Ronald Christensen
Department of Mathematics and Statistics
University of New Mexico

Statistical Learning:
A Second Course in Regression

Springer

To Emma and Dewey

Preface

"Critical assessment of data is the essential task of the educated mind."
Professor Garrett G. Fagan, Pennsylvania State University.
The last words in his audio course *The Emperors of Rome*, The Teaching Company.

"Statistical Learning is the uncritical assessment of data."
Professor Ronald Christensen, University of New Mexico.
The last words in his ignominious career prior to retirement.

"In the beginning was the Linear Model. And the Linear Model was with Statistics.
And the Linear Model was Statistics."
Professor Fletcher Christensen, University of New Mexico.

"There are essentially only two worthwhile things anybody knows how to do in
Statistics: linear approximations and simulations."
Professor Ronald Christensen, University of New Mexico. (Previous, less accurate,
version was "Linear Models and MCMC.")

Preface to the First and Last Edition

Due to the current (overwhelming?) popularity of *statistical learning*, I decided to consolidate the statistical learning material from some of my other books into one source, *independent* of the other books, and to add in some additional statistics topics that get designated as statistical learning. This came about because I recently completed new editions of three books: Christensen (2015), *Analysis of Variance, Design, and Regression: Linear Models for Unbalanced Data (ANREG-II)*, Christensen (2020), *Plane Answers to Complex Questions: The Theory of Linear Models*

(PA or for the fifth edition PA-V) and Christensen (2019) *Advanced Linear Modeling: Statistical Learning and Dependent Data (ALM-III)*. The new editions include material on statistical learning as it applies to traditional topics in Statistics but (despite the subtitle for one of them) none of those books really focus on statistical learning. The most efficient way to perform the consolidation seemed to be by assuming that the reader had already been exposed to a course in regression analysis. The book *does not* presuppose that the reader knows linear model theory. Consolidating all of this material into a cogent whole turned out to be far more work than I was expecting. Although this book should be accessible without any of the other books, I frequently refer to the other books for details and theory not contained in this one. I like to point out that the number of references to my other work is at least as much about sloth as it is about ego.

This book's computer code http://www.stat.unm.edu/~fletcher/R-SL.pdf is being pieced together from that of the applications book *ANREG-II*, http://www.stat.unm.edu/~fletcher/Rcode.pdf, as well as that for *ALM-III*, http://www.stat.unm.edu/~fletcher/R-ALMIII.pdf, with additions for the topics covered here that were not included in either of those other books. The data files for this book can be downloaded from https://www.stat.unm.edu/~fletcher/SL-Data.zip with the data in Table x.y appearing in file SLx-y.dat.

After reviewing standard linear regression in Chapters 1 and 2 we observe that most methods of nonparametric regression are merely applications of standard linear regression but that they often employ estimates of the regression parameters that are alternatives to the traditional least squares estimates. After doing standard regression, we introduce binomial and binary regression. These methods include (regularized, nonparametric) logistic regression and support vector machines. Finally, we introduce some topics from multivariate analysis that are commonly used in statistical learning.

The presentation assumes that the reader has encountered basic ideas of probability such as expected values, variances, covariances, and, for some topics near the end of the book, conditional expectations and densities of multivariate distributions. (*Any* calculus based statistics course should be sufficient but perhaps not necessary background.) We do not do any sophisticated probability, the reader needs merely not to be freaked out by the concepts. An extensive background in analysis of variance (ANOVA) is not necessary but I find ANOVA unavoidable when discussing a few of the topics. The whole idea of generalized additive models is based on an analogy with multifactor ANOVA and the discussion of regression trees also requires some knowledge of multifactor ANOVA. Moreover, there exist strong relationships between one-way multivariate ANOVA (MANOVA) and one of the discrimination procedures discussed, namely, linear discriminant analysis (LDA). For readers without an extensive ANOVA background, Appendix B contains an example of a three-factor ANOVA and Appendix C contains an example of a three-factor MANOVA. Starting with three-factor examples is really dumping you into the deep end of the pool, so each three-factor analysis is introduced using its equivalent one-factor analysis.

I would like to thank Carlos Torres Inga and Kevin J. Kloeppel for comments that led to improvements in the discussion of cluster analysis. Penny Darsey caught a number of typos. Also, I forgot to thank Andrew Ng in *ALM-III* for posting a video that helped me to understand support vector machines, namely "Support Vector Machines — Optimization Objective."

Contents

Chapter 1
Linear Regression

Abstract This chapter reviews basic ideas of linear regression.

Regression involves predicting values of a dependent variable from a collection of other (predictor) variables. Linear regression employs a prediction function that is a linear combination of the values of the predictor variables. Most forms of nonparametric regression are actually linear regression methods. The complete set of predictor variables can include not only whatever original predictor variables that were measured but nonlinear transformations of those original predictor variables. This allows predictor functions that are complicated nonlinear functions of the original measured predictors while still being linear combinations of the complete set of predictors.

Traditionally, all observed variables in regression were measurement variables in the sense that they resulted from measurements taken on objects. As such, these variables typically have measurement units associated with them like millimeters, grams, inches, pounds, etc. It can also be useful to incorporate as predictor variables factor/categorical variables that are used to indicate group membership. To incorporate categorical predictors into a linear regression, they need to be replaced by a collection of 0-1 indicators that identify each of the various group categories. Transformations of categorical predictors serve no useful purpose unless they change the group structure. (For example, if we think that two groups act alike, we can transform the categorical predictor so that they become the same group.)

For simplicity, in this chapter we review linear regression methods using illustrations that employ only the original measured variables. Later chapters discuss systematic methods for defining transformations of the original predictors. Nonetheless, the essential behavior of linear regression models in no way depends on how the predictor variables are obtained, so this review really applies equally well to nearly all of our approaches to nonparametric regression.

1.1 An Example

The Coleman Report data were given in Mosteller and Tukey (1977). The data consist of measurement variables from schools in the New England and Mid-Atlantic states. The predictor variables are x_1, staff salaries per pupil; x_2, percentage of sixth graders whose fathers have white-collar jobs; x_3, a composite measure of socioeconomic status; x_4, the mean score of a verbal test given to the teachers; and x_5, the mean educational level of the sixth graders' mothers (one unit equals two school years). The dependent variable y is the mean verbal test score for sixth graders. The data are given in Table 1.1. Figures 1.1 through 1.4 provide pairwise plots all of the variables, i.e., a scatterplot matrix of the variables.

Table 1.1 *Coleman Report* data.

School	y	x_1	x_2	x_3	x_4	x_5
1	37.01	3.83	28.87	7.20	26.60	6.19
2	26.51	2.89	20.10	−11.71	24.40	5.17
3	36.51	2.86	69.05	12.32	25.70	7.04
4	40.70	2.92	65.40	14.28	25.70	7.10
5	37.10	3.06	29.59	6.31	25.40	6.15
6	33.90	2.07	44.82	6.16	21.60	6.41
7	41.80	2.52	77.37	12.70	24.90	6.86
8	33.40	2.45	24.67	−0.17	25.01	5.78
9	41.01	3.13	65.01	9.85	26.60	6.51
10	37.20	2.44	9.99	−0.05	28.01	5.57
11	23.30	2.09	12.20	−12.86	23.51	5.62
12	35.20	2.52	22.55	0.92	23.60	5.34
13	34.90	2.22	14.30	4.77	24.51	5.80
14	33.10	2.67	31.79	−0.96	25.80	6.19
15	22.70	2.71	11.60	−16.04	25.20	5.62
16	39.70	3.14	68.47	10.62	25.01	6.94
17	31.80	3.54	42.64	2.66	25.01	6.33
18	31.70	2.52	16.70	−10.99	24.80	6.01
19	43.10	2.68	86.27	15.03	25.51	7.51
20	41.01	2.37	76.73	12.77	24.51	6.96

It is of interest to examine the correlations between y and each of the predictor variables.

	x_1	x_2	x_3	x_4	x_5
Correlation with y	0.192	0.753	0.927	0.334	0.733

Of the five variables, x_3 has the highest correlation. It explains more of the y variability than any other single variable. Variables x_2 and x_5 also have reasonably high correlations with y. Low correlations exist between y and both x_1 and x_4. Interestingly, x_1 and x_4 turn out to be more important in explaining y than either x_2 or x_5.

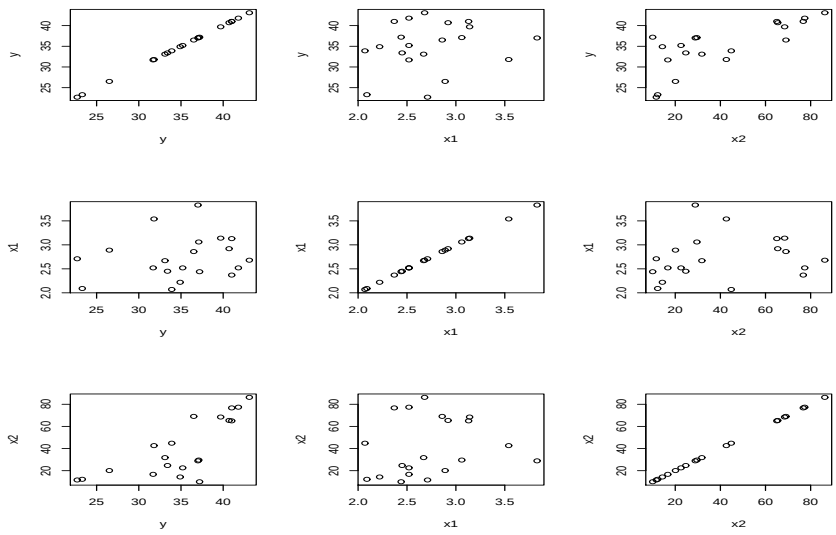

Fig. 1.1 Scatterplot matrix for *Coleman Report* data.

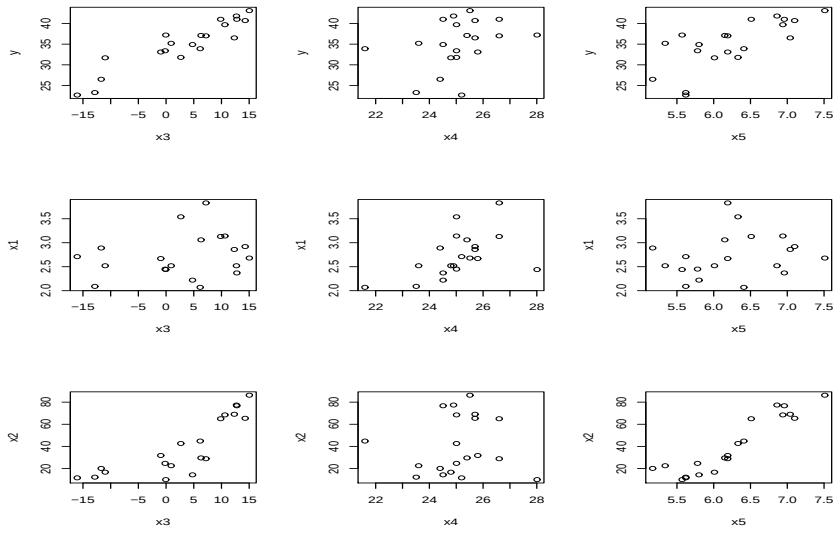

Fig. 1.2 Scatterplot matrix for *Coleman Report* data.

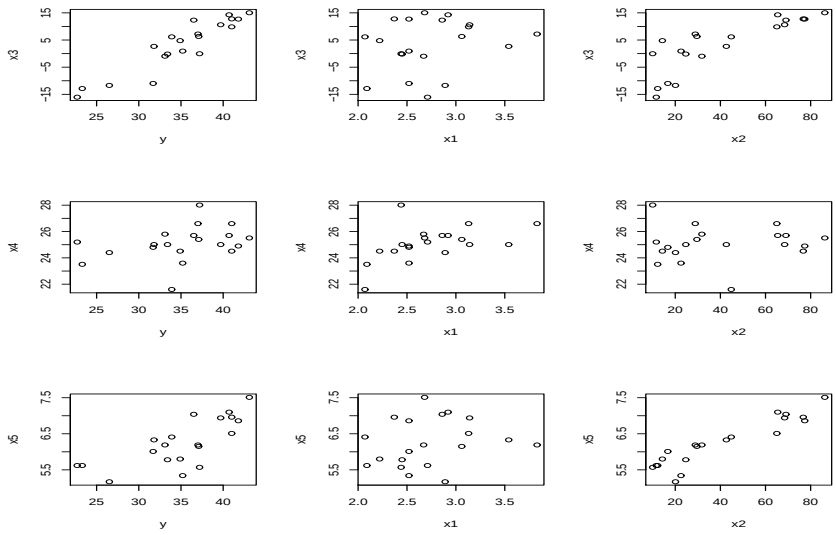

Fig. 1.3 Scatterplot matrix for *Coleman Report* data.

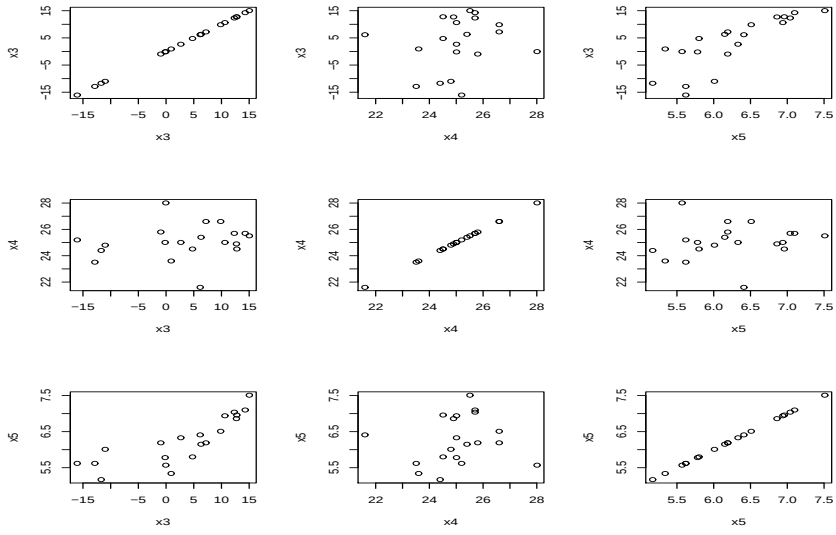

Fig. 1.4 Scatterplot matrix for *Coleman Report* data.

However, the explanatory powers of x_1 and x_4 only manifest themselves after x_3 has been fitted to the data.

1.2 Inferential Procedures

A basic linear model for the *Colman Report* data that does not involve transforming the predictors is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i, \tag{1}$$

$i = 1, \ldots, 20$, where the ε_is are unobservable independent $N(0, \sigma^2)$ random variables and the βs are fixed unknown parameters. Fitting Model (1) with a computer program typically yields a table of coefficients with parameter estimates, standard errors for the estimates, t ratios for testing whether the parameters are zero, P values, and a three line analysis of variance table.

Table of Coefficients: Model (1)

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	19.95	13.63	1.46	0.165
x_1	−1.793	1.233	−1.45	0.168
x_2	0.04360	0.05326	0.82	0.427
x_3	0.55576	0.09296	5.98	0.000
x_4	1.1102	0.4338	2.56	0.023
x_5	−1.811	2.027	−0.89	0.387

Analysis of Variance: Model (1)

Source	df	SS	MS	F	P
Regression	5	582.69	116.54	27.08	0.000
Error	14	60.24	4.30		
Total	19	642.92			

From just these two tables of statistics much can be learned. In particular, the estimated regression equation is

$$\hat{y} = 19.9 - 1.79x_1 + 0.0436x_2 + 0.556x_3 + 1.11x_4 - 1.81x_5.$$

Substituting the observed values x_{ij}, $j = 1, \ldots, 5$ for the x_js in the estimated regression equation gives the fitted (predicted) values $\hat{y}_i$ and the residuals $\hat{\varepsilon}_i = y_i - \hat{y}_i$.

The estimated regression equation *describes* the relationship between y and the predictor variables for the *current* data; *it does not imply a causal relationship*. If we go out and increase the percentage of sixth graders whose fathers have white-collar jobs by 1%, i.e., increase x_2 by one unit, we cannot infer that mean verbal test scores will tend to increase by 0.0436 units. In fact, we cannot think about any of the variables in a vacuum. No variable has an effect in the equation apart from the *observed values* of all the other variables. If we conclude that some variable can

be eliminated from the model, we cannot conclude that the variable has no effect on y, we can only conclude that the variable is not necessary to explain *these* data. The same variable may be very important in explaining other, rather different, data collected on the same variables. All too often, people choose to interpret the estimated regression coefficients as if the predictor variables cause the value of y but the estimated regression coefficients simply describe an observed relationship. Frankly, since the coefficients do not describe a causal relationship, many people, including the author, find regression coefficients to be remarkably uninteresting quantities. What this model is good at is predicting values of y for new cases that are similar to those in the current data. In particular, such new cases should have predictor variables with values similar to those in the current data.

The t statistics for testing $H_0 : \beta_k = 0$ were reported in the table of coefficients. For example, the test of $H_0 : \beta_4 = 0$ has

$$t_{obs} = \frac{\hat{\beta}_4}{\mathrm{SE}(\hat{\beta}_4)} = \frac{1.1102}{0.4338} = 2.56.$$

The P value is

$$P = \Pr[|t(dfE)| \geq 2.56] = 0.023.$$

The value 0.023 indicates a reasonable amount of evidence that variable x_4 is needed in the model. We can be reasonably sure that dropping x_4 from the model harms the explanatory (predictive) power of the model. In particular, with a P value of 0.023, the test of the null model with $H_0 : \beta_4 = 0$ is rejected at the $\alpha = 0.05$ level (because $0.05 > 0.023$), but the test is not rejected at the $\alpha = 0.01$ level (because $0.023 > 0.01$).

A 95% confidence interval for β_3 has endpoints $\hat{\beta}_3 \pm t(0.975, dfE)\,\mathrm{SE}(\hat{\beta}_3)$. From a t table, $t(0.975, 14) = 2.145$ and from the table of coefficients the endpoints are

$$0.55576 \pm 2.145(0.09296).$$

The confidence interval is $(0.356, 0.755)$, so the data are consistent with β_3 being between 0.356 and 0.755.

The primary value of the analysis of variance table is that it gives the degrees of freedom, the sum of squares, and the mean square for error. The mean squared error is the estimate of σ^2, and the sum of squares error and degrees of freedom for error are vital for comparing various regression models. The degrees of freedom for error are $n - 1 -$ (the number of predictor variables). The minus 1 is an adjustment for fitting the intercept β_0.

The analysis of variance table also gives the test for whether any of the x variables help to explain y, i.e., of whether $y_i = \beta_0 + \varepsilon_i$ is an adequate model. This test is rarely of interest because it is almost always highly significant. It is a poor scholar who cannot find any predictor variables that are related to the measurement of primary interest. (Ok, I admit to being a little judgmental here.) The test of

$$H_0 : \beta_1 = \cdots = \beta_5 = 0$$

is based on

$$F_{obs} = \frac{MSReg}{MSE} = \frac{116.5}{4.303} = 27.08$$

and (typically) is rejected for large values of F. The numerator and denominator degrees of freedom come from the ANOVA table. As suggested, the corresponding P value in the ANOVA table is infinitesimal, i.e., zero to three decimal places. Thus these x variables, as a group, help to explain the variation in the y variable. In other words, it is possible to predict the mean verbal test scores for a school's sixth grade class from the five x variables measured. Of course, the fact that some predictive ability exists does not mean that the predictive ability is sufficient to be useful.

The *coefficient of determination*, R^2, measures the predictive ability of the model. It is the squared correlation between the $(\hat{y}_i, y_i)$ pairs and also is the percentage of the total variability in y that is explained by the x variables. If this number is large, it suggests a substantial predictive ability. In this example

$$R^2 \equiv \frac{SSReg}{SSTot} = \frac{582.69}{642.92} = 0.906,$$

so 90.6% of the total variability is explained by the regression model. This large percentage suggests that the five x variables have substantial predictive power. However, we will see that a large R^2 does not imply that the model is good in absolute terms. It may be possible to show that this model does not fit the data adequately. In other words, this model is explaining much of the variability but we may be able to establish that it is not explaining as much of the variability as it ought. Conversely, a model with a low R^2 value may be the perfect model but the data may simply have a great deal of variability. Moreover, even an R^2 of 0.906 may be inadequate for the predictive purposes of the researcher, while in some situations an R^2 of 0.3 may be perfectly adequate. It depends on the purpose of the research. Finally, a large R^2 may be just an unrepeatable artifact of a particular data set. The coefficient of determination is a useful tool but it must be used with care. When using just x_3 to predict y, R^2 is the square of the correlation between the two variables, so $R^2 = (0.927)^2 = 0.86$.

1.2.1 Computing commands

Performing multiple regression without a computer program is impractical. Mintab's reg command is menu driven, hence very easy to use. SAS's regression procedures are a bit more complicated, but example commands are easily followed, as are the commands for Minitab, most of which can be avoided by using the menus. (Minitab and SAS code for *ANREG-II* can be found at `http://www.stat.unm.edu/~fletcher/MinitabCode.pdf`.) I have little personal experience with programs like SPSS, SYSTAT, and JMP. R, on the other hand, is not a program but a programming language and is more complicated to use. Because multiple linear regression is the fundamental model considered in this book, we present some R code for it. The link to R code for this book was given in the Preface.

The following R code should work for computing most of the statistics used in this chapter and the next. Of course you have to replace the location of the data file `C:\\tab1-1.dat` with the location where you stored the data. For completeness, this code includes some procedures not yet discussed but that should have been discussed in a first course on regression.

```
coleman <- read.table("C:\\tab1-1.dat",sep="",
      col.names=c("School","x1","x2","x3","x4","x5","y"))
attach(coleman)
coleman
summary(coleman)

#Coefficient and ANOVA tables
co <- lm(y ~ x1+x2+x3+x4+x5)
cop=summary(co)
cop
anova(co)

#Confidence intervals
confint(co, level=0.95)

#Predictions
new = data.frame(x1=2.07, x2=9.99,x3=-16.04,x4= 21.6, x5=5.17)
predict(co,new,se.fit=T,interval="confidence")
predict(co,new,interval="prediction")

# Diagnostics table
infv = c(y,co$fit,hatvalues(co),rstandard(co),rstudent(co),
                                              cooks.distance(co))
inf=matrix(infv,I(cop$df[1]+cop$df[2]),6,dimnames =
                  list(NULL,c("y", "yhat", "lev","r","t","C")))
inf

# Normal and fitted values plots
qqnorm(rstandard(co),ylab="Standardized residuals")
plot(co$fit,rstandard(co),xlab="Fitted",
     ylab="Standardized residuals",main="Residual-Fitted plot")

#Wilk-Francia Statistic
rankit=qnorm(ppoints(rstandard(co),a=I(3/8)))
ys=sort(rstandard(co))
Wprime=(cor(rankit,ys))^2
Wprime
```

1.3 General Statement of Model

In general we consider a dependent variable y that is a random variable of interest. We also consider $p-1$ nonrandom predictor variables $x_1,\ldots,x_{p-1}$. These can be original measurements or transformations of original measurements. The general multiple (linear) regression model relates n observations on y to a linear combination

of the corresponding observations on the x_js plus a random error ε. In particular, we assume

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{p-1} x_{i,p-1} + \varepsilon_i,$$

where the subscript $i = 1, \ldots, n$ indicates different observations and the ε_is are independent $N(0, \sigma^2)$ random variables. The β_js and σ^2 are unknown constants and are the fundamental parameters of the regression model.

Estimates of the β_js are obtained by the method of least squares. The least squares estimates are those that minimize

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \cdots - \beta_{p-1} x_{i,p-1})^2 .$$

In this function the y_is and the x_{ij}s are all known quantities. Least squares estimates have a number of interesting statistical properties. If the errors are independent with mean zero, constant variance, and are normally distributed, the least squares estimates are maximum likelihood estimates (MLEs) and minimum variance unbiased estimates (MVUEs). If we keep the assumptions of mean zero and constant variance but weaken the independence assumption to that of the errors being merely uncorrelated and stop assuming normal distributions, the least squares estimates are best (minimum variance) linear unbiased estimates (BLUEs). For proofs of these statements, see *PA*.

