breiman et al., 1984). The trees are grown large, and not pruned; for classification, the trees are grown until each terminal node contains members of only one class, while for regression they are grown until each terminal node contains a small number of cases.

5.4.2.1 Properties of Random Forests

Random Forests inherit the positive characteristics of trees listed in Section 5.3. Although bagged trees are technically a special case of Random Forests in which m is chosen equal to the number of predictor variables, the small values of m typically used by Random Forests can lead to strikingly different properties; Random Forests are considerably faster than bagged trees and frequently more accurate.

The key to the accuracy of Random Forest predictions is low bias and low correlation among trees (Breiman, 2001). Low bias is achieved by growing large trees; low correlation results from making the trees as dissimilar as possible, while still maintaining low bias. In bagging, the trees differ only because they are fit to different bootstrap samples. Intuitively, this gives trees that are too similar to give low correlation because they tend to make mistakes in similar places. In Random Forests, random sampling of a small number (m) of predictors at each node forces the trees to be quite different, reducing correlation and improving predictive power.

Other properties that make Random Forests attractive for high-dimensional oncology data analysis are that they:

- Have an inbuilt method of assessing generalization error
- Do not require tuning of many parameters
- Provide measures of variable importance
- Can be used for very unbalanced data sets

- Can handle high-dimensional data without formal variable selection
- Provide proximities that can be used for clustering data

Each of these points is addressed in more detail in the remainder of this section.

5.4.2.2 Generalization Error

When a bootstrap sample is taken from the data, some observations do not make it into the bootstrap sample. These are called "out-of-bag data," and can be used to give an internal estimate of generalization error. To get this out-of-bag error rate, each tree is used to predict the response variable for the out-of-bag data and these predictions are saved. At any point, generalization error can be estimated for each observation by averaging the error rate of the predictions from the trees for which it was out-of-bag. For classification, class-wise error rates are computed by averaging over observations from the same class. An overall error rate is computed by averaging over all the observations.

5.4.2.3 Tuning Parameters

There are two obvious tuning parameters for Random Forests, namely the number of trees and m, the number of variables to sample at each node. However, in practice, Random Forests are not particularly sensitive to the choice of either of these parameters and the default choices of 500 trees and $m = \sqrt{p}$, where p is the total number of predictors, work well for many classification problems. In fact, Breiman (2001) shows that adding more trees to a Random Forest does not lead to overfitting, so the only real concern with the number of trees is that it should be large enough, and this can be checked using the out-of-bag error rate. The value of m may be chosen by fitting the first few trees in the forest and selecting m using the out-of-bag data described in the previous paragraph.

There is some risk that using the out-of-bag data to pick the number of trees and m may compromise the estimate of generalization error (see more discussion on this topic in Chapter 4 of this volume). However, Random Forests are not very sensitive to these tuning parameters, so fine-tuning is not required and the effects should be relatively small, as demonstrated by Diaz-Uriarte and Alvarez de Andres (2006).

Other tuning parameters may be necessary for specialized problems. In regression, it is necessary to control the depth of the trees or the minimum number of cases in the terminal nodes. Again, out-of-bag data can be used with care, understanding that over-tuning will lead to bias in the out-of-bag error rate estimate.

5.4.2.4 Variable Importance

Random Forests use an unusual but intuitive measure of variable importance. To measure the importance of variable k for a single tree in the forest, the out-of-bag observations are passed down the tree and the predicted values are computed. Next,

the values of variable k are randomly permuted, keeping all the other predictor variables fixed. These modified out-of-bag data are passed down the tree and the predicted values are computed. This process is repeated for all the trees, giving two sets of out-of-bag predictions for each observation: one set obtained from real data, the other set from variable-k-permuted data. The difference between the error rate from the modified data and the error rate for the real data gives a measure of variable importance for the observation. For classification, class-wise variable importance is computed by averaging over observations from the same class. Overall variable importance is computed by averaging over all the observations.

Intuitively, the permutation-based importance of variable k is an estimate of how much the prediction error on a test set would increase if the value of variable k were randomly permuted in the test set. In this sense, it is similar to the coefficient-based measures of importance used in methods such as linear regression or logistic regression – they measure how much the prediction would change if the value of the predictor increased by one unit. Quite a different measure is obtained, for both Random Forests and classical methods, if variable k is removed and the model is refit.

If an important predictor variable is correlated with other predictor variables, Random Forests sometimes splits on one and sometimes on another, due to the random choice of predictors at each node. Therefore, Random Forests tends to identify all of the correlated predictors as important if any one of them is important.

One attractive feature of all tree-based methods is their ability to capture complex interactions between predictors. If Random Forests captures such an interaction, the variables involved are likely to show up as "important" because randomly permuting one of them destroys the predictive power of the interaction.

5.4.2.5 Unequal Class Sizes

Unbalanced data sets, where some classes are much smaller than others, present a challenge to many classifiers. A naive classifier will work on getting the large classes right while allowing a high error rate on the small classes. Random Forests has an effective method for weighting the classes to give balanced results in unbalanced data (`http://www.math.usu.edu/~adele/forests`). One reason to do this is that the important predictor variables may be different when the method is forced to pay greater attention to the small class. Even in the balanced case, the weights can be adjusted to give lower error rates to decisions that have a high misclassification cost. For example, it is often more serious to incorrectly conclude that someone is healthy than it would be to incorrectly conclude that someone is ill.

5.4.2.6 Proximities

One of the difficult aspects of high-dimensional data analysis is that it is not obvious how to get a good "feel" for the data. Are there interesting patterns or structures, such as sub-groups within the known classes? Are there outliers? In a multiclass situation, are some of the groups separated while others overlap? Random Forests

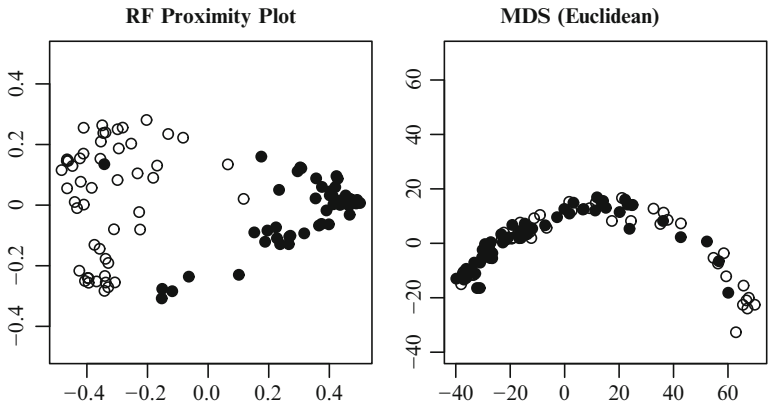


Fig. 5.3 MDS plot from the Random Forests proximities (*left*) and from Euclidean distance (*right*). *Solid circles* represent cancer cases, *open circles* represent controls.

provide a way to look at the data to give some insight into these questions. This is done by computing a measure of proximity between each pair of observations. The proximity between two observations is the proportion of the time that they end up in the same terminal node, where the proportion is taken over the trees in the forest. If two observations are always in the same terminal node, their proximity will be 1. If they are never in the same terminal node, their proximity will be 0. A distance matrix is derived from the proximities, and classical multidimensional scaling is used to obtain a two- or three-dimensional plot. Each point on the plot represents one of the observations and the distances between the points reproduce, as closely as possible, the proximity-based distances. Such a plot can be used to pick out subgroups of cases that almost always stay together in the trees, or outliers that are almost always alone in a terminal node.

A natural question at this point is whether it would be just as good to use multidimensional scaling on a conventional distance, such as Euclidean distance or one of the other distances commonly used in cluster analysis. This can certainly be done, but one of the difficulties is that a conventional distance can be dominated by noisy and uninformative predictors that may drown out the effects of the predictors that are important.

Figure 5.3 illustrates a proximity plot and the corresponding Euclidean distance MDS plot for the prostate cancer microarray data described in Section 5.5. The proximity plot reveals much more structure than the plot based on Euclidean distances, including an outlier that could be of interest to the investigators.

5.4.2.7 Missing Value Imputation

Random Forests imputes missing values using the proximities described above. The procedure is iterative: an initial forest is built using median imputation, proximities are calculated, and new imputations are obtained by a proximity-weighted average

for a continuous predictor or a proximity-weighted vote for a categorical predictor. A new forest is built, giving new proximities and imputations. Usually five or six iterations are sufficient to give stable imputations. Although no formal analysis has been done, the fact that the method uses proximity-based nearest neighbors suggests that it will be valid if values of the predictors are missing at random.

5.4.3 Boosted Trees

The idea behind boosted tree classifiers is to form an ensemble by fitting trees to weighted versions of the data. Initially, all observations have the same weight. As the ensemble grows, the weights are adjusted based on knowledge of the problem. The weights of frequently misclassified observations are increased, while those of seldom-misclassified observations are decreased. Heuristically, the trees are encouraged to tailor themselves to the difficult cases. The various algorithms differ in their choice of their:

- Individual trees
- Method of changing the weights
- Procedure for combining to give the final prediction

The trees used in boosting are typically small, sometimes as small as "stumps," which have only one split. Typically, the trees are combined by weighted voting or averaging, with highly accurate trees getting more weight than do less accurate ones. More sophisticated boosting methods use shrinkage methods such as the lasso (Tibshirani, 1996) to select this second set of weights.

In this section, two boosting methods are presented, namely AdaBoost, for historical reasons, and Gradient boosting. Only the classification context is discussed in this chapter. For more details on boosting for regression, see Chapter 10 of Hastie et al. (2001).

5.4.3.1 Adaboost

Boosting originated with the AdaBoost classifier of Freund and Schapire (1996). Consider a 2-class problem with data $(x_i, y_i), i = 1, \ldots, n$, where x_i denotes the predictor variables and y_i denotes the class variable for the ith subject, with $y_i = -1$ or $y_i = 1$ to represent the class. AdaBoost fits J classifiers as follows:

Adaboost Algorithm

1. Initialize the weights $w_i = 1/n$, for $i = 1, \ldots, n$.
2. For $j = 1, \ldots, J$:
   a) Fit a classifier $C_j(x)$ using weights w_i for $i = 1, \ldots, n$.
   b) Find predicted values $\hat{y}_i = C_j(x_i)$, for $i = 1, \ldots, n$.

c) Compute the weighted error rate: $e_j = \sum_{i=1}^{n} w_i I(\hat{y}_i \neq y_i) / \sum_{i=1}^{n} w_i$.
d) Let $\alpha_j = \log((1-e_j)/e_j)$.
e) If case i was misclassified in step 2b), multiply its weight by $\exp(\alpha_j)$.

3. AdaBoost predicts class -1 if $\sum_{j=1}^{J} \alpha_j C_j(x) < 0$ and class 1 otherwise.

Note that α_j is small if the weighted error rate at step j is large, so that α_j is larger for more accurate classifiers in step 3. In step 3, the values $\alpha_1, \ldots, \alpha_J$ could be normalized without changing the predictions, so the α_j are referred to as "weights" even though they are not normalized.

Adaboost uses very simple classifiers in step 2(a). A popular choice is a tree with just a single split, leading to the name "boosted stumps." At first it was believed that AdaBoost could not overfit, even if the number of trees, J, was increased indefinitely. In fact, this claim turns out to be incorrect, and there are examples for which generalization error decreases for a while but ultimately starts to increase as J becomes very large. Therefore, J should be chosen using some form of cross-validated estimate of generalization error.

5.4.3.2 Gradient Boosting Machines

Friedman et al. (2000) show that AdaBoost fits an additive model by stage-wise optimization of an objective function. This work was extended by Friedman (2001) resulting in methods called "Gradient Boosting Machines."

Gradient boosting seeks to find $f^{\star}(x)$ to minimize $E(L(y, f(x)))$ with respect to f for some loss function L, where the expectation is taken over the joint distribution of x and y. Typical loss functions are least squares for regression and the negative log-likelihood for the multinomial for classification.

Gradient boosting approximates $f^{\star}(x)$ by an additive expression of the form

$$\hat{f}(x) = \sum_{j=0}^{J} \rho_j T(x, a_j)$$

where $T(x, a_j)$ is a simple function of x with parameters a_j. The parameters ρ_j and $a_j, j = 1, \ldots, J$ are estimated using the following procedure.

Gradient Boosting Machines

1. Initialize $f_0(x) = \arg\min_{\beta} \sum_{i=1}^{n} L(y_i, \beta)$.
2. For $j = 1, \ldots, J$:
   a) $\tilde{y}_i = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{j-1}(x)}$ for $i = 1, \ldots, n$.
   b) Fit $T(x, a)$ to $(x_i, \tilde{y}_i), i = 1, \ldots, n$ and denote the estimated parameters a_j.
   c) Let $\rho_j = \arg\min_{\rho} \sum_{i=1}^{n} L(y_i, f_{j-1}(x_i)) + \rho T(x_i, a_j)$.
   d) Let $f_j(x) = f_{j-1}(x) + \rho_j T(x, a_j)$.

Step 2(a) requires the loss function L to be differentiable and computes "pseudo-residuals" in the direction of the negative gradient of the loss at the current $\hat{f}(x)$. In step 2(b) the fitting is typically done by least squares. A further modification known as "stochastic gradient boosting" (Friedman, 2002) fits $T(x,a)$ to a random subsample from the data in step 2(b). For trees, step 2(b) involves fitting a tree, and in step 2(c) a value of ρ_j is found separately for each node of the tree $T(x,a_j)$, with corresponding modifications to step 2(d). Finally, a shrinkage parameter λ can be used in step 2(d) to give $f_j(x) = f_{j-1}(x) + \lambda\rho_j T(x,a_j)$. The value of λ is a tuning parameter of the method. In general, smaller values of λ will require more trees, that is, larger J, to give comparable accuracy.

Properties of Gradient Boosting Machines

Like Random Forests, tree-based Gradient Boosting Machines (GBM) share many of the positive characteristics of trees listed in Section 5.3 and they are extremely powerful predictors. The jury is out on whether they are generally more powerful than Random Forests; sometimes they are, sometimes not. They *do* appear to require more tuning than Random Forests and the effects of this are illustrated in Section 5.5. Like Random Forests, Gradient Boosting Machines provide measures of variable importance and cross-validated estimates of generalization error. They also provide partial dependence plots and an ANOVA decomposition to aid interpretation.

5.5 Example: Prostate Cancer Microarrays

Random Forests and Gradient Boosting Machines are illustrated using a data set on prostate cancer (Singh et al., 2002). These data have 6,033 gene expression values for 102 arrays, namely 50 normal samples and 52 tumor samples. Of interest with these data was predicting disease class, normal or tumor, based on the gene expression values. The data were preprocessed by Dettling (2004). All computations were performed in R (R Development Core Team, 2007) using `randomForest` (Liaw and Wiener, 2002), and `gbm` (Ridgeway, 2007).

Initially, RF and GBM were run with randomly selected subsets of predictors to get an idea of CPU times in terms of the number of predictors. All parameters except the number of trees were set at default values and the `gbm.fit` function was used instead of `gbm` itself to avoid the formula interface, which can slow down the methods considerably. CPU times, measured by the average of 5 runs on a 2GHz machine, are given in Table 5.1. The actual values are machine dependent, but for both methods, CPU increases proportionally to the number of predictors. Times for GBM are slightly higher than for RF, and both methods increase proportionally to the number of predictors.

To evaluate accuracy, tenfold cross-validation was performed for both methods, repeated ten times, and averaged. Default values were chosen for all parameters. The average error rates were 9.4% and 14.23% for RF and GBM, respectively. When the

Table 5.1 CPU time (seconds) for RF and GBM.

Method	Number of trees	Number of predictors					
		1,000	2,000	3,000	4,000	5,000	6,000
RF	100	1.1	2.3	3.5	4.6	5.7	6.6
GBM	100	4.2	8.3	12.6	16.9	21.1	24.9
RF	500	5.5	11.1	17.0	22.2	28.2	32.2
GBM	500	8.2	16.2	24.2	32.5	40.4	47.8
RF	1,000	10.7	22.2	34.1	44.4	56.2	65.2
GBM	1,000	13.3	26.1	38.9	52.0	64.8	76.6

Table 5.2 Sensitivity to tuning parameter choice. Table gives number of errors for prostate cancer data, computed by tenfold cross-validation with the specified choice of tuning parameters and everything else set at default values.

		Number of Trees				
		100	250	500	750	1,000
Random Forests	m					
	25	13	13	10	10	12
	50	12	9	8	9	7
	75	9	10	**8**	9	9
	100	7	9	7	7	7
	150	8	8	7	7	7
	200	9	7	7	7	7
	300	7	7	8	7	7
Gradient Boosting Machines	Shrinkage					
	0.0001	50	43	25	14	12
	0.0005	27	13	11	11	10
	0.0010	**13**	11	10	9	9
	0.0015	14	11	10	10	9
	0.0020	11	10	11	9	8
	0.0025	9	11	9	7	7
	0.005	10	9	7	6	6
	0.01	11	7	6	6	6
	0.05	7	6	6	6	6
	0.1	5	7	6	6	6
	0.5	9	7	7	12	15
	1.0	19	9	18	18	20

number of trees for GBM was increased to 1,000, its average error rate dropped to 10.2% and when the number of trees was increased to 1,500, the average error rate was 9.6%.

To get an idea of how sensitive the methods are to tuning parameter choice, tenfold cross-validation was used to estimate the generalization error for different choices of the number of trees and of m for RF, and shrinkage for GBM. The results are presented in Table 5.2. Default values are shown in bold with the default value of $m = \sqrt{6033}$ approximated to 75. For RF, results are similar for all values of m greater than the default, and for all numbers of trees. There is some suggestion that

100 trees is not quite enough, and that $m = 25$ is also not enough. Nonetheless, the ratio of largest error rate to smallest error rate is less than 2.

Examination of the GBM algorithm suggests that the number of trees affects the amount of shrinkage required. For a large number of trees, smaller values of the shrinkage parameter may give good results, while for fewer trees, a larger value of the shrinkage parameter is necessary. Table 5.2 confirms these ideas. For a small number of trees, too little or too much shrinkage gives poor results. Similarly, for a fixed amount of shrinkage there is some evidence that fitting either too few or too many trees gives poor results. There is a large middle ground, where GBM give results that are comparable to, or a little better than, RF. Nonetheless, finding that middle ground is time consuming for large data sets, since the number of trees required for good performance tends to be greater for GBM than for RF.

Variable importance was computed for RF, but could not be computed for GBM due to problems with the R formula interface for large numbers of predictors. To evaluate variable importance for the two methods, the data set was first reduced to the 500 most important predictors according to the RF permutation variable importance. RF and GBM were used to select the 50 most important predictors from this reduced data set, with default values for RF and 500 trees with shrinkage = 0.05 for GBM. Error rates for the methods using the 50 selected predictors were estimated using ten repetitions of tenfold cross-validation. For RF the average error rate was 5.8% and for GBM it was 4.6%. Of course, these estimates are biased due to the fact that the predictors were chosen using the data at hand, but they suggest that if the number of predictors is sufficiently small, GBM may do slightly better than RF.

