The University of Waikato
Te Whare Wānanga o Waikato

A New Approach to Fitting Linear Models in High Dimensional Spaces

Yong Wang

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science at The University of Waikato.

November 2000

Abstract

This thesis presents a new approach to fitting linear models, called "pace regression", which also overcomes the dimensionality determination problem. Its optimality in minimizing the expected prediction loss is theoretically established, when the number of free parameters is infinitely large. In this sense, pace regression outperforms existing procedures for fitting linear models. Dimensionality determination, a special case of fitting linear models, turns out to be a natural by-product. A range of simulation studies are conducted; the results support the theoretical analysis.

Through the thesis, a deeper understanding is gained of the problem of fitting linear models. Many key issues are discussed. Existing procedures, namely OLS, AIC, BIC, RIC, CIC, CV(d), BS(m), RIDGE, NN-GAROTTE and LASSO, are reviewed and compared, both theoretically and empirically, with the new methods.

Estimating a mixing distribution is an indispensable part of pace regression. A measure-based minimum distance approach, including probability measures and nonnegative measures, is proposed, and strongly consistent estimators are produced. Of all minimum distance methods for estimating a mixing distribution, only the nonnegative-measure-based one solves the minority cluster problem, what is vital for pace regression.

Pace regression has striking advantages over existing techniques for fitting linear models. It also has more general implications for empirical modeling, which are discussed in the thesis.

Acknowledgements

First of all, I would like to express my sincerest gratitude to my supervisor, Ian Witten, who has provided a variety of valuable inspiration, assistance, and suggestions throughout the period of this project. His detailed constructive comments, both technically and linguistically, on this thesis (and lots of other things) are very much appreciated. I have benefited enormously from his erudition and enthusiasm in various ways in my research. He carefully considered my financial situations, often before I have done so myself, and, offered research and teaching assistant work to help me out; thus I was able to study without any financial worry.

I would also like to thank my co-supervisor, Geoff Holmes, for his productive suggestions to improve my research and its presentation. As the chief of the machine learning project Weka, he financially supported my attendance at several international conferences. His kindness and encouragement throughout the last few years are much valued.

I owe a particular debt of gratitude to Alastair Scott (Department of Statistics, University of Auckland), who became my co-supervisor in the final year of the degree, because of the substantial statistics component of the work. His kindness, expertise, and timely guidance helped me tremendously. He provided valuable comments on almost every technical point, as well as the writing in the thesis; encouraged me to investigate a few further problems; and made it possible for me to use computers in his department and hence to implement pace regression in S-PLUS/R.

Thanks also go to Ray Littler and Thomas Yee, who spent their precious time dis-

cussing pace regression with me and offered illuminating suggestions. Kai Ming Ting introduced me to the M5 numeric predictor, the re-implementation of which was taken as the initial step of the PhD project and eventually led me to become interested in the dimensionality determination problem.

Mark Apperley, the head of the Department of Computer Science, University of Waikato, helped me financially, along with many other things, by offering a teaching assistantship, a part-time lecturership, funding for attending international conferences, and funding for visiting the University of Calgary for six months (an arrangement made by Ian Witten). I am also grateful to the Waikato Scholarships Office for offering the three-year University of Waikato Full Scholarship.

I have also received various help from many others in the Department of Computer Science, in particular, Eibe Frank, Mark Hall, Stuart Inglis, Steve Jones, Masood Masoodian, Gordon Paynter, Bernhard Pfahringer, Steve Reeves, Phil Treweek, Len Trigg, and David McWha. It was so delightful playing soccer with these people.

I would like to express my appreciation to many people in the Department of Statistics, University of Auckland, in particular, Paul Copertwait, Brian McArdle, Renate Meyer, Russell Millar, Geoff Pritchard, and George Seber, for allowing me to sit in on their courses, providing lecture notes, and marking my assignments and tests, which helped me to learn how to apply statistical knowledge and express it gracefully. Thanks are also due to Jenni Holden, John Huakau, Ross Ihaka, Alan Lee, Chris Triggs, and others for their hospitality.

Finally, and most of all, I am grateful to my family and extended family for their tremendous tolerance and support—in particular, my wife, Cecilia, and my children, Campbell and Michelle. It is a pity that I often have to leave them, although staying with them is so enjoyable. The time-consuming nature of this work looks so fascinating to Campbell and Michelle that they plan to go to their daddy's university first, instead of a primary school, after they leave their kindergarden.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Empirical modeling builds models from data, as opposed to *analytical modeling* which derives models from mathematical analysis on the basis of first principles, background knowledge, reasonable assumptions, etc. One major difficulty in empirical modeling is the handling of the noise embedded in the data. By *noise* here we mean the uncertainty factors, which change from time to time and are usually best described by probabilistic models. An empirical modeling procedure generally utilizes a model prototype with some adjustable, free parameters and employs an optimization criterion to find the values of these parameters in an attempt to minimize the effect of noise. In practice, "optimization" usually refers to the best explanation of the data, where the explanation is quantified in some mathematical measure; for example, least squares, least absolute deviation, maximum likelihood, maximum entropy, minimum cost, etc.

The number of free parameters, more precisely, the *degrees of freedom*, plays an important role in empirical modeling. One well-known phenomenon concerning these best data explanation criteria is that any given data can be increasingly explained as the number of free parameters increases—in the extreme case when this number is equal to the number of observations, the data is fully explained. This phenomenon

is intuitive, for any effect can be explained given enough reasons. Nevertheless, the model that fully explains the data in this way usually has little predictive power on future observations, because it explains the noise so much that the underlying relationship is obscured. Instead, a model that includes fewer parameters may perform significantly better in prediction.

While free parameters are inevitably required by the model prototype, it seems natural to ask the question: how many parameters should enter the final estimated model? This is a classic issue, formally known as the *dimensionality determination problem* (or the *dimensionality reduction*, *model selection*, *subset selection*, *variable selection*, etc. *problem*). It was first brought to wide awareness by H. Akaike (1969) in the context of time series analysis. Since then, it has drawn research attention from many scientific communities, in particular, statistics and computer science (see Chapter 2).

Dimensionality determination is a key issue in empirical modeling. Although most research interest in this problem focuses on fitting linear models, it is widely acknowledged that the problem plagues almost all types of model structure in empirical modeling.[1] It is inevitably encountered when building various types of model, such as generalized additive models (Hastie and Tibshirani, 1990), tree-based models (Breiman et al., 1984; Quinlan, 1993), projection pursuit (Friedman and Stuetzle, 1981), multiple adaptive regression splines (Friedman, 1991), local weighted models (Cleveland, 1979; Fan and Gijbels, 1996; Atkeson et al., 1997; Loader, 1999), rule-based learning (Quinlan, 1993), and instance-based learning (Li, 1987; Aha et al., 1991), to name a few. There is a large literature devoted to the issue of dimensionality determination. Many procedures have been proposed to address the problem—most of them applicable to both linear and other types of model structure (see Chapter 2). Monographs that address the general topics of dimensionality determination, in particular for linear regression, include Linhart and

[1] It does not seem to be a problem for the estimation of a mixing distribution (see Chapter 5), since larger set of candidate support points does not jeopardize, if not improve, the estimation of the mixing distribution. This observation, however, does not generalize to every kind of unsupervised learning. For example, the loss function employed in a clustering application does take the number of clusters into account for, say, computational reasons.

Zucchini (1989), Rissanen (1989), Miller (1990), Burnham and Anderson (1998), and McQuarrie and Tsai (1998).

So far this problem is often considered as an independent issue of modeling principles. Unfortunately, it turns out to be so fundamental in empirical modeling that it casts a shadow over the general validity of many existing modeling principles, such as the least squares principle, the (generalized) maximum likelihood principle, etc. This is because it takes but one counter-example to refute the correctness of a generalization, and all these principles fail to solve the dimensionality determination problem. We believe its solution relates closely to fundamental issues of modeling principles, which have been the subject of controversy throughout the history of theoretical statistics, without any consensus being reached.

R. A. Fisher (1922) identifies three problems that arise in data reduction: (1) model specification, (2) parameter estimation in the specified model, and (3) the distribution of the estimates. In much of the literature, dimensionality determination is considered as belonging to the first type of problem, while modeling principles belong to the second. However, the considerations in the preceding paragraph imply that dimensionality determination belongs to the second problem type. To make this point more explicit, we modify Fisher's first two problems slightly, into (1) specifying the candidate model space, and (2) determining the optimal model from the model space. This modification is consistent with modern statistical decision theory (Wald, 1950; Berger, 1985), and will be taken as the starting point of our work.

This thesis proposes a new approach to solving the dimensionality determination problem, which turns out to be a special case of *general modeling*, i.e., the determination of the values of free parameters. The key idea is to consider the uncertainty of estimated models (introduced in Section 1.3), which results in a methodology similar to empirical Bayes (briefly reviewed in Section 2.8). The issue of modeling principles is briefly covered in Section 1.2. Two principles will be employed, which naturally lead to a solution to the dimensionality determination problem, or more generally, fitting linear models. We will theoretically establish optimality in the sense of k-asymptotics (i.e., an infinite number of free parameters) for the proposed

procedures. This implies the tendency of increasing performance as k increases and justifies the appropriateness of our procedures being applied in high-dimensional spaces. Like many other approaches, we focus for simplicity on fitting linear models with normally distributed noise, but the ideas can probably be generalized to many other situations.

1.2 Modeling principles

In the preceding section, we discussed the optimization criterion used in empirical modeling. This relates to the fundamental question of modeling: among the candidate models, which one is the best? This is a difficult and controversial issue, and there is a large literature devoted to it; see, for example, Berger (1985) and Brown (1990) and the references cited therein.

In our work, two separate but complementary principles are adopted for successful modeling. The first one follows the utility theory in decision analysis (von Neumann and Morgenstern, 1944; Wald, 1950; Berger, 1985), where *utility* is a mathematical measure of the decision-maker's satisfaction. The principle for best decision-making, including model estimation, is to maximize the expected utility. Since we are more concerned with loss—the negative utility—we rephrase this into the fundamental principle for empirical modeling as follows.

The Minimum Expected Loss Principle. Choose the model from the candidate model space with smallest expected loss in predicting future observations.

To minimize the expected loss over future observations, we often need to make assumptions so that future observations can be related to the observations which are used for model construction. By assumptions we mean statistical distribution, background knowledge, loss information, etc. Of course, the performance of the resulting model over future observations also depends on how far these assumptions are from future reality.

Further, we define that a *better model* implies smaller expected loss and *equivalent models* have equal expected loss. This principle is then applied to find the *best model* among all candidates.

It may seem that the principle does not address the issue of model complexity (e.g., the number of parameters), which is important in many practical applications. This is in fact not true, because the definition of the loss could also include loss incurred due to model complexity—for example, the loss for comprehension and computation. Thus model complexity can be naturally taken into account when minimizing the expected loss. In practice, however, this kind of loss is often difficult to quantify appropriately, and even more difficult to manipulate mathematically. When this is the case, a second parsimony principle can become helpful.

The Simplicity Principle. Of the equivalent models, choose the simplest.

We will apply this principle to *approximately* equivalent models, i.e., when the difference between the expected losses of the models in consideration is tolerable. It is worth pointing out that, though similar, this principle differs from the generally understood Occam's razor, in that the former considers equal expected losses, while the latter equal data explanation. We believe that it is inappropriate to compare models based on data explanation, and return to this in the next section.

"Tolerable" here is meant only in the sense of statistical significance. In practice, model complexity may be further reduced when other factors, such as data precision, prediction significance and cost of model construction, are considered. These issues, however, are less technical and will not be used in our work.

Although the above principles may seem simple and clear, their application is complicated in many situations by the fact that the distribution of the true model is usually unknown, and hence the expected loss is uncomputable. Partly due to this reason, a number of modeling principles exist which are widely used and extensively investigated. Some of the most important ones are the least squares principle (Legendre, 1805; Gauss, 1809), the maximum likelihood principle (Fisher, 1922, 1925;

Barnard, 1949; Birnbaum, 1962), the maximum entropy principle (Shannon, 1948; Kullback and Leibler, 1951; Kullback, 1968), the minimax principle (von Neuman, 1928; von Neumann and Morgenstern, 1944; Wald, 1939, 1950), the Bayes principle (Bayes, 1763), the minimum description length (MDL) principle (Rissanen, 1978, 1983), Epicurus's principle of multiple explanations, and Occam's razor. Instead of discussing these principles here, we examine a few major ones retrospectively in Section 4.5, after our work is presented.

In contrast to the above analytical approach, model evaluation can also be undertaken empirically, i.e., by data. The data used for evaluation can be obtained from another independent sample or from data resampling (for the latter, see Efron and Tibshirani, 1993; Shao and Tu, 1995). While empirical evaluation has found wide application in practice, it is more an important aid in evaluating models than a general modeling principle. One significant limitation is that results obtained from experimenting with data rely heavily on the experiment design and hence are inconclusive. Further, they are often computationally expensive and only apply to finite candidates. Almost all major existing data resampling techniques are still under intensive theoretic investigation—and some theoretic results in fitting linear models have already revealed their limitations; see Section 2.6.

1.3 The uncertainty of estimated models

It is known that in empirical modeling, the noise embedded in data is propagated into the estimated models, causing *estimation uncertainty*. Fisher's third problem concerns this kind of uncertainty. An example is the estimation of confidence interval. The common approach to decreasing the uncertainty of an estimated model is to increase the number of observations, as suggested by the Central Limit Theorem and the Laws of Large Numbers.

Traditionally, as mentioned above, a modeling procedure outputs the model that best explains the data among the candidate models. Since each estimated model is

uncertain to some extent, the model that optimally explains the data is the winner in a competition based on the combined effect of both the true predictive power and the uncertainty effect. If there are many independent candidate models in the competition, the "optimum" can be significantly influenced by the uncertainty effect and hence the optimal data explanation may be good only by chance. An extreme situation is that when all candidates have no predictive power—i.e., only the uncertainty effect is explained. Then, the more the candidate explains the data, the farther it is away from the true model and hence the worse it performs over future observations. In this extreme case, the expected loss of the winner increases without bound as the number of candidates increases—a conclusion that can be easily established from order statistics. The competition phenomenon is also investigated by Miller (1990), where he defines the *selection* or *competition bias* of the selected model. This bias generally increases with the number of candidate models.

This implies that a modeling procedure which is solely based on the maximization of data explanation is not appropriate. To take an extreme case, it could always find the existence of "patterns" in a completely random sequence, provided only that enough number of candidate models are used. This kind of modeling practice is known in unsavory terms as "data dredging", "fishing expeditions", "data mining" (in the old sense[2]), and "torturing the data until they confess" (Miller, 1990, p.12).

An appropriate modeling procedure, therefore, should take the uncertainty effect into consideration. It would be ideal if this effect could be separated from the model's data explanation, so that the comparison between models is based on their predictive power only. Inevitably, without knowing the true model, it is impossible to filter the uncertainty effect out of the observed performance completely. Nevertheless, since we know that capable participants tend to perform well and incapable ones tend to perform badly, we should be able to gain some information about the competition from the observed performance of all participants, use this information to generate a better estimate of the predictive power of each model, and even update the model estimate to a better one by stripping off the expected uncertainty effect.

[2] *Data mining* now denotes "the extraction of previously unknown information from databases that may be large, noisy and have missing data" (Chatfield, 1995).

Eventually, the model selection or, more generally, the modeling task can be based on the estimated predictive power, not simply the data explanation. In short, the modeling procedure relies on not only random data, but also *random models*.

This methodological consideration forms the basis of our new modeling approach. The practical situation may be so difficult that no general mathematical tools are available for providing solutions. For example, there may be many candidate models, generated by different types of modeling procedure, which are not statistically independent at all. The competition can be very hard to analyze.

In this thesis, this idea is applied to solving the dimensionality determination problem in fitting *linear* models, which seems a less difficult problem than the above general formulation. To be more precise, it is applied to fitting linear models in the general sense, which in fact subsumes the dimensionality determination problem. We shall transform the full model that contains all parameters into an orthogonal space to make the dimensional models independent. The observed dimensional models serve to provide an estimation of the overall distribution of the true dimensional models, which in turn helps to update each observed dimensional model to a better estimate, in expectation, of the true dimensional model. The output model is re-generated by an inverse transformation.

Although started from a different viewpoint, i.e., that of competing models, our approach turns out to be very similar to the empirical Bayes methodology pioneered by Robbins (1951, 1955, 1964, 1983); see Section 2.8.

1.4 Contributions

Although the major ideas in this thesis could be generalized to other types of model structure, we focus on fitting linear models, with a strong emphasis on the dimensionality determination problem, which, as we will demonstrate, turns out to be a special case of modeling in the general sense.

As we have pointed out, the problem under investigation relates closely to the fundamental principles of modeling. Therefore, despite the fact that we would like to focus simply on solving mathematical problems, arguments in some philosophical sense seem inevitable. The danger is that they may sound too "philosophical" to be sensible.

The main contributions of the work are as follows.

The competition viewpoint. Throughout the thesis, the competition viewpoint is adopted systematically. This not only provides the key to fitting linear models quantitatively, including a solution to the dimensionality determination problem, but also helps to answer, at least qualitatively, some important questions in the general case of empirical modeling.

The phenomenon of competing models for fitting linear models have been investigated by other authors; for example, Miller (1990), Donoho and Johnstone (1994), Foster and George (1994), and Tibshirani and Knight (1997). However, as shown later, these existing methods can still fail in some particular cases. Furthermore, none of them achieves k-asymptotic optimality generally, a property enjoyed by our new procedures.

The competition phenomenon pervades in various contexts of empirical modeling; for example, in orthogonalization selection (see Section 4.6).

Predictive objective of modeling. Throughout the work, we emphasize the performance of constructed models over future observations, by explicitly incorporating it into the minimum expected loss principle (Section 1.2) and relating it analytically to our modeling procedures (see, e.g., Sections 2.2 and 3.6). Although this goal is often used in simulation studies of modeling procedures in the literature, it is rarely addressed analytically, which seems ironical. Many existing modeling procedures do not directly examine this relationship, although it is acknowledged that they are derived from different considerations.

Employing the predictive objective emphasizes that it is reality that governs theory, not vice versa. The determination of loss function according to specific applications is another example.

In Section 5.6, we use this same objective for evaluating the accuracy of clustering procedures.

Pace regression. A new approach to fitting linear models, *pace regression*, is proposed, based on considering competing models, where "pace" stands for "*Projection Adjustment by Contribution Estimation.*" Six related procedures are developed, denoted PACE_1 to PACE_6, that share a common fundamental idea—estimating the distribution of the effects of variables from the data and using this to improve modeling. The optimality of these procedures, in the sense of k-asymptotics, is theoretically established. The first four procedures utilize OLS subset selection, and outperform existing OLS methods for subset selection, including OLS itself (by which we mean the OLS fitted model that includes all parameters). By abandoning the idea of selection, PACE_5 achieves the highest prediction accuracy of all. It even surpasses the OLS estimate when all variables have large effects on the outcome—in some cases by a substantial margin. Unfortunately, the extensive numerical calculations that PACE_5 requires may limit its application in practice. However, PACE_6 is a very good approximation to PACE_5, and is computationally efficient.

For deriving pace regression procedures, some new terms, such as the "contribution" of a model and also the H- and h-functions, are introduced. They are able to indicate whether or not a model, or a dimensional model, contributes, in expectation, to a prediction of future observations, and are employed to obtain the pace regression procedures.

We realize that this approach is similar to the empirical Bayes methodology (see Section 2.8), which, according to the author's knowledge, has not been such used for fitting linear models (as well as other model structures), despite the fact that empirical Bayes has long been recognized as a breakthrough in statistical infer-

ence (see, e.g., Neyman, 1962; Maritz and Lwin, 1989; Kotz and Johnson, 1992; Efron, 1998; Carlin and Louis, 2000).

Review and comparisons. The thesis also provides in Chapter 2 a general review that covers a wide range of key issues in fitting linear models. Comparisons of the new procedures with existing ones are placed appropriately throughout the thesis.

Throughout the review, comparisons and some theoretic work, we also attempt to provide a unified view for most approaches to fitting linear models, such as OLS, OLS subset selection, model selection based on data resampling, shrinkage estimation, empirical Bayes, etc. Further, our work in fitting linear models seems to imply that supervised learning requires unsupervised learning—more precisely, the estimation of a mixing distribution—as a fundamental step for handling competition phenomenon.

Simulation studies. Simulation studies are conducted in Chapter 6. The experiments range from artificial to practical datasets, illustration to application, high to low noise level, large to small number of variables, independent to correlated variables, etc. The procedures used in the experiments include OLS, AIC, BIC, RIC, CIC, PACE_2, PACE_4, PACE_6, LASSO and NN-GAROTTE.

Experimental results generally support our theoretic conclusions.

Efficient, reliable and consistent estimation of a mixing distribution. Pace regression requires an estimator of the mixing distribution of arbitrary mixture distribution. In Chapter 5, a new minimum distance approach based on a nonnegative measure is shown to provide efficient, reliable and consistent estimation of the mixing distribution. The existing minimum distance methods are efficient and consistent, yet not reliable, due to the minority cluster problem (see Section 5.5), while the maximum likelihood methods are reliable and consistent, yet not efficient in comparison with the minimum distance approach.

Reliable estimation is a finite sample issue but is vital in pace regression, since the misclustering of even a single isolated point could result in unbounded loss in prediction. Efficiency may also be desired in practice, especially when it is necessary to fit linear models repeatedly. Strong consistency of each new estimator is established.

We note that the estimation of a mixing distribution is an independent topic with its own value and has a large literature.

Implementation. Pace regression, together with some other ideas in the thesis, are implemented in the programming languages of S-PLUS (tested with versions 3.4 and 5.1)—in a compatible way with R (version 1.1.1)—and FORTRAN 77. The source code is given in the Appendix, and also serves to provide full details of the ideas in the thesis. Help files in S format have been written for the most important functions.

All the source code and help files are freely available from the author, and could be redistributed and/or modified under the terms of version 2 of the GNU General Public License as published by the Free Software Foundation.

1.5 Outline

The thesis consists of seven chapters and one appendix.

Chapter 1 briefly introduces the general idea of the whole work. It defines the dimensionality determination problem, and points outs that a shadow is cast on the validity of many empirical modeling principles by their failure to solve this problem. Our approach to this problem adopts the viewpoint of competing models, due to estimation uncertainty, and utilizes two complementary modeling principles: minimum expected loss, and simplicity.

Some important issues in fitting linear models are reviewed in Chapter 2, in an at-

tempt to provide a solid basis for the work presented later. They include linear models and distances, OLS subset selection, asymptotics, shrinkage, data resampling, RIC and CIC procedures, and empirical Bayes, etc. The basic method of applying the general ideas of competing models to linear regression is embedded in this chapter.

Chapter 3, the core of the thesis, formally proposes and theoretically evaluates pace regression. New concepts, such as contribution, H- and h-functions, are defined in Section 3.3 and their roles in modeling are illustrated in Section 3.4, respectively. Six procedures of pace regression are formally defined in Section 3.5, while k-asymptotic optimality is established in Section 3.6.

A few important issues are discussed in Chapter 4, some completing the definition of the pace regression procedures in special situations, others expanding their implications into a broader arena, and still others raising open questions.

In Chapter 5, the minimum distance estimation of a mixing distribution is investigated, along with a brief review of maximum likelihood estimation. It focuses on establishing consistency of the proposed probability-measure-based and nonnegative-measure-based estimators. The minority cluster problem, vital for pace regression, is discussed in Section 5.5. Section 5.6 further empirically investigates the accuracy of these clustering estimators in situations of overlapping clusters.

Results of simulation studies of pace regression and other procedures for fitting linear models are given in Chapter 6.

The final chapter briefly summarizes the whole work and provides some topics for future work and open questions.

The source code in S and FORTRAN, and some help files, are listed in the Appendix.

Chapter 2

Issues in fitting linear models

2.1 Introduction

The basic idea of regression analysis is to fit a linear model to a set of data. The classical ordinary least squares (OLS) estimator is simple, computationally cheap, and has well-established theoretical justification. Nevertheless, the models it produces are often less than satisfactory. For example, OLS does not detect redundancy in the set of independent variables that are supplied, and when a large number of variables are present, many of which are redundant, the model produced usually has worse predictive performance on future data than simpler models that take fewer variables into account.

Many researchers have investigated methods of *subset selection* in an attempt to neutralize this effect. The most common approach is OLS subset selection: from a set of OLS-fitted subset models, choose the one that optimizes some pre-determined modeling criterion. Almost all these procedures are based on the idea of thresholding variation reduction: calculating how much the variation of the model is increased if each variable in turn is taken away, setting a threshold on this amount, and discarding variables that contribute less than the threshold. The rationale is that a noisy variable—i.e., one that has no predictive power—usually reduces the variation only marginally, whereas the variation accounted for by a meaningful variable is larger and grows with the variable's significance.

Many well-known procedures, including FPE (Akaike, 1970), AIC (Akaike, 1973), C_p (Mallows, 1973) and BIC (Schwarz, 1978) follow this approach. While these certainly work well for some data sets, extensive practical experience and many simulation studies have exposed shortcomings in them all. Often, for example, a certain proportion of redundant variables are included in the final model (Derksen and Keselman, 1992). Indeed, there are data sets for which a full regression model outperforms the selected subset model unless most of the variables are redundant (Hoerl et al., 1986; Roecker, 1991).

Shrinkage methods offer an alternative to OLS subset selection. Simulation studies show that the technique of *biased ridge regression* can outperform OLS subset selection, although it generates a more complex model (Frank and Friedman, 1993; Hoerl et al., 1986). The shrinkage idea was further explored by Breiman (1995) and Tibshirani (1996), who were able to generate models that are less complex than ridge regression models yet still enjoy higher predictive accuracy than OLS subset models. Empirical evidence presented in these papers suggests that shrinkage methods yield greater predictive accuracy than OLS subset selection when a model has many noisy variables, or at most a moderate number of variables with moderate-sized effects—whereas they perform worse when there are a few variables that have a dramatic effect on the outcome.

These problems are systematic: the performance of modeling procedures can be related to the effects of variables and the extent of these effects. Researchers have sought to understand these phenomena and use them to motivate new approaches. For example, Miller (1990) investigated the selection bias that is introduced when the same data is used both to estimate the coefficients and to choose the subsets. New procedures, including the little bootstrap (Breiman, 1992), RIC (Donoho and Johnstone, 1994; Foster and George, 1994), and CIC (Tibshirani and Knight, 1997) have been proposed. While these undoubtedly produce good models for many data sets, we will see later that there is no single approach that solves these systematic problems in a general sense.

Our work in this thesis will show that these problems are the tip of an iceberg. They

are manifestations of a much more general phenomenon that can be understood by examining the expected contributions that individual variables make in an orthogonal decomposition of the estimated model. This analysis leads to a new approach called "pace regression"; see Chapter 3.

We investigate model estimation in a general sense that subsumes subset selection. We do not confine our efforts to finding the best of the subset models; instead we address the whole space of linear models and regard subset models as a special case. But it is an important special case, because simplifying the model structure has wide applications in practice, and we will use it extensively to help sharpen our ideas.

In this chapter, we introduce and review some important issues in linear regression to provide a basis for the analysis that follows. We briefly review the major approaches to model selection, focusing on their failure to solve the systematic problems raised above. A common thread emerges: the key to solving the general problem of model selection in linear regression lies in the distribution of the effects of the variables that are involved.

2.2 Linear models and the distance measure

Given k explanatory variables, the response variable, and n independent observations, a linear model can be written in the following matrix form,

$$y = X\beta^* + \epsilon, \tag{2.1}$$

where y is the n-dimensional response vector, X the $n \times k$ design matrix, β^* the k-dimensional parameter vector of the true, underlying, model, and each element of the noise component ϵ is independent and identically distributed according to $N(0, \sigma^2)$. We assume for the most part that the variance σ^2 is known; if not, it can be estimated using the OLS estimator $\hat{\sigma}^2$ of the full model.

With variables defined, a linear model is uniquely determined by a parameter vec-

tor β. Therefore, we use $\mathcal{M}$ to denote any model, $\mathcal{M}(\beta)$ the model with parameter vector β, and $\mathcal{M}^*$ as shorthand for the underlying model $\mathcal{M}(\beta^*)$. The entire space of linear models is $\mathbb{M}_k = \{\mathcal{M}(\beta) : \beta \in \mathbb{R}^k\}$. Note that models considered (and produced) by the OLS method, OLS subset selection methods, and shrinkage methods are all subclasses of $\mathbb{M}_k$. Any zero entry in β^* corresponds to a truly redundant variable. In fact, variable selection is not a problem independent from parameter estimation, or more generally, modeling; it is just a special case in which the discarded variables correspond to zero entries in the estimated parameter vector.

Further, we need a general distance measure between any two models in the space, which supersedes the usually-used loss function in the sense that minimizing the expected distance between the estimated and true models also minimizes (perhaps approximately) the expected loss. We use a general distance measure rather than the loss function only, because other distances are also of interest. Defining a distance measure coincides too with the notion of a metric space—which is how we envisage the model space.

Now we define the distance measure by relating it to the commonly-used quadratic loss. Given a design matrix X, the prediction vector of the model $\mathcal{M}(\beta)$ is $y_{\mathcal{M}(\beta)} = X\beta$. In particular, the true model $\mathcal{M}^*$ predicts the output vector $y^* = y_{\mathcal{M}^*} = X\beta^*$. The distance between two models is defined as

$$\mathcal{D}(\mathcal{M}(\beta_1), \mathcal{M}(\beta_2)) = ||y_{\mathcal{M}(\beta_1)} - y_{\mathcal{M}(\beta_2)}||^2/\sigma^2, \tag{2.2}$$

where $||\cdot||$ denotes the L_2 norm. Note that our real interest, the quadratic loss $||y_{\mathcal{M}} - y^*||^2$ of the model $\mathcal{M}$, is directly related to the distance $\mathcal{D}(\mathcal{M}, \mathcal{M}^*)$ by a scaling factor σ^2. Therefore, in the case that σ^2 is unknown and $\hat{\sigma}^2$ is used instead, any conclusion concerning the distance using σ^2 remains true asymptotically (Section 3.6).

Of course, $||y_{\mathcal{M}} - y^*||^2$ is only the loss under the x-fixed assumption—i.e., the design matrix X remains unchanged from future data. Another possible assumption used in fitting linear models is x-random—i.e., each explanatory variable is a ran-

dom variable in the sense that both training and future data are drawn independently from the same distribution. The implications of x-fixed and x-random assumptions in subset selection are discussed by Thompson (1978); see also Miller (1990), Breiman (1992) and Breiman and Spector (1992) for different treatment of two cases, in particular when n is small. Although our work presented later will not directly investigate the x-random situation, it is well known that the two assumptions converge as $n \to \infty$, and hence procedures optimal in one situation are also optimal in the other, asymptotically.

The performance of models (produced by modeling procedures) is thus naturally ordered in terms of their expected distances from the true model. The modeling task is hence to find a model that is as close as possible, in expectation, to the true one. Two equivalent models, say $\mathcal{M}(\beta_1)$ and $\mathcal{M}(\beta_2)$, denoted by $\mathcal{M}(\beta_1) = \mathcal{M}(\beta_2)$, have equal expected values of this distance.

2.3 OLS subset models and their ordering

An OLS *subset model* is one that uses a subset of the k candidate variables and whose parameter vector is an OLS fit. When determining the best subset to use, it is common practice to generate a sequence of $k+1$ *nested models* $\{\mathcal{M}_j\}$ with increasing numbers j of variables. $\mathcal{M}_0$ is the *null model* with no variables and $\mathcal{M}_k$ is the *full model* with all variables included. The OLS estimate of model $\mathcal{M}_j$'s parameter vector is $\hat{\beta}_{\mathcal{M}_j} = (X'_{\mathcal{M}_j} X_{\mathcal{M}_j})^{-1} X'_{\mathcal{M}_j} y$, where $X_{\mathcal{M}_j}$ is the $n \times j$ design matrix for model $\mathcal{M}_j$. Let $P_{\mathcal{M}_j} = X_{\mathcal{M}_j}(X'_{\mathcal{M}_j} X_{\mathcal{M}_j})^{-1} X'_{\mathcal{M}_j}$ be the orthogonal projection matrix from the original k-dimensional space onto the reduced j-dimensional space. Then $\hat{y}_{\mathcal{M}_j} = P_{\mathcal{M}_j} y$ is the OLS estimate of $y^*_{\mathcal{M}_j} = P_{\mathcal{M}_j} y^*$.

One way of determining subset models is to include the variables in a pre-defined order using prior knowledge about the modeling situation. For example, in time series analysis it usually makes good sense to give preference to closer points when selecting autoregressive terms, while when fitting polynomials, lower-degree terms

are often included before higher-degree ones. When the variable sequence is pre-defined, a total of $k + 1$ subset models are considered.

In the absence of prior ordering, a data-driven approach must be used to determine appropriate subsets. The final model could involve any subset of the variables. Of course, computing and evaluating all 2^k models rapidly becomes computationally infeasible as k increases. Techniques that are used in practice include forward, backward, and stepwise ranking of variables based on partial-F ratios (Thompson, 1978).

The difference between the prior ordering and data-driven approaches affects the subset selection procedure. If the ordering of variables is pre-defined, subsets are determined independently of the data, which implies that the ratio between the residual sum of squares and the estimated variance can be assumed to be F distributed. The subset selection criteria FPE, AIC, and C_p all make this assumption. However, data-driven ordering complicates the situation. Candidate variables compete to enter and leave the model, causing competition bias (Miller, 1990). It is certainly possible to use FPE, AIC and C_p in this situation, but they lack theoretical support, and in practice they perform worse than when the variable order is correctly pre-defined. For example, suppose underfitting is negligible and the number of redundant variables increases without bound. Then the predictive accuracy of the selected model and its expected number of redundant variables both tend to constant values when the variable order is pre-defined (Shibata, 1976), whereas in the data-driven scenario they both increase without bound.

Pre-defining the ordering makes use of prior knowledge of the underlying model. As is only to be expected, this will improve modeling if the information is basically correct, and hinder it otherwise. In practice, a combination of pre-defined and data-driven ordering is often used. For example, when certain variables are known to be relevant, they should definitely be kept in the model; also, it is common practice to always retain the constant term.

2.4 Asymptotics

We will be concerned with two asymptotic situations: *n-asymptotics*, where the number of observations increases without bound, and *k-asymptotics*, where the number of variables increases without bound. Usually k-asymptotics implies n-asymptotics; for example, our k-asymptotic conclusion implies, as least, $n-k \to \infty$ (see Section 3.6). In this section we review some n-asymptotic results. The remainder of the thesis is largely concerned with k-asymptotics, which justify the applicability of the proposed procedures in situation of large data sets with many possible variables—i.e., a situation when subset selection makes sense.

The model selection criteria FPE, AIC and C_p are *n-asymptotically equivalent* (Shibata, 1981) in the sense that they depend on threshold values of F-statistic that become the same—in this case, 2—as n approaches infinity. With reasonably large sample sizes, the performance of different n-asymptotically equivalent criteria are hardly distinguishable, both theoretically and experimentally. When discussing asymptotic situations, we use AIC to represent all three criteria. Note that this definition of equivalence is less general than ours, as defined in Sections 1.2 and 2.2; ours will be used throughout the thesis.

Asymptotically speaking, the change in the residual sum of squares of a significant variable is $O(n)$, whereas that of a redundant variable has a upper bound $O(1)$, in probability, and a upper bound $O(\log\log n)$, almost surely; see, e.g., Shibata (1986), Zhao et al. (1986), and Rao & Wu (1989). The model estimator generated by a threshold function bounded between $O(1)$ and $O(n)$ is weakly consistent in terms of model dimensionality, whereas one whose threshold function is bounded between $O(\log\log n)$ and $O(n)$ is strongly consistent.

Some model selection criteria are n-asymptotically strongly consistent. Examples include BIC (Schwarz, 1978), ϕ (Hannan and Quinn, 1979), GIC (Zhao et al., 1986), and Rao and Wu (1989). These all replace AIC's threshold of 2 by an increasing function of n bounded between $O(\log\log n)$ and $O(n)$. The function value usually

exceeds 2 (unless n is very small), giving a threshold that is larger than AIC's. However, employing the rate of convergence does not guarantee a satisfactory model in practice. For any finite data set, a higher threshold runs a greater risk of discarding a non-redundant variable that is only barely contributive. Criteria such as AIC that are n-asymptotically inconsistent do not necessarily perform worse than consistent ones in finite samples.

These OLS subset selection criteria all minimize a quantity that becomes, in the sense of n-asymptotic equivalence,

$$||y - y_{\mathcal{M}_j}||^2/\sigma^2 + \tau j \tag{2.3}$$

with respect to the dimensionality parameter j, where τ is the threshold value. Denote the n-asymptotic equivalence of procedures or models by $=_n$ (and k-asymptotic equivalence by $=_k$). We write (2.3) in parameterized form as $\text{OLSC}(\tau)$, where $\text{OLS} =_n \text{OLSC}(0)$, $\text{AIC} =_n \text{OLSC}(2)$ and $\text{BIC} =_n \text{OLSC}(\log n)$. The model selected by criterion (2.3) is denoted by $\mathcal{M}^{\text{OLSC}(\tau)}$. Since equivalent procedures imply equivalent estimated models (and vice versa), we can also write $\mathcal{M}^{\text{OLS}} =_n \mathcal{M}^{\text{OLSC}(0)}$, $\mathcal{M}^{\text{AIC}} =_n \mathcal{M}^{\text{OLSC}(2)}$ and $\mathcal{M}^{\text{BIC}} =_n \mathcal{M}^{\text{OLSC}(\log n)}$.

2.5 Shrinkage methods

Shrinkage methods provide an alternative to OLS subset selection. Ridge regression gives a biased estimate of the model's parameter vector that depends on a ridge parameter. Increasing this quantity shrinks the OLS parameters toward zero. This may give better predictions by reducing the variance of predicted values, though at the cost of a slight increase in bias. It often improves the performance of the OLS estimate when some of the variables are (approximately) collinear. Experiments show that ridge regression can outperform OLS subset selection if most variables have small to moderate effects (Tibshirani, 1996). Although standard ridge regression does not reduce model dimensionality, its lesser known variants do (Miller, 1990).

The *nn-garrote* (Breiman, 1995) and *lasso* (Tibshirani, 1996) procedures zero some parameters and shrink others by defining linear inequality constraints on the parameters. Experiments show that these methods outperform ridge regression and OLS subset selection when predictors have small to moderate numbers of moderate-sized effects, whereas OLS subset selection based on C_p prevails over others for small numbers of large effects (Tibshirani, 1996).

All shrinkage methods rely on a parameter: the ridge parameter for ridge regression, the garrote parameter for the nn-garrote, and the tuning parameter for the lasso. In each case the parameter value significantly influences the result. However, there is no consensus on how to determine suitable values, which may partly explain the unstable performance of these methods. In Section 3.4, we offer a new explanation of shrinkage methods.

2.6 Data resampling

Standard techniques of data resampling, such as cross-validation and the bootstrap, can be applied to the subset selection problem. Theoretical work has shown that, despite their computational expense, these methods perform no better than the OLS subset selection procedures. For example, under weak conditions, Shao (1993) shows that the model selected by leave-d-out cross-validation, denoted $\text{CV}(d)$, is n-asymptotically consistent only if $d/n \to 1$ and $n-d \to \infty$ as $n \to \infty$. This suggests that, perhaps surprisingly, the training set in each fold should be chosen to be as small as possible, if consistency is desired. Zhang (1993) further establishes that under similar conditions, $\text{CV}(d) = \text{OLSC}((2n-d)/(n-d))$ n-asymptotically. This means that $\text{AIC} = \text{CV}(1)$ and $\text{BIC} = \text{CV}(n(\log n - 2)/(\log n - 1))$ n-asymptotically.

The behavior of the bootstrap for subset selection is examined by Shao (1996), who proves that if the bootstrap using sample size m, $\text{BS}(m)$, satisfies $m/n \to 0$, it is n-asymptotically equivalent to $\text{CV}(n-m)$; in particular, $\text{BS}(n) = \text{CV}(1)$ n-asymptotically. Therefore, $\text{BS}(m) = \text{OLSC}((n+m)/m)$ n-asymptotically.

One problem with these data resampling methods is the difficulty of choosing an appropriate number of folds d for cross-validation, or an appropriate sample size m for the bootstrap—both affect the corresponding procedures. More fundamentally, their asymptotic equivalence to OLSC seems to imply that these data resampling techniques fail to take the competition phenomenon into account and hence are unable to solve the problems raised in Section 2.1.

2.7 The RIC and CIC

When there is no pre-defined ordering of variables, it is necessary to take account of the process by which a suitable ordering is determined. The expected value of the ith largest squared t-statistic of k noisy variables approaches $2\log(k/i)$ as k increases indefinitely. This property can help with variable selection.

The soft thresholding procedure developed in the context of wavelets (Donoho and Johnstone, 1994) and the RIC for subset selection (Foster and George, 1994) both aim to eliminate all non-contributory variables, up to the largest, by replacing the threshold 2 in AIC with $2\log k$; that is, RIC $=$ OLSC$(2\log k)$. The more variables, the higher the threshold. When the true hypothesis is the null hypothesis (that is, there are no contributive variables), or the contributive variables all have large effects, RIC finds the correct model by eliminating all noisy variables up to the largest. However, when there are significant variables with small to moderate effects, these can be erroneously eliminated by the higher threshold value.

The CIC procedure (Tibshirani and Knight, 1997) adjusts the training error by taking into account the average covariance of the predictions and responses, based on the permutation distribution of the dataset. In an orthogonally decomposed model

space, the criterion simplifies to

$$\text{CIC}(j) = ||y - y_{\mathcal{M}_j}||^2 + 2\text{E}^0\left[\sum_{i=1}^{j} t^2_{(i:k)}\right]\hat{\sigma}^2 \tag{2.4}$$

$$\approx ||y - y_{\mathcal{M}_j}||^2 + 4\sum_{i=1}^{j}\log(k/i)\hat{\sigma}^2, \tag{2.5}$$

where $t^2_{(i:k)}$ is the ith largest squared t-statistic out of k, and E^0 is the expectation over the permutation distribution. As this equation shows, CIC uses a threshold value that is twice the expected sum of the squared t statistics of the j largest noisy variables out of k. Because $\lim_{k\to\infty} P[t^2_{(1:k)} \geq 2\text{E}t^2_{(1:k)}] = 0$, this means that, for the null hypothesis, even the largest noisy variable is almost always eliminated from the model. Furthermore, this has the advantage over RIC that smaller contributive variables are more likely to be recognized and retained, due to the uneven, stairwise threshold of CIC for each individual variable.

Nevertheless, shortcomings exist. For example, if most variables have strong effects and will certainly not be discarded, the remaining noisy variables are treated by CIC as though they were the smallest out of k noisy variables—whereas in reality, the number should be reduced to reflect the smaller number of noisy variables. An overfitted model will likely result. Analogously, underfitting will occur when there are just a few contributive variables (Chapter 6 gives an experimental illustration of this effect.) CIC is based on an expected ordering of the squared t-statistics for noisy variables, and does not deal properly with situations where variables have mixed effects.

2.8 Empirical Bayes

Although our work started from the viewpoint of competing models, the solution discussed in the later chapters turns out to be very similar to the methodology of empirical Bayes. In this section, we briefly review this paradigm, in an attempt to provide a simple and solid basis for introducing the ideas in our approach. However,

Sections 4.6 and 7.3 show that the competing models viewpoint seems to fall beyond the scope of the existing empirical Bayes methods.

2.8.1 Empirical Bayes

The empirical Bayes methodology is due to Robbins (1951, 1955, 1964, 1983). After its appearance, it was quickly considered to be a breakthrough in the theory of statistical decision making (cf. Neyman, 1962). There is a large literature devoted to it. Books that provide introductions and discussions include Berger (1985), Maritz and Lwin (1989), Carlin and Louis (1996), and Lehmann and Casella (1998). Carlin and Louis (2000) give a recent review.

The fundamental idea of empirical Bayes, encompassing the *compound decision problem* (Robbins, 1951)[1], is to use data to estimate the prior distribution (and/or the posterior distribution) in Bayesian analysis, instead of resorting to a subjective prior in the conventional way. Despite the term "empirical Bayes" it is a frequentist approach, and Lindley notes in his discussion of Copas (1969) that "there is no one less Bayesian than an empirical Bayesian."

A typical empirical Bayes problem involves observations $x_1, \ldots, x_k$, independently sampled from distributions $F(x_i; \theta_i^*)$, where θ_i^* may be completely different for each individual i. Denote $\mathbf{x} = (x_1, \ldots, x_k)^t$ and $\boldsymbol{\theta}^* = (\theta_1^*, \ldots, \theta_k^*)^t$. Let $\boldsymbol{\theta}$ be an estimator of $\boldsymbol{\theta}^*$. Given a loss function, say,

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = ||\boldsymbol{\theta} - \boldsymbol{\theta}^*||^2 = \sum_{i=1}^{k} (\theta_i - \theta_i^*)^2, \tag{2.6}$$

the problem is to find the optimal estimator that minimizes the expected loss.

As we know, the usual maximum likelihood estimator is $\boldsymbol{\theta}^{\mathrm{ML}} = \mathbf{x}$, which is also the Bayes estimator with respect to the non-informative constant prior. The empirical Bayes estimator, however, is different. Since $x_1, \ldots, x_k$ are independent,

[1] Throughout the thesis, we adopt the widely-used term *empirical Bayes*, although the term *compound decision* probably makes more sense in our context of competing models.

we can consider that $\theta_1^*, \ldots, \theta_k^*$ are i.i.d. from a common prior distribution $G(\theta^*)$. Therefore, it is possible to estimate $G(\theta^*)$ from $x_1, \ldots, x_k$ based on the relationship

$$F_k(x) \approx \int F(x; \theta^*) \, \mathrm{d}G(\theta^*). \tag{2.7}$$

From this relationship, it is possible to find an estimator $\widehat{G}(\theta^*)$ of $G(\theta^*)$ from the known $F_k(x)$ and $F(x; \theta^*)$ (we will discuss how to do this in Chapter 5). Once $\widehat{G}(\theta^*)$ has been found, the remaining steps are exactly like Bayesian analysis with the subjective prior replaced by the estimated $\widehat{G}(\theta^*)$, resulting in the empirical Bayes estimator

$$\theta_i^{\mathrm{EB}}(i, \mathbf{x}) = \frac{\int_\Theta \theta^* f(x_i; \theta^*) \, \mathrm{d}\widehat{G}(\theta^*)}{\int_\Theta f(x_i; \theta^*) \, \mathrm{d}\widehat{G}(\theta^*)}. \tag{2.8}$$

Note that the data are used twice: once for the estimation of $G(\theta^*)$ and again for updating each individual θ_i^{ML} (or x_i).