In checking assumptions we often use the predictions (fitted values) $\hat{y}$ corresponding to the observed values of the predictor variables, i.e.,

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_{p-1} x_{i,p-1},$$

$i = 1, \ldots, n$. Residuals are the values

$$\hat{\varepsilon}_i = y_i - \hat{y}_i.$$

The other fundamental parameter to be estimated, besides the β_js, is the variance σ^2. The *sum of squares error* is

$$SSE = \sum_{i=1}^{n} \hat{\varepsilon}_i^2$$

and the estimate of σ^2 is the *mean squared error* (*residual mean square*)

$$MSE = SSE/(n-p).$$

The *MSE* is an unbiased estimate of σ^2 in that $\mathrm{E}(MSE) = \sigma^2$. Under the standard normality assumptions, *MSE* is the minimum variance unbiased estimate of σ^2. However, the maximum likelihood estimate of σ^2 is $\hat{\sigma}^2 = SSE/n$, We will never use the MLE of σ^2. (Some programs for fitting generalized linear models, when used to fit standard linear models, report the MLE rather than the *MSE*.)

Details of the estimation procedures are given in Chapter 2.

1.4 Regression Surfaces and Prediction

One of the most valuable aspects of regression analysis is its ability to provide good predictions of future observations. Of course, to obtain a prediction for a new value y we need to know the corresponding values of the predictor variables, the x_js. Moreover, to obtain good predictions, the values of the x_js need to be similar to those on which the regression model was fitted. Typically, a fitted regression model is only an approximation to the true relationship between y and the predictor variables. These approximations can be very good, but, because they are only approximations, they are not valid for predictor variables that are dissimilar to those on which the approximation was based. Trying to predict for x_j values that are far from the original data is always difficult. Even if the regression model is true and not an approximation, the variance of such predictions is large. When the model is only an approximation, the approximation is typically invalid for such predictor variables and the predictions can be utter nonsense.

The regression surface for the Coleman data is the set of all values z that satisfy

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5$$

for some values of the predictor variables. The estimated regression surface is

$$z = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_3 + \hat{\beta}_4 x_4 + \hat{\beta}_5 x_5.$$

There are two problems of interest. The first is estimating the value z on the regression surface for a fixed set of predictor variables. The second is predicting the value of a new observation to be obtained with a fixed set of predictor variables. For any set of predictor variables, the estimate of the regression surface and the prediction are identical. What differs are the standard errors associated with the different problems.

Consider estimation and prediction at

$$(x_1, x_2, x_3, x_4, x_5) = (2.07, 9.99, -16.04, 21.6, 5.17).$$

These are the minimum values for each of the variables, so there will be substantial variability in estimating the regression surface at this point. The estimator (predictor) is

$$\begin{aligned}\hat{y} = \hat{\beta}_0 + \sum_{j=1}^{5} \hat{\beta}_j x_j &= 19.9 - 1.79(2.07) + 0.0436(9.99) \\ &\quad + 0.556(-16.04) + 1.11(21.6) - 1.81(5.17) = 22.375.\end{aligned}$$

For constructing 95% t intervals, the percentile needed is $t(0.975, 14) = 2.145$.

The 95% confidence interval for the point $\beta_0 + \sum_{j=1}^{5} \beta_j x_j$ on the regression surface uses the standard error for the regression surface, which is

$$\text{SE}(Surface) = 1.577.$$

The standard error is obtained from the regression program and depends on the specific value of $(x_1, x_2, x_3, x_4, x_5)$. The formula for the standard error is given in Section 2.4. This interval has endpoints

$$22.375 \pm 2.145(1.577),$$

which gives the interval

$$(18.992, 25.757).$$

The 95% prediction interval is

$$(16.785, 27.964).$$

This is about 4 units wider than the confidence interval for the regression surface. The standard error for the prediction interval can be computed from the standard error for the regression surface.

$$\text{SE}(Prediction) = \sqrt{MSE + SE(Surface)^2}.$$

In this example,

$$\text{SE}(Prediction) = \sqrt{4.303 + (1.577)^2} = 2.606,$$

and the prediction interval endpoints are

$$22.375 \pm 2.145(2.606).$$

We mentioned earlier that even if the regression model is true, the variance of predictions is large when the x_j values for the prediction are far from the original data. We can use this fact to identify situations in which the predictions are unreliable because the locations are too far away. Let $p-1$ be the number of predictor variables so that, including the intercept, there are p regression parameters. Let n be the number of observations. A sensible rule of thumb is that we should start worrying about the validity of the prediction whenever

$$\frac{\text{SE}(Surface)}{\sqrt{MSE}} \geq \sqrt{\frac{2p}{n}}$$

and we should be very concerned about the validity of the prediction whenever

$$\frac{\text{SE}(Surface)}{\sqrt{MSE}} \geq \sqrt{\frac{3p}{n}}.$$

Recall from regression analysis that the leverage for a case is a number between 0 and 1 that measures the distance between the predictor variables for the case in question and the average of the predictor variables from the entire data. Leverages greater than $2p/n$ and $3p/n$ cause similar levels of concern to those mentioned in the previous paragraph. We are comparing $\mathrm{SE}(Surface)/\sqrt{MSE}$ to the square roots of these guidelines because, for cases in the data, the ratio in question is the square root of the leverage. In our example, $p = 6$ and $n = 20$, so

$$\frac{\mathrm{SE}(Surface)}{\sqrt{MSE}} = \frac{1.577}{\sqrt{4.303}} = 0.760 < 0.775 = \sqrt{\frac{2p}{n}}.$$

The location of this prediction is near the boundary of those locations for which we feel comfortable making predictions.

1.5 Comparing Regression Models

A frequent goal in regression analysis is to find the simplest model that provides an adequate explanation of the data. In examining the full model with all five x variables, there is little evidence that any of x_1, x_2, or x_5 are needed in the regression model. The t tests reported in Section 1.2 for the corresponding regression parameters gave P values of 0.168, 0.427, and 0.387. We could drop *any one* of the three variables without significantly harming the model. While this does not imply that all three variables can be dropped without harming the model, dropping the three variables makes an interesting point of departure.

Fitting the reduced model

$$y_i = \beta_0 + \beta_3 x_{i3} + \beta_4 x_{i4} + \varepsilon_i$$

gives

Table of Coefficients

Predictor	$\hat{\beta}_k$	$\mathrm{SE}(\hat{\beta}_k)$	t	P
Constant	14.583	9.175	1.59	0.130
x_3	0.54156	0.05004	10.82	0.000
x_4	0.7499	0.3666	2.05	0.057

Analysis of Variance

Source	df	SS	MS	F	P
Regression	2	570.50	285.25	66.95	0.000
Error	17	72.43	4.26		
Total	19	642.92			

We can test whether this reduced model is an adequate explanation of the data as compared to the full model. The sum of squares for error from the full model was reported in Section 1.2 as $SSE(Full) = 60.24$ with degrees of freedom $dfE(Full) =$

14 and mean squared error $MSE(Full) = 4.30$. For the reduced model we have $SSE(Red.) = 72.43$ and $dfE(Red.) = 17$. The test statistic for the adequacy of the reduced model is

$$F_{obs} = \frac{[SSE(Red.) - SSE(Full)]/[dfE(Red.) - dfE(Full)]}{MSE(Full)}$$
$$= \frac{[72.43 - 60.24]/[17 - 14]}{4.30} = 0.94.$$

F has $[dfE(Red.) - dfE(Full)]$ and $dfE(Full)$ degrees of freedom in the numerator and denominator, respectively. Here F is about 1, so it is not significant. In particular, 0.94 is less than $F(0.95, 3, 14)$, so a formal $\alpha = 0.05$ level one-sided F test does not reject the adequacy of the reduced model. In other words, the 0.05 level one-sided test of the null model with $H_0 : \beta_1 = \beta_2 = \beta_5 = 0$ is not rejected. (I also ignored the fact that I looked at the data and let that guide my choice of a reduced model. It is not surprising when dropping variables that seem unimportant turns out to be ok but the criterion that guided my choice did not assure that the variables would be unimportant.)

This test lumps the three variables x_1, x_2, and x_5 together into one big test. It is possible that the uselessness of two of these variables could hide the fact that one of them is (marginally) significant when added to the model with x_3 and x_4. To fully examine this possibility, we need to fit three additional models. Each variable should be added, in turn, to the model with x_3 and x_4. We consider in detail only one of these three models, the model with x_1, x_3, and x_4. From fitting this model, the t statistic for testing whether x_1 is needed in the model turns out to be -1.47. This has a P value of 0.162, so there is little indication that x_1 is useful. We could also construct an F statistic as illustrated previously. The sum of squares for error in the model with x_1, x_3, and x_4 is 63.84 on 16 degrees of freedom, so

$$F_{obs} = \frac{[72.43 - 63.84]/[17 - 16]}{63.84/16} = 2.16.$$

Note that, up to round-off error, $F = t^2$. The tests are equivalent and the P value for the F statistic is also 0.162. F tests are only equivalent to a corresponding t test when the numerator of the F statistic has one degree of freedom. Methods similar to these establish that neither x_2 nor x_5 are important when added to the model that contains x_3 and x_4.

Here we are testing two models: the full model with x_1, x_3, and x_4 against a reduced model with only x_3 and x_4. Both of these models are special cases of a biggest model that contains all of x_1, x_2, x_3, x_4, and x_5. In *ANREG-II* Subsection 3.1.1, for cases like this, we recommended an alternative F statistic,

$$F_{obs} = \frac{[72.43 - 63.84]/[17 - 16]}{4.30} = 2.00,$$

where the denominator MSE of 4.30 comes from the biggest model (as would the denominator degrees of freedom).

In testing the reduced model with only x_3 and x_4 against the full five-variable model, we observed that one might miss recognizing a variable that was (marginally) significant. In this case we did not miss anything important. However, if we had taken the reduced model as containing only x_3 and tested it against the full five-variable model, we would have missed the importance of x_4. The F statistic for this test turns out to be only 1.74.

In the model with x_1, x_3, and x_4, the t test for x_4 turns out to have a P value of 0.021. As seen in the table given previously, if we drop x_1 and use the model with only x_3, and x_4, the P value for x_4 goes to 0.057. Thus dropping a weak variable, x_1, can make a reasonably strong variable, x_4, look weaker. There is a certain logical inconsistency here. If x_4 is important in the x_1, x_3, x_4 model or the full five-variable model (P value 0.023), it is illogical that dropping some of the other variables could make it unimportant. Even though x_1 is not particularly important by itself, it augments the evidence that x_4 is useful. The problem in these apparent inconsistencies is that the x variables are all related to each other; this is known as the problem of *collinearity*. One reason for using the alternative F tests that employ $MSE(Big.)$ in the denominator is that it ameliorates this phenomenon.

Although a reduced model may be an adequate substitute for a full model on a particular set of data, it does *not* follow that the reduced model will be an adequate substitute for the full model with any data collected on the variables in the full model.

1.5.1 General discussion

Suppose that we want to compare two regression models, say,

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{q-1} x_{i,q-1} + \cdots + \beta_{p-1} x_{i,p-1} + \varepsilon_i \tag{1}$$

and

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_{q-1} x_{i,q-1} + \varepsilon_i. \tag{2}$$

The key fact here is that all of the variables in Model (2) are also in Model (1). In this comparison, we dropped the last variables $x_{i,q}, \ldots, x_{i,p-1}$ for notational convenience only; the discussion applies to dropping any group of variables from Model (1). *Throughout, we assume that Model (1) gives an adequate fit to the data and then compare how well Model (2) fits the data with how well Model (1) fits*. Before applying the results of this subsection, the validity of the model (1) assumptions should be evaluated.

We want to know if the variables $x_{i,q}, \ldots, x_{i,p-1}$ are needed in the model, i.e., whether they are useful predictors. In other words, we want to know if Model (2) is an adequate model; whether it gives an adequate explanation of the data. The variables $x_q, \ldots, x_{p-1}$ are extraneous if and only if $\beta_q = \cdots = \beta_{p-1} = 0$. The test we

develop can be considered as a test of

$$H_0 : \beta_q = \cdots = \beta_{p-1} = 0.$$

Parameters are very tricky things; you never get to see the value of a parameter. I strongly prefer the interpretation of testing one model against another model rather than the interpretation of testing whether $\beta_q = \cdots = \beta_{p-1} = 0$. In practice, useful regression models are rarely correct models, although they can be *very* good approximations. Typically, we do not really care whether Model (1) is true, only whether it is useful, but dealing with parameters in an incorrect model becomes tricky.

In practice, we are looking for a (relatively) succinct way of summarizing the data. The smaller the model, the more succinct the summarization. However, we do not want to eliminate useful explanatory variables, so we test the smaller (more succinct) model against the larger model to see if the smaller model gives up significant explanatory power. Note that the larger model always has at least as much explanatory power as the smaller model because the larger model includes all the variables in the smaller model plus some more.

Applying model testing procedures to this problem yields the following test: Reject the hypothesis

$$H_0 : \beta_q = \cdots = \beta_{p-1} = 0$$

at the α level if

$$F \equiv \frac{[SSE(Red.) - SSE(Full)] \big/ (p-q)}{MSE(Full)} > F(1-\alpha, p-q, n-p).$$

The notation $SSE(Red.) - SSE(Full)$ focuses on the ideas of full and reduced models. Other notations that focus on variables and parameters are also commonly used. One can view the model comparison procedure as fitting Model (2) first and then seeing how much better Model (1) fits. The notation based on this refers to the (extra) *sum of squares for regressing* on $x_q, \ldots, x_{p-1}$ *after* regressing on $x_1, \ldots, x_{q-1}$ and is written

$$SSR(x_q, \ldots, x_{p-1} | x_1, \ldots, x_{q-1}) \equiv SSE(Red.) - SSE(Full).$$

This notation assumes that the model contains an intercept. Alternatively, one can think of fitting the parameters $\beta_q, \ldots, \beta_{p-1}$ after fitting the parameters $\beta_0, \ldots, \beta_{q-1}$. The relevant notation refers to the *reduction in sum of squares* (for error) due to fitting $\beta_q, \ldots, \beta_{p-1}$ after $\beta_0, \ldots, \beta_{q-1}$ and is written

$$R(\beta_q, \ldots, \beta_{p-1} | \beta_0, \ldots, \beta_{q-1}) \equiv SSE(Red.) - SSE(Full).$$

Note that it makes perfect sense to refer to $SSR(x_q, \ldots, x_{p-1} | x_1, \ldots, x_{q-1})$ as the reduction in sum of squares for fitting $x_q, \ldots, x_{p-1}$ after $x_1, \ldots, x_{q-1}$.

It was mentioned earlier that the degrees of freedom for $SSE(Red.) - SSE(Full)$ is $p-q$. Note that $p-q$ is the number of variables to the left of the vertical bar

in $SSR(x_q,\ldots,x_{p-1}|x_1,\ldots,x_{q-1})$ and the number of parameters to the left of the vertical bar in $R(\beta_q,\ldots,\beta_{p-1}|\beta_0,\ldots,\beta_{q-1})$.

A point that is quite clear when thinking of model comparisons is that if you change either model, (1) or (2), the test statistic and thus the test changes. This point continues to be clear when dealing with the notations $SSR(x_q,\ldots,x_{p-1}|x_1,\ldots,x_{q-1})$ and $R(\beta_q,\ldots,\beta_{p-1}|\beta_0,\ldots,\beta_{q-1})$. If you change any variable on either side of the vertical bar, you change $SSR(x_q,\ldots,x_{p-1}|x_1,\ldots,x_{q-1})$. Similarly, the parametric notation $R(\beta_q,\ldots,\beta_{p-1}|\beta_0,\ldots,\beta_{q-1})$ is also perfectly precise, but confusion can easily arise when dealing with parameters if one is not careful. For example, when testing, say, $H_0:\ \beta_1=\beta_3=0$, the tests are completely different in the three models

$$y_i=\beta_0+\beta_1x_{i1}+\beta_3x_{i3}+\varepsilon_i, \tag{3}$$

$$y_i=\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\beta_3x_{i3}+\varepsilon_i, \tag{4}$$

and

$$y_i=\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\beta_3x_{i3}+\beta_4x_{i4}+\varepsilon_i\ . \tag{5}$$

In Model (3) the test is based on $SSR(x_1,x_3)\equiv R(\beta_1,\beta_3|\beta_0)$, i.e., the sum of squares for regression (*SSReg*) in the model with only x_1 and x_3 as predictor variables. In Model (4) the test uses

$$SSR(x_1,x_3|x_2)\equiv R(\beta_1,\beta_3|\beta_0,\beta_2).$$

Model (5) uses $SSR(x_1,x_3|x_2,x_4)\equiv R(\beta_1,\beta_3|\beta_0,\beta_2,\beta_4)$. In all cases we are testing $\beta_1=\beta_3=0$ *after* fitting all the other parameters in the model. In general, we think of testing $H_0:\ \beta_q=\cdots=\beta_{p-1}=0$ after fitting $\beta_0,\ldots,\beta_{q-1}$.

If the reduced model is obtained by dropping out only one variable, e.g., if $q-1=p-2$, the parametric hypothesis is $H_0:\ \beta_{p-1}=0$. We have just developed an F test for this and we have earlier used a t test for the hypothesis. In multiple regression the F test is equivalent to the t test. It follows that the t test must be considered as a test for the parameter *after fitting all* of the other parameters in the model. In particular, the t tests reported in the table of coefficients when fitting a regression tell you only whether a variable can be dropped relative to the model that contains all the other variables. These t tests cannot tell you whether more than one variable can be dropped from the fitted model. If you drop any variable from a regression model, all of the t tests change. It is only for notational convenience that we are discussing testing $\beta_{p-1}=0$; the results hold for any β_k.

The *SSR* notation can also be used to find *SSE*s. Consider models (3), (4), and (5) and suppose we know $SSR(x_2|x_1,x_3)$, $SSR(x_4|x_1,x_2,x_3)$, and the *SSE* from Model (5). We can easily find the *SSE*s for models (3) and (4). By definition,

$$\begin{aligned} SSE(4) &= [SSE(4)-SSE(5)]+SSE(5)\\ &= SSR(x_4|x_1,x_2,x_3)+SSE(5). \end{aligned}$$

Also

$$\begin{aligned} SSE(3) &= [SSE(3) - SSE(4)] + SSE(4) \\ &= SSR(x_2|x_1,x_3) + \{SSR(x_4|x_1,x_2,x_3) + SSE(5)\}. \end{aligned}$$

Moreover, we see that

$$\begin{aligned} SSR(x_2,x_4|x_1,x_3) &= SSE(3) - SSE(5) \\ &= SSR(x_2|x_1,x_3) + SSR(x_4|x_1,x_2,x_3). \end{aligned}$$

Note also that we can change the order of the variables.

$$SSR(x_2,x_4|x_1,x_3) = SSR(x_4|x_1,x_3) + SSR(x_2|x_1,x_3,x_4).$$

1.6 Sequential Fitting

Multiple regression analysis is largely impractical without the aid of a computer. One specifies a regression model and the computer returns the vital statistics for that model. Many computer programs actually fit a sequence of models rather than fitting the model all at once.

EXAMPLE 1.6.1. Suppose you want to fit the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \varepsilon_i.$$

Many regression programs actually fit the sequence of models

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \varepsilon_i, \\ y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i, \\ y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i, \\ y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \varepsilon_i. \end{aligned}$$

The sequence is determined by the order in which the variables are specified. If the identical model is specified in the form

$$y_i = \beta_0 + \beta_3 x_{i3} + \beta_1 x_{i1} + \beta_4 x_{i4} + \beta_2 x_{i2} + \varepsilon_i,$$

the end result is exactly the same but the sequence of models is

$$\begin{aligned} y_i &= \beta_0 + \beta_3 x_{i3} + \varepsilon_i, \\ y_i &= \beta_0 + \beta_3 x_{i3} + \beta_1 x_{i1} + \varepsilon_i, \\ y_i &= \beta_0 + \beta_3 x_{i3} + \beta_1 x_{i1} + \beta_4 x_{i4} + \varepsilon_i, \\ y_i &= \beta_0 + \beta_3 x_{i3} + \beta_1 x_{i1} + \beta_4 x_{i4} + \beta_2 x_{i2} + \varepsilon_i. \end{aligned}$$

Frequently, programs that fit sequences of models also provide sequences of sums of squares. Thus the first sequence of models yields

$$SSR(x_1),\ SSR(x_2|x_1),\ SSR(x_3|x_1,x_2),\ \text{and}\ SSR(x_4|x_1,x_2,x_3)$$

while the second sequence yields

$$SSR(x_3),\ SSR(x_1|x_3),\ SSR(x_4|x_3,x_1),\ \text{and}\ SSR(x_2|x_3,x_1,x_4).$$

These can be used in a variety of ways. For example, as shown at the end of the previous section, to test

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_3 x_{i3} + \varepsilon_i$$

against

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \varepsilon_i$$

we need $SSR(x_2,x_4|x_3,x_1)$. This is easily obtained from the second sequence as

$$SSR(x_2,x_4|x_3,x_1) = SSR(x_4|x_3,x_1) + SSR(x_2|x_3,x_1,x_4). \qquad \square$$

EXAMPLE 1.6.2. If we fit the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i$$

to the *Coleman Report* data, we get the sequential sums of squares listed below.

Source	*df*	*Seq SS*	Notation
x_1	1	23.77	$SSR(x_1)$
x_2	1	343.23	$SSR(x_2\|x_1)$
x_3	1	186.34	$SSR(x_3\|x_1,x_2)$
x_4	1	25.91	$SSR(x_4\|x_1,x_2,x_3)$
x_5	1	3.43	$SSR(x_5\|x_1,x_2,x_3,x_4)$

Recall that the *MSE* for the five-variable model is 4.30 on 14 degrees of freedom.

From the sequential sums of squares we can test a variety of hypotheses related to the full model. For example, we can test whether variable x_5 can be dropped from the five-variable model. The F statistic is $3.43/4.30$, which is less than 1, so the effect of x_5 is insignificant. This test is equivalent to the t test for x_5 given in Section 2 when fitting the five-variable model. We can also test whether we can drop both x_4 and x_5 from the full model. The F statistic is

$$F_{obs} = \frac{(25.91 + 3.43)/2}{4.30} = 3.41.$$

$F(0.95,2,14) = 3.74$, so this F statistic provides little evidence that the pair of variables is needed. (The relative importance of x_4 is somewhat hidden by combining

it in a test with the unimportant x_5.) Similar tests can be constructed for dropping x_3, x_4, and x_5, for dropping x_2, x_3, x_4, and x_5, and for dropping x_1, x_2, x_3, x_4, and x_5 from the full model. The last of these is just the ANOVA table F test.

We can also make a variety of tests related to 'full' models that do not include all five variables. In the previous paragraph, we found little evidence that the pair x_4 and x_5 help explain the data in the five-variable model. We now test whether x_4 can be dropped when we have already dropped x_5. In other words, we test whether x_4 adds explanatory power to the model that contains x_1, x_2, and x_3. The numerator has one degree of freedom and is $SSR(x_4|x_1,x_2,x_3) = 25.91$. The usual denominator mean square for this test is the *MSE* from the model with x_1, x_2, x_3, and x_4, i.e., $\{14(4.303)+3.43\}/15$. (For numerical accuracy we have added another significant digit to the *MSE* from the five-variable model. The *SSE* from the model without x_5 is just the *SSE* from the five-variable model plus the sequential sum of squares $SSR(x_5|x_1,x_2,x_3,x_4)$.) Our best practice would be to construct the test using the same numerator mean square but the *MSE* from the five-variable model in the denominator of the test. Using this second denominator, the F statistic is $25.91/4.30 = 6.03$. Corresponding F percentiles are $F(0.95,1,14) = 4.60$ and $F(0.99,1,14) = 8.86$, so x_4 may be contributing to the model. If we had used the *MSE* from the model with x_1, x_2, x_3, and x_4, the F statistic would be equivalent to the t statistic for dropping x_4 that is obtained when fitting this four-variable model.

If we wanted to test whether x_2 and x_3 can be dropped from the model that contains x_1, x_2, and x_3, the usual denominator is $[14(4.303)+25.91+3.43]/16 = 5.60$. (The *SSE* for the model without x_4 or x_5 is just the *SSE* from the five-variable model plus the sequential sum of squares for x_4 and x_5.) Again, we would alternatively use the *MSE* from the five-variable model in the denominator. Using the first denominator, the test is

$$F_{obs} = \frac{(343.23+186.34)/2}{5.60} = 47.28,$$

which is much larger than $F(0.999,2,16) = 10.97$, so there is overwhelming evidence that variables x_2 and x_3 cannot be dropped from the x_1, x_2, x_3 model.