5.6 Software

Commercial software for classification and regression trees, Random Forests, and gradient boosting is available from http://www.salford-systems.com. R packages include `rpart` (Therneau and Atkinson., 2007), `randomForest` (Liaw and Wiener, 2002), and `gbm` (Ridgeway, 2007), and these, along with R (R Development Core Team, 2007), are available from the CRAN Web site `http://www.cran.r-project.org`. Open source FORTRAN software for Random Forests is available from `http://www.math.usu.edu/~adele/forests`. This is the only available version that uses class weighting to deal with unequal class sizes.

5.7 Recent Research and Oncology Applications

Wu et al. (2003) compared a variety of classification methods for mass spectrometry data of ovarian cancer samples. The paper provides an overview of linear and quadratic discriminant analysis, *k*-nearest neighbors, bagging, boosting, and

Random Forest algorithms. These methods were applied to the mass spectrometry data consisting of spectra on each of 47 ovarian cancer patients and 44 normal patients, and each method was used to predict cancer status. The Random Forest classifier outperformed the other methods, providing the lowest prediction error rate.

Shi et al. (2005) demonstrated the ability of Random Forests to successfully differentiate renal cell carcinoma sample classes (clear vs. nonclear tumors in 366 patients) based on tissue protein abundance microarray data. The Random Forests method was also used to discover "clinically well-defined" subgroups, including low- and high-grade clear cell patients, within these sample classes based on relatively few protein-level markers.

In an early study on microarray data Dudoit et al. (2002) compared Random Forests to other classifiers for three cancer data sets. The work was extended by Lee et al. (2005), who compared 21 methods on 7 data sets, of which 6 were cancer data. Although Lee et al. (2005) did no formal testing, their results for Random Forests were almost always better than the other tree-based methods they considered, including two versions of boosted trees, and similar to those from SVMs for all of the cancer data sets. As part of their comparison, Lee et al. (2005) used three forms of gene selection to reduce the number of genes to 50 before using the classifiers. Random Forests does not require this sort of variable selection preprocessing, and may have given even better results if it had been used without variable selection.

Diaz-Uriarte and Alvarez de Andres (2006) presented a user-friendly web interface `http://genesrf.bioinfo.cnio.es` to the RF algorithm for classification based on gene expression microarray data. Their application supported the competitive results of Random Forests compared to other methods such as diagonal linear discriminant analysis, k-nearest neighbor, and support vector machines. They conclude "random forest and gene selection using random forest should probably become part of the 'standard tool-box' of methods for class prediction and gene selection with microarray data." Application of Random Forests to breast cancer was presented by some of the same authors in Alvarez et al. (2005).

Variations of boosting to classify tumors using gene expression data were considered by Dettling and Buhlmann (2003). In Dettling (2004), Random Forests compared favorably to two boosting methods for six cancer microarray data sets.

Bureau et al. (2005) used Random Forests' variable importance to identify SNPs predictive of phenotypes, and Heidema et al. (2006) found that Random Forests were able to handle a large number of predictors and that they were useful in variable reduction.

In addition to the classification of phenotypes or treatment conditions, Pang et al. (2006) employed Random Forests with gene expression microarray data to identify important pathways relevant to the phenotype differences. A pathway is a set of genes that together serve a common biological function. One benefit to focusing on pathway analysis is that the final set of important variables, namely the selected genes, are more easily interpretable because of their shared membership in a specific pathway. The authors compared Random Forests with linear discriminant analysis, neural network, bagging, support vector machines, k-nearest neighbor, and naive Bayes. Random Forests performed favorably, with the lowest or second-lowest error

rates, when applied to a variety of data sets, including three cancer data sets – one involving breast cancer and two involving lung cancer.

After identifying recurrent expression patterns across many human gene expression microarray data sets, Huang et al. (2007) identified thousands of potential functional modules, or groups of functionally related genes. They then used Random Forests to make functional predictions for 779 and 116 unknown genes, with a validation accuracy of 70%.

Munro et al. (2006) used Random Forests to predict the presence of transitional cell carcinoma (TCC) in proteomic (SELDI) profiles, with good reproducibility in a validation set collected 6 months after the initial analysis.

References

Alvarez, S., Diaz-Uriarte, R., Osorio, A., Barroso, A., Melchor, L., Paz, M. F., Honrado, E., Rodriguez, R., Urioste, M., Valle, L., Diez, O., Cigudosa, J. C., Dopazo, J., Esteller, M., and Benitez, J. (2005). A predictor based on the somatic genomic changes of the brca1/brca2 breast cancer tumors identifies the non-brca1/brca2 tumors with brca1 promoter hypermethylation. *Clinical Cancer Research*, 11(3):1146–1153.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 26(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth, Boca Raton, FL.

Bureau, A., Dupuis, J., Falls, K., Lunetta, K. L., Hayward, B., Keith, T. P., and Eerdewegh, P. V. (2005). Identifying snps predictive of phenotype using random forests. *Genetic Epidemiology*, 28(2):171–182.

Cutler, A. and Stevens, J. R. (2006). Random forests for microarrays. In Kimmel, A. and Oliver, B., editors, *DNA Microarrays, Part B: Databases and Statistics, Volume 411 (Methods in Enzymology)*. Academic Press, San Diego, CA.

Dettling, M. (2004). Bagboosting for tumor classification with gene expression data. *Bioinformatics*, 20(18):3583–3593.

Dettling, M. and Buhlmann, P. (2003). Boosting for tumor classification with gene expression data. *Bioinformatics*, 19(9):1061–1069.

Diaz-Uriarte, R. and Alvarez de Andres, S. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157.

Dudoit, S., Fridlyand, J., and Speed, T. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pp. 148–156.

Friedman, J. (1991). Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19(1):1–141.

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232.

Friedman, J. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion). *Annals of Statistics*, 28(2):337–407.

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

Heidema, A. G., Boer, J. M., Nagelkerke, N., Mariman, E. C., van der A, D. L., and Feskens, E. J. (2006). The challenge for genetic epidemiologists: how to analyze large numbers of snps in relation to complex diseases. *BMC Genetics*, 7(23).

Huang, Y., Li, H., Hu, H., Yan, X., Waterman, M., Huang, H., and Zhou, X.J. (2007). Systematic discovery of functional modules and context-specific functional annotation of human genome. *Bioinformatics*, 23:222–229.

Lee, J. W., Lee, J. B., Park, M., and Song, S. H. (2005). An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis*, 48(4):869–885.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Munro, N. P., Cairns, D. A., Clarke, P., Rogers, M., Stanley, A. J., Barrett, J. H., Harnden, P., Thompson, D., Eardley, I., Banks, R. E., and Knowles, M. A. (2006). Urinary biomarker profiling in transitional cell carcinoma. *International Journal of Cancer*, 119(11):2642–2650.

Pang, H., Lin, A., Holford, M., Enerson, B., Lu, B., Lawton, M., Floyd, E., and Zhao, H. (2006). Pathway analysis using random forests classification and regression. *Bioinformatics*, 22(16):2028–2036.

R Development Core Team. (2007). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Ridgeway, G. (2007). *gbm: Generalized Boosted Regression Models*. R package version 1.6-3.

Shi, T., Seligson, D., Belldegrun, A., Palotie, A., and Horvath, S. (2005). Tumor classification by tissue microarray profiling: random forest clustering applied to renal cell carcinoma. *Modern Pathology*, 18:547–557.

Singh, D., Febbo, P., Ross, K., Jackson, D., Manola, J., Ladd, C., Tamayo, P., Renshaw, A., D'Amico, A., Richie, J., Lander, E., Loda, M., Kantoff, P., Golub, T., and Sellers, W. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209.

Stamey, T., Kabalin, J., McNeal, J., Johnstone, I., Freiha, F., Redwine, E., and Yang, N. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. ii. radical prostatectomy treated patients. *Journal of Urology*, 16:1076–1083.

Therneau, T. M. and Atkinson., B. (2007). *rpart: Recursive Partitioning*. R port by Brian Ripley. R package version 3.1-36.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58:267–288.

Wu, B., Abbot, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., and Zhao, H. (2003). Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data. *Bioinformatics*, 19(13):1636–1643.

Chapter 6
Support Vector Machine Classification for High-Dimensional Microarray Data Analysis, With Applications in Cancer Research

Hao Helen Zhang

In a classification problem, we are given a set of labeled samples from two or more distinctive classes, e.g., different cancer types. The task is to learn a rule that can categorize these samples based on their attributes such as diagnostic indicators or gene expression profiles. A good classification rule should generalize well, i.e., it ought to be able to accurately classify new unlabeled samples taken from the same target population.

High-dimensional expression microarray data have been rapidly accumulated in the field of cancer research in recent years. They impose new challenges to conventional classification methods, mainly due to their unique "high dimension low sample size" data structure. That is, these data in general contain large numbers of variables, typically tens of thousands of genes, but much smaller numbers of assayed tumor samples which are often less than hundreds. Effective and reliable classification methods are hence demanded to discover hidden patterns in these high-dimensional data.

In the past decade, the support vector machine (SVM) has become a powerful classification tool used in various research fields, due to its superior prediction accuracy, nonlinear classification feature, and ability to handle high-dimensional data. In this chapter, we first review the basic principles of the SVM and its variant formulations, then demonstrate how these methods overcome the *curse of dimensionality* and thus, become suitable to accurately identify differentially expressed gene signatures and build reliable classification models in cancer research.

H.H. Zhang
Department of Statistics, North Carolina State University, 2501 Founders Drive, Raleigh, NC 27613, USA
email: hzhang@stat.ncsu.edu

X. Li, R. Xu (eds.), *High-Dimensional Data Analysis in Cancer Research*, Applied Bioinformatics and Biostatistics in Cancer Research, DOI 10.1007/978-0-387-69765-9_6,

6.1 Classification Problems: A Statistical Point of View

6.1.1 Binary Classification Problems

We start with the simple case of two-class or binary classification. In general, we are given a training set consisting of n labeled data points

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}, \quad \mathbf{x}_i \in \mathcal{R}^d, \ y_i \in \{-1, +1\},$$

from which the classification rule is induced. Here $\mathbf{x}_i$ is a d-vector of input variables or predictors, and the label y_i indicates which class the ith data point belongs to. Our task is to construct a classifier rule ϕ which separates the positive class from the negative class. Mathematically, the classifier ϕ is a mapping

$$\phi : \mathcal{R}^d \longrightarrow \{-1, +1\},$$

which assigns a class label to a sample based on its input vector. In practice, we generally train a real-valued function $f : \mathcal{R}^d \longrightarrow \mathcal{R}$ from the training set, and then construct the associated classification rule as $\phi(\mathbf{x}) = \text{sign}[f(\mathbf{x})]$.

Any binary classifier may make one of two possible types of incorrect decisions: classifying a sample from -1 class to $+1$ class, or vice versa. They are, respectively, the *false positive* decision and the *false negative* decision. Each mistaken decision is typically associated with a cost or penalty C. In particular, let $C(y_1, y_2)$ be the cost paid for classifying a sample from class y_1 to class y_2. In the binary case, we have $C(-1, +1)$ for the false positive cost, $C(+1, -1)$ for the false negative cost, and $C(+1, +1) = C(-1, -1) = 0$ because no cost is paid for correct classifications. We normally do not require that $C(+1, -1) = C(-1, +1)$, since one type of misclassification may be more costly than the other. The generalization performance of any classifier ϕ is measured by its overall cost on the target population, which will be described more rigorously in the following statistical framework.

6.1.2 Bayes Rule for Binary Classification

From the statistical perspective, the training samples $\{\mathbf{x}_i, y_i\}, i = 1, \cdots, n$ can be regarded as n independent realizations from a unknown joint distribution $P(\mathbf{X}, Y)$. Define

$$p(\mathbf{x}) = \Pr(Y = +1 | \mathbf{X} = \mathbf{x}),$$

which is the conditional probability of a random sample $(\mathbf{X}, Y)$ from the target population belonging to $+1$ class given $\mathbf{X} = \mathbf{x}$.

For any classifier ϕ, its loss at the data pair $(\mathbf{x}, y)$ is $C(y, \phi(\mathbf{x}))$. The generalization performance of ϕ is typically measured by its risk function, or the expected (average) cost over the target population,

$$R(\phi) = E\left[C(Y, \phi(\mathbf{X})\right].$$

Here the expectation is taken with respect to the joint distribution $P(\mathbf{X}, Y)$. General theoretical and empirical methods for risk estimation are reviewed in Chapter 4. The *Bayes* classification rule, ϕ_{B}, is the optimal rule which minimizes the risk function $R(\phi)$, i.e., $\phi_{\mathrm{B}}(\mathbf{x}) = \mathrm{argmin}_{\phi} R(\phi)$. For a binary classification associated with the cost C, it is easy to show that the Bayes rule is given by

$$\phi_{\mathrm{B}}(\mathbf{x}) = \mathrm{sign}\left[p(\mathbf{x}) - \frac{C(-1,+1)}{C(-1,+1)+C(+1,-1)}\right]. \tag{6.1}$$

In practice, a simple choice of C is the $0-1$ cost: $C(y, \phi(\mathbf{x})) = 1$ if $y \neq \phi(\mathbf{x})$; $= 0$ otherwise. Here C can also be expressed as a function of the quantity $y\phi(\mathbf{x})$, i.e.,

$$C\left(y, \phi(\mathbf{x})\right) = \left[-y\phi(\mathbf{x})\right]_*, \quad \text{where } [\tau]_* = 1 \text{ if } \tau > 0; \ = 0 \text{ otherwise.} \tag{6.2}$$

In this special case, the Bayes rule minimizes the expected misclassification rate (EMR) $E\left[-Y\phi(\mathbf{X})\right]_*$ and the resulting optimal rule is

$$\phi_{\mathrm{B}}(\mathbf{x}) = \mathrm{sign}\left[p(\mathbf{x}) - \frac{1}{2}\right]. \tag{6.3}$$

Note (6.3) is true as long as equal costs $C(-1,+1) = C(+1,-1)$ are used. Hand (1997) and Lin et al. (2002) give more results on the Bayes rule in other nonstandard classification settings.

We would like to point out that the Bayes rule is the golden rule for a particular problem, which in general is not available without the knowledge of $p(\mathbf{x})$. Therefore, most classification methods attempt to approximate the Bayes rule, by either estimating $p(\mathbf{x})$ or directly estimating $\mathrm{sign}\left[p(\mathbf{x}) - \frac{1}{2}\right]$, based on the training data. For example, the logistic regression belongs to the former type, and the SVM algorithm belongs to the latter type.

6.1.3 Multiclass Classification Problems

In multiclass classification problems, the training set consists of n samples from k distinctive classes ($k \geq 3$),

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}, \quad \mathbf{x}_i \in \mathcal{R}^d, \ y_i \in \{1, \cdots, k\}.$$

The task is to learn a classification rule $\phi(\mathbf{x}) : \mathcal{R}^d \to \{1, \cdots, k\}$ from the training set. In this case, the misclassification cost C can be represented as a $k \times k$ matrix, with the entry $C(l,m)$ representing the cost of classifying a sample from class l to class m for $l, m = 1, \ldots, k$. We have $C(l,l) = 0$ for all $l = 1, \ldots, k$ because correct decisions are not penalized.

To achieve a multiclass classification, each classifier $\phi(\mathbf{x})$ is inherently associated with a set of k functions

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})),$$

where $f_l(\mathbf{x}) : \mathcal{R}^d \to \mathcal{R}$ indicates the strength of evidence that $\mathbf{x}$ belongs to class l. The decision rule is defined as

$$\phi(\mathbf{x}) = \arg \max_{l=1,\ldots,k} f_l(\mathbf{x}). \tag{6.4}$$

6.1.4 Bayes Rule for Multiclass Classification

Assume that the samples in the training set are independently drawn from a joint probability distribution $P(\mathbf{x}, y)$. Let

$$p_l(\mathbf{x}) = \Pr(Y = l | \mathbf{X} = \mathbf{x}), \quad \text{for } l = 1, \ldots, k,$$

where $p_l(\mathbf{x})$ is the conditional probability of a random sample $(\mathbf{X}, Y)$ belonging to class l given $\mathbf{X} = \mathbf{x}$. For a classifier ϕ, its loss at the data point $(\mathbf{x}, y)$ is $C(y, \phi(\mathbf{x}))$, and its generalization performance is measured by the expected (average) misclassification cost over the target population:

$$E\left[C(Y, \phi(\mathbf{X}))\right] = E_{\mathbf{X}}\left[\sum_{l=1}^{k} C(l, \phi(\mathbf{X})) p_l(\mathbf{X})\right]. \tag{6.5}$$

The multiclass Bayes rule which minimizes (6.5) is then given by

$$\phi_{\mathrm{B}}(\mathbf{x}) = \mathrm{argmin}_{m=1,\ldots,k}\left[\sum_{l=1}^{k} C(l, m) p_l(\mathbf{x})\right]. \tag{6.6}$$

In the simple case of (equal) $0-1$ cost, $C(l, m) = I(l \neq m)$, where I is the indicator function. The cost matrix is a $k \times k$ matrix filled with 1's except the entries on the main diagonal which are zeros. The Bayes rule, which minimizes the generalization error $E[I(Y \neq \phi(\mathbf{X}))] = \Pr(Y \neq \phi(\mathbf{X}))$, is given by

$$\phi_{\mathrm{B}}(\mathbf{x}) = \mathrm{argmax}_{l=1,\ldots,k} p_l(\mathbf{x}). \tag{6.7}$$

Therefore the Bayes rule assigns $\mathbf{x}$ to its most likely class.

Similar to the binary case, the Bayes rule is available only if all the underlying conditional distribution $p_l(\mathbf{x})$'s are known, which however is not true in practice. Many statistical methods have been proposed to approximate the Bayes rule based on the training data. Classical statistical methods are designed to estimate $p_l(\mathbf{x})$'s and then assign $\mathbf{x}$ to the class corresponding to the highest probability. One such example is the multicategory logit models (Agresti, 2002). An alternative class of

methods are the large-margin methods such as the multiclass SVMs, which directly target on the Bayes rule (6.6) and (6.7). The second class of methods tend to deliver better performance in practice, since estimating $\phi_B(\mathbf{x})$ is essentially estimating the relative rank of $p_l(\mathbf{x})$'s, which in general is an easier task than estimating $p_l(\mathbf{x})$'s themselves.

6.2 Support Vector Machine for Two-Class Classification

The support vector machine (SVM), proposed by Boser et al. (1992), Cortes and Vapnik (1995), and Vapnik (1998), is one type of large margin classifiers. It was originally motivated by maximizing the geometric margin of a separating hyperplane for linearly separable binary classification problems. It was then generalized to linearly nonseparable problems and nonlinear classification problems. The SVM is attractive in its ability to condense the information contained in the training set and find a decision surface determined by certain points in the training set. It has proven to be effective in achieving the state-of-the-art performance in various applications. See Burges (1998) and Cristianini and Shawe-Taylor (2000) for a tutorial on the basic concept of the binary SVM.

6.2.1 Linear Support Vector Machines

The linear SVM attempts to find a linear function $f(\mathbf{x}) = \mathbf{w}'\mathbf{x} + b$ from the training set in some optimal sense, where $\mathbf{w} = (w_1, \ldots, w_d)'$. The classification rule is then constructed as $\phi(\mathbf{x}) = \text{sign}[f(\mathbf{x})]$. There are two types of linear SVM classifiers: the *hard margin* SVM and the *soft margin* SVM, depending on whether the training data is linearly separable or not.