The empirical Bayes estimator can be theoretically justified in the asymptotic sense. The Bayes risk of the estimator converges to the true Bayes risk as $k \to \infty$, i.e., as if the true distribution $G(\theta^*)$ were known. Since no estimator can reduce the Bayes risk below the true Bayes risk, it is *asymptotically optimal* (Robbins, 1951, 1955, 1964). Clearly $\boldsymbol{\theta}^{\mathrm{ML}}$ is inferior to $\boldsymbol{\theta}^{\mathrm{EB}}$ in this sense.

Empirical Bayes can be categorized in different ways. The best known division is between *parametric empirical Bayes* and *nonparametric empirical Bayes*, depending on whether or not the prior distribution assumes a parametric form—for example, a normal distribution.

Applications of empirical Bayes are surveyed by Morris (1983) and Maritz and Lwin (1989). The method is often used either to generate a prior distribution based on past experience (see, e.g., Berger, 1985), or to solve compound decision problems when a loss function in the form of (2.6) can be assumed. In most cases, the parametric empirical Bayes approach is adopted (see, e.g., Robbins, 1983; Maritz and Lwin, 1989), where the mixing distribution has a parametric interpretation. To

my knowledge, it is rarely used in the general setting of empirical modeling, which is the main concern of the thesis. Further, the less frequently used nonparametric empirical Bayes appears more appropriate in our approach.

2.8.2 Stein estimation

Shrinkage estimation or Stein estimation is another frequentist effort, and has close ties to parametric empirical Bayes. It was originated by Stein (1955), who found, surprisingly, that the usual maximum likelihood estimator for the mean of a multivariate normal distribution, i.e., $\boldsymbol{\theta}^{\text{ML}} = \mathbf{x}$, is inadmissible when $k > 2$. He proposed a new estimator

$$\boldsymbol{\theta}^{\text{JS}} = \left[1 - \frac{(k-2)\sigma^2}{||\mathbf{x}||^2}\right]\mathbf{x}, \tag{2.9}$$

which was later shown by James and Stein (1961) to have smaller mean squared error than the usual estimator for every $\boldsymbol{\theta}^*$, when $k > 2$, although the new estimator is itself inadmissible. This new estimator, known as the James-Stein estimator, improves over the usual estimator by shrinking it toward zero. Subsequent development of shrinkage estimation goes far beyond this case, for example, positive-part James-Stein estimator, shrinking toward the common mean, toward a linear subspace, etc. (cf. Lehmann and Casella, 1998).

The connection of the James-Stein estimator to parametric empirical Bayes was established in the seminal papers of Efron and Morris (1971, 1972a,b, 1973a,b, 1975, 1977). Among other things, they showed that the James-Stein estimator is exactly the parametric empirical Bayes estimator if $G(\theta^*)$ is assumed to be a normal distribution and the unbiased estimator of the shrinkage factor is used.

Although our work does not adopt the Stein estimation approach and focuses only on asymptotic results, it is interesting to note that while the general empirical Bayes estimator achieves asymptotic optimality, the James-Stein estimator, and a few other similar ones, uniformally outperform the usual ML estimator even for finite k (> 2).

So far there does not seem to have been found an estimator which both dominates the usual ML estimator for finite k and is k-asymptotically optimal for arbitrary $G(\theta^*)$.

The James-Stein estimator is applied to linear regression by Sclove (1968) in the form

$$\widetilde{\beta} = (1 - \frac{(k-2)\sigma^2}{RSS})\widehat{\beta}, \tag{2.10}$$

where $\widehat{\beta}$ is the OLS estimate of the regression coefficients and RSS is the regression sum of squares. Note that when RSS is large in comparison with the numerator $(k-2)\sigma^2$, shrinkage has almost no effect. Alternative derivations of this estimator are given by Copas (1983).

2.9 Summary

In this chapter, we reviewed the major issues in fitting linear models, as well as procedures such as OLS, OLS subset selection, shrinkage, asymptotics, x-fixed *vs.* x-random, data resampling, etc. From the discussion, a common thread has emerged: each procedure's performance is closely related to the distribution of the effects of the variables it involves. This consideration is analogous to the idea underlying empirical Bayes methodology, and will be exploited in later chapters.

Chapter 3

Pace regression

3.1 Introduction

In Chapter 2, we reviewed major issues in fitting linear models and introduced the empirical Bayes methodology. From this brief review, it becomes clear that the major procedures for linear regression all fail to solve the systematic problems raised in Section 2.1 in any general sense. It has also emerged that each procedure's performance relates closely to the proportions of the different effects of the individual variables. It seems that the essential feature of any particular regression problem is the distribution of the effects of the variables it involves. This raises three questions: how to define this distribution; how to estimate it from the data, if indeed this is possible; and how to formulate satisfactory general regression procedures if the distribution is known. This chapter answers these questions.

In Section 3.2, we introduce the orthogonal decomposition of models. In the resulting model space, the effects of variables, which correspond to dimensions in the space, are statistically independent. Once the effects of individual variables have been teased out in this way, the distribution of these effects is easily defined.

The second question asks whether we can estimate this distribution from the data. The answer is "yes." Moreover, estimators exist which are strongly consistent, in the sense of k-asymptotics (and also n-asymptotics). In fact, the estimation is sim-

ply a clustering problem—to be more precise, it involves estimating the mixing distribution of a mixture. Section 3.6 shows how to perform this.

We answer the third question by demonstrating three successively more powerful techniques, and establish optimality in the sense of k-asymptotics. First, following conventional ideas of model selection, the distribution of the effects of the variables can be used to derive an optimal threshold for OLS subset model selection. The resulting estimator is provably superior to all existing OLS subset selection techniques that are based on a thresholding procedure. Second, by showing that there are limitations to the use of thresholding variation reduction, we develop an improved selection procedure that does not involve thresholding. Third, abandoning the use of selection entirely results in a new adjustment technique that substantially outperforms all other procedures—outperforming OLS even when all variables have large effects. Section 3.5 formally introduces these procedures, which are all based on analyzing the dimensional contributions of the estimated models introduced in Section 3.3, and discusses their properties. Section 3.4 illustrates the procedures through examples.

Two optimalities, corresponding to the minimum prior and posterior expected losses respectively, are defined in Section 3.4. The optimality of each proposed estimator is theoretically established in the sense of k-asymptotics, which further implies n-asymptotics. When the noise variance is unknown and replaced by the OLS unbiased estimator $\hat{\sigma}^2$, the realization of the optimality requires $(n - k) \to \infty$; see Corollary 3.12.1 in Section 3.6.

Some technical details are postponed to the appendix in Section 3.8.

3.2 Orthogonal decomposition of models

In this section, we discuss the orthogonal decomposition of linear models, and define some special distances that are of particular interest in our modeling task. We will also briefly discuss in Section 3.2.3 the advantages of using an orthogo-

nal model space, the OLS subset models, and the choice of an orthonormal basis. We assume that no variables are collinear—that is, a model with k variables has k degrees of freedom. We return to the problem of collinear variables in Section 4.3.

Following the notation introduced in Section 2.2, given a model $\mathcal{M}(\beta)$ with parameter vector β, its prediction vector is $y_{\mathcal{M}} = X\beta$ where X is the $n \times k$ design matrix. This vector is located in the space spanned by the k separate n-vectors that represent the values of the individual variables. For any orthonormal basis of this space $b_1, b_2, \ldots, b_k$ (satisfying $||b_j|| = 1$), let $P_1, P_2, \ldots, P_k$ be the corresponding projection matrices from this space onto the axes. $y_{\mathcal{M}}$ decomposes into k components $P_1 y_{\mathcal{M}}, P_2 y_{\mathcal{M}}, \ldots, P_k y_{\mathcal{M}}$, each being a projection onto a different axis. Clearly the whole is the sum of the parts: $y_{\mathcal{M}} = \sum_{j=1}^{k} P_j y_{\mathcal{M}}$.

3.2.1 Decomposing distances

The distance $\mathcal{D}(\mathcal{M}(\beta_1), \mathcal{M}(\beta_2))$ between models $\mathcal{M}(\beta_1)$ and $\mathcal{M}(\beta_2)$ has been defined in (2.2). Although this measure involves the noise variance σ^2 for convenience of both analysis and computation, it is $||y_{\mathcal{M}(\beta_1)} - y_{\mathcal{M}(\beta_2)}||^2$ that is the center of interest.

Given an orthogonal basis, the distance between two models can be decomposed as follows:

$$\mathcal{D}(\mathcal{M}(\beta_1), \mathcal{M}(\beta_2)) = \sum_{j=1}^{k} \mathcal{D}_j(\mathcal{M}(\beta_1), \mathcal{M}(\beta_2)), \tag{3.1}$$

where

$$\mathcal{D}_j(\mathcal{M}(\beta_1), \mathcal{M}(\beta_2)) = ||P_j y_{\mathcal{M}(\beta_1)} - P_j y_{\mathcal{M}(\beta_2)}||^2 / \sigma^2 \tag{3.2}$$

is the jth *dimensional distance* between the models. The property of *additivity* of distance in this orthogonal space will turn out to be crucial for our analysis: the distance between the models is equal to the sum of the distances between the

models' projections.

Denote by $\mathcal{M}_0$ the *null model* $\mathcal{M}(0)$, whose every parameter is zero. The distance between $\mathcal{M}$ and the null model is the *absolute distance* of $\mathcal{M}$, denoted by $\mathcal{A}(\mathcal{M})$; that is, $\mathcal{A}(\mathcal{M}) = \mathcal{D}(\mathcal{M}, \mathcal{M}_0)$. Decomposing the absolute distance yields

$$\mathcal{A}(\mathcal{M}) = \sum_{j=1}^{k} \mathcal{A}_j(\mathcal{M}), \tag{3.3}$$

where $\mathcal{A}_j(\mathcal{M}) = ||P_j y_{\mathcal{M}}||^2/\sigma^2$ is the jth *dimensional absolute distance* of $\mathcal{M}$.

3.2.2 Decomposing the estimation task

Two models are of central interest in the process of estimation: the true model $\mathcal{M}^*$ and the estimated model $\mathcal{M}$. In Section 2.2, we defined the distance between them as the *loss* of the estimated model, denoted by $\mathcal{L}(\mathcal{M})$; that is, $\mathcal{L}(\mathcal{M}) = \mathcal{D}(\mathcal{M}, \mathcal{M}^*)$. We also related this loss function to the quadratic loss incurred when predicting future observations, as required by the minimum expected loss principle.

Being a distance, the loss can be decomposed into dimensional components

$$\mathcal{L}(\mathcal{M}) = \sum_{j=1}^{k} \mathcal{L}_j(\mathcal{M}), \tag{3.4}$$

where $\mathcal{L}_j(\mathcal{M}) = \mathcal{D}_j(\mathcal{M}, \mathcal{M}^*) = ||P_j y_{\mathcal{M}} - P_j y_{\mathcal{M}^*}||^2/\sigma^2$ is the jth *dimensional loss* of model $\mathcal{M}$.

Orthogonal decomposition breaks the estimation task down into individual estimation tasks for each of the k dimensions. $\mathcal{A}_j(\mathcal{M}^*)$ is the underlying absolute distance in the jth dimension, and $\mathcal{A}_j(\mathcal{M})$ is an estimate of it. The loss incurred by this estimate is $\mathcal{L}_j(\mathcal{M})$. The sum of the losses in each dimension is the total loss $\mathcal{L}(\mathcal{M})$ of the model $\mathcal{M}$. This reduces the modeling task to estimating $\mathcal{A}_j(\mathcal{M}^*)$ for all j. Once these estimated distances have been found for each dimension $j = 1, \ldots, k$, the estimated model can be reconstructed from them.

Our estimation process has two steps: an initial estimate of $\mathcal{A}_j(\mathcal{M}^*)$ followed by a further refinement stage. This implies that there is some information not used in the initial estimate that can be exploited to improve it; we use the loss function to guide refinement. The first step is to find a relationship between the initial estimate $\mathcal{A}_j(\mathcal{M})$ and $\mathcal{A}_j(\mathcal{M}^*)$. The classic OLS estimate $\widehat{\mathcal{M}}$, which has parameter vector $\widehat{\beta} = (X'X)^{-1}X'y$, provides a basis for such a relationship, because it is well known that for all j, $\mathcal{A}_j(\widehat{\mathcal{M}})$ are independently, noncentrally χ^2 distributed with one degree of freedom and noncentrality parameter $\mathcal{A}_j(\mathcal{M}^*)/2$ (Schott, 1997, p.390). We write this as

$$\mathcal{A}_j(\widehat{\mathcal{M}}) \sim \chi^2_1(\mathcal{A}_j(\mathcal{M}^*)/2) \text{ independently for all } j. \tag{3.5}$$

When σ^2 is unknown and therefore replaced by the unbiased OLS estimate $\hat{\sigma}^2$, the χ^2 distribution in (3.5) becomes an F distribution: $\mathcal{A}_j(\widehat{\mathcal{M}}) \sim F(1, n-k, \mathcal{A}_j(\mathcal{M}^*)/2)$ (asymptotically) independently for all j. The F-distribution can be accurately approximated by (3.5) when $n - k \gg 0$.

This relationship forms a cornerstone of our work. In Section 3.8.1 we discuss how to re-generate the model from the updated absolute distances.

Updating signed projections. An alternative approach to updating absolute distances is to update signed projections, where the signs of the projections can be determined by the directions of the axis in an orthogonal space. Both approaches share the same methodology, and use a very similar analysis and derivation to obtain corresponding modeling procedures.

Our work throughout the thesis focuses on updating absolute distances. It seems to us that updating absolute distances is generally more appropriate than the alternative, although we do not exclude the possibility of using the latter in practice. Our arguments for this preference are given in Section 4.7.

3.2.3 Remarks

Advantages of an orthogonal model space. There are at least two advantages to using orthogonally decomposed models for estimation. The first is additivity: the distance between two models is the sum of their distances in each dimension. This convenient property is inherited by two special distances: the absolute distance of the model itself and the loss function of the estimated model. Of course, any quantity that involves addition and subtraction of distances between models is additive too.

The second advantage is the independence of the model's components in the different dimensions. This makes the dimensional distances between models independent too. In particular, the absolute distances in each dimension, and the losses incurred by an estimated model in each dimension—as well as any measure derived from these by additive operators—are independent between one dimension and another.

These two features allow the process of estimating the overall underlying model to be broken down into estimating its components in each dimension separately. Furthermore, the distribution of the effects of variables—precisely dimensions—can be accurately defined.

A special case: OLS subset models. It is well known that OLS subset models are a special case of orthogonally decomposed models in which each variable is associated with an orthogonal dimension. This makes it easy to simplify the model structure: discarding variables in a certain order is the same as deleting dimensions in the same order. If the discarded variables are actually redundant, deleting them makes the model more accurate.

Denote the nested subset models of $\mathcal{M}$, from the null model up to the full one, by $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_k$ respectively. Denote by $y_{\mathcal{M}_j}$ the prediction vector of the j-dimensional model $\mathcal{M}_j$, and by $P_{\mathcal{M}_j} = X_{\mathcal{M}_j}(X'_{\mathcal{M}_j}X_{\mathcal{M}_j})^{-1}X'_{\mathcal{M}_j}$ the orthogonal projection matrix from the space of k dimensions to the j-dimensional sub-

space corresponding to $\mathcal{M}_j$. Then $y_{\mathcal{M}_j} = P_{\mathcal{M}_j} y_{\mathcal{M}}$ and $P_j = P_{\mathcal{M}_j} - P_{\mathcal{M}_{j-1}}$. Furthermore, the jth orthonormal base can be written as $b_j = P_j y / ||P_j y|| = (y_{\mathcal{M}_j} - y_{\mathcal{M}_{j-1}}) / ||(y_{\mathcal{M}_j} - y_{\mathcal{M}_{j-1}})||$.

Choice of an orthonormal basis. The choice of an orthonormal basis depends on the particular application. In some cases, there may exist an obvious, reasonable orthonormal basis—for example, orthogonal polynomials in polynomial regression, or eigenvectors in principal component analysis. In other situations where no obvious basis exists, it is always possible to construct one from the given data using, say, a partial-F test (Section 2.3). The latter, more general, data-driven approach is employed in our implementation and simulation studies (Chapter 6). However, we note that the partial-F test needs to observe y-values before model construction, and hence can cause a problem of orthogonalization selection (Section 4.6), which is an analogous phenomenon to model selection.

3.3 Contributions and expected contributions

We next explore a new measure for a model: its *contribution*. An estimated model's contribution is zero for the null model and reaches a maximum when the model is the same as the underlying one. It can be decomposed into k independent, additive components in k-dimensional orthogonal space—the "dimensional contributions." In practice, these quantities are random variables, and we can define both a *cumulative expected contribution function* and its derivative of of the estimated model. These two functions can be estimated for any particular regression problem, and will turn out to play key roles in understanding the modeling process and in building actual models in practice.

3.3.1 $C(A)$ and $EC(A)$

Definition 3.1 *The contribution of an estimate $\mathcal{M}$ of the underlying model $\mathcal{M}^*$ is defined to be*

$$\mathcal{C}(\mathcal{M}) = \mathcal{L}(\mathcal{M}_0) - \mathcal{L}(\mathcal{M}). \tag{3.6}$$

The contribution is actually the difference between the losses of two models: the null and the estimated. Therefore, its sign and value indicate the merits of the estimated model against the null one: a positive contribution means that the estimated model is better than the null one in terms of predictive accuracy; a negative value means it is worse; and zero means they are equivalent.

Since $\mathcal{A}(\mathcal{M}^*)$ is a constant, maximizing the (expected) contribution $\mathcal{C}(\mathcal{M})$ is equivalent to minimizing the (expected) loss $\mathcal{L}(\mathcal{M})$, where the latter is our goal of modeling. Therefore, we now convert this goal into maximizing the (expected) contribution. Using the contribution, rather than dealing directly with losses, is particularly useful in understanding the task of model selection, as will soon become clear.

Given a k-dimensional orthogonal basis, the contribution function decomposes into k components that retain the properties of additivity and dimensional independence:

$$\mathcal{C}(\mathcal{M}) = \sum_{j=1}^{k} \mathcal{C}_j(\mathcal{M}) \tag{3.7}$$

where

$$\mathcal{C}_j(\mathcal{M}) = \mathcal{L}_j(\mathcal{M}_0) - \mathcal{L}_j(\mathcal{M}) \tag{3.8}$$

is the jth *dimensional contribution* of the model $\mathcal{M}$.

In the subset selection task, each dimension is either retained or discarded. It is clear that this decision should be based on the sign of the corresponding dimensional contribution. If a dimension's contribution is positive, retaining it will give better

predictive accuracy than discarding it, and conversely, if the contribution is negative then discarding it will improve accuracy. If the dimensional contribution is zero, it makes no difference to predictive accuracy whether that dimension is retained or not.

In following we focus on a single dimension, the jth. The results of individual dimensions can easily be combined because dimensions are independent and the contribution measure is additive. Focusing on the contribution in this dimension, we write ${a_j}^2 = \mathcal{A}_j(\mathcal{M})$ and ${a_j^*}^2 = \mathcal{A}_j(\mathcal{M}^*)$. Without loss of generality, assume that $a_j^* \geq 0$. If the projection of $y_{\mathcal{M}}$ in the jth dimension is in the same direction as that of $y_{\mathcal{M}^*}$, then a_j is the positive square root of $\mathcal{A}_j(\mathcal{M})$; otherwise it is the negative square root. In either case, the contribution can be written

$$\mathcal{C}_j(\mathcal{M}) = {a_j^*}^2 - (a_j - a_j^*)^2. \tag{3.9}$$

Clearly, $\mathcal{C}_j(\mathcal{M})$ is zero when a_j is 0 or $2a_j^*$. When a_j lies between these values, the contribution is positive. For any other values of a_j, it is negative. The maximum contribution is achieved when $a_j = a_j^*$, and has value ${a_j^*}^2$, which occurs when—in this dimension—the estimated model is the true model. This is obviously the best that the estimated model can do in this dimension.

In practice, however, only ${a_j}^2$ is available. Neither the value of ${a_j^*}^2$ nor the directional relationship between the two projections are known. Denote $\mathcal{C}_j(\mathcal{M})$ by $C({a_j}^2)$, altering the notion of the contribution of $\mathcal{M}$ in this dimension to the contribution of ${a_j}^2$. $C({a_j}^2)$ is used below as shorthand for $C({a_j}^2; {a_j^*}^2, s_j) = \mathcal{C}_j(\mathcal{M})$, where s_j is the sign of a_j. In the following we drop the subscript j when only one dimension is under consideration, giving a^2 and ${a^*}^2$ for ${a_j}^2$ and ${a_j^*}^2$ respectively. We also use A for a^2 since it is this, rather than a, that is available; likewise we use A^* for ${a^*}^2$.

We have argued that the performance of an estimated model can be analyzed in terms of the value of its contribution in each dimension. Unfortunately, this value is unavailable in practice. What can be computed, as we will show, is the condi-

tional expected value $\mathrm{E}_A[C(A; A^*, s)|A, \cdot]$ where the expectation is taken over all the uncertainties concerning A^* and s. The "$\cdot$" here represents the prior knowledge that is available about A^* and s (if any). In our work presented later, the "$\cdot$" is replaced by three quantities: A^*, $G(A^*)$ (distribution of A^*), and $\widehat{A}$ plus $G(A^*)$. The k-asymptotic "availability" of $G(A^*)$ (i.e., strongly consistent estimators) ensures our k-asymptotic conclusions.

Note that the expectation is always conditional on A. The value of A is available in practice, for example, the dimensional absolute distance of the OLS full model. Therefore, the expected contribution is a function of A, and we simply denote it by $EC(A)$ (equivalently, $\mathrm{E}_A C(A)$) as a general representation for all possible situations.

3.3.2 $H(A)$ and $h(A)$

In practice, the estimate A is a random variable. For example, according to (3.5), the OLS estimate is $\widehat{A} \sim \chi^2_1(A^*/2)$. Analogously to the definitions of CDF and pdf of a random variable, we define the *cumulative expected contribution function* and its derivative of A, denoted by $H(A)$ and $h(A)$ respectively. For convenience, we call $H(A)$ and $h(A)$ the H- and *h-functions* of A.

Definition 3.2 *The cumulative expected contribution function of A is defined as*

$$H(A) = \int_0^A EC(t) f(t)\, \mathrm{d}t, \tag{3.10}$$

where $f(A)$ is the pdf of A.

Further,

$$h(A) = \frac{\mathrm{d}H(A)}{\mathrm{d}A}. \tag{3.11}$$

Immediately, we have

$$EC(A) = \frac{h(A)}{f(A)}. \tag{3.12}$$

Note that the definitions of $H(A)$ and $h(A)$, as well as the relationship (3.12), encapsulate conditional situations. For example, with known A^*, we have $EC(A; A^*) = h(A; A^*)/f(A; A^*)$, with known $G(A^*)$, $EC(A; G) = h(A; G)/f(A; G)$, and with known $\widehat{A}$ plus $G(A^*)$, $EC(A; \widehat{A}, G) = h(A; \widehat{A}, G)/f(A; \widehat{A}, G)$. Further, H and h are mutually derivable under the same condition.

3.3.3 $h(\widehat{A}; A^*)$

Now we derive an expression for $h(\widehat{A})$ given A^*, where $\widehat{A}$ is the OLS estimate of the dimensional absolute distance. Let $\widehat{A} = \hat{a}^2$ and $a^* = +\sqrt{A^*}$, where $\hat{a}$ is a signed random variable distributed on the real line according to the pdf $p(\hat{a}; a^*)$. Note that from the property (3.5), we have a special situation that $\hat{a} \sim N(a^*, 1)$, and thus $\widehat{A} = \hat{a}^2 \sim \chi_1^2(a^{*2}/2)$, where

$$p(\hat{a}; a^*) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(\hat{a}-a^*)^2}{2}}. \tag{3.13}$$

It is this special case that motivates the utilization of the pdf of $\hat{a}$, instead of that of $\widehat{A}$ directly. This is because the pdf of the noncentral chi-squared distribution has an infinite series form, which is inconvenient for both analysis and computation.

Throughout the thesis, we only use $p(\hat{a}; a^*)$ in the form (3.13), although the following derivation works generally once $p(\hat{a}; a^*)$ is known.

With known $p(\hat{a}; a^*)$, the CDF of $\widehat{A}$ given A^* is

$$F(\widehat{A}; A^*) = \int_{-\sqrt{\widehat{A}}}^{\sqrt{\widehat{A}}} p(t; \sqrt{A^*})\, \mathrm{d}t, \tag{3.14}$$

hence

$$f(\widehat{A};A^*)=\frac{dF(\widehat{A};A^*)}{d\widehat{A}}=\frac{p(\sqrt{\widehat{A}};\sqrt{A^*})+p(-\sqrt{\widehat{A}};\sqrt{A^*})}{2\sqrt{\widehat{A}}}. \tag{3.15}$$

Using (3.9), rewrite the contribution of $\widehat{A}$ given A^* by a two-argument function $c(\hat{a};a^*)$

$$C(\widehat{A})=c(\hat{a};a^*)=a^{*2}-(\hat{a}-a^*)^2. \tag{3.16}$$

Only the sign of $\hat{a}$ can affect the value of the contribution, and so the expected contribution of $\widehat{A}$ given A^* is

$$EC(\widehat{A};A^*)=\frac{c(\sqrt{\widehat{A}};\sqrt{A^*})p(\sqrt{\widehat{A}};\sqrt{A^*})+c(-\sqrt{\widehat{A}};\sqrt{A^*})p(-\sqrt{\widehat{A}};\sqrt{A^*})}{p(\sqrt{\widehat{A}};\sqrt{A^*})+p(-\sqrt{\widehat{A}};\sqrt{A^*})}. \tag{3.17}$$

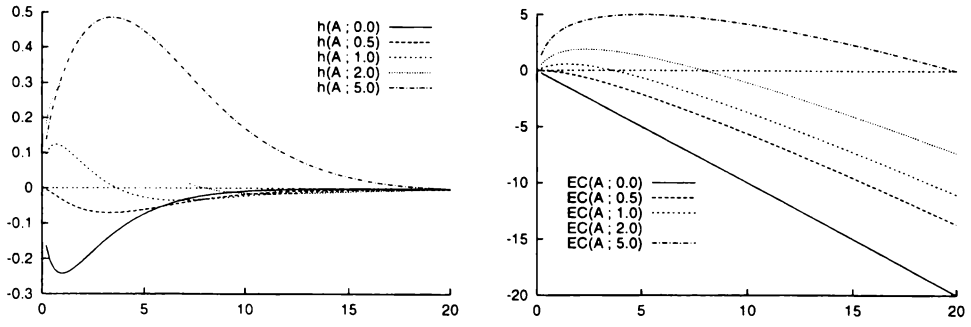
Using (3.12),

$$h(\widehat{A};A^*)=\frac{c(\sqrt{\widehat{A}};\sqrt{A^*})p(\sqrt{\widehat{A}};\sqrt{A^*})+c(-\sqrt{\widehat{A}};\sqrt{A^*})p(-\sqrt{\widehat{A}};\sqrt{A^*})}{2\sqrt{\widehat{A}}}. \tag{3.18}$$

In particular, $h(0;A^*)=0$ for every A^* (see Appendix 3.8.2). This gives the following theorem.

Theorem 3.3 *The $h(\widehat{A};A^*)$ of the* OLS *estimate $\widehat{A}$ given A^* is determined by (3.18), while the pdf $f(\widehat{A};A^*)$ is determined by (3.15).*

Because $EC(A)=h(A)/f(A)$ by (3.12), and $f(A)$ is always positive, the value of $h(A)$ has the same sign as $EC(A)$. Therefore the sign of h can be used as a criterion to determine whether a dimension should be discarded or not. Within a *positive interval*, where $h(A)>0$, A is expected to contribute positively to the predictive accuracy, whereas within a *negative one*, where $h(A)<0$, it will do the opposite. At a zero of $h(A)$ the expected contribution is zero.



(a) (b)

Figure 3.1: $h(\widehat{A}; A^*)$ and $EC(\widehat{A}; A^*)$, for $A^* = 0, .5, 1, 2, 5$.

Figure 3.1(a) shows examples of $h(\widehat{A}; A^*)$ where A^* is $0, 0.5, 1, 2$ and 5. All the curves start from the origin. When $A^* = 0$, the curve first decreases as $\widehat{A}$ increases, and then gradually increases, approaching the horizontal axis asymptotically and never rising above it. As the value of A^* grows, the $h(\widehat{A}; A^*)$ curve generally rises. However, it always lies below the horizontal axis until A^* becomes 0.5. When $A^* > 0.5$, there is one positive interval. A maximum is reached not far from A^* (the maximum approaches A^* as the latter increases), and thereafter each curve slowly descends to meet the axis at around $4A^*$ (the ordinate approaches this value as A^* increases). Thereafter the interval remains negative; within it, each h-function reaches a minimum and then ascends to approach the axis asymptotically from below.

Most observations in the last paragraph are included in the following theorem. For convenience, denote the zeros of h by $\mathcal{Z}_1$, $\mathcal{Z}_2$, $\mathcal{Z}_3$ in increasing order along the horizontal axis, and assume that h is properly defined at ∞. Since A^* in practice is always finite, we only consider the situation $A^* < \infty$.

Theorem 3.4 *Properties of* $h(\widehat{A}; A^*)$.

1. *Every* $h(\widehat{A}; A^*)$ *has three zeros (two of which may coincide).*

2. *When* $A^* \leq 0.5$, $\mathcal{Z}_1 = \mathcal{Z}_2 = 0$, $\mathcal{Z}_3 = \infty$; *when* $A^* > 0.5$, $\mathcal{Z}_1 = 0$, $0 < \mathcal{Z}_2 < \infty$, $\mathcal{Z}_3 = \infty$.

3. $\lim_{A^* \to \infty}(\mathcal{Z}_2 - 4A^*) = 0$.

4. $h(\widehat{A}; A^*) > 0$ *for* $\widehat{A} \in (0, \mathcal{Z}_2)$ *and* $h(\widehat{A}; A^*) < 0$ *for* $\widehat{A} \in (\mathcal{Z}_2, \infty)$.

5. *When* $A^* > 0.5$, $h(\widehat{A}; A^*)$ *has a unique maximum, at* $A_{\max}$, *say. Then* $\lim_{A^* \to \infty}(A_{\max} - A^*) = 0$.

6. $h(\widehat{A}; A^*)$ *is continuous for every* $\widehat{A}$ *and* A^*.

The proof can be easily established using formulae (3.13)–(3.18). Almost all these properties are evident in Figure 3.1(a). The critical value 0.5—the largest value of A^* for which $h(\widehat{A}; A^*)$ has no positive interval—can be obtained by setting to zero the first derivative of $h(\widehat{A}; A^*)$ with respect to $\widehat{A}$ at point $\widehat{A} = 0$. The derivatives around $\widehat{A} = 0$ can be obtained using the Taylor expansion of $h(\widehat{A}; A^*)$ with respect to $\sqrt{\widehat{A}}$ (see Appendix 3.8.2).

As noted earlier, the sign of h can be used as a criterion for subset selection. In Section 3.4, Figure 3.1(a) is interpreted from a subset selection perspective.

Figure 3.1(b) shows the expected contribution curves $EC(\widehat{A}; A^*)$. The location of the maximum converges to A^* as $A^* \to \infty$, and it converges very quickly—when $A^* = 2$, the difference is almost unobservable.

3.3.4 $h(A; G)$

When deriving the H- and h-functions, we have assumed that the underlying dimensional absolute distance A^* is given. However, A^*, is, in practice, unknown—only the value of A is known. To allow the functions to be calculated, we consider A^* to be a random variable with a distribution $G(A^*)$, a distribution which is estimable from a sample of A. This is exactly the problem of estimating a mixing distribution. In our situation, the pdf of the corresponding mixture distribution has the general form

$$f(A; G) = \int_{\mathbb{R}^+} f(A; A^*)\, \mathrm{d}G(A^*), \tag{3.19}$$

where $A^* \in \mathbb{R}^+$, the nonnegative half real line. $G(A^*)$ is the mixing distribution function, $f(A; A^*)$ the pdf of the component distribution, and $f(A; G)$ the pdf of the mixture distribution. Section 3.6 discusses how to estimate the mixing distribution from a sample of A.

If the mixing distribution $G(A^*)$ is given, the h-function of A can be obtained from the following theorem.

Theorem 3.5 *Let A^* be distributed according to $G(A^*)$ and A be a random variable sampled from the mixture distribution defined in (3.19). Then*

$$h(A; G) = \int_{\mathbb{R}^+} h(A; A^*)\, \mathrm{d}G(A^*), \tag{3.20}$$

where $h(A; A^*)$ is the h-function determined by $f(A; A^*)$.

The proof follows easily from the definition of h.

No matter whether the underlying mixing distribution $G(A^*)$ is discrete or continuous, it can always be estimated by a discrete one while reaching the same k-asymptotic conclusions. Further, in practice, a discrete estimate of an arbitrary $G(A^*)$ is often easier to obtain without sacrificing prediction accuracy, and easier to manipulate for further computation. Suppose the corresponding pdf $g(A^*)$ of a discrete $G(A^*)$ is defined as

$$g(\alpha_i^*) = w_i, \text{ where } \sum_{i=1}^{m} w_i = 1. \tag{3.21}$$

Then (3.19) can be re-written as

$$f(A; G) = \sum_{i=1}^{m} w_i f(A; \alpha_i^*), \tag{3.22}$$

and (3.20) as

$$h(A;G) = \sum_{i=1}^{m} w_i h(A;\alpha_i^*). \tag{3.23}$$

Although the general forms (3.19) and (3.20) are adopted in the following analysis, it is (3.22) and (3.23) that are used in practical computations. Note that if $h(A;G)$ and $f(A;G)$ are known, the expected contribution of A given $G(A^*)$ is given by (3.12) as $h(A;G)/f(A;G)$.

Since all $\widehat{A}_j$'s of an OLS estimated model which is decomposed in an orthogonal space are statistically independent (see (3.5)), they form a sample from the mixture distribution with a discrete mixing distribution $G(A^*)$. The pdf of the mixture distribution takes the form (3.22), while the pdf of the component distribution is provided by (3.15). Likewise the h-function of the mixture has the form (3.23), while the component h-function is given by (3.18).

From now on, the mixing distribution $G(A^*)$ becomes our major concern. If the two functions $f(A;A^*)$ and $h(A;A^*)$ are well defined (as they are in OLS estimation), $G(A^*)$ uniquely determines $f(A;G)$ and $h(A;G)$ by (3.19) and (3.20) respectively. The following sections analyze the modeling process with known $G(A^*)$, show how to build the best model with known $G(A^*)$, and finally tackle the question of estimating $G(A^*)$.

3.4 The role of $H(A)$ and $h(A)$ in modeling

The H- and h-functions illuminate our understanding of the modeling process, and help with building models too. Here we use them to illustrate issues associated with the OLS subset selection procedures described in Section 2.1, and to elucidate new, hitherto unreported, phenomena. While we concentrate on OLS subset selection criteria, we touch on shrinkage methods too.

We also illustrate with examples the basis for the new procedures that are formally

defined in the next section. Not only does subset selection by thresholding variation reduction severely restrict the modeling space, but the very idea of subset selection is a limited one—when a wider modeling space is considered, better estimators emerge. Consequently we expand our horizon from subset selection to the general modeling problem, producing a final model that is not a least-squares fit at all. This improves on OLS modeling even when the projections on all dimensions are significant.

The methodology we adopt is suggested by contrasting the model-based selection problem that we have studied so far with the "dimension-based selection" that is used in principal component analysis. *Dimension-based selection* tests each orthogonal dimension independently for elimination, whereas *model-based selection* analyzes a set of orthogonal nested models in sequence (as discussed in Section 2.3, the sequence may be defined a priori or computed from the data). In dimension-based selection, deleting a dimension removes the transformed variable associated with it, and although this reduces the number of dimensions, it does not necessarily reduce the number of original variables.

Chapter 2 showed that all the OLS subset selection criteria share the idea of thresholding the amount by which the variation is reduced in each dimension. CIC sets different threshold values for each individual dimension depending on its rank in the ordering of all dimensions in the model, whereas other methods use fixed thresholds. While straightforward for dimension-based selection, this needs some adjustment in model-based selection because the variation reductions of the nested models may not be in the desired decreasing order. The necessary adjustment, if a few variables are tested, is to compare the reductions of the variation by these variables with the sum of their corresponding threshold values. The central idea remains the same.

Therefore, the key issue in OLS subset selection is the choice of threshold. We denote these schemes by OLSC(τ), where τ is the threshold (see (2.3)). The optimum value of τ is denoted by τ^*. We now consider how τ^* can be determined using the H-function, assuming that all dimensions have the same known H. We begin with dimension-based selection and tackle the nested model situation later.

As discussed earlier, our modeling goal is to minimize the expected loss. Here we consider two expected losses, the *prior expected loss* $\mathrm{E}^{\mathcal{M}}\mathrm{E}_{\mathcal{M}}[\mathcal{L}(\mathcal{M})]$ and the *posterior expected loss* $\mathrm{E}_{\mathcal{M}}[\mathcal{L}(\mathcal{M})]$—or equivalently in our context $\mathrm{E}^{A}\mathrm{E}_{A}[\mathcal{L}(\mathcal{M})]$ and $\mathrm{E}_{A}[\mathcal{L}(\mathcal{M})]$ respectively—which correspond to the situations before and after seeing the data. Note that by A here we mean all A's of the model $\mathcal{M}$. We call the corresponding optimalities *prior optimality* and *posterior optimality*. In practice, posterior optimality is usually more appropriate to use after the data is observed. However, prior optimality, without taking specific observations into account, provides a general picture about the modeling problem under investigation. More importantly, both optimalities converge to each other as the number of observations approaches infinity, i.e., k-asymptotically here. In the following theorem, we relate τ^* to prior optimality, while in the next section we see how the estimators of two optimalities converge to each other.

Theorem 3.6 *Given an orthogonal decomposition of a model space $\mathbb{M}_k$, let $\mathcal{M}^* \in \mathbb{M}_k$ be any underlying model and $\mathcal{M} \in \mathbb{M}_k$ be its estimate. Assume that all dimensional absolute distances of $\mathcal{M}$ have the same distribution function $F(A)$ and thus the same H-function. Rank the $\mathcal{A}_j(\mathcal{M})$'s in decreasing order as j increases. Then the estimator $\mathcal{M}^{\mathrm{OLSC}(\tau^*)}$ of $\mathcal{M}^*$ possesses prior optimality among all $\mathcal{M}^{\mathrm{OLSC}(\tau)}$ if and only if*

$$\tau^* = \arg\min_{A \geq 0} \; H(A). \tag{3.24}$$

Proof. For dimension j, we know from (3.8) that

$$\mathcal{L}_j(\mathcal{M}) = \mathcal{L}_j(\mathcal{M}_0) - \mathcal{C}_j(\mathcal{M}).$$

$\mathrm{OLSC}(\tau)$ discards dimension j if $\mathcal{A}_j(\mathcal{M}) \leq \tau$, so

$$\mathcal{C}_j(\mathcal{M}^{\mathrm{OLSC}(\tau)}) = \begin{cases} 0 & \text{if } \mathcal{A}_j(\mathcal{M}) \leq \tau \\ \mathcal{C}_j(\mathcal{M}) & \text{if } \mathcal{A}_j(\mathcal{M}) > \tau, \end{cases}$$

where $A = \mathcal{A}_j(\mathcal{M})$ is a random variable which is distributed according to the CDF $F(A)$. Thus

$$\begin{aligned}
&\int_{\mathbb{R}^+} \mathrm{E}_A[\mathcal{L}_j(\mathcal{M}^{\mathrm{OLSC}(\tau)})]\,\mathrm{d}F(A) \\
&= \int_{\mathbb{R}^+} \mathrm{E}_A[\mathcal{L}_j(\mathcal{M}_0)]\,\mathrm{d}F(A) - \int_{\tau}^{\infty} \mathrm{E}_A[\mathcal{C}_j(\mathcal{M})]\,\mathrm{d}F(A) \\
&= \mathcal{L}_j(\mathcal{M}_0) - H(\infty) + H(\tau).
\end{aligned}$$

Taking advantage of additivity and dimensional independence, sum the above equation over all k dimensions:

$$\mathrm{E}^A\mathrm{E}_A[\mathcal{L}(\mathcal{M}^{\mathrm{OLSC}(\tau)})] = \mathcal{L}(\mathcal{M}_0) - kH(\infty) + kH(\tau). \tag{3.25}$$

$H(\tau)$ is the only term on the right-hand side of (3.25) that varies with τ. Thus minimizing the prior expected loss is equivalent to minimizing $H(\tau)$, and *vice versa*. This completes the proof. □

Theorem 3.6 requires the dimensional absolute distances of the initially estimated model—e.g., the OLS full model—to be sorted into decreasing order. This is easily accomplished in the dimension-based situation, but not in the model-based situation. However, the nested models are usually invariably generated in a way that attempts to establish such an order. If so, this condition is approximately satisfied in practice and thus $\mathcal{M}^{\mathrm{OLSC}(\tau^*)}$ is a good approximation to the minimum prior expected loss estimator even in the model-based situation.

From Theorem 3.6, we have

Corollary 3.6.1 *Properties of* τ^*.

1. $\tau^* = \arg\min_{\mathcal{Z}\in\{\mathcal{Z}_i\}} H(\mathcal{Z})$, *where* $\{\mathcal{Z}_i\}$ *is the set of zeros of* h.
2. *If* $\tau^* > 0$, $h(\tau^*-) > 0$; *if* $\tau^* < \infty$, $h(\tau^*+) < 0$.
3. $H(\tau^*) \leq 0$ *and* $H(\infty) - H(\tau^*) \geq 0$.

4. *If there exists* A *such that* $H(\infty) - H(A) > 0$, *then* $\tau^* < \infty$.

Properties 1 and 2 show that the optimum τ^* must be a zero of h—moreover, one that separates a negative interval to the left from a positive interval to the right (unless $\tau^* = 0$ or ∞). Properties 3 and 4 narrow the set of zeros that includes the optimal value τ^*, and thus help to establish which one is the optimum.

Four examples follow. The first two illustrate Theorem 3.6 in a dimension-based situation in which each dimension is processed individually. In the first example, each dimension's underlying A^* is known—equivalently, its $h(A; A^*)$ is known. In the second, the underlying value of each dimensional absolute distance is chosen from two possibilities, and only the mixing distribution of these two values and the corresponding h-functions are known.

The last two examples introduce the ideas for building models that we will explore in the next section.

Throughout these examples, notice that the dimensional contributions are only ever used in expected-value form, and the component h-function is the OLS $h(\widehat{A}; A^*)$. Further, we always take the k-asymptotic viewpoint, implying that the distribution $G(A^*)$ can be assumed known.

Example 3.1 *Subset selection from a single mixture.* Consider the function $h(\widehat{A}; A^*)$ illustrated in Figure 3.1(a). We suppose that all dimensions have the same $h(\widehat{A}; A^*)$.

Noisy dimensions, and ones whose effects are undetectable. A noisy dimension, for which $h(\widehat{A}; 0)$ is always negative, will be eliminated from the model no matter how large its absolute distance $\widehat{A}$. Since $\lim_{A^* \to 0} h(\widehat{A}; A^*) = h(\widehat{A}; 0)$, non-redundant dimensions behave more like noisy ones as their underlying effect decreases—in other words, their contribution eventually becomes undetectable. When $A^* \leq 0.5$, any contribution is completely overwhelmed by the noise, and no subset selection procedure can detect it.

Dimensions with small vs. large effects. When the estimate resides in a negative interval of h, its contribution is negative. All hs, no matter how large their A^*, have at least one negative interval $(4A^*, \infty)$. This invalidates all subset selection schemes that eliminate dimensions based on thresholding their variation reductions with a fixed threshold, because a large estimate $\widehat{A}$ does not necessarily mean that the corresponding variable is contributive—its contribution also depends on A^*. The reason that threshold-type selection works at all is that the estimate $\widehat{A}$ in a dimension whose effect is large is less likely to fall into a negative interval than one whose effect is small.

The OLSC(τ) *criterion.* The OLS subset selection criterion OLSC(τ) eliminates dimensions whose OLS estimate falls below the threshold τ, where $\tau = 2$ for AIC, $\log n$ for BIC, $2 \log k$ for RIC, and the optimal value is τ^* as defined in (3.24). Since dimensions should be discarded based on the sign of their expected contribution, we consider three cases: dimensions with zero and small effects, those with moderate effects, and those with large effects.

When a dimension is redundant, i.e. $A^* = 0$, it should always be discarded no matter how large the estimate $\widehat{A}$. This can only be done by OLSC(τ^*), with $\tau^* = \infty$ in this case. Whenever $\tau < \infty$, dimensions whose $\widehat{A}$ exceeds τ are kept inside the model: thus a certain proportion of redundant variables are included in the final model. Dimensions with small effects behave similarly to noisy ones, and the threshold value $\tau^* = \infty$ is still best—which results in the null model.

Suppose that dimensions have moderate effects. As the value of A^* increases from zero, the value of the cumulative expected contribution $H(\tau)$ will at some point change sign. At this point, the model found by OLSC(τ), which heretofore has been better than the full model, becomes worse than it. Hence there is a value of A^* for which the predictive ability of $\mathcal{M}^{\text{OLSC}(\tau)}$ is the same as that of the full model. Furthermore, there exists a value of A^* at which the predictive ability of the null model is the same as that of the full model. In these cases, model $\mathcal{M}^{\text{OLSC}(\tau^*)}$ is either the null model or the full one, since τ^* is either 0 or ∞ depending on the

value of A^*.

When each dimension has a large effect—large enough that the position of the second zero of $h(\widehat{A}; A^*)$ is at least τ—any OLSC(τ) with fixed $\tau > 0$ will inevitably eliminate contributive dimensions. This means that the full model is a better one than $\mathcal{M}^{\text{OLSC}(\tau)}$. Furthermore, OLSC($\tau^*$) with $\tau^* = 0$ will always choose the full model, which is the optimal model for every $\mathcal{M}^{\text{OLSC}(\tau)}$.

Shrinkage methods in orthogonal space. In orthogonal regression, when $X'X$ is a diagonal matrix, contribution functions help explain why shrinkage methods work. These methods shrink the parameter values of OLS models and use smaller values than the OLS estimates. This may or may not change the signs of the OLS estimated parameters; however, for orthogonal regressions, the signs of the parameters are left unchanged. In this situation, therefore, shrinking parameters is tantamount to shrinking the $\widehat{A}$'s. Ridge regression shrinks all the parameters while NN-GAROTTE and LASSO shrink the larger parameters and zero the smaller ones.

When A^* is small, it is possible to choose a shrinkage parameter that will shrink $\widehat{A}$'s that lie between A^* and $4A^*$ to around A^*, and shrink the negative contributions outside $4A^*$ to become positively contributive—despite the fact that $\widehat{A}$ around the maximum point A^* are shrunk to smaller values. This may give the resulting model lower predictive error than any model selected by OLSC(τ), including $\tau = \tau^*$. Zeroing the smaller $\widehat{A}$'s by NN-GAROTTE and LASSO does not guarantee better predictive accuracy than ridge regression, for these dimensions might be contributive. When A^* is large, shrinkage methods perform badly because the distribution of $\widehat{A}$ tends to be sharper around A^*. This is why OLS subset selection often does better in this situation.