The argument for basing tests on the *MSE* from the five-variable model is that it is less subject to bias than the other *MSE*s. In the test given in the previous paragraph, the *MSE* from the usual 'full' model incorporates the sequential sums of squares for x_4 and x_5. A reason for doing this is that we have tested x_4 and x_5 and are not convinced that they are important. As a result, their sums of squares are incorporated into the error. Even though we may not have established an overwhelming case for the importance of either variable, there is some evidence that x_4 is a useful predictor when added to the first three variables. The sum of squares for x_4 may or may not be large enough to convince us of its importance but it is large enough to change the *MSE* from 4.30 in the five-variable model to 5.60 in the x_1, x_2, x_3 model. In general, if you test terms and pool them with the Error whenever the test is insignificant, you are biasing the *MSE* that results from this pooling. □

In general, *when given the ANOVA table and the sequential sums of squares, we can test any model in the sequence against any reduced model that is part of the*

sequence. We cannot *use these statistics to obtain a test involving a model that is not part of the sequence*.

1.7 Reduced Models and Prediction

Fitted regression models are, not surprisingly, very dependent on the observed values of the predictor variables. We have already discussed the fact that fitted regression models are particularly good for making predictions but only for making predictions on new cases with predictor variables that are similar to those used in fitting the model. Fitted models are not good at predicting observations with predictor variable values that are far from those in the observed data. We have also discussed the fact that in evaluating a reduced model we are evaluating whether the reduced model is an adequate explanation of the data. An adequate reduced model should serve well as a prediction equation but only for new cases with predictor variables similar to those in the original data. It should not be overlooked that *when using a reduced model for prediction, new cases need to be similar to the observed data on* all *predictor variables and not just on the predictor variables in the reduced model*.

Good prediction from reduced models requires that new cases be similar to observed cases on all predictor variables because of the process of selecting reduced models. Predictor variables are eliminated from a model if they are not necessary to explain the data. This can happen in two ways. If a predictor variable is truly unrelated to the dependent variable, it is both proper and beneficial to eliminate that variable. The other possibility is that a predictor variable may be related to the dependent variable but that the relationship is hidden by the nature of the observed predictor variables. In the *Coleman Report* data, suppose the true response depends on both x_3 and x_5. We know that x_3 is clearly the best single predictor but the observed values of x_5 and x_3 are closely related; the sample correlation between them is 0.819. Because of their high correlation *in these data*, much of the actual dependence of y on x_5 could be accounted for by the regression on x_3 alone. Variable x_3 acts as a surrogate for x_5. As long as we try to predict new cases that have values of x_5 and x_3 similar to those in the original data, a reduced model based on x_3 should work well. Variable x_3 should continue to act as a surrogate. On the other hand, if we tried to predict a new case that had an x_3 value similar to that in the observed data but where the pair x_3, x_5 was not similar to x_3, x_5 pairs in the observed data, the reduced model that uses x_3 as a surrogate for x_5 would be inappropriate. Predictions could be very bad and, if we thought only about the fact that the x_3 value is similar to those in the original data, we might expect the predictions to be good. Unfortunately, when we eliminate a variable from a regression model, we typically have no idea if it is eliminated because the variable really has no effect on y or because its effect is being masked by some other set of predictor variables. For further discussion of these issues see Mandel (1989a, b).

Of course there is reason to hope that predictions will typically work well for reduced models. If the data come from an observational study in which the cases are

some kind of sample from a population, there is reason to expect that future cases that are *sampled in the same way* will behave similarly to those in the original study. In addition, if the data come from an experiment in which the predictor variables are under the control of the investigator, it is reasonable to expect the investigator to select values of the predictor variables that cover the full range over which predictions will be made. Nonetheless, regression models give good approximations and good predictions only within the range of the observed data and, when a reduced model is used, the definition of the range of the observed data includes the values of all predictor variables that were in the full model. In fact, *even this statement is too weak*. When using a reduced model or even when using the full model for prediction, *new cases need to be similar to the observed cases in all relevant ways*. If there is some unmeasured predictor that is related to y and if the observed predictors are highly correlated with this unmeasured variable, then for good prediction a new case needs to have a value of the unmeasured variable that is similar to those for the observed cases. In other words, *the variables in any model may be acting as surrogates for some unmeasured variables and to obtain good predictions the new cases must be similar on both the observed predictor variables and on these unmeasured variables*.

Prediction should work well whenever $(x_{i1}, x_{i2}, \ldots, x_{i,p-1}, y_i)$, $i = 1, \ldots, n$ constitutes a random sample from some population and when the point we want to predict, say y_0, corresponds to predictor variables $(x_{01}, x_{02}, \ldots, x_{0,p-1})$ that are sampled from the same population. In practice, we rarely have this ideal, but the ideal illuminates what can go wrong in practice.

1.8 Collinearity

Collinearity exists when the predictor variables $x_1, \ldots, x_{p-1}$ are correlated. We have n observations on each of these variables, so we can compute the sample correlations between them. Typically, the x variables are assumed to be fixed and not random. For data like the *Coleman Report*, we have a sample of schools so the predictor variables really are random. But for the purpose of fitting the regression we treat them as fixed. (Probabilistically, we look at the conditional distribution of y given the predictor variables.) In some applications, the person collecting the data actually has control over the predictor variables so they truly are fixed. If the x variables are fixed and not random, there is some question as to what a correlation between two x variables means. Actually, we are concerned with whether the observed predictor variables are *orthogonal*, but that turns out to be *equivalent to having sample correlations of zero* between the x variables. Nonzero sample correlations indicate nonorthogonality, thus we need not concern ourselves with the interpretation of sample correlations between nonrandom samples. (Technically, for orthogonal and uncorrelated predictors to mean the same thing, the predictor variables should have their sample means subtracted from them. I tend to use the terms

interchangeably in this book because of my perception that "uncorrelated" is a less daunting term to people in a second course on regression.)

In regression, it is almost unheard of to have x variables that display no collinearity (correlation) [unless the variables are constructed to have no correlation]. In other words, observed x variables are almost never orthogonal. The key ideas in dealing with collinearity were previously incorporated into the discussion of comparing regression models. In fact, the methods discussed earlier were built around dealing with the collinearity of the x variables. This section merely reviews a few of the main ideas.

1. The estimate of any parameter, say $\hat{\beta}_2$, depends on *all* the variables that are included in the model.
2. The sum of squares for any variable, say x_2, depends on *all* the other variables that are included in the model. For example, none of $SSR(x_2)$, $SSR(x_2|x_1)$, and $SSR(x_2|x_3,x_4)$ would typically be equal.
3. Suppose the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$$

is fitted and we obtain t statistics for each parameter. If the t statistic for testing $H_0 : \beta_1 = 0$ is small, we are led to the model

$$y_i = \beta_0 + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i.$$

If the t statistic for testing $H_0 : \beta_2 = 0$ is small, we are led to the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_3 x_{i3} + \varepsilon_i.$$

However, if the t statistics for both tests are small, we are *not* led to the model

$$y_i = \beta_0 + \beta_3 x_{i3} + \varepsilon_i.$$

To arrive at the model containing only the intercept and x_3, one must at some point use the model containing only the intercept and x_3 as a reduced model.
4. A moderate amount of collinearity has little effect on predictions and therefore little effect on SSE, R^2, and the explanatory power of the model. Collinearity increases the variance of the $\hat{\beta}_k$s, making the estimates of the parameters less reliable. (I told you not to rely on parameters anyway.) Depending on circumstances, sometimes a large amount of collinearity can have an effect on predictions. Just by chance, one may get a better fit to the data than can be justified scientifically.

The complications associated with points 1 through 4 all vanish if the sample correlations between the x variables are *all zero*.

Many computer programs will print out a matrix of correlations between the variables. One would like to think that if all the correlations between the x variables are reasonably small, say less than 0.3 or 0.4, then the problems of collinearity would not be serious. Unfortunately, that is simply not true. To avoid difficulties

with collinearity, not only do all the correlations need to be small but *all of the partial correlations among the x variables must be small*. Thus, small correlations alone do not ensure small collinearity.

EXAMPLE 1.8.1. The correlations among predictors for the Coleman data are given below.

	x_1	x_2	x_3	x_4	x_5
x_1	1.000	0.181	0.230	0.503	0.197
x_2	0.181	1.000	0.827	0.051	0.927
x_3	0.230	0.827	1.000	0.183	0.819
x_4	0.503	0.051	0.183	1.000	0.124
x_5	0.197	0.927	0.819	0.124	1.000

A visual display of these relationships was provided in Figures 1.1–1.4.

Note that x_3 is highly correlated with x_2 and x_5. Since x_3 is highly correlated with y, the fact that x_2 and x_5 are also quite highly correlated with y is not surprising. Recall that the correlations with y were given in Section 1. Moreover, since x_3 is highly correlated with x_2 and x_5, it is also not surprising that x_2 and x_5 have little to add to a model that already contains x_3. We have seen that it is the two variables x_1 and x_4, i.e., the variables that do not have high correlations with either x_3 or y, that have the greater impact on the regression equation.

Having regressed y on x_3, the sample correlations between y and any of the other variables are no longer important. Having done this regression, it is more germane to examine the partial correlations between y and the other variables after adjusting for x_3. (Recall that the sample partial correlation between, say, x_4 and y given x_3 is the just sample correlation between the residuals from fitting y on x_3 and the residuals from fitting x_4 on x_3.) However, as we will see in our discussion of variable selection in Chapter 5, even this has its drawbacks. □

As long as points 1 through 4 are kept in mind, a moderate amount of collinearity is not a big problem. For severe collinearity, there are four common approaches: a) classical ridge regression, b) generalized inverse regression, c) principal components regression, and d) canonical regression. Classical ridge regression is probably the best known of these methods, cf. Section 4.2. The other three methods are closely related and seem quite reasonable. Principal components regression is discussed in Section 4.1. While these methods were originally developed for dealing with collinearity, they are now often used to deal with overfitting, i.e., fitting so many predictor variables that prediction becomes unreliable. Another procedure, *lasso regression*, is becoming increasingly popular for dealing with overfitting but it is considerably more difficult to understand how it works, cf. Section 4.3.

1.9 More on Model Testing

In this section, we take the opportunity to introduce various methods of defining reduced models. To this end we introduce some new data, a subset of the *Chapman data*.

EXAMPLE 1.9.1. Dixon and Massey (1983) report data from the Los Angeles Heart Study supervised by J. M. Chapman. The variables are y, weight in pounds; x_1, age in years; x_2, systolic blood pressure in millimeters of mercury; x_3, diastolic blood pressure in millimeters of mercury; x_4, cholesterol in milligrams per dl; x_5, height in inches. The data from 60 men are given in Table 1.2.

Table 1.2 L. A. heart study data.

i	x_1	x_2	x_3	x_4	x_5	y	i	x_1	x_2	x_3	x_4	x_5	y
1	44	124	80	254	70	190	31	42	136	82	383	69	187
2	35	110	70	240	73	216	32	28	124	82	360	67	148
3	41	114	80	279	68	178	33	40	120	85	369	71	180
4	31	100	80	284	68	149	34	40	150	100	333	70	172
5	61	190	110	315	68	182	35	35	100	70	253	68	141
6	61	130	88	250	70	185	36	32	120	80	268	68	176
7	44	130	94	298	68	161	37	31	110	80	257	71	154
8	58	110	74	384	67	175	38	52	130	90	474	69	145
9	52	120	80	310	66	144	39	45	110	80	391	69	159
10	52	120	80	337	67	130	40	39	106	80	248	67	181
11	52	130	80	367	69	162	41	40	130	90	520	68	169
12	40	120	90	273	68	175	42	48	110	70	285	66	160
13	49	130	75	273	66	155	43	29	110	70	352	66	149
14	34	120	80	314	74	156	44	56	141	100	428	65	171
15	37	115	70	243	65	151	45	53	90	55	334	68	166
16	63	140	90	341	74	168	46	47	90	60	278	69	121
17	28	138	80	245	70	185	47	30	114	76	264	73	178
18	40	115	82	302	69	225	48	64	140	90	243	71	171
19	51	148	110	302	69	247	49	31	130	88	348	72	181
20	33	120	70	386	66	146	50	35	120	88	290	70	162
21	37	110	70	312	71	170	51	65	130	90	370	65	153
22	33	132	90	302	69	161	52	43	122	82	363	69	164
23	41	112	80	394	69	167	53	53	120	80	343	71	159
24	38	114	70	358	69	198	54	58	138	82	305	67	152
25	52	100	78	336	70	162	55	67	168	105	365	68	190
26	31	114	80	251	71	150	56	53	120	80	307	70	200
27	44	110	80	322	68	196	57	42	134	90	243	67	147
28	31	108	70	281	67	130	58	43	115	75	266	68	125
29	40	110	74	336	68	166	59	52	110	75	341	69	163
30	36	110	80	314	73	178	60	68	110	80	268	62	138

For now, our interest is not in analyzing the data but in illustrating modeling techniques. We fitted the basic multiple regression model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i. \quad (1)$$

The table of coefficients and ANOVA table follow.

Table of Coefficients: Model (1)

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	−112.50	89.56	−1.26	0.214
x_1-age	0.0291	0.2840	0.10	0.919
x_2-sbp	0.0197	0.3039	0.06	0.949
x_3-dbp	0.7274	0.4892	1.49	0.143
x_4-chol	−0.02103	0.04859	−0.43	0.667
x_5-ht	3.248	1.241	2.62	0.011

Analysis of Variance: Model (1)

Source	df	SS	MS	F	P
Regression	5	7330.4	1466.1	3.30	0.011
Residual Error	54	24009.6	444.6		
Total	59	31340.0			

One plausible reduced model is that systolic and diastolic blood pressure have the same regression coefficient, i.e, $H_0 : \beta_2 = \beta_3$. Incorporating this into Model (1) gives

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_2 x_{i3} + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i \\ &= \beta_0 + \beta_1 x_{i1} + \beta_2 (x_{i2} + x_{i3}) + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i, \end{aligned} \quad (2)$$

which involves regressing y on the four variables x_1, $x_2 + x_3$, x_4, x_5. The fitted equation is

$$\hat{y} = -113 + 0.018 x_1 + 0.283 (x_2 + x_3) - 0.0178 x_4 + 3.31 x_5.$$

The ANOVA table

Analysis of Variance for Model (2).

Source	df	SS	MS	F	P
Regression	4	6941.9	1735.5	3.91	0.007
Residual Error	55	24398.1	443.6		
Total	59	31340.0			

leads to the test statistic for whether the reduced model fits,

$$F_{obs} = \frac{(24398.1 - 24009.6)/(55 - 54)}{444.6} \doteq 1.$$

The reduced model based on the sum of the blood pressures fits as well as the model with the individual blood pressures.

The table of coefficients for Model (2)

Table of Coefficients: Model (2)

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	−113.16	89.45	−1.27	0.211
x_1-age	0.0182	0.2834	0.06	0.949
x_2+x_3	0.2828	0.1143	2.47	0.016
x_4-chol	−0.01784	0.04841	−0.37	0.714
x_5-ht	3.312	1.237	2.68	0.010

shows a significant effect for the sum of the blood pressures. Although neither blood pressure looked important in the table of coefficients for the full model, we find that the sum of the blood pressures is a good predictor of weight, with a positive regression coefficient. Although high blood pressure is not likely to cause high weight, there is certainly a correlation between weight and blood pressure, so it is plausible that blood pressure could be a good predictor of weight. The reader should investigate whether x_2, x_3, and x_2+x_3 are all acting as surrogates for one another, i.e., whether it is sufficient to include *any one* of the three in the model, after which the others add no appreciable predictive ability.

Another plausible idea, perhaps more so for other dependent variables rather than weight, is that it could be the difference between the blood pressure readings that is important. In this case, the corresponding null hypothesis is $H_0 : \beta_2+\beta_3 = 0$. Writing $\beta_3 = -\beta_2$, the model becomes

$$\begin{aligned} y_i &= \beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}-\beta_2 x_{i3}+\beta_4 x_{i4}+\beta_5 x_{i5}+\varepsilon_i \\ &= \beta_0+\beta_1 x_{i1}+\beta_2(x_{i2}-x_{i3})+\beta_4 x_{i4}+\beta_5 x_{i5}+\varepsilon_i. \end{aligned} \tag{3}$$

With

Analysis of Variance for Model (3)

Source	df	SS	MS	F	P
Regression	4	4575.5	1143.9	2.35	0.065
Residual Error	55	26764.5	486.6		
Total	59	31340.0			

the test statistic for whether the reduced model fits is

$$F_{obs} = \frac{(26764.5-24009.6)/(55-54)}{444.6} = 6.20.$$

The one-sided P value is 0.016, i.e., $6.20 = F(1-.016,1,54)$. Clearly the reduced model fits inadequately. Replacing the blood pressures by their difference does not predict as well as having the blood pressures in the model.

It would have worked equally well to have written $\beta_3 = -\beta_2$ and fitted the reduced model

$$y_i = \beta_0+\beta_1 x_{i1}+\beta_3(x_{i3}-x_{i2})+\beta_4 x_{i4}+\beta_5 x_{i5}+\varepsilon_i.$$

Tests for proportional coefficients are similar to the previous illustrations. For example, we could test if the coefficient for x_2 (sbp) is 4 times smaller than for x_3

(dbp). To test $H_0 : 4\beta_2 = \beta_3$, the reduced model becomes

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + 4\beta_2 x_{i3} + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i \\ &= \beta_0 + \beta_1 x_{i1} + \beta_2 (x_{i2} + 4x_{i3}) + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i. \end{aligned}$$

We leave it to the reader to evaluate this hypothesis.

Now let's test whether the regression coefficient for diastolic blood pressure is 0.5 units higher than for systolic. The hypothesis is $H_0 : \beta_2 + 0.5 = \beta_3$. Substitution gives

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + (\beta_2 + 0.5) x_{i3} + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i \\ &= 0.5x_{i3} + \beta_0 + \beta_1 x_{i1} + \beta_2 (x_{i2} + x_{i3}) + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i. \end{aligned} \tag{4}$$

The term $0.5x_{i3}$ is a known constant for each observation i, often called an *offset*. Such terms are easy to handle in linear models, just take them to the other side of the equation,

$$y_i - 0.5x_{i3} = \beta_0 + \beta_1 x_{i1} + \beta_2 (x_{i2} + x_{i3}) + \beta_4 x_{i4} + \beta_5 x_{i5} + \varepsilon_i, \tag{5}$$

and fit the model with the new dependent variable $y_i - 0.5x_{i3}$.

The fitted regression equation is

$$\hat{y} - 0.5x_3 = -113 + 0.026x_1 + 0.097(x_2 + x_3) - 0.0201x_4 + 3.27x_5$$

or

$$\hat{y} = -113 + 0.026x_1 + 0.097x_2 + 0.597x_3 - 0.0201x_4 + 3.27x_5.$$

The ANOVA table for the reduced model (5) is

Analysis of Variance for Model (5)

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	4	3907.7	976.9	2.23	0.077
Residual Error	55	24043.1	437.1		
Total	59	27950.8			

It may not be obvious but Model (5) can be tested against the full model (1) in the usual way. Since x_{i3} is already included in Model (1), subtracting 0.5 times it from y_i has little effect on Model (1): the fitted values differ only by the constant $0.5x_{i3}$ being subtracted; the residuals and degrees of freedom are identical. Performing the test of Model (5) versus Model (1) gives

$$F_{obs} = \frac{(24043.1 - 24009.6)/(55 - 54)}{444.6} = 0.075$$

for a one-sided P value of 0.79, so the equivalent reduced models (4) and (5) are consistent with the data.

We could similarly test whether the height coefficient is 3.5 in Model (1), i.e., test $H_0 : \beta_5 = 3.5$ by fitting

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + 3.5 x_{i5} + \varepsilon_i$$

or

$$y_i - 3.5 x_{i5} = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \varepsilon_i. \tag{6}$$

Fitting Model (6) gives the regression equation

$$\hat{y} - 3.5 x_5 = -130 + 0.045 x_1 + 0.019 x_2 + 0.719 x_3 - 0.0203 x_4$$

or

$$\hat{y} = -130 + 0.045 x_1 + 0.019 x_2 + 0.719 x_3 - 0.0203 x_4 + 3.5 x_5.$$

The ANOVA table is

Analysis of Variance for Model (6)

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	4	3583.3	895.8	2.05	0.100
Residual Error	55	24027.9	436.9		
Total	59	27611.2			

and testing the models in the usual way gives

$$F_{obs} = \frac{(24027.9 - 24009.6)/(55 - 54)}{444.6} = 0.041$$

for a one-sided P value of 0.84. The reduced model (6) is consistent with the data.

Alternatively, we could test $H_0 : \beta_5 = 3.5$ from the original table of coefficients for Model (1) by computing

$$t_{obs} = \frac{3.248 - 3.5}{1.241} = -0.203$$

and comparing the result to a $t(54)$ distribution. The square of the t statistic equals the F statistic.

Finally, we illustrate a simultaneous test of the last two hypotheses, i.e., we test $H_0 : \beta_2 + 0.5 = \beta_3;\ \beta_5 = 3.5$. The reduced model is

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + (\beta_2 + 0.5) x_{i3} + \beta_4 x_{i4} + 3.5 x_{i5} + \varepsilon_i \\ &= 0.5 x_{i3} + 3.5 x_{i5} + \beta_0 + \beta_1 x_{i1} + \beta_2 (x_{i2} + x_{i3}) + \beta_4 x_{i4} + \varepsilon_i \end{aligned}$$

or

$$y_i - 0.5 x_{i3} - 3.5 x_{i5} = \beta_0 + \beta_1 x_{i1} + \beta_2 (x_{i2} + x_{i3}) + \beta_4 x_{i4} + \varepsilon_i. \tag{7}$$

The fitted regression equation is

$$\hat{y} - .5 x_3 - 3.5 x_5 = -129 + 0.040 x_1 + 0.094 (x_2 + x_3) - 0.0195 x_4$$

or

$$\hat{y} = -129 + 0.040 x_1 + 0.094 x_2 + 0.594 x_3 - 0.0195 x_4 + 3.5 x_5.$$

The ANOVA table is

Analysis of Variance for Model (7)

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	3	420.4	140.1	0.33	0.806
Residual Error	56	24058.8	429.6		
Total	59	24479.2			

and testing Model (7) against Model (1) in the usual way gives

$$F_{obs} = \frac{(24058.8 - 24009.6)/(56 - 54)}{444.6} = 0.055,$$

for a one-sided P value of 0.95. In this case, the high one-sided P value is probably due less to any problems with Model (7) and due more to me looking at the table of coefficients for Model (1) and choosing a null hypothesis that seemed consistent with the data. Typically, *hypotheses should be suggested by previous theory or data, not inspection of the current data.*

1.10 Diagnostics

One tool for checking the assumptions of regression models is looking at diagnostic statistics. If problems with assumptions become apparent, one way to deal with them is to try transformations, e.g., *ANREG-II* Section 7.3. Among the methods discussed there, only the circle of transformations does not apply to all linear regression models. In particular, the discussion of transforming x at the end of *ANREG-II* Section 7.3 takes on new importance in multiple regression because multiple regression involves several predictor variables, each of which is a candidate for Box–Tidwell transformation. Incidentally, the modified Box–Tidwell procedure evaluates each predictor variable separately, so it involves adding only one predictor variable $x_{ij}\log(x_{ij})$ to the multiple regression model at a time.

Table 1.3 contains a variety of measures for checking the assumptions of the multiple regression model with five predictor variables that was fitted to the *Coleman Report* data. The table includes case indicators, the data y, the predicted values $\hat{y}$, the leverages, the standardized residuals r, the standardized deleted residuals t, and Cook's distances C. All of these are standard regression diagnostics. Recall that leverages measure the distance between the predictor variables of a particular case and the overall center of those data. Cases with leverages near 1 dominate any fitted regression. As a rule of thumb, leverages greater than $2p/n$ cause concern and leverages greater than $3p/n$ cause (at least mild) consternation. Here n is the number of observations in the data and p is the number of regression coefficients, including the intercept. The standardized deleted residuals t contain essentially the same information as the standardized residuals r but t values can be compared to a $t(dfE - 1)$ distribution to obtain a formal test of whether a case is consistent with the other

data. (A formal test based on the r values requires a more exotic distribution than the $t(dfE-1)$.) Cook's distance for case i is defined as

$$C_i = \frac{\sum_{h=1}^{n}\left(\hat{y}_h - \hat{y}_{h[i]}\right)^2}{pMSE}, \tag{1}$$

where $\hat{y}_h$ is the predictor of the hth case and $\hat{y}_{h[i]}$ is the predictor of the hth case when case i has been removed from the data. Cook's distance measures the effect of deleting case i on the prediction of all of the original observations.