6.2.1.1 Separable Case

Consider the simple classification problem where the training data can be perfectly separated into their classes by some hyperplane $\mathbf{w}'\mathbf{x} + b = 0$. Here $\mathbf{w} \in \mathcal{R}^d$ is normal to the hyperplane and $b \in \mathcal{R}$ is the intercept. For any separating hyperplane, we define its *geometric margin* as $d_+ + d_-$, where d_+ and d_- are, respectively, the shortest distances from the closest positive data point and the closest negative data point to the hyperplane. The linear SVM is proposed to seek the *optimal margin classifier* in the following sense: the separating hyperplane perfectly separates the training data and has the largest geometric margin. In other words, the SVM classifier solves the following problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \tfrac{1}{2}\mathbf{w}'\mathbf{w} \\ \text{subject to} \quad & y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1, \;\; i = 1, \ldots, n. \end{aligned} \tag{6.8}$$

Denote the optimal solution to (6.8) by $(\widehat{\mathbf{w}}, \hat{b})$. The resulting classification rule is then $\phi(\mathbf{x}) = \text{sign}[\widehat{\mathbf{w}}'\mathbf{x} + \hat{b}]$, which is called the *hard margin* linear SVM classifier.

The problem in (6.8) is a convex and quadratic programming (QP) problem. Classical Lagrange duality theory (Fletcher, 1987) can transform the original problem (6.8) to its Wolf dual problem, which is easier to solve than the original problem in this case. The dual problem of (6.8), involved with the Lagrange multiplies $\alpha_i \geq 0, i = 1, \ldots, n$, can be expressed as

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i'\mathbf{x}_j - \sum_{i=1}^{n}\alpha_i, \\ \text{subject to} \quad & \sum_{i=1}^{n}\alpha_i y_i = 0; \quad \alpha_i \geq 0, \ \ i = 1, \ldots, n, \end{aligned} \tag{6.9}$$

where $\alpha = (\alpha_1, \ldots, \alpha_n)'$. Denote the solution to (6.9) as $\hat{\alpha}$. The inputs $\mathbf{x}_i$ for which $\hat{\alpha}_i > 0$ are called *support vectors*. From the Karush–Kuhn–Tucker (KKT) condition (Fletcher, 1987), the optimal hyperplane must satisfy the following equations:

$$\hat{\alpha}_i[y_i(\widehat{\mathbf{w}}'\mathbf{x}_i + \hat{b}) - 1] = 0, \quad i = 1, \ldots, n.$$

Therefore, all the support vectors must lie on the margins of the SVM classifier, since they meet the equality constraint $y_i[\widehat{\mathbf{w}}'\mathbf{x}_i + \hat{b}] = 1$. After $\widehat{\alpha}$ is solved from (6.9), we can compute the normal vector of the optimal hyperplane as

$$\widehat{\mathbf{w}} = \sum_{i=1}^{n}\hat{\alpha}_i y_i \mathbf{x}_i. \tag{6.10}$$

The intercept $\hat{b}$ can be derived using the KKT condition, and any support vector with $\hat{\alpha}_i > 0$ may be used to solve $\hat{b}$ as $\hat{b} = y_i - \widehat{\mathbf{w}}'\mathbf{x}_i$. For numerical stability, Lin et al. (2002) suggested an equivalent but more robust formula to compute $\hat{b}$ as

$$\hat{b} = \frac{\sum_{i=1}^{n}\hat{\alpha}_i(1-\hat{\alpha}_i)[y_i - \widehat{\mathbf{w}}'\mathbf{x}_i]}{\sum_{i=1}^{n}\hat{\alpha}_i(1-\hat{\alpha}_i)}.$$

Let N be the number of support vectors, then the linear SVM solution can be expressed as $\hat{f}(\mathbf{x}) = \sum_{i=1}^{N}\hat{\alpha}_i y_i \mathbf{x}_i + \hat{b}$.

6.2.1.2 Nonseparable Case

In general situations, the training data is not necessarily linearly separable, i.e., there may not exist any hyperplane which perfectly separates the training points

into two classes. Therefore, no hyperplane will satisfy the constraints specified in (6.8). Cortes and Vapnik (1995) suggested introducing the nonnegative slack variables ξ_i's to relax the constraints in (6.8) as

$$y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n.$$

If a data point $(\mathbf{x}_i, y_i)$ is misclassified, we must have $\xi_i > 1$, so the quantity $\sum_{i=1}^n \xi_i$ actually forms an upper bound on the training error $\sum_{i=1}^n [-y_i f(\mathbf{x_i})]_*$. Recall that the function $[\tau]_*$ was defined in (6.2). Naturally, a small $\sum_{i=1}^n \xi_i$ is preferred. This leads to the following *soft margin* linear SVM optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \sum_{i=1}^n \xi_i + \frac{\lambda}{2}\mathbf{w}'\mathbf{w}, \\ \text{subject to} \quad & y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \xi_i, \ \ i = 1, \ldots, n. \\ & \xi_i \geq 0, \ \ i = 1, \ldots, n. \end{aligned} \tag{6.11}$$

The constant $\lambda > 0$ is a tuning parameter which controls the penalty imposed on the training error. The larger λ is, the smaller penalty is imposed on the upper bound of the training error.

Notice the two constraints in (6.11) can be combined as

$$\xi_i \geq \max\{0, 1 - y_i(\mathbf{w}'\mathbf{x}_i + b)\} = [1 - y_i(\mathbf{w}'\mathbf{x}_i + b)]_+, \quad i = 1, \ldots, n,$$

where $[\tau]_+ = \tau$ if $\tau \geq 0$; $= 0$ otherwise. Therefore, we have the following equivalent formulation of (6.11):

$$\min_{\mathbf{w},b} \quad \sum_{i=1}^n \left[1 - y_i(\mathbf{w}'\mathbf{x}_i + b)\right]_+ + \frac{\lambda}{2}\mathbf{w}'\mathbf{w}. \tag{6.12}$$

The function $[1 - yf]_+$ is termed as the *hinge* loss function in literature. Similar to the separable case, (6.11) can be solved by its dual problem

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i'\mathbf{x}_j - \sum_{i=1}^n \alpha_i, \\ \text{subject to} \quad & \sum_{i=1}^n \alpha_i y_i = 0; \quad 0 \leq \alpha_i \leq \lambda^{-1}, \ \ i = 1, \ldots, n. \end{aligned} \tag{6.13}$$

Denote the solution to (6.13) as $\hat{\alpha}_i$'s. The SVM classifier is then given by:

$$\begin{aligned} \widehat{\mathbf{w}} &= \sum_{i=1}^n \hat{\alpha}_i y_i \mathbf{x}_i, \\ \hat{b} &= y_i - \widehat{\mathbf{w}}'\mathbf{x}_i, \ \text{ for some } \ 0 < \hat{\alpha}_i < \lambda^{-1}. \end{aligned} \tag{6.14}$$

Numerically, it is wise to compute $\hat{b}$ by taking the average over all the training points satisfying $0 < \hat{\alpha}_i < \lambda^{-1}$.

6.2.2 Nonlinear Support Vector Machines

As suggested in Cortes and Vapnik (1995) and Vapnik (1998), the linear SVM can be generalized to the nonlinear SVM by first mapping the input vector $\mathbf{x}$ into a high-dimensional feature space $\mathcal{F}$, often infinite dimensional, and then fitting a linear SVM in $\mathcal{F}$. Define the map $\psi : \mathcal{R}^d \to \mathcal{F}$. The nonlinear SVM then solves

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} \left[1 - y_i(\mathbf{w}'\psi(\mathbf{x}_i) + b)\right]_+ + \frac{\lambda}{2}\mathbf{w}'\mathbf{w}, \tag{6.15}$$

where $\mathbf{w}$ is the normal vector of the hyperplane $\mathbf{w}'\psi(\mathbf{x}_i) + b = 0$ constructed in $\mathcal{F}$. The map ψ is usually chosen to enrich the features used for classification by introducing nonlinear transformations of $\mathbf{x}$. For example, a polynomial map

$$\psi(\mathbf{x}) = (x_1^2, \ldots, x_d^2, x_1x_2, \ldots, x_dx_{d-1}), \tag{6.16}$$

increases the number of features from d to d^2 by including all the second degree polynomials of $\mathbf{x}$.

The optimization problem (6.15) can be solved through its dual problem:

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \psi(\mathbf{x}_i)'\psi(\mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i, \\ \text{subject to} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0; \quad 0 \le \alpha_i \le \lambda^{-1}, \ \ i = 1, \ldots, n. \end{aligned} \tag{6.17}$$

To classify a future unlabeled sample with input $\mathbf{x}$, we apply the decision rule

$$\text{sign}\left[\hat{f}(\mathbf{x})\right] = \text{sign}\left[\sum_{i=1}^{n} \hat{\alpha}_i y_i \psi(\mathbf{x}_i)'\psi(\mathbf{x}) + \hat{b}\right]. \tag{6.18}$$

Minimizing the objective function in (6.17) and evaluating the decision rule (6.18) both require the computation of inner products $\psi(\mathbf{x}_i)'\psi(\mathbf{x})$ in a high-dimensional space, not requiring the knowledge of the explicit function form ψ. Therefore, these expensive calculations can be significantly reduced by using a positive definite kernel function K, which satisfies

$$K(\mathbf{x}, \mathbf{z}) = \psi(\mathbf{x})'\psi(\mathbf{z}), \quad \forall \mathbf{x}, \mathbf{z} \in \mathcal{R}^d.$$

For example, the kernel $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}'\mathbf{z})^2 = \psi(\mathbf{x})'\psi(\mathbf{z})$ holds for the second degree polynomial map in (6.16). Vapnik (1998), Schölkopf and Smola (2002), and Shawe-Taylor and Cristianini (2004) give a very detailed treatment on the kernel SVM. A special strength of using a *kernel trick* is to introduce nonlinearity and to deal with arbitrarily structured data. In practice, a variety of kernels have been used to construct different SVM classifiers, including

$$K(\mathbf{x},\mathbf{z}) = (\mathbf{x}'\mathbf{z}+1)^m, \qquad (m\text{th degree polynomial kernel})$$
$$K(\mathbf{x},\mathbf{z}) = \exp(\|\mathbf{x}-\mathbf{z}\|^2/2\sigma^2), \quad (\text{Gaussian kernel})$$

and the Sobolev space kernel (Wahba, 1990). The feature space $\mathcal{F}$ corresponding to the latter two types of kernels is of infinite dimension. Given a kernel function K, the SVM prediction can be easily done using the sign of

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{N} \hat{\alpha}_i y_i K(\mathbf{x}_i,\mathbf{x}) + \hat{b},$$

where N is the number of support vectors.

6.2.3 Regularization Framework for SVM

It was shown by Wahba (1999) and Evgeniou et al. (1999) that the SVM can be derived as the solution to a regularization problem in a reproducing kernel Hilbert space (RKHS). For an introduction on the RKHS theory, see Wahba (1990). Let $\mathcal{H}_K$ be an RKHS with the reproducing kernel $K(\mathbf{x},\mathbf{z}), \mathbf{x},\mathbf{z} \in \mathcal{R}^d$. Wahba (1999) proved that the SVM classifier associated with the kernel K solves the following variational problem in $\mathcal{H}_K$:

$$\min_f \sum_{i=1}^{n} [1 - y_i f(\mathbf{x}_i)]_+ + \lambda \|h\|^2_{\mathcal{H}_K}, \tag{6.19}$$

over all the functions with the form $f(\mathbf{x}) = h(\mathbf{x}) + b$, $h \in \mathcal{H}_K$ and $b \in \mathcal{R}$. Here $\|h\|^2_{\mathcal{H}_K}$ is a penalty functional measuring the roughness of f, and the regularization parameter $\lambda > 0$ controls the trade-off between the training error upper bound and the classifier complexity. Using the representer theorem of Kimeldorf and Wahba (1971), it is easy to show the minimizer of (6.19) lies in a finite dimensional space and has the representation

$$\hat{f}_\lambda(\mathbf{x}) = \sum_{i=1}^{n} \hat{c}_i K(\mathbf{x}_i,\mathbf{x}) + \hat{b}.$$

Statistical properties of the SVM were carefully studied in Lin (2002) and Zhang (2004). In particular, the following lemma reveals the connection between the SVM classifier and the Bayes rule.

Lemma 1. *(Lin et al., 2002) Let C be the misclassification cost for a binary classifier f. The minimizer of $E[1 - Yf(\mathbf{X})]_+$ is sign* $\left[p(\mathbf{x}) - \frac{C(-1,+1)}{C(-1,+1)+C(+1,-1)}\right]$.

Lemma 1 shows that the population minimizer for the hinge loss function is the Bayes rule $\phi_{\mathrm{B}}(\mathbf{x})$. In other words, if the reproducing kernel Hilbert space is rich enough such that the sign function, which is generally not smooth, can be approximated arbitrarily well in the L_2 norm by the functions in the RKHS, then the SVM solution $\hat{f}_\lambda(\mathbf{x})$ directly approaches the Bayes rule $\phi_{\mathrm{B}}(\mathbf{x})$ as $n \to \infty$. This property is known as *Fisher consistent* for classification and holds for any fixed d. There is

a large body of literature on the method of regularization in function estimation. Cox and O'Sullivan (1990) provides a general framework for studying the asymptotic properties of such methods. The convergence rate of the SVM in the standard data setting with d fixed can be found in Lin (2002) and Zhang (2004). For high-dimension low sample size data with $d \gg n$, Hall et al. (2005) recently revealed some properties of the SVM classifier based on the geometric representation of data in the new asymptotic sense, which assumes $d \to \infty$ with n fixed.

6.3 Support Vector Machines for Multiclass Problems

6.3.1 One-versus-the-Rest and Pairwise Comparison

One common strategy for a multiclass classification problem is to transform the multiclass problem into a series of binary problems. Two popular choices are the one-versus-rest approach and the pairwise comparison; see a detailed review in Weston and Watkins (1999) and Schölkopf and Smola (2002).

The one-versus-the-rest approach is designed to train k binary SVM classifiers $f_1, \ldots, f_k$, with each f_l separating class l from the remaining classes. These binary classifiers are then combined to make a final prediction according to the maximal output, i.e., $\phi(\mathbf{x}) = \text{argmax}_{l=1,\ldots,k} f_l(\mathbf{x})$. The one-versus-the-rest approach is very easy to implement in practice, so it is widely used in practice. However, this method may give poor classification performance in the absence of a dominating class (Lee et al., 2004). In other words, when none of $p_l(\mathbf{x})$'s is greater than $1/2$, the approach may break down. Another disadvantage of the one-versus-the-rest approach is that the resulting binary problems may be very unbalanced (Fung and Mangasarian, 2001). For example, if the number of classes is large, the class containing a smaller fraction of training data tends to be ignored in nonseparable cases, which may degrade the generalization performance of the classifier.

The pairwise comparison approach also trains multiple classifiers, each separating a pair of classes from each other. For a problem with k classes, this results in totally $k(k-1)/2$ binary classifiers, and the final decision rule is then constructed by a voting scheme among all the classifiers. One concern about this approach is that a large number of training tasks may be involved. For example, if $k = 10$, then we need to train 45 binary classifiers. The individual classifiers, however, usually solve smaller and easier optimization problems than in the one-versus-the-rest approach, because smaller training sets are involved and the classes have less overlap (Schölkopf and Smola, 2002).

6.3.2 Multiclass Support Vector Machines (MSVMs)

Arguably the most elegant scheme for multiclass problems is to construct a multiclass classifier which separates k classes simultaneously. The binary SVM objective

function has been modified in various ways for discriminating k classes at the same time, including Vapnik (1998), Weston and Watkins (1999), Bredensteiner and Bennett (1999), Guermeur (2002), Lee et al. (2004), Liu et al. (2004), and Wang and Shen (2007). The multiclass SVM (MSVM) generally requires the joint estimation of multiple functions $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$, each real-valued function $f_l(\mathbf{x})$ indicating the strength of evidence that $\mathbf{x}$ belongs to class l, and assigns $\mathbf{x}$ to a class with the largest value of $f_l(\mathbf{x})$.

Given an arbitrary sample point $(\mathbf{x}, y)$, a reasonable decision vector $\mathbf{f}(\mathbf{x})$ should encourage a large value for $f_y(\mathbf{x})$ and small values for $f_l(\mathbf{x}), l \neq y$. Define the vector of relative differences as

$$\mathbf{g} = (f_y(\mathbf{x}) - f_1(\mathbf{x}), \ldots, f_y(\mathbf{x}) - f_{y-1}(\mathbf{x}), f_y(\mathbf{x}) - f_{y+1}(\mathbf{x}), \ldots, f_y(\mathbf{x}) - f_k(\mathbf{x})).$$

Liu et al. (2004) called the vector $\mathbf{g}(\mathbf{f}(\mathbf{x}), y)$ the *generalized functional margin* of $\mathbf{f}$, which characterizes the correctness and strength of classification of $\mathbf{x}$ by $\mathbf{f}$. For example, $\mathbf{f}$ indicates a correct classification of $(\mathbf{x}, y)$ if $\mathbf{g}(\mathbf{f}(\mathbf{x}), y) > \mathbf{0}_{k-1}$.

The basic idea of the multiclass SVM is to impose a penalty based on the values of $f_y(\mathbf{x}) - f_l(\mathbf{x})$'s. In Weston and Watkins (1999), a penalty is imposed only if $f_y(\mathbf{x}) < f_l(\mathbf{x}) + 2$ for $l \neq y$. This implies that even if $f_y(\mathbf{x}) < 1$, a penalty is not imposed as long as $f_l(\mathbf{x})$ is sufficiently small for $l \neq y$; similarly, if $f_l(\mathbf{x}) > 1$ for $l \neq y$, we do not pay a penalty if $f_y(\mathbf{x})$ is sufficiently large. In summary, this loss function can be represented as

$$L(y, \mathbf{f}(\mathbf{x})) = \sum_{l \neq y} [2 - (f_y(\mathbf{x}) - f_l(\mathbf{x}))]_+. \tag{6.20}$$

In Lee et al. (2004), a different loss function was used

$$L(y, \mathbf{f}(\mathbf{x})) = \sum_{l \neq y} [f_l(\mathbf{x}) + 1]_+. \tag{6.21}$$

Liu and Shen (2006) suggested

$$L(y, \mathbf{f}(\mathbf{x})) = [1 - \min_l \{f_y(\mathbf{x}) - f_l(\mathbf{x})\}]_+, \tag{6.22}$$

To avoid the redundancy, a sum-to-zero constraint $\sum_{l=1}^{k} f_l = 0$ is sometimes enforced as in Guermeur (2002), Lee et al. (2004), and Liu et al. (2004).

For linear classification problems, we have $f_l(\mathbf{x}) = \mathbf{w}_l' \mathbf{x} + b_l, l = 1, \cdots, k$. The sum-to-zero constraint can be replaced by

$$\sum_{l=1}^{k} b_l = 0, \quad \sum_{l=1}^{k} \mathbf{w}_l = \mathbf{0}. \tag{6.23}$$

To achieve the nonlinear classification, we assume $f_l(\mathbf{x}) = \mathbf{w}_l'\psi(\mathbf{x}) + b_l$, where $\psi(\mathbf{x})$ represents the basis functions in the feature space $\mathcal{F}$. Similar to the binary classification, the nonlinear MSVM can be conveniently solved using a kernel function.