Example 3.2 *Subset selection from a double mixture.* Suppose

$$h(\widehat{A}) = \frac{k_1}{k} h(\widehat{A}; \alpha_1^*) + \frac{k_2}{k} h(\widehat{A}; \alpha_2^*), \tag{3.26}$$

where $k = k_1 + k_2$. For k_1 dimensions the underlying A^* is α_1^*, while for the

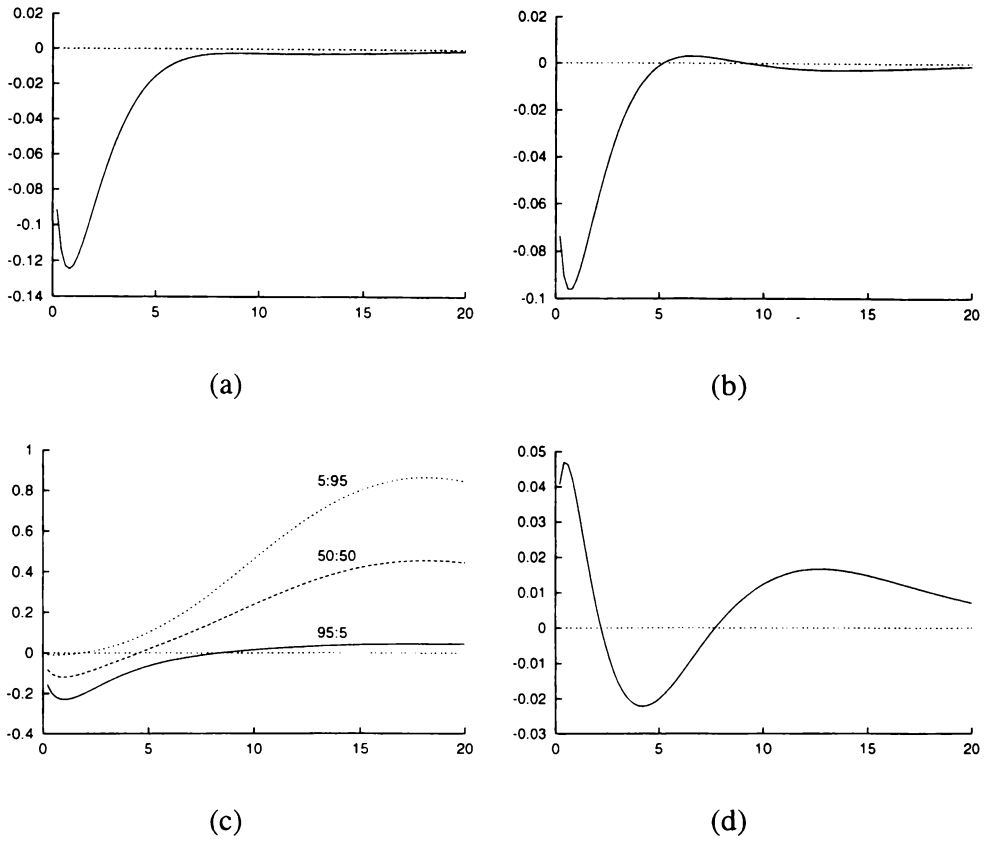


Figure 3.2: $h(\widehat{A}) = \frac{k_1}{k} h(\widehat{A}; \alpha_1^*) + \frac{k_2}{k} h(\widehat{A}; \alpha_2^*)$. (a). $\alpha_1^* = 0, \alpha_2^* = 3, k_1 : k_2 = 80 : 20$. (b). $\alpha_1^* = 0, \alpha_2^* = 3, k_1 : k_2 = 75 : 25$. (c). $\alpha_1^* = 0, \alpha_2^* = 20$, $k_1 : k_2$ are respectively $5 : 95, 50 : 50, 95 : 5$. (d). α_1^* and α_2^* are carefully set with fixed $k_1 : k_2 = 95 : 5$ such that $H(\mathcal{Z}_3) = 0$.

remaining k_2 dimensions it is α_2^*. $\widehat{A}$ is an observation sampled from the mixture distribution $f(\widehat{A}) = \frac{k_1}{k} f(\widehat{A}; \alpha_1^*) + \frac{k_2}{k} f(\widehat{A}; \alpha_2^*)$. Altering the values of k_1, α_1^*, k_2 and α_2^* yields the different hs illustrated in Figure 3.2. The optimal threshold τ^* of OLSC(τ) is discussed.

In Figure 3.2(a), where $\alpha_1^* = 0$, $\alpha_2^* = 3$ and $k_1 : k_2 = 80 : 20$, no positive interval exists despite the fact that there are 20 non-redundant dimensions. This is because the effect of all the non-redundant dimensions is overwhelmed by the noisy ones. In principle, no matter how large its effect, any dimension can be overwhelmed by a sufficient number of noisy ones. In this case $\tau^* = \infty$ and OLSC(τ^*) selects the null model.

In Figure 3.2(b), which is obtained from the previous situation by altering $k_1 : k_2$ to $75 : 25$, there is a positive interval. But $H(\infty)$ is the minimum of all zeros, so that τ^* remains ∞ and the model chosen by OLSC(τ^*) is still the null one. The contributive dimensions are still submerged by the noisy ones.

If a finite threshold value τ^* exists, it must satisfy $H(\infty) - H(\tau^*) > 0$ (Property 4 of Corollary 3.6.1). It can take on any nonnegative value by adjusting the four parameters k_1, α_1^*, k_2 and α_2^*. Figure 3.2(c) shows three functions h, obtained by setting α_1^* and α_2^* to 0 and 20 respectively and making the ratio between k_1 and k_2 $5 : 95$, $50 : 50$, and $95 : 5$. As these curves show, the corresponding values of τ^* are about 2, 4 and 8.

In Figure 3.2(d), $k_1 : k_2 = 95 : 5$ and α_1^* and α_2^* are set to make $H(\mathcal{Z}_3) = 0$, where $\mathcal{Z}_3$ is the third zero of h. In this case, there are two possibilities for τ^*: the origin $\mathcal{Z}_1$, and $\mathcal{Z}_3$. $\mathcal{Z}_3$ gives a simpler model. Notice that the number of parameters of the two models is in the approximate ratio $5 : 100$. However, the balance is easily broken—for example, if α_1^* increases slightly, then there is a single value 0 for τ^*. Although the larger model has slightly smaller predictive error than the smaller one, it is much more complex. Here is a situation where a far more succinct model can be obtained with a small sacrifice in predictive accuracy.

Example 3.3 *Subset selection based on the sign of h.* Although OLSC(τ^*) is optimal among every OLSC(τ), it has limitations. It always deletes dimensions whose $\widehat{A}$'s fall below the threshold, and retains the remaining ones. Thus it may delete dimensions that lie in the positive intervals of h, and retain ones in the negative intervals. We know that the sign of h is the sign of the expected contribution, and the selection can be improved by using this fact: we simply retain dimensions whose $h(\widehat{A})$ is positive and discard the remainder.

In the next section we formalize this idea and prove that its predictive accuracy always improves upon OLSC(τ^*). Because the positive and negative intervals of h can lie anywhere along the half real line, this procedure may retain dimensions with smaller $\widehat{A}$ and discard ones with larger $\widehat{A}$. Or it may delete a dimension whose

$\widehat{A}$ lies between those of the other dimensions. For example, in Figure 3.2(b), the new procedure will keep all the dimensions whose $\widehat{A}$'s lie within the small positive interval, despite the fact that OLSC(τ^*) chooses the null model.

Example 3.4 *General modeling.* So far we have only discussed subset selection, where the estimate $\widehat{A}$ is either altered to zero or remains the same. A natural question is whether better results might be obtained by relaxing this constraint—and indeed they can be. For example, in Example 3.1, where all dimensions are noisy, OLSC(τ^*) deletes them all and chooses the null model. This in effect replaces the estimate $\widehat{A}$ by the underlying A^*, which is 0 in this case. Similarly, when all estimates $\widehat{A}$ have the same underlying A^* (which is non-zero), and all $\widehat{A}$'s are updated to A^*, the estimated model improves significantly—even though no dimension is redundant.

A similar thing happens when A is sampled from a mixture distribution. In Figure 3.2(a), the h-function of the mixture has no positive interval, although 20 out of 100 dimensions have $A^* = 3$. The best that subset selection can do is to discard all dimensions. However, a dimension with $\widehat{A} = 20$—despite $h(\widehat{A})$ being less than 0—is unlikely to be a redundant one; it is more likely to belong to the group for which $A^* = 3$. Altering its OLS estimate $\widehat{A}$ from 20 to 3 is likely to convert the dimension into a contributive one.

The next section formulates a formal estimation procedure based on this idea.

3.5 Modeling with known $G(A^*)$

We now assume that the underlying $G(A^*)$ is known and define a group of six procedures, collectively called *pace regression*, which build models by adjusting the orthogonal projections based on estimations of the expected dimensional contributions. They are denoted $\text{PACE}_1, \text{PACE}_2, \ldots, \text{PACE}_6$, and the model produced by PACE_i is written $\mathcal{M}^{\text{PACE}_i}$. We will discuss in the next section how to estimate $G(A^*)$.

$G(A^*)$ is the distribution function of a random variable A^* that represents the dimensional absolute distance of the underlying model $\mathcal{M}^*$. These distances, denoted by $A_1^*, \ldots, A_k^*$, form a sample of size k from $G(A^*)$. When the mixing distribution $G(A^*)$ is known, (3.19) can be used to obtain the mixture pdf $f(A; G)$ from the component distribution pdf $f(A; A^*)$. From this, along with the component h-function $h(A; A^*)$, the functions $h(A; G)$ and $H(A; G)$ can be found from (3.20) and (3.12) respectively. As Theorem 3.3 shows, OLS estimation provides the expressions for both $f(A; A^*)$ and $h(A; A^*)$.

The pace procedures consist of two steps. The first, which is the same for all procedures, generates an initial model $\mathcal{M}$. We always use the OLS full model $\widehat{\mathcal{M}}$ for this, although any initial model can be used so long as the component distribution and the component h-function are available. Decomposing $\widehat{\mathcal{M}}$ in a given orthogonal space yields the model's dimensional absolute distances, say $\widehat{A}_1, \ldots, \widehat{A}_k$. These are in fact a sample from a mixture distribution $F(\widehat{A}; G)$ with known component distribution $F(\widehat{A}; A^*)$ and mixing distribution $G(A^*)$. In the second step, the final model is generated from either (3.37) or (3.38), where $\widetilde{A}_1, \ldots, \widetilde{A}_k$ are obtained by updating $\widehat{A}_1, \ldots, \widehat{A}_k$. The new procedures differ in how the updating is done.

To characterize the resulting performance, we define a class of estimators and show that pace estimators are optimal within this class, or a subclass of it. The class of estimators $\mathfrak{M}_k$ (where k is the number of orthogonal dimensions) is as follows: given the initial model $\widehat{\mathcal{M}}$ and its absolute distances in an orthogonal decomposed model space, every member of $\mathfrak{M}_k$ is an updating of $\widehat{\mathcal{M}}$ by (3.37) and *vice versa*, where the updating is entirely dependent on the set of absolute distances of $\widehat{\mathcal{M}}$. Clearly, $\mathcal{M}^{\text{OLSC}(\tau)} \in \mathfrak{M}_k$ for any τ.

Various corollaries below establish that the pace estimators are better than others. Each estimator's optimality is established by a theorem that applies to a specific subclass, and proving each corollary reduces to exhibiting an example where the performance is actually better. These examples are easily obtainable from the illustrations in Section 3.4. When we say that one estimator is "better than" (or "equivalent to") another, we mean in the sense of posterior expected loss. With one

exception, better estimators have lower prior expected loss (the exception, as noted below, is Corollary 3.8.1, which compares PACE_1 and PACE_3). Refer to Section 3.4 for definitions of prior and posterior expected losses.

Of the six procedures, PACE_1 and PACE_2 perform model-based selection, that is, they select a subset model from a sequence. Procedures PACE_3 and PACE_4 address the dimension-based situation, where each orthogonal dimension is tested and selected (or not) individually. If these procedures are used for a sequence of orthogonal nested models, the resulting model may not belong to the sequence. The last two procedures, PACE_5 and PACE_6, are not selection procedures. Instead, they update the absolute distances of the estimated model to values chosen appropriately from the nonnegative half real line.

Theorem 3.6 shows that there is an optimal threshold for threshold-type OLS subset selection that minimizes the prior expected loss. We use this idea for nested models.

Procedure 1 (PACE_1). *Given a sequence of orthogonal nested models, let* $\tau = \arg\min_{\mathcal{Z}\in\{\mathcal{Z}_i\}} H(\mathcal{Z})$, *where* $\{\mathcal{Z}_i\}$ *is the set of zeros of* h. *Output the model in the sequence selected by* $\text{OLSC}(\tau)$.

According to Corollary 3.6.1, Procedure 1 finds the optimal threshold τ^*. Therefore $\text{PACE}_1 = \text{OLSC}(\tau^*)$ (and $\mathcal{M}^{\text{PACE}_1} = \mathcal{M}^{\text{OLSC}(\tau^*)}$). Since the sequence of dimensional absolute distances of the model $\widehat{\mathcal{M}}$ is not necessarily always decreasing, $\mathcal{M}^{\text{OLSC}(\tau^*)}$ is not guaranteed to be the minimum prior expected loss estimator. However, in practice these distances do generally decrease, and so in the nested model situation $\mathcal{M}^{\text{OLSC}(\tau^*)}$ is an excellent approximation to the minimum prior expected loss estimator. In this sense, PACE_1 is superior to other OLS subset selection procedures—OLS, AIC, BIC, RIC, and CIC—for these do not use the optimal threshold τ^*. In particular, the procedure CIC uses a threshold that depends on the number of variables in the subset model as well as the total number of variables. The relative performance of these procedures depends on the particular experiments used for comparison, since datasets exist for which any selection criterion's threshold coincides with the optimal value τ^*.

Instead of approximating the optimum as PACE_1 does, PACE_2 always selects the optimal model from a sequence of nested models, where optimality is in the sense of posterior expected loss.

Procedure 2 (PACE_2). *Among a sequence of orthogonal nested models, output the one which has the largest value of* $\sum_{i=1}^{j} h(\widehat{A}_i)/f(\widehat{A}_i)$ *for* $j = 1, 2, \ldots, k$.

Theorem 3.7 *Given* $G(A^*)$, $\mathcal{M}^{\text{PACE}_2}$ *has the smallest posterior expected loss in a subclass of* $\mathfrak{M}_k$ *in which each estimator can only select from the sequence of orthogonal nested models that is provided.*

Proof. Let $\mathcal{M}_j$ be the selected model, then $\widetilde{A}_i = 0$ if $i > j$ and $\widetilde{A}_i = \widehat{A}_i$ if $i \leq j$. From the definition of dimensional contribution (3.8), we have

$$\text{E}_{\{\widehat{A}_i\}}[\mathcal{L}(\mathcal{M}_j)] = \mathcal{L}(\mathcal{M}_0) - \sum_{i=1}^{j} EC(\widehat{A}_i). \tag{3.27}$$

Because $EC(\widehat{A}_i) = h(\widehat{A}_i)/f(\widehat{A}_i)$ by (3.12), minimizing $\text{E}_{\{\widehat{A}_i\}}[\mathcal{L}(\mathcal{M}_j)]$ is equivalent to maximizing $\sum_{i=1}^{j} h(\widehat{A}_i)/f(\widehat{A}_i)$ with respect to j. This completes the proof. □

Corollary 3.7.1 *Given* $G(A^*)$ *and a sequence of orthogonal nested models,* $\mathcal{M}^{\text{PACE}_2}$ *is a better estimator than* $\mathcal{M}^{\text{OLSC}(\tau)}$ *for any* τ. *This includes* $\mathcal{M}^{\text{PACE}_1}$, $\mathcal{M}^{\text{OLS}}$, $\mathcal{M}^{\text{AIC}}$, $\mathcal{M}^{\text{BIC}}$, $\mathcal{M}^{\text{RIC}}$ *and* $\mathcal{M}^{\text{CIC}}$.

Since, according to Section 2.6, the cross-validation subset selection procedure $\text{CV}(d) =_n \text{OLSC}((2n - d)/(n - d))$ and the bootstrap subset selection procedure $\text{BS}(m) =_n \text{OLSC}((n + m)/m)$, we have

Corollary 3.7.2 *Given* $G(A^*)$ *and a sequence of orthogonal nested models,* n*-asymptotically* $\mathcal{M}^{\text{PACE}_2}$ *is a better estimator than* $\mathcal{M}^{\text{CV}(d)}$ *for any* d *and* $\mathcal{M}^{\text{BS}(m)}$ *for any* m.

In fact, the difference between the models generated by PACE_1 and PACE_2 is small, because we have

Corollary 3.7.3 *Given $G(A^*)$, if the elements of $\{\widehat{A}_j; j = 1, \ldots, k\}$ are in decreasing order as j increases, $\mathcal{M}^{\text{PACE}_1} =_k \mathcal{M}^{\text{PACE}_2}$ a.s.*

Proof. Since $\{\widehat{A}_j; j = 1, \ldots, k\}$ are in decreasing order as j increases, $\mathcal{M}^{\text{PACE}_2}$ is in effect the model that minimizes $\int_{\mathbb{R}^+} \mathrm{E}_{\widehat{A}}[\mathcal{L}(\mathcal{M}_j)]\, \mathrm{d}F_k(\widehat{A})$ with respect to j, where $F_k(\widehat{A})$ is the Kolmogorov empirical CDF of $\widehat{A}$, and $\mathcal{M}^{\text{PACE}_1}$ is the model that minimizes $\int_{\mathbb{R}^+} \mathrm{E}_{\widehat{A}}[\mathcal{L}(\mathcal{M}_j)]\, \mathrm{d}F(\widehat{A})$ with respect to j. Since, almost surely, $F_k(\widehat{A}) \to F(\widehat{A})$ uniformally as $k \to \infty$ (Glivenko-Cantelli theorem), it implies that, almost surely, $\int_{\mathbb{R}^+} \mathrm{E}_{\widehat{A}}[\mathcal{L}(\mathcal{M}_j)]\, \mathrm{d}F_k(\widehat{A}) \to \int_{\mathbb{R}^+} \mathrm{E}_{\widehat{A}}[\mathcal{L}(\mathcal{M}_j)]\, \mathrm{d}F(\widehat{A})$ as $k \to \infty$, due to the Helly-Bray theorem (see, e.g., Galambos, 1995). Therefore, minimizing the prior expected loss is equivalent to minimizing the posterior expected loss, k-asymptotically. We complete the proof with $\mathcal{M}^{\text{PACE}_1} =_k \mathcal{M}^{\text{PACE}_2}$ a.s. □

These two procedures show how to select the best model in a sequence of $k + 1$ nested models. However, no model sequence can guarantee that the optimal model is one of the nested models. Thus we now consider dimension-based modeling, where the final model can be a combination of any dimensions. With selection, the number of potential models given the projections on k dimensions is as large as 2^k. When the orthogonal basis is provided by a sequence of orthogonal nested models, this kind of selection means that the final model may not be one of the nested models, and its parameter vector may not be the OLS fit in terms of the original variables.

Procedure 3 (PACE_3). *Let $\tau = \arg\min_{\mathcal{Z} \in \{\mathcal{Z}_i\}} H(\mathcal{Z})$, where $\{\mathcal{Z}_i\}$ is the set of zeros of h. Set $\widetilde{A}_j = 0$ if $\widehat{A}_j \leq \tau$; otherwise $\widetilde{A}_j = \widehat{A}_j$. Output the model determined by $\{\widetilde{A}_1, \ldots, \widetilde{A}_k\}$.*

Theorem 3.8 *Given $G(A^*)$, $\mathcal{M}^{\text{PACE}_3}$ is the minimum prior expected loss estimator of $\mathcal{M}^*$ in an estimator class which is the subclass of $\mathfrak{M}_k$ in which every estimator is determined by $\{\widetilde{A}_1, \ldots, \widetilde{A}_k\}$ where $\widetilde{A}_j \in \{0, \widehat{A}_j\}$ for all j.*

The proof is omitted; it is similar to that of Theorem 3.6.

Since $\mathcal{M}^{\text{PACE}_1}$ is the model determined by $\{\widetilde{A}_1, \ldots, \widetilde{A}_k\}$ where $\widetilde{A}_j$ is either $\widehat{A}_j$ or 0 depending on whether the associated variable is included or not, this estimator belongs to the class described in Theorem 3.8. This gives the following corollary.

Corollary 3.8.1 *Given $G(A^*)$, $\mathcal{M}^{\text{PACE}_3}$ is a better estimator (in the sense of prior expected loss) than $\mathcal{M}^{\text{PACE}_1}$.*

The difference between the models generated by PACE_1 and PACE_3 is also small, because

Corollary 3.8.2 *Given $G(A^*)$, if the elements of $\{\widehat{A}_j; j = 1, \ldots, k\}$ are in decreasing order as j increases, $\mathcal{M}^{\text{PACE}_1} = \mathcal{M}^{\text{PACE}_3}$.*

As we have seen, whether or not a dimension is contributive is indicated by the sign of the corresponding h-function. This leads to the next procedure.

Procedure 4 (PACE_4). *Set $\widetilde{A}_j = 0$ if $h(\widehat{A}_j) \leq 0$; otherwise $\widetilde{A}_j = \widehat{A}_j$. Output the model determined by $\{\widetilde{A}_1, \ldots, \widetilde{A}_k\}$.*

PACE_4 does not rank dimensions in order of absolute distance and eliminate those with smaller distances, as do conventional subset selection procedures and the preceding pace procedures. Instead, it eliminates dimensions that are not contributive in the estimated model irrespective of the magnitude of their dimensional absolute distance. It may eliminate a dimension with a larger absolute distance than another dimension that is retained. (In fact the other procedures may end up doing this occasionally, but they do so only because of incorrect ranking of variables.)

Theorem 3.9 *Given $G(A^*)$, $\mathcal{M}^{\text{PACE}_4}$ has the smallest posterior expected loss in the subclass of $\mathfrak{M}_k$ in which every estimator is determined by $\{\widetilde{A}_1, \ldots, \widetilde{A}_k\}$ where $\widetilde{A}_j \in \{0, \widehat{A}_j\}$ for all j.*

The proof is omitted; it is similar to that of Theorem 3.7.

Because the estimator class defined in Theorem 3.9 covers the classes defined in Theorems 3.7 and 3.8,

Corollary 3.9.1 *Given $G(A^*)$, $\mathcal{M}^{\text{PACE}_4}$ is a better estimator than $\mathcal{M}^{\text{PACE}_2}$ and $\mathcal{M}^{\text{PACE}_3}$.*

PACE_1, PACE_2, PACE_3 and PACE_4 are all selection procedures: each updated dimensional absolute distance of the estimated model must be either 0 or $\widehat{A}_j$. The optimal value of $\widetilde{A}_j$ is often neither of these. If the possible values are chosen from $\mathbb{R}^+$ instead, the best updated estimate $\widetilde{A}_j$ is the one that maximizes the expected contribution of the jth dimension given $\widehat{A}_j$ and $G(A^*)$. The optimality is achieved over an uncountably infinite set of potential models. This relaxation can improve performance dramatically even when there are no noisy dimensions.

Procedure 5 (PACE_5). *Output the model determined by $\{\widetilde{A}_1, \ldots, \widetilde{A}_k\}$, where*

$$\widetilde{A}_j = \arg\max_{A \in \mathbb{R}^+} \int_{\mathbb{R}^+} \frac{h(A; A^*)}{f(A; A^*)} f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*). \tag{3.28}$$

Theorem 3.10 *Given $G(A^*)$, $\mathcal{M}^{\text{PACE}_5}$ has the smallest posterior expected loss of all estimators in $\mathfrak{M}_k$.*

Proof. Each OLS estimate $\widehat{A}_j$ is an observation sampled from the mixture pdf $f(\widehat{A}; G)$ determined by the component pdf $f(\widehat{A}; G)$ and the mixing distribution $G(A^*)$. If $\widehat{A}_j$ is replaced by any $A \in \mathbb{R}^+$, the expected contribution of A given $\widehat{A}_j$ and $G(A^*)$ is

$$EC(A; \widehat{A}_j, G) = \frac{\int_{\mathbb{R}^+} EC(A; A^*) f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)}{\int_{\mathbb{R}^+} f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)}. \tag{3.29}$$

Since $EC(A; A^*) = h(A; A^*)/f(A; A^*)$ from (3.12) and $\int_{\mathbb{R}^+} f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)$ is constant for every A, the integration on the right-hand side of (3.28) actually maximizes over the expected contribution of A. From (3.8), this is equivalent to

minimizing the expected loss in the jth dimension. Because all dimensions are independent, the posterior expected loss of the updated model given the set $\{\widehat{A}_j\}$ and $G(A^*)$ is minimized. □

Corollary 3.10.1 *Given $G(A^*)$, $\mathcal{M}^{\text{PACE}_5}$ is a better estimator than $\mathcal{M}^{\text{PACE}_4}$.*

This motivates a new shrinkage method: shrink the magnitude (or the sum of the magnitude) of the orthogonal projections of the model $\widehat{\mathcal{M}}$. This is equivalent to updating the OLS estimate $\widehat{\mathcal{M}}$ to a model that satisfies $\widetilde{A}_j \leq \widehat{A}_j$ for every j. Since all shrinkage estimators of this type obviously yield a member of $\mathfrak{M}_k$, $\mathcal{M}^{\text{PACE}_5}$ is a better estimator than any of them.

Although we do not pursue this direction because of the optimality of $\mathcal{M}^{\text{PACE}_5}$, this helps us to understand other shrinkage estimators. Models produced by shrinkage methods in the literature—ridge regression (including ridge regression for subset selection), NN-GAROTTE and LASSO—do not necessarily require an orthogonal space and hence do not always belong to this subclass, and so we cannot show that the new estimator is superior to them in general. However, in the important special case of orthogonal regression, when the column vectors of X are taken as the orthogonal axis, the models produced by all these shrinkage methods do belong to this subclass (see Example 3.1). Therefore,

Corollary 3.10.2 *Given $G(A^*)$, $\mathcal{M}^{\text{PACE}_5}$ is a better estimator for orthogonal regression than $\mathcal{M}^{\text{RIDGE}}$, $\mathcal{M}^{\text{NN-GAROTTE}}$ and $\mathcal{M}^{\text{LASSO}}$.*

A general explicit solution to (3.28) does not seem to exist. Rather than resorting to numerical techniques, however, a good approximate solution is available. Considering that

$$\frac{h(A;A^*)}{f(A;A^*)} = \frac{c(\sqrt{A};\sqrt{A^*})p(\sqrt{A};\sqrt{A^*}) + c(-\sqrt{A};\sqrt{A^*})p(-\sqrt{A};\sqrt{A^*})}{p(\sqrt{A};\sqrt{A^*}) + p(-\sqrt{A};\sqrt{A^*})}, \quad (3.30)$$

the dominant part on the right-hand side is $c(\sqrt{A};\sqrt{A^*})p(\sqrt{A};\sqrt{A^*})$—and its dominance increases dramatically as A^* increases. Replacing $h(A;A^*)/f(A;A^*)$ in (3.28) by $c(\sqrt{A};\sqrt{A^*})$, we obtain the following approximation to PACE$_5$.

Procedure 6 (PACE$_6$). *Output the model determined by* $\{\widetilde{A}_1,\ldots,\widetilde{A}_k\}$ *where*

$$\widetilde{A}_j = \arg\max_{A\in\mathbb{R}^+} \int_{\mathbb{R}^+} c(\sqrt{A};\sqrt{A^*}) f(\widehat{A}_j;A^*)\,\mathrm{d}G(A^*). \tag{3.31}$$

Equation (3.31) can be solved by setting the first derivative of the right-hand side to zero, resulting in

$$\widetilde{A}_j = \left[\frac{\int_{\mathbb{R}^+}\sqrt{A^*} f(\widehat{A}_j;A^*)\,\mathrm{d}G(A^*)}{\int_{\mathbb{R}^+} f(\widehat{A}_j;A^*)\,\mathrm{d}G(A^*)}\right]^2. \tag{3.32}$$

In (3.28), (3.31) and (3.32) the true distribution $G(A^*)$ is discrete (as in (3.21)), so they become respectively

$$\widetilde{A}_j = \arg\max_{A\in\mathbb{R}^+} \sum_{i=1}^{m} \frac{h(A;\alpha_i^*)}{f(A;\alpha_i^*)} f(\widehat{A}_j;\alpha_i^*) w_i, \tag{3.33}$$

$$\widetilde{A}_j = \arg\max_{A\in\mathbb{R}^+} \sum_{i=1}^{m} c(\sqrt{A};\sqrt{\alpha_i^*}) f(\widehat{A}_j;\alpha_i^*) w_i, \tag{3.34}$$

and

$$\widetilde{A}_j = \left[\frac{\sum_{i=1}^m \sqrt{\alpha_i^*} f(\widehat{A}_j;\alpha_i^*) w_i}{\sum_{i=1}^m f(\widehat{A}_j;\alpha_i^*) w_i}\right]^2. \tag{3.35}$$

The following loose bound can be obtained for the increased posterior expected loss suffered by the PACE$_6$ approximation.

Theorem 3.11

$$0 \le \mathrm{E}[\mathcal{L}_j(\mathcal{M}^{\mathrm{PACE}_6})|\widehat{A}_j, G(A^*)] - \mathrm{E}[\mathcal{L}_j(\mathcal{M}^{\mathrm{PACE}_5})|\widehat{A}_j, G(A^*)] < 2e^{-1}. \tag{3.36}$$

Proof. See Appendix 3.8.3.

Appendix 3.8.3 actually obtains the tighter bound $4\sqrt{\widetilde{A}_j^{\text{PACE}_6}\alpha^*}e^{-2\sqrt{\widetilde{A}_j^{\text{PACE}_6}\alpha^*}}$, where α^* is the support point of the distribution $G(A^*)$ that maximizes $c(\sqrt{\widetilde{A}_j^{\text{PACE}_6}}; \sqrt{A^*}) - EC(\widetilde{A}_j^{\text{PACE}_6}; A^*)$. This bound rises from zero at the origin in terms of $\widetilde{A}_j^{\text{PACE}_6}\alpha^*$, achieves the maximum $2e^{-1}$ at the point $\widetilde{A}_j^{\text{PACE}_6}\alpha^* = 0.5$, and thereafter drops exponentially to zero as $\widetilde{A}_j^{\text{PACE}_6}\alpha^*$ increases. It follows that the increased posterior expected loss caused by approximating PACE_6 is usually close to zero.

Remarks

All these pace procedures adjust the magnitude of the orthogonal projections of the OLS estimate $\widehat{\mathcal{M}}$, based on an estimate of the expected dimensional contributions. Among them, PACE_5 and PACE_6 go the furthest: each projection of $\widehat{\mathcal{M}}$ onto the orthogonal axis can be adjusted to any nonnegative value and the adjusted value achieves (or approximately achieves) the greatest expected contribution, corresponding to the minimum posterior expected loss. These two procedures can shrink, retain or even expand the values of the absolute dimensional distances. Surprising though it may sound, increasing a zero distance to a much higher value can improve predictive accuracy.

Of the six pace procedures, PACE_2, PACE_4 and PACE_6 are most appropriate for practical applications. PACE_6 generates a very good approximation to the model from PACE_5, which is the best of the six procedures. Procedure PACE_2 chooses the best member of a sequence of subset models that is provided to it, which is useful if prior information dictates the sequence of subset models. PACE_1 and PACE_3 involve numerical integration and have higher (posterior) expected loss than other procedures. PACE_4, which is a lower expected loss procedure than PACE_2, is useful for dimension-based subset selection.

3.6 The estimation of $G(A^*)$

Now it is time to consider how to estimate $G(A^*)$ from $\widehat{A}_1, \widehat{A}_2, \ldots, \widehat{A}_k$, which are the dimensional absolute distances of the OLS estimate $\widehat{\mathcal{M}}$. Once this is accomplished, the procedures described in the last section become fully defined by replacing the true $G(A^*)$, which we assumed in the last section was known, with the estimate. The estimation of $G(A^*)$ is an independent step in these modeling procedures, and can be investigated independently. It critically influences the quality of the final model—better estimates of $G(A^*)$ give better estimators for the underlying model.

$\widehat{A}_1, \widehat{A}_2, \ldots, \widehat{A}_k$ are actually a sample from a mixture distribution whose component distribution $F(\widehat{A}; A^*)$ is known and whose mixing distribution is $G(A^*)$. (Strictly speaking, this sample is taken without replacement. However, this is asymptotically the same as sampling with replacement, and so does not affect our theoretical results since they are be established in the asymptotic sense.) Estimating $G(A^*)$ from data points $\widehat{A}_1, \widehat{A}_2, \ldots, \widehat{A}_k$ is tantamount to estimating the mixing distribution. Note that the mixture here is a countable one—the underlying $G(A^*)$ has support at $A_1^*, A_2^*, \ldots, A_k^*$, and the number of support points is unlimited as $k \to \infty$. Chapter 5 tackles this problem in a general context.

The following theorem guarantees that if the mixing distribution is estimated sufficiently well, the pace regression procedures continue to enjoy the various properties proved above in the limit of large k.

Theorem 3.12 *Let $\{G_k(A^*)\}$ be a sequence of CDF estimators. If $G_k(A^*) \to_w G(A^*)$ a.s. as $k \to \infty$, and the known $G(A^*)$ is replaced by the estimator $G_k(A^*)$, Theorems 3.6–3.11 (and all their corollaries) hold k-asymptotically a.s.*

Proof. According to the Helly-Bray theorem, $G_k(A^*) \to_w G(A^*)$ a.s. as $k \to \infty$ implies the almost sure pointwise convergence of all the objective functions used in these theorems (and their corollaries) to the underlying corresponding functions, because these functions are continuous. This further implies the almost sure conver-

gence of the optimal values and of the locations where these optima are achieved, as $k \to \infty$. This completes the proof. □

All of the above results utilize the loss function $||y_{\mathcal{M}} - y^*||^2/\sigma^2$. However, our real interest is $||y_{\mathcal{M}} - y^*||^2$. Therefore we need the following corollary.

Corollary 3.12.1 *If the loss function $||y_{\mathcal{M}} - y^*||^2/\sigma^2$ is replaced by $||y_{\mathcal{M}} - y^*||^2$ in Theorems 3.6–3.11 (and all their corollaries),*

1. *Theorem 3.12 continues to hold, if σ^2 is known;*
2. *Theorem 3.12 holds almost surely as $n \to \infty$ if σ^2 is replaced with an n-asymptotically strongly consistent estimator.*

It is well known that both the unbiased OLS estimator $\hat{\sigma}^2$ (for $(n - k) \to \infty$ as $n \to \infty$) and the biased maximum likelihood estimator (for $k/n \to 0$) are n-asymptotically strongly consistent.

In view of Theorem 3.12, any estimator of the mixing distribution is able to provide the desired theoretic results in the limit if it is *strongly consistent* in the sense that, almost surely, it converges weakly to the underlying mixing distribution as $k \to \infty$. From Chapter 5, the maximum likelihood estimator and a few minimum distance estimators are known to be strongly consistent. If any of these estimators are used to obtain $G_k(A^*)$, Theorem 3.12 is secured. This, finally, closes our circle of analysis.

However, it is pointed out in Chapter 5 that all the minimum distance methods except the nonnegative-measure-based one suffer from a serious defect in a finite-sample situation: they may completely ignore small numbers of data points in the estimated mixture, no matter how distant they are from the dominant data points. This severely impacts their use in our modeling procedures, because the value of one dimensional absolute distance is frequently quite different to all the others—and this implies that the underlying absolute distance has a very high probability of being different too. In addition, the maximum likelihood approach, which does

not seem to have this minority cluster problem, can be used for pace regression if computational cost is not an issue.

For all these estimators, the following three conditions, due to Robbins (1964), must be satisfied in order to ensure strong consistency (see also Section 5.3.1).

(C1) $F(x;\theta)$ is continuous on $\mathcal{X} \times \Theta$.

(C2) Define $\mathcal{G}$ to be the class of CDFs on Θ. If $F_{G_1} = F_{G_2}$ for $G_1, G_2 \in \mathcal{G}$, then $G_1 = G_2$.

(C3) Either Θ is a compact subset of $\mathbb{R}$, or $\lim_{\theta \to \pm\infty, \theta \in \Theta} F(x;\theta)$ exists for each $x \in \mathcal{X}$ and is not a distribution function on $\mathcal{X}$.

Pace regression involves mixtures of $\chi^2_1(A^*/2)$, where A^* is the mixing parameter. Conditions C1 and C3 are clearly satisfied. C2, the identifiability condition, is verified in Appendix 3.8.4.

3.7 Summary

This chapter explores and formally presents a new approach to linear regression. Not only does this approach yield accurate prediction models, it also reduces model dimensionality. It outperforms other modeling procedures in the literature, in the sense of k-asymptotics and, as we will see in Chapter 6, it produces satisfactory results in simulation studies for finite k.

We have limited our investigation to linear models with normally distributed noise, but the ideas are so fundamental that we believe they will soon find application in other realms of empirical modeling.

3.8 Appendix

3.8.1 Reconstructing model from updated absolute distances

Once final estimates for the absolute distances in each dimension have been found, the model needs to be reconstructed from them. Consider how to build a model from a set of absolute distances, denoted by $A_1, \ldots, A_k$. Let $\alpha = (t_1\sqrt{A_1}, \ldots, t_k\sqrt{A_k})'$, where t_j is either $+1$ or -1 depending on whether or not the jth projection of the prediction vector $y_{\widehat{\mathcal{M}}}$ has the same direction as the orthogonal base b_j. This choice of t_j's value is based on the fact that the projections of $y_{\widehat{\mathcal{M}}}$ and $y_{\mathcal{M}^*}$ are most likely in the same direction—i.e., any other choice would degrade the estimate.

Our estimate of the parameter vector is

$$\beta = (X'X)^{-1}X'B\alpha\sigma, \tag{3.37}$$

where B is a column matrix formed from the bases $b_1, \ldots, b_k$. For example, if $\{A_1, \ldots, A_k\}$ are the OLS estimates of the absolute distances in the corresponding dimensions, (3.37) gives β the value of the OLS estimate $\widehat{\beta}$.

It may be that not all the A_j's are available, but a prediction vector is known that corresponds to all missing A_j's. This situation will occur if some dimensions are forced to be in the final model—for example, the constant term, or dimensions that give very great reductions in variation (very large A_j's). Suppose the number of dimensions with known A_j's is k', and call the overall prediction vector for the remaining $k - k'$ dimensions y_{rest}. Then the estimated parameter vector is

$$\beta = (X'X)^{-1}X'(y_{\text{rest}} + B\alpha\sigma), \tag{3.38}$$

where B is an $n \times k'$ matrix and α a k'-vector.

The estimation of β from A_j's is fully described by (3.37) and (3.38). However, in practice the computation takes a different, more efficient, route. Once the $n \times k$

approximation equation of the original least-squares problem has been orthogonally transformed, finding the least squares solution reduces to solving a matrix equation

$$U\beta = d, \tag{3.39}$$

where U is a $k \times k$ upper-triangular matrix and d is a k-vector (Lawson and Hanson, 1974, 1995). As a matter of fact, the square of the jth element in d is exactly the OLS estimate $\widehat{A}_j\sigma^2$. When a new set of estimates, say $\widetilde{A}_j$ $(j = 1, \ldots, k)$, is obtained, the corresponding estimate of β^* is the solution of (3.39) with the jth element in d replaced by $\sqrt{\widetilde{A}_j}\sigma$ without changing sign. If not all $\widehat{A}_j$'s are known, so that (3.38) is used instead of (3.37), only dimensions with known $\widehat{A}_j$'s are replaced.

3.8.2 The Taylor expansion of $h(A; A^*)$ with respect to $\sqrt{A}$

Let $a = \sqrt{A} \geq 0$ and $a^* = \sqrt{A^*} \geq 0$. Denote $\psi = 1/\sqrt{2\pi}$. Because

$$c(a; a^*) = a^{*2} - (a - a^*)^2 = 2aa^* - a^2$$

and

$$p(a; a^*) = \psi e^{-\frac{(a-a^*)^2}{2}} = \psi e^{\frac{2aa^*-a^2}{2}} e^{-\frac{a^{*2}}{2}}$$

hence

$$\begin{aligned}
&\frac{c(a; a^*)p(a; a^*)}{\psi e^{-\frac{a^{*2}}{2}}} \\
= \;& (2a^* - a)ae^{\frac{2a^*-a}{2}a} \\
= \;& (2a^* - a)a + \frac{(2a^* - a)^2}{2}a^2 + \frac{(2a^* - a)^3}{8}a^3 + O(a^4) \\
= \;& 2a^*a + (-1 + 2a^{*2})a^2 + (-2a^* + a^{*3})a^3 + O(a^4)
\end{aligned}$$

Similarly

$$c(-a;a^*) = a^{*2} - (a+a^*)^2 = -2aa^* - a^2$$

and

$$p(-a;a^*) = \psi e^{-\frac{(a+a^*)^2}{2}} = \psi e^{\frac{-a^2-2aa^*}{2}} e^{-\frac{a^{*2}}{2}}$$

hence

$$\begin{aligned}
&\frac{c(-a;a^*)p(-a;a^*)}{\psi e^{-\frac{a^{*2}}{2}}} \\
&= (-2a^* - a)ae^{\frac{-2a^*-a}{2}a} \\
&= -(2a^* + a)a + \frac{(2a^*+a)^2}{2}a^2 - \frac{(2a^*+a)^3}{8}a^3 + O(a^4) \\
&= -2a^*a + (-1 + 2a^{*2})a^2 + (2a^* - a^{*3})a^3 + O(a^4)
\end{aligned}$$

Therefore

$$\begin{aligned}
\frac{h(A;A^*)}{\psi e^{-\frac{a^{*2}}{2}}} &= \frac{c(a;a^*)p(a;a^*) + c(-a;a^*)p(-a;a^*)}{2a\psi e^{-\frac{a^{*2}}{2}}} \\
&= (-1 + 2a^{*2})a + O(a^3)
\end{aligned}$$

that is,

$$h(A;A^*) = \frac{1}{\sqrt{2\pi}} e^{-\frac{A^*}{2}} [(-1 + 2A^*)\sqrt{A} + O(A^{\frac{3}{2}})]. \tag{3.40}$$

3.8.3 Proof of Theorem 3.11

To prove Theorem 3.11, we need a simple lemma.

Lemma 1 *For any* $A \geq 0$ *and* $A^* \geq 0$,

$$0 \leq c(\sqrt{A};\sqrt{A^*}) - EC(A;A^*) < 4\sqrt{AA^*}e^{-4\sqrt{AA^*}} \leq 2e^{-1}. \tag{3.41}$$

Proof. For every $A \geq 0$ and $A^* \geq 0$, $c(\sqrt{A}; \sqrt{A^*}) \geq 0$ and $c(-\sqrt{A}; \sqrt{A^*}) \leq 0$, hence

$$\begin{aligned}
& c(\sqrt{A}; \sqrt{A^*}) \\
= & \frac{c(\sqrt{A}; \sqrt{A^*})p(\sqrt{A}; \sqrt{A^*}) + c(\sqrt{A}; \sqrt{A^*})p(-\sqrt{A}; \sqrt{A^*})}{p(\sqrt{A}; \sqrt{A^*}) + p(-\sqrt{A}; \sqrt{A^*})} \\
\geq & \frac{c(\sqrt{A}; \sqrt{A^*})p(\sqrt{A}; \sqrt{A^*}) + c(-\sqrt{A}; \sqrt{A^*})p(-\sqrt{A}; \sqrt{A^*})}{p(\sqrt{A}; \sqrt{A^*}) + p(-\sqrt{A}; \sqrt{A^*})} \\
= & EC(A; A^*),
\end{aligned}$$

and

$$\begin{aligned}
& c(\sqrt{A}; \sqrt{A^*}) - EC(A; A^*) \\
= & \frac{[c(\sqrt{A}; \sqrt{A^*}) - c(-\sqrt{A}; \sqrt{A^*})]p(-\sqrt{A}; \sqrt{A^*})}{p(\sqrt{A}; \sqrt{A^*}) + p(-\sqrt{A}; \sqrt{A^*})} \\
= & \frac{4\sqrt{AA^*}p(-\sqrt{A}; \sqrt{A^*})}{p(\sqrt{A}; \sqrt{A^*}) + p(-\sqrt{A}; \sqrt{A^*})} \\
< & \frac{4\sqrt{AA^*}p(-\sqrt{A}; \sqrt{A^*})}{p(\sqrt{A}; \sqrt{A^*})} \\
= & 4\sqrt{AA^*}e^{-2\sqrt{AA^*}} \\
\leq & 2e^{-1},
\end{aligned}$$

thus completing the proof of the lemma. □

Proof of Theorem 3.11. According to the definition of dimensional contribution, to prove Theorem 3.11 we need to show that

$$0 \leq EC(\widetilde{A}_j^{\text{PACE}_5}; \widehat{A}_j, G(A^*)) - EC(\widetilde{A}_j^{\text{PACE}_6}; \widehat{A}_j, G(A^*)) < 2e^{-1}, \tag{3.42}$$

where $G(A^*)$, no matter continuous or discrete, is the mixing distribution function used in both procedures. The first inequality in (3.42) is obvious because $\widetilde{A}_j^{\text{PACE}_5}$ is the optimal solution. For the second inequality, from the above lemma and the fact

that $\widetilde{A}_j^{\text{PACE}_6}$ is the optimal solution of (3.34), we have

$$\begin{aligned}
& EC(\widetilde{A}_j^{\text{PACE}_5}; \widehat{A}_j, G(A^*)) \\
= & \frac{\int_{\mathbb{R}^+} EC(\widetilde{A}_j^{\text{PACE}_5}; A^*) f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)}{\int_{\mathbb{R}^+} f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)} \\
\leq & \frac{\int_{\mathbb{R}^+} c(\sqrt{\widetilde{A}_j^{\text{PACE}_5}}; \sqrt{A^*}) f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)}{\int_{\mathbb{R}^+} f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)} \\
\leq & \frac{\int_{\mathbb{R}^+} c(\sqrt{\widetilde{A}_j^{\text{PACE}_6}}; \sqrt{A^*}) f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)}{\int_{\mathbb{R}^+} f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)}.
\end{aligned}$$

Hence

$$\begin{aligned}
& EC(\widetilde{A}_j^{\text{PACE}_5}; \widehat{A}_j, G(A^*)) - EC(\widetilde{A}_j^{\text{PACE}_6}; \widehat{A}_j, G(A^*)) \\
\leq & \frac{\int_{\mathbb{R}^+} \{c(\sqrt{\widetilde{A}_j^{\text{PACE}_6}}; \sqrt{A^*}) - EC(\widetilde{A}_j^{\text{PACE}_6}; A^*)\} f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)}{\int_{\mathbb{R}^+} f(\widehat{A}_j; A^*)\, \mathrm{d}G(A^*)}.
\end{aligned}$$

According to the lemma, $c(\sqrt{\widetilde{A}_j^{\text{PACE}_6}}; \sqrt{A^*}) - EC(\widetilde{A}_j^{\text{PACE}_6}; A^*) \geq 0$ for every A^*, and let

$$\alpha^* = \underset{\{A^*\}}{\arg\max}\ c(\sqrt{\widetilde{A}_j^{\text{PACE}_6}}; \sqrt{A^*}) - EC(\widetilde{A}_j^{\text{PACE}_6}; A^*),$$

where $\{A^*\}$ is the set of support points of $G(A^*)$. Therefore,

$$\begin{aligned}
& EC(\widetilde{A}_j^{\text{PACE}_5}; \widehat{A}_j, G(A^*)) - EC(\widetilde{A}_j^{\text{PACE}_6}; \widehat{A}_j, G(A^*)) \\
\leq & c(\sqrt{\widetilde{A}_j^{\text{PACE}_6}}; \sqrt{\alpha^*}) - EC(\widetilde{A}_j^{\text{PACE}_6}; \alpha^*) \\
< & 4\sqrt{\widetilde{A}_j^{\text{PACE}_6} \alpha^*} e^{-2\sqrt{\widetilde{A}_j^{\text{PACE}_6} \alpha^*}} \\
\leq & 2e^{-1},
\end{aligned}$$

which finishes the proof of Theorem 3.11. □

3.8.4 Identifiability for mixtures of $\chi_1^2(A^*/2)$ distributions

Although we have been unable to locate the following theorem in the literature, it seems unlikely to be original. Note that here, identifiability applies only to the situation where the mixing function is limited to being a CDF. Note that the identifiability of mixtures using CDFs as mixing functions implies the identifiability of mixtures using any finite nonnegative functions as mixing functions (Lemma 4 in Section 5.4.3), as required by (5.19).