Table 1.3 Diagnostics: *Coleman Report*, full data.

Case	y	$\hat{y}$	Leverage	r	t	C
1	37.01	36.66	0.482	0.23	0.23	0.008
2	26.51	26.86	0.486	−0.24	−0.23	0.009
3	36.51	40.46	0.133	−2.05	−2.35	0.107
4	40.70	41.17	0.171	−0.25	−0.24	0.002
5	37.10	36.32	0.178	0.42	0.40	0.006
6	33.90	33.99	0.500	−0.06	−0.06	0.001
7	41.80	41.08	0.239	0.40	0.38	0.008
8	33.40	33.83	0.107	−0.22	−0.21	0.001
9	41.01	40.39	0.285	0.36	0.34	0.008
10	37.20	36.99	0.618	0.16	0.16	0.007
11	23.30	25.51	0.291	−1.26	−1.29	0.110
12	35.20	33.45	0.403	1.09	1.10	0.133
13	34.90	35.95	0.369	−0.64	−0.62	0.040
14	33.10	33.45	0.109	−0.18	−0.17	0.001
15	22.70	24.48	0.346	−1.06	−1.07	0.099
16	39.70	38.40	0.157	0.68	0.67	0.014
17	31.80	33.24	0.291	−0.82	−0.81	0.046
18	31.70	26.70	0.326	2.94	4.56	0.694
19	43.10	41.98	0.285	0.64	0.63	0.027
20	41.01	40.75	0.223	0.14	0.14	0.001

Figures 1.5 and 1.6 are plots of the standardized residuals versus normal scores and against the predicted values. (W' is the Shapiro-Francia statistic which is the squared sample correlation between the ordered standardized residuals and the normal scores, thus high values are consistent with normality.) The largest standardized residual, that for case 18, appears to be somewhat unusually large. To test whether the data from case 18 are consistent with the other data, we can compare the standardized deleted residual to a $t(dfE-1)$ distribution. From Table 1.3, the t residual is 4.56. The corresponding P value is 0.0006. Actually, we chose to perform the test on the t residual for case 18 only because it was the largest of the 20 t residuals. Because the test is based on the largest of the t values, it is appropriate to multiply the P value by the number of t statistics considered. This gives $20 \times 0.0006 = 0.012$, which is still a small P value. There is considerable evidence that the data of case

18 are inconsistent, for whatever reason, with the other data. This fact cannot be discovered from a casual inspection of the raw data.

The only point of any concern with respect to the leverages is case 10. Its leverage is 0.618, while $2p/n = 0.6$. This is only a mildly high leverage and case 10 seems well behaved in all other respects; in particular, C_{10} is small, so deleting case 10 has very little effect on predictions.

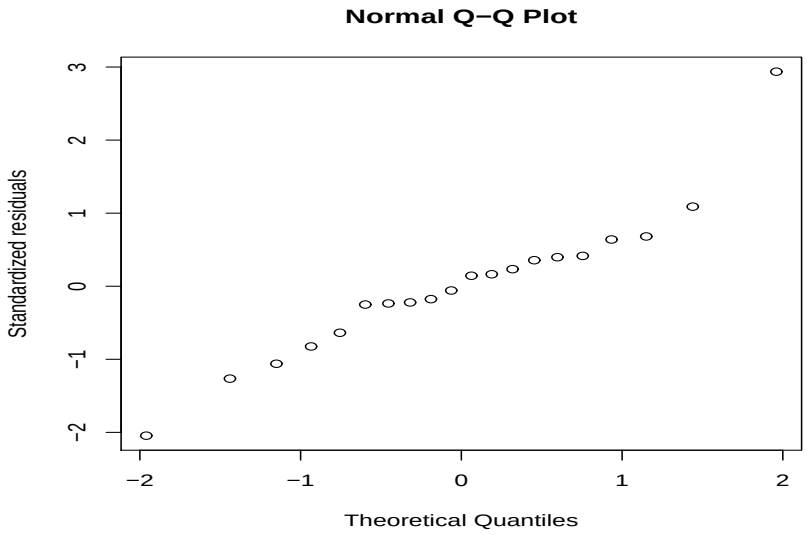

Fig. 1.5 Normal plot, full data, $W' = 0.903$.

We now reconsider the analysis with case 18 deleted. The regression equation is

$$y = 34.3 - 1.62x_1 + 0.0854x_2 + 0.674x_3 + 1.11x_4 - 4.57x_5$$

and $R^2 = 0.963$. Table 1.4 contains the table of coefficients. Table 1.5 contains the analysis of variance. Table 1.6 contains diagnostics. Note that the *MSE* is less than half of its previous value when case 18 was included in the analysis. It is no surprise that the *MSE* is smaller, since the case being deleted is often the single largest contributor to the *SSE*. Correspondingly, the regression parameter t statistics in Table 1.3 are all much more significant. The actual regression coefficient estimates have changed a bit but not greatly. Predictions have not changed radically either, as can be seen by comparing the predictions given in Tables 1.3 and 1.6. Although the predictions have not changed radically, they have changed more than they would have if we deleted any observation other than case 18. From the definition of Cook's distance given in Equation (1), C_{18} is precisely the sum of the squared differences between the predictions in Tables 1.3 and 1.6 divided by 6 times the *MSE* from the

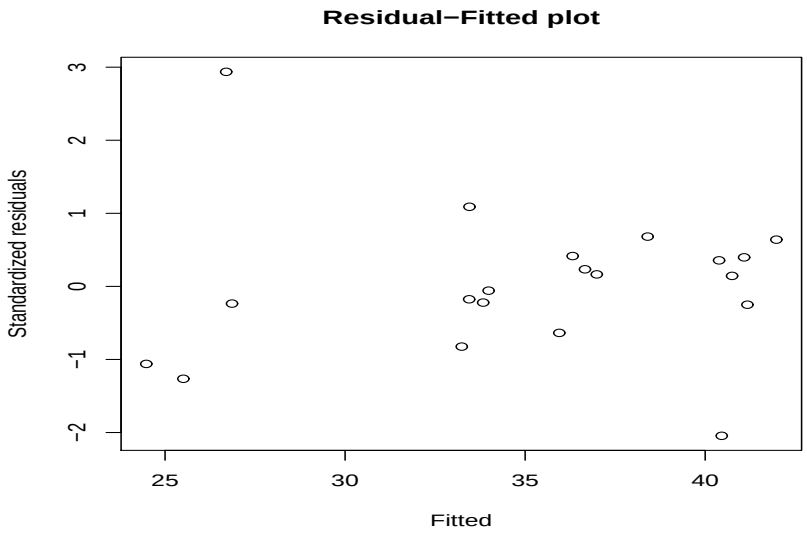

Fig. 1.6 Standardized residuals versus predicted values, full data.

full data. From Table 1.3, Cook's distance when dropping case 18 is much larger than Cook's distance from dropping any other case.

Table 1.4 Table of Coefficients: Case 18 deleted.

Predictor	$\hat{\beta}$	SE($\hat{\beta}$)	t	P
Constant	34.287	9.312	3.68	0.003
x_1	−1.6173	0.7943	−2.04	0.063
x_2	0.08544	0.03546	2.41	0.032
x_3	0.67393	0.06516	10.34	0.000
x_4	1.1098	0.2790	3.98	0.002
x_5	−4.571	1.437	−3.18	0.007

Table 1.5 Analysis of Variance: Case 18 deleted.

Source	df	SS	MS	F	P
Regression	5	607.74	121.55	68.27	0.000
Error	13	23.14	1.78		
Total	18	630.88			

Consider again Table 1.6 containing the diagnostic statistics when case 18 has been deleted. Case 10 has moderately high leverage but seems to be no real problem. Figures 1.7 and 1.8 give the normal plot and the standardized residual versus predicted value plot, respectively, with case 18 deleted. Figure 1.8 is particularly interesting. At first glance, it appears to have a horn shape opening to the right. But there are only three observations on the left of the plot and many on the right, so one would *expect* a horn shape because of the data distribution. Looking at the right of the plot, we see that in spite of the data distribution, much of the horn shape is due to a single very small residual. If we mentally delete that residual, the remaining residuals contain a hint of an upward opening parabola. The potential outlier is case 3. From Table 1.6, the standardized deleted residual for case 3 is -5.08, which yields a raw P value of 0.0001, and if we adjust for having 19 t statistics, the P value is 0.0019, still an extremely small value. Note also that in Table 1.3, when case 18 was included in the data, the standardized deleted residual for case 3 was somewhat large but not nearly so extreme.

Table 1.6 Diagnostics: Case 18 deleted.

Case	y	$\hat{y}$	Leverage	r	t	C
1	37.01	36.64	0.483	0.39	0.37	0.023
2	26.51	26.89	0.486	−0.39	−0.38	0.024
3	36.51	40.21	0.135	−2.98	−5.08	0.230
4	40.70	40.84	0.174	−0.12	−0.11	0.001
5	37.10	36.20	0.179	0.75	0.73	0.020
6	33.90	33.59	0.504	0.33	0.32	0.018
7	41.80	41.66	0.248	0.12	0.12	0.001
8	33.40	33.65	0.108	−0.20	−0.19	0.001
9	41.01	41.18	0.302	−0.15	−0.15	0.002
10	37.20	36.79	0.619	0.50	0.49	0.068
11	23.30	23.69	0.381	−0.37	−0.35	0.014
12	35.20	34.54	0.435	0.66	0.64	0.055
13	34.90	35.82	0.370	−0.87	−0.86	0.074
14	33.10	32.38	0.140	0.58	0.57	0.009
15	22.70	22.36	0.467	0.35	0.33	0.017
16	39.70	38.25	0.158	1.18	1.20	0.044
17	31.80	32.82	0.295	−0.91	−0.90	0.058
18		24.28	0.483			
19	43.10	41.44	0.292	1.48	1.56	0.151
20	41.01	41.00	0.224	0.00	0.00	0.000

With cases 3 and 18 deleted, the regression equation becomes

$$\hat{y} = 29.8 - 1.70x_1 + 0.0851x_2 + 0.666x_3 + 1.18x_4 - 4.07x_5.$$

The R^2 for these data is 0.988. The table of coefficients is in Table 1.7, the analysis of variance is in Table 1.8, and the diagnostics are in Table 1.9.

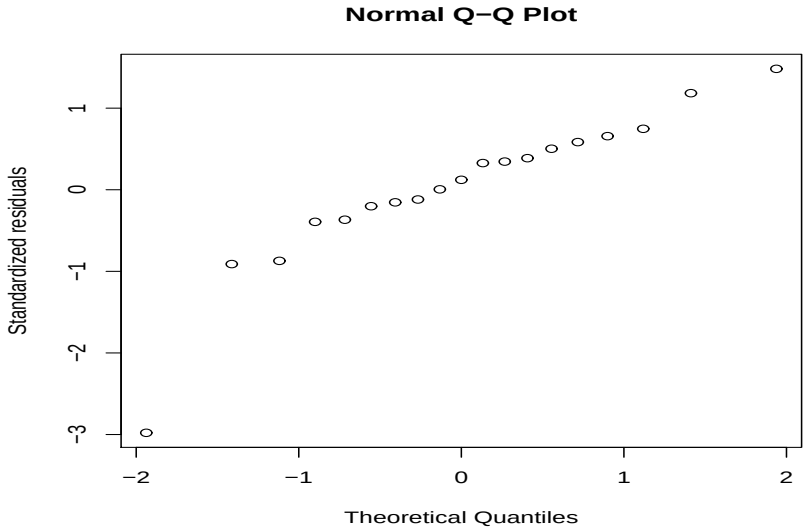

Fig. 1.7 Normal plot, case 18 deleted, $W' = 0.852$.

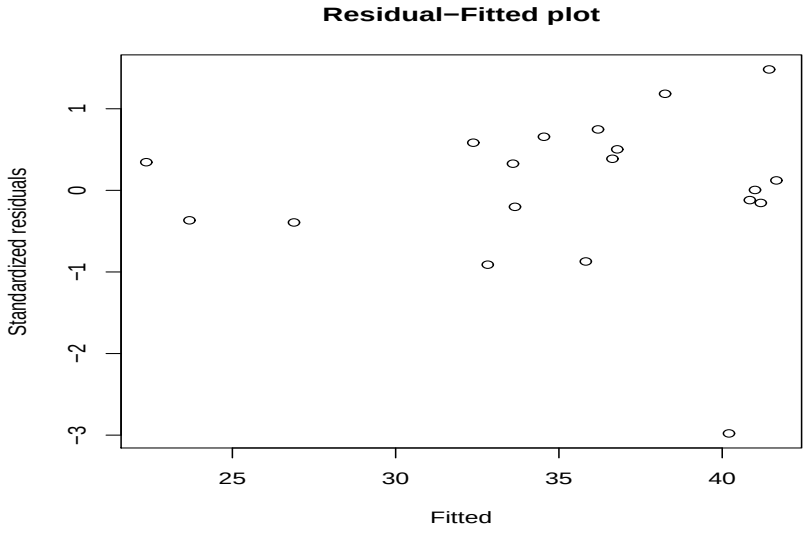

Fig. 1.8 Standardized residuals versus predicted values, case 18 deleted.

Table 1.7 Table of Coefficients: Cases 3 and 18 deleted.

Predictor	$\hat{\beta}$	SE($\hat{\beta}$)	t	P
Constant	29.758	5.532	5.38	0.000
x_1	−1.6985	0.4660	−3.64	0.003
x_2	0.08512	0.02079	4.09	0.001
x_3	0.66617	0.03824	17.42	0.000
x_4	1.1840	0.1643	7.21	0.000
x_5	−4.0668	0.8487	−4.79	0.000

Table 1.8 Analysis of Variance: Cases 3 and 18 deleted.

Source	df	SS	MS	F	P
Regression	5	621.89	124.38	203.20	0.000
Error	12	7.34	0.61		
Total	17	629.23			

Deleting the outlier, case 3, again causes a drop in the MSE, from 1.78 with only case 18 deleted to 0.61 with both cases 3 and 18 deleted. This creates a corresponding drop in the standard errors for all regression coefficients and makes them all appear to be more significant. The actual estimates of the regression coefficients do not change much from Table 1.4 to Table 1.7. The largest changes seem to be in the constant and in the coefficient for x_5.

From Table 1.9, the leverages, t statistics, and Cook's distances seem reasonable. Figures 1.9 and 1.10 contain a normal plot and a plot of standardized residuals versus predicted values. Both plots look good. In particular, the suggestion of lack of fit in Figure 1.8 appears to be unfounded. Once again, Figure 1.10 could be misinterpreted as a horn shape but the 'horn' is due to the distribution of the predicted values.

Ultimately, someone must decide whether or not to delete unusual cases based on subject matter considerations. There is only moderate statistical evidence that case 18 is unusual and case 3 does not look severely unusual unless one previously deletes case 18. Are there subject matter reasons for these schools to be unusual? Will the data be more or less representative of the appropriate population if these data are deleted?

1.11 Final Comment

The maxim for unbalanced data, and regression data are typically unbalanced, is that *if you change anything, you change everything*. If you change a predictor variable in a model, you change the meaning of the regression coefficients (to the extent that they have any meaning), you change the estimates, the fitted values, the residuals,

Table 1.9 Diagnostics: Cases 3 and 18 deleted.

Case	y	$\hat{y}$	Leverage	r	t	C
1	37.01	36.83	0.485	0.33	0.31	0.017
2	26.51	26.62	0.491	−0.20	−0.19	0.007
3		40.78	0.156			
4	40.70	41.43	0.196	−1.04	−1.05	0.044
5	37.10	36.35	0.180	1.07	1.07	0.041
6	33.90	33.67	0.504	0.42	0.41	0.030
7	41.80	42.11	0.261	−0.46	−0.44	0.012
8	33.40	33.69	0.108	−0.39	−0.38	0.003
9	41.01	41.56	0.311	−0.84	−0.83	0.053
10	37.20	36.94	0.621	0.54	0.52	0.078
11	23.30	23.66	0.381	−0.58	−0.57	0.035
12	35.20	34.24	0.440	1.65	1.79	0.356
13	34.90	35.81	0.370	−1.47	−1.56	0.212
14	33.10	32.66	0.145	0.60	0.59	0.010
15	22.70	22.44	0.467	0.46	0.44	0.031
16	39.70	38.72	0.171	1.38	1.44	0.066
17	31.80	33.02	0.298	−1.85	−2.10	0.243
18		24.50	0.486			
19	43.10	42.22	0.332	1.37	1.43	0.155
20	41.01	41.49	0.239	−0.70	−0.68	0.025

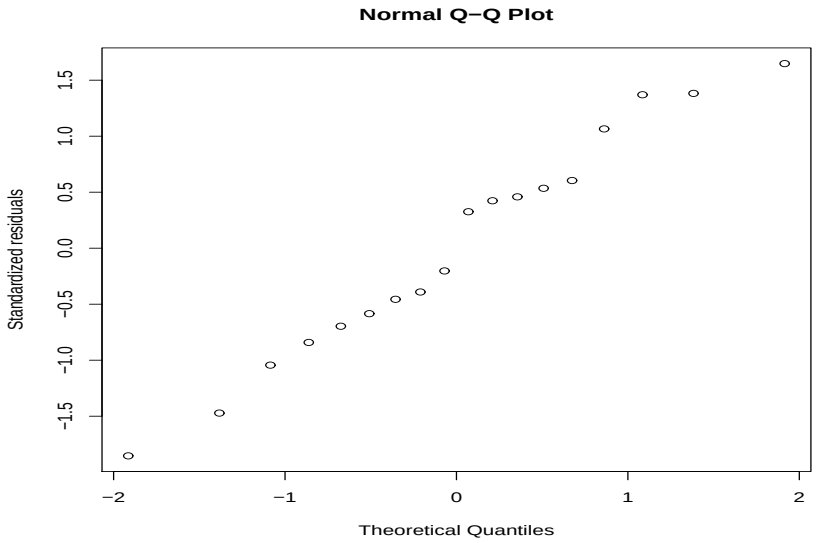

Fig. 1.9 Normal plot, cases 3 and 18 deleted, $W' = 0.979$.

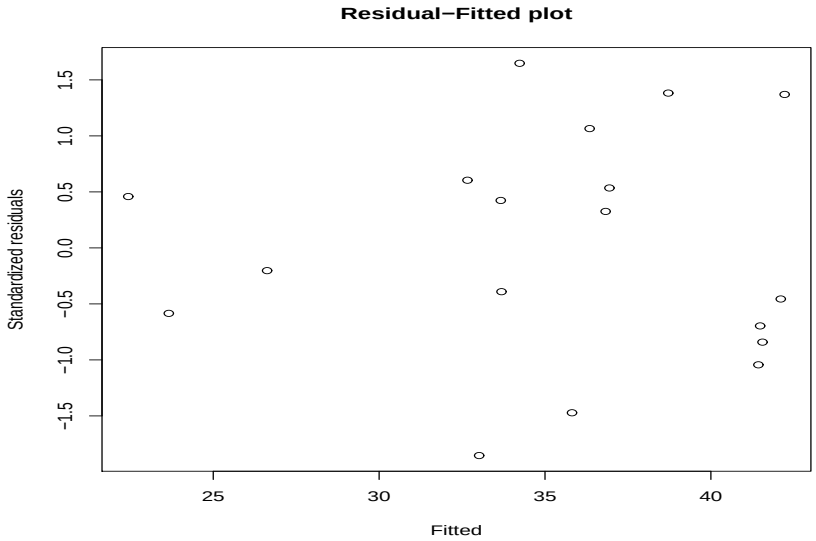

Fig. 1.10 Standardized residuals versus predicted values, cases 3 and 18 deleted.

the leverages: *everything*! If you drop out a data point, you change the meaning of the regression coefficients, the estimates, the fitted values, the residuals, the leverages: *everything*! If you change anything, you change everything. There are a few special cases where this is not true, but they are only special cases.

1.12 Exercises

EXERCISE 1.12.1. Younger (1979, p. 533) presents data from a sample of 12 discount department stores that advertise on television, radio, and in the newspapers. The variables x_1, x_2, and x_3 represent the respective amounts of money spent on these advertising activities during a certain month while y gives the store's revenues during that month. The data are given in Table 1.10. Complete the following tasks using multiple regression.

(a) Give the theoretical model along with the relevant assumptions.
(b) Give the fitted model, i.e., repeat (a) substituting the estimates for the unknown parameters.
(c) Test $H_0 : \beta_2 = 0$ versus $H_A : \beta_2 \neq 0$ at $\alpha = 0.05$.
(d) Test the hypothesis $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$.
(e) Give a 99% confidence interval for β_2.

(f) Test whether the reduced model $y_i = \beta_0 + \beta_1 x_{i1} + \varepsilon_i$ is an adequate explanation of the data as compared to the full model.
(g) Test whether the reduced model $y_i = \beta_0 + \beta_1 x_{i1} + \varepsilon_i$ is an adequate explanation of the data as compared to the model $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$.
(h) Write down the ANOVA table for the 'full' model used in (g).
(i) Construct an added variable plot for adding variable x_3 to a model that already contains variables x_1 and x_2. Interpret the plot.
(j) Compute the sample partial correlation $r_{y3\cdot 12}$. What does this value tell you?

Table 1.10 Younger's advertising data.

Obs.	y	x_1	x_2	x_3	Obs.	y	x_1	x_2	x_3
1	84	13	5	2	7	34	12	7	2
2	84	13	7	1	8	30	10	3	2
3	80	8	6	3	9	54	8	5	2
4	50	9	5	3	10	40	10	5	3
5	20	9	3	1	11	57	5	6	2
6	68	13	5	1	12	46	5	7	2

EXERCISE 1.12.2. The information below relates y, a second measurement on wood volume, to x_1, a first measurement on wood volume, x_2, the number of trees, x_3, the average age of trees, and x_4, the average volume per tree. Note that $x_4 = x_1/x_2$. Some of the information has not been reported, so that you can figure it out on your own.

Table of Coefficients

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	23.45	14.90		0.122
x_1	0.93209	0.08602		0.000
x_2		0.4721	1.5554	0.126
x_3	−0.4982	0.1520		0.002
x_4	3.486	2.274		0.132

Analysis of Variance

Source	df	SS	MS	F	P
Regression	4	887994			0.000
Error					
Total	54	902773			

Sequential

Source	df	SS
x_1	1	883880
x_2	1	183
x_3	1	3237
x_4	1	694

(a) How many observations are in the data?
(b) What is R^2 for this model?
(c) What is the mean squared error?
(d) Give a 95% confidence interval for β_2.
(e) Test the null hypothesis $\beta_3 = 0$ with $\alpha = 0.05$.
(f) Test the null hypothesis $\beta_1 = 1$ with $\alpha = 0.05$.
(g) Give the F statistic for testing the null hypothesis $\beta_3 = 0$.
(h) Give $SSR(x_3|x_1,x_2)$ and find $SSR(x_3|x_1,x_2,x_4)$.
(i) Test the model with only variables x_1 and x_2 against the model with all of variables x_1, x_2, x_3, and x_4.
(j) Test the model with only variables x_1 and x_2 against the model with variables x_1, x_2, and x_3.
(k) Should the test in part (g) be the same as the test in part (j)? Why or why not?
(l) For estimating the point on the regression surface at $(x_1,x_2,x_3,x_4) = (100,25,50,4)$, the standard error of the estimate for the point on the surface is 2.62. Give the estimated point on the surface, a 95% confidence interval for the point on the surface, and a 95% prediction interval for a new point with these x values.
(m) Test the null hypothesis $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$ with $\alpha = 0.05$.

EXERCISE 1.12.3. Atkinson (1985) and Hader and Grandage (1958) have presented Prater's data on gasoline. The variables are y, the percentage of gasoline obtained from crude oil; x_1, the crude oil gravity °API; x_2, crude oil vapor pressure measured in lbs/in^2; x_3, the temperature, in °F, at which 10% of the crude oil is vaporized; and x_4, the temperature, in °F, at which all of the crude oil is vaporized. The data are given in Table 1.3. Find a good model for predicting gasoline yield from the other four variables.