Furthermore, we can represent the MSVM as the solution to an optimization problem in the RKHS. Assume that $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}, \ldots, f_k(\mathbf{x})) \in \prod_{l=1}^{k}(\{1\} + \mathcal{H}_K)$ with the sum-to-zero constraint. Then an MSVM classifier can be derived by solving

$$\min_{\mathbf{f}} \sum_{i=1}^{n} L(y_i, \mathbf{f}(\mathbf{x}_i)) + \lambda \sum_{l=1}^{k} \|h_l\|_{\mathcal{H}_K}^2, \tag{6.24}$$

where $f_l(\mathbf{x}) = h_l(\mathbf{x}) + b_l, h_l \in \mathcal{H}_K$, $b_l \in \mathcal{R}$, and $L(\cdot,\cdot)$ is the loss function defined in (6.20), (6.21), or (6.22). The following lemma establishes the connection between the MSVM classifier and the Bayes rule.

Lemma 2. *(Lee et al., 2004): Let* $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$ *be the minimizer of* $E[L(Y, \mathbf{f}(\mathbf{x}))]$ *defined in* (6.21) *under the sum-to-zero constraint. Then*

$$arg \max_{l=1,\ldots,k} f_l(\mathbf{x}) = \phi_{\mathrm{B}}(\mathbf{x}).$$

6.4 Parameter Tuning and Solution Path for SVM

6.4.1 Tuning Methods

To implement the SVM classification, one needs to prespecify the values for the tuning parameters, including the regularization parameter λ and the parameters involved in the kernel function. For example, σ^2 in the Gaussian kernel is a tuning parameter. Selection of the tuning parameters is critical to the performance of SVMs, since their values have a direct impact on the generalization accuracy of a classifier. Most software packages provide a common or default value for the tuning parameters, which however is not necessarily the best choice given a particular problem. In practice, the adaptive tuning is often preferred, which is generally done by minimizing an estimate of the generalization error or some other related performance measures. Numerically, the adaptive tuning can be implemented either by a gradient descent algorithm or by a grid search over a wide range of parameter values.

There have been a host of tuning methods proposed for parameter tuning in the SVMs, including the leave-out-one cross-validation (LOOCV), fivefold cross-validation (5-CV), generalized cross-validation (GCV; Wahba et al., 2000), and the $\xi\alpha$ bound method (Joachims, 2000). Duan et al. (2001) gives a thorough evaluation and comparison for these tuning measures in various settings. Parameter tuning can be time-consuming when there are multiple parameters involved in the training. Chaplle et al. (2002) proposed a feasible approach for tuning a large number of parameters.

6.4.2 Entire Solution Path for SVM

Given a fixed tuning parameter λ, solving the SVM problem defined in (6.8), (6.11), and (6.19) requires quadratic programming. The computation cost can be large when a grid search for λ is needed. Hastie et al. (2004) developed an efficient algorithm which can derive the *entire path* of SVM solutions with essentially the computational cost of a single fit. One important discovery in their paper is that the SVM solution $[\widehat{\mathbf{w}}(\lambda), \hat{b}(\lambda)]$ has a piecewise linear trajectory in $1/\lambda$. So by identifying the break points along the path, their algorithm can get the exact SVM solution for any value of λ rapidly. This path-finder algorithm greatly speeds up the adaptive selection of tuning parameters. The solution path of the multiclass SVM was recently derived by Lee and Cui (2006).

6.5 Sparse Learning with Support Vector Machines

Since its invention, the SVM has been widely applied to many real-world problems and demonstrated superior performance, especially for high-dimensional and low sample size data. However, there are two limitations in the standard SVM approaches. Firstly, the prediction accuracy of SVMs may suffer from the presence of redundant variables, as shown by Hastie et al. (2001) and Guyon et al. (2002). One main reason is the decision rule of the SVM utilizes all the variables without discrimination. Secondly, when the true model is sparse, i.e., only a subset of input variables are involved with the underlying classification boundary, the SVM solution is less attractive in providing insight on individual variable effects since its decision rule is hardly sparse across variables.

In the modern age, new technologies of data collection and management are rapidly producing data sets of ever increasing samples sizes and dimensions (the number of variables), which often include superfluous variables. To enhance the generalization performance of a learning algorithm, it is important to identify important variables and build parsimonious classifiers with high interpretability and improved prediction performance (Kittler, 1986). Take for example the cancer classification using microarray gene expression data. Accurate identification of different cancer types is critical to achieve effective treatment and longer survival time of patients. Each type of cancer may be characterized by a group of abnormally expressed genes, called a "signature." Since gene expression arrays measure tens of thousands of genes, choosing the ones that comprise a signature that can accurately classify cancer subtypes remains a major challenge. It is therefore desired to improve the standard SVM by adding the feature of variable selection, which helps to build highly interpretable classifiers with competitive generalization performance.

General methods for variable selection in regression settings are reviewed in Chapter 2. For the SVM, one popular class of variable selection methods have been proposed based on learning with some shrinkage penalty (Bradley and Mangasarian, 1998; Gunn and Kandola, 2002; Zhu et al., 2003; Bach et al., 2004; Zou and Yuan,

2008; Zhang et al., 2006). Another class of methods are kernel scaling methods including Weston et al. (2000) and Grandvalet and Canu (2002). A special issue on variable and feature selection published by *Journal of Machine Learning Research* in 2003 introduced other approaches like Bi et al. (2003) and Rakotomamonjy (2003). These methods build automatic dimension reduction into classification, so they do not require a variable prescreening process such as the gene screening process often used in cancer classification with microarray data (Dudoit et al., 2002).

6.5.1 Variable Selection for Binary SVM

6.5.1.1 L_1 SVM

Linear Classification

The L_1 SVM was proposed by Bradley and Mangasarian (1998) and Zhu et al. (2003) to achieve simultaneous variable selection and classification. The idea of the L_1 SVM is to replace the squared L_2 norm of $\mathbf{w}$ in (6.12) with the L_1 norm:

$$\min_{\mathbf{w},b} \quad \sum_{i=1}^{n} \left[1 - y_i(\mathbf{w}'\mathbf{x}_i + b)\right]_+ + \lambda \|\mathbf{w}\|_1, \tag{6.25}$$

where $\|\mathbf{w}\|_1 = \sum_{j=1}^{d} |w_j|$. The L_1 penalty, also known as the LASSO penalty, was first studied by Tibshirani (1996) for regression problems. With a sufficiently large value of λ, this penalty can shrink small coefficients to exactly zeros and hence achieve a continuous variable selection.

Different from the standard L_2 SVM which requires quadratic programming techniques, the problem (6.25) is solved using linear programming. Zhu et al. (2003) studied the solution property of the L_1 SVM by considering the following equivalent formulation of (6.25)

$$\min_{\mathbf{w},b} \quad \sum_{i=1}^{n} [1 - y_i(\mathbf{w}'\mathbf{x}_i + b)]_+, \quad \text{subject to } \|\mathbf{w}\|_1 \le s, \tag{6.26}$$

where $s > 0$ is the tuning parameter which has the same role as λ. They showed that the solution path $[\widehat{\mathbf{w}}(s), \hat{b}(s)]$ to (6.26) is a piecewise linear function in the tuning parameter s, and proposed an efficient algorithm to compute the whole solution path for the L_1 SVM. Furthermore, the numerical results in Zhu et al. (2003) suggested that the L_1 SVM may outperform the standard SVM for high-dimensional problems, especially when there are redundant noise features. Recently, Fung and Mangasarian (2004) developed a fast Newton algorithm to solve the dual problem of (6.25) by only using a linear equation solver. Their algorithm was shown to be very effective for high-dimensional input space, even when $d \gg n$.

Nonlinear Classification

There are two ways to generalize the linear L_1 SVM for nonlinear classifications. The first way is to map the input vector into a high-dimensional feature space $\mathcal{F}$ and fit the linear L_1 SVM in $\mathcal{F}$. Define the map $\psi : \mathcal{R}^d \to \mathcal{F}$, and the nonlinear L_1 SVM is obtained by solving

$$\min_{\mathbf{w},b} \quad \sum_{i=1}^{n} \left[1 - y_i(\mathbf{w}'\psi(\mathbf{x}_i) + b)\right]_+ + \lambda \|\mathbf{w}\|_1. \tag{6.27}$$

The resulting classifier is sign $\left[\widehat{\mathbf{w}}'\psi(\mathbf{x}) + \hat{b}\right]$. This formulation was used in Zhu et al. (2003).

The second approach to formulate the nonlinear L_1 SVM is through a regularization in the RKHS framework. Instead of applying the L_1 penalty to the normal vector $\mathbf{w}$, Zhang (2006) suggested to impose a penalty functional on the functions. They first consider the function ANOVA decomposition for the nonlinear classifier f,

$$f(\mathbf{x}) = b + \sum_{j=1}^{d} f_j(x_j) + \sum_{j<k} f_{jk}(x_j, x_k) + \text{all higher-order interactions}, \tag{6.28}$$

where b is constant, f_j's are the main effects, f_{jk}'s are the two-factor interactions, and so on. The identifiability of the components in the right side of (6.28) is assured by side conditions through averaging operators. In practice, we generally truncate the decomposition (6.28) by retaining low-order interaction terms for easy computation and interpretation. Correspondingly, the entire space is truncated to its subspace $\mathcal{H} = \{1\} \oplus_{j=1}^{q} \mathcal{H}^j$, where $\mathcal{H}^j$'s are q orthogonal subspaces. The space $\mathcal{H}$ is an RKHS with the induced norm $\|\cdot\|_{\mathcal{H}}$. Zhang (2006) then proposed to solve

$$\min_{f \in \mathcal{H}} \quad \sum_{i=1}^{n} [1 - y_i f(\mathbf{x}_i)]_+ + \lambda \sum_{j=1}^{q} \|P^j f\|_{\mathcal{H}}, \tag{6.29}$$

where $P^j f$ is the projection of f onto the subspace $\mathcal{H}^j$. The penalty in (6.29), first proposed by Lin and Zhang (2006) for nonparametric regression and termed as the COSSO penalty, applies a soft-thresholding operation to functional components and hence achieves sparse solutions. In the special case of additive models, $f(\mathbf{x}) = b + \sum_{j=1}^{d} f_j(x_j)$, (6.29) becomes

$$\min_{f \in \mathcal{H}} \quad \sum_{i=1}^{n} [1 - y_i f(\mathbf{x}_i)]_+ + \lambda \sum_{j=1}^{d} \|f_j\|_{\mathcal{H}},$$

therefore the selection of main effect components is equivalent to variable selection. The empirical performance of the SVM with COSSO penalty was demonstrated in Zhang (2006) in term of its classification and variable selection accuracy. Gunn and Kandola (2002) also suggested a general framework to build interpretable classifiers with sparse kernels.

6.5.1.2 SCAD SVM

Recently Zhang et al. (2006) considered the sparse SVM with another type of penalty form: smoothly clipped absolute deviation (SCAD; Fan and Li, 2001). In linear regression models, Fan and Li (2001) first proposed the SCAD penalty for linear regression and argued that it can overcome the bias issue of the L_1 penalty. Essentially, it has been shown that the SCAD penalty produces sparse solutions by thresholding small estimates to zero and provides nearly unbiased estimates for large coefficients. Mathematically, the SCAD penalty function is expressed as

$$p_\lambda(|w|) = \begin{cases} \lambda|w| & \text{if} \quad |w| \le \lambda, \\ -\frac{(|w|^2 - 2a\lambda|w| + \lambda^2)}{2(a-1)} & \text{if} \quad \lambda < |w| \le a\lambda, \\ \frac{(a+1)\lambda^2}{2} & \text{if} \quad |w| > a\lambda, \end{cases} \tag{6.30}$$

where $a > 2$ and $\lambda > 0$ are tuning parameters. The SCAD function, plotted in Figure 6.1, is a quadratic spline function with two knots at λ and $a\lambda$. Except for the singularity at 0, the function $p_\lambda(w)$ is symmetric and nonconvex, and has a continuous first-order derivative. Though having the same form as the L_1 penalty around zero, the SCAD applies a constant penalty to large coefficients, whereas the L_1 penalty increases linearly as $|w|$ increases. It is this feature that guards the SCAD penalty against producing biases for estimating large coefficients. The linear SCAD SVM solves

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} [1 - y_i(\mathbf{w}'\mathbf{x}_i + b)]_+ + \sum_{j=1}^{d} p_\lambda(|w_j|). \tag{6.31}$$

There are two tuning parameters: λ and a in (6.31). Here λ balances the trade-off between data fitting and the model parsimony: a too small λ may lead to an overfitted classifier with little sparsity; while a too large λ may produce a very sparse classifier but with poor classification accuracy. Parameter tuning is thus needed to ensure a proper solution. Fan and Li (2001) showed that $a = 3.7$ is a good choice for most problems in practice. The nonlinear SCAD SVM can be easily extended as

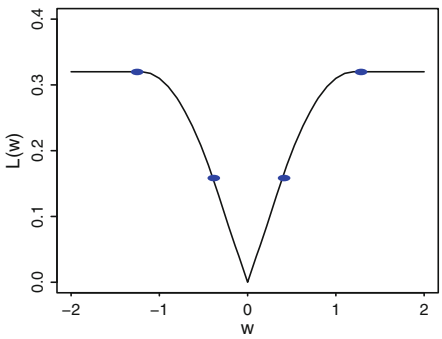


Fig. 6.1 The SCAD penalty function with $\lambda = 0.4$ and $a = 3$.

$$\min_{\mathbf{w},b} \sum_{i=1}^{n} [1 - y_i(\mathbf{w}'\psi(\mathbf{x}_i) + b)]_+ + \sum_{j=1}^{d} p_\lambda(|w_j|). \tag{6.32}$$

6.5.2 Variable Selection for Multiclass SVM

For multiclass classification problems, variable selection becomes more complex than in the binary case, since fitting the MSVM requires the estimation of multiple discriminating functions and each function has its own subset of important predictors. Recall that for k-class classification, we need to train the function vector $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$, where $f_l(\mathbf{x}) = \mathbf{w}_l'\psi(\mathbf{x}) + b_l$ and $w_l = (w_{l1}, \ldots, w_{ld})'$ for $l = 1, \ldots, k$. The final classification rule is $\phi(\mathbf{x}) = \text{argmax}_{l=1,\ldots,k} f_l(\mathbf{x})$.

6.5.2.1 L_1 Multiclass SVM

To achieve variable selection, Wang and Shen (2007) proposed to impose the L_1 penalty on the coefficients of the MSVM and solve

$$\min_{\mathbf{w}_l, b_l; l=1,\ldots,k} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)) + \lambda \sum_{l=1}^{k} \sum_{j=1}^{d} |w_{lj}|, \tag{6.33}$$

under the sum-to-zero constraint $\sum_{l=1}^{k} b_l = 0$, $\sum_{l=1}^{k} \mathbf{w}_l = \mathbf{0}$. In (6.33), the loss function L is a generalized hinge loss defined in (6.20), (6.21), or (6.22). Wang and Shen (2007) developed a statistical learning theory for the optimal solution to (6.33), quantifying the convergence rate of the generalization error of the L_1-MSVM. They also derived the entire solution path for the linear L1 MSVM. Lee and Cui (2006) provided a framework to generalize the linear L1 MSVM for nonlinear classification using the COSSO penalty (Lin and Zhang, 2006).

6.5.2.2 Supnorm Multiclass SVM

The L_1 penalty used in (6.33) does not distinguish the source of coefficients, i.e., it treats all the coefficients equally and totally ignores whether they correspond to the same variable or different variables. Intuitively, if one variable is not important, it would be desired to shrink all the coefficients associated with that variable to zeros simultaneously. Motivated by this, Zhang et al. (2008) proposed the Supnorm MSVM, which penalizes the supnorm of the coefficients associated with a certain variable. In particular, for each variable x_j, define the collection of all the coefficients associated with it as a vector $\mathbf{w}_{(j)} = (w_{1j}, \ldots, w_{kj})'$, and its supnorm as $\|\mathbf{w}_{(j)}\|_\infty = \max_{l=1,\ldots,k} |w_{lj}|$. In this way, the importance of x_j is measured by its largest absolute coefficient. The Supnorm MSVM was proposed to solve

$$\min_{\mathbf{w}_l, b_l; l=1,\ldots,k} \quad \sum_{i=1}^{n} L(y_i, \mathbf{f}(\mathbf{x}_i)) + \lambda \sum_{j=1}^{d} \|\mathbf{w}_{(j)}\|_\infty,$$
$$\text{subject to} \quad \sum_{l=1}^{k} b_l = 0, \ \sum_{l=1}^{k} \mathbf{w}_l = \mathbf{0}. \tag{6.34}$$

For three-class problems, the Supnorm MSVM is equivalent to the L_1 MSVM after adjusting the tuning parameters. Empirical studies in Zhang et al. (2008) showed that the Supnorm MSVM tends to achieve a higher degree of model parsimony than the L_1 MSVM without compromising the classification accuracy.

In (6.34), the same tuning parameter λ is used for all the terms $\|\mathbf{w}_{(j)}\|_\infty$ in the penalty, which may be too restrictive. Zhang et al. (2008) further suggested penalizing different variables with different penalties according to their relative importance. Ideally, larger penalties should be imposed on redundant variables to eliminate them from the final model more easily, while smaller penalties are used for important variables to retain them in the fitted classifier. In particular, the adaptive Supnorm MSVM solves the following problem:

$$\min_{\mathbf{w}_l, b_l; l=1,\ldots,k} \quad \sum_{i=1}^{n} L(y_i, \mathbf{f}(\mathrm{x}_i)) + \lambda \sum_{j=1}^{d} \tau_j \|\mathbf{w}_{(j)}\|_\infty,$$
$$\text{subject to} \quad \sum_{l=1}^{k} b_l = 0, \ \sum_{l=1}^{k} \mathbf{w}_l = \mathbf{0}, \tag{6.35}$$

where the weights $\tau_j \geq 0$ are adaptively chosen such that large values are used for unimportant variables and small values for important variables. Let $(\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_d)$ be the standard MSVM solution. Zhang et al. (2008) used

$$\tau_j = \frac{1}{\|\tilde{\mathbf{w}}_{(j)}\|_\infty}, \quad j = 1, \ldots, d,$$

which were shown to perform well in their numerical examples. If $\|\tilde{\mathbf{w}}_{(j)}\|_\infty = 0$, which implies an infinite penalty imposed on w_{lj}'s, then all the coefficients $\hat{w}_{lj}, l = 1, \ldots, k$, associated with x_j, are shrunk to zero altogether.

6.6 Cancer Data Analysis Using SVM

Modern DNA microarray technologies make it possible to monitor mRNA expressions of thousands of genes simultaneously. The gene expression profiles have been used for classification and diagnostic prediction of cancers recently. The SVM has been successfully applied to cancer classification and demonstrated high prediction accuracy. Since each type of cancer is often characterized by a group of abnormally expressed genes, an effective approach to gene selection helps to classify different cancer types more accurately and lead to a better understanding of genetic signatures

in cancers, In the following, we demonstrate the applications of the standard SVM and sparse SVMs to two data sets: one for two-type cancer classification and the other for multitype cancer classification. Interestingly, when $d > n$, a linear classifier often shows better performance than a nonlinear classifier (Hastie et al., 2001), even though nonlinear classifiers are known to be more flexible.