Theorem 3.13 *The mixture of $\chi_1^2(A^*/2)$ distributions, where A^* is the mixing parameter, is identifiable.*

Proof. Let $P(x;\mu)$ be the CDF of the distribution $N(\mu,1)$, and $F(A;A^*)$ be the CDF of the distribution $\chi_1^2(A^*/2)$. From the definition of the $\chi_1^2(A^*/2)$ distribution and the symmetry of the normal distribution,

$$\begin{aligned} F(A;A^*) &= P(\sqrt{A};\sqrt{A^*}) - P(-\sqrt{A};\sqrt{A^*}) \\ &= P(\sqrt{A};\sqrt{A^*}) + P(\sqrt{A};-\sqrt{A^*}) - 1, \end{aligned} \tag{3.43}$$

where $P(\sqrt{A};\sqrt{A^*})$ is the CDF of $N(\sqrt{A^*},1)$ and $P(\sqrt{A};-\sqrt{A^*})$ is the CDF of $N(-\sqrt{A^*},1)$. Therefore, if the mixture of $F(A;A^*)$ were unidentifiable, the mixture of normal distributions where the mean μ is the mixing parameter would be unidentifiable. Clearly this contradicts the well-known identifiability result for mixtures of normal distributions (Teicher, 1960), thus completing the proof. □

Chapter 4

Discussion

4.1 Introduction

Several issues related to pace regression deserve further discussion. Some are necessary to complete the definition of the pace regression procedures in special situations. Others expand their implications into a broader arena, while yet others raise interesting open questions.

4.2 Finite k *vs.* k-asymptotics

We have seen that the pace regression procedures are optimal in a k-asymptotic sense. Larger numbers of variables tend to produce estimators that are closer to optimal. If there are only a few candidate variables, pace regression will not necessarily outperform other methods. Since k is inevitably finite in practice, it is worth expanding on this.

The central idea of pace regression is to decompose the prediction vector of an estimated model into k orthogonal components, and then adjust each according to aggregated magnitude information from all components. The more diverse the magnitudes of the different components, the less they can inform the adjustment of

any particular one. If one component's magnitude differs greatly from that of all others, there is little basis on which to alter its value.

Pace regression shines when many variables have similar effects—a common special situation is when many variables have zero effect. As the effects of the variables disperse, pace regression's superiority over other procedures fades. When the effect of each variable is isolated from that of all others, the pace estimator is exactly the OLS one. In principle, the worst case is when no improvement over OLS is possible.

Although pace regression is k-asymptotically optimal, this does not mean that increasing the number of candidate variables necessarily improves prediction. New contributive variables should increase predictive accuracy, but new redundant variables will decrease it. Pre-selection of variables based on background knowledge will always help modeling, if suitable variables are selected.

4.3 Collinearity

Pace regression, like almost any other linear regression procedure, fails when presented with (approximately) collinear variables. Hence collinearity should be detected and handled before applying the procedure. We suggest eliminating collinearity by discarding variables. The number of candidate variables k should be reduced accordingly, because collinearity does not provide new independent dimensions, a prerequisite of pace regression. In other words, the model has the same degrees of freedom without collinearity as with it. Appropriate variables can be identified by examining the matrix $X'X$, or QR-transforming X (see, e.g., Lawson and Hanson, 1974, 1995; Dongarra et al., 1979; Anderson et al., 1999).

Note that OLS subset selection procedures are sometimes described as a protection against collinearity. However, the fact is that none of these automatic procedures can reliably eliminate collinearity, for collinearity does not necessarily imply that the projections of the vector y are small.

4.4 Regression for partial models

The full OLS model forms the basis for pace regression. In some situations, however, the full model may be unavailable. For example, there may be too many candidate variables—perhaps more than the number of observations. Although the largest available partial model can be substituted for the full model, this causes practical difficulties.

The clustering step in pace regression must take *all* candidate variables into account. This is possible so long as the statistical test used to determine the initial partial model can supply an approximate distribution for the dimensional absolute distances of the variables that do not participate in it. For example, the partial-F test may be used to discard variables based on forward selection; typically, dimensional absolute distances are smaller for the discarded variables than for those used in the model. It seems likely that a sufficiently accurate approximate distribution for the dimensional absolute distances of the discarded variables can be derived for this test, though further investigation is necessary to confirm this.

Instead of providing an approximate distribution for the discarded variables, it is also possible to estimate the mixing distribution directly by assuming that the effects of these variables are located in an interval of small values. Estimation of the mixing distribution can then proceed as usual, provided the boundaries of fitting intervals are all chosen to be outside this interval. Our implementation adopts this approach.

In many practical applications of data mining, some kind of *feature selection* (see, e.g. Liu and Motoda, 1998; Hall, 1999) is performed before a formal modeling procedure is invoked, which is helpful when there are too many features (or variables) available for the procedure to handle computationally. However, it is generally not acknowledged that bias is introduced by discarding variables without passing relevant information on to the modeling procedure—though admittedly most modeling procedures cannot make use of this kind of information.

Estimating the noise variance, should it be unknown a priori, is another issue that

is affected when only a partial initial model is available. The noise component will contain the effects of all variables that are not included in the partial model. Moreover, because of competition between variables, the OLS estimate of σ^2 from the partial model is biased downwards. How to compensate for this is an interesting topic worthy of further investigation.

4.5 Remarks on modeling principles

As noted in Section 1.2, pace regression measures the success of modeling using two separate principles. The primary one is accuracy, or the minimization of expected loss. The secondary one is parsimony, or a preference for the smallest model, and is only used when it does not conflict with the first—that is, to decide between several models that have about the same accuracy. The secondary principle is essential if dimensionality reduction is of interest. For example, if the support points found by the clustering step are not all zero, none of the upgraded $\widetilde{A}_j$'s by PACE_5 and PACE_6 will be zero. But many may be tiny, and eliminating tiny $\widetilde{A}_j$ has negligible influence on predictive ability.

In the following, we discuss four widely-accepted general principles of modeling by fitting linear models. When the goal is to minimize the expected loss, the (k-asymptotic) superiority of pace regression casts doubt on these principles. In fact, so does the existing empirical Bayes methodology, including Stein estimation—however, it does not seem to have been applied to fitting linear models in a way that is directly evaluated based on the loss function; see the review in Section 2.8.

First, pace regression challenges the general least squares principle—or perhaps more precisely, the unbiasedness principle. All six pace procedures outperform OLS estimation: PACE_1 and PACE_3 in the sense of prior expected loss and the remainder in the sense of posterior expected loss. According to the Gauss-Markov Theorem, OLS yields a "best linear unbiased estimator" (or BLUE). OLS's inferiority, however, is due precisely to the unbiasedness constraint, which fails to utilize all the infor-

mation implicit in the data. It is well known that biased estimators such as subset selection and shrinkage can outperform the OLS estimator in particular situations. We have shown that pace regression estimators, which are also biased, outperform OLS in all situations. In fact, in modern statistical decision theory, the requirement of unbiased estimation is usually discarded.

Second, pace regression challenges the maximum likelihood principle. In the linear regression situation, the maximum likelihood estimator is exactly the same as the OLS one.

Third, some uses of Bayes' rule come under attack. The Bayesian estimator BIC is threshold-based OLS subset selection, where the a priori density is used to determine the threshold. Yet the six pace regression procedures equal or outperform the best OLS subset selection estimator. This improvement is not based on prior information—the general validity of which has long been questioned—but on hitherto unexploited information that is implicit in the very same data. Furthermore, when the non-informative prior is used—and the use of the non-informative prior is widely accepted, even by many non-Bayesians—the Bayes estimator is the same as the maximum likelihood estimator, i.e., the OLS estimator, and therefore inferior to the pace regression estimators. Although *hierarchical Bayes* (also known as *Bayes empirical Bayes*), which involves some higher level, subjective prior, can produce similar results to empirical Bayes (see, e.g., Berger, 1985; Lehmann and Casella, 1998), both differ essentially in whether to utilize prior information or data information for the distribution of true parameters (or models).

Fourth, questions arise concerning complexity-based modeling. According to the minimum description length (MDL) principle (Rissanen, 1978), the best model is the one that minimizes the sum of the model complexity and the data complexity given the model. In practice the first part is an increasing function of the number of parameters required to define the model, while the second is the resubstitution error. Our analysis and experiments do not support this principle. We have found—see the examples and analysis in Sections 3.4 and 3.5 and the experiments in Chapter 6—that pace regression may choose models of the same or even larger size, and with

larger resubstitution errors, than those of other procedures, yet gives much smaller prediction errors on independent test sets. In addition, the MDL estimator derived by Rissanen (1978) is the same as the BIC estimator, which has already been shown inferior to the pace regression estimators in terms of our prediction criterion.

4.6 Orthogonalization selection

Our analysis assumes that the dimensional models associated with the axes in an orthogonal space are independent. This statement, however, is worth further examination. When a suitable orthogonalization is given before seeing the data (or more precisely, the y values), this is clearly true. The problem is that when the orthogonalization is generated from the data, say, by a partial-F test, it corresponds to a selection from a group of candidate orthogonalizations. Such selection will definitely affect the pace estimators (and many other types of estimator which employ orthogonalization).

The effect of selecting orthogonalization can be shown by extreme examples. After eliminating collinearity (see Section 4.3), we could always find an orthogonalization such that $\widehat{A}_1 = \widehat{A}_2 = \cdots = \widehat{A}_k$, in which case the updating of the PACE_5 and PACE_6 estimators will choose the same values of the OLS estimates. Or, possibly, these $\widehat{A}_j$ values could be adjusted by selecting an appropriate orthogonal basis to be "distributed" like a uni-component χ^2 sample, therefore PACE_5 and PACE_6 will update them to the center of this distribution. Generally speaking, such updating will not achieve the goal of better estimation, because it manipulates the data too much.

Despite these phenomena, it is not suggested that the procedures are only useful when an orthogonalization is provided in advance. Rather, employing the data to set up an orthogonal basis can help to eliminate redundant variables. This phenomenon resembles the issue of model selection—here it is just *orthogonalization selection*—because both involve manipulating the data and thus affect the estimation. The

selection of an orthogonalization is a competition with uncertainty factors in which the winner is the one that best satisfies the selection criterion, e.g., the partial-F test.

Orthogonalization selection influences the proposed estimators, yet its effect remains unclear to us. Although experiments with pace regression produce satisfactory results (see Chapter 6) without taking this effect into account, the theoretical analysis of pace regression is incomplete when the orthogonalization is based upon the data. We leave this problem open: it is a future extension of pace estimation.

Despite the optimality of pace regression given an orthogonal basis, different orthogonal systems result in different estimated models. This obviously contradicts the invariance principle. We consider this question as follows.

An orthogonal system, like any other assumption used in empirical modeling, can be pre-determined by employing prior information (background knowledge) or determined from data. Which of these is appropriate depends on knowing (or assuming) which helps the modeling most. An orthogonal system that is natural in an application is not necessarily the best to employ. In situations when there are many redundant variables, the data-driven approach based on the partial-F test seems to be a good choice.

4.7 Updating signed $\hat{a}_j$?

All proposed pace procedures concern the mixture of $\widehat{A}_j$'s, or the estimation of $G(A^*)$. An interesting question arises: is it possible, or indeed better, to estimate and use the mixing distribution of the mixture of signed $\hat{a}_j$, where the sign of each $\hat{a}_j$ is determined by the direction of each orthogonal axis? Hence each $\hat{a}_j$ is independent and normally distributed with mean a_j^*. Once the mixing distribution, say, $G(a^*)$, is obtained from the given $\hat{a}_j$'s through an estimation procedure, each $\hat{a}_j$ could be updated in a similar way to the proposed pace estimators. Note that the sign of $\hat{a}_j$ here is different from that defined in Section 3.3.1. Both $\hat{a}_j$ and a_j^* are signed values here, while previously a_j^* is always nonnegative.

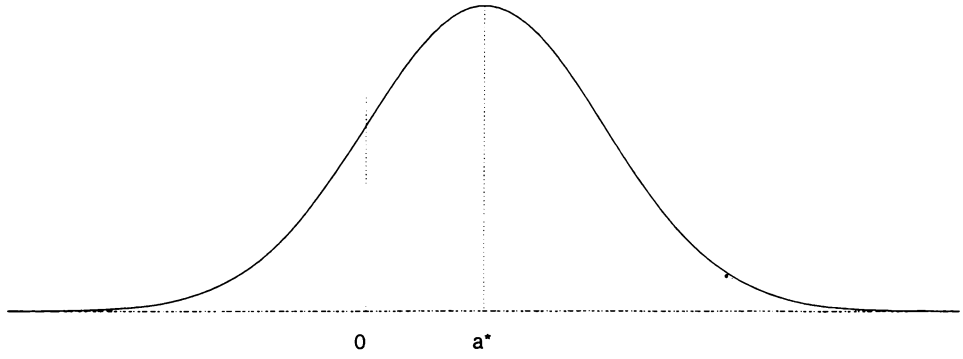


Figure 4.1: Uni-component mixture

Employing only the unsigned values implies that any potential influence of the signs is shielded out of the estimation, no matter whether it improves or worsens the estimate. Therefore, it is worth examining situations where signs help and where they hinder. At first glance, using $G(a^*)$ rather than $G(A^*)$ seems to produce better estimates, since it employs more information—information about the sign as well as the magnitude. Nevertheless, our answer to this question is negative: it is possible but *generally* not as good. Consider the following examples.

Start from a simple, ideal situation in which the magnitude of all projections are equal. Further, assume that the directions of the orthogonal axes are chosen so that the signs of $a_1^*, a_2^*, \ldots, a_k^*$ are the same, i.e., $a_1^* = a_2^* = \cdots = a_k^* = a^* (> 0$, without loss of generality). Then the mixture is a uni-component normal distribution with mean a^*, as shown in Figure 4.1. The best estimator using $G(a^*)$—k-asymptotically equivalent to using a strongly consistent estimator of $G(a^*)$—can ideally update all $\hat{a}_j$ to a^*, resulting in zero expected loss. In contrast, however, the estimator based on $G(A^*)$ will only update the magnitude, but not the sign, i.e., the data on the negative half real line will keep the same negative sign with the updated magnitude. Obviously in this case the estimator based on $G(a^*)$ should be better than the other one. Note this is an ideal situation. Also, for finite sample, the improvement only takes effect for the data points around the origin.

The second example involves asymmetric mixtures. Since in practice we are unlikely to be so fortunate that all a_j^* have the same sign, consider a less ideal situa-

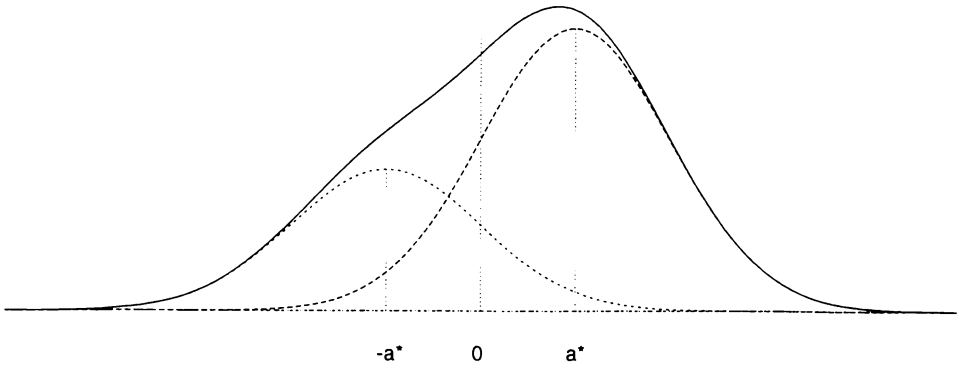


Figure 4.2: Asymmetric mixture

tion: there are more positive a_j^* than negative a_j^*. In Figure 4.2, a mixture of two components is displayed, where two-thirds of the a_j^*'s are set to a^* and the rest to $-a^*$. In this asymmetric case, the advantage of using $G(a^*)$ over using $G(A^*)$ decreases, because the ratio of data points that change sign after updating decreases. Further, in the finite sample situation, the value of a^* estimated from the set $\{\hat{a}_j\}$ is less accurate than from the set $\{\widehat{A}_j\}$, since the same number of data points are more sparsely distributed when using $\{\hat{a}_j\}$ than when using $\{\widehat{A}_j\}$; see also Section 4.2.

More importantly, the assumption of an asymmetric mixture implies the existence of correlation between the determined axial directions and the projections of the true response vector. This is equivalent to employing prior information, whose effect, as always, may upgrade or deteriorate the estimate, depending on how correct it is. For example, if we know beforehand that all the true parameters are positive, we might be able to determine an orthogonal basis that takes this information into account, thus improving estimation. However, more generally, we have no idea whether the directions of the chosen basis are somehow correlated with the directions of the true projections. Without this kind of prior information, we can often expect to obtain a symmetric mixture, as shown in Figure 4.3. Choosing axial directions without any implication about the projection directions is equivalent in that the axial directions are randomly chosen.

In the case of symmetric $G(a^*)$, updating involves only updating the magnitude, not the sign, just as when using $G(A^*)$. However, a better estimate of $G(A^*)$ than

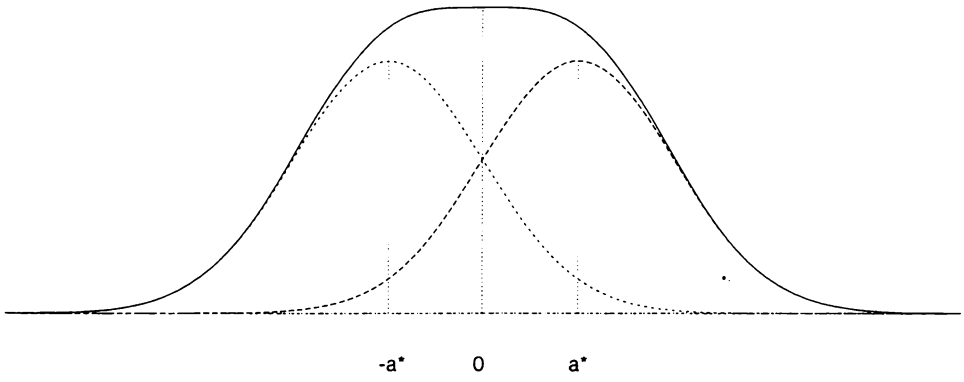


Figure 4.3: Symmetric mixture

of $G(a^*)$ is obtained from the same number of data points, because they are more closely packed together. Furthermore, random assignment of signs implies extra, imposed uncertainty, thus deteriorating estimation.

In the above, only one single magnitude value is considered, but the idea extends to situations with multiple or even continuous magnitude values. Without useful prior information to suggest the choice of axial directions, one would expect a symmetric mixture to be obtained. In this case, it is better to employ $G(A^*)$ than $G(a^*)$.

Chapter 5

Efficient, reliable and consistent estimation of an arbitrary mixing distribution

5.1 Introduction

A common practical problem is to fit an underlying statistical distribution to a sample. In some applications, this involves estimating the parameters of a single distribution function—e.g. the mean and variance of a normal distribution. In others, an appropriate mixture of elementary distributions must be found—e.g. a set of normal distributions, each with its own mean and variance. Pace regression, as pointed out in Section 3.6, concerns mixtures of noncentral χ_1^2 distributions.

In many situations, the cumulative distribution function (CDF) of a *mixture distribution* (or *model*) has the form

$$F_G(x) = \int_\Theta F(x;\theta)\, dG(\theta), \tag{5.1}$$

where $\theta \in \Theta$, the parameter space, and $x \in \mathcal{X}$, the sample space. This gives the CDF of the mixture distribution $F_G(x)$ in terms of two more elementary distributions: the *component distribution* $F(x;\theta)$ and the *mixing distribution* $G(\theta)$. The

former has a single unknown parameter θ,[1] while the latter gives a CDF for θ. For example, $F(x;\theta)$ might be the normal distribution with mean θ and unit variance, where θ is a random variable distributed according to $G(\theta)$.

The mixing distribution $G(\theta)$ can be either continuous or discrete. In the latter case, $G(\theta)$ is composed of a number of mass points, say, $\theta_1, \ldots, \theta_k$ with masses $w_1, \ldots, w_k$ respectively, satisfying $\sum_{i=1}^{k} w_i = 1$. Then (5.1) can be re-written as

$$F_G(x) = \sum_{i=1}^{k} w_i F(x;\theta_i), \tag{5.2}$$

each mass point providing a component, or cluster, in the mixture with the corresponding weight, where these points are also known as the *support points* of the mixture. If the number of components k is finite and known *a priori*, the mixture distribution is called *finite*; otherwise it is treated as *countably infinite*. The qualifier "countably" is necessary to distinguish this case from the situation with continuous $G(\theta)$, which is also infinite.

The main topic in mixture models—which is also the problem that we will address in this chapter—is the estimation of $G(\theta)$ from sampled data that are independent and identically distributed according to the unknown distribution $F_G(x)$. We will focus on the estimation of an *arbitrary* mixing distribution, i.e., the true $G(\theta)$ "is treated completely unspecified as to whether it is discrete, continuous or in any particular family of distributions; this will be called *nonparametric mixture model*" (Lindsay, 1995, p.8).

For tackling this problem, two main approaches exist in the literature (see Section 5.2), the *maximum likelihood* and *minimum distance approaches*, although the maximum likelihood approach can also be viewed as a special minimum distance approach that uses the Kullback–Leibler distance (Titterington et al., 1985). It is generally believed that the maximum likelihood approach should produce (slightly) better estimation through solving complicated nonlinear equations, while the (usual)

[1] In fact, θ could be a vector, but we only focus on the simplest, univariate situation, which is all that is needed for pace regression.

minimum distance approach, which only requires solving linear equations with linear constraints, offers the advantage of computational efficiency. This advantage makes the minimum distance approach to be the main interest in our work, because fitting linear models can be embedded in a large modeling system and required repeatedly.

The minimum distance approach can be further categorized based on the distance measure used, for example, *CDF-based*, *pdf-based*, etc. As we show in more detail in Section 5.5, existing minimum distance methods all suffer from a serious drawback in finite-data situations (the *minority cluster problem*): small outlying groups of data points can be completely ignored in the clusters that are produced. To rectify this, a new minimum distance method using a nonnegative measure and the idea of local fitting is proposed, which solves the problem while remaining as computationally efficient as other minimum distance methods. Before proposing this method, we generalize the CDF-based method and introduce the probability-measure-based method for reasons of completeness. Theoretical results of strong consistency of the proposed estimators will also be established. *Strong consistency* here means that, almost surely, the estimator sequences converge weakly to any given $G(\theta)$ as the sample size approaches infinity, i.e.,

$$\Pr(\lim_{n\to\infty} G_n(\theta) = G(\theta), \theta \text{ any continuity point of } G) = 1. \tag{5.3}$$

The minority cluster phenomenon seems to have been overlooked, presumably for three reasons: small amounts of data may be assumed to represent a small loss; a few data points can easily be dismissed as outliers; and in the limit the problem evaporates because the estimators are strongly consistent. However, often these reasons are inappropriate: the loss function may be sensitive to the distance between clusters; the small number of outlying data points may actually represent small, but important, clusters; and any practical clustering situation will necessarily involve finite data. Pace regression is such an example (see Chapter 3).

Note that the maximum likelihood approach (reviewed in Subsection 5.2.1) does

not seem to have this problem. The overall likelihood function is the product of each individual likelihood function, hence any single point that is not in the "neighborhood" of the resulting clusters will make the product zero, which is obviously not the maximum.

Finally, it is worth mentioning that although our application involves estimating the mixing distribution in an empirical modeling setting, there are many other applications. Lindsay (1995) provides an extensive discussion, including known component densities, the linear inverse problem, random effects models, repeated measures models, latent class and latent trait models, missing covariates and data, random coefficient regression models, empirical and hierarchical Bayes, nuisance parameter models, measurement error models, de-convolution problems, robustness and contamination models, over-dispersion and heterogeneity, clustering, etc. Many of these subjects "have [a] vast literature of their own." Books by Titterington et al. (1985), McLachlan and Basford (1988) also provide extensive treatment in considerable depth about general and specific topics in mixture models.

In this chapter we frequently use the term "clustering" for the estimation of a mixing distribution. In our context, both terms have a similar meaning, although in the literature they have slightly different connotations.

The structure of this chapter is as follows. Existing estimation methods for mixing distributions are briefly reviewed in Section 5.2. Section 5.3 generalizes the CDF-based approach, and develops a general proof of the strong consistency of the mixing distribution estimator. Then this proof is adapted for measure-based estimators, including the probability-measure-based and nonnegative-measure-based ones, in Section 5.4. Section 5.5 describes the minority cluster problem and illustrates with experiments how the new method overcomes it. Simulation studies on predictive accuracy of these minimum distance procedures in the case of overlapping clusters are given in Section 5.6.

5.2 Existing estimation methods

In this section, we briefly review the existing methods of the two main approaches to the estimation of a mixing distribution: the maximum likelihood and minimum distance approaches. These methods work generally for all types of component distribution.

The Bayesian approach and its variations are not included, since this methodology is usually used to control the number of resulting support points, or to mitigate the effect of outliers, which in our context does not appear to be a problem.

5.2.1 Maximum likelihood methods

The likelihood function for an independent sample from the mixture distribution has the form

$$L(G) = \prod_{i=1}^{n} L_i(G), \tag{5.4}$$

where $L_i(G)$ has the integral form $\int L_i(\theta) dG(\theta)$. Lindsay's (1995) monograph provides an excellent coverage of the theory, geometry, computation and applications of maximum likelihood (ML) estimation; see also Böhning (1995) and Lindsay and Lesperance (1995).

The idea of finding the nonparametric maximum likelihood estimator (NPMLE) of a mixing distribution was originated by Robbins (1950) in an abstract. It was later substantially developed by Kiefer and Wolfowitz (1956), who provided conditions that ensure consistency of the ML estimator.

Algorithmic computation of the NPMLE, however, only began twenty years later. Laird (1978) proved that the NPMLE is a step function with no more than n steps, where n is the sample size. She also suggested using the EM algorithm (Dempster et al., 1977) to find the NPMLE. An implementation is provided by DerSimo-

nian (1986, 1990). Since the EM algorithm often converges very slowly, this is a not popular method for finding the NPMLE (although it has applications in estimating finite mixtures).

Gradient-based algorithms are more efficient than EM-based ones. Well-known examples include the *vertex direction method* (VDM) and its variations (e.g., Fedorov, 1972; Wu, 1978a,b; Böhning, 1982; Lindsay, 1983), the *vertex exchange method* (VEM) (Böhning, 1985, 1986; Böhning et al., 1992), the *intra-simplex direction method* (ISDM) (Lesperance and Kalbfleisch, 1992), and the *semi-infinite programming method* (SIP) (Coope and Watson, 1985; Lesperance and Kalbfleisch, 1992). Lesperance and Kalbfleisch (1992) point out that ISDM and SIP are stable and very fast—in one of their experiments, both need only 11 iterations to converge, in contrast to 2177 for VDM and 143 for VEM.

The computation of the NPMLE has been significantly improved within the last twenty-five years. Nevertheless, due to the essence of nonlinear optimization, the computational cost of all these ML methods increases dramatically as the number of final support points increases, and in practice this number can be as large as the sample size. The minimum distance approach is more efficient, since often it only involves a linear or quadratic mathematical programming problem.

5.2.2 Minimum distance methods

The idea of the minimum distance method is to define some measure of the goodness of the clustering and optimize this by suitable choice of a mixing distribution $G_n(\theta)$ for a sample of size n. We generally want the estimator to be strongly consistent as $n \to \infty$, in the sense defined in Section 5.1, for an arbitrary mixing distribution. We also want to take advantage of any special structure of mixtures to come up with an efficient algorithmic solution.

We begin with some notation. Let $x_1, \ldots, x_n$ be a sample chosen according to the mixture distribution, and suppose (without loss of generality) that the sequence is

ordered so that $x_1 \leq x_2 \leq \cdots \leq x_n$. Let $G_n(\theta)$ be a discrete estimator of the underlying mixing distribution with a set of support points at $\{\theta_{nj}; j = 1, \ldots, k_n\}$. Each θ_{nj} provides a component of the final clustering with weight $w_{nj} \geq 0$, where $\sum_{j=1}^{k_n} w_{nj} = 1$. Given the support points, obtaining $G_n(\theta)$ is equivalent to computing the weight vector $\mathbf{w}_n = (w_{n1}, w_{n2}, \ldots, w_{nk_n})^t$. Denote by $F_{G_n}(x)$ the estimated mixture CDF with respect to $G_n(\theta)$.

Two minimum distance estimators were proposed in the late 1960s. Choi and Bulgren (1968) used

$$\frac{1}{n}\sum_{i=1}^{n}[F_{G_n}(x_i) - i/n]^2 \tag{5.5}$$

as the distance measure. Minimizing this quantity with respect to G_n yields a strongly consistent estimator. This estimator can be slightly improved upon by using the Cramér-von Mises statistic

$$\frac{1}{n}\sum_{i=1}^{n}[F_{G_n}(x_i) - (i - 1/2)/n]^2 + 1/(12n^2), \tag{5.6}$$

which essentially replaces i/n in (5.5) with $(i - \frac{1}{2})/n$ without affecting the asymptotic result. As might be expected, this reduces the bias for small-sample cases, as was demonstrated empirically by Macdonald (1971) in a note on Choi and Bulgren's paper.

At about the same time, Deely and Kruse (1968) used the sup-norm associated with the Kolmogorov-Smirnov test. The minimization is over

$$\sup_{1 \leq i \leq n} \{|F_{G_n}(x_i) - (i - 1)/n|, |F_{G_n}(x_i) - i/n|\}, \tag{5.7}$$

and this leads to a linear programming problem. Deely and Kruse also established the strong consistency of their estimator G_n.

Ten years later, the above approach was extended by Blum and Susarla (1977) to approximate density functions by using any function sequence $\{f_n\}$ that satisfies

$\sup|f_n - f_G| \to 0$ a.s. as $n \to \infty$, where f_G is the underlying pdf of the mixture. Each f_n can be obtained, for example, by a kernel-based density estimator. Blum and Susarla approximated the function f_n by the overall mixture pdf f_{G_n}, and established the strong consistency of the estimator G_n under weak conditions.

For reason of simplicity and generality, we will denote the approximation between two mathematical entities of the same type by $\approx$, which implies the minimization with respect to an estimator of a distance measure between the entities on either side. The types of entity involved in this thesis include vector, function and measure, and we use the same symbol $\approx$ for each.

In the work reviewed above, two kinds of estimator are used: CDF-based (Choi and Bulgren, Macdonald, and Deely and Kruse) and pdf-based (Blum and Susarla). CDF-based estimators involve approximating an empirical distribution with an estimated one F_{G_n}. We write this as

$$F_{G_n} \approx F_n, \tag{5.8}$$

where F_n is the Kolmogorov empirical CDF,

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \delta_{[x_i,\infty)}(x), \tag{5.9}$$

or indeed any empirical CDF that converges to it. Pdf-based estimators involve the approximation between probability density functions:

$$f_{G_n} \approx f_n, \tag{5.10}$$

where f_{G_n} is the estimated mixture pdf and f_n is the empirical pdf described above.

The entities in (5.8) and (5.10) are functions. When the approximation is computed, however, it is computed between vectors that represent the functions. These vectors contain the function values at a particular set of points, which we call "fitting points." In the work reviewed above, the fitting points are chosen to be the data

points themselves.

5.3 Generalizing the CDF-based approach

In this section we generalize the CDF-based approach using the form (5.8). A well-defined CDF-based estimator needs to specify (a) the set of support points, (b) the set of fitting points, (c) the empirical function, and (d) the distance measure. The following generalization, due to the considerations given in Subsection 5.3.3, will cover all of the above four aspects. In Subsection 5.3.1, conditions for establishing the estimator's strong consistency are given. The proof of strong consistency is given in Subsection 5.3.2.

5.3.1 Conditions

Conditions for defining the estimators and ensuring strong consistency are given in this section. They are divided into two groups. The first includes the continuity condition (Condition 5.1), the identifiability condition (Condition 5.2), and Condition 5.3. Their satisfaction needs to be checked in practice. The second group, which can always be satisfied, is used to precisely define the estimator. They are Condition 5.4 for the selection of (potential) support points, Conditions 5.5–5.6 for distance measure, Conditions 5.7 for empirical function, and Conditions 5.8–5.9 for fitting points.

Condition 5.1 *$F(x;\theta)$ is a continuous function over $\mathcal{X} \times \Theta$.*

Condition 5.2 *Define $\mathcal{G}$ to be the class of CDFs on Θ. If $F_G = F_H$ for $G, H \in \mathcal{G}$, then $G = H$.*

Condition 5.2 is known as the identifiability condition. The identifiability problem of mixture distributions has attracted much research attention since the seminal pa-

pers by Teicher (1960, 1961, 1963). References and summaries can be found in Prakasa Rao (1992).

Condition 5.3 *Either* Θ *is a compact subset of* $\mathbb{R}$, *or* $\lim_{\theta \to \pm\infty, \theta \in \Theta} F(x;\theta)$ *exist for each* $x \in \mathcal{X}$ *and are not distribution functions on* $\mathcal{X}$.

Conditions 5.1–5.3 were initially considered by Robbins (1964), and are essential to establish the strong consistency of the estimator of the mixing distribution given the uniform convergence of the mixture estimator.

Condition 5.4 *Define* $\mathcal{G}_n$ *to be the class of discrete distributions on* Θ *with support at* $\theta_{n1}, \ldots, \theta_{nj_n}$, *where the* $\{\theta_{nj}\}$ *are chosen so that for any* $G \in \mathcal{G}$ *there is a sequence* $\{G_n^w\}$ *with* $G_n^w \in \mathcal{G}_n$ *that converges weakly to* G *a.s.*

Condition 5.4 concerns the selection of support points. In fact, a weakly convergent sequence $\{G_n^w\}$, instead of the almost surely weakly convergent sequence, is always obtainable. This is because if Θ is compact, a weakly convergent sequence $\{G_n^w\}$ for any G can be obtained by, say, equally spacing θ_{nj} throughout Θ; if Θ is not compact, some kind of mapping of the equally spaced points in a compact space onto Θ can provide such a sequence.

Despite this fact, we adopt a weaker condition here that only requires a sequence which, *almost surely*, converges weakly to any given G. We will discover that this relaxation does not change the conclusion about the strong consistency of the resulting estimator. The advantage of using a weaker condition is that the set of support points $\{\theta_{nj}\}$ can be adapted to the given sample, often resulting in a data-oriented and hence probably smaller set of support points, which implies less computation and possibly higher accuracy. For example, each data point could provide a support point, suggested by the unicomponent maximum likelihood estimators. This set always contains a sequence $\{G_n^w\}$ which converges weakly to any given G a.s.

Let $d(\mathbf{u}, \mathbf{v})$ be a distance measure between two k-vectors $\mathbf{u}$ and $\mathbf{v}$. This distance is not necessarily a metric; that is, the triangle inequality may not be satisfied. Denote

the L_p-norm by $||\cdot||_p$. Conditions 5.5 and 5.6 provide a wide class of distance measures.

Condition 5.5 *Either*

(i) $d(\mathbf{u},\mathbf{v}) = ||\mathbf{u}-\mathbf{v}||_\infty$

or

(ii) if $\lim_{k\to\infty}\frac{1}{k}||\mathbf{u}-\mathbf{v}||_1 \neq 0$, then $\lim_{k\to\infty} d(\mathbf{u},\mathbf{v}) \neq 0$.

Condition 5.6 $d(\mathbf{u},\mathbf{v}) \leq h(||\mathbf{u}-\mathbf{v}||_\infty)$, *where* $h(z)$ *is a univariate, non-decreasing function and* $\lim_{z\to 0} h(z) = h(0) = 0$.

Examples of distances between two k-vectors satisfying conditions 5.5 and 5.6 are $||\mathbf{u}-\mathbf{v}||_p/\sqrt[p]{k}$ $(0<p<\infty)$, $||\mathbf{u}-\mathbf{v}||_p^p/k$ $(0<p<\infty)$, and $||\mathbf{u}-\mathbf{v}||_\infty$.

Let F_n be any empirical function obtained from observations (not necessarily a CDF) satisfying

Condition 5.7 $\lim_{n\to\infty}||F_G - F_n||_\infty = 0$ *a.s.*

According to the Glivenko-Cantelli theorem, the Kolmogorov empirical CDF satisfies this condition. Here $\delta_I(x)$ is the indicator function: $\delta_I(x) = 1$ if $x \in I$, and 0 if otherwise. Other empirical functions which (a.s.) uniformally converge to the Kolmogorov empirical CDF as $n \to \infty$ satisfy this condition as well. Empirical functions other than the Kolmogorov empirical CDF result in the same strong consistency conclusion, but can provide flexibility for better estimation in the finite data situation. For example, an empirical CDF obtained from the Cramer-von Mises statistic can be easily constructed, which is different from the Kolmogorov empirical CDF but uniformally convergent to it.

Denote the chosen set of fitting points on $\mathcal{X}$ by a vector $\mathbf{a}_n = (a_{n1}, a_{n2}, \ldots, a_{ni_n})^t$. The number of points i_n needs to satisfy

Condition 5.8 $\lim_{n\to\infty} i_n = \infty$.

Let $\mathcal{B}$ be the set of all Borel subsets on $\mathcal{X}$. Denote by $i(I)$ the number of fitting points that are located within the subset $I \in \mathcal{B}$, and by $P_G(I)$ the probability measure determined by F_G over I.

Condition 5.9 *There exists a positive constant c such that, for any* $I \in \mathcal{B}$,

$$\lim_{n\to\infty} \frac{i(I)}{i_n} \geq cP_G(I) \ \ a.s. \tag{5.11}$$

Conditions 5.8–5.9 are satisfied when the set of fitting points is an exact copy of the set of data points, as in the minimum distance methods reviewed in Subsection 5.2.2. Using a larger set of fitting points when the number of data points is small makes the resulted estimator more accurate within tolerable computational cost. Using fewer fitting points when there are many data points over a small interval decreases the computational burden without sacrificing much accuracy. In the finite sample situation, Condition 5.9 gives great freedom in the choice of fitting points. For example, more points can be placed around specific areas for better estimation. All these can be done without sacrificing strong consistency.

5.3.2 Estimators

For any discrete distribution $G_n(\theta) = \sum_{j=1}^{j_n} w_{nj}\delta_{[\theta_{nj},\infty)}(\theta)$ on Θ, the corresponding mixture distribution is

$$F_{G_n}(x) = \int F(x;\theta)\,\mathrm{d}G_n(\theta) = \sum_{j=1}^{j_n} w_{nj}F(x;\theta_{nj}). \tag{5.12}$$

Let $\mathbf{a}_n$ be a vector of elements from $\mathcal{X}$; for example, the set of fitting points. Denote the vectors of the function values $F_{G_n}(x)$ and $F_n(x)$ corresponding to the elements of $\mathbf{a}_n$ by $F_{G_n}(\mathbf{a}_n)$ and $F_n(\mathbf{a}_n)$. The distance between two vectors $F_{G_n}(\mathbf{a}_n)$ and $F_n(\mathbf{a}_n)$ is written

$$S_n(G_n) = d(F_{G_n}(\mathbf{a}_n), F_n(\mathbf{a}_n)). \tag{5.13}$$

The estimator G_n is defined as the one minimizing $S_n(G_n)$. We have

Theorem 5.1 *Under Conditions 5.1–5.9,* $\lim_{n\to\infty} G_n \to_w G$ *a.s.*

Proof. The proof of the theorem consists of three steps.

(1). $\lim_{n\to\infty} S_n(G_n) = 0$ a.s.

Clearly, for any (discrete or continuous) distribution H on Θ, the distribution F_H on $\mathcal{X}$ is pointwise continuous under Condition 5.1. $G_n^w \to_w G$ a.s. (Condition 5.3) implies $F_{G_n^w} \to F_G$ pointwise a.s. according to the Helly-Bray theorem. This further implies uniform convergence (Pólya, 1920), that is, $\lim_{n\to\infty} ||F_{G_n^w} - F_G||_\infty = 0$ a.s. Because $\lim_{n\to\infty} ||F_G - F_n||_\infty = 0$ a.s. by Condition 5.7, and

$$||F_{G_n^w} - F_n||_\infty \leq ||F_{G_n^w} - F_G||_\infty + ||F_G - F_n||_\infty, \tag{5.14}$$

then $\lim_{n\to\infty} ||F_{G_n^w} - F_n||_\infty = 0$ a.s. Since $S_n(G_n^w) \leq h(||F_{G_n^w}(\mathbf{a}_n) - F_n(\mathbf{a}_n)||_\infty) \leq h(||F_{G_n^w} - F_n||_\infty)$, this means that $\lim_{n\to\infty} S_n(G_n^w) = 0$ a.s. by Condition 5.6. Because the estimator G_n is obtained by a minimization procedure over $S_n(\cdot)$, for every $n \geq 1$

$$S_n(G_n) \leq S_n(G_n^w). \tag{5.15}$$

Therefore, $\lim_{n\to\infty} S_n(G_n) = 0$ a.s.

(2). $\lim_{n\to\infty} ||F_{G_n} - F_G||_\infty = 0$ a.s.

Let H be an arbitrary element in the set $\{G_n : n \to \infty$ and $\lim_{n\to\infty} ||F_{G_n} - F_G||_\infty \neq 0\}$, i.e. $||F_H - F_G||_\infty \neq 0$. Then $\exists x_0$, $F_H(x_0) \neq F_G(x_0)$.

Because F_H and F_G are both continuous and non-decreasing, and $F_H(-\infty) = F_G(-\infty) = 0$ and $F_H(\infty) = F_G(\infty) = 1$, there must exist an interval I_0 in the range of F_G near $F_G(x_0)$ such that the length ΔF of I_0 is positive and $\inf |F_H(x) - F_G(x)| \geq \epsilon > 0$ for all $x \in F_G^{-1}(I_0)$, where $F_G^{-1}(I_0)$ denotes the inverse mapping of F_G over I_0. (Note that $F_G^{-1}(I_0)$ may not be unique, but the conclusion remains the

same.) Because $||F_G - F_n||_\infty \to 0$ a.s., when n is sufficiently large, $||F_G - F_n||_\infty$ can be made arbitrarily small a.s. Therefore, within $F_G^{-1}(I_0)$, $\inf |F_H(x) - F_n(x)| \geq \epsilon$ a.s. as $n \to \infty$. By Condition 5.9, $\lim_{n\to\infty} i(F_G^{-1}(I_0))/i_n \geq cP_G(F_G^{-1}(I_0)) \geq c\Delta F$ a.s. By Conditions 5.5, either $S_n(G_H) \geq \epsilon$ a.s. if $d(\mathbf{u}, \mathbf{v}) = ||\mathbf{u} - \mathbf{v}||_\infty$, or $\lim_{n\to\infty} \frac{1}{i_n} \sum_{i=1}^{i_n} |F_H(\mathbf{a}_n) - F_n(\mathbf{a}_n)| \geq c\Delta F$ a.s. if $d(\mathbf{u}, \mathbf{v})$ is defined otherwise. In either situation, $S_n(H) \neq 0$ a.s.

From the last paragraph, the arbitrariness of H and the continuity of F_H in a metric space with respect to H, we have

$$\Pr(\lim_{n\to\infty} S_n(G_n) \neq 0 | \lim_{n\to\infty} ||F_{G_n} - F_G||_\infty \neq 0) = 1. \tag{5.16}$$

By step (1), immediately, $||F_{G_n} - F_G||_\infty \to 0$ a.s.

(3). $G_n \to_w G$ a.s.

Robbins (1964) proved that $||F_{G_n} - F_G||_\infty \to 0$ a.s. implies that $G_n \to_w G$ a.s. under Conditions 5.1–5.3. □

Clearly, the generalized estimator obtained by minimizing (5.13) covers Choi and Bulgren (1968) and Cramér-von Mises statistic (Macdonald, 1971). It can be further generalized as follows to cover Deely and Kruse (1968), who use two empirical CDFs.

Let $\{F_{ni}, i = 1, \ldots, m\}$ be a set of empirical CDFs, where each F_{ni} satisfies Condition 5.7. Denote

$$S_{ni}(G_n) = d(F_{G_n}(\mathbf{a}_n), F_{ni}(\mathbf{a}_n)). \tag{5.17}$$

Theorem 5.2 *Let G_n be the estimator by minimizing* $\max_{1\leq i\leq m} S_{ni}(G_n)$. *Then under Conditions 5.1–5.9, $G_n \to_w G$ a.s.*

The proof is similar to that of Theorem 5.1 and thus omitted.

The proof of Theorem 5.1 is inspired by the work of Choi and Bulgren (1968) and Deely and Kruse (1968). The main difference in our proof is the second step. We claim this proof is more general: it covers their results, while their proof does not cover ours. A similar proof will be used in the next section for measure-based estimators.