Table 1.11 Prater's gasoline–crude oil data.

y	x_1	x_2	x_3	x_4	y	x_1	x_2	x_3	x_4
6.9	38.4	6.1	220	235	24.8	32.2	5.2	236	360
14.4	40.3	4.8	231	307	26.0	38.4	6.1	220	365
7.4	40.0	6.1	217	212	34.9	40.3	4.8	231	395
8.5	31.8	0.2	316	365	18.2	40.0	6.1	217	272
8.0	40.8	3.5	210	218	23.2	32.2	2.4	284	424
2.8	41.3	1.8	267	235	18.0	31.8	0.2	316	428
5.0	38.1	1.2	274	285	13.1	40.8	3.5	210	273
12.2	50.8	8.6	190	205	16.1	41.3	1.8	267	358
10.0	32.2	5.2	236	267	32.1	38.1	1.2	274	444
15.2	38.4	6.1	220	300	34.7	50.8	8.6	190	345
26.8	40.3	4.8	231	367	31.7	32.2	5.2	236	402
14.0	32.2	2.4	284	351	33.6	38.4	6.1	220	410
14.7	31.8	0.2	316	379	30.4	40.0	6.1	217	340
6.4	41.3	1.8	267	275	26.6	40.8	3.5	210	347
17.6	38.1	1.2	274	365	27.8	41.3	1.8	267	416
22.3	50.8	8.6	190	275	45.7	50.8	8.6	190	407

EXERCISE 1.12.4. Analyze the Chapman data of Example 1.9.1. Find a good model for predicting weight from the other variables.

EXERCISE 1.12.5. Table 1.12 contains a subset of the pollution data analyzed by McDonald and Schwing (1973). The data are from various years in the early 1960s. They relate air pollution to mortality rates for various standard metropolitan statistical areas in the United States. The dependent variable y is the total age-adjusted mortality rate per 100,000 as computed for different metropolitan areas. The predictor variables are, in order, mean annual precipitation in inches, mean January temperature in degrees F, mean July temperature in degrees F, population per household, median school years completed by those over 25, percent of housing units that are sound and with all facilities, population per sq. mile in urbanized areas, percent nonwhite population in urbanized areas, relative pollution potential of sulphur dioxide, annual average of percent relative humidity at 1 pm. Find a good predictive model for mortality.

Alternatively, you can obtain the complete data from the Internet statistical service STATLIB by going to `http://lib.stat.cmu.edu/datasets/` and clicking on "pollution." The data consist of 16 variables on 60 cases.

EXERCISE 1.12.6. Go to `http://lib.stat.cmu.edu/datasets/` and click on "bodyfat." There are data for 15 variables along with a description of the data.

(a) Using the body density measurements as a dependent variable, perform a multiple regression using all of the other variables except body fat as predictor variables. What variables can be safely eliminated from the analysis? Discuss any surprising or expected results in terms of the variables that seem to be most important.
(b) Using the body fat measurements as a dependent variable, perform a multiple regression using all of the other variables except density as predictor variables. What variables can be safely eliminated from the analysis? Discuss any surprising or expected results in terms of the variables that seem to be most important.

Table 1.12 Pollution data.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y
36	27	71	3.34	11.4	81.5	3243	8.8	42.6	59	921.870
35	23	72	3.14	11.0	78.8	4281	3.5	50.7	57	997.875
44	29	74	3.21	9.8	81.6	4260	0.8	39.4	54	962.354
47	45	79	3.41	11.1	77.5	3125	27.1	50.2	56	982.291
43	35	77	3.44	9.6	84.6	6441	24.4	43.7	55	1071.289
53	45	80	3.45	10.2	66.8	3325	38.5	43.1	54	1030.380
43	30	74	3.23	12.1	83.9	4679	3.5	49.2	56	934.700
45	30	73	3.29	10.6	86.0	2140	5.3	40.4	56	899.529
36	24	70	3.31	10.5	83.2	6582	8.1	42.5	61	1001.902
36	27	72	3.36	10.7	79.3	4213	6.7	41.0	59	912.347
52	42	79	3.39	9.6	69.2	2302	22.2	41.3	56	1017.613
33	26	76	3.20	10.9	83.4	6122	16.3	44.9	58	1024.885
40	34	77	3.21	10.2	77.0	4101	13.0	45.7	57	970.467
35	28	71	3.29	11.1	86.3	3042	14.7	44.6	60	985.950
37	31	75	3.26	11.9	78.4	4259	13.1	49.6	58	958.839
35	46	85	3.22	11.8	79.9	1441	14.8	51.2	54	860.101
36	30	75	3.35	11.4	81.9	4029	12.4	44.0	58	936.234
15	30	73	3.15	12.2	84.2	4824	4.7	53.1	38	871.766
31	27	74	3.44	10.8	87.0	4834	15.8	43.5	59	959.221
30	24	72	3.53	10.8	79.5	3694	13.1	33.8	61	941.181
31	45	85	3.22	11.4	80.7	1844	11.5	48.1	53	891.708
31	24	72	3.37	10.9	82.8	3226	5.1	45.2	61	871.338
42	40	77	3.45	10.4	71.8	2269	22.7	41.4	53	971.122
43	27	72	3.25	11.5	87.1	2909	7.2	51.6	56	887.466
46	55	84	3.35	11.4	79.7	2647	21.0	46.9	59	952.529
39	29	75	3.23	11.4	78.6	4412	15.6	46.6	60	968.665
35	31	81	3.10	12.0	78.3	3262	12.6	48.6	55	919.729
43	32	74	3.38	9.5	79.2	3214	2.9	43.7	54	844.053
11	53	68	2.99	12.1	90.6	4700	7.8	48.9	47	861.833
30	35	71	3.37	9.9	77.4	4474	13.1	42.6	57	989.265
50	42	82	3.49	10.4	72.5	3497	36.7	43.3	59	1006.490
60	67	82	2.98	11.5	88.6	4657	13.5	47.3	60	861.439
30	20	69	3.26	11.1	85.4	2934	5.8	44.0	64	929.150
25	12	73	3.28	12.1	83.1	2095	2.0	51.9	58	857.622
45	40	80	3.32	10.1	70.3	2682	21.0	46.1	56	961.009
46	30	72	3.16	11.3	83.2	3327	8.8	45.3	58	923.234
54	54	81	3.36	9.7	72.8	3172	31.4	45.5	62	1113.156
42	33	77	3.03	10.7	83.5	7462	11.3	48.7	58	994.648
42	32	76	3.32	10.5	87.5	6092	17.5	45.3	54	1015.023
36	29	72	3.32	10.6	77.6	3437	8.1	45.5	56	991.290
37	38	67	2.99	12.0	81.5	3387	3.6	50.3	73	893.991
42	29	72	3.19	10.1	79.5	3508	2.2	38.8	56	938.500
41	33	77	3.08	9.6	79.9	4843	2.7	38.6	54	946.185
44	39	78	3.32	11.0	79.9	3768	28.6	49.5	53	1025.502
32	25	72	3.21	11.1	82.5	4355	5.0	46.4	60	874.281

Chapter 2
Matrix Formulation

Abstract In this chapter we use matrices to write regression models. Properties of matrices are reviewed in Appendix A. The economy of notation achieved through using matrices allows us to arrive at some interesting new insights and to derive several of the important properties of regression analysis.

2.1 Random Vectors

In this section we discuss vectors and matrices that are made up of random variables rather than just numbers. For simplicity, we focus our discussion on vectors that contain 3 rows, but the results are completely general.

Let y_1, y_2, and y_3 be random variables. From these, we can construct a 3×1 random vector, say

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}.$$

The expected value of the random vector is just the vector of expected values of the random variables. For the random variables write $\mathrm{E}(y_i) = \mu_i$, then

$$\mathrm{E}(Y) \equiv \begin{bmatrix} \mathrm{E}(y_1) \\ \mathrm{E}(y_2) \\ \mathrm{E}(y_3) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} \equiv \mu.$$

In other words, expectation of a random vector is performed elementwise. In fact, the expected value of any random matrix (a matrix consisting of random variables) is the matrix made up of the expected values of the elements in the random matrix. Thus if w_{ij}, $i = 1,2,3$, $j = 1,2$ is a collection of random variables and we write

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{33} \end{bmatrix},$$

then

$$\mathrm{E}(W) \equiv \begin{bmatrix} \mathrm{E}(w_{11}) & \mathrm{E}(w_{12}) \\ \mathrm{E}(w_{21}) & \mathrm{E}(w_{22}) \\ \mathrm{E}(w_{31}) & \mathrm{E}(w_{33}) \end{bmatrix}.$$

We also need a concept for random vectors that is analogous to the variance of a random variable. This is the *covariance matrix*, sometimes called the *dispersion matrix*, the *variance matrix*, or the *variance-covariance matrix*. The covariance matrix is simply a matrix consisting of all the variances and covariances associated with the vector Y. Write

$$\mathrm{Var}(y_i) = \mathrm{E}(y_i - \mu_i)^2 \equiv \sigma_{ii}$$

and

$$\mathrm{Cov}(y_i, y_j) = \mathrm{E}[(y_i - \mu_i)(y_j - \mu_j)] \equiv \sigma_{ij}.$$

Two subscripts are used on σ_{ii} to indicate that it is the variance of y_i *rather than* writing $\mathrm{Var}(y_i) = \sigma_i^2$.

The covariance matrix of our 3×1 vector Y is

$$\mathrm{Cov}(Y) = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}.$$

When Y is 3×1, the covariance matrix is 3×3. If Y were 20×1, $\mathrm{Cov}(Y)$ would be 20×20. The covariance matrix is always symmetric because $\sigma_{ij} = \sigma_{ji}$ for any i, j. The variances of the individual random variables lie on the diagonal that runs from the top left to the bottom right. The covariances lie off the diagonal.

In general, if Y is an $n \times 1$ random vector and $\mathrm{E}(Y) = \mu$, then $\mathrm{Cov}(Y) = \mathrm{E}[(Y - \mu)(Y - \mu)']$. In other words, $\mathrm{Cov}(Y)$ is the expected value of the random matrix $(Y - \mu)(Y - \mu)'$.

Three simple rules about expectations and covariance matrices can take one a long way in the theory of regression. The first two are matrix analogues of basic results for linear combinations of random variables. In fact, to prove these matrix results, one really only needs the random variable results. The third result is important but a little more complicated to prove. All of these results are establishing in *PA*.

Proposition 2.1.1. Let A be a fixed $r \times n$ matrix, let c be a fixed $r \times 1$ vector, and let Y be an $n \times 1$ random vector with $\mathrm{E}(Y) = \mu$ and $\mathrm{Cov}(Y) = V$, then

1. $\mathrm{E}(AY + c) = A\mu + c$,
2. $\mathrm{Cov}(AY + c) = AVA'$,
3. If $r = n$, then $\mathrm{E}(Y'AY) = \mathrm{tr}(AV) + \mu' A \mu$.

2.2 Matrix Formulation

2.2.1 Simple linear regression in matrix form

The usual model for simple linear regression is

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i \quad i = 1, \ldots, n, \tag{1}$$

$\mathrm{E}(\varepsilon_i) = 0$, $\mathrm{Var}(\varepsilon_i) = \sigma^2$, and $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$ for $i \neq j$. As vectors equation (1) can be written as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_1 + \varepsilon_1 \\ \beta_0 + \beta_1 x_2 + \varepsilon_2 \\ \vdots \\ \beta_0 + \beta_1 x_n + \varepsilon_n \end{bmatrix}.$$

These two vectors are equal if and only if the corresponding elements are equal, which occurs if and only if equation (1) holds.

We can also see by multiplying and adding the matrices on the right-hand side below that

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

$$Y_{n\times 1} \;=\; X_{n\times 2} \quad \beta_{2\times 1} \;+\; e_{n\times 1}$$

This gives the simple linear regression model in the form of a general linear model for which Y is an $n \times 1$ vector of observations on the dependent variable, X is an $n \times p$ matrix of observed predictor variables that usually includes a column of ones associated with fitting an intercept term (for simple linear regression $p = 2$), β is a $p \times 1$ vector of unobservable regression parameters, and e is an unobservable $n \times 1$ vector of random error terms.

The model conditions on the ε_is translate into matrix terms as

$$\mathrm{E}(e) = 0$$

where 0 is the $n \times 1$ matrix containing all zeros and

$$\mathrm{Cov}(e) = \sigma^2 I$$

where I is the $n \times n$ identity matrix. By definition, the covariance matrix $\mathrm{Cov}(e)$ has the variances of the ε_is down the diagonal. The variance of each individual ε_i is σ^2, so all the diagonal elements of $\mathrm{Cov}(e)$ are σ^2, just as in $\sigma^2 I$. The covariance matrix $\mathrm{Cov}(e)$ has the covariances of distinct ε_is as its off-diagonal elements. The covariances of distinct ε_is are all 0, so all the off-diagonal elements of $\mathrm{Cov}(e)$ are zero, just as in $\sigma^2 I$.

Table 2.1 Weights for various heights.

Ht.	Wt.	Ht.	Wt.
65	120	63	110
65	140	63	135
65	130	63	120
65	135	72	170
66	150	72	185
66	135	72	160

EXAMPLE 2.2.1. Height and weight data are given in Table 2.1 for 12 individuals. In matrix terms, the simple linear regression (SLR) model for regressing weights (y) on heights (x) is

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 65 \\ 1 & 65 \\ 1 & 65 \\ 1 & 65 \\ 1 & 66 \\ 1 & 66 \\ 1 & 63 \\ 1 & 63 \\ 1 & 63 \\ 1 & 72 \\ 1 & 72 \\ 1 & 72 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

The observed dependent variable data for this example are

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 120 \\ 140 \\ 130 \\ 135 \\ 150 \\ 135 \\ 110 \\ 135 \\ 120 \\ 170 \\ 185 \\ 160 \end{bmatrix}.$$

We could equally well rearrange the order of the observations to write

$$\begin{bmatrix} y_7 \\ y_8 \\ y_9 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 63 \\ 1 & 63 \\ 1 & 63 \\ 1 & 65 \\ 1 & 65 \\ 1 & 65 \\ 1 & 65 \\ 1 & 66 \\ 1 & 66 \\ 1 & 72 \\ 1 & 72 \\ 1 & 72 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}$$

in which the x_i values are ordered from smallest to largest. □

2.2.2 One-way ANOVA

A one-way analysis of variance (ANOVA) model is written

$$y_{ij} = \mu_i + \varepsilon_{ij} \tag{2}$$

or

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij} \tag{3}$$

for, say, $i = 1, \ldots, a$ and $j = 1, \ldots, N_i$.

EXAMPLE 2.2.2. Let $a = 3$ and $(N_1, N_2, N_3) = (2, 3, 2)$. Model (2) is a regression model and it is an exercise to write it in matrix form. The vector version of equation (3) for these values is

$$\begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \\ y_{23} \\ y_{31} \\ y_{32} \end{bmatrix} = \begin{bmatrix} \mu + \alpha_1 + \varepsilon_{11} \\ \mu + \alpha_1 + \varepsilon_{12} \\ \mu + \alpha_2 + \varepsilon_{21} \\ \mu + \alpha_2 + \varepsilon_{22} \\ \mu + \alpha_2 + \varepsilon_{23} \\ \mu + \alpha_3 + \varepsilon_{31} \\ \mu + \alpha_3 + \varepsilon_{32} \end{bmatrix}.$$

Model (3) holds for $a = 3$ and these N_is if and only if these two vectors are equal. The vector on the right hand side can be obtained from multiplication and addition so that

$$\begin{bmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{22} \\ y_{23} \\ y_{31} \\ y_{32} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{12} \\ \varepsilon_{21} \\ \varepsilon_{22} \\ \varepsilon_{23} \\ \varepsilon_{31} \\ \varepsilon_{32} \end{bmatrix}$$

$$Y \quad = \quad X \quad \beta \quad + \quad e$$

Note that the last three columns of X added together give the first column of X. That causes the rank of this 7×4 matrix to be 3 which is less than the number of columns $p = 4$, so this will not satisfy our forthcoming mathematical condition for being a regression model. □

2.2.3 The general linear model

The standard general linear model is

$$Y = X\beta + e, \quad \mathrm{E}(e) = 0, \quad \mathrm{Cov}(e) = \sigma^2 I.$$

Y is an $n \times 1$ vector of observable random variables. X is an $n \times p$ matrix of known constants. β is a $p \times 1$ vector of unknown (regression) parameters. e is an $n \times 1$ vector of unobservable random errors. It will typically be assumed that $n \geq p$. Mathematically, *regression is any general linear model where the rank of X is p.* (This was not the case for model (3).) In a general linear model, the number of functionally distinct mean parameters is the rank of X.

Using Proposition 2.1.1 and the fact that $X\beta$ is a fixed vector, it follows that

$$\mathrm{E}(Y) = X\beta$$

because

$$\mathrm{E}(Y) = \mathrm{E}(X\beta + e) = X\beta + \mathrm{E}(e) = X\beta + 0 = X\beta.$$

Moreover,

$$\mathrm{Cov}(Y) = \sigma^2 I$$

because

$$\mathrm{Cov}(Y) = \mathrm{Cov}(X\beta + e) = \mathrm{Cov}(e) = \sigma^2 I.$$

In particular, since we know X but we do not know β, all that $\mathrm{E}(Y) = X\beta$ really tells us is that $\mathrm{E}(Y) \in C(X)$. In particular, if we have two different models for the same Y vector, say, $Y = X_1\beta_1 + e_1$ and $Y = X_2\beta_2 + e_2$, if we happen to have $C(X_1) = C(X_2)$, these two models are telling us the same thing about $\mathrm{E}(Y)$ and are therefore referred to as *equivalent models*. In particular, when using least squares estimates, two equiv-

alent models give the same fitted values and thus the same residuals. This does not necessarily happen when using estimates that are alternatives to least squares.

EXAMPLE 2.2.3. *Multiple regression.*
In non-matrix form, the multiple regression model is

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_{p-1} x_{i,p-1} + \varepsilon_i, \quad i = 1, \ldots, n, \tag{4}$$

where

$$\mathrm{E}(\varepsilon_i) = 0, \quad \mathrm{Var}(\varepsilon_i) = \sigma^2, \quad \mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0,\ i \neq j.$$

In matrix terms this can be written as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1,p-1} \\ 1 & x_{21} & x_{22} & \cdots & x_{2,p-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{n,p-1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{p-1} \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}.$$

$$Y_{n\times 1} \;=\; X_{n\times p} \qquad \beta_{p\times 1} \;+\; e_{n\times 1}$$

Multiplying and adding the right-hand side gives

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \cdots + \beta_{p-1} x_{1,p-1} + \varepsilon_1 \\ \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \cdots + \beta_{p-1} x_{2,p-1} + \varepsilon_2 \\ \vdots \\ \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_{p-1} x_{n,p-1} + \varepsilon_n \end{bmatrix},$$

which holds if and only if (4) holds. The conditions on the ε_is translate into

$$\mathrm{E}(e) = 0,$$

where 0 is the $n \times 1$ matrix consisting of all zeros, and

$$\mathrm{Cov}(e) = \sigma^2 I,$$

where I is the $n \times n$ identity matrix. □

EXAMPLE 2.2.4. In Example 2.2.1 we illustrated the matrix form of a SLR using data on heights and weights. We now illustrate some of the models from Chapter 3 applied to these data.

A cubic model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \varepsilon_i \tag{5}$$

is

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 65 & 65^2 & 65^3 \\ 1 & 65 & 65^2 & 65^3 \\ 1 & 65 & 65^2 & 65^3 \\ 1 & 65 & 65^2 & 65^3 \\ 1 & 66 & 66^2 & 66^3 \\ 1 & 66 & 66^2 & 66^3 \\ 1 & 63 & 63^2 & 63^3 \\ 1 & 63 & 63^2 & 63^3 \\ 1 & 63 & 63^2 & 63^3 \\ 1 & 72 & 72^2 & 72^3 \\ 1 & 72 & 72^2 & 72^3 \\ 1 & 72 & 72^2 & 72^3 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

Some of the numbers in X are getting quite large, i.e., $65^3 = 274,625$. The model has better numerical properties if we compute $\bar{x}_\cdot = 69.4166\bar{6}$ and replace model (5) with the equivalent model

$$y_i = \gamma_0 + \gamma_1(x_i - \bar{x}_\cdot) + \gamma_2(x_i - \bar{x}_\cdot)^2 + \beta_3(x_i - \bar{x}_\cdot)^3 + \varepsilon_i$$

and its matrix form

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & (65-\bar{x}_\cdot) & (65-\bar{x}_\cdot)^2 & (65-\bar{x}_\cdot)^3 \\ 1 & (65-\bar{x}_\cdot) & (65-\bar{x}_\cdot)^2 & (65-\bar{x}_\cdot)^3 \\ 1 & (65-\bar{x}_\cdot) & (65-\bar{x}_\cdot)^2 & (65-\bar{x}_\cdot)^3 \\ 1 & (65-\bar{x}_\cdot) & (65-\bar{x}_\cdot)^2 & (65-\bar{x}_\cdot)^3 \\ 1 & (66-\bar{x}_\cdot) & (66-\bar{x}_\cdot)^2 & (66-\bar{x}_\cdot)^3 \\ 1 & (66-\bar{x}_\cdot) & (66-\bar{x}_\cdot)^2 & (66-\bar{x}_\cdot)^3 \\ 1 & (63-\bar{x}_\cdot) & (63-\bar{x}_\cdot)^2 & (63-\bar{x}_\cdot)^3 \\ 1 & (63-\bar{x}_\cdot) & (63-\bar{x}_\cdot)^2 & (63-\bar{x}_\cdot)^3 \\ 1 & (63-\bar{x}_\cdot) & (63-\bar{x}_\cdot)^2 & (63-\bar{x}_\cdot)^3 \\ 1 & (72-\bar{x}_\cdot) & (72-\bar{x}_\cdot)^2 & (72-\bar{x}_\cdot)^3 \\ 1 & (72-\bar{x}_\cdot) & (72-\bar{x}_\cdot)^2 & (72-\bar{x}_\cdot)^3 \\ 1 & (72-\bar{x}_\cdot) & (72-\bar{x}_\cdot)^2 & (72-\bar{x}_\cdot)^3 \end{bmatrix} \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \beta_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

This third-degree polynomial is the largest polynomial that we can fit to these data. Two points determine a line, three points determine a quadratic, and with only four district x values in the data, we cannot fit a model greater than a cubic. Incidentally, it is not a typo that β_3 appears in the both versions of the model. Unlike the γ_js and their corresponding β_js, β_3 turns out to have the same meaning in both models.

For some models it is convenient to transform x into a variable that takes values between 0 and 1. Define $\tilde{x} = (x-63)/9$ so that

$$(x_1, \ldots, x_{12}) = (65,65,65,65,66,66,63,63,63,72,72,72)$$

transforms to

$$(\tilde{x}_1, \ldots, \tilde{x}_{12}) = (2/9,2/9,2/9,2/9,1/3,1/3,0,0,0,1,1,1).$$

A model based on cosines

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 \cos(\pi \tilde{x}_i) + \beta_3 \cos(\pi 2 \tilde{x}_i) + \varepsilon_i$$

becomes

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 65 & \cos(2\pi/9) & \cos(4\pi/9) \\ 1 & 65 & \cos(2\pi/9) & \cos(4\pi/9) \\ 1 & 65 & \cos(2\pi/9) & \cos(4\pi/9) \\ 1 & 65 & \cos(2\pi/9) & \cos(4\pi/9) \\ 1 & 66 & \cos(\pi/3) & \cos(2\pi/3) \\ 1 & 66 & \cos(\pi/3) & \cos(2\pi/3) \\ 1 & 63 & \cos(0) & \cos(0) \\ 1 & 63 & \cos(0) & \cos(0) \\ 1 & 63 & \cos(0) & \cos(0) \\ 1 & 72 & \cos(\pi) & \cos(2\pi) \\ 1 & 72 & \cos(\pi) & \cos(2\pi) \\ 1 & 72 & \cos(\pi) & \cos(2\pi) \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

A *Haar wavelet* model involves indicator functions for sets. The indicator function for a set A is $\mathscr{I}_A(u)$ where $\mathscr{I}_A(u) = 1$ if $u \in A$ and $\mathscr{I}_A(u) = 0$ if $u \notin A$. A Haar wavelet model is

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 \mathscr{I}_{[0,.50)}(\tilde{x}_i) + \beta_3 \mathscr{I}_{[.5,1]}(\tilde{x}_i) + \varepsilon_i$$

and becomes

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 65 & 1 & 0 \\ 1 & 65 & 1 & 0 \\ 1 & 65 & 1 & 0 \\ 1 & 65 & 1 & 0 \\ 1 & 66 & 1 & 0 \\ 1 & 66 & 1 & 0 \\ 1 & 63 & 1 & 0 \\ 1 & 63 & 1 & 0 \\ 1 & 63 & 1 & 0 \\ 1 & 72 & 0 & 1 \\ 1 & 72 & 0 & 1 \\ 1 & 72 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

Notice that the last two columns of the X matrix add up to a column of 1s, like the first column. This causes the rank of the 12×4 model matrix X to be only 3, so the model is not a regression model. Dropping either of the last two columns (or the first column) does not change the model in any meaningful way but makes the model a regression.