6.6.1 Binary Cancer Classification for UNC Breast Cancer Data

In the following, we present the performance of SCAD SVM on a real data set. More detailed analysis can be found in Zhang et al. (2006). These data consist of three public microarray gene expression data sets, respectively, from Stanford (Perou et al., 2000), Rosetta (Veer et al., 2002), and Singapore (Sotiriou et al., 2003), combined into a large data set. The original three data sets have, respectively, 5,974 genes and 104 patients, 24,187 genes and 97 patients, and 7,650 genes and 99 patients. In Hu et al. (2005), the three data sets are imputed for missing values, combined, and then corrected to adjust the batch bias, and the final combined set (provided by Dr. C. M. Perou from UNC) contains 2,924 genes and 300 patients. The primary interest is to select important genes and use them to distinguish two subtypes of breast cancer: Luminal and non-Luminal. We separate the whole data into three folds according to their source information. The binary SVM, L_1 SVM, SCAD SVM classifiers are all trained on two folds and tested on the remaining one. Tenfold cross-validation within the training set is used to choose λ.

Table 6.1 reports the cross-validation error in each learning and the average error rate is given in the last column. The first two rows are the error rates of the binary SVM classifiers trained, respectively, with the top 50 and 100 genes, which are ranked and selected using the t-test (Pan, 2002; Furey et al., 2000). The third row is the result for the binary SVM with all of the 2,924 genes. Note the gene preselection is not needed for the L_1 SVM and the SCAD-SVM. In the learning from Stanford data, the SCAD SVM has the lowest error rate 0.115. In the learning from Rosetta data, the SCAD SVM and the SVM with all genes are equally best. In the learning from Singapore data, the SVM with all genes is best and the SCAD SVM is the second best. The SVMs with top 50 and 100 genes do not perform so well, which may be due to the fact that the ranking method selects individual genes separately and ignore their correlations. Overall speaking, the SCAD SVM gives the lowest average error rate among all the methods.

Table 6.1 Cross-validation error rates for UNC breast cancer data.

	Stanford	Rosetta	Singapore	Average
SVM (with top 50)	0.202	0.217	0.192	0.203
SVM (with top 100)	0.192	0.206	0.111	0.170
SVM (with all genes)	0.154	0.175	0.051	0.127
L_1 SVM	0.125	0.216	0.081	0.141
SCAD SVM	0.115	0.175	0.061	0.117

Table 6.2 Number of selected genes for UNC breast cancer data.

	Stanford	Rosetta	Singapore	Average
L_1 SVM	59	63	72	65
SCAD SVM	15	19	31	22

Table 6.2 gives the number of genes selected by the L_1 SVM and the SCAD SVM. In summary, the L_1 SVM selects $59 \sim 72$ genes in three runs, while the SCAD SVM only selects $15 \sim 31$ genes. Therefore, the SCAD SVM builds a more parsimonious prediction model but with a higher classification accurate than the L_1 SVM for this data set. In Zhang et al. (2006), Figure 5 lists the UniGene identifiers of all the genes that are most frequently selected by the three methods and the classical t-test procedure.

6.6.2 Multi-type Cancer Classification for Khan's Children Cancer Data

Four different MSVMs are applied to the children cancer data set of Khan et al. (2001), available at http://research.nhgri.nih.gov/microarray/Supplement/. The data set consists of a training set of size 63 and a test set of size 20, and contains totally 2,308 genes. Khan et al. (2001) classified the small round blue cell tumors (SRBCTs) of childhood into four classes: neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL), and the Ewing family of tumors (EWS) using cDNA gene expression profiles. The distribution of the four tumor categories in the training and test sets is given in Table 6.3. Note that Burkitt lymphoma (BL) is a subset of NHL.

To analyze the data, we first standardize the data sets by applying a simple linear transformation based on the training data. Consequently, the standardized observations have mean 0 and variance 1 across genes. Then we rank all genes using their marginal relevance in class separation by adopting a simple criterion used in Dudoit et al. (2002). In particular, the relevance measure for gene g is defined to be the ratio of between classes sum of squares to within class sum of squares as follows:

$$R(g) = \frac{\sum_{i=1}^{n}\sum_{l=1}^{k} I(y_i = l)(\bar{x}_{\cdot g}^{(l)} - \bar{x}_{\cdot g})^2}{\sum_{i=1}^{n}\sum_{l=1}^{k} I(y_i = l)(x_{ig} - \bar{x}_{\cdot g}^{(l)})^2},$$

where n is the size of the training set, $\bar{x}_{\cdot g}^{(l)}$ denotes the average expression level of gene g for class l observations, and $\bar{x}_{\cdot g}$ is the overall mean expression level of gene g in the training set. To compare the variable selection performance of different methods, we select the top 100 genes which have the largest R values and bottom

Table 6.3 Distribution of four tumor categories for children cancer data.

Data set	NB	RMS	BL	EWS	Total
Training set	12	20	8	23	63
Test set	6	5	3	6	20

Table 6.4 Classification and gene selection results for children cancer data.

	Number of Selected Genes		Error	
	Top 100	Bottom 100	LOOCV	Test
MSVM	100	100	0	0
L1 MSVM	62	1	0	1
Supnorm MSVM	53	0	0	1
Adaptive Supnorm MSVM	50	0	0	1

100 genes which have the smallest R values, and build each classifier with these 200 genes. We would like to point out that the relevance measure R is equivalent to the F statistic for ANOVA problems. The classification errors and the selection frequency of these 200 genes are reported for each method.

Table 6.4 shows that all the four MSVMs have zero LOOCV error, and 0 or 1 misclassification error on the testing set. In terms of gene selection, two Supnorm MSVMs are able to eliminate all the bottom 100 genes, and they use around 50 genes out of the top 100 genes to build classifiers with competitive classification performance to other methods. The standard SVM does not have a feature of gene selection. This data set has been analyzed by many other methods, including Khan et al. (2001), Lee and Lee (2003), Dudoit et al. (2002), Tibshirani et al. (2002), and Tang and Zhang (2005), and nearly all of them yield 0 or 1 test error.

Acknowledgment Zhang's research was supported in part by National Science Foundation DMS-0405913, DMS-0645293 and by National Institute of Health NIH/NCI R01 CA-085848.

References

Agresti, A. (2002). *Categorical Data Analysis*. Wiley-Interscience, New York.

Bach, F., Lanckriet, G. R., and Jordan, M. I. (2004). Multiple kernel learning, conic duality, and the smo algorithm. In *Proceeding of the Twenty-First International Conference on Machine Learning*, Vol. 69, ACM, New York.

Bi, J., Bennett, K. P., Embrechts, M., Breneman, C. M., and Song, M. (2003). Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, 3:1229–1243.

Boser, B. E., Guyon, I. M., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. In Haussler, D., editor, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152. ACM Press, Pittsburgh, PA.

Bradley, P. S. and Mangasarian, O. L. (1998). Feature selection via concave minimization and support vector machines. In Shavlik, J., editor, *Machine Learning Proceedings of the Fifteenth International Conference (ICML '98)*, pages 82–90. Morgan Kaufmann, San Francisco, CA.

Bredensteiner, E. J. and Bennett, K. P. (1999). Multicategory classification by support vector machines. *Computational Optimization and Applications*, 12:35–46.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167.

Chaplle, O., Vapnik, V., Bousquet, O., and Mukherjee, S. (2002). Choosing kernel parameters for support vector machines. *Machine Learning*, 46:131–159.

Cortes, C. and Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20:1–25.

Cox, D. and O'Sullivan, F. (1990). Asymptotic analysis of penalized likelihood and related estimator. *Annals of Statistics*, 18:1676–1695.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK.

Duan, K., Keerthi, S., and Poo, A. (2001). Evaluation of simple performance measures for tuning svm hyperparameters. Technical Report CD-01-11, Department of Mechanical Engineering, National University of Singapore.

Dudoit, S., Fridlyand, J., and Speed, T. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of American Statistical Association*, 97:77–87.

Evgeniou, T., Pontil, M., and Poggio, T. (1999). A unified framework for regularization networks and support vector machines. Technical report, M.I.T. Artificial Intelligence Laboratory and Center for Biological and Computational Learning Department of Brain and Cognitive Sciences.

Fan, J. and Li, R. Z. (2001). Variable selection via penalized likelihood. *Journal of the American Statistical Association*, 96:1348–1360.

Fletcher, R. (1987). *Practical Methods of Optimization*. Wiley-Interscience, New York, NY.

Fung, G. and Mangasarian, O. L. (2001). Multicategory proximal support vector machine classifiers. Technical Report 01-06, University of Wisconsin-Madison, Data Mining Institute.

Fung, G. and Mangasarian, O. L. (2004). A feature selection newton method for support vector machine classification. *Computational Optimization and Applications Journal*, 28(2):185–202.

Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schurmmer, M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16:906–914.

Grandvalet, Y. and Canu, S. (2002). Adaptive scaling for feature selection in SVMs. *Neural Information Processing Systems*, 553–560.

Guermeur, Y. (2002). Combining discriminant models with new multi-class SVMs. *Pattern Analysis and Applications*, 5:168–179.

Gunn, S. R. and Kandola, J. S. (2002). Structural modeling with sparse kernels. *Machine Learning*, 48:115–136.

Guyon, I., Weston, J., and Barnhill, S. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422.

Hall, P., Marrson, S., and Neeman, A. (2005). Geometric representation for high dimension low sample size data. *Journal of Royal Statistical Society, B*, 67:427–444.

Hand, D. J. (1997). *Construction and Assessment of Classification Rules*. John Wiley and Sons, Chichester, England.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Element of Statistical Learning*. Springer, New York.

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. (2004). The entire regularization path for the support vector machines. *Journal of Machine Learning Research*, 5:1391–1415.

Hu, Z., Fan, C., Marron, J. S., He, X., Qaqish, B. F., Karaca, G., Livasy, C., Carey, L., Reynolds, E., Dressler, L., Nobel, A., Parker, J., Ewend, M. G., Sawyer, L. R., Xiang, D., Wu, J., Liu, Y., Karaca, M., Nanda, R., Tretiakova, M., Orrico, A. R., Dreher, D., Palazzo, J. P., Perreard, L., Nelson, E., Mone, M., Hansen, H., Mullins, M., Quackenbush, J. F., Olapade, O. I., Bernard,

B. S., and Perou, C. M. (2005). The molecular portraits of breast tumors are conserved across microarray platforms. submitted.

Joachims, T. (2000). Estimating the generalization performance of an SVM efficiently. In *Proceedings of ICML-00, 17th International Conference on Machine Learning*, Morgan Kaufman, San Francisco, 431–438.

Khan, J., Wei, J., Ringer, M., Saal, L., Ladanyi, M., Westerman, F., Berthold, F., Schwab, M., Antonescu, C., Peterson, C., and Meltzer, P. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural network. *Nature Medicine*, Jun.; 7(6):673–679.

Kimeldorf, G. and Wahba, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–85.

Kittler, J. (1986). Feature selection and extraction. In T.Y.Young and K.-S. Fu, editors, *Handbook of Pattern Recognition and Image Processing*. Academic Press, New York.

Lee, Y. and Cui, Z. (2006). Characterizing the solution path of multicategory support vector machines. *Statistica Sinica*, 16:391–409.

Lee, Y. and Lee, C. (2003). Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*, 19:1132–1139.

Lee, Y., Lin, Y., and Wahba, G. (2004). Multicategory support vector machines, theory, and application to the classification of microarray data and satellite ra diance data. *Journal of American Statistical Association*, 99:67–81.

Lin, Y. (2002). SVM and the Bayes rule in classification. *Data Mining and Knowledge Discovery*, 6:259–275.

Lin, Y. and Zhang, H. H. (2006). Component selection and smoothing in smoothing spline analysis of variance models. *Annals of Statistics*, 34:2272–2297.

Lin, Y., Lee, Y., and Wahba, G. (2002). Support vector machines for classification in nonstandard situations. *Machine Learning*, 46:191–202.

Liu, Y. and Shen, X. (2006). Multicategory psi-learning and support vector machine: computational tools. *Journal of American Statistical Association*, 99:219–236.

Liu, Y., Shen, X., and Doss, H. (2004). Multicategory psi-learning and support vector machine: computational tools. *Journal of Computational and Graphical Statistics*, 14:219–236.

Pan, W. (2002). A comparative review of statistical methods for discovering differently expressed genes in replicated microarray experiments. *Bioinformatics*, 18:546–554.

Perou, C., Srlie, T., Eisen, M., van de Rijn, M., Jeffrey, S., Rees, C., Pollack, J., Ross, D., Johnsen, H., Akslen, L., Fluge, O., Pergamenschikov, A., Williams, C., Zhu, S., Lning, P., Brresen-Dale, A., Brown, P., and Botstein, D. (2000). Molecular portraits of human breast tumors. *Nature*, 406:747–752.

Rakotomamonjy, A. (2003). Variable selection using svm-based criteria. *Journal of Machine Learning Research*, 3:1357–1370.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press, Cambridge, MA.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Recognition*. Cambridge University Press, Cambridge, UK.

Sotiriou, C., Neo, S., McShane, L., Korn, E., Long, P., Jazaeri, A., Martiat, P., Fox, S., Harris, A., and Liu, E. (2003). Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proceedings of the National Academy of Sciences*, 100(18):10393–10398.

Tang, Y. and Zhang, H. H. (2005). Multiclass proximal support vector machines. *Journal of Computational and Graphical Statistics*, 15:339–355.

Tibshirani, R. J. (1996). Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society, B*, 58:267–288.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences USA*, 99:6567–6572.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, New York.

Veer, L. V., Dai, H., van de Vijver, M., He, Y., Hart, A., Mao, M., Peterse, H., van der Kooy, K., Marton, M., Witteveen, A., Schreiber, G., Kerkhoven, R., Roberts, C., Linsley, P., Bernards, R., and Friend, S. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536.

Wahba, G. (1990). *Spline Models for Observational Data*, volume 59. SIAM. CBMS-NSF Regional Conference Series in Applied Mathematics.

Wahba, G. (1999). Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In Scholkopt, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods–Support Vector Learning*. MIT Press, Cambridge, MA.

Wahba, G., Lin, Y., and Zhang, H. H. (2000). Generalized approximate cross validation for support vector machines, or, another way to look at margin-like quantities. In Smola, Bartlett, Scholkopf, and Schurmans, editors, *Advances in Large Margin Classifiers*. MIT Press.

Wang, L. and Shen, X. (2007). On l1-norm multiclass support vector machines: methodology and theory. *Journal of American Statistical Association*, 102:583–594.

Weston, J. and Watkins, C. *Multi-class support vector machines*, In Verleysen, M., editor, *Proceedings of ESANN99*, Brussels, D. Facto Press (1999).

Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V. Feature selection for SVMs. *In Advances in Neural Information Processing Systems (NIPS)* 13, (2000). (Edited by: TK Leen, TG Dietterich, V Tresp). MIT Press 2001, 668–674.

Zhang, T. (2004). Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 32:56–85.

Zhang, H. (2006). Variable selection for support vector machines via smoothing spline anova. *Statistica Sinica*, 16:659–674.

Zhang, H., Ahn, J., Lin, X., and Park, C. (2006). Gene selection using support vector machines with nonconvex penalty. *Bioinformatics*, 22:88–95.

Zhang, H., Liu, Y., Wu, Y., and Zhu, J. (2008). Variable selection for multicategory SVM via supnorm regularization. *The Electronic Journal of of Statistics*. to appear.

Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2003). 1-norm support vector machines. *NIPS* 16. MIT Press.

Zou, H. and Yuan, M. (2008). The F_∞ support vector machines. *Statistica Sinica*, 18:379–398.

Chapter 7
Bayesian Approaches: Nonparametric Bayesian Analysis of Gene Expression Data

Sonia Jain

Microarray gene expression experiments in molecular oncology studies are escalating in popularity. As a result, there is a growing need for flexible statistical methods to reliably extract biologically important information from the massive amount of data obtained from these high-throughput assays. The clustering of gene expression profiles to identify sets of genes that exhibit similar expression patterns is of particular interest in determining gene co-expression and co-regulation. In this chapter, we review Bayesian approaches to analyze microarray data, such as EBArrays (Newton et al., 2004), probability of expression (POE) (Garrett and Parmigiani, 2003), and infinite Bayesian mixture models (Medvedovic and Sivaganesan, 2002). Later in the chapter, we specifically focus on a model-based latent class approach to clustering gene expression profiles. A nonparametric Bayesian mixture model is used to model microarray data that are assumed to arise from underlying heterogeneous mechanisms. To compute these Bayesian models, split–merge Markov chain Monte Carlo is employed to avoid computational problems such as poor mixing and slow convergence to the posterior distribution. We demonstrate the utility of split–merge methods in this high-dimensional application by considering data from a leukemia microarray study.

7.1 Introduction

The analysis of gene expression microarray data in oncology is a fertile area of research that has bred a wide assortment of statistical data mining techniques for a variety of genomic applications, such as gene identification, drug discovery, disease diagnosis, and of course, profiling gene expression patterns. The advent of

S. Jain
Division of Biostatistics and Bioinformatics, University of California, San Diego, 9500 Gilman Drive, MC-0717, La Jolla, CA 92093-0717, USA
email: sojain@ucsd.edu

X. Li, R. Xu (eds.), *High-Dimensional Data Analysis in Cancer Research*, Applied Bioinformatics and Biostatistics in Cancer Research, DOI 10.1007/978-0-387-69765-9_7,

microarray technology allows biologists to study the expression profiles of thousands of genes simultaneously, which can provide a better picture of genetic pathways by elucidating genes that are co-regulated or redundant. From a statistical perspective, the vast amount of potentially highly correlated data leads to some interesting methodological questions regarding experimental design and the analysis of large and complex data sets. Statistical techniques that have been developed to analyze microarray data have played an important role in the discovery and understanding of the molecular basis of disease, particularly cancer. Examples include Eisen et al. (1998) and Golub et al. (1999).

Before discussing Bayesian methodology, a brief introduction to genomic terminology is given. Griffiths et al. (1996) provides a thorough introduction. The most basic unit of heredity is the *gene*, which is a sequence of nucleotides (a segment of *DNA* or *deoxyribonucleic acid*) located in a specific position on a particular chromosome that encodes a genetic product, usually a *protein*. Often, researchers are interested in *gene expression*, a process by which a gene's coded information is converted into one of its products. Expressed genes include those that are *transcribed* into *mRNA* (*messenger ribonucleic acid*) and subsequently *translated* into a protein. Expressed genes at the transcription level are usually the focus of microarray experiments, since expressed genes are a good indication of a cell's activity.

Without going into the functional details, a microarray is a large collection of DNA templates to which unknown samples of mRNA are matched based on base-pairing rules of nucleotides (known as *hybridization*) by some automated process. These experiments monitor mRNA expression of thousands of genes in a single experiment. There are several types of microarray experiments, of which cDNA (*complementary DNA*) microarrays and high-density oligonucleotide arrays are the most common (see Schena, 1999, for further details).

Parametric and nonparametric Bayesian models are a critical tool in analyzing high-dimensional data sets, particularly in the context of DNA microarrays, in which the data is potentially highly correlated. In Section 7.2, we provide an overview of some Bayesian methods that have been applied to gene expression analysis, such as EBArrays (Newton et al., 2001), (POE) (Garrett and Parmigiani, 2003), and infinite Bayesian mixture models (Medvedovic and Sivaganesan, 2002).