5.3.3 Remarks

In the above, we have generalized the estimation of a mixing distribution based on approximating mixture CDFs. As will be shown in Section 5.5, the CDF-based approach does not solve the minority cluster problem. However, this generalization is included because of the following considerations:

1. It is of theoretic interest to show that there actually exists a large class of minimum distance estimators, not just a few individual ones, that can provide strongly consistent estimates of an arbitrary mixing distribution.

2. We would like to formalize the definition of the estimators of a mixing distribution, including the selection of support points and fitting points. For example, Choi and Bulgren (1968) did not discuss how the set of support points $\{\theta_{nj}\}$ should be chosen for their estimator. Clearly, the consistency of the estimator relies on how this set is chosen.

3. A more general class of estimators may provide flexibility and adaptability in practice without losing strong consistency. Condition 5.4 is such an example.

4. The CDF-based approach is a special and simpler case of the measure-based approach. It provides a starting point for the more complicated situation.

5.4 The measure-based approach

This section extends the CDF-based approach to the measure-based one. The idea is to approximate the empirical measure with the estimated measure over intervals, which we call *fitting intervals*. The two measures are represented by two vectors that contain values that the measures take over the fitting intervals. Then distance in the vector space is minimized with respect to the candidate mixing distributions.

Two approaches are considered further: the probability-measure-based (or PM-based) approach and the nonnegative-measure-based (NNM-based) approach. For the former, the empirical measure P_n is approximated by the estimated probability measure P_{G_n}, which can be written

$$P_{G_n} \approx P_n, \tag{5.18}$$

where G_n is a discrete CDF on Θ. For the latter, we abandon the normalization constraint for G_n so that the estimate is only a nonnegative measure, thus not necessarily a probability measure. We write this as

$$P_{G'_n} \approx P_n, \tag{5.19}$$

where G'_n is a discrete function with nonnegative mass at support points. G'_n can be normalized afterwards to become a distribution function, if necessary.

As with the CDF-based estimators, we need to specify (a) the set of support points, (b) the set of fitting intervals, (c) the empirical measure, and (d) the distance measure. Only (b) and (c) differ from CDF-based estimation. They are given in Subsection 5.4.1 for the PM-based approach, and modified in Subsection 5.4.3 to fit the NNM-based approach. Strong consistency results for both estimators are established in Subsections 5.4.2 and 5.4.3.

5.4.1 Conditions

For the PM-based approach, we need only specify conditions for fitting intervals and the empirical measure—the other conditions remain identical to those in Subsection 5.3.1. We first discuss how to choose the empirical measure, then give conditions (Conditions 5.11–5.14) for selecting fitting intervals.

Denote the value of a measure P over an interval I by $P(I)$. Let P_n be an empirical measure (i.e., a measure obtained from observations, but not necessarily a probability measure). Define the corresponding nondecreasing empirical function F_n (hence not necessarily a CDF) as

$$F_n(x) = P_n((-\infty, x]). \tag{5.20}$$

Clearly P_n and F_n are uniquely determined by each other on Borel sets. We have the following condition for P_n.

Condition 5.10 *The F_n corresponding to P_n satisfies Condition 5.7.*

If F_n is the Kolmogorov empirical CDF as defined in (5.9), the corresponding P_n over a subset $I \in \mathcal{B}$ is

$$P_n(I) = \frac{1}{n} \sum_{i=1}^{n} \delta_I(x_i). \tag{5.21}$$

An immediate result following Condition 5.10 is $\lim_{n\to\infty} ||P_G(I) - P_n(I)||_\infty = 0$ a.s. for all $I \in \mathcal{B}$.

We determine the set of fitting intervals by the set of the right endpoints and a function $l(x)$ which, given a right endpoint, generates a left endpoint for the fitting interval. Given the set of right endpoints $\{a_{n1}, \ldots, a_{ni_n}\}$, the set of fitting intervals is determined as $\mathcal{I}_n = \{I_{n1}, \ldots, I_{ni_n}\} = \{(l(a_{n1}), a_{n1}), \ldots, (l(a_{ni_n}), a_{ni_n})\}$. Setting each interval to be open, closed or semi-open will not change the asymptotic conclusion, because of the continuity condition (Condition 5.1).

In order to establish strong consistency, we require $l(x)$ to satisfy the following.

Condition 5.11 *For some constant $\delta_x > 0$, $\forall x \in \mathcal{X}$, $x - l(x) \geq \delta_x$, except for those x's such that $l(x) \notin \mathcal{X}$; for the exceptions, let $l(x) = \inf \mathcal{X}$.*

An example of such a function is $l(x) = x - a$ for $x - a \in \mathcal{X}$, and $l(x) = \inf \mathcal{X}$ if otherwise. Further, define

$$l^k(x) = \begin{cases} x, & \text{if } k = 0; \\ l(l^{k-1}(x)), & \text{if } k > 0. \end{cases} \tag{5.22}$$

The following conditions govern the selection of fitting intervals; in particular, they determine the number and locations of right endpoints.

Condition 5.12 $\lim_{n\to\infty} i_n = \infty$.

We use an extra CDF $F_a(x)$ to determine the distribution of the right endpoints of fitting intervals. If possible, we always replace it with $F_G(x)$ so that fitting intervals can be determined dynamically based on the sample; otherwise, some additional techniques are needed. We will discuss this issue in Subsection 5.4.4.

Condition 5.13 *$F_a(x)$ is a strictly increasing CDF throughout $\mathcal{X}$.*

Denote by $P_a(I)$ its corresponding probability measure over a subset $I \in \mathcal{B}$. Clearly $P_a(I) \neq 0$ unless the Lebesgue measure of I is zero for every $I \in \mathcal{B}$. Denote by $i(I)$ the number of points in the intersection between the set of the right endpoints $\{a_{ni}; i = 1, \ldots, i_n\}$ and a subset I on $\mathcal{X}$.

Condition 5.14 *There exists a constant $c > 0$ such that, for all $I \in \mathcal{B}$,*

$$\lim_{n\to\infty} \frac{i(I)}{i_n} \geq cP_a(I) \ \ a.s. \tag{5.23}$$

If $F(x;\theta)$ is strictly increasing throughout $\mathcal{X}$ for every $\theta \in \Theta$, $F_G(x)$ is strictly increasing throughout $\mathcal{X}$ as well. In this situation we can use $F_G(x)$ to substitute for $F_a(x)$. Using $F_G(x)$ instead of the vague $F_a(x)$ to guide the selection of fitting intervals simplifies the selection and makes it data-oriented. For example, set $i_n = n$, and $a_{ni} = x_i$, $i = 1, \ldots, n$. Then we have $\lim_{n\to\infty} i(I)/n = P_G(I)$ a.s., thus satisfying Condition 5.14, where c is a value satisfying $0 < c \leq 1$.

5.4.2 Approximation with probability measures

$P(I)$ denotes the value of a probability measure P over $I \in \mathcal{B}$. Further, let $P(\mathcal{I}_n)$ denote the vector of the values of the probability measure over the set of fitting intervals. Our task is to find an estimator $G_n \in \mathcal{G}_n$ (defined in Condition 5.4) of the mixing distribution CDF which minimizes

$$S_n(G_n) = d(P_{G_n}(\mathcal{I}_n), P_n(\mathcal{I}_n)). \tag{5.24}$$

The proof of the following theorem follows that of Theorem 5.1. The difference is that here we deal with probability measure and the set of fitting intervals of (possibly) finite length as defined by Conditions 5.11–5.14, while Theorem 5.1 tackles CDFs, which use intervals all starting from $-\infty$.

Theorem 5.3 *Under Conditions 5.1–5.6 and 5.10–5.14,* $\lim_{n\to\infty} G_n \to_w G$ *a.s.*

The proof of this theorem needs two lemmas.

Lemma 2 *Under Condition 5.1 and 5.11, if* $||F_H - F_G||_\infty \neq 0$ *for* $H, G \in \mathcal{G}$, *there exists a point* x_0 *such that* $P_H(l(x_0), x_0) \neq P_G(l(x_0), x_0)$.

Proof of Lemma 2. Assume $P_H(l(x), x) = P_G(l(x), x)$ for every $x \in \mathcal{X}$. $F_H(x) = \sum_{k=1}^{\infty} P_H(l^k(x), l^{k-1}(x))$ and $F_G(x) = \sum_{k=1}^{\infty} P_G(l^k(x), l^{k-1}(x))$ for every $x \in \mathcal{X}$ (note that since F_H and F_G are continuous, the probability measure over a countable

set of singletons is zero). Thus $F_H(x) = F_G(x)$ for all $x \in \mathcal{X}$. This contradicts $||F_H - F_G||_\infty \neq 0$, thus completing the proof. □

Lemma 3 *If there is a point $x_0 \in \mathcal{X}$ such that $P_H(l(x_0), x_0) \neq P_G(l(x_0), x_0)$ for $H, G \in \mathcal{G}$, then given conditions 5.1, 5.5, 5.10-5.14, $\lim_{n\to\infty} S_n(H) \neq 0$ a.s.*

Proof of Lemma 3. Since $F_H(x)$ and $F_G(x)$ are both continuous, $P_H(l(x), x)$ and $P_G(l(x), x)$ are continuous functions of x. There must be an interval I of nonzero length near x_0 such that for every point $x \in I$, $P_H(l(x), x) - P_G(l(x), x) \neq 0$. According to the requirement for $P_a(x)$, we can assume $P_a(I) = \epsilon_P > 0$. By Conditions 5.12 and 5.14, no fewer than $i_n c \epsilon_P$ points are chosen as the right endpoints of fitting intervals from I a.s. as $n \to \infty$. Since

$$||P_H(\mathcal{I}_n) - P_n(\mathcal{I}_n)||_\infty \geq ||P_H(\mathcal{I}_n) - P_G(\mathcal{I}_n)||_\infty - ||P_G(\mathcal{I}_n) - P_n(\mathcal{I}_n)||_\infty \quad (5.25)$$

and $\lim_{n\to\infty} ||P_G(\mathcal{I}_n) - P_n(\mathcal{I}_n)||_\infty = 0$, we have $\lim_{n\to\infty} ||P_H(\mathcal{I}_n) - P_n(\mathcal{I}_n)||_\infty \geq \lim_{n\to\infty} ||P_H(\mathcal{I}_n) - P_G(\mathcal{I}_n)||_\infty > 0$ a.s. Also, $\lim_{n\to\infty} \frac{1}{i_n} ||P_H(\mathcal{I}_n) - P_n(\mathcal{I}_n)||_1 > 0$ a.s. as $n \to \infty$. This implies $\lim_{n\to\infty} S_n(H) \neq 0$ a.s. by Condition 5.5. □

Proof of Theorem 5.3. The proof of the theorem consists of three steps.

(1). $\lim_{n\to\infty} S_n(G_n) = 0$ a.s.

Following the first step in the proof of Theorem 5.1, we obtain $\lim_{n\to\infty} ||F_{G_n^w} - F_n||_\infty = 0$ a.s. Immediately, $\lim_{n\to\infty} ||P_{G_n^w} - P_n||_\infty = 0$ a.s. for all Borel sets on $\mathcal{X}$. Since $S_n(G_n^w) \leq h(||P_{G_n^w}(\mathcal{I}_n) - P_n(\mathcal{I}_n)||_\infty) \leq h(||P_{G_n^w} - P_n||_\infty)$, this means that $\lim_{n\to\infty} S_n(G_n^w) = 0$ a.s. by Condition 5.6. Because the estimator G_n is obtained by a minimization procedure of $S_n(\cdot)$, for every $n \geq 1$

$$S_n(G_n) \leq S_n(G_n^w). \quad (5.26)$$

Hence $\lim_{n\to\infty} S_n(G_n) = 0$ a.s.

(2). $\lim_{n\to\infty} ||F_{G_n} - F_G||_\infty = 0$ a.s.

Let H be an arbitrary element in the set $\{G_n; n \to \infty : \lim_{n\to\infty} ||F_{G_n} - F_G||_\infty \neq 0\}$, i.e. $||F_H - F_G||_\infty \neq 0$. According to Lemmas 2 and 3, $\lim_{n\to\infty} S_n(H) \neq 0$ a.s. Therefore, due to the arbitrariness of H, the continuity of F_H, and step (1), $||F_{G_n} - F_G||_\infty \to 0$ a.s., as $n \to \infty$.

(3). $G_n \to_w G$ a.s.

The proof is the same as the third step in the proof of Theorem 5.1. □

In Subsection 5.4.1, each fitting interval I_i is uniquely determined by the right endpoint a_{ni}—the left endpoint is given by a function $l(a_{ni})$. If we instead determine each fitting interval in the reverse order—i.e., given the left endpoint, compute the right endpoint—the following theorem is obtained. The function $l(x)$ for computing the right endpoint can be defined analogously to Subsection 5.4.1. The proof is similar and omitted.

Theorem 5.4 *Let G_n be the estimator described in the last paragraph. Then under the same conditions as in Theorem 5.3,* $\lim_{n\to\infty} G_n \to_w G$ *a.s.*

Further, let the ratio between the number of intervals in a subset of the set of fitting intervals and i_n be greater than a positive constant as $n \to \infty$, and the intervals in this subset are either all in the form $(l(a_{ni}), a_{ni})$ or all in the form $(a_{ni}, l(a_{ni}))$ and satisfy Conditions 5.13 and 5.14. We have the following theorem, whose proof is similar and omitted too.

Theorem 5.5 *Let G_n be the estimator described in the last paragraph. Then under the same conditions as in Theorem 5.3,* $\lim_{n\to\infty} G_n \to_w G$ *a.s.*

Theorem 5.5 provides flexibility for practical applications with finite samples. This issue will be discussed in Subsection 5.4.4.

In fact, $F_a(x)$ can always be replaced with $F_G(x)$ in Condition 5.9 for establishing the theorems in this subsection, even if $F(x;\theta)$ is not strictly increasing throughout

$\mathcal{X}$. This is because the normalization constraint of the estimator G_n ensures that F_H in Lemma 3 is a distribution function on $\mathcal{X}$, thus $F_H(-\infty) = F_G(-\infty) = 0$ and $F_H(\infty) = F_G(\infty) = 1$. This implies that if at any point x_0, $F_H(x_0) \neq F_G(x_0)$, there must exist an interval I near x_0, satisfying $P_G(I) \neq 0$, such that $F_H(x) - F_G(x) \neq 0, \forall x \in I$, which can be used to establish Lemma 3.

We will see that the estimator investigated in the next subsection could not always replace $F_a(x)$ with $F_G(x)$.

Barbe (1998) investigated the properties of mixing distribution estimators, obtained by approximating the empirical probability measure with the estimated probability measure. He established theoretical results based on certain properties of the estimator. However, he did not develop an operational estimation scheme. The estimators discussed in this subsection also form a wider class than the one he used. For example, P_n may not be a probability measure.

5.4.3 Approximation with nonnegative measures

In the last subsection, we discussed the approximation of empirical measures with estimated probability measures. Now we consider throwing away the normalization constraint. As in (5.19), G'_n is a discrete function with nonnegative mass at support points. Let $G'_n(\theta) = \sum_{j=1}^{j_n} w'_{nj}\delta_{(-\infty,\theta_{nj}]}(\theta)$ with every $w'_{nj} \geq 0$. The mixture function becomes

$$F_{G'_n}(x) = \int F(x;\theta)\, \mathrm{d}G'_n(\theta) = \sum_{j=1}^{j_n} w'_{nj} F(x;\theta_{nj}). \tag{5.27}$$

Note that the w_{nj}'s are not assumed to sum to one, hence neither G'_n nor $F_{G'_n}$ are CDFs. Accordingly, the corresponding nonnegative measure is not a probability measure. Denote this measure by $P_{G'_n}(I)$ for any $I \in \mathcal{B}$. The estimator G'_n is the one that minimizes

$$S_n(G'_n) = d(P_{G'_n}(\mathcal{I}_n), P_n(\mathcal{I}_n)). \tag{5.28}$$

Of course, G_n' can be normalized afterwards.

We will establish strong consistency of G_n'. Based on the work in the last subsection, the result is almost straightforward. We need to use the general meaning of weak convergence for functions, instead of CDFs. In Condition 5.2, $\mathcal{G}$ is changed to $\mathcal{G}'$, the class of finite, nonnegative, nondecreasing functions defined on $\mathcal{X}$ such that $G'(-\infty) = 0$ for every $G' \in \mathcal{G}'$. Also, in Condition 5.4, $\mathcal{G}_n$ is replaced with $\mathcal{G}_n'$. The identifiability is now about general functions. In fact, we have

Lemma 4 *The identifiability of mixture F_G where $G \in \mathcal{G}$ implies the identifiability of mixture $F_{G'}$ where $G' \in \mathcal{G}'$.*

Proof. Assume $F_{G'} = F_{H'}$, where $G', H' \in \mathcal{G}'$. Then $F_{G'}(\infty) = F_{H'}(\infty)$, i.e., $\int_\Theta G'\, d\theta = \int_\Theta H'\, d\theta$. Let $G = G'/\int_\Theta G'\, d\theta$ and $H = H'/\int_\Theta H'\, d\theta$. Hence $F_G = F_H$. Since both G and H are CDFs on Θ, $G = H$, or equivalently, $G' = H'$, thus completing the proof. □

Also, we need a lemma which is similar to Theorem 2 in Robbins (1964).

Lemma 5 *Under Conditions 5.1-5.3 with changes described in the paragraph before Lemma 4, if $\lim_{n\to\infty} ||F_{G_n'} - F_G||_\infty = 0$ a.s., then $G_n' \to_w G$ a.s.*

The proof is intactly the same as that of Theorem 2 in Robbins (1964) and thus omitted.

Lemma 6 *Let G_n be a CDF obtained by normalizing G_n' and $\lim_{n\to\infty} G_n' \to_w G$ a.s. where $G \in \mathcal{G}$. Then $\lim_{n\to\infty} G_n \to_w G$ a.s.*

Let G_n' be the solution obtained by minimizing $S_n(G_n')$ in (5.28). We have

Theorem 5.6 *Under similar conditions with changes mentioned above, the conclusions of Theorems 5.3-5.5 remain true if G_n is replaced with G_n' (or the estimator obtained by normalizing G_n').*

The proof of Theorem 5.6 follows exactly that of Theorem 5.3. For step 3, Lemma 5 should be used instead.

It was shown in the last subsection that $F_G(x)$ can always be replaced with $F_a(x)$ in Condition 5.9 to establish strong consistency for G_n obtained from (5.18). This is, however, not so for G'_n obtained from (5.19). For example, if $F(x;\theta)$ is bounded on $\mathcal{X}$ (e.g., uniform distribution), then $F_G(x)$ may have intervals with zero P_G-measure. Mixing distributions, with or without components bounded within zero P_G-measure intervals, have the same $S_n(\cdot)$ value as $n \to \infty$.

5.4.4 Considerations for finite samples

From the previous subsections, it becomes clear that there is a large class of strongly consistent estimators. The behavior of these estimators, however, differs radically in the finite sample situation. In the discussion of a few finite sample issues that follows, we will see that, without careful design, some estimators can perform badly. We take an algorithmic viewpoint and ensure that the estimation is workable, fast and accurate in practice. The conditions carefully defined in Subsections 5.3.1 and 5.4.1 provide such flexibility and adaptability.

Of all minimum distance estimators proposed so far, only the nonnegative-measure-based one is able to solve the minority cluster problem (see Section 5.5), in the sense that it can always reliably detect small clusters when their locations are far away from dominant data points. The experiments conducted in Sections 5.5 and 5.6 also seem to suggest that these estimators are generally more accurate and stable than other minimum distance estimators. We focus on the NNM-based estimator.

The conditions in Subsection 5.3.1 show that, without losing strong consistency, the set of support points and the set of fitting intervals can be determined without taking the data points into account. Nevertheless, unless sufficient prior knowledge is provided about the support points and the distribution of data points, such an independent procedure is unlikely to provide satisfactory results. An algorithm

that determines the two sets before seeing the data would not work for many general cases, because the underlying mixing distribution changes from application to application. The sample needs to be considered when determining the two sets.

The support points could be suggested by the sample and the component distribution. As mentioned earlier, each data point could generate a support point using the unicomponent maximum likelihood estimator. For example, for the mixture of normal distributions where the mean is the mixing parameter, data points can be directly taken as support points.

Each data point can be taken as an endpoint of a fitting interval, while the function $l(x)$ can be decided based on the component distribution. In the above example of a mixture of normal distributions, we can set, say, $l(x) = x - 3\sigma$. Note that fitting intervals should not be chosen so large that data points may exert an influence on the approximation at a remote place, or so small that the empirical (probability) measures are not accurate enough.

The number of support points and the number of fitting intervals can be adjusted according to Conditions 5.4, 5.12–5.14. For example, when there are few data points, more support points and fitting intervals can be used to get a more accurate estimate within a tolerable computational cost; when there are a large number of data points in a small region, fewer support points and fitting intervals can be used, which substantially decreases computational cost while keeping the estimate accurate.

Basing the estimate on solving (5.19) has computational advantages in some cases. The solution of an equation like the following,

$$\begin{pmatrix} \mathbf{A}'_{G'_n} & \underline{0} \\ \underline{0} & \mathbf{A}''_{G'_n} \end{pmatrix} \begin{pmatrix} \mathbf{w}'_n \\ \mathbf{w}''_n \end{pmatrix} \approx \begin{pmatrix} \mathbf{p}'_n \\ \mathbf{p}''_n \end{pmatrix}, \tag{5.29}$$

subject to $\mathbf{w}'_n \geq \mathbf{0}$ and $\mathbf{w}''_n \geq \mathbf{0}$, is the same as combining the solutions of two subequations $\mathbf{A}'_{G'_n}\mathbf{w}'_n \approx \mathbf{p}'_n$ subject to $\mathbf{w}'_n \geq \mathbf{0}$, and $\mathbf{A}''_{G'_n}\mathbf{w}''_n \approx \mathbf{p}''_n$ subject to $\mathbf{w}''_n \geq \mathbf{0}$. Each sub-equation can be further partitioned. This implies that clusters separating from one another can be estimated separately. Further, normalizing the

separated clusters' weight vectors before recombining them can produce a better estimate. The computational burden decreases significantly as the extent of separation increases. The best case is that each data point is isolated from every other one, suggesting that each point represents a cluster comprising a single point. In this case, the computational complexity is only linear.

We wish to preserve strong consistency, and determine the support points and fitting intervals from the data. Under our conditions, however, an extra CDF $F_a(x)$, which should be strictly increasing, is needed to determine fitting intervals. It can be replaced with $F_G(x)$ when $F(x;\theta)$ is strictly increasing, making it data-oriented. But when $F(x;\theta)$ is not strictly increasing, $F_G(x)$ cannot substitute for $F_a(x)$. Since the selection of fitting intervals should better make use of the sample, this is a dilemma.

However, according to Theorem 5.5 (and its counterpart for G'_n, Theorem 5.6), $F_a(x)$ could be a CDF used to determine only part of all fitting intervals, no matter how small the ratio is. This gives complete flexibility for almost anything we want to do in the finite sample situation. The only problem is that if $P_G(x)$ is used as the selection guide for fitting intervals, unwanted components may appear at locations where no data points are nearby. This can be overcome by providing enough fitting intervals at locations that may produce components. This, in turn, suggests that support points should avoid locations where there are no data points around. Also, this reduces the number of fitting intervals and hence computational cost.

In fact, for finite samples, using the strictly increasing $P_G(x)$ as the selection guide for fitting intervals suffers from the problem described in the last paragraph. Almost all distributions have dominant parts and dwindle quickly away from the dominant parts, hence they are effectively bounded in finite sample situations. Therefore the strategy discussed in the last paragraph for bounded distributions should be used for all distributions—i.e., the determination of fitting intervals needs to take support points into account, so that all regions within these support points' sphere of influence are covered by a large enough number of fitting intervals. The information about the support points' sphere of influence is available from the component distribution and the number of data points. For regions on which support points have tiny

influence in terms of probability value, it is not necessary to provide fitting intervals there, because the equation

$$\begin{pmatrix} \mathbf{A}_{G_n'} \\ \underline{0} \end{pmatrix} \mathbf{w}_n \approx \begin{pmatrix} \mathbf{p}_n \\ 0 \end{pmatrix} \tag{5.30}$$

has the same solution as $\mathbf{A}_{G_n'} \mathbf{w}_n \approx \mathbf{p}_n$.

Finally, in terms of distance measures, not all of those determined by Conditions 5.5 and 5.6 can provide straightforward algebraic solutions to (5.18) and (5.19). The chosen distance measure determines the type of the mathematical programming problem. If a quadratic distance measure is chosen, a least squares solution under constraints will be pursued. This approach can be viewed as an extension of Choi and Bulgren (1968) from CDF-based to measure-based. Elegant and efficient methods LSE and NNLS provided in Lawson and Hanson (1974, 1995) can be employed. If the sup-norm distance measure is used, solutions can be found by the efficient simplex method in linear programming. This approach can be viewed as an extension of Deely and Kruse (1968).

5.5 The minority cluster problem

This section discusses the minority cluster problem, and illustrates it with simple experiments. The minimum distance approach based on nonnegative measures provides a solution, while other minimum distance ones fail. This problem is vital for pace regression, since ignoring even one isolated data point in the estimated mixing distribution can cause unbounded loss. The discussion below is intended to be intuitive and practical.

5.5.1 The problem

Although they perform well asymptotically, the minimum distance methods described above suffer from the finite-sample problem discussed earlier: they can neglect small groups of outlying data points no matter how far they lie from the dominant data points. The underlying reason is that the objective function to be minimized is defined globally rather than locally. A global approach means that the value of the estimated probability density function at a particular place will be influenced by all data points, no matter how far away they are. This can cause small groups of data points to be ignored even if they are a long way from the dominant part of the data sample. From a probabilistic point of view, however, there is no reason to subsume distant groups within the major clusters just because they are relatively small.

The ultimate effect of suppressing distant minority clusters depends on how the clustering is applied. If the application's loss function depends on the distance between clusters, the result may prove disastrous because there is no limit to how far away these outlying groups may be. One might argue that small groups of points can easily be explained away as outliers, because the effect will become less important as the number of data points increases—and it will disappear in the limit of infinite data. However, in a finite-data situation—and all practical applications necessarily involve finite data—the "outliers" may equally well represent small minority clusters. Furthermore, outlying data points are not really treated as outliers by these methods—whether or not they are discarded is merely an artifact of the global fitting calculation. When clustering, the final mixture distribution should take all data points into account—including outlying clusters if any exist. If practical applications demand that small outlying clusters are suppressed, this should be done in a separate stage.

In distance-based clustering, each data point has a far-reaching effect because of two global constraints. One is the use of the cumulative distribution function; the other is the normalization constraint $\sum_{j=1}^{k_n} w_{nj} = 1$. These constraints may sacrifice

a small number of data points—at any distance—for a better overall fit to the data as a whole. Choi and Bulgren (1968), the Cramer-von Mises statistic (Macdonald, 1971), and Deely and Kruse (1968) all enforce both the CDF and the normalization constraints. Blum and Susarla (1977) drop the CDF, but still enforce the normalization constraint. The result is that these clustering methods are only appropriate for finite mixtures without small clusters, where the risk of suppressing clusters is low.

Here we address the general problem for arbitrary mixtures. Of course, the minority cluster problem exists for all types of mixture—including finite mixtures.

5.5.2 The solution

Now that the source of the problem has been identified, the solution is clear, at least in principle: drop both the approximation of CDFs, as Blum and Susarla (1977) do, and the normalization constraint—no matter how seductive it may seem. This consideration leads to the nonnegative-measure-based approach described in Subsection 5.4.3.

To define the estimation procedure fully, we need to determine (a) the set of support points, (b) the set of fitting intervals, (c) the empirical measure, and (d) the distance measure. Theoretical analysis in the previous sections guarantees a strongly consistent estimator: here we discuss these in an intuitive manner.

Support points. The support points are usually suggested by the data points in the sample. For example, if the component distribution $F(x;\theta)$ is the normal distribution with mean θ and unit variance, each data point can be taken as a support point. In fact, the support points are more accurately described as *potential* support points, because their associated weights may become zero after solving (5.19)—and, in practice, many often do.

Fitting intervals. The fitting intervals are also suggested by the data points. In the normal distribution example with known standard deviation σ, each data point x_i can provide one interval, such as $[x_i - 3\sigma, x_i]$, or two, such as $[x_i - 3\sigma, x_i]$ and

$[x_i, x_i + 3\sigma]$, or more. There is no problem if the fitting intervals overlap. Their length should not be so large that points can exert an influence on the clustering at an unduly remote place, nor so small that the empirical measure is inaccurate. The experiments reported below use intervals of a few standard deviations around each data point, and, as we will see, this works well.

Empirical measure. The empirical measure can be the probability measure determined by the Kolmogorov empirical CDF, or any measure that converges to it. The fitting intervals discussed above can be open, closed, or semi-open. This will affect the empirical measure if data points are used as interval boundaries, although it does not change the values of the estimated measure because the corresponding distribution is continuous. In small-sample situations, bias can be reduced by careful attention to this detail—as Macdonald (1971) discusses with respect to Choi and Bulgren's (1968) method.

Distance measure. The choice of distance measure determines what kind of mathematical programming problem must be solved. For example, a quadratic distance will give rise to a least squares problem under linear constraints, whereas the sup-norm gives rise to a linear programming problem that can be solved using the simplex method. These two measures have efficient solutions that are globally optimal.

5.5.3 Experimental illustration

Experiments are conducted in the following to illustrate the failure of existing minimum distance methods to detect small outlying clusters, and the improvement achieved by the new scheme. The results also suggest that the new method is more accurate and stable than the others.

When comparing clustering methods, it is not always easy to evaluate the clusters obtained. To finesse this problem we consider simple artificial situations in which the proper outcome is clear. Some practical applications of clusters do provide objective evaluation functions—for example, in our setting of empirical modeling

(see Section 5.6).

The methods used are Choi and Bulgren (1968) (denoted CHOI), MacDonald's application of the Cramér-von Mises statistic (CRAMÉR), the method with the probability measure (PM), and the method with the nonnegative measure (NNM). In each case, equations involving non-negativity and/or linear equality constraints are solved as quadratic programming problems using the elegant and efficient procedures NNLS and LSEI provided by Lawson and Hanson (1974, 1995). All four methods have the same computational time complexity.

Experiment 5.1 *Reliability of clustering algorithms.* We set the sample size n to 100 throughout the experiments. The data points are artificially generated from a mixture of two clusters: n_1 points from $N(0,1)$ and n_2 points from $N(100,1)$. The values of n_1 and n_2 are in the ratios $99:1$, $97:3$, $93:7$, $80:20$ and $50:50$.

Every data point is taken as a potential support point in all four methods: thus the number of potential components in the clustering is 100. For PM and NNM, fitting intervals need to be determined. In the experiments, each data point x_i provides two fitting intervals, $[x_i-3, x_i]$ and $[x_i, x_i+3]$. Any data point located on the boundary of an interval is counted as half a point when determining the empirical measure over that interval.

These choices are admittedly crude, and further improvements in the accuracy and speed of PM and NNM are possible that take advantage of the flexibility provided by (5.18) and (5.19). For example, accuracy will likely increase with more—and more carefully chosen—support points and fitting intervals. The fact that it performs well even with crudely chosen support points and fitting intervals testifies to the robustness of the method.

Our primary interest in this experiment is the weights of the clusters that are found. To cast the results in terms of the underlying models, we use the cluster weights to estimate values for n_1 and n_2. Of course, the results often do not contain exactly two clusters—but because the underlying cluster centers, 0 and 100, are well separated

compared to their standard deviation of 1, it is highly unlikely that any data points from one cluster will fall anywhere near the other. Thus we use a threshold of 50 to divide the clusters into two groups: those near 0 and those near 100. The final cluster weights are normalized, and the weights for the first group are summed to obtain an estimate $\hat{n}_1$ of n_1, while those for the second group are summed to give an estimate $\hat{n}_2$ of n_2.

		$n_1 = 99$ $n_2 = 1$	$n_1 = 97$ $n_2 = 3$	$n_1 = 93$ $n_2 = 7$	$n_1 = 80$ $n_2 = 20$	$n_1 = 50$ $n_2 = 50$
CHOI	Failures	86	42	4	0	0
	$\hat{n}_1/\hat{n}_2$	99.9/0.1	99.2/0.8	95.8/4.2	82.0/18.0	50.6/49.4
	SD($\hat{n}_1$)	0.36	0.98	1.71	1.77	1.30
CRAMÉR	Failures	80	31	1	0	0
	$\hat{n}_1/\hat{n}_2$	99.8/0.2	98.6/1.4	95.1/4.9	81.6/18.4	49.7/50.3
	SD($\hat{n}_1$)	0.50	1.13	1.89	1.80	1.31
PM	Failures	52	5	0	0	0
	$\hat{n}_1/\hat{n}_2$	99.8/0.2	98.2/1.8	94.1/5.9	80.8/19.2	50.1/49.9
	SD($\hat{n}_1$)	0.32	0.83	0.87	0.78	0.55
NNM	Failures	0	0	0	0	0
	$\hat{n}_1/\hat{n}_2$	99.0/1.0	96.9/3.1	92.8/7.2	79.9/20.1	50.1/49.9
	SD($\hat{n}_1$)	0.01	0.16	0.19	0.34	0.41

Table 5.1: Experimental results for detecting small clusters

Table 5.1 shows results for each of the four methods. Each cell represents one hundred separate experimental runs. Three figures are recorded. At the top is the number of times the method failed to detect the smaller cluster, that is, the number of times $\hat{n}_2 = 0$. In the middle are the average values for $\hat{n}_1$ and $\hat{n}_2$. At the bottom is the standard deviation of $\hat{n}_1$ and $\hat{n}_2$ (which are equal). These three figures can be thought of as measures of reliability, accuracy and stability respectively.

The top figures in Table 5.1 show clearly that only NNM is always reliable in the sense that it never fails to detect the smaller cluster. The other methods fail mostly when $n_2 = 1$; their failure rate gradually decreases as n_2 grows. The center figures show that, under all conditions, NNM gives a more accurate estimate of the correct values of n_1 and n_2 than the other methods. As expected, CRAMÉR shows a noticeable improvement over CHOI, but it is very minor. The PM method has lower failure rates and produces estimates that are more accurate and far more stable (indicated

by the bottom figures) than those for CHOI and CRAMÉR—presumably because it is less constrained. Of the four methods, NNM is clearly and consistently the winner in terms of all three measures: reliability, accuracy and stability.

The results of the NNM method can be further improved. If the decomposed form (5.29) is used instead of (5.19), and the solutions of the sub-equations are normalized before combining them—which is feasible because the two underlying clusters are so distant from each other—the correct values are obtained for $\hat{n}_1$ and $\hat{n}_2$ in virtually every trial.

5.6 Simulation studies of accuracy

In this section, we study the predictive accuracy of clustering procedures through simulations. Since the component distribution has a limited sphere of influence in finite sample situations, data points should have little effect on the estimation of the probability density function at far away points. Because NNM uses local fitting, it may also have some advantage in accuracy over procedures that adopt global fitting, even in the case of overlapping clusters. We describe the results of two experiments.

Unlike supervised learning, there seems to be no widely accepted criterion for evaluating clustering results (Hartigan, 1996). However, for the special case of estimating a mixing distribution, it is reasonable to employ the quadratic loss function (2.6) used in empirical Bayes or, equivalently, the loss function (3.4) in pace regression (when the variance is a known constant), and this is what we adopt. More details concerning their use are given later.

Experiment 5.2 investigates a mixture of normal distributions with the mean as the mixing parameter. The underlying mixing distribution is normal as well—information that is not used by the clustering procedures we tested. This type of mixture has many practical applications in empirical Bayes analysis, as well as in Bayesian analysis; see, for example, Berger (1985), Maritz and Lwin (1989), and Carlin and Louis (1996). The second experiment considers a mixture of noncentral

χ_1^2 distributions with the noncentrality parameter as the mixing one. The underlying mixing distribution follows Breiman (1995), where he uses it for subset selection. This experiment closely relates to pace regression.

Unlike Experiment 5.1, data points in Experiment 5.2 and 5.3 are mixed so closely that there is no clear grouping. Therefore, reliability is not a concern for them. It is worth mentioning, however, that ignoring even one isolated point in other situations, as in Experiment 5.1, may dramatically increase the loss.

Experiment 5.2 *Accuracy of clustering procedures for mixtures of normal distributions.* We consider the mixture distribution

$$\begin{aligned} x_i|\mu_i &\sim N(\mu_i, 1) \\ \mu_i &\sim N(0, \sigma_\mu^2), \end{aligned}$$

where $i = 1, 2, \ldots, 100$. The simulation results given in Table 5.2 are losses averaged over twenty runs. For each run, 100 data points are generated from the mixture distribution. Given these data points, the mixing distribution is estimated by the four procedures CHOI, CRAMÉR, PM and NNM, and then each x_i is updated to be x_i^{EB} using (2.8). The overall loss for all updated estimates is calculated by (2.6), i.e., $\sum_{i=1}^{100} ||x_i^{\text{EB}} - \mu_i||^2$.

Eleven cases with different values of σ_μ^2 are studied. Generally, as σ_μ^2 increases, the prediction losses increase. This is due to the decreasing number of neighboring points to support updating; see also Section 4.2. For almost every case, PM and NNM perform similarly, both outperforming CHOI and CRAMÉR with a clear margin.

Experiment 5.3 *Accuracy of clustering procedures for mixtures of χ_1^2 distributions.* In this experiment, we consider a mixture of noncentral χ_1^2 distributions using the noncentrality parameter as the mixing one, i.e., the mixture (3.19) or (3.22). The underlying mixing distribution is a modified version of a design by Breiman (1995); see also Experiment 6.6. Since this experiment relates to linear regression, we adopt the notation used for pace regression in Chapter 3.

σ_μ^2	CHOI	CRAMÉR	PM	NNM
0	4.00	3.93	3.27	3.95
1	59.2	59.1	57.4	57.8
2	75.6	75.5	73.4	73.8
3	86.2	86.2	84.2	84.2
4	89.1	88.8	87.0	87.0
5	96.5	96.5	93.6	93.5
6	100.0	100.0	97.7	97.8
7	100.4	100.5	97.8	97.8
8	102.3	101.9	100.3	100.3
9	104.9	104.8	102.1	102.1
10	104.9	104.8	102.0	102.0

Table 5.2: Experiment 5.2. Average loss of clustering procedures over mixtures of normal distributions.

Breiman considers two clustering situations in terms of the β_j ($j = 1, \ldots, k$) values. One, with $k = 20$, has two clusters around 5 and 15 and a cluster at zero. The other, with $k = 40$, has three clusters around 10, 20 and 30 and a cluster at zero. We determine β_j in the same way as Breiman. Then we set $a_j^* = \alpha\beta_j$ and $A_j^* = a_j^{*2}$, where α is always $\frac{1}{3}$ for both cases. The value of α is so chosen that the clusters are neither too far away from nor too close to each other.

The sample of the mixture distribution is the set $\{A_j\}$, where each $A_j = a_j^2$ and $a_j \sim N(a_j^*, 1)$. Each clustering procedure is applied to the given sample, $A_1, A_2, \ldots, A_k$, to produce an estimated mixing distribution $G_k(A^*)$. Then each A_j is updated by PACE$_6$ to obtain $\widetilde{A}_j$. The loss function (3.4) in pace regression, that is the quadratic loss function (2.6), is used to obtain the true prediction loss for each estimate. The signs of $\tilde{a}_j$ and a_j^* are taken into account when calculating the loss, where a_j^* is always positive and $\tilde{a}_j$ has the same sign as a_j. Therefore, the overall loss is $\sum_{j=1}^{k} ||\tilde{a}_j - a_j^*||^2$.

The simulation results are given in Table 5.3, each figure being an average over 100 runs. The variable rc (same as used by Breiman) is the radius of each cluster in the true mixing distribution, i.e., larger rc implies more data points in each cluster. It can be observed from these results that NNM (and probably PM as well) is always a safe method to apply. For small rc, both CHOI and CRAMÉR produce less

rc	CHOI	CRAMÉR	PM	NNM
$k = 20$				
1	15.9	14.1	9.4	11.1
2	13.3	13.1	13.2	13.0
3	15.3	15.7	16.1	16.1
4	17.5	18.0	18.5	18.5
5	19.6	19.9	20.5	20.4
$k = 40$				
1	55.0	48.8	17.4	17.4
2	29.7	29.9	26.5	26.7
3	32.3	32.5	33.9	33.8
4	36.7	37.0	38.4	38.4
5	40.7	40.8	41.4	41.4

Table 5.3: Experiment 5.3. Average losses of clustering procedures over mixtures of χ^2_1 distributions.

accurate or sometimes very bad results—note that in this experiment data points are not easily separable. For large rc, however, CHOI and CRAMÉR do seem to produce slightly more accurate results than the other two. This suggests that CHOI and CRAMÉR may have a small advantage in predictive accuracy when clustering is within a relatively small area and there are no small clusters in this area. This slightly better performance is probably due to CHOI and CRAMÉR's use of larger intervals, when there is no risk of suppressing small amounts of data points.

5.7 Summary

The idea exploited in this chapter is simple: a new minimum distance method which is able to reliably detect small groups of data points, including isolated ones. Reliable detection in this case is important in empirical modeling. Existing minimum distance methods do not cope well with this situation.

The maximum likelihood approach does not seem to suffer from this problem. Nevertheless, it often consumes more computational cost than the minimum distance one. For many applications, fast estimation is important. When pace regression is

embedded in a more sophisticated estimation task, it may be necessary to repeatedly fit linear models.

The predictive accuracy of minimum distance estimators is also empirically investigated in Section 5.6 for situations with overlapping clusters. Although further investigation is needed, NNM seems generally preferable.

Despite the simplicity of the underlying idea, most of this chapter is devoted to a theoretical aspect of new estimators—strong consistency. This is a necessary condition for establishing asymptotic properties of pace regression, and gives a guarantee for large sample behaviour.

Chapter 6

Simulation studies

6.1 Introduction

It is time for some experimental results to illustrate the idea of pace regression and give some indication how it performs in practice. The results are very much in accordance with the theoretical analysis in Chapter 3. In addition, they illustrate some effects that have been discussed in previous chapters.

We will test AIC, BIC, RIC, CIC (in the form (2.5)), PACE_2, PACE_4, and PACE_6. The OLS full model and the null model—which are actually generated by procedures OLS (being $\text{OLSC}(0)$) and $\text{OLSC}(\infty)$—are included for comparison. Procedures PACE_1, PACE_3, and PACE_5 are not included because they involve numerical solution and can be approximated by PACE_2, PACE_4, and PACE_6, respectively.

Two shrinkage methods, NN-GAROTTE and LASSO,[1] are used in Experiments 6.1, 6.2 and 6.7. Because the choice of shrinkage parameter in each case is somewhat controversial, we use the best estimate for this parameter obtained by data resampling—more precisely, five-fold cross-validation over an equally-spaced grid of twenty-one discrete values for the shrinkage ratio. This choice is computationally intensive, which is why results from these methods are not included in all experiments. The Stein estimation version of LASSO (Tibshirani, 1996) is used in

[1]The procedure LASSO is downloaded from StatLib (http://lib.stat.cmu.edu/S/) and NN-GAROTTE from Breiman's ftp site at UCB (ftp://ftp.stat.berkeley.edu/pub/users/breiman).

Experiments 6.1 and 6.2, due to the heavy time consumption of the cross-validation version.[2] It is acknowledged that in these two experiments the cross-validation version may yield better results, as suggested in Experiment 6.7.

We use the partial-F test and backward elimination of variables to set up the orthogonalization, and employ the unbiased $\hat{\sigma}^2$ estimate of the OLS full model. The non-central χ^2 distribution is used instead of the F-distribution for computational reasons. To estimate the mixing distribution $G(A^*)$, we adopt the minimum distance procedure based on nonnegative measure (NNM) given (5.19) or equivalently (5.28) in Chapter 5, along with the quadratic programming algorithm NNLS provided by Lawson and Hanson (1974, 1995). Support points are the set $\{\widehat{A}_j\}$ plus zero (if there is any $\widehat{A}_j$ near zero), except that points lying in the gap between zero and three are discarded in order to simplify the model. This gap helps selection criteria PACE_2 and PACE_4 to properly eliminate dimensions in situations with redundant variables, because when the h-function's sign near the origin is negative—which is crucial for both to eliminate redundant dimensions—it could be incorrectly estimated to be positive (note that $h(0+; A^*) > 0$ when $A^* \geq 0.5$)). This choice is, of course, slightly biased towards situations with redundant variables, but does not sacrifice much prediction accuracy in any situation, and thus can be considered as an application of the simplicity principle (Section 1.2). For PACE_6, $\widetilde{A}_j$'s smaller than 0.5 are discarded to simplify the model as well (see also Section 4.5). These are rough and ready decisions, taken for practical expediency. For more detail, refer to the source code of the S-PLUS implementation in the Appendix.

Recalling the modeling principles that we discussed in Section 1.2, our interests focus mainly on predictive accuracy, and the estimated model complexities appear as a natural by-product.

Both artificial and practical datasets are included in our experiments. One main advantage of using artificial datasets is that the experiments are completely under control. Since the true model is known, the prediction errors can be obtained ac-

[2] In our simulations, NN-GAROTTE takes about two weeks on our computer to obtain the results for Experiments 6.1 and 6.2, while LASSO^{CV} appears to be several times slower.

curately and the estimated model complexity can be compared with the true one. Furthermore, we can design experiments that cover any situations of interest.

Section 6.2 considers artificial datasets that contain only statistically independent variables. Hence the column vectors of the design matrix form a nearly, yet not completely, orthogonal basis.

Section 6.3 investigates two more realistic situations using artificial datasets. In one, studied by Miller (1990), the effects of variables are geometrically distributed. The other considers clusters of coefficients, following the experiment of Breiman (1995).

As any modeling procedures should finally be applied to solve real problems, Section 6.4 extends the investigation to real world datasets. Twelve practical datasets are used to compare the modeling procedures using cross-validation.

6.2 Artificial datasets: An illustration

To illustrate the ideas of pace regression, in this section we give three experimental examples that compare pace regression with other procedures in terms of both prediction accuracy and model complexity, using artificial datasets with statistically independent variables. In the first, contributive variables have small effects; in the second they have large effects. The third example tests the influence of the number of candidate variables on pace regression. Results are given in the form of both tables and graphs.