Another thing we could do is fit one SLR model to the heights below 65.5 and a different SLR to heights above 65.5. The corresponding matrix model can be written

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 0 & 0 & 1 & 66 \\ 0 & 0 & 1 & 66 \\ 1 & 63 & 0 & 0 \\ 1 & 63 & 0 & 0 \\ 1 & 63 & 0 & 0 \\ 0 & 0 & 1 & 72 \\ 0 & 0 & 1 & 72 \\ 0 & 0 & 1 & 72 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

Here β_1 and β_2 are the intercept and slope for the heights below 65.5 whereas β_3 and β_4 are the intercept and slope for the heights above 65.5. Alternatively, we could rewrite the model as

$$\begin{bmatrix} y_7 \\ y_8 \\ y_9 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 63 & 0 & 0 \\ 1 & 63 & 0 & 0 \\ 1 & 63 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 0 & 0 & 1 & 66 \\ 0 & 0 & 1 & 66 \\ 0 & 0 & 1 & 72 \\ 0 & 0 & 1 & 72 \\ 0 & 0 & 1 & 72 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} + \begin{bmatrix} \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

This makes it a bit clearer that we are fitting a SLR to the points with small x values and a separate SLR to cases with large x values. The pattern of 0s in the X matrix ensure that the small x values only involve the intercept and slope parameters β_1 and β_2 for the line on the first partition set and that the large x values only involve the intercept and slope parameters β_3 and β_4 for the line on the second partition set.

Fitting this model can also be accomplished by fitting the model

$$\begin{bmatrix} y_7 \\ y_8 \\ y_9 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 63 & 0 & 0 \\ 1 & 63 & 0 & 0 \\ 1 & 63 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 66 & 1 & 66 \\ 1 & 66 & 1 & 66 \\ 1 & 72 & 1 & 72 \\ 1 & 72 & 1 & 72 \\ 1 & 72 & 1 & 72 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \gamma_0 \\ \gamma_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

Here we have changed the first two columns to make them agree with the SLR of Example 2.2.1. However, notice that if we subtract the third column from the first column we get the first column of the previous version. Similarly, if we subtract the fourth column from the second column we get the second column of the previous version. This model has intercept and slope parameters β_0 and β_1 for the first partition and intercept and slope parameters $(\beta_0+\gamma_0)$ and $(\beta_1+\gamma_1)$ for the second partition. Thus γ_0 and γ_1 are the change in intercept and slope when going from the first partition to the second.

Another equivalent model makes an adjustment in the predictor variable based on the splitting point 65.5.

$$\begin{bmatrix} y_7 \\ y_8 \\ y_9 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 63 & 0 & 0 \\ 1 & 63 & 0 & 0 \\ 1 & 63 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 65 & 0 & 0 \\ 1 & 66 & 1 & 66-65.5 \\ 1 & 66 & 1 & 66-65.5 \\ 1 & 72 & 1 & 72-65.5 \\ 1 & 72 & 1 & 72-65.5 \\ 1 & 72 & 1 & 72-65.5 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \gamma_0 \\ \gamma_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

It is not hard to see that this has the same column space as the previous model. The reason for illustrating this model is that dropping the intercept adjustment for large values gives the reduced model

$$\begin{bmatrix} y_7 \\ y_8 \\ y_9 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 63 & 0 \\ 1 & 63 & 0 \\ 1 & 63 & 0 \\ 1 & 65 & 0 \\ 1 & 65 & 0 \\ 1 & 65 & 0 \\ 1 & 65 & 0 \\ 1 & 66 & 66-65.5 \\ 1 & 66 & 66-65.5 \\ 1 & 72 & 72-65.5 \\ 1 & 72 & 72-65.5 \\ 1 & 72 & 72-65.5 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \gamma_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix},$$

which turns out to be the linear spline model with a knot a 65.5. In other words, this model forces the two lines to meet one another at the height value 65.5.

Because of the particular structure of these data with 12 observations but only four distinct values of x, except for the Haar wavelet and linear spline models, all of these models are equivalent to one another and all of them are equivalent to a model with the one-way ANOVA regression matrix formulation

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \\ \varepsilon_7 \\ \varepsilon_8 \\ \varepsilon_9 \\ \varepsilon_{10} \\ \varepsilon_{11} \\ \varepsilon_{12} \end{bmatrix}.$$

The models are equivalent in that they all give the same column spaces and therefore the same least squares fitted values, residuals, and degrees of freedom for error. □

2.3 Least Squares Estimation

The regression estimates given by standard computer programs are least squares estimates. For simple linear regression, the least squares estimates are the values of β_0 and β_1 that minimize

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 . \tag{1}$$

For multiple regression, the least squares estimates of the β_js minimize

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \cdots - \beta_{p-1} x_{i,p-1})^2 .$$

In matrix terms these can both be written as minimizing

$$(Y - X\beta)'(Y - X\beta). \tag{2}$$

The form in (2) is just the sum of the squares of the elements in the vector $(Y - X\beta)$.

We now give the general form for the least squares estimate of β in regression problems.

Proposition 2.3.1. If $r(X) = p$, then $\hat{\beta} = (X'X)^{-1} X'Y$ is the least squares estimate of β.

PROOF: Note that $(X'X)^{-1}$ exists only because in a regression problem the rank of X is p so that $X'X$ is a $p \times p$ matrix of rank p and hence invertible. (Note that $X'X$ is also symmetric.) The proof stems from rewriting the function to be minimized.

$$\begin{aligned}(Y - X\beta)'(Y - X\beta) &= \left(Y - X\hat{\beta} + X\hat{\beta} - X\beta\right)' \left(Y - X\hat{\beta} + X\hat{\beta} - X\beta\right) \\ &= \left(Y - X\hat{\beta}\right)' \left(Y - X\hat{\beta}\right) + \left(Y - X\hat{\beta}\right)' \left(X\hat{\beta} - X\beta\right) \\ &\quad + \left(X\hat{\beta} - X\beta\right)' \left(Y - X\hat{\beta}\right) + \left(X\hat{\beta} - X\beta\right)' \left(X\hat{\beta} - X\beta\right).\end{aligned} \tag{3}$$

Consider either one of the two cross-product terms from the previous expression, say, $\left(X\hat{\beta} - X\beta\right)' \left(Y - X\hat{\beta}\right)$. Using the definition of $\hat{\beta}$ given in the proposition,

$$\begin{aligned}\left(X\hat{\beta} - X\beta\right)' \left(Y - X\hat{\beta}\right) &= \left[X\left(\hat{\beta} - \beta\right)\right]' \left(Y - X\hat{\beta}\right) \\ &= \left(\hat{\beta} - \beta\right)' X' \left(Y - X\left(X'X\right)^{-1} X'Y\right) \\ &= \left(\hat{\beta} - \beta\right)' X' \left(I - X\left(X'X\right)^{-1} X'\right) Y\end{aligned}$$

but

$$X' \left(I - X\left(X'X\right)^{-1} X'\right) = X' - \left(X'X\right)\left(X'X\right)^{-1} X' = X' - X' = 0.$$

Thus

$$\left(X\hat{\beta} - X\beta\right)' \left(Y - X\hat{\beta}\right) = 0$$

and similarly

$$\left(Y - X\hat{\beta}\right)' \left(X\hat{\beta} - X\beta\right) = 0.$$

Eliminating the two cross-product terms in (3) gives

$$(Y-X\beta)'(Y-X\beta)=\left(Y-X\hat{\beta}\right)'\left(Y-X\hat{\beta}\right)+\left(X\hat{\beta}-X\beta\right)'\left(X\hat{\beta}-X\beta\right).$$

This form is easily minimized. The first of the terms on the right-hand side does not depend on β, so the β that minimizes $(Y-X\beta)'(Y-X\beta)$ is the β that minimizes the second term $\left(X\hat{\beta}-X\beta\right)'\left(X\hat{\beta}-X\beta\right)$. The second term is non-negative because it is the sum of squares of the elements in the vector $X\hat{\beta}-X\beta$ and it is minimized by making it zero. This is accomplished by choosing $\beta=\hat{\beta}$. □

EXAMPLE 2.3.2. *Simple linear regression.*
We now show that Proposition 2.3.1 gives the usual algebraic estimates for simple linear regression. Assume the model

$$y_i=\beta_0+\beta_1 x_i+\varepsilon_i \quad i=1,\ldots,n.$$

and recall that

$$\hat{\beta}_1=\frac{\sum_{i=1}^n(x_i-\bar{x}_\cdot)y_i}{\sum_{i=1}^n(x_i-\bar{x}_\cdot)^2}=\frac{\sum_{i=1}^n(x_i-\bar{x}_\cdot)(y_i-\bar{y}_\cdot)}{\sum_{i=1}^n(x_i-\bar{x}_\cdot)^2}=\frac{s_{xy}}{s_x^2}=\frac{(\sum_{i=1}^n x_iy_i)-n\bar{x}_\cdot\bar{y}_\cdot}{(n-1)s_x^2}$$

with

$$\hat{\beta}_0=\bar{y}_\cdot-\hat{\beta}_1\bar{x}_\cdot.$$

Now write

$$X=\begin{bmatrix}1 & x_1\\ 1 & x_2\\ \vdots & \vdots\\ 1 & x_n\end{bmatrix}$$

so that

$$X'X=\begin{bmatrix}n & \sum_{i=1}^n x_i\\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2\end{bmatrix}.$$

Inverting this matrix gives

$$\left(X'X\right)^{-1}=\frac{1}{n\sum_{i=1}^n x_i^2-(\sum_{i=1}^n x_i)^2}\begin{bmatrix}\sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i\\ -\sum_{i=1}^n x_i & n\end{bmatrix}.$$

The denominator in this term can be simplified by observing that

$$n\sum_{i=1}^n x_i^2-\left(\sum_{i=1}^n x_i\right)^2=n\left(\sum_{i=1}^n x_i^2-n\bar{x}_\cdot^2\right)=n\sum_{i=1}^n(x_i-\bar{x}_\cdot)^2.$$

Note also that

$$X'Y=\begin{bmatrix}\sum_{i=1}^n y_i\\ \sum_{i=1}^n x_iy_i\end{bmatrix}.$$

Finally, we get

$$
\begin{aligned}
\hat{\beta} &= \left(X'X\right)^{-1}X'Y \\
&= \frac{1}{n\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2}\begin{bmatrix}\sum_{i=1}^{n}x_i^2 & -\sum_{i=1}^{n}x_i\\ -\sum_{i=1}^{n}x_i & n\end{bmatrix}\begin{bmatrix}\sum_{i=1}^{n}y_i\\ \sum_{i=1}^{n}x_iy_i\end{bmatrix} \\
&= \frac{1}{n\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2}\begin{bmatrix}\sum_{i=1}^{n}x_i^2\sum_{i=1}^{n}y_i-\sum_{i=1}^{n}x_i\sum_{i=1}^{n}x_iy_i\\ -\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i+n\sum_{i=1}^{n}x_iy_i\end{bmatrix} \\
&= \frac{1}{\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2}\begin{bmatrix}\bar{y}_.\sum_{i=1}^{n}x_i^2-\bar{x}_.\sum_{i=1}^{n}x_iy_i\\ \left(\sum_{i=1}^{n}x_iy_i\right)-n\bar{x}_.\bar{y}_.\end{bmatrix} \\
&= \frac{1}{\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2}\begin{bmatrix}\bar{y}_.\sum_{i=1}^{n}x_i^2-n\bar{x}_.^2\bar{y}_.-\left\{\bar{x}_.\left(\sum_{i=1}^{n}x_iy_i\right)-\left(n\bar{x}_.^2\bar{y}_.\right)\right\}\\ \sum_{i=1}^{n}(x_i-\bar{x}_.)(y_i-\bar{y}_.)\end{bmatrix} \\
&= \frac{1}{\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2}\begin{bmatrix}\bar{y}_.\left(\sum_{i=1}^{n}x_i^2-n\bar{x}_.^2\right)-\bar{x}_.\left\{\sum_{i=1}^{n}x_iy_i-n\bar{x}_.\bar{y}_.\right\}\\ \hat{\beta}_1\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2\end{bmatrix} \\
&= \frac{1}{\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2}\begin{bmatrix}\bar{y}_.\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2-\bar{x}_.\left\{\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)\left(y_i-\bar{y}_.\right)\right\}\\ \hat{\beta}_1\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2\end{bmatrix} \\
&= \begin{bmatrix}\bar{y}_.-\hat{\beta}_1\bar{x}_.\\ \hat{\beta}_1\end{bmatrix} = \begin{bmatrix}\hat{\beta}_0\\ \hat{\beta}_1\end{bmatrix}.
\end{aligned}
$$

The alternative regression model

$$y_i = \beta_{*0} + \beta_1\left(x_i-\bar{x}_.\right)+\varepsilon_i \quad i=1,\ldots,n$$

is easier to work with. Write the model in matrix form as

$$Y = Z\beta_* + e$$

where

$$Z = \begin{bmatrix}1 & \left(x_1-\bar{x}_.\right)\\ 1 & \left(x_2-\bar{x}_.\right)\\ \vdots & \vdots\\ 1 & \left(x_n-\bar{x}_.\right)\end{bmatrix}$$

and

$$\beta_* = \begin{bmatrix}\beta_{*0}\\ \beta_1\end{bmatrix}.$$

We need to compute $\hat{\beta}_* = (Z'Z)^{-1}Z'Y$. Observe that

$$Z'Z = \begin{bmatrix}n & 0\\ 0 & \sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2\end{bmatrix},$$

$$\left(Z'Z\right)^{-1} = \begin{bmatrix}\frac{1}{n} & 0\\ 0 & 1\Big/\sum_{i=1}^{n}\left(x_i-\bar{x}_.\right)^2\end{bmatrix},$$

$$Z'Y = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} (x_i - \bar{x}_\cdot)\, y_i \end{bmatrix},$$

and

$$\hat{\beta}_* = \left(Z'Z\right)^{-1} Z'Y = \begin{bmatrix} \bar{y}_\cdot \\ \sum_{i=1}^{n} (x_i - \bar{x}_\cdot)\, y_i \Big/ \sum_{i=1}^{n} (x_i - \bar{x}_\cdot)^2 \end{bmatrix} = \begin{bmatrix} \hat{\beta}_{*0} \\ \hat{\beta}_1 \end{bmatrix}.$$

These are the usual estimates. □

Recall that least squares estimates have a number of other properties, for proofs cf. e.g. *PA*. If the errors are independent with mean zero, constant variance, and are normally distributed, the least squares estimates are maximum likelihood estimates and minimum variance unbiased estimates. If the errors are merely uncorrelated with mean zero and constant variance, the least squares estimates are best (minimum variance) linear unbiased estimates.

In multiple regression, simple algebraic expressions for the parameter estimates are not possible. The only nice equations for the estimates are the matrix equations. Applying Proposition 2.1.1, we can find the expected value and covariance matrix of the least squares estimate $\hat{\beta}$. In particular, we show that $\hat{\beta}$ is an *unbiased* estimate of β by showing

$$\mathrm{E}\left(\hat{\beta}\right) = \mathrm{E}\left(\left(X'X\right)^{-1} X'Y\right) = \left(X'X\right)^{-1} X'\mathrm{E}(Y) = \left(X'X\right)^{-1} X'X\beta = \beta.$$

To find variances and standard errors we need $\mathrm{Cov}\left(\hat{\beta}\right)$. In particular, recall that the inverse of a symmetric matrix is symmetric and that $X'X$ is symmetric.

$$\begin{aligned}
\mathrm{Cov}\left(\hat{\beta}\right) &= \mathrm{Cov}\left[\left(X'X\right)^{-1} X'Y\right] \\
&= \left[\left(X'X\right)^{-1} X'\right] \mathrm{Cov}(Y) \left[\left(X'X\right)^{-1} X'\right]' \\
&= \left[\left(X'X\right)^{-1} X'\right] \mathrm{Cov}(Y) X \left[\left(X'X\right)^{-1}\right]' \\
&= \left(X'X\right)^{-1} X' \mathrm{Cov}(Y)\, X \left(X'X\right)^{-1} \\
&= \left(X'X\right)^{-1} X' \left(\sigma^2 I\right) X \left(X'X\right)^{-1} \\
&= \sigma^2 \left(X'X\right)^{-1} X'X \left(X'X\right)^{-1} \\
&= \sigma^2 \left(X'X\right)^{-1}.
\end{aligned}$$

EXAMPLE 2.3.2 CONTINUED. For simple linear regression we only need to take the inverse of a 2×2 matrix so the covariance matrix becomes

$$\begin{aligned}
\mathrm{Cov}\left(\hat{\beta}\right) &= \sigma^2 \left(X'X\right)^{-1} \\
&= \sigma^2 \frac{1}{n\sum_{i=1}^n \left(x_i - \bar{x}_\cdot\right)^2} \begin{bmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{bmatrix} \\
&= \sigma^2 \frac{1}{n\sum_{i=1}^n \left(x_i - \bar{x}_\cdot\right)^2} \begin{bmatrix} \sum_{i=1}^n x_i^2 - n\bar{x}_\cdot^2 + n\bar{x}_\cdot^2 & -n\bar{x}_\cdot \\ -n\bar{x}_\cdot & n \end{bmatrix} \\
&= \sigma^2 \frac{1}{n\sum_{i=1}^n \left(x_i - \bar{x}_\cdot\right)^2} \begin{bmatrix} \sum_{i=1}^n \left(x_i - \bar{x}_\cdot\right)^2 + n\bar{x}_\cdot^2 & -n\bar{x}_\cdot \\ -n\bar{x}_\cdot & n \end{bmatrix} \\
&= \sigma^2 \begin{bmatrix} \frac{1}{n} + \frac{\bar{x}_\cdot^2}{\sum_{i=1}^n (x_i - \bar{x}_\cdot)^2} & \frac{-\bar{x}_\cdot}{\sum_{i=1}^n (x_i - \bar{x}_\cdot)^2} \\ \frac{-\bar{x}_\cdot}{\sum_{i=1}^n (x_i - \bar{x}_\cdot)^2} & \frac{1}{\sum_{i=1}^n (x_i - \bar{x}_\cdot)^2} \end{bmatrix}.
\end{aligned}$$

□

In vector/matrix terms the least squares *fitted (predicted) values* are

$$\hat{Y} \equiv X\hat{\beta} = MY,$$

where we define the *perpendicular projection operator (ppo) onto C(X)* as

$$M = X\left(X'X\right)^{-1}X'.$$

Note that M is both symmetric ($M = M'$) and *idempotent* ($MM = M$). The *residuals* are

$$\hat{e} \equiv Y - \hat{Y} = Y - X\hat{\beta} = (I - M)Y.$$

Proposition 2.1.1 leads to

$$\mathrm{E}(\hat{Y}) = X\beta; \qquad \mathrm{Cov}(\hat{Y}) = \sigma^2 M$$

and

$$\mathrm{E}(\hat{e}) = 0; \qquad \mathrm{Cov}(\hat{e}) = \sigma^2(I - M).$$

For example,

$$\begin{aligned}
\mathrm{Cov}(\hat{e}) &= \mathrm{Cov}([I - M]Y) \\
&= [I - M]\mathrm{Cov}(Y)[I - M]' \\
&= [I - M]\sigma^2 I[I - M]' \\
&= \sigma^2\left(I - M - M' + MM'\right) \\
&= \sigma^2\left(I - M\right).
\end{aligned}$$

The last equality follows because M is symmetric and idempotent.

The sum of squares for error is just the sum of the squared residuals, so

$$SSE = \hat{e}'\hat{e}.$$

Proposition 2.1.1 part 3 then leads to the result

$$\mathrm{E}(MSE) = \sigma^2$$

because, after recalling that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$, $\mathrm{tr}(A+B) = \mathrm{tr}(A) + \mathrm{tr}(B)$, and noting that

$$\mathrm{tr}(M) = \mathrm{tr}\left[X\left(X'X\right)^{-1}X'\right] = \mathrm{tr}\left[X'X\left(X'X\right)^{-1}\right] = \mathrm{tr}\left[I_p\right] = p,$$

we see that

$$\mathrm{E}(MSE) = \frac{1}{n-p}\mathrm{E}(\hat{e}'\hat{e}) = \frac{1}{n-p}\left\{\mathrm{tr}[\sigma^2(I-M)] + 0\right\} = \sigma^2\frac{n-p}{n-p} = \sigma^2.$$

2.4 Inference

We begin by examining the analysis of variance table for the regression model (2.2.4). We then discuss tests, confidence intervals, and prediction intervals.

There are two frequently used forms of the ANOVA table:

Source	df	SS	MS
β_0	1	$n\bar{y}_\cdot^2 \equiv C$	$n\bar{y}_\cdot^2$
Regression	$p-1$	$\hat{\beta}'X'X\hat{\beta} - C$	$SSReg/(p-1)$
Error	$n-p$	$Y'Y - C - SSReg$	$SSE/(n-p)$
Total	n	$Y'Y$	

and the more often used form

Source	df	SS	MS
Regression	$p-1$	$\hat{\beta}'X'X\hat{\beta} - C$	$SSReg/(p-1)$
Error	$n-p$	$Y'Y - C - SSReg$	$SSE/(n-p)$
Total	$n-1$	$Y'Y - C$	

Note that $Y'Y = \sum_{i=1}^n y_i^2$, $C = n\bar{y}_\cdot^2 = \left(\sum_{i=1}^n y_i\right)^2/n$, and $\hat{\beta}'X'X\hat{\beta} = \hat{\beta}'X'Y = Y'MY$. The difference between the two tables is that the first includes a line for the intercept or grand mean while in the second the total has been corrected for the grand mean.

The coefficient of determination can be computed as

$$R^2 = \frac{SSReg}{Y'Y - C}.$$

This is the ratio of the variability explained by the predictor variables to the total variability of the data. Note that $(Y'Y - C)/(n-1) = s_y^2$, the sample variance of the ys without adjusting for any structure except the existence of a possibly nonzero mean.