Later in the chapter, we focus on a particular aspect of microarray analysis: clustering gene expression profiles to determine genetic co-expression and co-regulation. Cluster analysis is often considered the last "stage" of analysis in a microarray experiment (after experimental design, image analysis, pre-processing, normalization, and determination of differential gene expression) and is sometimes considered exploratory. Model-based approaches to clustering, in which the probability distribution of observed data is estimated by a statistical model, have been shown empirically to frequently outperform traditional clustering methods, such as agglomerative hierarchical algorithms and k-means clustering (Yeung et al., 2003).

We model gene expression microarray data as arising from a mixture of normal distributions and place a Dirichlet process prior on the mixing distribution, so that a countably infinite number of mixture components can be modeled. We compute this Dirichlet process mixture via a nonconjugate form of Gibbs sampling (Neal, 2000)

and a nonincremental split–merge Metropolis–Hastings technique (Jain, 2002; Jain and Neal, 2004, 2007). Because gene expression data are both high dimensional and highly correlated, the incremental Gibbs sampler exhibits convergence problems such as poor mixing and slow convergence. The nonincremental split–merge technique overcomes these problems by moving groups of observations in a single update and avoiding low probability states. We demonstrate the improved performance of the split–merge technique in Section 7.5 by considering a leukemia data set.

This chapter is organized as follows. Section 7.2 provides an overview of Bayesian approaches to high-dimensional microarray data analysis. In Section 7.3, we present the Dirichlet process mixture model used for statistical inference in the clustering of gene expression data. Section 7.4 describes the split–merge Markov chain Monte Carlo technique that estimates the Dirichlet process mixture model. Section 7.5 illustrates the application of the nonparametric Bayesian mixture model in analyzing leukemia microarray data. Section 7.6 provides concluding remarks.

7.2 Bayesian Analysis of Microarray Data

The role that Bayesian methods play in biomedical applications has increased over the past decade (Beaumont and Rannala, 2004). The Bayesian paradigm is often considered an appealing choice for modeling, since historical and prior knowledge can be naturally incorporated into the Bayesian framework via the prior distribution. Due to advances in statistical computation, in particular, Markov chain Monte Carlo (see Tierney, 1994; Gilks et al., 1996), it is now feasible to estimate empirically these analytically intractable models.

Bayesian inference provides a flexible framework for high-dimensional and complex data analysis. Bayesian methods have been widely used in genomic applications such as gene expression microarray analysis, SNP analysis, protein mass spectrometry, and serial analysis of gene expression (SAGE). An excellent recent reference that describes only Bayesian approaches to microarray and proteomic analyses is Do et al. (2006). The focus of this chapter is on DNA microarray analysis.

Aside from the high dimensionality inherent in microarray data, this type of data also poses challenges due to its complex hierarchical error structure (potential contamination due to biological noise and equipment/calibration error), variability of expression levels across genes, and lack of information regarding the number of expected clusters. A hierarchical Bayesian model can be used to model these complexities by specifying appropriate prior distributions and combining the priors with the observed microarray data to yield a posterior clustering allocation of the observations to mixture components. Here, we outline several Bayesian approaches to microarray analysis, including EBArrays (Newton et al., 2004), POE (Garrett and Parmigiani, 2003), and infinite Bayesian mixture models (Medvedovic and Sivaganesan, 2002).

7.2.1 EBarrays

EBarrays is a parametric empirical Bayes approach proposed by Newton and Kendziorski (2003), Kendziorski et al. (2003), and Newton et al. (2001); Newton et al. (2004); Newton et al. (2006). This approach takes a hierarchical mixture model approach to address the issue of differential gene expression. The model is considered hierarchical since at the lower level of this mixture model, the mixture component distributions model the conditional variation in the expression profiles given their expected means. At higher levels of this hierarchy, there is an overlying distribution that describes the variation in the expected means. This allows for sharing of properties across genes, since the genes are connected by having expression values drawn from a common probability distribution.

Specifically, the model represents the probability distribution of expression intensities $\mathbf{y}_j = (y_{j1}, \ldots, y_{jI})$ measured on gene j, where I is the total number of samples. When studying differential gene expression between two groups, it is possible that the I samples are exchangeable and that the distribution of the gene expression intensities is not influenced by group membership. In this case, the authors denote this as "equivalent expression" or EE_j for gene j. Formally, they consider the gene expression intensities y_{ji} as independent random departures from a gene-specific mean μ_j; that is, the intensities are drawn from an observation distribution $f_{\text{obs}}(\cdot|\mu_j)$.

On the other hand, if the group membership impacts the distribution of the gene intensities, then this is "differential expression" or DE_j, and two distinct means, μ_{j1} and μ_{j2}, are necessary for each group. Kendziorski and colleagues assume that the gene intensities are independent and identically distributed from a common distribution $\pi(\mu)$, such that information sharing across genes may occur. Note that if the μ_j's are fixed effects, then borrowing information across genes would not be possible.

A second component to this model is the discrete mixing weight, p, which specifies the proportion of genes that are differentially expressed (DE). Hence, $1-p$ represents the proportion of genes equivalently expressed (EE). The distribution for data $\mathbf{y}_j = (\mathbf{y}_{j1}, \ldots, y_{jI})$ arising from EE gene j is:

$$f_0(\mathbf{y}_j) = \int \left(\prod_{i=1}^{I} f_{\text{obs}}(y_{ji}|\mu) \right) \pi(\mu) \mathrm{d}\mu \tag{7.1}$$

If, on the other hand, gene j is differentially expressed, then the data follow the following distribution:

$$f_1(\mathbf{y}_j) = f_0(\mathbf{y}_{j1}) f_0(\mathbf{y}_{j2}) \tag{7.2}$$

because different means characterize the two distinct subsets of data $\mathbf{y}_{j1}$ and $\mathbf{y}_{j2}$.

The marginal distribution of the data is given as:

$$p f_1(\mathbf{y}_j) + (1-p) f_0(\mathbf{y}_j) \tag{7.3}$$

Using Bayes' rule, Kendziorski et al. (2003) show that the posterior probability of differential expression is then:

$$\frac{pf_1(\mathbf{y}_j)}{pf_1(\mathbf{y}_j)+(1-p)f_0(\mathbf{y}_j)} \tag{7.4}$$

The EBarrays method can be extended to beyond two groups to handle multiple patterns of expression. If there are $m+1$ distinct patterns of expression possible for data $\mathbf{y}_j = (y_{j1},\ldots,y_{jI})$ including the null pattern of equivalent expression across all samples, then (7.3) can be generalized to the following mixture model:

$$\sum_{k=0}^{m} p_k f_k(\mathbf{y}_j) \tag{7.5}$$

where p_k are the mixing weights, and component densities, f_k, represent the predictive distribution of measurements for each distinct pattern. The posterior probability of expression for pattern k generalizes to:

$$P(k|\mathbf{y}_j) \propto p_k f_k(\mathbf{y}_j). \tag{7.6}$$

The posterior probabilities provides the basis for statistical inference for each gene's expression pattern and can be used to classify genes into clusters.

This method is implemented in the Bioconductor package called `EBarrays` for two types of models: log-normal model and Gamma–Gamma model. Both of these models assume a constant coefficient of variation, which needs to be verified by model diagnostics. One of the benefits of the EBarrays approach is that the sources of variability for the entire gene expression profile is modeled simultaneously rather than applying a basic statistical test (e.g., modified t-test) repeatedly for each gene and then drawing inferences by a post-hoc statistical adjustment.

7.2.2 Probability of Expression

Another well-known Bayesian approach to microarray analysis is probability of expression (POE) first described by Parmigiani et al. (2002). Other POE references include Garrett and Parmigiani (2003) and Garrett-Mayer and Scharpf (2006). POE is a powerful microarray meta-analysis tool. Often, data from different microarray platforms are not directly comparable. POE takes continuous gene expression data from any array platform and reexpresses it on a three-component categorical scale, so that the data may be easily compared.

The data are categorized as either over-expressed, under-expressed, or no change. Garrett and Parmigiani (2003) use the following notation to represent these categories: $e_{ji} = -1$ if gene j is under-expressed in sample i, $e_{ji} = 0$ if gene j has baseline (no change) expression in sample i, and $e_{ji} = 1$ if gene j is over-expressed in sample i.

For each gene j, the probability distribution of data y_{ji} given e_{ji} is denoted by $f_{e,j}$, where

$$y_{ji}|(e_{ji} = e) \sim f_{e,j}(\cdot) \tag{7.7}$$

where $e \in \{-1, 0, 1\}$.

A mixing weight is defined to represent the proportion of over-expressed samples in gene j, denoted as, π_j^+. Under-expressed samples are represented by π_j^-. The proportion of all differentially expressed samples is then $\pi_j = \pi_j^+ + \pi_j^-$. The e_{ji}'s are assumed to be independent, conditional on the π_j's and f's.

Bayes' rule is utilized to estimate the probability of over- or under-expression for each gene. Specifically, the probability that a gene is over-expressed (p_{ji}^+) is

$$\begin{aligned} p_{ji}^+ &= P(e_{ji} = 1|y_{ji}, \omega) \\ &= \frac{\pi_j^+ f_{1,j}(y_{ji})}{\pi_j^+ f_{1,j}(y_{ji}) + \pi_j^- f_{-1,j}(y_{ji}) + (1 - \pi_j^+ - \pi_j^-) f_{0,j}(y_{ji})} \end{aligned} \tag{7.8}$$

where ω denotes all unknown parameters.

Similarly, the probability that a gene is under-expressed (p_{ji}^-) is:

$$\begin{aligned} p_{ji}^- &= P(e_{ji} = 1|y_{ji}, \omega) \\ &= \frac{\pi_j^- f_{-1,j}(y_{ji})}{\pi_j^+ f_{1,j}(y_{ji}) + \pi_j^- f_{-1,j}(y_{ji}) + (1 - \pi_j^+ - \pi_j^-) f_{0,j}(y_{ji})} . \end{aligned} \tag{7.9}$$

POE uses a latent variable approach, in which a parametric Bayesian hierarchical mixture model is employed. Currently, the authors have implemented their technique using the uniform distribution for $f_{1,j}$ and $f_{-1,j}$, and a normal distribution for $f_{0,j}$. Model identifiability limits the choices for these densities. Gene-specific parameters are created so that information may be borrowed across genes.

Markov chain Monte Carlo methods are used to estimate the Bayesian mixture model. In particular, the Metropolis–Hastings method is utilized (Metropolis et al., 1953; Hastings, 1970). Once this model has been fit, a POE scale is created as a function of the posterior estimates of the model parameters. The POE scale is defined as the differences in the values of p_{ji}^+ and p_{ji}^-; hence, the POE scale is defined from -1 to 1. Genes with positive probability of over-expression have a POE range between 0 and 1, while genes with positive probability of under-expression have a POE value between -1 and 0. Note that in this representation, it is not possible for a particular y_{ji} to have positive probability of both over- and under-expression.

The POE value for gene j in sample i is represented as:

$$p_{ji} = p_{ji}^+ + p_{ji}^- \tag{7.10}$$

Since gene expression experiments are expensive and studies are often conducted on a small number of samples, combining data statistically across experiments and platforms is advantageous. POE is implemented as a library in the statistical software, R.

7.3 Nonparametric Bayesian Mixture Model

Yeung et al. (2003) showed empirically that model-based approaches to clustering, such as finite mixture models, in which the probability distribution of observed data is estimated by a statistical model, frequently outperform traditional clustering methods. The criteria used in these comparisons were cluster accuracy and cluster stability. Further, Medvedovic and Sivaganesan (2002) and Medvedovic et al. (2004) demonstrated that a Bayesian clustering approach, via infinite mixture models, produces more realistic clustering allocations compared to finite mixture models. Medvedovic and Sivaganesan (2002) and Medvedovic et al. (2004) applied a Dirichlet process mixture model (similar to Neal, 2000) to microarray data and estimated the posterior distribution via Gibbs sampling. Please note that the Dirichlet process mixture model is commonly referred to as a nonparametric Bayesian model, even though it has a countably infinite number of parameters. The term nonparametric in this context refers to the types of applications that these Bayesian models can be applied to, such as density estimation and scatterplot smoothing. Medvedovic et al. (2004) have noted that more efficient sampling schemes other than traditional Gibbs sampling, such as Jain and Neal (2004), required further investigation to determine if this would improve convergence to the posterior distribution. The purpose of the remainder of this chapter is to examine this proposition.

We will consider the following Dirichlet process mixture model, where we assume that the observations arise from an underlying mixture of simple parametric distributions having the form $F(\theta)$. In particular, we assume that each observation is drawn from a normal distribution and that the elements are independent and identically distributed.

We will assume that the component normal distribution's parameters, the mean μ and variance τ^{-1}, are independently drawn from some mixing distribution G. Instead of requiring G to take a parametric form, we place a Dirichlet process prior (Antoniak, 1974; Blackwell and MacQueen, 1973; Ferguson, 1983), a distribution over the space of distribution functions, on G. This produces the following hierarchical mixture model:

$$\begin{aligned} y_{ih} \mid \mu_{ih},\ \tau_{ih} &\sim F(y_{ih}; \mu_{ih}, \tau_{ih}) \ = \ N(y_{ih}; \mu_{ih}, \tau_{ih}^{-1} \boldsymbol{I}) \\ (\mu_{ih},\ \tau_{ih}) \mid G &\sim G \\ G &\sim DP(G_0, \alpha) \\ G_0 &= N(\mu; w, B^{-1}) \cdot \text{Gamma}(\tau; r, R) \end{aligned} \tag{7.11}$$

where G_0 and α are the two Dirichlet process prior parameters. G_0 is the base measure, while α is a concentration or "smoothing" parameter that takes values greater than zero. The parameter α determines the extent that the observations are clustered together, and may be either fixed to a particular value or treated as a random variable and given a hyperprior. If α is large, we expect a large number of distinct clusters drawn from the prior. If α is small, however, then there is high probability that we will fit a smaller number of mixture components; that is, there is a high probability that a class indicator will be drawn from a previously used component. Here, subscript i refers to attributes rather than different observations. Under G_0, which is the prior over the entire vector over attributes, the model parameters (μ_h, τ_h) are independent.

The probability density function for the prior distribution of μ given in (7.11) is

$$f(\mu \mid w, B) = \left(\frac{B}{2\pi}\right)^{\frac{1}{2}} \exp\left(\frac{-B}{2}(\mu - w)^2\right) \tag{7.12}$$

where B is a precision parameter.

The probability density function for the prior for τ is

$$f(\tau \mid r, R) = \frac{1}{R^r \Gamma(r)} \tau^{r-1} \exp\left(\frac{-\tau}{R}\right). \tag{7.13}$$

This parameterization of the Gamma density is adopted throughout this chapter.

Hyperpriors could be placed on w, B, r, and R to add another stage to this hierarchy if desired.

Equation (7.11) represents a countably infinite mixture model (Ferguson, 1983). When we integrate G over the Dirichlet process prior, the model parameters follow a generalized Pólya urn scheme (see Blackwell and MacQueen, 1973). As Neal (2000) showed, by integrating over G, the clustering property of the Dirichlet process is evident, since there is positive probability that some of the model parameters are identical. Thus, we can group identical model parameters into clusters, denoted c_i, which represents the latent class associated with observation i. That is, the sampling scheme for the stochastic class indicator, c_i, is

$$\begin{aligned} P(c_i = c \mid c_1, \ldots, c_{i-1}) &= \frac{n_{i,c}}{i - 1 + \alpha}, \quad \text{for } c \in \{c_j\}_{j<i} \\ P(c_i \neq c_j \text{ for all } j < i \mid c_1, \ldots, c_{i-1}) &= \frac{\alpha}{i - 1 + \alpha} \end{aligned} \tag{7.14}$$

where $n_{i,c}$ is the number of c_k for $k < i$ that are equal to c. The labeling of the indicator c_i is irrelevant in the above probabilities; all that matters is which c_i's are equal to each other. The probabilities shown in (7.14) define the Dirichlet process model.

7.4 Posterior Inference of the Bayesian Model

Posterior inference of the Dirichlet process mixture model is conducted via Markov chain Monte Carlo. Markov chain Monte Carlo methods were first used in statistical physics problems, but are now commonly employed in statistical modeling. In particular, Bayesian computations can often be complex due to high dimensionality or complexity of the posterior distribution, so direct sampling is infeasible. The basic idea of Markov chain Monte Carlo is simply performing Monte Carlo integration using Markov chains. Monte Carlo integration draws samples from the required distribution and forms sample averages to approximate expectations. It would be ideal if the samples generated are independent, but that is not necessary. The novelty of Markov chain Monte Carlo is to cleverly construct a Markov chain so that it has the distribution of interest, say π, as its stationary distribution. For a comprehensive review of Markov chain sampling methods, refer to Gilks et al. (1996) and Tierney (1994). In the following, we compare the performance of Gibbs sampling (Neal, 2000) to split–merge Markov chain Monte Carlo (Jain and Neal, 2004, 2007).

7.4.1 Gibbs Sampling

Following the articles by Geman and Geman (1984) and Gelfand and Smith (1990), Gibbs sampling became a very popular Markov chain Monte Carlo algorithm among Bayesian statisticians. Suppose that the distribution of interest is the joint distribution, $\pi(x)$, where $x = (x_1, \ldots, x_n)$. Each component could be a scalar, vector, or matrix. When direct sampling of $\pi(x)$ is not possible, but the conditional distribution for each x_i is available, Gibbs sampling is appropriate. A Gibbs sampling scan updates each x_i according to its conditional distribution given the current value of all the other x_j where $j \neq i$, which is denoted as x_{-i}:

$$\pi(x_i|x_{-i}) = \pi(x_i|x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

The new value for x_i is selected without reference to the former value that it replaces, and this new value is used immediately when drawing a value for the next component, x_{i+1}. The transition kernel is a product of these *full conditional distributions* for each individual update required to produce a single iteration of Gibbs sampling. The components may be updated via a random scan or systematic scan. The random scan satisfies detailed balance, but the systematic scan does not. However, π is an invariant distribution for the systematic scan Markov chain since it is an invariant distribution for each individual x_i update.

When G_0 is not a conjugate prior for F, we cannot analytically compute the integral $\int F(y_i, \phi)\, \mathrm{d}G_0(\phi)$. This leads to numerous computational difficulties. West et al. (1994) suggest numerical integration, but if the parameters, ϕ_c, are high-dimensional, this can become rather cumbersome. Another recommendation on how to calculate the integral is Monte Carlo integration based on samples from G_0, but as

MacEachern and Müller (1998) argue, this approximation can be quite inaccurate. Instead, MacEachern and Müller (1998) propose an exact Gibbs sampling algorithm that employs the addition of auxiliary parameters, which Neal (2000) (Algorithm 8) further improves upon to make the algorithm more efficient. Here, Neal's auxiliary method, which is used later in this chapter, is briefly described. Consult Neal (2000) for further details.

Auxiliary variable methods sample from a distribution π_x for x by sampling from some distribution π_{xy} for (x,y), with respect to which the marginal distribution of x is π_x. Neal extends this idea by considering the auxiliary variable y to be temporary during the Markov chain simulation.