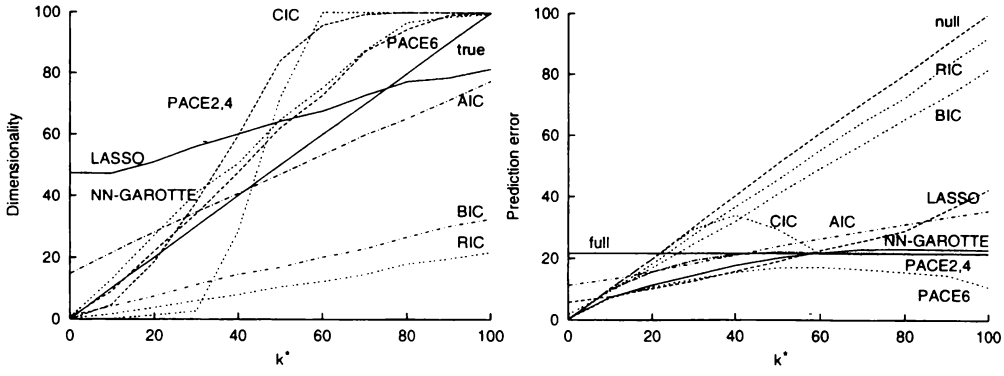
Experiment 6.1 *Variables with small effects.* The underlying model is $y = b_0 + \sum_{j=1}^{k} b_j x_j + N(0, \sigma^2)$, where $b_0 = 0, x_j \sim N(0,1)$, $\mathrm{Cov}(x_i, x_j) = 0$ for $i \neq j$, and $\sigma^2 = 200$. Each parameter b_j is either 1 or 0, depending on whether it has an effect or is redundant. For each sample, $n = 1000$ and $k = 100$ (excluding b_0 which is always included in the model). The number of non-redundant variables k^* is set to $0, 10, 20, \ldots, 100$. For each value, the result is the average of twenty runs over twenty independent training samples, tested on a large independent set. This test

k^*	0	10	20	30	40	50	60	70	80	90	100
Dim											
NN-GAR.	0.6	12.8	27.2	40.9	50.5	64.7	75.2	87.6	96.7	98.5	99.6
LASSO$^{\text{Stein}}$	47.4	47.2	51.0	56.0	60.0	64.2	67.6	72.8	77.4	78.6	81.5
AIC	14.8	21.4	28.1	34.5	40.6	46.9	53.4	59.9	65.2	71.2	77.6
BIC	1.1	4.2	7.4	11.1	14.5	16.7	20.1	22.9	26.6	30.1	32.5
RIC	0.2	1.6	3.7	6.0	7.9	10.4	12.2	14.6	17.9	19.6	21.8
CIC	0.0	0.2	1.3	2.6	28.4	72.0	100	100	100	100	100
PACE_2	0.0	4.6	18.6	37.8	59.6	84.0	95.7	99.4	100	100	100
PACE_4	0.0	4.6	18.8	37.9	59.6	84.2	95.8	99.4	100	100	100
PACE_6	0.6	8.9	22.1	34.0	47.6	62.0	72.8	87.2	94.5	99.1	100
PE											
NN-GAR.	0.4	7.3	11.3	14.4	17.6	20.1	21.9	23.0	23.1	23.0	22.9
LASSO$^{\text{Stein}}$	5.69	7.46	10.1	12.5	15.7	18.9	22.6	25.6	29.0	35.8	42.6
full	21.6	21.6	21.6	21.6	21.6	21.6	21.6	21.6	21.6	21.6	21.6
AIC	11.2	13.1	15.6	18.2	20.9	23.8	26.4	28.8	31.1	33.5	35.7
BIC	1.7	9.1	16.9	24.5	31.8	41.0	49.4	57.3	65.6	73.3	82.1
RIC	0.5	9.3	18.2	27.5	36.6	45.6	55.4	64.6	72.6	83.5	92.5
CIC	0.0	10.1	19.7	29.3	33.9	29.3	21.6	21.6	21.6	21.6	21.6
PACE_2	0.0	9.8	15.6	19.3	21.1	22.1	21.8	21.6	21.6	21.6	21.6
PACE_4	0.0	9.8	15.6	19.3	21.3	22.2	22.0	22.0	21.8	21.8	21.7
PACE_6	0.1	7.0	10.6	13.2	15.4	16.9	17.1	16.5	15.5	14.5	10.7

Table 6.1: Experiment 6.1. Average dimensionality and prediction error of the estimated models.

set is generated from the same model structure with 10,000 observations and zero noise variance, and hence can be taken as the true model. The results are shown in Figure 6.1 and recorded in Table 6.1.

Figure 6.1(a) and Figure 6.1(b) show the dimensionality (Dim) of the estimated models and their prediction errors (PE), respectively; the horizontal axis is k^* in both cases. In Figure 6.1(a), the solid line gives the size of the underlying models. Since prediction accuracy rather than model complexity is used as the standard for modeling, the best estimated model does not necessarily have the same complexity as the underlying one—dimensionality reduction is merely a byproduct of parameter estimation. Models generated by PACE_2, PACE_4 and PACE_6 find the underlying null hypothesis H_0 and the underlying full hypothesis H_f correctly; their seeming inconsistency between these extremes is in fact necessary for the estimated models to produce optimal predictions. AIC overfits H_0 and underfits H_f. BIC and RIC



(a) (b)

Figure 6.1: Experiment 6.1, $k = 100, \sigma^2 = 200$. (a) Dimensionality. (b) Prediction error.

both fit H_0 well, but dramatically underfit all the other hypotheses—including H_f. CIC successfully selects both H_0 and H_f but either underfits or overfits models in between. NN-GAROTTE generally chooses slightly larger models than PACE_6, but $\text{LASSO}^{\text{Stein}}$ significantly overfits for small k^* and underfits for large k^*.

In Figure 6.1(b), the vertical axis represents the average mean squared error in predicting independent test sets. The models built by BIC and RIC have errors nearly as great as the null model. AIC is slightly better for H_f than BIC and RIC, but fails on H_0. CIC eventually coincides with the full model as k^* increases, and produces large errors for some model structures between H_0 and H_f. It is interesting to note that PACE_2 always performs as well as the best of $\text{OLSC}(\infty)$ (the null model), RIC, BIC, AIC, CIC, and OLS (the full model): no OLS subset selection procedures produce models that are sensibly better. Recall that PACE_2 selects the optimal subset model from the sequence, and in this respect resembles PACE_1, which is the optimal threshold-based OLS subset selection procedure $\text{OLSC}(\tau^*)$. PACE_4 performs similarly to PACE_2. Remarkably, PACE_6 outperforms PACE_2 and PACE_4 by a large margin, even when there are no redundant variables. The shrinkage methods NN-GAROTTE and $\text{LASSO}^{\text{Stein}}$ are competitive with PACE_2 and PACE_4—better for small k^* and worse for large k^*—but can never outperform PACE_6 in any practical sense, which is consistent with Corollary 3.10.2 and Theorem 3.11.

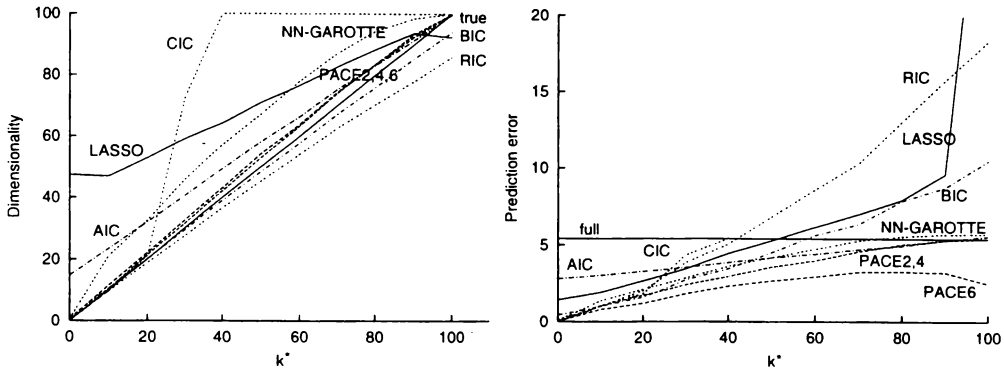
k^*	0	10	20	30	40	50	60	70	80	90	100
Dim											
NN-GAR.	0.6	21.0	32.6	45.7	57.1	67.2	78.0	87.0	94.3	98.5	100
LASSO$^{\text{Stein}}$	47.4	46.8	52.8	59.0	64.2	71.0	76.6	82.6	88.3	93.7	92.3
AIC	14.8	23.4	32.0	40.5	49.2	58.2	66.4	75.3	83.0	91.4	99.7
BIC	1.1	10.6	20.1	29.4	39.0	48.4	57.8	67.2	75.9	85.4	94.0
RIC	0.2	9.6	18.8	27.4	36.6	45.3	54.0	62.5	70.3	78.0	86.0
CIC	0.0	9.6	21.1	72.4	100	100	100	100	100	100	100
PACE$_2$	0.0	10.3	21.4	31.5	42.0	53.0	63.2	74.0	83.4	93.3	100
PACE$_4$	0.0	10.3	21.4	31.5	42.1	53.0	63.4	74.0	83.5	93.3	100
PACE$_6$	0.6	11.5	22.1	32.6	43.0	54.2	63.9	74.3	83.4	92.5	100
PE											
NN-GAR.	0.10	1.35	2.12	2.83	3.61	4.17	4.72	5.27	5.56	5.70	5.71
LASSO$^{\text{Stein}}$	1.42	1.91	2.65	3.43	4.43	5.27	6.15	7.00	7.94	9.61	35.60
full	5.40	5.40	5.40	5.40	5.40	5.40	5.40	5.40	5.40	5.40	5.40
AIC	2.80	3.00	3.27	3.53	3.85	4.16	4.40	4.70	4.95	5.32	5.56
BIC	0.43	1.00	1.59	2.71	3.44	4.50	5.62	6.35	7.88	8.76	10.5
RIC	0.12	1.00	2.03	3.82	5.04	6.81	8.59	10.3	13.1	15.8	18.3
CIC	0.00	0.99	1.83	4.31	5.40	5.40	5.40	5.40	5.40	5.40	5.40
PACE$_2$	0.00	1.06	1.71	2.40	2.95	3.56	3.96	4.57	5.03	5.29	5.40
PACE$_4$	0.00	1.06	1.71	2.40	2.92	3.58	3.98	4.57	5.04	5.29	5.40
PACE$_6$	0.03	0.80	1.19	1.82	2.31	2.66	2.92	3.24	3.28	3.22	2.48

Table 6.2: Experiment 6.2. Average dimensionality and prediction error of the estimated models.

Experiment 6.2 *Variables with large effects.* In this experiment the underlying model is the same as in the last experiment except that $\sigma^2 = 50$; results are shown in Figure 6.2 and Table 6.2.

The results are similar to those in the previous example. As Figure 6.2(a) shows, models generated by PACE$_2$, PACE$_4$ and PACE$_6$ lie closer to the line of underlying models than in the last example. AIC generally overfits by an amount that decreases as k^* increases. RIC and BIC generally underfit by an amount that increases as k^* increases. CIC settles on the full model earlier than in the last example. NN-GAROTTE and LASSO$^{\text{Stein}}$ generally choose models of larger dimensionalities than PACE$_2$, PACE$_4$ and PACE$_6$.

In terms of prediction error (Figure 6.2(b)), PACE$_2$ and PACE$_4$ are still the best of all OLS subset selection procedures, and are almost always better and never meaningfully worse than NN-GAROTTE and LASSO$^{\text{Stein}}$ (see also the discussion of shrinkage



(a) (b)

Figure 6.2: Experiment 6.2. $k = 100, \sigma^2 = 50$. (a) Dimensionality. (b) Prediction error.

estimation in situations with different variable effects; Example 3.1), while PACE_6 is significantly superior. When there are no redundant variables, PACE_6 chooses a full-sized model but uses different coefficients from OLS, yielding a model with much smaller prediction error than the OLS full model. This defies conventional wisdom, which views the OLS full model as the best possible choice when all variables have large effects.

Experiment 6.3 *Rate of convergence.* Our third example explores the influence of the number of candidate variables. The value of k is chosen to be $10, 20, \ldots, 100$ respectively, and for each value k^* is chosen as 0, $k/2$ and k; otherwise the experimental conditions are as in Experiment 6.1. Note that variables with small effects are harder to distinguish in the presence of noisy variables. Results are shown in Figure 6.3 and recorded in Table 6.3.

The pace regression procedures are always among the best in terms of prediction error, a property enjoyed by none of the conventional procedures. None of the pace procedures suffer noticeably as the number of candidate variables k decreases. Apparently, they are stable when k is as small as ten.

	k	10	20	30	40	50	60	70	80	90	100
Dim	AIC	1.7	2.5	5.0	5.8	7.8	8.2	12.3	10.9	13.2	14.8
($k^* = 0$)	BIC	0.0	0.2	0.3	0.2	0.2	0.5	0.8	0.5	0.9	1.1
	RIC	0.2	0.3	0.3	0.2	0.2	0.1	0.2	0.2	0.3	0.2
	CIC	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0
	$PACE_2$	0.9	0.3	0.3	0.2	0.1	0.1	0.1	0.1	0.1	0.0
	$PACE_4$	0.9	0.3	0.3	0.2	0.1	0.1	0.1	0.1	0.1	0.0
	$PACE_6$	0.8	0.7	1.1	0.6	0.3	0.6	1.4	0.5	0.6	0.6
PE	AIC	1.1	2.0	3.5	4.4	5.8	6.1	9.5	8.5	10.7	11.2
($k^* = 0$)	BIC	0.0	0.5	0.5	0.5	0.5	0.7	1.4	0.8	1.6	1.7
	RIC	0.3	0.6	0.5	0.5	0.4	0.3	0.6	0.4	0.7	0.5
	CIC	0.0	0.1	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0
	$PACE_2$	0.5	0.5	0.4	0.4	0.1	0.2	0.2	0.1	0.1	0.0
	$PACE_4$	0.5	0.5	0.4	0.4	0.1	0.2	0.2	0.1	0.1	0.0
	$PACE_6$	0.2	0.2	0.3	0.2	0.2	0.1	0.4	0.1	0.2	0.1
Dim	AIC	4.8	8.9	14.6	18.8	23.4	27.2	33.2	37.0	40.2	46.9
($k^* = \frac{k}{2}$)	BIC	2.0	3.5	6.2	6.5	7.9	10.6	12.0	13.3	13.9	16.7
	RIC	3.0	4.2	6.2	5.8	6.7	8.0	8.3	9.7	8.7	10.4
	CIC	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0
	$PACE_2$	7.2	14.6	22.3	32.8	41.0	45.9	57.9	59.1	65.6	84.0
	$PACE_4$	7.2	14.6	22.3	32.8	41.0	46.0	58.0	59.1	65.8	84.2
	$PACE_6$	5.9	11.1	18.2	24.8	31.4	34.0	44.8	46.6	51.0	62.0
PE	AIC	2.3	4.7	7.2	9.0	11.4	13.8	17.1	18.1	22.0	23.8
($k^* = \frac{k}{2}$)	BIC	3.5	8.0	11.0	15.8	20.2	22.9	27.8	32.2	37.3	41.0
	RIC	2.8	7.5	10.9	16.5	20.9	24.7	30.4	35.3	41.8	45.6
	CIC	2.8	5.4	8.0	11.3	12.5	16.9	19.7	21.6	29.4	29.3
	$PACE_2$	2.1	4.5	6.8	8.3	10.7	12.6	16.0	16.9	19.4	22.1
	$PACE_4$	2.1	4.5	6.8	8.3	10.8	12.7	16.2	16.9	19.5	22.2
	$PACE_6$	1.8	3.4	5.3	6.8	8.3	9.6	12.2	12.9	15.9	16.9
Dim	AIC	7.7	15.6	23.6	31.0	38.6	45.9	52.0	60.9	67.5	77.6
($k^* = k$)	BIC	3.3	7.2	11.1	13.2	15.2	19.9	22.1	24.4	27.4	32.5
	RIC	5.3	8.5	11.3	12.3	13.1	15.6	16.6	18.1	18.8	21.8
	CIC	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100
	$PACE_2$	9.4	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100
	$PACE_4$	9.4	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0	100
	$PACE_6$	9.2	20.0	29.9	39.6	49.9	60.0	70.0	80.0	90.0	100
PE	full	2.0	4.0	6.4	8.2	10.4	12.1	16.2	16.6	19.1	21.6
($k^* = k$)	AIC	3.5	7.0	10.3	13.8	17.1	21.2	26.7	28.3	33.1	35.7
	BIC	7.7	15.3	22.8	31.3	40.5	47.2	56.4	65.7	74.0	82.1
	RIC	5.6	13.9	22.6	32.1	42.5	51.3	61.9	71.8	82.4	92.5
	CIC	2.0	4.0	6.4	8.2	10.4	12.1	16.2	16.6	19.1	21.6
	$PACE_2$	2.3	4.0	6.4	8.2	10.4	12.1	16.2	16.6	19.1	21.6
	$PACE_4$	2.3	4.0	6.4	8.3	10.6	12.3	16.4	16.9	19.4	21.7
	$PACE_6$	1.2	1.0	2.6	2.7	4.5	5.2	7.8	8.0	10.4	10.7

Table 6.3: Experiment 6.3—for the true hypotheses, $k^* = 0$, $\frac{k}{2}$, and k, respectively.

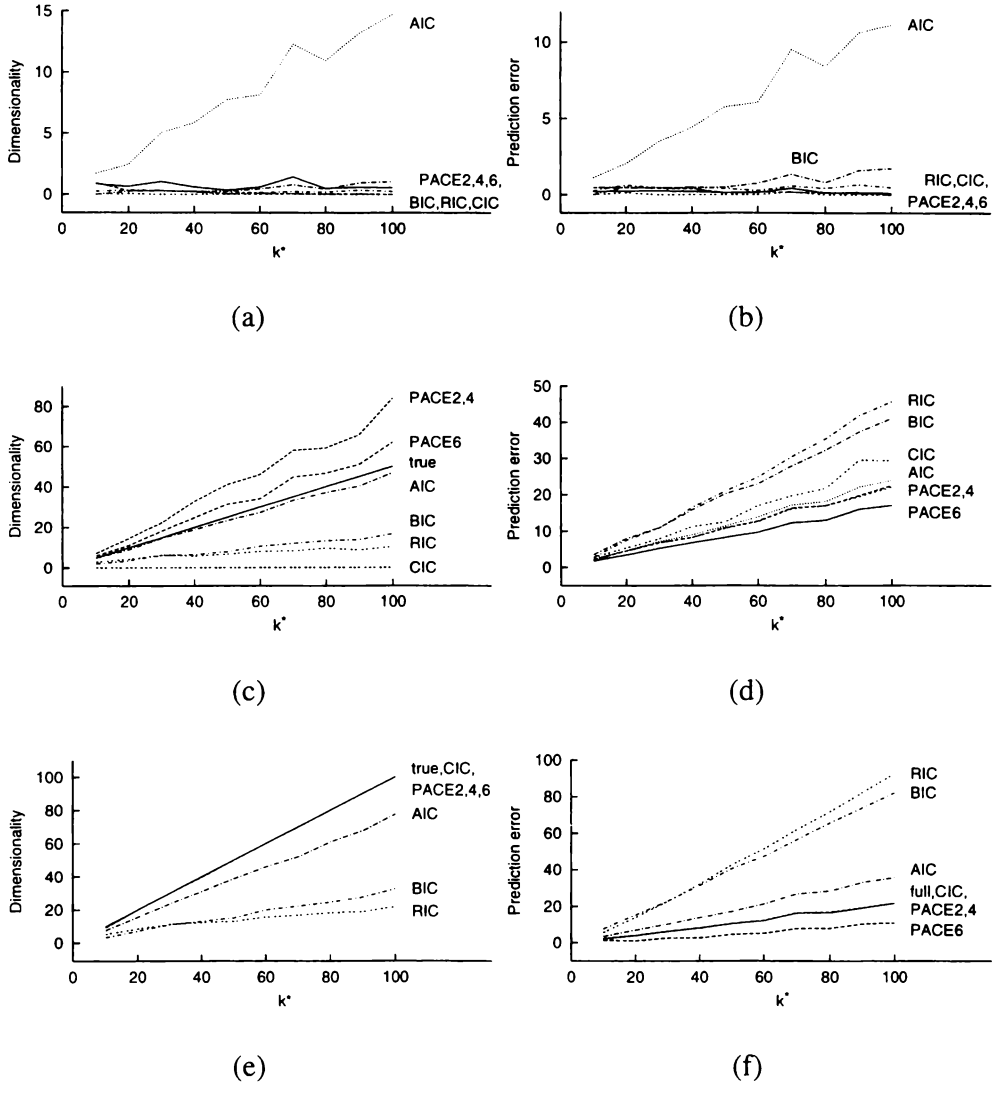


Figure 6.3: Experiment 6.3. (a), (c) and (e) show the dimensionalities of the estimated models when the underlying models are the null, half, and full hypotheses. (b), (d) and (f) show corresponding prediction errors over large independent test sets.

6.3 Artificial datasets: Towards reality

Miller (1990) considers an artificial situation for subset selection that assumes "geometric progression of values of the true projections", i.e., the geometric distribution of A^*. Despite the fact that, as he points out (p.181), "there is no evidence that this pattern is realistic", it is probably closer to reality than the models in the last section, as evidenced by the distribution of eigenvalues of a few practical datasets given in Table 6.2, Miller, 1990. Hence this section tests how pace regression compares with other methods in this artificial situation. Real world datasets will be tested in the next section.

We know that the extent of upgrading each $\widehat{A}_j$ by pace regression basically depends on the number of $\widehat{A}$ around it. In particular, PACE_2 and PACE_4, both essentially selection procedures, rely more on the number around the true threshold value τ^*. Experiment 6.4 uses a situation in which there are relatively more A_j^* around τ^*, so that PACE_2 and PACE_4 should perform similarly to how they would if τ^* were known. Experiment 6.5 roughly follows Miller's idea, in which case, however, few A_j^* are around τ^*, and hence pace regression will produce less satisfactory results.

Breiman (1995) investigates another artificial situation in which the nonzero true coefficient values are distributed in two or three groups, and the column vectors of X are normally, but not independently, distributed. Experiment 6.6 follows this idea.

For the experiments in this section, we include results for both the x-fixed and x-random situations. From these results, it can be seen that all procedures perform almost identically in both situations. We should not conclude, of course, that this will remain true in other cases, in particular for small n.

Experiment 6.4 *Geometrically distributed effects.* We now investigate a simple geometric distribution of effect. Except where described below, all other options in this experiment are the same as used previously.

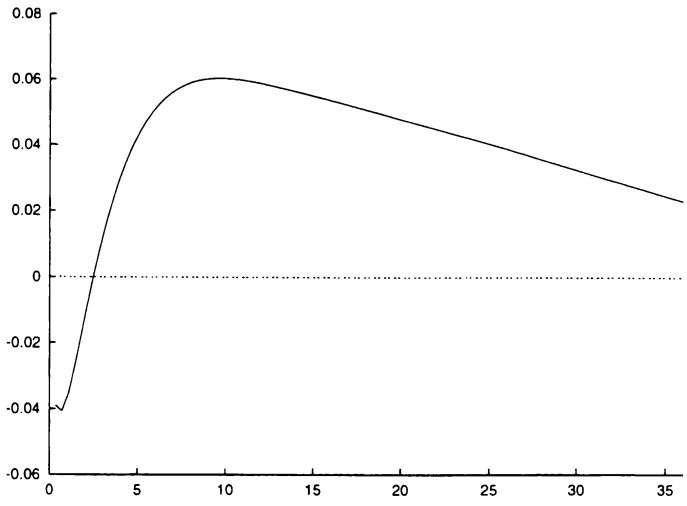


Figure 6.4: $h(\widehat{A}; G)$ in Experiment 6.4.

The linear model with 100 explanatory variables is used, i.e.,

$$y = \sum_{j=1}^{100} \beta_j x_j + \epsilon, \tag{6.1}$$

where the noise component $\epsilon \sim N(0, \sigma^2)$. The probability density function of $\widehat{A}$ is determined by the following mixture model

$$\begin{aligned} 100 \times f(\widehat{A}; G) \;=\; & 50 \times f(\widehat{A}; 0) + 25 \times f(\widehat{A}; 1) + 13 \times f(\widehat{A}; 4) + 6 \times f(\widehat{A}; 9) + \\ & 3 \times f(\widehat{A}; 16) + 2 \times f(\widehat{A}; 25) + 1 \times f(\widehat{A}; 36), \end{aligned} \tag{6.2}$$

i.e., $A^* = 0, 1, 4, 9, 16, 25$ and 36 with probabilities .5, .25, .13, .6, .3, .2 and .1 respectively.

The corresponding mixture h-function is shown in Figure 6.4. It can be seen that $\tau^* \approx 2.5$, for OLSC(τ). Hence this situation seems favorable to AIC, for (n-asymptotically) AIC = OLSC(2), which is nearly the optimal *selection* procedure here. Since here BIC = OLSC($\log n$) = OLSC(6.9) and RIC = OLSC($2 \log k$) = OLSC(9.2), both should be inferior to AIC.

The effects $A^* = 0, 1, 4, 9, 16, 25, 36$ are converted to parameters through the ex-

Procedure	x-fixed model		x-random model	
	Dim	PE	Dim	PE
full (OLS)	100.0	1.03	100.0	1.11
null	0.0	2.73	0.0	2.68
AIC	37.8	0.96	37.2	1.03
BIC	15.4	1.09	14.6	1.16
RIC	11.3	1.21	10.9	1.30
CIC	13.7	1.15	12.4	1.26
PACE_2	45.2	1.00	41.2	1.05
PACE_4	45.3	1.03	41.4	1.08
PACE_6	39.2	0.80	36.7	0.84

Table 6.4: Results of Experiment 6.4.

pression

$$\beta_j = \sqrt{\frac{A_j^* \sigma^2}{n}}. \tag{6.3}$$

We let $x_j \sim N(0,1)$ independently for all j. For each training set, let $\sigma^2 = 10$ and the number of observations $n = 1000$. Then the true model becomes

$$\begin{aligned} y \;=\; & 0 \times (x_1 + \cdots + x_{50}) + 0.1 \times (x_{51} + \cdots + x_{75}) + 0.2 \times (x_{76} + \cdots + x_{88}) + \\ & 0.3 \times (x_{89} + \cdots + x_{94}) + 0.4 \times (x_{95} + x_{96} + x_{97}) + 0.5 \times (x_{98} + x_{99}) + \\ & 0.6 \times x_{100} + \epsilon. \end{aligned} \tag{6.4}$$

To test the estimated model, two situations are considered: x-fixed and x-random. In the x-fixed situation, the training and test sets share the same X matrix, but only the training set contains the noise component ϵ. In the x-random situation, a large, independent, noise-free test set ($n = 10000$, and different X from the training set) is generated, and the coefficients in (6.4) are used to obtain the response vector.

Table 6.4 presents the experimental results. Each figure in the table is the average of 100 runs. The model dimensionality Dim is the number of remaining variables, excluding the intercept, in the trained model. PE is the mean squared prediction error by the trained model over the test set.

As shown in Table 6.4, the null model is clearly the worst in terms of prediction

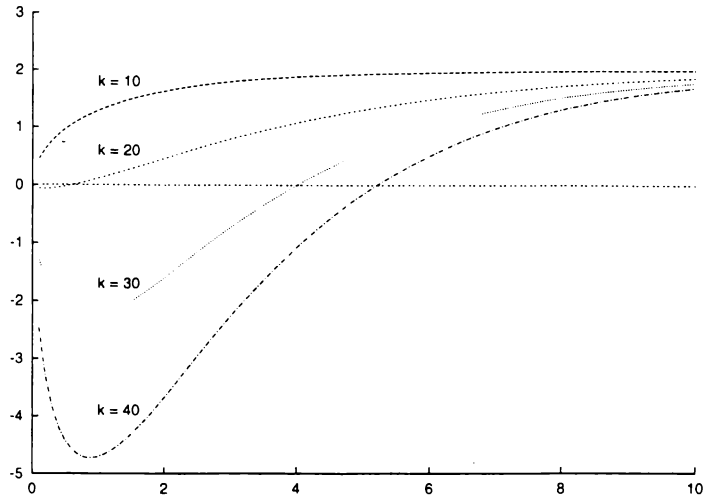


Figure 6.5: The mixture h-function for $k = 10, 20, 30, 40$ predictors, and $A_j^* = (10\alpha^{j-1})^2$ with $\alpha = 0.8$.

error. The full model is relatively good, even better than BIC, RIC and CIC—this is due to the high threshold values employed by the latter. AIC and three pace estimators are better than the full model. AIC, PACE_2 and PACE_4 perform similarly: AIC is slightly better because the other two have to estimate the threshold value while it uses a nearly optimal threshold value. PACE_6 beats all others by a clear margin, although it chooses models of similar dimensionality to AIC, PACE_2 and PACE_4.

Experiment 6.5 *Geometrically distributed effects, Miller's case.* We now look at Miller's (1990, pp.177–181) case where effects are distributed geometrically. In our experiment, however, we examine the situation with independent explanatory variables, whereas Miller (pp.186-187) performs similarity transformations on the variance-covariance matrix. Correlated explanatory variables will be inspected in Experiments 6.6 and 6.7.

Miller's assignment of the effects is equivalent to this: for a model with k predictors (excluding the constant term), the underlying $A_j^* = (10\alpha^{j-1})^2$ for $j = 1, \ldots, k$. In particular, he considers the situations $k = 10, 20, 30$ or 40 and $\alpha = 0.8$ or 0.6; i.e., for $\alpha = 0.8$, $A_j^* = 100, 64, 41, 26, 17, 11, 6.9, 4.4, 2.8, 1.8, \ldots$, and for $\alpha = 0.6$,

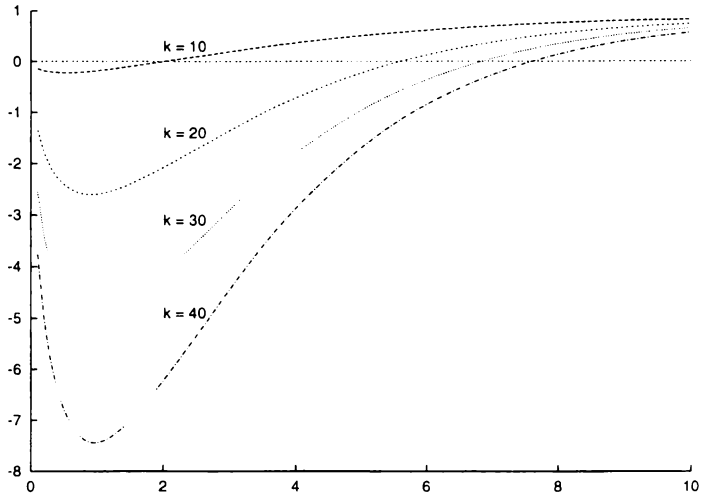


Figure 6.6: The mixture h-function for $k = 10, 20, 30, 40$ predictors, and $A_j^* = (10\alpha^{j-1})^2$ with $\alpha = 0.6$.

	$\alpha = 0.8$				$\alpha = 0.6$			
	$k = 10$	$k = 20$	$k = 30$	$k = 40$	$k = 10$	$k = 20$	$k = 30$	$k = 40$
$\tau^* \approx$	0	0.8	3.5	5	2	5.5	6.5	7.5

Table 6.5: τ^* for $\alpha = 0.8$ and 0.6 respectively in Experiment 6.5.

$A_j^* = 100, 36, 13, 4.7, 1.7, 0.6, 0.22, \ldots$. Figure 6.5 and 6.6 show the mixture h-functions for the $\alpha = 0.8$ and 0.6 situations respectively. From both figures, we can roughly estimate the values of τ^* for each situation, given in Table 6.5. When the threshold value of a model selection criterion is near τ^*, this criterion should resemble the optimal selection criterion in performance. This observation helps to explain the experimental results.

As in Experiment 6.4, each β_j is chosen according to (6.3), where $n = 1000$ and $\sigma^2 = 10$. Therefore the model for $\alpha = 0.8$ is of the form

$$\begin{aligned} y &= 1x_1 + 0.8x_2 + 0.64x_3 + 0.512x_4 + 0.41x_5 + 0.33x_6 + 0.26x_7 + \\ &\quad 0.21x_8 + 0.17x_9 + 0.13x_{10} + \cdots + 0.8^{k-1}x_k + \epsilon, \end{aligned} \tag{6.5}$$

and the model for $\alpha = 0.6$ is of the form

$$y = 1x_1 + 0.6x_2 + 0.36x_3 + 0.22x_4 + 0.13x_5 + 0.078x_6 + 0.047x_7 + 0.028x_8 + 0.017x_9 + 0.01x_{10} + \cdots + 0.6^{k-1}x_k + \epsilon. \quad (6.6)$$

Training and test sets are generated as in Experiment 6.4.

The experimental results are given in Table 6.6 and 6.7 for $\alpha = 0.8$ and 0.6 respectively. Among all these results, PACE_6 is clearly the best: it never (sensibly) loses to any other procedures, and often wins by a clear margin. In some cases, procedures PACE_2 and PACE_4 are inferior to the best selection criterion—despite the fact that, theoretically, they are k-asymptotically the best selection criteria. This is the price that pace regression pays for estimating $G(A^*)$ in the finite k situation. It could be slightly inferior to other selection procedures which have the threshold values near τ^*; nevertheless, no individual one of the OLS, null, AIC, BIC = $\text{OLSC}(6.9)$, RIC = $\text{OLSC}(2\log k)$ and CIC selection criteria can always be optimal.

The accuracy of pace estimators relies on the accuracy of the estimated $G(A^*)$, which further depends on the number of predictors (i.e., the number of A_j's)—more precisely, the number of A_j's around a particular point of interest, for example, τ^* for PACE_2 and PACE_4. In this experiment, there are often few data points close enough to τ^* to provide enough information for accurately estimating $G(A^*)$ around τ^*. In the cases $\alpha = 0.6$ and $k = 30$ and 40, for example, $\tau^* \approx 6.5$ and 7.5, but we only have $A_j^* = 100, 36, 13, 4.7, 1.7, 0.6, 0.22, \ldots$. For samples distributed like this, the clustering result around τ^* is unlikely to be accurate, implying higher expected loss for pace estimators.

Experiment 6.6 *Breiman's case.* This experiment explores a situation considered by Breiman (1995, pp.379–382).

According to Breiman (1995, p.379), the "values of the coefficients were renormalized so that in the X-controlled case, the average R^2 was around .85, in the X-random, about .75." Since details of the "re-normalization" are not given, the

Procedure	k	x-fixed model		x-random model	
		Dim	PE	Dim	PE
full (OLS)		10	0.11	10	0.11
null		0	2.79	0	2.72
AIC		8.61	0.13	8.6	0.14
BIC		6.53	0.24	6.61	0.24
RIC	10	7.37	0.18	7.53	0.19
CIC		9.87	0.11	9.81	0.12
$PACE_2$		9.45	0.12	9.44	0.12
$PACE_4$		10	0.11	10	0.11
$PACE_6$		10	0.11	10	0.11
full (OLS)		20	0.20	20	0.22
null		0	2.80	0	2.74
AIC		10.8	0.20	11.0	0.22
BIC		6.83	0.27	6.71	0.29
RIC	20	7.10	0.25	7.23	0.26
CIC		10.6	0.21	10.6	0.23
$PACE_2$		12.2	0.21	12.1	0.23
$PACE_4$		12.2	0.21	12.1	0.23
$PACE_6$		11.5	0.20	11.3	0.22
full (OLS)		30	0.31	30	0.30
null		0	2.75	0	2.74
AIC		12.7	0.27	12.1	0.25
BIC		7.06	0.27	7.15	0.27
RIC	30	7.11	0.27	7.21	0.27
CIC		8.40	0.26	8.42	0.26
$PACE_2$		13.4	0.28	11.6	0.26
$PACE_4$		13.4	0.28	11.6	0.26
$PACE_6$		12.4	0.23	11.3	0.23
full (OLS)		40	0.41	40	0.43
null		0	2.78	0	2.79
AIC		14.1	0.33	13.8	0.34
BIC		7.17	0.28	6.96	0.31
RIC	40	6.99	0.29	6.73	0.31
CIC		7.52	0.29	7.26	0.31
$PACE_2$		11.2	0.32	11.1	0.33
$PACE_4$		11.3	0.31	11.2	0.33
$PACE_6$		11.5	0.26	11.3	0.28

Table 6.6: Experiment 6.5. Results for $k = 10, 20, 30, 40$ predictors, and $A_j^* = (10\alpha^{j-1})^2$ with $\alpha = 0.8$.

Procedure	k	x-fixed model		x-random model	
		Dim	PE	Dim	PE
full (OLS)		10	0.11	10	0.11
null		0	1.57	0	1.61
AIC		4.92	0.10	5.13	0.11
BIC		3.23	0.13	3.44	0.13
RIC	10	3.77	0.12	3.96	0.11
CIC		5.12	0.11	5.3	0.11
$PACE_2$		5.28	0.12	5.75	0.11
$PACE_4$		10	0.11	10	0.11
$PACE_6$		10	0.11	10	0.11
full (OLS)		20	0.20	20	0.22
null		0	1.57	0	1.60
AIC		6.35	0.16	6.56	0.17
BIC		3.24	0.13	3.48	0.14
RIC	20	3.5	0.13	3.76	0.13
CIC		3.4	0.14	3.76	0.14
$PACE_2$		4.54	0.15	5.37	0.16
$PACE_4$		4.66	0.14	5.53	0.15
$PACE_6$		4.72	0.13	5.56	0.13
full (OLS)		30	0.31	30	0.30
null		0	1.55	0	1.54
AIC		8.5	0.22	7.92	0.20
BIC		3.59	0.14	3.37	0.15
RIC	30	3.61	0.14	3.41	0.15
CIC		3.28	0.15	3.12	0.16
$PACE_2$		4.59	0.18	3.88	0.18
$PACE_4$		4.73	0.16	4.05	0.17
$PACE_6$		5.48	0.14	4.72	0.14
full (OLS)		40	0.41	40	0.43
null		0	1.58	0	1.59
AIC		10.2	0.29	9.71	0.29
BIC		3.63	0.16	3.58	0.16
RIC	40	3.52	0.16	3.44	0.16
CIC		3.12	0.16	3.03	0.17
$PACE_2$		4.33	0.23	3.84	0.23
$PACE_4$		4.73	0.19	4.24	0.19
$PACE_6$		5.47	0.16	5.09	0.16

Table 6.7: Experiment 6.5. Results for $k = 10, 20, 30, 40$ predictors, and $A_j^* = (10\alpha^{j-1})^2$ with $\alpha = 0.6$.

Procedure	$rc = 1$		$rc = 2$		$rc = 3$		$rc = 4$		$rc = 5$	
	Dim	PE	Dim	PE	Dim	PE	Dim	PE	Dim	PE
	x-fixed ($k = 20, n = 40$)									
full (OLS)	20	0.54	20	0.54	20	0.54	20	0.54	20	0.54
null	0	3.65	0	4.00	0	4.88	0	5.76	0	6.58
AIC	4.77	0.32	6.4	0.42	7.45	0.48	8.00	0.51	8.69	0.53
BIC	3.01	0.21	3.97	0.36	5.07	0.46	5.86	0.53	6.38	0.58
RIC	2.32	0.14	2.98	0.34	3.83	0.49	4.41	0.59	4.84	0.67
CIC	2.25	0.14	3.04	0.37	3.92	0.53	5.09	0.62	6.01	0.64
$PACE_2$	3.47	0.20	5.46	0.41	6.89	0.51	8.19	0.57	9.41	0.58
$PACE_4$	3.62	0.20	5.74	0.40	7.22	0.51	8.61	0.56	9.72	0.58
$PACE_6$	3.76	0.16	5.46	0.34	6.99	0.46	8.17	0.51	8.99	0.54
	x-random ($k = 40, n = 80$)									
full (OLS)	40	1.06	40	1.06	40	1.06	40	1.06	40	1.06
null	0	0.41	0	2.60	0	3.22	0	3.85	0	4.43
AIC	8.46	0.48	10.2	0.55	12.1	0.67	13.1	0.75	13.9	0.80
BIC	3.94	0.30	5.51	0.38	6.74	0.52	7.55	0.64	8.06	0.71
RIC	2.27	0.26	3.64	0.31	4.85	0.50	5.25	0.64	5.60	0.76
CIC	1.22	0.30	3.38	0.31	4.47	0.53	5.17	0.67	6.03	0.76
$PACE_2$	3.50	0.36	5.83	0.41	8.31	0.60	10.8	0.75	12.7	0.80
$PACE_4$	4.55	0.34	6.45	0.40	9.08	0.59	11.7	0.74	13.4	0.80
$PACE_6$	5.04	0.27	7.37	0.33	9.72	0.47	11.5	0.60	13.0	0.66

Table 6.8: Experiment 6.6, both the x-fixed and x-random situations

normalization factor α used here sets $\beta_j^* = \alpha\beta_j$, for $j = 1, \ldots, k$, so that

$$R^2 \approx \frac{n\alpha^2 \sum_{j=1}^{k} {\beta_j}^2}{n\alpha^2 \sum_{j=1}^{k} {\beta_j}^2 + k}, \tag{6.7}$$

that is,

$$\alpha(n, k, R^2) \approx \sqrt{\frac{k}{n(1 - R^2) \sum_{j=1}^{k} {\beta_j}^2}}. \tag{6.8}$$

Therefore, in the x-fixed case

$$\alpha(40, 20, 0.85) \approx \sqrt{\frac{10}{3 \sum_{j=1}^{k} {\beta_j}^2}}, \tag{6.9}$$

and in the x-random case

$$\alpha(80, 40, 0.75) \approx \sqrt{\frac{2}{\sum_{j=1}^{k} {\beta_j}^2}}. \tag{6.10}$$

Given this normalization, subset selection procedures seem to produce models of similar dimensionalities to those given in Breiman's paper. Each β_j is obtained by Breiman's method, p.379.

Experimental results are given in Table 6.8. The performance of pace estimators are as in Experiment 6.5; PACE_6 is the best of all. However, PACE_2 and PACE_4 are not optimal selection criteria in all situations. They are intermediate in performance to other selection criteria—not the best, not the worst. While further investigation is needed to fully understand the results, a plausible explanation is that, as in Experiment 6.5, few data points are available to accurately estimate $G(A^*)$ around τ^*.

6.4 Practical datasets

This section presents simulation studies on practical datasets.

Experiment 6.7 *Practical datasets.* The datasets used are chosen from several that are available. They are chosen for their relatively large number of continuous or binary variables, to highlight the need for subset selection and also in accordance with our k-asymptotic conclusions. To simplify the situation and focus on the basic idea, non-binary nominal variables originally contained in some datasets are discarded, and so are observations with missing values; future extensions of pace regression may take these issues into account.

Table 6.7 lists the names of these datasets, together with the number of observations n and the number of variables k in each dataset. The datasets Autos (Automobile), Cpu (Computer Hardware), and Cleveland (Heart Disease—Processed Cleveland)

	Autos	Bankbill	Bodyfat	Cholesterol	Cleveland	Cpu
n / k	159 / 16	71 / 16	252 / 15	297 / 14	297 / 14	209 / 7
	Horses	Housing	Ozone	Pollution	Prim9	Uscrime
n / k	102 / 14	506 / 14	203 / 10	60 / 16	500 / 9	47 / 16

Table 6.9: Practical datasets used in Experiment 6.7.

are obtained from the Machine Learning Repository at UCI[3] (Blake et al., 1998). Bankbill, Horses and Uscrime are from OzDASL[4] (Australasian Data and Story Library). Bodyfat, Housing (Boston) and Pollution are from StatLib[5]. Ozone is from Leo Breiman's Web site[6]. Prim9 is available from the S-PLUS package. Cholesterol is used by Kilpatrick and Cameron-Jones (1998). Most of these datasets are widely used for testing numeric predictors.

Pace regression procedures, together with others, are applied to these datasets without much scrutiny. This is less than optimal, and often inappropriate. For example, some datasets are intrinsically nonlinear, and/or the noise component is not normally distributed.

Although the results obtained are generally supportive, careful interpretation is necessary. First, the finite situation may not be large enough to support our k-asymptotic conclusions, or the elements in the set $\{\widehat{A}_j\}$ are too separated to support meaningful updating (Section 4.2). In fact, it is well known that no estimator is uniformally optimal in all finite situations. Second, the choice of these datasets may not be extensive enough, or they could be biased in some unknown way. Third, since the true model is unknown and thus cross-validation results are used instead, there is no consensus on the validity of data resampling techniques (Section 2.6).

The experimental results are given in Table 6.7. Each figure is the average of twenty runs of 10-fold cross-validation results, where the "Dim" column gives the average estimated dimensionality (excluding the intercept) and the "PE(%)" column the average squared prediction error relative to the sample variance of the response vari-

[3]http://www.ics.uci.edu/~mlearn/MLRepository.html
[4]http://www.maths.uq.edu.au/~gks/data/index.html
[5]http://lib.stat.cmu.edu/datasets/
[6]ftp://ftp.stat.berkeley.edu/pub/users/breiman

Procedure	Autos		Bankbill		Bodyfat		Cholesterol	
	Dim	PE (%)	Dim	PE (%)	Dim	PE (%)	Dim	PE (%)
NN-GAR.	10.6	21.5	10.39	7.45	8.10	2.76	10.38	101.1
LASSO^{CV}	9.72	21.5	9.71	8.65	2.99	2.73	5.75	97.4
full (OLS)	15.00	21.4	15.00	7.86	14.00	2.75	13.00	97.1
null	0.00	100.5	0.00	101.01	0.00	100.48	0.00	100.4
AIC	7.82	23.6	9.64	8.76	3.38	2.71	4.31	97.4
BIC	3.87	23.9	8.73	8.99	2.24	2.67	2.54	97.6
RIC	3.57	23.6	8.02	9.78	2.24	2.67	2.65	97.0
CIC	5.00	24.5	12.54	8.56	2.23	2.68	2.23	99.9
PACE$_2$	12.95	21.9	13.77	8.20	2.24	2.68	4.42	97.4
PACE$_4$	12.99	21.8	13.80	8.20	2.25	2.68	4.50	97.3
PACE$_6$	10.51	21.1	12.67	8.34	2.32	2.66	4.35	96.2
	Cleveland		Cpu		Horses		Housing	
	Dim	PE (%)	Dim	PE (%)	Dim	PE (%)	Dim	PE (%)
NN-GAR.	9.60	47.6	5.10	25.5	4.90	109	7.50	28.0
LASSO^{CV}	9.10	47.8	5.15	23.4	1.23	97	6.53	28.1
full (OLS)	13.00	48.2	6.00	18.4	13.00	106	13.0	28.3
null	0.00	100.3	0.00	100.7	0.00	102	0.0	100.3
AIC	8.24	50.2	5.06	18.0	3.87	108	11.0	28.0
BIC	5.31	51.3	4.94	17.9	2.50	106	10.7	28.8
RIC	5.41	51.3	5.00	17.7	2.14	108	10.8	28.5
CIC	11.46	49.1	6.00	18.4	0.29	107	11.9	28.2
PACE$_2$	12.35	48.5	5.21	18.2	2.65	107	11.6	28.2
PACE$_4$	12.35	48.6	5.21	18.2	2.68	108	11.6	28.2
PACE$_6$	11.17	49.3	5.14	18.2	2.98	102	11.4	28.2
	Ozone		Pollution		Prim9		Uscrime	
	Dim	PE (%)	Dim	PE (%)	Dim	PE (%)	Dim	PE (%)
NN-GAR.	5.78	32.2	7.69	69.0	6.90	58.3	8.81	60.9
LASSO^{CV}	5.62	32.3	5.32	47.9	6.43	54.1	9.67	49.8
full (OLS)	9.00	31.6	15.00	60.8	7.00	53.5	15.00	52.1
null	0.00	100.6	0.00	102.1	0.00	100.2	0.00	102.9
AIC	5.54	31.6	7.24	58.3	7.00	53.5	6.91	49.3
BIC	3.60	34.3	5.04	65.3	4.13	54.6	5.26	49.3
RIC	4.24	33.5	4.51	66.3	5.87	54.8	3.81	52.7
CIC	5.92	31.7	5.64	67.1	7.00	53.5	5.00	53.4
PACE$_2$	7.17	31.7	8.96	62.7	6.78	53.8	9.16	54.4
PACE$_4$	7.17	31.7	9.06	62.9	6.85	53.8	9.21	54.9
PACE$_6$	6.08	31.5	7.86	56.0	6.97	53.8	7.78	50.7

Table 6.10: Results for practical datasets in Experiment 6.7.

able (i.e., the null model should have roughly 100% relative error).