EXAMPLE 2.4.1. *Simple linear Regression.*
For simple linear regression, we know that

$$SSReg = \hat{\beta}_1^2 \sum_{i=1}^{n} (x_i - \bar{x}_\cdot)^2 = \hat{\beta}_1 \sum_{i=1}^{n} (x_i - \bar{x}_\cdot)^2 \hat{\beta}_1.$$

We will examine the alternative model

$$y_i = \beta_{*0} + \beta_1 (x_i - \bar{x}_\cdot) + \varepsilon_i,$$

which we denote in matrix terms as $Y = Z\beta_* + e$. Note that $C = n\hat{\beta}_{*0}^2$, so the general form for *SSReg* reduces to the simple linear regression form because

$$\begin{aligned} SSReg &= \hat{\beta}_*' Z' Z \hat{\beta}_* - C \\ &= \begin{bmatrix} \hat{\beta}_{*0} \\ \hat{\beta}_1 \end{bmatrix}' \begin{bmatrix} n & 0 \\ 0 & \sum_{i=1}^n (x_i - \bar{x}_\cdot)^2 \end{bmatrix} \begin{bmatrix} \hat{\beta}_{*0} \\ \hat{\beta}_1 \end{bmatrix} - C \\ &= \hat{\beta}_1^2 \sum_{i=1}^{n} (x_i - \bar{x}_\cdot)^2. \end{aligned}$$

The same result can be obtained from $\hat{\beta}' X' X \hat{\beta} - C$ but the algebra is more tedious. □

To obtain tests and confidence regions we need to make additional distributional assumptions. In particular, we assume that the y_is have independent normal distributions. Equivalently, we take

$$\varepsilon_1, \ldots, \varepsilon_n \text{ indep. } N(0, \sigma^2).$$

To test the hypothesis

$$H_0 : \beta_1 = \beta_2 = \cdots = \beta_{p-1} = 0,$$

use the analysis of variance table test statistic

$$F = \frac{MSReg}{MSE}.$$

Under H_0,

$$F \sim F(p-1, n-p).$$

We can also perform a variety of t tests for individual regression parameters β_k. The procedures fit into a general technique based on identifying 1) the parameter, 2) the estimate, 3) the standard error of the estimate, and 4) the distribution of $(Est - Par)/SE(Est)$, cf. *ANREG-II*, Chapter 3. The parameter of interest is β_k. Having previously established that

$$\mathrm{E} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_{p-1} \end{bmatrix} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix},$$

it follows that for any $k = 0, \ldots, p-1$,

$$\mathrm{E}\left(\hat{\beta}_k\right) = \beta_k.$$

This shows that $\hat{\beta}_k$ is an unbiased estimate of β_k. Before obtaining the standard error of $\hat{\beta}_k$, it is necessary to identify its variance. The covariance matrix of $\hat{\beta}$ is $\sigma^2 (X'X)^{-1}$, so the variance of $\hat{\beta}_k$ is the $(k+1)$st diagonal element of $\sigma^2 (X'X)^{-1}$. The $(k+1)$st diagonal element is appropriate because the first diagonal element is the variance of $\hat{\beta}_0$ not $\hat{\beta}_1$. If we let a_k be the $(k+1)$st diagonal element of $(X'X)^{-1}$ and estimate σ^2 with MSE, we get a standard error for $\hat{\beta}_k$ of

$$\mathrm{SE}\left(\hat{\beta}_k\right) = \sqrt{MSE}\sqrt{a_k}.$$

Under normal errors, the appropriate reference distribution is

$$\frac{\hat{\beta}_k - \beta_k}{\mathrm{SE}(\hat{\beta}_k)} \sim t(n-p).$$

Standard techniques now provide tests and confidence intervals. For example, a 95% confidence interval for β_k has endpoints

$$\hat{\beta}_k \pm t(.975, n-p)\,\mathrm{SE}(\hat{\beta}_k)$$

where $t(.975, n-p)$ is the 97.5th percentile of a t distribution with $n-p$ degrees of freedom.

A $(1-\alpha)100\%$ simultaneous confidence region for $\beta_0, \beta_1, \ldots, \beta_{p-1}$ consists of all the β vectors that satisfy

$$\frac{\left(\hat{\beta} - \beta\right)' X'X \left(\hat{\beta} - \beta\right) \Big/ p}{MSE} \leq F(1-\alpha, p, n-p).$$

This region also determines joint $(1-\alpha)100\%$ confidence intervals for the individual β_ks with limits

$$\hat{\beta}_k \pm \sqrt{pF(1-\alpha, p, n-p)}\,\mathrm{SE}(\hat{\beta}_k).$$

These intervals are an application of Scheffé's method of multiple comparisons, cf. Chapter 6.

We can also use the Bonferroni method to obtain joint $(1-\alpha)100\%$ confidence intervals with limits

$$\hat{\beta}_k \pm t\left(1 - \frac{\alpha}{2p}, n-p\right)\,\mathrm{SE}(\hat{\beta}_k).$$

Finally, we consider estimation of the point on the surface that corresponds to a given set of predictor variables and the prediction of a new observation with a given set of predictor variables. Let the predictor variables be $x_1, x_2, \ldots, x_{p-1}$. Combine these into the row vector

$$x' = (1, x_1, x_2, \ldots, x_{p-1}).$$

The point on the surface that we are trying to estimate is the parameter $x'\beta = \beta_0 + \sum_{j=1}^{p-1} \beta_j x_j$. The least squares estimate is $x'\hat{\beta}$, which can be thought of as a 1×1 matrix. The variance of the estimate is

$$\mathrm{Var}\left(x'\hat{\beta}\right) = \mathrm{Cov}\left(x'\hat{\beta}\right) = x'\mathrm{Cov}\left(\hat{\beta}\right)x = \sigma^2 x'\left(X'X\right)^{-1}x,$$

so the standard error is

$$\mathrm{SE}\left(x'\hat{\beta}\right) = \sqrt{MSE}\sqrt{x'\left(X'X\right)^{-1}x} \equiv \mathrm{SE}(Surface).$$

This is the standard error of the estimated regression surface. The appropriate reference distribution is

$$\frac{x'\hat{\beta} - x'\beta}{\mathrm{SE}\left(x'\hat{\beta}\right)} \sim t(n-p)$$

and a $(1-\alpha)100\%$ confidence interval has endpoints

$$x'\hat{\beta} \pm t\left(1 - \frac{\alpha}{2}, n-p\right)\mathrm{SE}(x'\hat{\beta}).$$

When predicting a new observation, the point prediction is just the estimate of the point on the surface but the standard error must incorporate the additional variability associated with a new observation. The original observations were assumed to be independent with variance σ^2. It is reasonable to assume that a new observation is independent of the previous observations and has the same variance. Thus, in the prediction we have to account for the variance of the new observation, which is σ^2, plus the variance of the estimate $x'\hat{\beta}$, which is $\sigma^2 x'\left(X'X\right)^{-1}x$. This leads to a variance for the prediction of $\sigma^2 + \sigma^2 x'\left(X'X\right)^{-1}x$ and a standard error of

$$\sqrt{MSE + MSE\, x'\left(X'X\right)^{-1}x} = \sqrt{MSE\left[1 + x'\left(X'X\right)^{-1}x\right]} \equiv \mathrm{SE}(Prediction).$$

Note that

$$\mathrm{SE}(Prediction) = \sqrt{MSE + [\mathrm{SE}(Surface)]^2}.$$

The $(1-\alpha)100\%$ prediction interval has endpoints

$$x'\hat{\beta} \pm t\left(1 - \frac{\alpha}{2}, n-p\right)\sqrt{MSE\left[1 + x'\left(X'X\right)^{-1}x\right]}.$$

Results of this section constitute the theory behind most of the applications in Sections 1.1, 1.2, and 1.3.

2.5 Diagnostics

Let $x_i' = (1, x_{i1}, \ldots, x_{i,p-1})$ be the ith row of X, then the ith fitted value is

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_{p-1} x_{i,p-1} = x_i' \hat{\beta}$$

and the corresponding residual is

$$\hat{\varepsilon}_i = y_i - \hat{y}_i = y_i - x_i' \hat{\beta}.$$

As mentioned earlier, the vector of predicted (fitted) values is

$$\hat{Y} = \begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} x_1' \hat{\beta} \\ \vdots \\ x_n' \hat{\beta} \end{bmatrix} = X\hat{\beta}.$$

The vector of residuals is

$$\begin{aligned} \hat{e} &= Y - \hat{Y} \\ &= Y - X\hat{\beta} \\ &= Y - X(X'X)^{-1}X'Y \\ &= \left(I - X(X'X)^{-1}X'\right) Y \\ &= (I - M) Y \end{aligned}$$

where

$$M \equiv X(X'X)^{-1}X'$$

is the perpendicular projection operator (matrix) onto $C(X)$. M is the key item in the analysis of the general linear model, cf. *PA*. Since M is symmetric ($M = M'$) and idempotent ($MM = M$),

$$\begin{aligned} SSE &= \sum_{i=1}^{n} \hat{\varepsilon}_i^2 \\ &= \hat{e}'\hat{e} \\ &= [(I - M)Y]'[(I - M)Y] \\ &= Y'(I - M' - M + M'M)Y \\ &= Y'(I - M)Y. \end{aligned}$$

Another common way of writing SSE is

$$SSE = \left[Y - X\hat{\beta}\right]' \left[Y - X\hat{\beta}\right].$$

EXERCISE 2.1 Use the form $SSE = Y'(I-M)Y$ and Proposition 2.1.1 part 3 to show that $\mathrm{E}[Y'(I-M)Y] = (n-p)\sigma^2$ and $\mathrm{E}(MSE) = \sigma^2$. (This is an alternate proof for a result at the end of Section 2.3.)

We can now define the standardized residuals. Recall that the covariance matrix of the residual vector $\hat{e}$ is $\mathrm{Cov}(\hat{e}) = \sigma^2(I-M)$. Typically, the covariance matrix is not diagonal, so the residuals are not uncorrelated. Nonetheless, the variance of a particular residual $\hat{\varepsilon}_i$ is σ^2 times the ith diagonal element of $(I-M)$. The ith diagonal element of $(I-M)$ is the ith diagonal element of I, 1, minus the ith diagonal element of M, say, m_{ii}. Thus

$$\mathrm{Var}(\hat{\varepsilon}_i) = \sigma^2(1-m_{ii})$$

and the standard error of $\hat{\varepsilon}_i$ is

$$\mathrm{SE}(\hat{\varepsilon}_i) = \sqrt{MSE(1-m_{ii})}.$$

The ith *standardized residual* is defined as

$$r_i \equiv \frac{\hat{\varepsilon}_i}{\sqrt{MSE(1-m_{ii})}}.$$

The *leverage* of the ith case is defined to be m_{ii}, the ith diagonal element of M. Some people like to think of M as the 'hat' matrix because it transforms Y into $\hat{Y}$, i.e., $\hat{Y} = X\hat{\beta} = MY$. More common than the name 'hat matrix' is the consequent use of the notation h_i for the ith leverage, thus $h_i \equiv m_{ii}$. In any case, the leverage can be interpreted as a measure of how unusual x_i' is relative to the other rows of the X matrix, cf. *PA*.

PA also discusses the computation of standardized deleted residuals and Cook's distance.

2.6 Basic Notation and Concepts

It seems useful to review and consolidate in one place the basic notation and ideas to be used for linear models (unless defined otherwise for particular purposes). Anything that is not obvious is explained in *PA*.

A linear model has $Y = X\beta + e$ where Y is an $n \times 1$ vector of observable random variables, X is an $n \times p$ matrix of known values, β is a $p \times 1$ vector of fixed but unknown coefficients, and e is an $n \times 1$ vector of unobservable random errors. For this to be a linear model we need $\mathrm{E}(e) = 0$ so that $\mathrm{E}(Y) = X\beta$. A *standard linear model* assumes that an individual observation or error has variance σ^2 and that $\mathrm{Cov}(Y) = \mathrm{Cov}(e) = \sigma^2 I$. The assumption that the observations have a multivariate normal distribution is written $Y \sim N(X\beta, \sigma^2 I)$.

The column space of X is denoted $C(X)$. Since $\mathrm{E}(Y) = X\beta$ but we do not know β, all we really know is that $\mathrm{E}(Y) \in C(X)$. In fact,

$$C(X) \equiv \{\mu | \mu = X\beta \text{ for some } \beta\}.$$

A partitioned linear model is written $Y = X\beta + Z\gamma + e$ where Z is also a matrix of known values and γ is also a vector of fixed, unknown coefficients. If Z has s columns, write

$$X = [X_1, \ldots, X_p] = \begin{bmatrix} x_1' \\ \vdots \\ x_n' \end{bmatrix}; \quad Z = [Z_1, \ldots, Z_s] = \begin{bmatrix} z_1' \\ \vdots \\ z_n' \end{bmatrix}.$$

For any vector v, $\|v\|^2 \equiv v'v$ is the squared (Euclidean) length of v. The Euclidean inner product between two vectors u and v is $u'v$. They are perpendicular (orthogonal), written $v \perp u$, if $v'u = 0$.

A^- denotes a generalized inverse of the matrix A, i.e., any matrix that satisfies $AA^-A = A$. If A^{-1} exists, it is the unique generalized inverse of A. $r(A)$ denotes the rank of A and $\mathrm{tr}(A)$ denotes its trace.

$M = X(X'X)^-X'$ denotes the unique perpendicular projection operator (ppo) onto the column space of X. (With tongue slightly in cheek) the *Fundamental Theorem of Least Squares Estimation* is that in a linear model, $\hat{\beta}$ is a least squares estimate if and only if

$$X\hat{\beta} = MY.$$

The proof works pretty much the same as in Section 3 when $r(X) < p$ except that you have to use $(X'X)^-$ instead of the inverse. In particular, least squares estimates have the form $\hat{\beta} = (X'X)^-X'Y$. Any one of the infinite number of generalized inverses gives a perfectly good least squares estimate. For regression models wherein $r(X) = p$, $(X'X)^{-1}$ always exists, so a generalized inverse is not needed and the unique estimate that satisfies the fundamental theorem is $\hat{\beta} = (X'X)^{-1}X'Y$.

M has the properties that $M = M'$ and $MX = X$. (These are easy to see when $(X'X)^{-1}$ exists.) The property $MX = X$ implies that $MM = M$, i.e., that M is idempotent. More generally, M_A denotes the ppo onto $C(A)$, so in particular $M \equiv M_X$. If $C(X) \subset C(A)$, we can find a matrix B such that $X = AB$. Moreover, we can use this fact to show that if $C(X) = C(A)$, then $M = M_A$.

$C(A)^\perp$ denotes the orthogonal complement of $C(A)$, i.e. all the vectors that are orthogonal to $C(A)$. If $C(X) \subset C(A)$, $C(X)^\perp_{C(A)}$ denotes the orthogonal complement of $C(X)$ with respect to $C(A)$, i.e. all vectors in $C(A)$ that are orthogonal to $C(X)$.

An $r \times c$ matrix of 1s is denoted J_r^c with $J_n \equiv J_n^1$ and $J \equiv J_n$ with similar notation for matrices of 0s.

This is all common notation and, except for the use of M and J, it is pretty much standard notation. (Some authors prefer P and $\mathbf{1}$.)

2.7 Weighted Least Squares

Generalized least squares is a method for dealing with observations that have nonconstant variances and nonzero correlations. In this section, we deal with the simplest form in which we assume zero correlations between observations.

Our standard regression model has

$$Y = X\beta + e, \quad \mathrm{E}(e) = 0, \quad \mathrm{Cov}(e) = \sigma^2 I.$$

We now consider a model for data that do not all have the same variance. In this model, we assume that the *relative* sizes of the variances are known but that the variances themselves are unknown. In this simplest form of weighted regression, we have a covariance structure that changes from $\mathrm{Cov}(e) = \sigma^2 I$ to $\mathrm{Cov}(e) = \sigma^2 D(w)^{-1}$. Here $D(w)$ is a diagonal matrix with *known* weights $w = (w_1, \ldots, w_n)'$ along the diagonal. The covariance matrix involves $D(w)^{-1}$, which is just a diagonal matrix having diagonal entries that are $1/w_1, \ldots, 1/w_n$. The variance of an observation y_i is σ^2/w_i. If w_i is large relative to the other weights, the relative variance of y_i is small, so it contains more information than other observations and we should place more weight on it. Conversely, if w_i is relatively small, the variance of y_i is large, so it contains little information and we should place little weight on it. For all cases, w_i is a measure of how much relative weight should be placed on case i. Note that the weights are relative, so we could multiply or divide them all by a constant and obtain essentially the same analysis. Obviously, in standard regression the weights are all taken to be 1.

In matrix form, our new model is

$$Y = X\beta + e, \quad \mathrm{E}(e) = 0, \quad \mathrm{Cov}(e) = \sigma^2 D(w)^{-1}. \tag{1}$$

In this model all the observations are uncorrelated because the covariance matrix is diagonal. We do not know the variance of any observation because σ^2 is unknown. However, we do know the relative sizes of the variances because we know the weights w_i. It should be noted that when model (1) is used to make predictions, it is necessary to specify weights for any future observations.

The analysis of the weighted regression model (1) is based on changing it into a standard regression model. The trick is to create a new diagonal matrix that has entries $\sqrt{w_i}$. In a minor abuse of notation, we write this matrix as $D(\sqrt{w})$. We now multiply model (1) by this matrix to obtain

$$D(\sqrt{w})Y = D(\sqrt{w})X\beta + D(\sqrt{w})e. \tag{2}$$

It is not difficult to see from Proposition 2.1.1 that

$$\mathrm{E}\big[D(\sqrt{w})e\big] = D(\sqrt{w})\mathrm{E}(e) = D(\sqrt{w})0 = 0$$

and

$$\text{Cov}\left[D(\sqrt{w})e\right] = D(\sqrt{w})\text{Cov}(e)D(\sqrt{w})' = D(\sqrt{w})\left[\sigma^2 D(w)^{-1}\right]D(\sqrt{w}) = \sigma^2 I.$$

Thus equation (2) defines a standard regression model. By Proposition 2.3.1, the least squares regression estimates from model (2) are

$$\begin{aligned}\hat{\beta} &= \left\{[D(\sqrt{w})X]'[D(\sqrt{w})X]\right\}^{-1}[D(\sqrt{w})X]'[D(\sqrt{w})Y] \\ &= [X'D(w)X]^{-1}X'D(w)Y.\end{aligned}$$

The estimate of β given above is referred to as a weighted least squares estimate because rather than minimizing $[Y - X\beta]'[Y - X\beta]$, the estimates are obtained by minimizing

$$[Y - X\beta]'D(w)[Y - X\beta] = \left[D(\sqrt{w})Y - D(\sqrt{w})X\beta\right]'\left[D(\sqrt{w})Y - D(\sqrt{w})X\beta\right].$$

Thus the original minimization problem has been changed into a similar minimization problem that incorporates the weights. The sum of squares for error from model (2) is

$$\begin{aligned}SSE = \left[D(\sqrt{w})Y - D(\sqrt{w})X\hat{\beta}\right]'&\left[D(\sqrt{w})Y - D(\sqrt{w})X\hat{\beta}\right] \\ &= \left[Y - X\hat{\beta}\right]'D(w)\left[Y - X\hat{\beta}\right].\end{aligned}$$

The dfE are unchanged from a standard model and MSE is simply SSE divided by dfE. Standard errors are found in much the same manner as usual except now

$$\text{Cov}\left(\hat{\beta}\right) = \sigma^2[X'D(w)X]^{-1}.$$

Because the $D(w)$ matrix is diagonal, it is very simple to modify a computer program for standard regression to allow the analysis of models like (1). Of course, to make a prediction, a weight must now be specified for the new observation. Essentially the same idea of rewriting model (1) as the standard regression model (2) works even when $D(w)$ is not a diagonal matrix, cf. *PA*, Sections 2.7 and 3.8).

2.8 Variance-Bias Tradeoff

For a standard linear model the least squares estimates are the best linear unbiased estimates and for independent normal data they are the best unbiased estimates. They are best in the sense of having smallest variances. However, it turns out that you can often get better point estimates by incorporating a little bias. A little bit of bias can sometimes eliminate a great deal of variance, making for an overall better estimate.

Suppose the standard linear model

$$Y = X\beta + e, \quad \mathrm{E}(e) = 0, \quad \mathrm{Cov}(e) = \sigma^2 I, \tag{1}$$

is correct and consider fitting a *reduced* model

$$Y = X_0\gamma + e, \quad \text{with } C(X_0) \subset C(X). \tag{2}$$

Most often in regression analysis a reduced model is obtained by tossing out some of the predictor variables in the full model. However, Section 1.9 introduced a variety of other methods for obtaining interesting reduced models. If $\mathrm{E}(Y) \neq X_0\gamma$, using the reduced linear model to estimate $X\beta$ creates bias. In this section we examine how even incorrect reduced models can improve estimation if the bias they introduce is small relative to the variability in the model.

EXAMPLE 2.8.1. Consider fitting a linear model with an intercept and three predictors. I am going to fit the full model using ordinary least squares. You, however, think that the regression coefficients for the second and third variable should be the same and that they should be half of the coefficient for the first variable. You incorporate that into your model using the ideas of Section 1.9. If you are correct, your fitted values will be twice as good as mine! But even if you are wrong, if you are close to being correct, your fitted values will still be better than mine. We now explore these claims in some generality. □

Under the standard linear model (1), the best fitted values one could ever have are $X\beta$ but we don't know β. For estimated fitted values, say, $F(Y)$, their quality can be measured by looking at

$$\mathrm{E}\left\{[F(Y) - X\beta]'[F(Y) - X\beta]\right\}.$$

For least squares estimates $X\hat{\beta} = MY$, using Proposition 2.1.1 part 3 gives

$$\mathrm{E}\left[(MY - X\beta)'(MY - X\beta)\right] = \mathrm{E}\left[(Y - X\beta)'M(Y - X\beta)\right] = \mathrm{tr}[M\sigma^2 I] = \sigma^2 r(X).$$

Now consider the reduced model (2) with M_0 the ppo onto $C(X_0)$. If we estimate the fitted values from the reduced model, i.e. $X_0\hat{\gamma} = M_0 Y$, and the reduced model is true, i.e., $X\beta = X_0\gamma$,

$$\begin{aligned}\mathrm{E}\left[(M_0Y - X\beta)'(M_0Y - X\beta)\right] &= \mathrm{E}\left[(Y - X_0\gamma)'M_0(Y - X_0\gamma)\right] \\ &= \mathrm{tr}[M_0\sigma^2 I] = \sigma^2 r(X_0).\end{aligned}$$

If the reduced model is true, since $r(X_0) \leq r(X)$, we are better off using the reduced model.

In Example 2.8.1, my fitting the full model with $X = [J, X_1, X_2, X_3]$ gives $\mathrm{E}[(MY - X\beta)'(MY - X\beta)] = 4\sigma^2$. Your reduced model has $\beta_2 = \beta_3 = 2\beta_1$, so using ideas from Section 1.9 $X_0 = [J, 2X_1 + X_2 + X_3]$. This leads to $\mathrm{E}[(M_0Y - X\beta)'(M_0Y - X\beta)] = 2\sigma^2$, and the conclusion that your fitted values are

twice as good as mine when your model is correct. (One could argue that it would be more appropriate to look at the square roots, so $\sqrt{2}$ times better?)

What about the claim that you don't have to be correct to do better than me, you only have to be close to correct? You being correct is $\mathrm{E}(Y) = X_0\gamma$. You will be close to correct if the true $\mathrm{E}(Y)$ is close to $X_0\gamma$, i.e., if $X\beta \doteq X_0\gamma$ for some γ. In particular, you will be close to correct if the true mean $X\beta$ is close $C(X_0)$, which happens if $X\beta \doteq M_0X\beta$, i.e., if $X\beta$ is close to its perpendicular projection onto $C(X)$. In general, because $M_0Y - X\beta = M_0(Y - X\beta) - (I - M_0)X\beta$ where $M_0(Y - X\beta) \perp (I - M_0)X\beta$, *PA* shows

$$\mathrm{E}\|M_0Y - X\beta\|^2 = \mathrm{E}\left[(M_0Y - X\beta)'(M_0Y - X\beta)\right] = \sigma^2 r(X_0) + \|X\beta - M_0X\beta\|^2.$$

We have written the expected squared distance as a variance term that is the product of the observation variance σ^2 and the model size $r(X_0)$ plus a bias term that measures the squared distance of how far the reduced model is from being true. If the reduced model is true, the bias is zero. But even when the reduced model is not true, if a reduced model with $r(X_0)$ substantially smaller than $r(X)$ is close to being true, specifically if

$$\|X\beta - M_0X\beta\|^2 < \sigma^2[r(X) - r(X_0)],$$

the fitted values of the reduced model will be better estimates than the original least squares estimates.

In Example 2.8.1, if the squared distance between the truth, $X\beta$, and the expected value of your reduced model estimate, $M_0X\beta$, is less than $2\sigma^2$, you will do better than me. Of course we cannot know how close the truth is to the reduced model expected value, but in Subsection 5.1.3 we will see that Mallow's C_p statistic estimates $r(X_0) + \|X\beta - M_0X\beta\|^2/\sigma^2$, so it gives us an idea about how much better (or worse) a reduced model is doing than the full model. In the context of variable selection, dropping predictor variables with regression coefficients close to zero should result in improved fitted values because the reduced model without those predictors should have $M_0X\beta \doteq X\beta$.

Most biased estimation methods used in practice are immensely more complicated than this discussion. Variable selection methods are perhaps the most widely used methods of creating biased estimates. They use the data to determine an appropriate reduced model, so X_0 is actually a function of Y, say $X_0(Y)$. The computations made here for a fixed X_0 no longer apply.

Most of the alternative estimates discussed later are also biased. If the choice of components in principal component regression is made without reference to Y, then computations similar to those made here are possible. Ridge regression is also relatively tractable. The lasso and other penalized least squares estimates are harder to evaluate.

Bayesian methods, when using a proper prior on β, also provide biased estimates. Whether or not they actually improve the estimates, in the sense discussed here, depends on how well the prior reflects reality. *PA* introduces Bayesian regression while Christensen et al. (2010) go into much more detail.