This idea is utilized when updating the c_i to avoid integrating with respect to G_0. The permanent state of the Markov chain consists of c_i and the ϕ_c. However, when c_i is updated, temporary auxiliary variables are introduced that represent values for the parameters of components which have no other observations associated with them. Then, Gibbs sampling is performed to update the c_i, but with respect to the distribution including these auxiliary parameters. The number of auxiliary components (and corresponding auxiliary parameters) is a tuning parameter, designated here by v (corresponding to m in Neal, 2000).

The conditional prior used in this version of Gibbs sampling will depend on whether c_i is associated with an existing or an auxiliary component. The probability of c_i being equal to a c in $\{1,\ldots,k\}$ will be $n_{-i,c}/(n-1+\alpha)$, where k is the number of distinct c_j for $j \neq i$ and $n_{-i,c}$ is the number of times c occurs among the c_j for $j \neq i$. The probability of c_i having some other value will be $\alpha/(n-1+\alpha)$, which is split equally among the v auxiliary components.

If $c_i = c_j$ for some $j \neq i$, the auxiliary parameters are drawn independently from G_0. But, if $c_i \neq c_j$ for all $j \neq i$, then it is associated with one of the v auxiliary parameters. The corresponding ϕ is equal to the existing ϕ_{c_i}. The ϕ values for the other auxiliary components are drawn independently from G_0.

A Gibbs sampling update for c_i is performed by drawing a new value from its conditional distribution using the following probabilities:

$$P(c_i = c \mid c_{-i}, y_i, \phi_1, \ldots, \phi_{(k+v)}) = \begin{cases} b\,\dfrac{n_{-i,c}}{n-1+\alpha}\,F(y_i,\phi_c),\ 1 \leq c \leq k \\ b\,\dfrac{\alpha/v}{n-1+\alpha}\,F(y_i,\phi_c),\ k < c \leq (k+v) \end{cases} \tag{7.15}$$

where $n_{-i,c}$ is the number of c_j for $j \neq i$ that are equal to c, k is the number of distinct c_j for $j \neq i$, v is the number of auxiliary parameters, $F(y_i,\phi_c)$ is the likelihood for observation i, and b is the appropriate normalizing constant. After this update, all ϕ not associated with a mixture component are discarded. The remaining ϕ_c can then be updated via Gibbs sampling or some other Markov chain update that leaves the posterior distribution invariant.

7.4.2 Split–Merge Markov Chain Monte Carlo

Jain and Neal have developed split–merge Metropolis–Hastings procedures for both conjugate and nonconjugate Dirichlet process mixture models (Jain and Neal, 2004, 2007). A model is considered conjugate if G_0 is a conjugate prior for F. A specific type of nonconjugate prior that we will consider is the *conditionally conjugate* family of priors. In conditionally conjugate models, the pair, F and G_0, is conditionally conjugate in one model parameter if the remaining parameters are held fixed. Here, we employ the nonconjugate version of this technique, and assume that F is conditionally conjugate to G_0 in (7.11), so the model parameters, ϕ_{c_i}, cannot be integrated away. The state of the Markov chain consists of the mixture component indicators, c_i, and the model parameters.

The Jain and Neal (2007) sampler offers nonincremental moves that yield significant changes to the allocation of observations to mixture components in a single iteration. A *split* move involves separating a single mixture component into two distinct components, while a *merge* move combines two distinct mixture components together. The split–merge proposal densities are evaluated by a Metropolis–Hastings procedure (Metropolis et al., 1953; Hastings, 1970) in which split or merge proposals are constructed by exploiting properties of a *restricted* Gibbs sampling scan on the component indicators, c_i and model parameters. The Gibbs sampling scan is restricted in that it is only performed on a subset of the data (i.e., observations associated with the merged component that is proposed to be split) and will only allocate observations between two mixture components.

To achieve more reasonable split proposals, several intermediate restricted Gibbs sampling scans are conducted prior to the final restricted Gibbs sampling scan, which is used to calculate the Metropolis–Hastings acceptance probability. The result of the last intermediate Gibbs sampling scan is denoted as the random *launch* state, from which the restricted Gibbs sampling transition probability is explicitly calculated. The number of intermediate restricted Gibbs sampling scans is considered a tuning parameter of this algorithm.

For a merge proposal, there are several ways to combine items in two components to one component, and intermediate restricted Gibbs sampling scans are utilized to obtain an appropriate merge launch state. A description of the steps involved in this algorithm, details to compute the Metropolis–Hastings acceptance probability, and a discussion of the validity of the conjugate version of the split–merge Metropolis–Hastings algorithm are provided in Jain and Neal (2007).

We employ the nonconjugate split–merge procedure as an exploratory technique to sift out potential clusters and subclusters. In the following, we will consider the nonconjugate split–merge procedure as a method to cluster genomics data and will compare this method's performance to Neal's auxiliary Gibbs sampling method (Neal, 2000). We intend to demonstrate that our method can detect patterns in high dimensions that perhaps might be overlooked by standard Markov chain Monte Carlo techniques.

7.5 Leukemia Gene Expression Example

In this section, we describe data from Golub et al. (1999) and apply the Dirichlet process mixture model to cluster gene expression profiles. The performances of two computational methods, Gibbs sampling and split–merge Markov chain Monte Carlo, are compared.

7.5.1 Leukemia Data

The microarray data will consist of an $n \times m$ matrix, where n denotes the experimental condition (e.g., patients) and m represents the number of genes. Typically, $n < m$. If y_{ij} is the expression level of gene j for experimental condition i, then $y_i = (y_{i1}, \ldots, y_{im})$ represents the expression profile for the ith subject. In the mixture model setting, we will assume that each observed patient expression profile arises from an underlying mixture of simple parametric distributions having the form $F(\theta)$. That is, each expression profile arises from an unknown latent class. Expression profiles generated from the same pattern form clusters of similar expression profiles. For microarray data, we will assume that each element of the patient's expression profile arises from a normal distribution and that the elements are independent and identically distributed. The normal mixture model is a realistic choice here because of its flexibility in modeling a number of heterogeneous populations simultaneously and its simplicity in constructing conditional distributions.

The Golub et al. (1999) data compares gene expression in two types of leukemia: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML) for 72 patients. The microarrays used were Affymetrix high-density oligonucleotide arrays consisting of 6,817 human genes. The patients were classified as 47 cases of ALL (38 B-cell ALL and 9 T-cell ALL) and 25 cases of AML. Originally, these data were split into a training and test set, but for our purposes, we combine the two.

Data from microarrays require some processing to yield a normalized matrix of intensity values. Following the exact process described by Dudoit et al. (2002b), three procedures were applied to the data: "(i) thresholding: floor of 100 and ceiling of 16,000; (ii) filtering: exclusion of genes with max/min ≤ 5 and (max $-$ min) ≤ 500, where max and min refer respectively to the maximum and minimum expression levels of a particular gene across mRNA samples; (iii) base 10 logarithmic transformation." The observations were further transformed by standardizing the patients (observations) to have mean 0 and variance 1 across the genes as suggested by Dudoit et al. (2002a), which differs from Golub et al. (1999). According to Yang et al. (2002), a scale adjustment is necessary "to prevent the expression levels in one particular array from dominating the average expression levels across arrays." These transformations produce a $72 \times 3{,}571$ matrix with no missing values. We will consider a random subset of 100 genes in this analysis. We further transformed this data by standardizing the expression levels for each gene to have a mean of 0 and a variance of 1.

Our objective is to classify the patients according to leukemia type by using information revealed by the gene expression data. The goal is to obtain a similar classification based on clinical findings as described by Golub et al. (1999), but perhaps further subclassification of the patients is possible. A countably infinite number of components mixture model is ideal in this situation, since we usually do not know a priori how many clusters are present in the data. Furthermore, in disease classification, such as types of tumors, we may wish to detect subcategories within a particular class of tumors that may arise, for instance, due to differences in gender, molecular structure, or time of onset. Bayesian methods naturally handle genes that may overlap among the components, which allows the detection of different, interacting genetic pathways that contribute to disease.

7.5.2 Implementation

Auxiliary Gibbs sampling is compared to split–merge Markov chain Monte Carlo. For each algorithm, all observations were assigned to the same mixture component for the initial state, and each algorithm was run for 10,000 iterations, but the first 5,000 iterations are shown in the trace plots (i.e., sampled values in the Markov chain vs. iteration number). All simulations were performed on Matlab.

Performance measures that were considered include trace plots over time and computation time per iteration. The trace plots show three values which represent the fractions of observations associated with the most common, two most common, and three most common mixture components.

The split–merge algorithm has four adjustable parameters, in which a specific version of the algorithm is denoted as follows: Split–Merge $(\cdot,\cdot,\cdot,\cdot)$. The first number in parentheses is the number of intermediate Gibbs sampling scans to reach the launch state for the split proposal, the second is the number of Metropolis–Hastings updates in a single iteration, the third is the number of complete incremental sampling scans after the final Metropolis–Hastings update, and the final parameter is the number of intermediate Gibbs sampling scans to reach the launch state for the merge proposal.

7.5.3 Results

The priors were set as follows: The Dirichlet process parameter, α, was set to one for all simulations. The parameters for the priors of the parameters have been set to the same values over all genes as follows: $w = 0$, $B = 1$, $r = 1$, and $R = 5$, where $\mu \sim \text{Normal}(w, B^{-1})$ and $\tau \sim \text{Gamma}(r, R)$. It would be wise to set these parameters using prior knowledge obtained from the oncologists, since it is unlikely that each gene has the same likelihood of being expressed. Alternatively, the model could be

extended to allow the hyperparameters to vary. We provide results for the nonconjugate split–merge procedure, Split–Merge (50,1,1,30), and Gibbs sampling with $v = 3$ auxiliary parameters.

Since the classification described by Golub et al. (1999) is based on preexisting clinical data, we do not expect to obtain an identical classification of tumors. However, we do believe that similarities should be observed. From Figure 7.1, it is evident that Gibbs Sampling is not able to separate the data and leaves all observations in the same mixture component. Gibbs Sampling will take much longer to reach equilibrium as it is stuck in a single component. On the other hand, Split–Merge splits the data into two major clusters. These groups correspond to the clinical classification in that the ALL and AML groups have been separated into distinct clusters. The subgroup ALL-T consisting of nine patients has not been distinguished from the ALL-B group, but that may be simply due to the randomized selection of the 100 genes or unrealistic priors.

Table 7.1 shows a contingency table comparing the clinical classification of the tumors to the two clusters obtained from the split–merge procedure at iteration 9000. McNemar's test is performed, in which groups ALL-T and ALL-B have been combined. The p-value for this test is 0.2373, indicating the difference in distributions between the split–merge clusters and clinical findings is not significant.

It has been noted that the data are merged from a training and test set, where the data were obtained from two different labs at different times. According to Dudoit et al. (2002b), the test set (consisting of 34 patients) is more heterogeneous and includes a broader range of samples that were subjected to different preparation protocols. This may also have some bearing as to why ALL-T is consistently merged with ALL-B.

As a final check, the simulations were repeated by starting the simulation from a typical state of the competing method's equilibrium distribution. Gibbs sampling stayed in the two-component state that it was started from. This suggests that the two-component split–merge state has high posterior probability; that is, the split was not due to any problem with the split–merge procedure.

When the initial state was changed so that each observation was assigned to a different mixture component, the two samplers again produced conflicting results. Figure 7.2 shows that Gibbs sampling is stuck in three components, while split–merge finds two major components. Note that this version of split–merge has been slightly modified by adding 20 restricted Gibbs sampling scans for only the parameters prior to the usual 50 intermediate Gibbs sampling scans when conducting a split proposal. These additional scans are intended to improve the split proposals. By inspection, the classifications obtained by both Gibbs sampling and split–merge seem quite reasonable. Tables 7.2 and 7.3 show contingency tables comparing each sampler's clusters (at iteration 5000) to the clinical classification.

To determine if these configurations are high-probability states from the posterior distribution, each sampler was started from a typical state from the other sampler. Gibbs sampling remains in the three-component configuration as before. However, split–merge quickly moves (within one hundred iterations) from the three-component configuration to two components. This indicates that the two-component

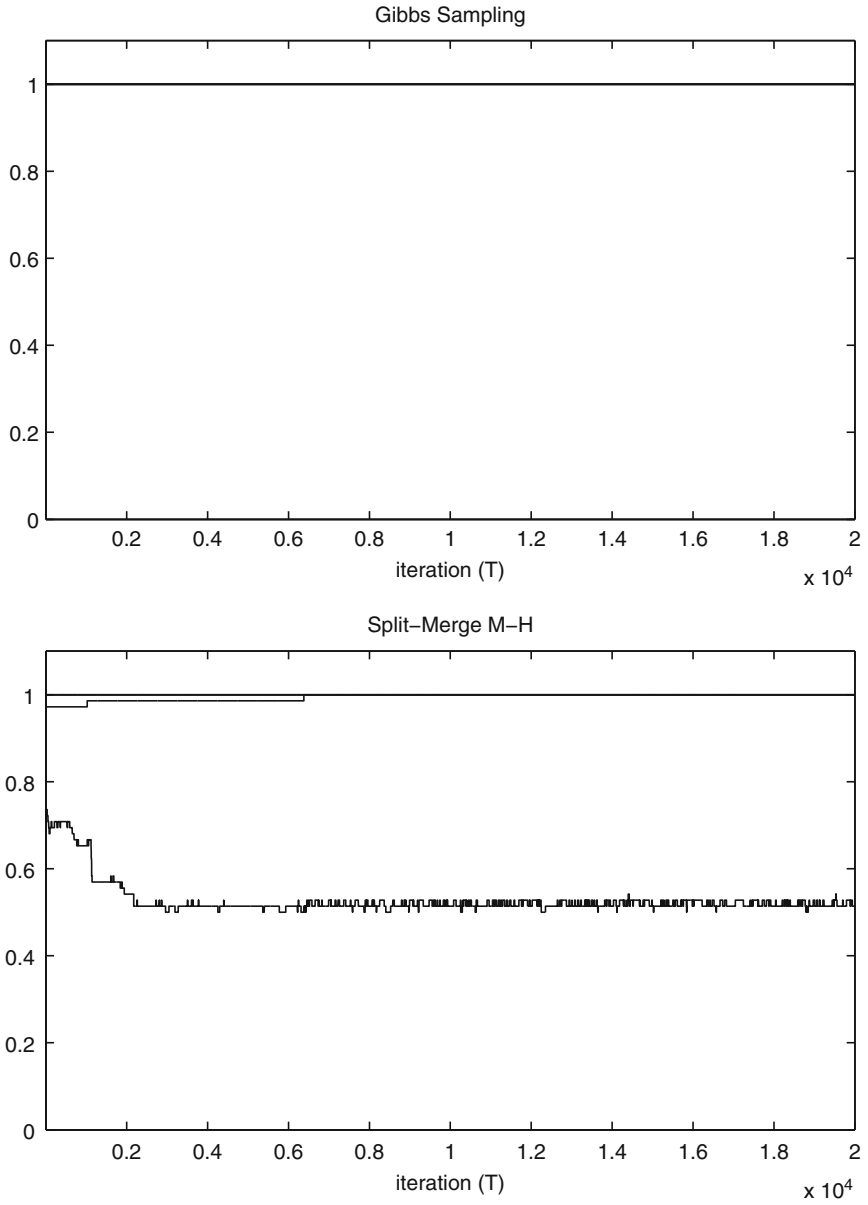


Fig. 7.1 Trace plots comparing Gibbs sampling to split–merge Metropolis–Hastings using leukemia microarray data. The initial state consists of all observations in the same mixture component.

state has high probability given the priors, while the three-component state is quite unlikely. Once again, Gibbs sampling is stuck in an atypical configuration, where it is unable to delete a component.

Table 7.1 2×2 contingency table comparing a clinical classification of tumors to clusters obtained from the split–merge procedure.

Tumor	Split–merge cluster 1	Split–merge cluster 2
AML	1	24
ALL	34	13

From Tables 7.1 and 7.3, it is apparent that the two slightly different split–merge techniques with different initialization states produce different clusterings. To determine if the additional scans improve the split–merge technique's convergence, each split–merge procedure was initialized from a typical state from the other method. Results indicate that there is no obvious improvement by the addition of extra Gibbs sampling scans for the parameters only. Each split–merge sampler remained in its initialization state for approximately 2,000 iterations. While the two different states obtained by the two split–merge procedures are high-probability configurations, it is not clear that either split–merge technique has indeed converged to the correct equilibrium distribution yet. Perhaps additional restricted Gibbs sampling scans for the split proposals are necessary to improve mixing between these high-probability states.

Since the analysis is performed under a Bayesian paradigm, it would be beneficial to obtain more guidance or perform sensitivity analysis regarding the choice of priors in these situations. Alternatively, when no prior information is available, additional stages may be added to this hierarchical model to estimate priors. The priors chosen in this chapter were selected so that the range of data would be covered with little prior opinion, and we assumed that each gene had the same prior distribution (which is likely untrue). It seems natural to choose a prior that would favor genes known to be important.

7.6 Discussion

In conclusion, this chapter provides a brief overview of some parametric and nonparametric Bayesian approaches to high-dimensional gene expression analysis. We briefly reviewed EBArrays (Newton et al., 2004), POE (Garrett and Parmigiani, 2003), and infinite Bayesian mixture models (Medvedovic and Sivaganesan, 2002), and focused on a specific nonparametric Bayesian model, the Dirichlet process mixture model, to analyze patients' gene expression profiles via split–merge Markov chain Monte Carlo. The Bayesian paradigm provides a flexible framework for high-dimensional statistical inference in the presence of strongly correlated data.