It can be seen from Table 6.7 that the pace estimators, especially PACE_6, generally produce the best cross-validation results in terms of prediction error. This is consistent with the results on artificial datasets in the previous sections. LASSO^{CV} also generally gives good results on these datasets, in particular Horses and Pollution, while NN-GAROTTE does not seem to perform so well.

6.5 Remarks

In this chapter, experiments have been carried out over many datasets, ranging from artificial to practical. Although these are all finite k situations, pace estimators perform reasonably well, especially PACE_6, in terms of squared error, in accordance with the k-asymptotic conclusions.

We also notice that pace estimators, mainly PACE_2 and PACE_4, perform slightly worse than the best of the other selection estimators in some situations. This is the price that pace regression has to pay for estimating $G(A^*)$ in finite situations. In almost all cases, given the true $G(A^*)$, the experimental results are predictable.

Unlike other modeling procedures used in the experiments, pace estimators may have room for further improvement. This is because an estimator of the mixing distribution that is better than the currently used NNM may be available—e.g., the ML estimator. Even for the NNM estimator itself, the choices of potential support points, fitting intervals, empirical measure, and distance measure can all affect the results. The current choices are not necessarily optimal.

Chapter 7

Summary and final remarks

7.1 Summary

This thesis presents a new approach to fitting linear models, called "pace regression", which also overcomes the dimensionality determination problem introduced in Section 1.1. It endeavors to minimize the expected prediction loss, which leads to the k-asymptotic optimality of pace regression as established theoretically. In orthogonal spaces, it outperforms, in the sense of expected loss, many existing procedures for fitting linear models when the number of free parameters is infinitely large. Dimensionality determination turns out to be a natural by-product of the goal of minimizing expected loss. These theoretical conclusions are supported by simulation studies in Chapter 6.

Pace regression consists of six procedures, denoted by $\text{PACE}_1, \ldots, \text{PACE}_6$, respectively, which are optimal in their corresponding model spaces (Section 3.5). Among them, PACE_5 is the best, but requires numeric integration and thus is computationally expensive. However, it can be approximated very well by PACE_6, which provides consistently satisfactory results in our simulation studies.

The main idea of these procedures, explained in Chapter 3, is to decompose the initially estimated model—we always use the OLS full model—in an orthogonal space into dimensional models, which are statistically independent. These dimensional

models are then employed to obtain an estimate of the distribution of the true effects in each individual dimension. This allows the initially estimated dimensional models to be upgraded by filtering out the estimated expected uncertainty, and hence overcomes the competition phenomena in model selection based on data explanation. The concept of "contribution", which is introduced in Section 3.3, can be used to indicate whether and how much a model outperforms the null one. It is employed throughout the remainder of the thesis to explain the task of model selection and to obtain pace regression procedures.

Chapter 4 discusses some related issues. Several major modeling principles are examined retrospectively in Section 4.5. Orthogonalization selection—a competition phenomenon analogous to model selection—is covered in Section 4.6. The possibility of updating signed projections is discussed in Section 4.7.

Although starting from the viewpoint of competing models, the idea of this work turns out to be similar to the empirical Bayes methodology pioneered by Robbins (1951, 1955, 1964), and has a close relation with Stein or shrinkage estimation (Stein, 1955; James and Stein, 1961). The shrinkage idea was explored in linear regression by Sclove (1968), Breiman (1995) and Tibshirani (1996), for example.

Through this thesis, we have gained a deep understanding of the problem of fitting linear models by minimizing expected quadratic loss. Key issues in fitting linear models are introduced in Chapter 2, and throughout the thesis major existing procedures for fitting linear models are reviewed and compared, both theoretically and empirically, with the proposed ones. They include OLS, AIC, BIC, RIC, CIC, CV(d), BS(m), RIDGE, NN-GAROTTE and LASSO.

Chapter 5 investigates the estimation of an arbitrary mixing distribution. Although this is an independent topic with its own value, it is an essential part of pace regression. Two minimum distance approaches based on the probability measure and on a nonnegative measure are introduced. While all the estimators are strongly consistent, the minimum distance approach based on a nonnegative measure is both computationally efficient and reliable. The maximum likelihood estimator can be

reliable but not efficient; other minimum distance estimators can be efficient but not reliable. By "reliable" here we mean that no isolated data points are ignored in the estimated mixing distribution. This is vital for pace regression, because one single ignored point can cause unbounded loss. The predictive accuracy of these minimum distance procedures in the case of overlapping clusters is studied empirically in Section 5.6, giving results that generally favor the new methods.

Finally, it is worth pointing out that the optimality of pace regression is only in the sense of k-asymptotics, and hence does not exclude the use of OLS, and perhaps many other procedures for fitting linear models, in situations of finite k, especially when k is small or there are few neighboring data points to support a reasonable updating.

7.2 Unsolved problems and possible extensions

In this section, we list some unsolved problems and possible extensions of pace regression—most have appeared earlier in the thesis. Since the issues covered in the thesis are general and have broadly-based potential, we have not attempted to generate an extensive list of extensions.

Orthogonalization selection What is the effect of orthogonalization selection and orthogonalization selection methods on pace regression (Section 4.6)?

Other noise distributions Only normally distributed noise components are considered in the current implementation of pace regression. Extension to other types of distribution, even an empirical distribution, seems fairly straightforward.

Full model unavailable How should the situation when the full model is unavailable and the noise variance is estimated inaccurately (Section 4.4) be handled?

The x-random model It is known that the x-random model converges to the x-fixed model for large samples (Section 2.2). However, for small samples, the x-random assumption should in principle result in a different model, because it involves more uncertainty—namely sampling uncertainty.

Enumerated variables Enumerated variables can be incorporated into linear regression, as well as into pace regression, using dummy variables. However, if the transformation into dummy variables relies on the response vector (so as to reduce dimensionality), it involves competition too. This time the competition occurs among the multiple labels of each enumerated variable. Note that binary variables do not have this problem and thus can be used straightforwardly in pace regression as numeric variables.

Missing values Missing values are another unknowns: they could be filled in so as to minimize the expected loss.

Other loss functions In many applications concerning gains in monetary units, minimizing the expected absolute deviation, rather than the quadratic loss considered here, is more appropriate.

Other model structures The ideas of pace regression are applicable to many model structures, such as those mentioned in Section 1.1—although they may require some adaptation.

Classification Classification differs from regression in that it uses a discrete, often unordered, prediction space, rather than a continuous one. Both share similar methodology: a method that is applicable in one paradigm is often applicable in the other too. We would like to extend the ideas of pace regression to solve classification problems too.

Statistically dependent competing models Future research may extend into the broader arena concerning general issues of statistically dependent competing models (Section 7.3).

7.3 Modeling methodologies

In spite of rigorous mathematical analysis, empirical Bayes (including Stein estimation) is often criticized from a philosophical viewpoint. One seemingly obvious problem, which is still controversial today (see, e.g., discussions following Brown, 1990), is that completely irrelevant (or statistically unrelatedly distributed) events, given in arbitrary measurement units, can be employed to improve estimation of each other. Efron and Morris (1973a, p.379) comment on this phenomenon as follows:

> Do you mean that if I want to estimate tea consumption in Taiwan I will do better to estimate simultaneously the speed of light and the weight of hogs in Montana?

The empirical Bayes methodology contradicts some well-known statistical principles, such as conditionality, sufficiency and invariance.

In a non-technical introduction to Stein estimation, Efron and Morris (1977, pp.125-126), in the context of an example concerning rates of an endemic disease in cities, concluded that "the James-Stein method gives better estimates for a majority of cities, and it reduces the total error of estimation for the sum of all cities. It cannot be demonstrated, however, that Stein's method is superior for any particular city; in fact, the James-Stein prediction can be substantially worse." This clearly indicates that estimation should depend on the loss function employed in the specific application—a notion consistent with statistical decision theory.

Two different loss functions are of interest here. One is the loss concerning a particular player, and the other the overall loss concerning a group of players. Successful application of empirical Bayes implies that it is reasonable to employ overall loss in this application. The further assumption of statistical independence of the players' performance is needed in order to apply the mathematical tools that are available.

The second type of loss concerns the performance of a *team of players*. Although employing information about all players' past performance through empirical Bayes estimation may or may not improve the estimation of a particular player's true ability, it is always (technically, almost surely) the correct approach—in other words, asymptotically optimal using general empirical Bayes estimation and finitely better using Stein estimation—to estimating a team's true ability and hence to predicting its future performance.

A linear model—or any model—with multiple free parameters is such a team, each parameter being a player. Note that parameters could be completely irrelevant, but they are bound together into a team by the prediction task. It may seem slightly beyond the existing empirical Bayes methods, since the performance of these players could be, and often are, statistically dependent. The general question should be how to estimate the team's true ability, given statistically dependent performance of players.

Pace regression handles statistical dependence using *dummy players*, namely the absolute dimensional distances of the estimated model in an orthogonal space. This approach provides a general way of eliminating statistical dependence. There are other possible ways of defining dummy players, and from the empirical Bayes viewpoint it is immaterial how they are defined as long as they are statistically independent of each other. The choice of $A_1, \ldots, A_k$ in pace regression has these advantages: (1) they share the same scale of signal-to-noise ratio, and thus the estimation is invariant of the choice of measurement units; (2) the mixture distribution is well defined, with mathematical solutions available; (3) they are simply related to the loss function of prediction (in the x-fixed situation); (4) many applications have some zero, or almost zero, A_j^*'s, and updating each corresponding A_j often improves estimation and reduces dimensionality (see Section 4.2).

Nevertheless, using dummy players in this way is not a perfect solution. When the determination of orthogonal basis relies on the observed response, it suffers from the orthogonalization selection problem (Section 4.6). Information about the statistical dependence of the original players might be available, or simply estimable from

data. If so, it might be employable to improve the estimate of the team's true ability.

Estimation of a particular player's true ability using the loss function concerning this single player is actually estimation given the player's *name*, which is different from estimation given the player's *performance* while the performance of all players in a team is mixed together with their names lost or ignored. This is because the latter addresses the team ability given that performance, thus sharing the same methodology of estimating team ability. For example, pre-selecting parameters based on performance (i.e., data explanation) affects the estimation of the team ability; see Section 4.4.

This point generalizes to situations with many competing models. When the final model is determined based on its performance—rather than on some prior belief—after seeing all the models' performance, the estimation of its true ability should take into account all candidates' performance. In fact, the final model, assumed optimal, does not have to be one of the candidate models—it can be a modification of one model or a weighted combination of all candidates.

Modern computing technology allows the investigation of larger and larger data sets and the exploration of increasingly wide model spaces. *Data mining* (see, e.g., Fayyad et al., 1996; Friedman, 1997; Witten and Frank, 1999), as a burgeoning new technology, often produces many candidate models from computationally intensive algorithms. While this can help to find more appropriate models, the chance effects increase. How to determine the optimal model given a number of competing ones is precisely our concern above.

The subjective prior used in Bayesian analysis concerns the specification of the model space in terms of probabilistic weights, thus relating to Fisher's first problem (see Section 1.1). Without a prior, the model space would otherwise be specified implicitly with (for example) a uniform distribution. Both empirical Bayes and pace regression fully exploit data after all prior information—including specification of the model space—is given. Thus both belong to Fisher's second problem type. The story told by empirical Bayes and pace regression is that the data contains *all*

necessary information for the convergence of optimal predictive modeling, while the tips given by any prior information—the correctness of which is always subject to practical examination—play at most a role of speeding up this convergence.

"Torturing" data is deemed to be unacceptable modeling practice (see Section 1.3). In fact, however, empirical modeling always tortures the data to some extent—the larger the model space, the more the data is tortured. Since some model space, large or small, is indispensable, the correct approach is to take the effect of the necessary data torturing into account within the modeling procedure. This is what pace regression does. And does successfully.

Appendix A

Help files and source code

The ideas presented in the thesis are formally implemented in the computer languages of S-PLUS/R and FORTRAN 77. The whole program runs under versions 3.4 and 5.1 of S-PLUS (MathSoft, Inc.[1]), and under version 1.1.1 of R.[2] Help files for most important S-PLUS functions are included. The program is free software; it can be redistributed and/or modified under the terms of the GNU General Public License (version 2) as published by the Free Software Foundation.[3]

One extra software package is utilized in the implementation, namely the elegant NNLS algorithm by Lawson and Hanson (1974, 1995), which is public domain and obtainable from NetLib.[4]

The source code is organized in six files (five in S-PLUS/R and one in FORTRAN 77):

`disctfun.q` contains functions that handle discrete functions. They are mainly employed in our work to represent discrete pdfs; in particular, the discrete, arbitrary mixing distributions.

`ls.q` is basically an S-PLUS/R interface to functions in FORTRAN for solving linear system problems, including QR transformation, non-negative linear

[1] http://www.mathsoft.com/

[2] http://lib.stat.cmu.edu/R/CRAN/

[3] Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA. http://www.gnu.org/.

[4] http://www.netlib.org/lawson-hanson/all

regression, non-negative linear regression with equality constraints, upper-triangular equation solution.

mixing.q includes functions for the estimation of an arbitrary mixing distributions.

pace.q has functions about pace regression and for handling objects of class "pace".

util.q gives a few small functions.

ls.f has functions in FORTRAN for solving some linear systems problems.

A.1 Help files

bmixing:

See **mixing**.

disctfun:

```
Constructor of a Discrete Function


DESCRIPTION:
       Returns a univariate discrete  function  which  is  always
       zero except over a discrete set of data points.

USAGE:
       disctfun(data=NULL, values=NULL, classname, normalize=F,
                sorting=F)

REQUIRED ARGUMENTS:
       None.

OPTIONAL ARGUMENTS:
data:     The set of data points over which  the  function  takes
       non-zero values.
values:   The set of the function values over the specified  data
       points.
classname:     Extra class names; e.g., c("sorted", "normed"), if
       data are sorted and values are L1-normalized on input.
normalize:     Will L1-normalize values (so as to make a discrete
       pdf function).
sorting:    Will sort data.

VALUE:
       Returns an object of class "disctfun".

DETAILS:
       Each discrete function is stored in  a  (n  x  2)  matrix,
       where  the  first column stores the data points over which
       the function takes non-zero values, and the second  column
       stores the corresponding function values at these points.

SEE ALSO:
       disctfun.object, mixing.

EXAMPLES:
       disctfun()
       disctfun(c(1:5,10:6), 10:1, T, T)
```

disctfun.object:

Discrete Function Object

DESCRIPTION:

These are objects of class "disctfun". They represent the discrete functions which take non-zero values over a discrete set of data points .

GENERATION:

This class of objects can be generated by the contructor disctfun.

METHODS:

The "disctfun" class of objects has methods for these generic functions: sort, print, is, normalize, unique, plot, "+", "*", etc.

STRUCTURE:

Each discrete function is stored in a (n x 2) matrix, where the first column stores the data points over which the function takes non-zero values, and the second column stores the corresponding function values at these points.

Since it is a special matrix, methods for matrices can be used.

SEE ALSO:

disctfun, mixing.

fintervals:

Fitting Intervals

DESCRIPTION:

Returns a set of fitting intervals given the set of fitting points.

USAGE:

```
fintervals(fp, rb=0, cn=3, dist=c("chisq", "norm"),
     method=c("nnm", "pm", "choi", "cramer"), minb=0.5)
```

REQUIRED ARGUMENTS:

fp: fitting points. Allow being unsorted.

OPTIONAL ARGUMENTS:

rb: Only used for Chi-sqaured distribution. No fitting intervals' endpoints are allowed to fall into the interval (0,rb). If the left endpoint does, it is replaced by 0; if the right endpoint does, replaced by rb. This is useful for the function mixing() when ne != 0.

```
cn:     Only for normal distribution. The length of  each  fitting
        interval is 2*cn.
dist:      type of distribution; only implemented for  non-central
        Chi-sqaure ("chisq") with one degree of freedom and normal
        distribution with variance 1 ("norm").
method:    One of "nnm", "pm", "choi" and  "cramer"; see mixing.
minb:      Only for Chi-squared distribution and method "nnm".  No
        right  endpoint  is allowed between (0, minb); delete such
        intervals if generated.

VALUE:
        fitting intervals (stored in a two-coloumn matrix).

SEE ALSO:
        mixing, spoints.

EXAMPLES:
        fintervals(1:10)
        fintervals(1:10, rb=2)
```

lsqr:

```
QR Transformation of the Least Squares Problem


DESCRIPTION:
        QR-transform the least squares problem
                        A x = b
         into
                        R x = t(Q) b
         where
              A = Q R, and
              R is  an  upper-triagular  matrix  (could  be  rank-
        deficient)
              Q is an orthogonal matrix.

USAGE:
        lsqr(a,b,pvt,ks=0,tol=1e-6,type=1)

REQUIRED ARGUMENTS:
a:      matrix A
b:      vector b

OPTIONAL ARGUMENTS:
pvt:    column pivoting vector, only columns specified in pvt  are
        used  in  QR-transformed. If not provided, all columns are
        considered.
ks:     the first ks columns specified in pvt  are  QR-transformed
        in  the  same order.  (But they may be deleted due to rank
        deficiency.)
tol:    tolerance for collinearity
type:      used for finding the most suitable pivoting column.
               1    based on explanation ratio.
               2    based on largest diagonal element.
```

VALUE:

a: R (including the zeroed elements)

b: t(Q) b

pvt: pivoting index

ks: pseudorank of R (determined by the value of tol)

w: explanation matrix.

w[1,] sum of squares in a[,j] and b[,j]

w[2,] sum of unexplained squares

DETAILS:

This is an S-Plus/R interface to the function LSQR in FORTRAN. Refer to the source code of this function for more detail.

REFERENCES:

Lawson, C. L. and Hanson, Richard J. (1974, 1995). "Solving Least Squares Problems", Prentice-Hall.

Dongarra, J. J., Bunch, J.R., Moler, C.B. and Stewart, G.W. (1979). "LINPACK Users' Guide." Philadelphia, PA: SIAM Publications.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D. (1999). "LAPACK Users' Guide." Third edition. Philadelphia, PA: SIAM Publications.

EXAMPLES:

```
a <- matrix(rnorm(500),ncol=10)
b <- 1:50 / 50
lsqr(a,b)
```

mixing, bmixing:

Minimum Distance Estimator of an Arbitrary Mixing Distribution.

DESCRIPTION:

The minimum distance estimation of an arbitrary mixing distribution based on a sample from the mixture distribution, given the component distribution.

USAGE:

```
mixing(data, ne=0, t=3, dist=c("chisq", "norm"),
       method=c("nnm", "pm", "choi", "cramer"))
bmixing(data, ne = 0, t = 3, dist = c("chisq", "norm"),
        method = c("nnm", "pm", "choi", "cramer"),
        minb = 0.5)
```

REQUIRED ARGUMENTS:

data: A sample from the mixture distribution.

OPTIONAL ARGUMENTS:

ne: number of extra points with small values; only used for Chi-square distribution.

t: threshold for support points; only used for Chi-square distribution.

dist: type of distribution; only implemented for non-central Chi-square ("chisq") with one degree of freedom and normal distribution with variance 1 ("norm").

method: one of the four methods:

- "nnm": based on nonnegative measures (Wang, 2000)
- "pm": based on probability measures (Wang, 2000)
- "choi": based on CDFs (Choi and Bulren, 1968)
- "cramer": Cramer-von Mises statistic, also CDF-based (MacDonald, 1971).

The default is "nnm", which, unlike the other three, does not have the minority cluster problem.

minb: Only for Chi-squared distribution and method "nnm". No right endpoint is allowed between (0, minb); delete such intervals if any.

VALUE:

the estimated mixing distribution as an object of class "disctfun".

DETAILS:

The seperability of data points are tested by the function mixing() using the function separable(). Each block is handled by the function bmixing().

REFERENCES:

Choi, K., and Bulgren, W. B. (1968). An estimation procedure for mixtures of distributions. J. R. Statist. Soc. B, 30, 444-460.

MacDonald, P. D. M. (1971). Comment on a paper by Choi and Bulgren. J. R. Statist. Soc. B, 33, 326-329.

Wang, Y. (2000). A new approach to fitting linear models in high dimensional spaces. PhD thesis, Department of Computer Science, University of Waikato, New Zealand.

SEE ALSO:

pace, disctfun, separable, spoints, fintervals.

EXAMPLES:

```
mixing(1:10)                  # mixing distribution of nchiq
bmixing(1:10)
mixing(c(5:15, 50:60))          # two main blocks
bmixing(c(5:15, 50:60))
mixing(c(5:15, 50:60), ne=20) # ne affects the estimation
bmixing(c(5:15, 50:60), ne=20)
```

nnls, nnlse:

Problems NNLS and NNLSE

DESCRIPTION:

Problem NNLS (nonnegative least squares):

```
Minimize        || A x - b ||
subject to      x >= 0
```

Problem NNLSE (nonnegative least squares with equality constraints):

```
Minimize     || A x - b ||
subject to   E x = f
and          x >= 0
```

USAGE:

```
nnls(a,b)
nnlse(a, b, e, f)
```

REQUIRED ARGUMENTS:

a: matrix A
b: vector b
e: matrix E
f: vector f

VALUE:

x: the solution vector
rnorm: the Euclidean norm of the residual vector.
index: defines the sets P and Z as follows:
P: index[1:nsetp] INDEX(1)
Z: index[(nsetp+1):n]
mode: the success-failure flag with the following meaning:
1 The solution has been computed successfully.
2 The dimensions of the problem is bad,
either m <= 0 or n <= 0
3 Iteration count exceeded.
More than 3*n itereations.

DETAILS:

This is an S-Plus/R interface to the algorithm NNLS proposed by Lawson and Hanson (1974, 1995), and to the algorithm NNLSE by Haskell and Hanson (1981) and Hanson and Haskell (1982).

Problem NNLSE converges to Problem NNLS by considering

```
Minimize  || /  A\     /  b\ ||
          || |   | x - |   | ||
          || \e E/     \e f/ ||
subject to x >= 0
as e --> 0+.
```

Here the implemented nnlse only considers the situation that xj >= 0 for all j, while the original algorithm allows restriction on some xj only.

The two functions nnls and nnlse are tested based on the source code of NNLS (in FORTRAN 77), which is public domain and obtained from http://www.netlib.org/lawson-hanson/all.

REFERENCES:

Lawson, C. L. and Hanson, R. J. (1974, 1995). "Solving Least Squares Problems", Prentice-Hall.

Haskell, K. H. and Hanson, R. J. (1981). An algorithm for linear least squares problems with equality and nonnegativity constraints. Math. Prog. 21 (1981), pp. 98-118.

Hanson, R. J. and Haskell, K. H. (1982). Algorithm 587. Two algorithms for the linearly constrained least squares problem. ACM Trans. on Math. Software, Sept. 1982.

EXAMPLES:

```
a <- matrix(rnorm(500),ncol=10)
b <- 1:50 / 50
nnls(a,b)
e <- rep(1,10)
f <- 1
nnlse(a,b,e,f)
```

nnlse:

See **nnls**.

pace:

Pace Estimator of a Linear Regression Model

DESCRIPTION:

Returns an object of class "pace" that represents a fit of a linear model.

USAGE:

```
pace(x, y, va, yname=NULL, ycol=0, pvt=NULL, kp=0,
     intercept=T, ne=0, method=<<see below>>, tau=2,
     tol=1e-6)
```

REQUIRED ARGUMENTS:

x: regression matrix. It is a matrix, or a data.frame, or anything else that can be coerced into a matrix through function as.matrix().

OPTIONAL ARGUMENTS:

y: response vector. If y is missing, the last column in x or the column specified by ycol is the response vector.

va: variance of noise component.

yname: name of response vector.

ycol: column number in x for response vector, if y not provided.

pvt: pivoting vector which stores the column numbers that will be considered in the estimation. When NULL (default), all columns are taken into account.

kp: the ordering of the first kp (=0, by default) columns specified in pvt are pre-chosen. The estimation will not change their ordering, but they are suject to rank-deficiency examination.

intercept: if intercept should be added into the regression matrix.

ne: number of extra variables which are pre-excluded before calling this function due to small effects (i.e., A's). Only used for pace methods and cic.

method: one of the methods "pace6" (default), "pace2", "pace4", "olsc", "ols", "full", "null", "aic", "bic", "ric", "cic". See DETAILS and REFERENCES below.

tau: threshold value for the method "olsc(tau)".

tol: tolerance threshold for collinearity. = 1e-6, by default.

VALUE:

An object of class "pace" is returned. See pace.object for details.

DETAILS:

For methods pace2, pace4, pace6 and olsc, refer to Wang (2000) and Wang et al. (2000). Further, n-aymptotically, aic = olsc(2), bic = olsc(log(n)), and ric = olsc(2 log(k)), where n is the number of observations and k is the number of free parameters (or candidate variables).

Method aic is prospoed by Akaike (1973).

Method bic is by Schwarz (1978).

For method ric, refer to Donoho and Johnstone (1994) and Foster and George (1994)

Method cic is due to Tibshirani and Knight (1997).

Method ols is the ordinary least squares estimation including all parameters after eliminating (psuedo-) collinearity, while method full is the ols without eliminating collinearity. Method null returns the mean of the response, included for reason of completeness.

REFERENCES:

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. Proc. 2nd Int. Symp. Inform. Theory, suppl. Problems of Control and Information theory, pp267-281.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal Spatial Adaptation via Wavelet Shrinkage. Biometrika 81, 425-55.

Foster, D. and George, E. (1994). The risk inflation criterion for multiple regression. Annals of Statistics, 22, 1947-1975.

Schwarz, G. (1978). Estimating the dimension of a model. Annals of Statistics. Vol. 6, No. 2, March 1978. 461-464

Tibshirani, R. and Knight, K. (1997). The covariance inflation criterion for model selection . Technical report, November, 1997, Department of Statistics, University of Stanford.

Wang, Y. (2000). A new approach to fitting linear models in high dimensional spaces. PhD thesis, Department of Computer Science, University of Waikato, New Zealand.

Wang, Y., Witten, I. H. and Scott, A. (2000). Pace regression. (Submitted.)

SEE ALSO:

pace.object, predict.pace

EXAMPLES:

```
x <- matrix(rnorm(500), ncol=10)    # completely random
pace(x)   # equivalently, pace(x,method="pace6")
for(i in c("ols","null","aic","bic","ric","cic","pace2",
    "pace4","pace6")) {cat("---- METHOD", i,"--- \n\n");
     print(pace(x, method=i))}
y <- x %*% c(rep(0,5), rep(.5,5)) + rnorm(50)
                              # mixed effect response
for(i in c("ols","null","aic","bic","ric","cic","pace2",
    "pace4","pace6")) {cat("---- METHOD", i,"--- \n\n");
     print(pace(x, y, method=i))}
pace(x,y,method="olsc",tau=5)
pace(x,y, ne=20)    # influence of extra variables
```

pace.object:

Pace Regression Model Object

DESCRIPTION:

These are objects of class "pace". They represent the fit of a linear regression model by pace estimator.

GENERATION:

This class of objects is returned from the function pace to represent a fitted linear model.

METHODS:

The "pace" class of objects (at the moment) has methods for these generic functions: print, predict.

STRUCTURE:
coef: fitted coefficients of the linear model.
pvt: pivoting columns in the training set.
ycol: the column in the training set considered as the response vector.
intercept: intercept is used or not.
va: given or estimated noise variance (i.e., the unbiased OLS estimate).
ks: the final model dimensionality.
kc: psuedo-rank of the design matrix (determined due to the value of tol).
ndims: total number of dimensions.
nobs: number of observations.
ne: nubmer of extra variables with small effects.
call: function call.
A: observed absolute distances of the OLS full model.
At: updated absolute distances.
mixing: estimated mixing distribution from observed absolute distances.

SEE ALSO:
pace, predict.pace, mixing

predict.pace:

Predicts New Obsevations Using a Pace Linear Model Estimate.

DESCRIPTION:
Returns the predicted values for the response variable.

USAGE:
```
predict.pace(p, x, y)
```

REQUIRED ARGUMENTS:
p: an object of class "pace".
x: test set (may or may not contain the response vector.)

OPTIONAL ARGUMENTS:
y: response vector (if not stored in x).

VALUE:
pred: predicted response vector.
resid: residual vector (if applicable).

SEE ALSO:
pace, pace.object.

EXAMPLES:
```
x <- matrix(rnorm(500), ncol=10)
p <- pace(x)
```

```
xtest <- matrix(rnorm(500), ncol=10)
predict(p, xtest)
```

rsolve:

Solving Upper-Triangular Equation

DESCRIPTION:

Solve the upper-triangular equation

$$X * b = y$$

where X is an upper-triagular matrix (could be rank-deficient). Here the columns of X may be pre-indexed.

USAGE:

```
rsolve(x,y,pvt,ks)
```

REQUIRED ARGUMENTS:

x: matrix X

y: vector y

OPTIONAL ARGUMENTS:

pvt: column pivoting vector, only columns specified in pvt are used in solving the equation. If not provided, all columns are considered.

ks: the first ks columns specified in pvt are QR-transformed in the same order.

VALUE:

the solution vector b

DETAILS:

This is an S-Plus/R interface to the function RSOLVE in FORTRAN. Refer to the source code of this function for more detail.

REFERENCES:

Dongarra, J. J., Bunch, J.R., Moler, C.B. and Stewart, G.W. (1979). "LINPACK Users' Guide." Philadelphia, PA: SIAM Publications.

Press, W.H., Flanney, B.P., Teukkolky S.A., Vatterling, U.T. (1994). "Numerical Recipes in C." Cambridge University Press.

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D. (1999). "LAPACK Users' Guide." Third edition. Philadelphia, PA: SIAM Publications.

EXAMPLES:

```
a <- matrix(rnorm(500),ncol=10)
```

```
b <- 1:50 / 50
fit <- lsqr(a,b)
rsolve(fit$a,fit$b,fit$pvt)
```

separable:

Seperability of a Point from a Set of Points.

DESCRIPTION:

Test if a point could be separable from a set of data points, based on probability, for the purpose of estimating mixing distribution.

USAGE:

```
separable(data, x, ne=0, thresh=0.1,
          dist=c("chisq", "norm"))
```

REQUIRED ARGUMENTS:

data: incrementally sorted data points.

x: a single point, satisfying that either x <= min(data) or x >= max(data).

OPTIONAL ARGUMENTS:

ne: number of extra data points of small values (= 0, by default). Only used from chisq distribution.

thresh: a threshold value based on probability (= 0.1, by default). Larger value implies easier seperability.

dist: type of distribution; only implemented for non-central Chi-square ("chisq") with one degree of freedom and normal distribution with variance 1 ("norm").

VALUE:

T or F, implying whether x is separable from data.

DETAILS:

Used by the estimation of a mixing distribution so that the whole set of data points could be seperated into blocks to speed up estimation.

REFERENCES:

Wang, Y. (2000). A new approach to fitting linear models in high dimensional spaces. PhD thesis, Department of Computer Science, University of Waikato, New Zealand.

SEE ALSO:

mixing, pace

EXAMPLES:

```
separable(5:10, 25)          # Returns T
separable(5:10, 25, ne=20)   # Returns F, for extra small
                             # values not included in data.
```

spoints:

Support Points

DESCRIPTION:

Returns the set of candidate support points for the minimum distance estimation of an arbitrary mixing distribution.

USAGE:

```
spoints(data, ne=0, dist=c("chisq", "norm"), ns=100, t=3)
```

REQUIRED ARGUMENTS:

data: A sample of the mixture distribution.

OPTIONAL ARGUMENTS:

ne: number of extra points with small values. Only used for Chi-sqaured distribution.

dist: type of distribution; only implemented for non-central Chi-square ("chisq") with one degree of freedom and normal distribution with variance 1 ("norm").

ns: Maximum number of support points. Only for normal distribution.

t: No support points inside (0, t).

VALUE:

Support points (stored as a vector).

SEE ALSO:

mixing, fintervals.

EXAMPLES:

```
spoints(1:10)
```

A.2 Source code

disctfun.q:

```
# ------------------------------------------------------------
#
# Functions for manipulating discrete functions
#
# ------------------------------------------------------------

disctfun <- function(data=NULL, values=NULL, classname,
                     normalize = F, sorting = F)
{
  if( missing(data) ) n <- 0
  else n <- length(data)
  if( n == 0) {         # empty disctfun
    if(is.null( version$language ) )  {     # S language
      d <- NULL
      class(d) <- "disctfun"
      return (d)
    }
    else {                                  # R language
      d <- matrix(nrow=0,ncol=2,dimnames =
                  list(NULL,c("Points", "Values")))
      class(d) <- "disctfun"
      return (d)
    }
  }
  else {
    if( missing(values) ) values <- rep(1/n,n)
    else if (length(values) != n)
      stop("Lengths of data and values must match.")
    else if( normalize ) values <- values / sum(values)
    d <- array(c(data, values),dim=c(n,2),dimnames =
               list(paste("[",1:n,",]",sep=""),
                    c("Points", "Values")))
    class(d) <- "disctfun"
  }
  if( ! missing(classname) )
    class(d) <- unique( c(class(d), classname) )
  if( missing(values) || normalize )
    class(d) <- unique( c(class(d),"normed") )
  if( is.sorted(data) )
    class(d) <- unique( c(class(d), "sorted") )
  if( sorting && !is.sorted(d) ) sort.disctfun(d)
  else d
}

sort.disctfun <- function(d)
{
  if (length(d) == 0) return (d)
  if( is.sorted(d) ) return (d)
  n <- dim(d)[1]
  if( n != 1) {
```

```
    i <- sort.list(d[,1])
    cl <- class(d)
    d <- array(d[i,], dim=c(n,2),dimnames =
              list(paste("[",1:n,",]",sep=""),
                   c("Points", "Values")))
    class(d) <- cl
  }
  class(d) <- unique( c(class(d),"sorted") )
  d
}

is.disctfun <- function(d)
{
  if( is.na( match("disctfun", class(d) ) ) ) F
  else T
}

normalize.disctfun <- function(d)
{
  d[,2] <- d[,2] / sum(d[,2])
  class(d) <- unique( c(class(d),"normed") )
  d
}

print.disctfun <- function(d)
{
  if( length(d) == 0 ) cat("disctfun()\n")
  else if(dim(d)[1] == 0) cat("disctfun()\n")
  else print(unclass(d))
  invisible(d)
}

unique.disctfun <- function(d)
# On input, d is sorted.
{
  count <- 1
  if(dim(d)[1] >= 2) {
    for ( i in 2:dim(d)[1] )
      if( d[count,1] != d[i,1] ) {
        count <- count + 1
        d[count,] <- d[i,]
      }
      else d[count,2] <- d[count,2] + d[i,2]
    disctfun(d[1:count,1], d[1:count,2],"sorted")
  }
  else {
    class(d) <- c(class(d),"sorted")
    d
  }
}

"+.disctfun" <- function(d1,d2)
{
  if(! is.disctfun(d1) | ! is.disctfun(d2) )
    stop("the argument(s) not disctfun.")
```

```
  if( length(d1) == 0) d2
  else if (length(d2) == 0) d1
  else {
    d <- disctfun( c(d1[,1],d2[,1]), c(d1[,2],d2[,2]), sort = T )
    # In R, unique is not a generic function
    unique.disctfun(d)
  }
}

"*.disctfun" <- function(x,d)
{
  if( is.disctfun(d) ) {
    if( length(x) == 1 )
      { if( length(d) != 0 ) d[,2] <- x * d[,2] }
    else stop("Inappropriate type of argument.")
    # this seems necessary for R
    class(d) <- unique(c("disctfun", class(d)))
    d
  }
  else if( is.disctfun(x) ) {
    if( length(d) == 1 )
      { if( length(x) != 0 ) x[,2] <- d * x[,2] }
    else stop("Inappropriate type of argument.")
    # this seems necessary for R
    class(x) <- unique(c("disctfun", class(x)))
    x
  }
}

plot.disctfun <- function(d,xlab="x", ylab="y",lty=1,...)
{
  if( length(d) != 0)
    plot(d[,1],d[,2],xlab=xlab, ylab=ylab,type="h",...)
  else cat("NULL (disctfun); nothing ploted.\n")
}
```

ls.q:

```
# ------------------------------------------------------------
#
# Functions for solving linear system problems.
#
# ------------------------------------------------------------

nnls <- function(a,b)
{
  m <- as.integer(dim(a)[1])
  n <- as.integer(dim(a)[2])
  storage.mode(a) <- "double"
  storage.mode(b) <- "double"
  x <- as.double(rep(0, n))          # only for output
  rnorm <- 0
  storage.mode(rnorm) <- "double"  # only for output
  w <- x                             # n-vector of working space
```

```
  zz <- b                              # m-vector of working space
  index <- as.integer(rep(0,n))
  mode <- as.integer(0)                # = 1, success
  .Fortran("nnls",a,m,m,n,b,x=x,rnorm=rnorm,w,zz,index=index,
           mode=mode)[c("x","rnorm","index","mode")]
}

nnls.disctfun <- function(s, sp)
{
  index <- s$index[s$x[s$index]!=0]
  disctfun(sp[index],s$x[index], normalize=T, sorting=T)
}

nnlse <- function(a, b, e, f)
{
  eps <- 1e-3;              # eps is adjustable.

  nnls(rbind(e, a * eps), c(f, b * eps))
}

lsqr <- function(a,b,pvt,ks=0,tol=1e-6,type=1)
{
  if ( ! is.matrix(b) ) dim(b) <- c(length(b),1)
  if ( ! is.matrix(a) ) dim(a) <- c(1,length(a))
  m <- as.integer(dim(a)[1])
  n <- as.integer(dim(a)[2])
  if ( m !=  dim(b)[1] )
    stop("dim(a)[1] != dim(b)[1] in lsqr()")
  nb <- as.integer(dim(b)[2])
  if(missing(pvt) || is.null(pvt)) pvt <- 1:n
  else if( ! correct.pvt(pvt,n) ) stop("mis-specified pvt.")
  kp <- length(pvt)
  kn <- 0
  storage.mode(kn) <- "integer"
  storage.mode(kp) <- "integer"
  storage.mode(a) <- "double"
  storage.mode(b) <- "double"
  storage.mode(pvt) <- "integer"
  storage.mode(ks) <- "integer"
  storage.mode(tol) <- "double"
  storage.mode(type) <- "integer"
  if(ks < 0 || ks > kp) stop("ks out of range in lsqr()")
  w <- matrix(0, nrow = 2, ncol = n+nb) # working space
  storage.mode(w) <- "double"
  .Fortran("lsqr", a=a,m,m,n,b=b,m,nb,pvt=pvt,kp,ks=ks,w=w,tol,
           type)[c("a","b","pvt","ks","w")]
}

correct.pvt <- function(pvt,n) {
  ! ( max(pvt) > n ||                  # out of range
     min(pvt) < 1 ||                   # out of range
     any(duplicated(pvt))              # has duplicated
     )
}

rsolve <- function(x,y,pvt,ks)
```

```
{
  if ( ! is.matrix(y) ) dim(y) <- c(length(y),1)
  if ( ! is.matrix(x) ) dim(x) <- c(1,length(x))
  m <- as.integer(dim(x)[1])
  n <- as.integer(dim(x)[2])
  mn <- min(m,n)
  ny <- as.integer(dim(y)[2])
  if(missing(pvt)) {
    pvt <- 1:mn
  }
  if ( missing(ks) ) ks <- length(pvt)
  if(ks < 0 || ks > mn) stop("ks out of range in lsqr()")
  storage.mode(x) <- "double"
  storage.mode(y) <- "double"
  storage.mode(pvt) <- "integer"
  storage.mode(ks) <- "integer"
  .Fortran("rsolve", x,m,y=y,m,ny,pvt,ks)$y[1:ks,]
}
```

mixing.q:

```
# ------------------------------------------------------------
#
# Functions for the estimation of a mixing distribution
#
# ------------------------------------------------------------

spoints <- function(data, ne =0, dist = c("chisq", "norm"),
                    ns = 100, t = 3)
{
  if( ! is.sorted(data) ) data <- sort(data)

  dist <- match.arg(dist)
  n <- length(data)
  if(n < 2) stop("tiny size data.")

  if(dist == "chisq" && (data[1] < t || ne != 0) ) {
    sp <- 0
    ns <- ns -1
    if(data[n] == t) ns <- 1
  }
  else sp <- NULL
  switch (dist,
          "norm" = data, #seq(data[1], data[n], length=ns),
          c(sp, data[data >= t]) )
}

fintervals <- function( fp, rb = 0, cn = 3,
                        dist = c("chisq", "norm"),
                        method = c("nnm", "pm", "choi", "cramer"),
                        minb = 0.5)
{
  dist <- match.arg(dist)
  method <- match.arg(method)
```

```
  switch( method,
         "pm" = ,
         "nnm" = switch( dist,
           "chisq" = {f <- cbind(c(fp,invlfun(fp[(i<-fp>=minb)])),
                                 c(lfun(fp),fp[i]) )
                      if( rb >= 0 ) {
                        f[f[,1] > 0 & f[,1] < rb, 1] <- 0
                        f[f[,2] > 0 & f[,2] < rb, 2] <- rb
                      }
                      f
                    },
           "norm" = cbind( c(fp,fp-cn), c(fp+cn,fp) )),
         "choi" = ,
         "cramer" = switch( dist,
           "chisq" = cbind(pmax(min(fp) - 50, 0), fp),
           "norm" = cbind(min(fp) - 50, fp) )
         )
}

lfun <- function( x, c1 = 5, c2 = 20 )
# ------------------------------------------------------------
# function l(x)
#
# Returns the right boudary points of fitting intervals
# given the left ones
# ------------------------------------------------------------
{
  if(length(x) == 0) NULL
  else {
    x[x<0] <- 0
    c1 + x + c2 * sqrt(x)
  }
}

invlfun <- function( x, c1 = 5, c2 = 20 )
# ------------------------------------------------------------
# inverse l-function
# ------------------------------------------------------------
{
  if(length(x) == 0) NULL
  else {
    x[x<=c1] <- c1
    (sqrt( x - c1 + c2 * c2 / 4 ) - c2 / 2 )^2
  }
}

epm <- function(data, intervals, ne = 0, rb = 0, bw = 0.5)
# ------------------------------------------------------------
# empirical probability measure over intervals
# ------------------------------------------------------------
{
  n <- length(data)
  pm <- rep(0, nrow(intervals))
  for ( i in 1:n )
    pm <- pm + (data[i]>intervals[,1] & data[i]<intervals[,2]) +
      ( (data[i] == intervals[,1]) + (data[i] ==
```

```
                intervals[,2])) * bw
  i <- intervals[,1] < rb & intervals[,2] >= rb
  pm[i] <- pm[i] + ne
  pm / (n + ne)
}

probmatrix <- function( data, bins, dist = c("chisq", "norm") )
# ------------------------------------------------------------
# Probability matrix
# ------------------------------------------------------------
{
  dist <- match.arg(dist)
  n<-length(data)
  e <- matrix(nrow=nrow(bins),ncol=n)
  for (j in 1:n)
    e[,j] <- switch(dist,
                    "chisq" = pchisq1(bins[,2], data[j]) -
                    pchisq1(bins[,1], data[j]),
                    "norm" = pnorm(bins[,2], data[j]) -
                    pnorm(bins[,1], data[j]) )
  e
}

bmixing <- function(data, ne = 0, t = 3, dist = c("chisq","norm"),
                    method = c("nnm","pm","choi","cramer"),
                    minb=0.5)
{
  dist <- match.arg(dist)
  method <- match.arg(method)

  if( !is.sorted(data) ) {
    data <- sort(data)
    class(data) <- unique( c(class(data),"sorted") )
  }

  n <- length(data)

  if( n == 0 ) return (disctfun())
  if( n == 1 ) {
    if( dist == "chisq" && data < t ) return ( disctfun(0) )
    else  return ( disctfun(max( data )) )
  }

  # support points
  sp <- spoints(data, ne = ne, t = t, dist = dist)
  rb <- 0
  # fitting intervals
  if(ne != 0) {
    rb <- max(minb,data[min(3,n)])
    fi <- fintervals(data, rb = rb, dist = dist, method = method,
                     minb=minb)
  }
  else fi <- fintervals(data, dist = dist, method = method,
                        minb=minb)

  bw <- switch(method,
```

```
                "cramer" =,
                "nnm" =,
                "pm" = 0.5,
                "choi" = 1 )
  # empirical probability measure
  b <- epm(data, fi, ne=ne, rb=rb, bw=bw)
  a <- probmatrix(sp, fi, dist = dist)
  ns <- length(sp)
  nnls.disctfun(switch( method,
                        "nnm" = nnls(a,b),
                         nnlse(a,b,matrix(1,nrow=1,ncol=ns),1))
                , sp)
}

mixing <- function(data, ne = 0, t = 3, dist = c("chisq","norm"),
                   method = c("nnm","pm","choi","cramer"))
{
  dist <- match.arg(dist)
  method <- match.arg(method)

  n <- length(data)
  if( n <= 1 )
    return ( bmixing(data,ne,t=t,dist=dist,method=method) )

  if( ! is.sorted(data) ) data <- sort(data)
  d <- disctfun()
  start <- 1
  for( i in 1:(n-1) ) {
    if( separable(data[start:i], data[i+1],ne, dist=dist) &&
       separable(data[(i+1):n], data[i], dist=dist)  ) {
      x <- data[start:i]
      class(x) <- "sorted"
      d <- d + ( i - start + 1 + ne) *
        bmixing(x,ne,t=t,dist=dist,method=method)
      start <- i+1
      ne <- 0
    }
  }
  x <- data[start:n]
  class(x) <- "sorted"
  normalize( d + ( n - start + 1 + ne) *
            bmixing(x,ne,t=t,dist=dist,method=method) )
}

separable <- function(data, x, ne = 0, thresh = 0.10,
                      dist = c("chisq","norm") )
{
  if (length(x) != 1) stop("length(x) must be one.")

  dist <- match.arg(dist)
  n <- length(data)
  mi <- data[1]

  if (dist == "norm")
    if( sum( 1 - pnorm( abs( x - data) ) ) <= thresh ) T
    else F
```

```
  else {
    rx <- sqrt(x)
    p1 <- pnorm( abs(rx - sqrt(data)) )
    p2 <- pnorm( abs(rx - seq(0,sqrt(mi),len=3)) )
    if( sum( 1 - p1, ne/3 * (1 - p2) ) <= thresh ) T
    else F
  }
}

pmixture <- function(x, d, dist="chisq")
{
  n <- length(x)
  p <- rep(0, len=n)
  for( i in 1:length(d[,1]) )
    p <- p + pchisq1(x, d[i,1]) * d[i,2]
  p
}

plotmix <- function(d, xlim, ylim=c(0,1), dist="chisq",...)
{
  if( missing(xlim) ) {
    mi <- min(d[,1])
    ma <- max(d[,1])
    low <- max(0, sqrt(mi)-3)^2
    high <- (sqrt(ma) + 3)^2
    xlim <- c(low,high)
  }
  x <- seq(xlim[1], xlim[2], len=100)
  y <- pmixture(x, d, dist=dist)
  plot(x,y,type="l", xlim=xlim, ylim=ylim, ...)
}
```

pace.q:

```
# ------------------------------------------------------------
#
# Functions for pace regression
#
# ------------------------------------------------------------

contrib <- function(a, as)
{
  as^2 - (as - a)^2
}

bhfun <- function(A, As) {
  hrfun(sqrt(A),sqrt(As))
}

bhrfun <- function(a, as) # hrfun without being divided by (2*a)
{
  contrib(a,as) * dnorm(a,as) + contrib(-a,as) * dnorm(-a,as)
}
```

```
hfun <- function(A, As) {
  hrfun(sqrt(A),sqrt(As))
}

hrfun <- function(a, as) {
  (contrib(a,as)*dnorm(a,as)+contrib(-a,as)*dnorm(-a,as))/(2*a)
}

hmixfun <- function(A, d)
{
  n <- length(A)
  h <- rep(0,n)
  rc <- sqrt(d[,1])
  for( i in 1:n )
    if( A[i] != 0 ) h[i] <- h[i] + hrfun(sqrt(A[i]),rc) %*% d[,2]
  h
}

bffun <- function(A, As)
{
  bfrfun(sqrt(A), sqrt(As))
}

bfrfun <- function(a, as)
# ffun() without being divided by (2*a)
{
  dnorm(a, as) + dnorm(-a, as)
}

ffun <- function(A, As)
{
  frfun(sqrt(A), sqrt(As))
}

frfun <- function(a, as)
{
  ( dnorm(a, as) + dnorm(-a, as) ) / (2*a)
}

fmixfun <- function(A, d)
{
  n <- length(A)
  f <- rep(0,n)
  rc <- sqrt(d[,1])
  for( i in 1:n )
    f[i] <- f[i] + frfun(sqrt(A[i]),rc) %*% d[,2]
  f
}

pace2 <- function(A, d)
{
  k <- length(A)
  con <- cumsum( hmixfun(A, d) / fmixfun(A, d) )
  ma <- max(0,con)
  ima <- match(ma, con, nomatch=0)    # number of positive a's
  if(ima < k) A[(ima+1):k] <- 0       # if not the last one
```

```
  A
}

pace4 <- function(A, d)
{
  con <- hmixfun(A, d) / fmixfun(A, d)
  A[con <= 0] <- 0
  A
}

pace6 <- function(A, d, t = 0.5)
{
  for( i in 1:length(A) ) {
    de <- sum( bffun(A[i], d[,1]) * d[,2] )
    if(de > 0)
      A[i] <- (sum(sqrt(d[,1]) * bffun(A[i], d[,1])*d[,2])/de)^2
  }
  A[A<=t] <- 0
  A
}

pace <- function(x,y,va,yname=NULL,ycol=0,pvt=NULL,kp=0,
                 intercept=T,ne=0,
                 method=c("pace6","pace2","pace4","olsc","ols",
                   "ful ","null","aic","bic","ric","cic"),
                 tau=2,tol=1e-6)
{
  method <- match.arg(method)
  if ( is.data.frame(x) ) x <- as.matrix(x)
  ncol <- dim(x)[2]
  if(is.null(dimnames(x)[[2]]))
    dimnames(x) <- list(dimnames(x)[[1]],paste("X",1:ncol,sep=""))
  namelist <- dimnames(x)[[2]]
  if ( missing(y) ) {
    if(ycol == 0 || ycol == "last") ycol <- ncol
    y <- x[,ycol]
    if( is.null(yname) ) yname <- dimnames(x)[[2]][ycol]
    x <- x[,-ycol]
    if ( ! is.matrix(x) ) x <- as.matrix(x)
    dimnames(x) <- list(NULL,namelist[-ycol])
  }
  if ( ! is.matrix(y) ) y <- as.matrix(y)
  if (intercept) {
    x <- cbind(1,x)
    ki <- 1
    kp <- kp + 1
    if(! is.null(pvt)) pvt <- c(1,pvt+1)
    dimnames(x)[[2]][1] <- "(Intercept)"
  }
  else ki <- 0
  m <- as.integer(dim(x)[1])
  n <- as.integer(dim(x)[2])
  mn <- min(m,n)
  if ( m !=  dim(y)[1] )
    stop("dim(x)[1] != dim(y)[1]")
  # first kp columns in pvt[] will maintain the same order
```

```
kp <- max(ki,kp)
# QR transformation
if(method == "full" ) ans <- lsqr(x,y,pvt,tol=0,ks=kp)
else ans <- lsqr(x,y,pvt,ks=kp,tol=tol)
kc <- ks <- ans$ks      # kc is psuedo-rank (unchangable),
                        # ks is used as the model dimensionality
b2 <- ans$b^2
if(missing(va)) {
  if ( m - ks <= 0 ) va <- 0          # data wholly explained
  else va <- sum(b2[(ks+1):m])/(m-ks) # OLS variance estimator
}
At <- A <- d <- NULL
mi <- min(b2[ki:ks])
if(va > max(1e-10, mi * 1e-10) ) {     # Is va is tiny?
  A <- b2[1:ks] / va
  names(A) <- dimnames(x)[[2]][ans$pvt[1:ks]]
}
ne <- ne + (mn - ks)
# Check (asymptotically) olsc-equivalent methods.
switch( method,
       "full" = ,           # full is slightly different from
                            # ols in that collinearity not
                            # excluded
       "ols" = {            # ols = olsc(0)
         method <- "olsc"
         tau <- 0
       },
       "aic" = {            # aic = olsc(2)
         method <- "olsc"
         tau <- 2
       },
       "bic" = {            # bic = olsc(log(n))
         method <- "olsc"
         tau <- log(m)
       },
       "ric" = {            # ric = olsc(2log(k))
         method <- "olsc"
         tau <- 2 * log(ks)
       },
       "null" = {           # null = olsc(+inf)
         method <- "olsc"
         tau <- 2*max(b2) + 1e20
         ks <- kp
       }
       )
switch( method,             # default method is pace2,4,6
       "olsc" = {           # olsc-equivalent methods
         if( ! is.null(A) ) {
           At <- A
           names(At) <- names(A)
           if(kp < ks)
             ksp <- match(max(0,cum<-cumsum(At[(kp+1):ks]-tau)),
                          cum, nomatch=0)
           else ksp <- 0
           if(kp+ksp+1 <= ks) At[(kp+ksp+1):ks] <- 0
           ks <- kp + ksp
```

```
            ans$b[1:ks] <- sign(ans$b[1:ks])*sqrt(At[1:ks]*va)
          }
        },
        "cic" = {                    # cic method
          if( ! is.null(A) ) {
            At <- A
            names(At) <- names(A)
            kl <- ks - kp
            if(kp < ks)
              ksp <- match(max(0,cum <-
                               cumsum(At[(kp+1):ks] -
                                      4*log((ne+kl)/1:kl))),
                           cum, nomatch=0)
            else ksp <- 0
            if(kp+ksp+1 <= ks) At[(kp+ksp+1):ks] <- 0
            ks <- kp + ksp
            ans$b[1:ks] <- sign(ans$b[1:ks]) *
              sqrt(At[1:ks] * va)
          }
        },
        if( ! is.null(A) && ks > 1) { # runs when va is not tiny
          At <- A
          names(At) <- names(A)
          d <- mixing(A[2:ks],ne=ne)     # mixing distribution
          At[2:ks] <- get(method)(A[2:ks],d)
          # get rid of zero coefs
          ks <- ks - match(F, rev(At[2:ks] < 0.001),
                           nomatch = ks) + 1
          ans$b[1:ks] <- sign(ans$b[1:ks]) * sqrt(At[1:ks] * va)
        }
        else {   # delete variables that explain little data
          ave <- mean(b2[1:ks])
          ks <- ks - match(T, rev( b2[1:ks] > 1e-5 * ave),
                           nomatch=0) + 1
        }
        )
  coef <- rsolve(ans$a[1:ks,],ans$b[1:ks,],ans$pvt[1:ks],ks)
  i <- sort.list(ans$pvt[1:ks])
  pvt <- ans$pvt[i]
  coef <- coef[i]
  names(coef) <- dimnames(x)[[2]][pvt[1:ks]]
  if( ! is.null(yname) ) {
    coef <- as.matrix(coef)
    dimnames(coef)[[2]] <- yname
  }
  fit <- list(coef = coef, pvt = pvt, ycol = ycol, intercept =
              intercept, va = va, ks = ks, kc = kc, ndims = n,
              nobs = m, ne = ne, call = match.call() )
  if( !is.null(A)) fit$A <- A
  if( !is.null(At)) fit$At <- At
  if( !is.null(d)) fit$mixing <- unclass(d)
  class(fit) <- "pace"
  fit
}

print.pace <- function (p) {
```

```
  cat("Call:\n")
  print(p$call)
  cat("\n")
  cat("Coefficients:\n")
  print(p$coef)
  cat("\n")
  cat("Number of observations:",p$nobs,"\n")
  cat("Number of dimensions: ",length(p$coef)," (out of ", p$kc,
      ", plus ", p$ndims - p$kc, " abandoned for collinearity)\n",
      sep="")
}

predict.pace <- function(p,x,y)
{
  if( ! is.matrix(x) ) x <- as.matrix(x)
  n <- dim(x)[1]
  k <- length(p$pvt)
  if( p$intercept )
    if(k == 1) pred <- rep(p$coef[1],n)
    else {
      pred <- p$coef[1] + x[,p$pvt[2:k]-1,drop=FALSE] %*%
        p$coef[2:k]
    }
  else pred <- x[,p$pvt] %*% p$coef
  fit <- list(pred = pred)
  if(! missing(y)) fit$resid <- y - pred
  else if(p$ycol >= 1  && p$ycol <= ncol(x))
    fit$resid <- x[,p$ycol] - pred
  if( length(fit$resid) != 0 ) names(fit$resid) <- NULL
  fit
}
```

util.q:

```
# ------------------------------------------------------------
#
# Utility functions
#
# ------------------------------------------------------------

is.sorted <- function(data)
{
  if( is.na( match("sorted", class(data) ) ) ) F
  else T
}

is.normed <- function(data)
{
  if( is.na( match("normed", class(data) ) ) ) F
  else T
}

normalize <- function(x,...) UseMethod("normalize")
```

```
rchisq1 <- function(n, ncp = 0)
# rchisq() in S-Plus (up to version 5.1) does not work for
# non-central Chi-square distribution.
{
  rnorm( n, mean = sqrt(ncp) )^2
}

pchisq1 <- function(x, ncp = 0)
# The S+ language (both versions 3.4 and 5.1) fails to return 0
# for, say, pchisq(0,1,1). The same function in R seems to work
# fine. In the following, the probability value is set to zero
# when the location is at zero.
# Here we only consider the situation df = 1, using a new
# function name pchisq1. Also, efficiency is taken into account,
# within the application's accuracy requirement.
{
  i <- x == 0 | sqrt(x) - sqrt(ncp) < -10
  j <- sqrt(x) - sqrt(ncp) > 10
  p <- rep(0,length(x))
  p[j] <- 1
  p[!i & !j] <- pchisq(x[!i & !j], 1, ncp)
  p
}
```

ls.f:

```
c     ------------------------------------------------------
      subroutine upw2(a,mda,j,k,m,s2)
      double precision a(mda,*), s2, e2
      integer j,k,m

      e2 = a(k,j) * a(k,j)
      if( e2 >  .1 *  s2 ) then
         s2 = sum2(a,mda,j,k+1,m)
      else
         s2 = s2 - e2
      endif
      end

c     ------------------------------------------------------
c     Constructs single Householder orthogonal transformation
c     Input:  a(mda,*)   ; matrix A
c             mda        ; the first dimension of A
c             j          ; the j-th column
c             k          ; the k-th element
c             s2         ; s2 = d^2
c     Output: a(k,j)     ; = a(k,j) - d
c             d          ; diagonal element of R
c             q          ; = - u'u/2, where u is the transformation
c                        ; vector. Hence should not be zero.
c     ------------------------------------------------------
      subroutine hh1(a,mda,j,k,d,q,s2)
      integer mda,j,k
      double precision a(mda,*),s2,d,q

      if (a(k,j) .ge. 0) then
         d = - dsqrt(s2)
      else
         d = dsqrt(s2)
      endif
      a(k,j) = a(k,j) - d
      q = a(k,j) * d
      end

c     ------------------------------------------------------
c     Performs single Householder transformation on one column
c     of a matrix
c
c     Input:  a(mda,)
c             m           ; a(k..m,j) stores the transformation
c                         ; vectur u
c             j           ; the j-th column
c             k           ; the k-th element
c             q           ; q = -u'u/2; must be negative
c             b(mdb,)     ; stores the to-be-transformed vector
c             l           ; b(k..m,l) to be transformed
c     Output: b(k..m,l)   ; transformed part of b
c     ------------------------------------------------------
      subroutine hh2(a,mda,m,j,k,q,b,mdb,l)
```

```
      integer mda,m,k,j,mdb,l
      double precision a(mda,*),q,b(mdb,*),s,alpha

      s = 0.0
      do 10 i=k,m
         s = s + a(i,j) * b(i,l)
 10   continue
      alpha = s / q
      do 20 i=k,m
         b(i,l) = b(i,l) + alpha * a(i,j)
 20   continue
      end

c     ------------------------------------------------------
c     Constructs Givens orthogonal rotation matrix
c
c     Algorithm refers to P58, Chapter 10 of
c     C. L. Lawson and R. J. Hanson (1974).
c     "Solving Least Squares Problems". Prentice-Hall, Inc.
c     ------------------------------------------------------
      subroutine gg1(v1, v2, c, s)
      double precision v1, v2, c, s
      double precision ma, r

      ma = max(dabs(v1), dabs(v2))
      if (ma .eq. 0.0) then
         c = 1.0
         s = 0.0
      else
         r = ma * dsqrt((v1/ma)**2 + (v2/ma)**2)
         c = v1/r
         s = v2/r
      endif
      end

c     ------------------------------------------------------
c     Performs Givens orthogonal transformation
c
c     Algorithm refering to P59, Chapter 10 of
c     C. L. Lawson an R. J. Hanson (1974).
c     "Solving Least Squares Problems". Prentice-Hall, Inc.
c     ------------------------------------------------------
      subroutine gg2(c, s, z1, z2)
      double precision c, s, z1, z2
      double precision w

      w  = c * z1 + s * z2
      z2 = -s * z1 + c * z2
      z1 = w
      end

c     ------------------------------------------------------
      function sum(a,mda,j,l,h)
      integer mda,j,l,h,i
      double precision a(mda,*)
```

```
      sum = 0.0
      do 10 i=1,h
         sum = sum + a(i,j)
 10   continue
      end

c     ------------------------------------------------------
      function sum2(a,mda,j,l,h)
      integer mda,j,l,h,i
      double precision a(mda,*)

      sum2 = 0.0
      do 10 i=1,h
         sum2 = sum2 + a(i,j) * a(i,j)
 10   continue
      end

c     ------------------------------------------------------
c     stepwise least sqaures QR decomposition for equation
c             a x = b
c     i.e., a is QR-decomposed and b is QR-transformed
c     accordingly. Stepwise here means each calling of this
c     function either adds or deletes a column vector, into or
c     from the vector pvt(0..ks-1).
c
c     It is the caller's responsibility that the added column is
c     nonsingular
c
c     Input:  a(mda,*)  ; matrix a
c             m,n       ; (m x n) elements are occupied
c                       ; it is likely mda = m
c             b(mdb,*)  ; vector or matrix b
c                       ; if is matrix, only the first column
c                       ; may help the transformation
c             nb        ; (m x nb) elements are occupied
c             pvt(kp)   ; pivoting column indices
c             kp        ; kp elements in pvt() are considered
c             ks        ; on input, the firs ks columns indexed
c                       ; in pvt() are already QR-transformed.
c             j         ; the pvt(j)-th column of a will be added
c                       ; (if add = 1) or deleted (if add = -1)
c             w(2,*)    ; w(1,pvt) saves the total sum of squares
c                       ; of the used column vectors.
c                       ; w(2,pvt) saves the unexplained sum of
c                       ; squares.
c             add       ; = 1,  adds a new column and does
c                       ;       QR transformation.
c                       ; = -1, deletes the column
c     Output: a(,)      ; QR transformed
c             b(,)      ; QR transformed
c             pvt()     ;
c             ks        ; ks = ks + 1, if add = 1
c                       ; ks = ks - 1, if add = -1
c             w(2, *)   ; changed accordingly
c     ------------------------------------------------------
      subroutine steplsqr(a,mda,m,n,b,mdb,nb,pvt,kp,ks,j,w,add)
```

```
      integer mda,m,n,mdb,nb,pvt(*),j,ks,kp
      double precision a(mda,*),b(mdb,*),w(2,*),q,d,c,s
      integer pj,k,add,pk

      if( add .eq. 1 ) then
c     -- add pvt(j)-th column
         ks = ks + 1
         pj = pvt(j)
         pvt(j) = pvt(ks)
         pvt(ks) = pj

         call hh1(a,mda,pj,ks,d,q,w(2,pj))

         do 20 k = ks+1,kp
            pk = pvt(k)
            call hh2( a,mda,m,pj,ks,q,a,mda,pk )
            call upw2( a,mda,pk,ks,m,w(2,pk) )
 20      continue
         do 30 k = 1,nb
            call hh2( a,mda,m,pj,ks,q,b,mdb,k )
            call upw2( b,mdb,k,ks,m,w(2,n+k) )
 30      continue
         w(2, pj) = 0.0
         a(ks, pj) = d
         do 40 k = ks+1, m
            a(k,pj) = 0.0
 40      continue
 50      continue
      else
c     -- delete pvt(j)-th column (swap with pvt(ks) and
c     -- ks decrements)
         ks = ks - 1
         pj = pvt(j)
         do 110 i = j, ks
            pvt(i) = pvt(i+1)
 110     continue
         pvt(ks+1) = pj
c     -- givens rotation --
         do 140 i=j, ks
            call gg1( a(i,pvt(i)), a(i+1,pvt(i)), c, s )
            do 120 l=i, kp
               call gg2( c, s, a(i,pvt(l)), a(i+1,pvt(l)) )
 120        continue
            do 130 l = 1, nb
               call gg2( c, s, b(i,l), b(i+1,l))
 130        continue
 140     continue
         do 145 j = ks+1, kp
            pj = pvt(j)
            w(2,pj) = w(2,pj) + a(ks+1,pj) * a(ks+1,pj)
 145     continue
         do 150 l = 1, nb
            w(2,n+l) = w(2,n+l) + b(ks+1,l) * b(ks+1,l)
 150     continue
      endif
      end
```

```
c     ------------------------------------------------------
c     The pvt(j)-th column is chosen
c
c     choose the column with the largest diagonal element
c
c     type = 1 VIF-based, otherwise based on diagonal element
c
c     Input: pvt           ; column vector
c            ks            ; number of
c            j             ;
c     ------------------------------------------------------
      subroutine colj(pvt,ks,j,ln,w,tol,type)
      integer pvt(*),ks,j,l,ln,type
      double precision w(2,*),ma,now,tol

      ma = tol * 0.1
      do 10 l=ks+1,ln
         if (ks .eq. 0) type = 2
         now = colmax( w, pvt(l), type )
         if (now .le. ma) goto 10
         ma = now
         j = l
 10   continue
      if(ma .lt. tol) j = 0
      end

c     ------------------------------------------------------
c     returns the pivoting value, being subject to the
c     examination < tol
c     Input:  w(2,*)       ; w(1,j), the total sum of squares
c                          ; w(2,j), the unexplained sum of
c                          ; squares
c             j            ; the j-th column vector
c             type         ; = 1, VIF-based
c                          ; = 2, based on the diagonal element
c     Output: colmax       ; value
c     ------------------------------------------------------
      function colmax(w,j,type)
      integer j, type
      double precision w(2,*)

      if (type .eq. 1) then
         if(w(1,j) .le. 0.0) then
            colmax = 0.0
         else
            colmax = w(2,j) / w(1,j)
         endif
      else
         if(w(2,j) .le. 0.0) then
            colmax = 0.0
         else
            colmax = dsqrt( w(2,j) )
         endif
      endif
      end
```

```
c     ------------------------------------------------------
c     QR decomposition for the linear regression problem a x = b
c
c     Important note: on entry, the vector pvt[1:np] is given (and
c     supposedly correct), the first ks (could be zero) elements
c     in pvt(*) will always be kept inside the final selection.
c     This is useful when, for example, the constant term is
c     always included in the final model.
c     ------------------------------------------------------
      subroutine lsqr(a,mda,m,n,b,mdb,nb,pvt,kp,ks,w,tol,type)
      integer pvt(*),tmp,ld
      integer mda,m,n,mdb,nb,lp,kp,ks,type,pj
      double precision a(mda,*), b(mdb,*)
      double precision w(2,*),tol,ma

      if(tol .lt. 1e-20) tol = 1e-20
      ld = min(m,kp)
c     computes the squared sum for each column of a and b
      do 10 j=1,kp
         pj = pvt(j)
         w(1,pj) = sum2(a,mda,pj,1,m)
         w(2,pj) = w(1,pj)
 10   continue
      do 15 j=1,nb
         w(1,n+j) = sum2(b,mdb,j,1,m)
         w(2,n+j) = w(1,n+j)
 15   continue

c     The first ks columns defined in pvt(*) are pre-chosen.
c     However, they are subject to rank examination.
      lp = ks
      ks = 0
      do 17 j=1,lp
         pj = pvt(j)
         ma = colmax(w,pj,type)
         if(ma > tol) then
            call steplsqr(a,mda,m,n,b,mdb,nb,pvt,kp,ks,j,w,1)
            if ( ks .ge. ld) go to 18
         endif
 17   continue
 18   continue
c     lp is the number of pre-chosen column variables.
      lp = ks
      ld = min(ld,kp)

c     the rest columns defined in pvt(*) are subject to
c     rank examination
      do 50 k=ks+1,ld
c     -- choose the pivoting column j --
         call colj(pvt,ks,j,kp,w,tol,type)
         if(j .eq. 0) then
            kp = k - 1
            go to 60
         endif
         call steplsqr(a,mda,m,n,b,mdb,nb,pvt,kp,ks,j,w,1)
```

```
 50   continue
 60   continue

c     if ks is small, or less than n, forward is unnecessary.
c     Forward selection is only necessary when backward is
c     insufficient, such as there are too many (noncollinear)
c     variables.
      if( kp .gt. n .or. kp .gt. 200) then
         call forward(a,mda,ks,n,b,mdb,nb,pvt,kp,ks,lp,w)
      endif

      call backward(a,mda,ks,n,b,mdb,nb,pvt,kp,ks,lp,w)

      end

c     ------------------------------------------------------------
c     forward :  forward ordering based on data explanation
c             a x = b
c
c     Input:  a(mda,*)   ; matrix to be QR-transformed
c             m,n        ; (m x n) elements are occupied
c             b(mdb,nb)  ; (m x nb) elements are occupied
c             pvt(kp)    ; indices of pivoting columns
c             kp         ; the first kp elements in pvt()
c                        ; are to be used
c             ks         ; psuedo-rank of a
c             lp         ; the first lp elements indexed in pvt()
c                        ; are already QR-transformed and will not
c                        ; change.
c             w(2,n+nb)  ; w(1,) stores the total sum of each
c                        ; column of a and b.
c                        ; w(2,) stores the unexplained sum of
c                        ; squares
c     Output: a(,)       ; Upper triagular matrix by
c                        ; QR-transformation
c             b(,)       ; QR-transformed
c             pvt()      ; re-ordered
c     ------------------------------------------------------------
      subroutine forward(a,mda,m,n,b,mdb,nb,pvt,kp,ks,lp,w)
      integer pvt(*),tmp,ld
      integer mda,m,n,mdb,nb,kp,ks,pj,jma,jth,li
      double precision a(mda,*), b(mdb,*)
      double precision w(2,*),ma,sum,r

      do 1 j=lp+1,kp
         pj = pvt(j)
         w(2,pj) = sum2(a,mda,pj,lp+1,m)
 1    continue

      do 2 j=1,nb
         w(2,n+j) = sum2(b,mdb,j,lp+1,m)
 2    continue

      do 130 i=lp+1,ks
         ma = -1e20
         jma = 0
```

```
         do 20 j=i,kp
c     -- choose the pivoting column j based on data explanation --
            pj = pvt(j)
            sum = 0.0
            do 10 k = i, n
               sum = sum + a(k,pj) * b(k,1)
 10         continue
c           w(2,pj) should not be zero
            r = sum * sum / w(2,pj)
            if(r > ma) then
               ma = r
               jma = j
            endif
 20      continue

         if(jma .eq. 0) then
            ks = i
            goto 200
         else
            call steplsqr(a,mda,m,n,b,mdb,nb,pvt,kp,i-1,jma,w,1)
         endif
 130  continue

 200  continue

      end

c     ------------------------------------------------------------
c     backward variable elimination
c     ------------------------------------------------------------
      subroutine backward(a,mda,m,n,b,mdb,nb,pvt,kp,ks,lp,w)
      integer pvt(*),tmp,lp
      integer mda,m,n,mdb,nb,kp,ks,jmi,jth,kscopy,pj
      double precision a(mda,*), b(mdb,*)
      double precision w(2,*),ma,absd,sum
      double precision xxx(ks+nb),s2

c     backward elimination; the rank ks is not supposed to change
      kscopy = ks
      do 50 j = ks, lp+1, -1
         call coljd(a,mda,m,n,b,mdb,nb,pvt,kp,ks,lp+1,w,jmi)
         call steplsqr(a,mda,m,n,b,mdb,nb,pvt,kp,ks,jmi,w,-1)
 50   continue

c     restores the rank and data explanation
      ks = kscopy
      do 100 j=lp+1,kp
         pj = pvt(j)
         s2 = sum2(a,mda,pj,lp+1,ks)
         w(2,pj) = w(2,pj) - s2
 100  continue

      do 110 j=1,nb
         s2 = sum2(b,mdb,j,lp+1,ks)
         w(2,n+j) = w(2,n+j) - s2
 110  continue
```

```
      end

c     ------------------------------------------------------------
      subroutine coljd(a,mda,m,n,b,mdb,nb,pvt,kp,ks,jth,w,jmi)
      integer pvt(*),tmp,ld
      integer mda,m,n,mdb,nb,kp,ks,pj,jmi,jth
      double precision a(mda,*), b(mdb,*)
      double precision w(2,*),mi,val

      mi = dabs( b(ks,1) )
      jmi = ks
      do 50 j = jth, ks-1
         call coljdval(a,mda,b,mdb,pvt,ks,j,val)
         if (val .le. mi) then
            mi = val
            jmi = j
         endif
 50   continue
      end

c     ------------------------------------------------------------
      subroutine coljdval(a,mda,b,mdb,pvt,ks,jth,val)
      integer pvt(*)
      integer mda,mdb,ks,jth
      double precision a(mda,*), b(mdb,*)
      double precision val,c,s
      double precision xxx(ks)

      do 10 j = jth+1, ks
         xxx(j) = a(jth,pvt(j))
 10   continue
      val = b(jth,1)

      do 50 j = jth+1, ks
         call gg1( xxx(j), a(j,pvt(j)), c, s )
         do 30 l=j+1, ks
            xxx(l) = -s * xxx(l) + c * a(j,pvt(l))
 30      continue
         val = -s * val + c * b(j,1)
 50   continue
      val = dabs( val )
      end

c     ------------------------------------------------------------
      subroutine rsolve(a,mda,b,mdb,nb,pvt,ks)
      integer mda,mdb,nb,ks
      integer pvt(*)
      double precision a(mda,*),b(mdb,*)

      if(ks .le. 0) return
      do 50 k=1,nb
         b(ks,k) = b(ks,k) / a(ks,pvt(ks))
         do 40 i=ks-1,1,-1
            sum = 0.0
            do 30 j=i+1,ks
```

```
               sum = sum + a(i,pvt(j)) * b(j,k)
30          continue
            b(i,k) = (b(i,k) - sum) / a(i,pvt(i))
40       continue
50    continue
      end
```

References

Aha, D. W., Kibler, D. and Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, *6*, 37–66.

Akaike, H. (1969). Fitting autoregressive models for prediction. *Ann. Inst. Statist. Math.*, *21*, 243–247.

Akaike, H. (1970). Statistical predictor identification. *Ann. Inst. Statist. Math.*, *22*, 203–217.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *2nd International Symposium on Information Theory* (pp. 267–281). Akadémiai Kiadó, Budapest. (Reprinted in S. Kotz and N. L. Johnson (Eds.). (1992). Breakthroughs in Statistics. Volume 1, 610–624. Springer-Verlag.).

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Croz, J. D., Greenbaum, A., Hammarling, S., McKenney, A. and Sorensen, D. (1999). *LAPACK Users' Guide* (3rd Ed.). Philadelphia, PA: SIAM Publications.

Atkeson, C., Moorey, A. and Schaalz, S. (1997). Locally weighted learning. *Artificial Intelligence Review*, *11*, 11–73.

Barbe, P. (1998). Statistical analysis of mixtures and the empirical probability measure. *Acta Applicandae Mathematicae*, *50(3)*, 253–340.

Barnard, G. A. (1949). Statistical inference (with discussion). *J. Roy. Statist. Soc., Ser. B*, *11*, 115–139.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Phil. Trans. Roy. Soc.*, *53*, 370–418. Reprinted in (1958) *Biometrika, 45*, 293–315.

Berger, J. O. (1985). *Statistical decision theory and Bayesian analysis* (2nd Ed.). New York: Springer-Verlag.

Birnbaum, A. (1962). On the foundations of statistical inference (with discussion). *J. Amer. Statist. Assoc.*, *57*, 269–326. (Reprinted in S. Kotz and N. L. Johnson (Eds.). (1992). Breakthroughs in Statistics. Volume 1, 478–518. Springer-Verlag.).

Blake, C., Keogh, E. and Merz, C. J. (1998). UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Department of Information and Computer Science, University of California, Irvine.

Blum, J. R. and Susarla, V. (1977). Estimation of a mixing distribution function. *Ann. Probab.*, *5*, 200–209.

Böhning, D. (1982). Convergence of Simar's algorithm for finding the maximum likelihood estimate of a compound Poisson process. *Ann. Statist.*, *10*, 1006–1008.

Böhning, D. (1985). Numerical estimation of a probability measure. *Journal of Statistical Planning and Inference*, *11*, 57–69.

Böhning, D. (1986). A vertex-exchange-method in D-optimal design theory. *Metrika*, *33*, 337–347.

Böhning, D. (1995). A review of reliable algorithms for the semi-parametric maximum likelihood estimator of a mixture distribution. *Journal of Statistical Planning and Inference*, *47*, 5–28.

Böhning, D., Schlattmann, P. and Lindsay, B. (1992). C.A.MAN (computer assisted analysis of mixtures): Statistical algorithms. *Biometrics*, *48*, 283–303.

Breiman, L. (1992). The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error. *Journal of the American Statistics Association*, *87*, 738–754.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics*, *37(4)*, 373–384.

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont CA.

Breiman, L. and Spector, P. (1992). Submodel selection and evaluation in regression. the x-random case. *International Statistical Review*, *60*, 291–319.

Brown, L. D. (1990). An ancillarity paradox which appears in multiple linear regression (with discussion). *Ann. Statist.*, *18*, 471–538.

Burnham, K. P. and Anderson, D. R. (1998). *Model selection and inference : a practical information-theoretic approach*. New York: Springer.

Carlin, B. P. and Louis, T. A. (1996). *Bayes and Empirical Bayes Methods for Data Analysis*. London: Chapman & Hall.

Carlin, B. P. and Louis, T. A. (2000). Empirical Bayes: Past, present and future. *J. Am. Statist. Assoc.* (To appear).

Chatfield, C. (1995). Model uncertainty, data mining and statistical inference. *J. Roy. Statist. Soc. Ser. A*, *158*, 419–466.

Choi, K. and Bulgren, W. B. (1968). An estimation procedure for mixtures of distributions. *J. R. Statist. Soc. B*, *30*, 444–460.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplot. *Journal of the American Statistical Association*, *74*, 829–836.

Coope, I. D. and Watson, G. A. (1985). A projected Lagrangian algorithm for semi-infinite programming. *Mathematical Programming*, *32*, 337–293.

Copas, J. B. (1969). Compound decisions and empirical Bayes. *J. Roy. Statist. Soc. B, 31*, 397–425.

Copas, J. B. (1983). Regression, prediction and shrinkage (with discussions). *J. Roy. Statist. Soc. B, 45*, 311–354.

Deely, J. J. and Kruse, R. L. (1968). Construction of sequences estimating the mixing distribution. *Ann. Math. Statist., 39*, 286–288.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc., Ser. B, 39*, 1–22.

Derksen, S. and Keselman, H. J. (1992). Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology, 45*, 265–282.

DerSimonian, R. (1986). Maximum likelihood estimation of a mixing distribution. *J. Roy. Statist. Soc. Ser. C, 35*, 302–309.

DerSimonian, R. (1990). Correction to algorithm AS 221: Maximum likelihood estimation of a mixing distribution. *J. Roy. Statist. Soc. Ser. C, 39*, 176.

Dongarra, J. J., Bunch, J., Moler, C. and Stewart, G. (1979). *LINPACK Users' Guide*. Philadelphia, PA: SIAM Publications.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation via wavelet shrinkage. *Biometrika, 81*, 425–455.

Efron, B. (1998). R. A. Fisher in the 21st century. *Statistical Science, 13*, 95–126.

Efron, B. and Morris, C. N. (1971). Limiting the risk of Bayes and empirical Bayes estimators - Part I: The Bayes case. *J. Amer. Statist. Assoc., 66*, 807–815.

Efron, B. and Morris, C. N. (1972a). Empirical Bayes on vector observations: An extension of Stein's method. *Biometrika, 59*, 335–347.

Efron, B. and Morris, C. N. (1972b). Limiting the risk of Bayes and empirical Bayes estimators - Part II: The empirical Bayes case. *J. Amer. Statist. Assoc.*, *67*, 130–139.

Efron, B. and Morris, C. N. (1973a). Combining possibly related estimation problems (with discussion). *J. Roy. Statist. Soc., Ser. B*, *35*, 379–421.

Efron, B. and Morris, C. N. (1973b). Stein's estimation rule and its competitor - An empirical Bayes approach. *J. Amer. Statist. Assoc.*, *68*, 117–130.

Efron, B. and Morris, C. N. (1975). Data analysis using Stein's estimator and its generalizations. *J. Amer. Statist. Assoc.*, *70*, 311–319.

Efron, B. and Morris, C. N. (1977). Stein's paradox in statistics. *Scientific American*, *236(May)*, 119–127.

Efron, B. and Tibshirani, J. (1993). *An Introduction to the Bootstrap*. London: Chapman and Hall.

Fan, J. and Gijbels, I. (1996). *Local Polynomial Modeling and Its Applications*. Chapman & Hall.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (Eds.). (1996). *Advances in Knowledge Discovery and Data Mining*. MIT Press, Cambridge, MA.

Fedorov, V. V. (1972). *Theory of Optimal Experiments*. New York: Academic Press.

Fisher, R. A. (1922). On the mathemmatical foundations of theoretical statistics. *Philos. Trans. Roy. Soc. London, Ser. A*, *222*, 309–368. (Reprinted in S. Kotz and N. L. Johnson (Eds.). (1992). Breakthroughs in Statistics. Volume 1, 11–44. Springer-Verlag.).

Fisher, R. A. (1925). Theory of statistical estimation. *Proc. Camb. Phil. Soc.*, *22*, 700–725.

Foster, D. and George, E. (1994). The risk inflation criterion for multiple regression. *Annals of Statistics*, *22*, 1947–1975.

Frank, E. and Friedman, J. (1993). A statistical view of some chemometrics regression tools (with discussions). *Technometrics, 35*, 109–148.

Friedman, J. (1991). Multiple adaptive regression splines (with discussion). *Ann. Statist., 19*, 1–141.

Friedman, J. and Stuetzle, W. (1981). Projection pursuit regression. *JASA, 76*, 817–823.

Friedman, J. H. (1997). Data mining and statistics: What's the connection? In *Proceedings of the 29th Symposium on the Interface: Computing Science and Statistics*. The Interface Foundation of North America. May. Houston, Texas.

Galambos, J. (1995). *Advanced Probability Theory*. A series of Textbooks and Reference books/10. Marcel Dekker, Inc.

Gauss, C. F. (1809). Theoria motus corporum coelestium. *Werke, 7*. (English transl.: C. H. Davis. Dover, New York, 1963).

Hall, M. A. (1999). *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, Department of Computer Science, University of Waikato.

Hannan, E. J. and Quinn, B. G. (1979). The determination of the order of autoregression. *J. R. Statist. Soc. B, 41*, 190–195.

Hartigan, J. A. (1996). Introduction. In P. Arabie, L. J. Hubert and G. D. Soete (Eds.), *Clustering and Classification* (pp. 1–3). World Scientific Publ., River Edge, NJ.

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

Hoerl, R. W., Schuenemeyer, J. H. and Hoerl, A. E. (1986). A simulation of biased estimation and subset selection regression techniques. *Technometrics, 9*, 269–380.

James, W. and Stein, C. (1961). Estimation with quadratic loss. In *Proc. Fourth Berkeley Symposium on Math. Statist. Prob.*, 1 (pp. 311–319). University of

California Press. (Reprinted in S. Kotz and N. L. Johnson (Eds.). (1992). Breakthroughs in Statistics. Volume 1, 443–460. Springer-Verlag.).

Kiefer, J. and Wolfowitz, J. (1956). Consistercy of the maximum likelihood estimator in the presence of infinitely many incedental parameters. *Ann. Math. Statist.*, *27*, 886–906.

Kilpatrick, D. and Cameron-Jones, M. (1998). Numeric prediction using instance-based learning with encoding length selection. In N. Kasaboc, R. Kozma, K. Ko, R. O'Shea, G. Coghill and T. Gedeon (Eds.), *Progress in Connectionist-Based Information Systems*, Volume 2 (pp. 984–987). Springer-Verlag.

Kotz, S. and Johnson, N. L. (Eds.). (1992). *Breakthroughs in Statistics*, Volume 1. Springer-Verlag.

Kullback, S. (1968). *Information Theory and Statistics* (2nd Ed.). New York: Dover. Reprinted in 1978, Gloucester, MA: Peter Smith.

Kullback, S. and Leibler, R. (1951). On information and sufficiency. *Ann. Math. Statist.*, *22*, 79–86.

Laird, N. M. (1978). Nonparametric maximum likelihood estimation of a mixing distribution. *J. Amer. Statist. Assoc.*, *73*, 805–811.

Lawson, C. L. and Hanson, R. J. (1974, 1995). *Solving Least Squares Problems*. Prentice-Hall, Inc.

Legendre, A. M. (1805). Nouvelles méthods pur la determination des orbits des cométes. (Appendix: Sur la méthode des moindre carrés).

Lehmann, E. L. and Casella, G. (1998). *Theory of point estimation* (2nd Ed.). New York: Spinger-Verlag.

Lesperance, M. L. and Kalbfleisch, J. D. (1992). An algoritm for computing the nonparametric mle of a mixing distribution. *J. Amer. Statist. Assoc.*, *87*, 120–126.

Li, K.-C. (1987). Asymptotic optimality for C_p , C_L , cross-validation and generalized cross-validation: discrete index set. *Ann. Statist.*, *15*, 958–975.

Lindsay, B. G. (1983). The geometry of mixture likelihoods: A general theory. *Ann. Statist.*, *11*, 86–94.

Lindsay, B. G. (1995). *Mixture Models: Theory, Geometry, and Applications*, Volume 5 of *NSF-CBMS Regional Conference Series in Probability and Statistics*. Institute for Mathematical Statistics: Hayward, CA.

Lindsay, B. G. and Lesperance, M. L. (1995). A review of semiparametric mixture models. *Journal of Statistical Planning & Inference*, *47*, 29–39.

Linhart, H. and Zucchini, W. (1989). *Model Selection*. New York: Wiley.

Liu, H. and Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers.

Loader, C. (1999). *Local regression and likelihood*. Statistics and computing series. Springer.

Macdonald, P. D. M. (1971). Comment on a paper by Choi and Bulgren. *J. R. Statist. Soc. B*, *33*, 326–329.

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

Maritz, J. S. and Lwin, T. (1989). *Empirical Bayes Methods* (2nd Ed.). Chapman and Hall.

McLachlan, G. and Basford, K. (1988). *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York.

McQuarrie, A. D. and Tsai, C.-L. (1998). *Regression and time series model selection*. Singapore; River Edge, N.J.: World Scientific.

Miller, A. J. (1990). *Subset Selection in Regression*, Volume 40 of *Monographs on Statistics and Applied Probability*. Chapman & Hall.

Morris, C. N. (1983). Parametric empirical bayes inference: theory and applications. *J. Amer. Statist. Assoc.*, *78*, 47–59.

Neyman, J. (1962). Two breakthroughs in the theory of statistical decision making. *Rev. Inst. Internat. Statist.*, *30*, 11–27.

Pólya, G. (1920). Ueber den zentralen grenswertsatz der wahrscheinlichkeitstheorie und das momentenproblem. *Math. Zeit.*, *8*, 171.

Prakasa Rao, B. L. (1992). *Identifiability in Stochasitc Models: Characterization of Probability Distributions*. Academic Press, New York.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Mateo, Calif.: Morgan Kaufmann Publishers.

Rao, C. R. and Wu, Y. (1989). A strongly consistent procedure for model selection in a regression problem. *Technometrics*, *72(2)*, 369–374.

Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

Rissanen, J. (1983). A universal prior for integers and estimation by minimum description length. *Ann. Statist.*, *11*, 416–431.

Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry*, Volume 15 of *Series in Computer Science*. World Scientific Publishing Co.

Robbins, H. (1950). A generalization of the method of maximum likelihood: Estimating a mixing distribution (abstract). *Ann. Math. Statist.*, *21*, 314–315.

Robbins, H. (1951). Asymptotically subminimax solutions of compound statistical decision problems. In *Proc. Second Berkeley Symposium on Math. Statist. and Prob.*, 1 (pp. 131–148). University of California Press.

Robbins, H. (1955). An empirical Bayes approach to statistics. In *Proc. Third Berkeley Symposium on Math. Statist. and Prob.*, 1 (pp. 157–164). University of California Press. (Reprinted in S. Kotz and N. L. Johnson (Eds.). (1992). Breakthroughs in Statistics. Volume 1, 388–394. Springer-Verlag.).

Robbins, H. (1964). The empirical Bayes approach to statistical decision problems. *Ann. Math. Statist., 35*, 1–20.

Robbins, H. (1983). Some thoughts on empirical Bayes estimation. *Ann. Statist., 11*, 713–723.

Roecker, E. B. (1991). Prediction error and its estimation of subset selected models. *Technometrics, 33*, 459–468.

Schott, J. R. (1997). *Matrix analysis for statistics*. New York: Wiley.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics, 6(2)*, 461–464.

Sclove, S. L. (1968). Improved estimators for coefficients in linear regression. *J. Amer. Statist. Assoc., 63*, 596–606.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal, 27*, 398–403.

Shao, J. (1993). Linear model selection by cross-validation. *J. Amer. Statist. Assoc., 88*, 486–494.

Shao, J. (1996). Bootstrap model selection. *J. Amer. Statist. Assoc., 91*, 655–665.

Shao, J. and Tu, D. (1995). *The Jackknife and Bootstrap*. Springer Verlag.

Shibata, R. (1976). Selection of the order of an autoregressive model by Akaike's information criterion. *Biometrika, 63*, 117–126.

Shibata, R. (1981). An optimal selection of regression variables. *Biometrika, 68*, 45–54.

Shibata, R. (1986). Consistency of model selection and parameter estimation. In J. Gani. and M. Priestley (Eds.), *Essays in time series and applied processes* (pp. 27–41). J. Appl. Prob.

Stein, C. (1955). Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proc. Third Berkeley Symp. on Math. Statist. and Prob.*, 1 (pp. 197–206). University of California Press.

Teicher, H. (1960). On the mixture of distributions. *Ann. Math. Statist.*, *31*, 55–57.

Teicher, H. (1961). Identifiability of mixtures. *Ann. Math. Statist.*, *32*, 244–248.

Teicher, H. (1963). Identifiability of mixtures. *Ann. Math. Statist.*, *34*, 1265–1269.

Thompson, M. L. (1978). Selection of variables in multiple regression, part 1 and part 2. *International Statistical Review*, *46*, 1–21 and 129–146.

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

Tibshirani, R. and Knight, K. (1997). The covariance inflation criterion for model selection. Technical report, Department of Statistics, University of Stanford.

Titterington, D. M., Smith, A. F. M. and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. John Wiley & Sons.

von Neuman, J. (1928). Zur theorie der gesellschafstspiele. *Math. Annalen*, *100*, 295–320.

von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ.

Wald, A. (1939). Contributions to the theory of statistical estimation and hypothesis testing. *Ann. Math. Statist.*, *10*, 299–326.

Wald, A. (1950). *Statistical Decision Functions*. Wiley, New York.

Witten, I. H. and Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann.

Wu, C. F. (1978a). Some algorithmic aspects of the theory of optimal designs. *Ann. Statist.*, *6*, 1286–1301.

Wu, C. F. (1978b). Some iterative procedures for generating nonsigular optimal designs. *Comm. Statist.*, *7*, 1399–1412.

Zhang, P. (1993). Model selection via multifold cross-validation. *Ann. Statist.*, *21*, 299–313.

Zhao, L. C., Krishnaiah, P. R. and Bai, Z. D. (1986). On the detection of the number of signals in the presence of white noise. *Journal of Multivariate Analysis*, *20*, 1–25.