In particular, the split–merge techniques seem to be a promising tool that should be added to the statistical genomics repository. As an exploratory tool, it is useful in detecting clusters that other incremental Markov chain methods may not. Gibbs sampling is a popular Markov chain sampling scheme, since the full conditional

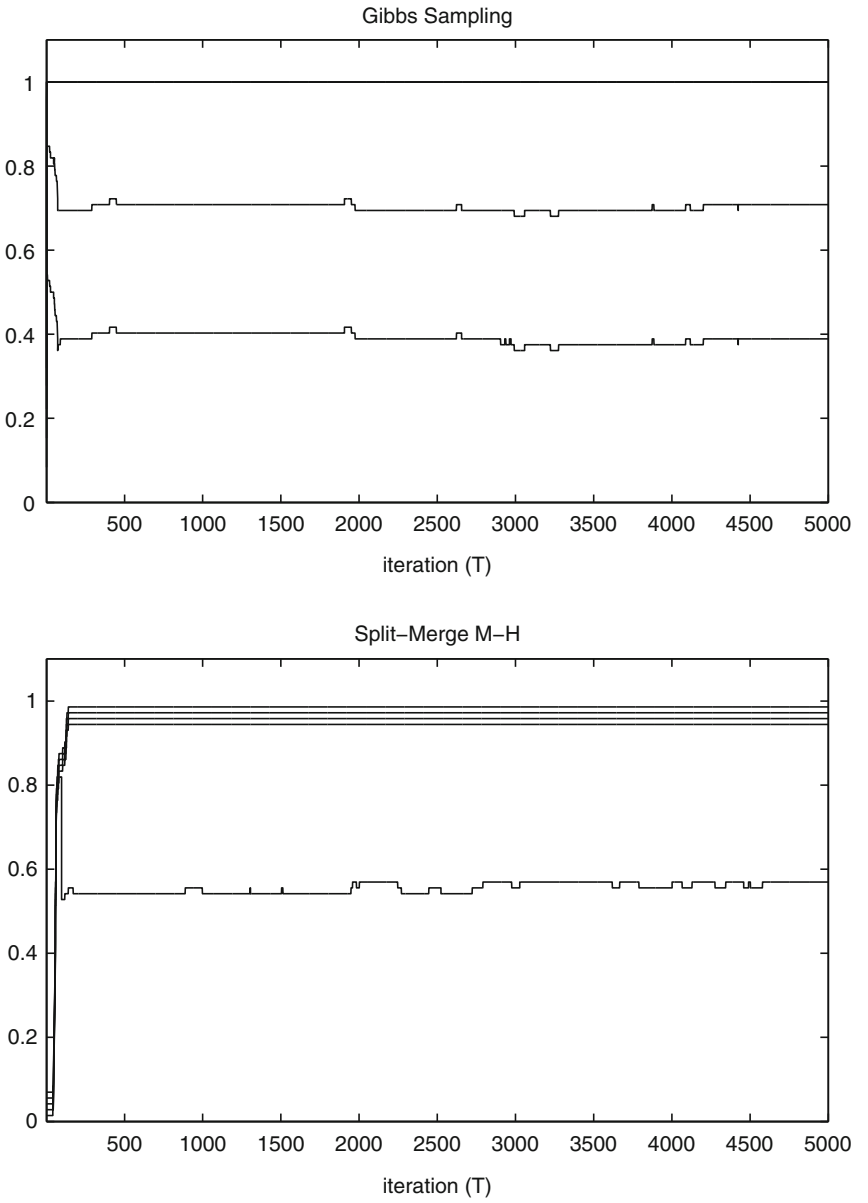


Fig. 7.2 Trace plots comparing Gibbs sampling to split–merge Metropolis–Hastings using leukemia microarray data. The initial state consists of each obervation in a different mixture component.

Table 7.2 Contingency table comparing a clinical classification of tumors to clusters obtained from the Gibbs sampling (GS) procedure.

Tumor	GS cluster 1	GS cluster 2	GS cluster 3
AML	23	2	0
ALL-B	0	17	21
ALL-T	0	2	7

Table 7.3 Contingency table comparing a clinical classification of tumors to clusters obtained from the split–merge procedure.

Tumour	Split–merge cluster 1	Split–merge cluster 2	Other split–merge clusters
AML	2	23	0
ALL	39	4	4

distributions are often simple to compute. Because the split–merge techniques employ Gibbs sampling to obtain an appropriate proposal density, it is actually quite straightforward to construct the split–merge techniques too. When clusters overlap in high dimensions, Gibbs sampling is not always a feasible solution. Since split–merge techniques are constructed to frequently split and merge components, these methods are recommended, even though they can be quite computationally intensive.

References

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2:1152–1174.

Beaumont, M. and Rannala, B. (2004). The Bayesian revolution in genetics. *Nature Reviews*, 5:251–261.

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via Pólya urn schemes. *Annals of Statistics*, 1:353–355.

Do, K.-A., Müller, P., and Vannucci, M. E. (2006). *Bayesian Inference for Gene Expression and Proteomics*. Cambridge University Press.

Dudoit, S., Fridlyand, J., and Speed, T. P. (2002a). Comparison of discrimination methods for the classification of tumours using gene expression data. *Journal of the American Statistical Association*, 97:77–87.

Dudoit, S., Yang, Y. H., Speed, T. P., and Callow, M. J. (2002b). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*, 12:111–139.

Eisen, M. B., Spellman, P., Brown, P., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Science, USA*, 95:14863–14868.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. In Rizvi, H. and Rustagi, J., editors, *Recent Advances in Statistics*, pp. 287–303. Academic Press.

Garrett, E. and Parmigiani, G. (2003). POE: statistical methods for qualitative analysis of gene expression. In Parmigiani, G., Garrett, E. S., Irizarry, R. A., and Zeger, S. L., editors, *The Analysis of Gene Expression Data: Methods and Software*, pp. 362–387. Springer.

Garrett-Mayer, E. and Scharpf, R. (2006). Models for probability of under- and overexpression: the POE scale. In Do, K.-A., Müller, P., and Vannucci, M., editors, *Bayesian Inference for Gene Expression and Proteomics*, pp. 137–154. Cambridge University Press.

Gelfand, A. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85:398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Gilks, W., Richardson, S., and Spiegelhalter, D. J., editors (1996). *Markov Chain Monte Carlo in Practice*. Chapman and Hall.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P.and Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537.

Griffiths, A. J. F., Miller, J. H., Suzuki, T., D., Lewontin, R. C., and Gelbart, W. M. (1996). *An Introduction to Genetic Analysis*. W. H. Freeman and Company, 6th edition.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.

Jain, S. (2002). *Split-Merge Techniques for Bayesian Mixture Models*. unpublished Ph.D. dissertation at University of Toronto.

Jain, S. and Neal, R. M. (2004). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182.

Jain, S. and Neal, R. M. (2007). Splitting and merging components of a nonconjugate Dirichlet process mixture model (with discussion). *Bayesian Analysis*, 2:445–472.

Kendziorski, C. M., Newton, M. A., Lan, H., and Gould, M. N. (2003). On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, 22:3899–3914.

MacEachern, S. N. and Müller, P. (1998). Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics*, 7:223–238.

Medvedovic, M. and Sivaganesan, S. (2002). Bayesian infinite mixture model-based clustering of gene expression profiles. *Bioinformatics*, 18:1194–1206.

Medvedovic, M., Yeung, K. Y., and Bumgarner, R. E. (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics*, 20:1222–1232.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265.

Newton, M. A. and Kendziorski, C. (2003). Parametric empirical Bayes methods for microarrays. In Parmigiani, G., Garrett, E. S., Irizarry, R. A., and Zeger, S. L., editors, *The Analysis of Gene Expression Data: Methods and Software*, pp. 254–271. Springer.

Newton, M. A., Kendziorski, C. M., Richmond, C. S., Blattner, F. R., and Tsui, K. W. (2001). On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology*, 8:37–52.

Newton, M. A., Noueiry, A., Sarkar, D., and Ahlquist, P. (2004). Detecting differential gene expression with a semiparametric hierarchical mixture model. *Biostatistics*, 5:155–176.

Newton, M. A., Wang, P., and Kendziorski, C. (2006). Hierarchical mixture models for expression profiles. In Do, K.-A., Müller, P., and Vannucci, M., editors, *Bayesian Inference for Gene Expression and Proteomics*, pp. 40–52. Cambridge University Press.

Parmigiani, G., Garrett, E. S., Anbazhagan, R., and Gabrielson, E. (2002). A statistical framework for expression-based molecular classification in cancer. *Journal of the Royal Statistical Society, Series B*, 64:717–736.

Schena, M. (1999). *DNA Microarrays: A Practical Approach*. Oxford University Press.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics*, 22:1701–1762.

West, M., Müller, P., and Escobar, M. D. (1994). Hierarchical priors and mixture models, with application in regression and density estimation. In Freeman, P. R. and Smith, A. F. M., editors, *Aspects of Uncertainty*, pp. 363–386. John Wiley & Sons.

Yang, Y. H., Buckley, M. J., Dudoit, S., and Speed, T. P. (2002). Comparison of methods for image analysis on cDNA microarray data. *Journal of Computational and Graphical Statistics*, 11:108–136.

Yeung, K. Y., Medvedovic, M., and Bumgarner, R. E. (2003). Clustering gene-expression data with repeated measurements. *Genome Biology*, 4:R34 (Epub).

Index

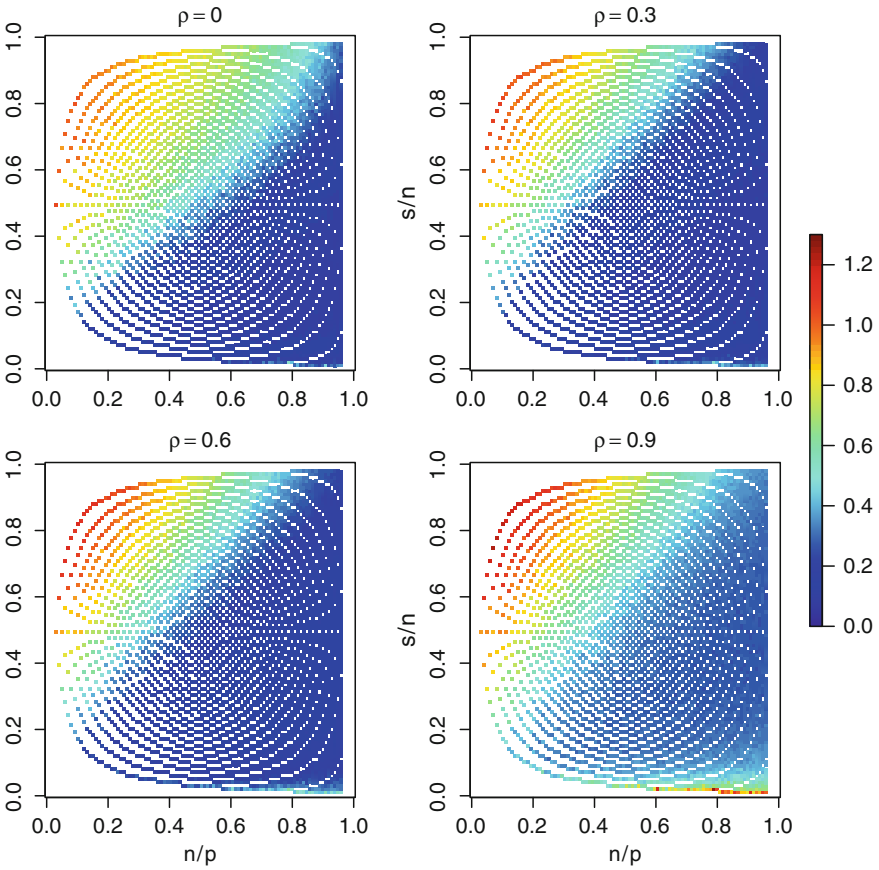


Fig. 2.1 Phase transition diagram with the sparse model recovered by LASSO with the tuning parameter selected by AIC. The number of variables is kept constant at $p = 100$. Columns of $\mathbf{X}$ exhibit compound symmetry correlation structure with $\rho = 0, .3, .6$, and .9.

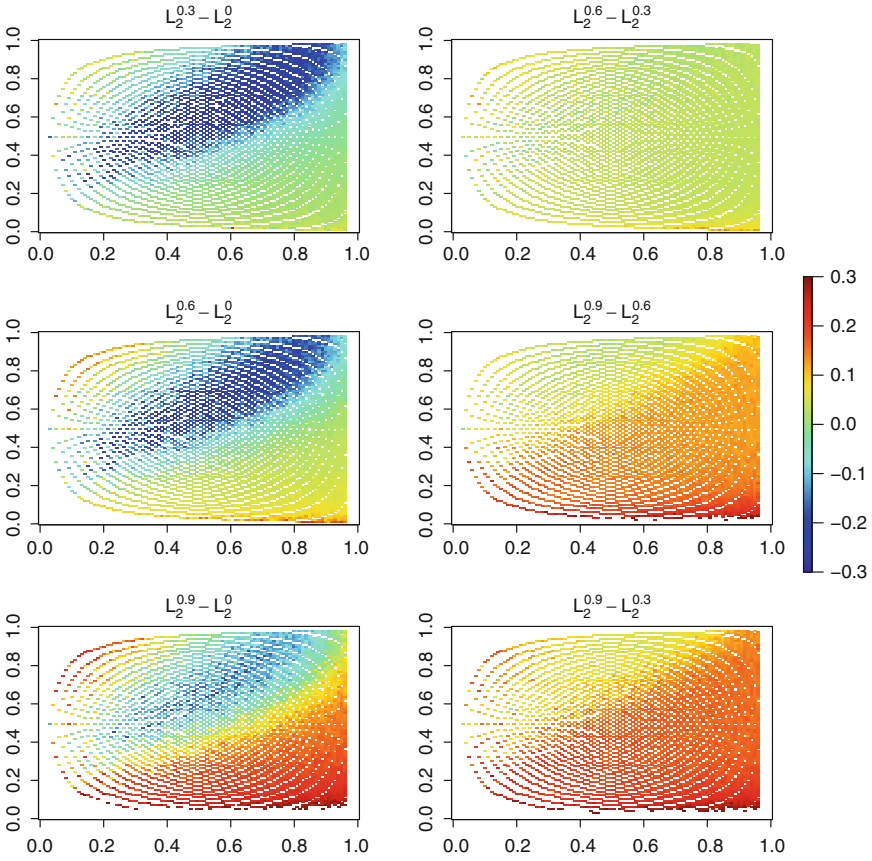


Fig. 2.2 Differences between the scaled estimation bias for the models estimated with LASSO, and the design matrix $\mathbf{X}$ exhibiting compound symmetry correlation structure with $\rho = 0, .3, .6,$ and .9. The notation L_2^{ρ} indicates the the scaled bias L_2 with the correlation ρ.

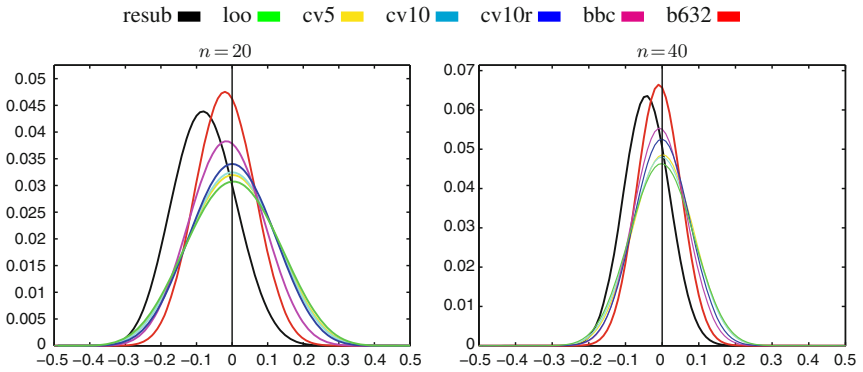


Fig. 4.2 Beta-fits of empirical deviation distribution. resub: "apparent" estimate; loo: leave-one-out; cv10r: repeated CV10; bbc: bias-corrected bootstrap; b632: .632 bootstrap.

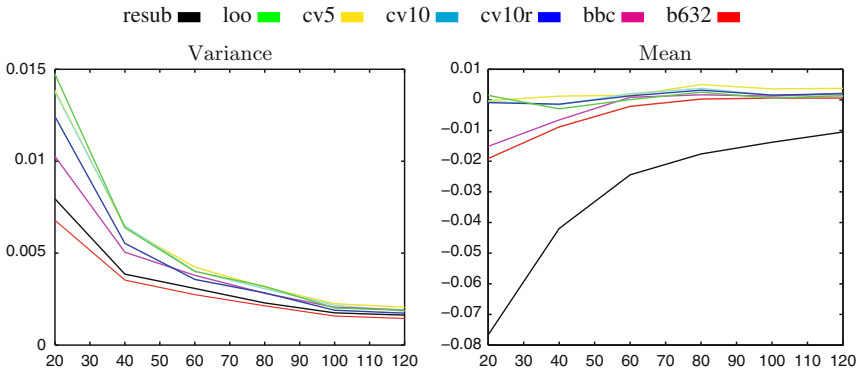


Fig. 4.3 Plots of the empirical deviation distribution.

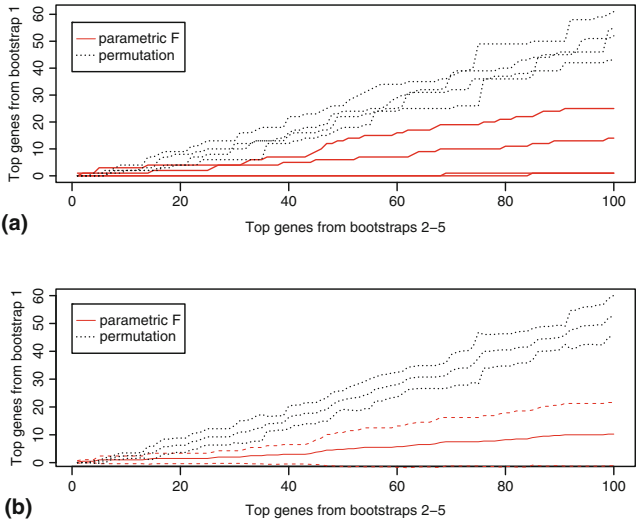


Fig. 4.4 Rediscovery rates estimated using bootstrap: **(a)** 5 individual runs; **(b)** estimated rediscovery rates and their 95% confidence intervals.

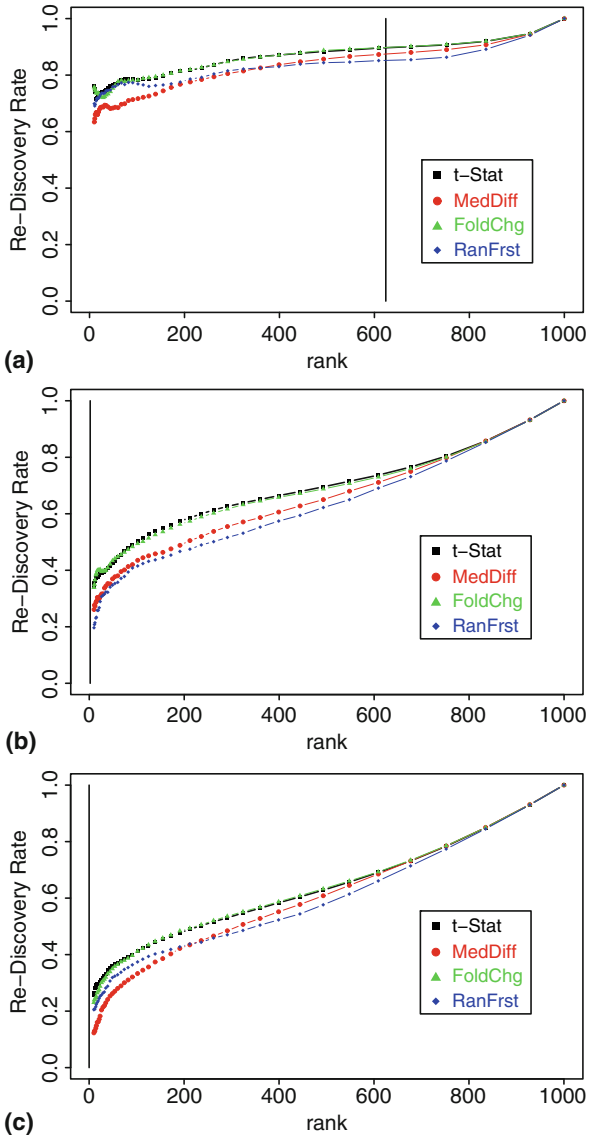


Fig. 4.5 RDCurve of gene selection associated with ER status and lymph node metastasis status, and a random data set. **(a)** RDCurve of gene selection for ER status; **(b)** RDCurve of gene selection for lymph metastasis status; **(c)** RDCurve of gene selection from noninformative data set. The vertical lines correspond to the number of genes selected with FDR < 0.05.