Chapter 3
Nonparametric Regression I

Abstract If the model you started with does not fit the data very well, an obvious thing to do is to fit a more complicated model. Nonparametric regression provides methods for fitting more complicated models. How can you tell if your original model did not fit very well? You can test it for lack of fit. How do you do that? You test it against a nonparametric regression model. This chapter illustrates lack-of-fit testing by fitting a simple linear regression (SLR) and comparing how it fits relative to some nonparametric regression models.

In analyzing data we often start with an initial model that is relatively complicated, that we hope fits reasonably well, and look for simpler versions that still fit the data adequately. *Lack of fit* involves an initial model that does not fit the data adequately. Most often, we start with a full model and look at reduced models. When dealing with lack of fit, our initial model is the reduced model, and we look for more complicated models that fit significantly better than the reduced model. In this chapter, we introduce methods for creating more complicated models and then testing lack of fit. The more complicated models constitute *nonparametric regression* models. The illustrations are for an initial simple linear regression model. This chapter provides an introduction. Chapter 7 goes into more detail.

3.1 Simple Linear Regression

We begin by fitting a simple linear regression.

EXAMPLE 3.1.1. *Hooker data.*
Forbes (1857) reported data on the relationship between atmospheric pressure and the boiling point of water that were collected in the Himalaya mountains by Joseph Hooker. Weisberg (1985, p. 28) presented a subset of 31 observations that are reproduced in Table 3.1.

A scatter plot of the data is given in Figure 3.1. The data appear to fit a line very closely. The usual summary tables follow for regressing pressure on temperature.

Table 3.1 Hooker data.

Case	Temperature	Pressure	Case	Temperature	Pressure
1	180.6	15.376	17	191.1	19.490
2	181.0	15.919	18	191.4	19.758
3	181.9	16.106	19	193.4	20.480
4	181.9	15.928	20	193.6	20.212
5	182.4	16.235	21	195.6	21.605
6	183.2	16.385	22	196.3	21.654
7	184.1	16.959	23	196.4	21.928
8	184.1	16.817	24	197.0	21.892
9	184.6	16.881	25	199.5	23.030
10	185.6	17.062	26	200.1	23.369
11	185.7	17.267	27	200.6	23.726
12	186.0	17.221	28	202.5	24.697
13	188.5	18.507	29	208.4	27.972
14	188.8	18.356	30	210.2	28.559
15	189.5	18.869	31	210.8	29.211
16	190.6	19.386			

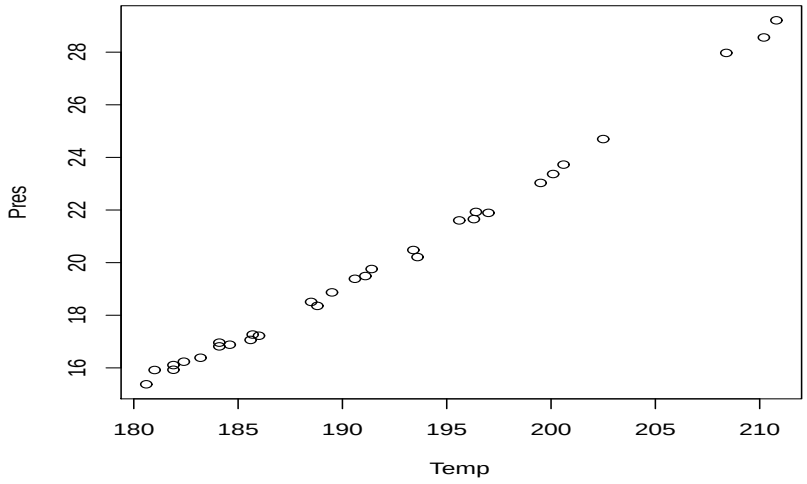

Fig. 3.1 Scatter plot of Hooker data.

Table of Coefficients: Hooker data – SLR.

Predictor	$\hat{\beta}_k$	$\mathrm{SE}(\hat{\beta}_k)$	t	P
Constant	−64.413	1.429	−45.07	0.000
Temperature	0.440282	0.007444	59.14	0.000

Analysis of Variance: Hooker data – SLR.

Source	df	SS	MS	F	P
Regression	1	444.17	444.17	3497.89	0.000
Error	29	3.68	0.13		
Total	30	447.85			

The coefficient of determination is exceptionally large:

$$R^2 = \frac{444.17}{447.85} = 99.2\%.$$

The plot of residuals versus predicted values is given in Figure 3.2. A pattern is very clear; the residuals form something like a parabola. In spite of a very large R^2 and a scatter plot that looks quite linear, the residual plot shows that a lack of fit obviously exists. After seeing the residual plot, you can go back to the scatter plot and detect suggestions of nonlinearity. These suggestions are even more clear in Figure 3.3 which displays the data with the least squares fitted line. The simple linear regression model is clearly inadequate, so we do not bother presenting a normal plot. □

Section 2 considers extending the simple linear regression model by fitting a polynomial in the predictor x. Section 3 considers some strange things that can happen when fitting high-order polynomials. Section 4 introduces the idea of extending the model by using functions of x other than polynomials. Section 5 looks at fitting the model to disjoint subsets of the data. Section 6 examines how the partitioning ideas of Section 5 lead naturally to the idea of fitting "splines." Section 7 gives a brief introduction to Fisher's famous lack-of-fit test.

The ideas of fitting models based on various functions of x and fitting models on subsets of the data (and then recombining the results) are fundamental in the field of *nonparametric regression*. When dealing with several measured predictor variables, methods for nonparametric regression are subject to a *curse of dimensionality*. One approach to removing the hex is discussed in the last two sections. For two predictor variables, Section 8 contrasts additive models with far more complicated interaction models. The generalized additive models introduced in Section 9 extend the ideas of Section 8 to more than two predictor variables. The problem with generalized additive models is that if the curse of dimensionality keeps you from fitting a full nonparametric regression model, the data will not be able to tell you whether the reduced generalized additive model adequately fits the data.

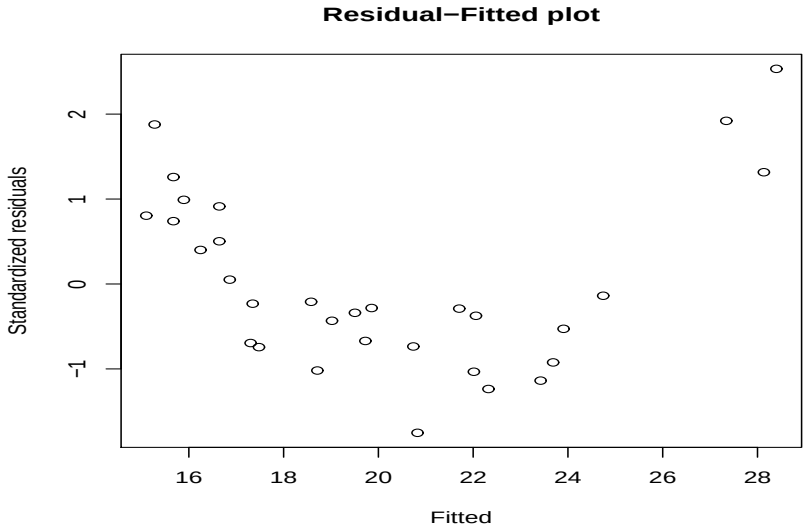

Fig. 3.2 Standardized residuals versus predicted values for Hooker data.

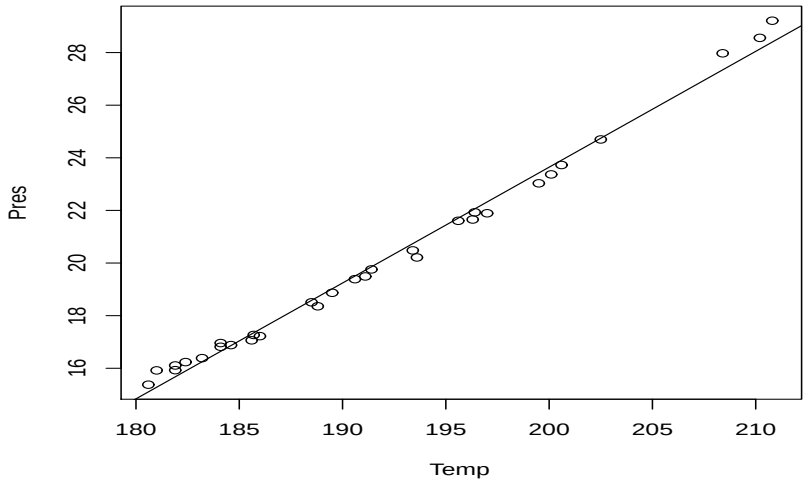

Fig. 3.3 Hooker data, linear fit.

3.2 Polynomial regression

With Hooker's data, the simple linear regression of pressure on temperature shows a lack of fit. The residual plot in Figure 3.2 clearly shows nonrandom structure. One method of dealing with this is to use a power transformation to eliminate the lack of fit, cf *ANREG-II*. In this section we introduce an alternative method called *polynomial regression*.

With a single predictor variable x, we can try to eliminate lack of fit in the simple linear regression $y_i = \beta_0 + \beta_2 x_i + \varepsilon_i$ by fitting larger models. In particular, we can fit the *quadratic* (parabolic) model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i.$$

We could also try a *cubic* model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \varepsilon_i,$$

the *quartic* model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \varepsilon_i,$$

or higher-degree polynomials. If we view our purpose as finding good, easily interpretable approximate models for the data, *high-degree polynomials can behave poorly*. As we will see later, the process of fitting the observed data can cause high-degree polynomials to give very erratic results in areas very near the observed data. A good approximate model should work well, not only at the observed data, but also near it. Thus, we focus on low-degree polynomials. The problem of erratic fits is addressed in the next section. We now examine issues related to fitting polynomials.

EXAMPLE 3.2.1. We fit a fifth-degree (quintic) polynomial to Hooker's data,

$$y_i = \gamma_0 + \gamma_1 x_i + \gamma_2 x_i^2 + \gamma_3 x_i^3 + \gamma_4 x_i^4 + \gamma_5 x_i^5 + \varepsilon_i. \tag{1}$$

Quite clearly this is a linear regression on the predictor variables x, x^2, x^3, x^4, x^5. Some computer programs on which I tried to fit the model to these data encountered numerical instability. They refused to fit it due to the collinearity of the predictor variables. (And possibly to the fact that the R^2 is so high.) To help with the numerical instability of the procedure, before computing the powers of the x variable I subtracted the mean $\bar{x}_\cdot = 191.787$. Thus, I actually fit,

$$y_i = \beta_0 + \beta_1 (x_i - \bar{x}_\cdot) + \beta_2 (x_i - \bar{x}_\cdot)^2 + \beta_3 (x_i - \bar{x}_\cdot)^3 + \beta_4 (x_i - \bar{x}_\cdot)^4 + \beta_5 (x_i - \bar{x}_\cdot)^5 + \varepsilon_i. \tag{2}$$

These two models are equivalent in that they always give the same fitted values, residuals, and degrees of freedom. In addition, $\gamma_5 \equiv \beta_5$ although none of the other γ_js are equivalent to the corresponding β_js. (The equivalences are obtained by the rather ugly process of actually multiplying out the powers of $(x_i - \bar{x}_\cdot)$ in model (2)

so that the model can be rewritten in the form of model (1).) The fitted model, (2), is summarized by the table of coefficients and the ANOVA table.

Table of Coefficients: Model (2).

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	19.7576	0.0581	340.19	0.000
$(x-\bar{x}_\cdot)$	0.41540	0.01216	34.17	0.000
$(x-\bar{x}_\cdot)^2$	0.002179	0.002260	0.96	0.344
$(x-\bar{x}_\cdot)^3$	0.0000942	0.0001950	0.48	0.633
$(x-\bar{x}_\cdot)^4$	0.00001522	0.00001686	0.90	0.375
$(x-\bar{x}_\cdot)^5$	−0.00000080	0.00000095	−0.84	0.409

Analysis of Variance: Model (2).

Source	df	SS	MS	F	P
Regression	5	447.175	89.435	3315.48	0.000
Error	25	0.674	0.027		
Total	30	447.850			

The most important things here are that we now know the SSE, dfE, and MSE from the fifth-degree polynomial. The ANOVA table also provides an F test for comparing the fifth-degree polynomial against the reduced model $y_i = \beta_0 + \varepsilon_i$, not a terribly interesting test.

Usually, *the only interesting t test for a regression coefficient in polynomial regression is the one for the highest term in the polynomial*. In this case the t statistic for the fifth-degree term is -0.84 with a P value of 0.409, so there is little evidence that we need the fifth-degree term in the polynomial. All the t statistics are computed as if the variable in question was the only variable being dropped from the fifth-degree polynomial. For example, it usually makes little sense to have a quintic model that does not include a quadratic term, so there is little point in examining the t statistic for testing $\beta_2 = 0$. One reason for this is that simple linear transformations of the predictor variable change the roles of lower-order terms. For example, something as simple as subtracting $\bar{x}_\cdot$ completely changes the meaning of γ_2 from model (1) to β_2 in model (2). Another way to think about this is that the Hooker data uses temperature measured in Fahrenheit as a predictor variable. The quintic model, (2), for the Hooker data is consistent with $\beta_2 = 0$ with a P value of 0.344. If we changed to measuring temperature in Celsius, there is no reason to believe that the new quintic model would still be consistent with $\beta_2 = 0$. When there is a quintic term in the model, a quadratic term based on Fahrenheit measurements has a completely different meaning than a quadratic term based on Celsius measurements. The same is true for all the other terms except the highest-order term, here the quintic term. On the other hand, the Fahrenheit and Celsius quintic models that include all lower-order terms are equivalent, just as the simple linear regressions based on Fahrenheit and Celsius are equivalent. Of course these comments apply to

all polynomial regressions. Exercise 3.7.7 explores the relationships among regression parameters for quadratic models that have and have not adjusted the predictor for its sample mean.

A lack-of-fit test is provided by testing the quintic model against the original simple linear regression model. The F statistic is

$$F_{obs} = \frac{(3.68 - 0.674)/(29 - 25)}{0.027} = 27.83$$

which is much bigger than 1 and easily significant at the 0.01 level when compared to an $F(4, 25)$ distribution. The test suggests lack of fit (or some other problem with the assumptions). □

3.2.1 Picking a polynomial

We now consider the problem of finding a small-order polynomial that fits the data well.

The table of coefficients for the quintic polynomial on the Hooker data provides a t test for whether we can drop each variable out of the model, but for the most part these tests are uninteresting. The only t statistic that is of interest is that for x^5. It makes little sense, when dealing with a fifth-degree polynomial, to worry about whether you can drop out, say, the quadratic term. The only t statistic of interest is the one that tests whether you can drop x^5 so that you could get by with a quartic polynomial. If you are then satisfied with a quartic polynomial, it makes sense to test whether you can get by with a cubic. In other words, what we would really like to do is fit the sequence of models

$$y_i = \beta_0 + \varepsilon_i, \tag{3}$$

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \tag{4}$$

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i, \tag{5}$$

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \varepsilon_i, \tag{6}$$

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \varepsilon_i, \tag{7}$$

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \beta_5 x_i^5 + \varepsilon_i, \tag{8}$$

and find the smallest model that fits the data. It is equivalent to fit the sequence of polynomials with x adjusted for its mean, $\bar{x}_\cdot$. In subsequent discussion we refer to SSEs and other statistics for models (3) through (8) as $SSE(3)$ through $SSE(8)$ with other similar notations that are obvious. Recall that models (1), (2), and (8) are equivalent.

Since this is just linear regression, many regression programs fit the overall model by fitting a sequence of models and provide key results from the sequence, namely,

the sequential sums of squares, which are simply the difference in error sums of squares for consecutive models in the sequence. Recall that you must specify the variables to the computer program in the order you want them fitted. For the Hooker data, sequential fitting of models (3) through (8) gives

Source	Model Comparison	df	Seq SS	F
$(x-\bar{x}.)$	$SSE(3)-SSE(4)$	1	444.167	16465.9
$(x-\bar{x}.)^2$	$SSE(4)-SSE(5)$	1	2.986	110.7
$(x-\bar{x}.)^3$	$SSE(5)-SSE(6)$	1	0.000	0.0
$(x-\bar{x}.)^4$	$SSE(6)-SSE(7)$	1	0.003	0.1
$(x-\bar{x}.)^5$	$SSE(7)-SSE(8)$	1	0.019	0.7

Using these and statistics reported in Example 3.2.1, the F statistic for dropping the fifth-degree term from the polynomial is

$$F_{obs} = \frac{SSE(7)-SSE(8)}{MSE(8)} = \frac{0.019}{0.027} = 0.71 = (-0.84)^2.$$

The corresponding t statistic reported earlier for testing $H_0 : \beta_5 = 0$ in model (2) was -0.84. The data are consistent with a fourth-degree polynomial.

The F test for dropping to a third-degree polynomial from a fourth-degree polynomial is

$$F_{obs} = \frac{SSE(6)-SSE(7)}{MSE(8)} = \frac{0.003}{0.027} = 0.1161.$$

In the denominator of the test we again use the MSE from the fifth-degree polynomial. *When doing a series of tests on related models one generally uses the MSE from the largest model in the denominator of all tests*, cf. *ANREG-II*, Subsection 3.1.1. The t statistic corresponding to this F statistic is $\sqrt{0.1161} \doteq 0.341$, *not* the value 0.90 reported earlier for the fourth-degree term in the table of coefficients for the fifth-degree model, (2). The t value of 0.341 is a statistic for testing $\beta_4 = 0$ in the fourth-degree model. The value $t_{obs} = 0.341$ is not quite the t statistic (0.343) you would get in the table of coefficients for fitting the fourth-degree polynomial (7) because the table of coefficients would use the MSE from model (7) whereas this statistic is using the MSE from model (8). Nonetheless, t_{obs} provides a test for $\beta_4 = 0$ in a model that has already specified that $\beta_5 = 0$ whereas $t = 0.90$ from the table of coefficients for the fifth-degree model, (2), is testing $\beta_4 = 0$ without specifying that $\beta_5 = 0$. Remember, *if you change anything, you change everything*.

The other F statistics listed are also computed as Seq SS$/MSE(8)$. From the list of F statistics, we can clearly drop any of the polynomial terms down to the quadratic term.

3.2.2 Exploring the chosen model

We now focus on the polynomial model that fits these data well: the quadratic model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i.$$

We have switched to fitting the polynomial without correcting the predictor for its mean value. Summary tables for fitting the quadratic model are

Table of Coefficients: Hooker data, quadratic model.

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	88.02	13.93	6.32	0.000
x	−1.1295	0.1434	−7.88	0.000
x^2	0.0040330	0.0003682	10.95	0.000

Analysis of Variance: Hooker data, quadratic model.

Source	df	SS	MS	F	P
Regression	2	447.15	223.58	8984.23	0.000
Error	28	0.70	0.02		
Total	30	447.85			

The MSE, regression parameter estimates, and standard errors are used in the usual way. The t statistics and P values are for the tests of whether the corresponding β parameters are 0. The t statistics for β_0 and β_1 are of little interest. The t statistic for β_2 is 10.95, which is highly significant, so the quadratic model accounts for a significant amount of the lack of fit displayed by the simple linear regression model. Figure 3.4 gives the data with the fitted parabola.

We will not discuss the ANOVA table in detail, but note that with two predictors, x and x^2, there are 2 degrees of freedom for regression. In general, if we fit a polynomial of degree a, there will be a degrees of freedom for regression, one degree of freedom for every term other than the intercept. Correspondingly, when fitting a polynomial of degree a, there are $n-a-1$ degrees of freedom for error. *The ANOVA table F statistic provides a test of whether the polynomial (in this case quadratic) model explains the data better than the model with only an intercept.*

The fitted values are obtained by substituting the x_i values into

$$\hat{y} = 88.02 - 1.1295x + 0.004033x^2.$$

The residuals are $\hat{\varepsilon}_i = y_i - \hat{y}_i$.

The coefficient of determination is computed and interpreted as before. It is the squared correlation between the pairs $(\hat{y}_i, y_i)$ and also $SSReg$ divided by the $SSTot$, so it measures the amount of the total variability that is explained by the predictor variables temperature and temperature squared. For these data, $R^2 = 99.8\%$, which is an increase from 99.2% for the simple linear regression model. It is not appropriate to compare the R^2 for this model to, say, the R^2 from the SLR model on

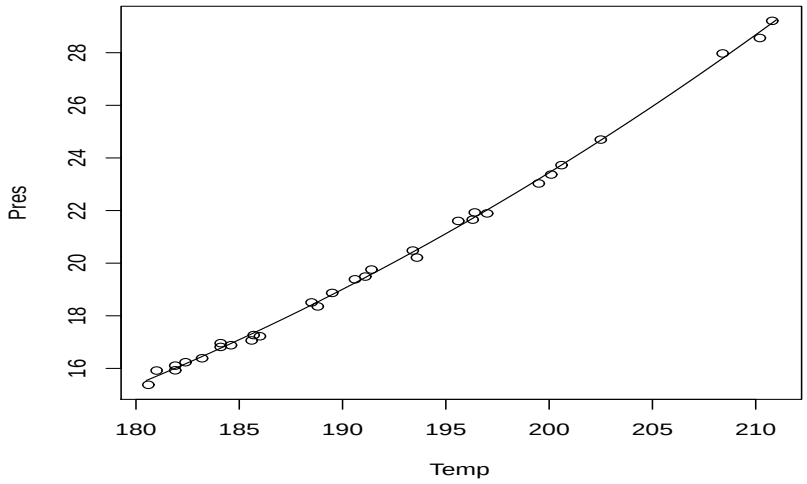

Fig. 3.4 Hooker data with quadratic fit.

the $\log(y)$ of *ANREG-II*, Section 7.3 because they are computed from data that use different scales. However, if we back transform the fitted log values to the original scale to give $\hat{y}_{i\ell}$ values and compute R^2_ℓ as the squared correlation between the $(\hat{y}_{i\ell}, y_i)$ values, then R^2_ℓ and R^2 are comparable.

The standardized residual plots for the quadratic model are given in Figures 3.5 and 3.6. The plot against the predicted values looks good, just as it did for the transformed y data examined in *ANREG-II*, Section 7.3. The normal plot for this model has a shoulder at the top but it looks much better than the normal plot for the simple linear regression on the log transformed data.

If we are interested in the mean value of pressure for a temperature of 205°F, the quadratic model estimate is (up to a little round-off error)

$$\hat{y} = 25.95 = 88.02 - 1.1295\,(205) + 0.004033\,(205)^2.$$

The standard error (as reported by the computer program) is 0.0528 and a 95% confidence interval is $(25.84, 26.06)$. This compares to a point estimate of 25.95 and a 95% confidence interval of $(25.80, 26.10)$ obtained in *ANREG-II*, Section 7.3 from regressing the log of pressure on temperature and back transforming. The quadratic model *prediction* for a new observation at 205°F is again 25.95 with a 95% prediction interval of $(25.61, 26.29)$. The corresponding back transformed prediction interval from the log transformed data is $(25.49, 26.42)$. In this example, the results of the two methods for dealing with lack of fit are qualitatively very similar, at least at 205°F.

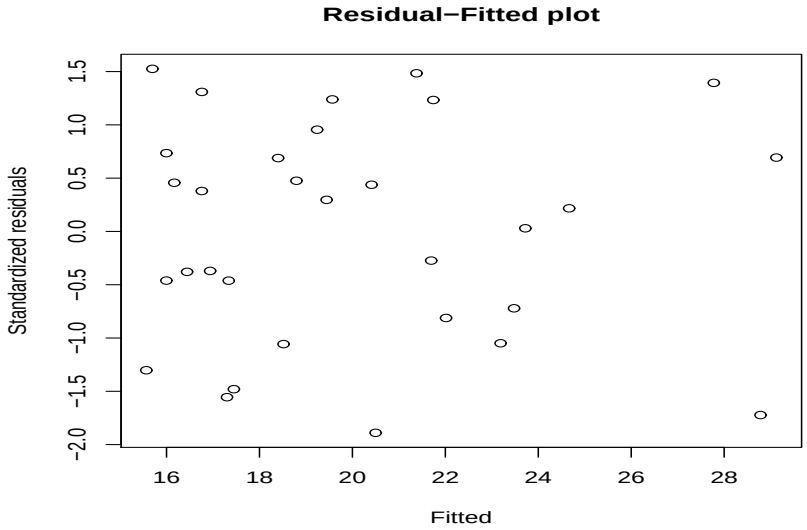

Fig. 3.5 Standardized residuals versus predicted values, quadratic model.

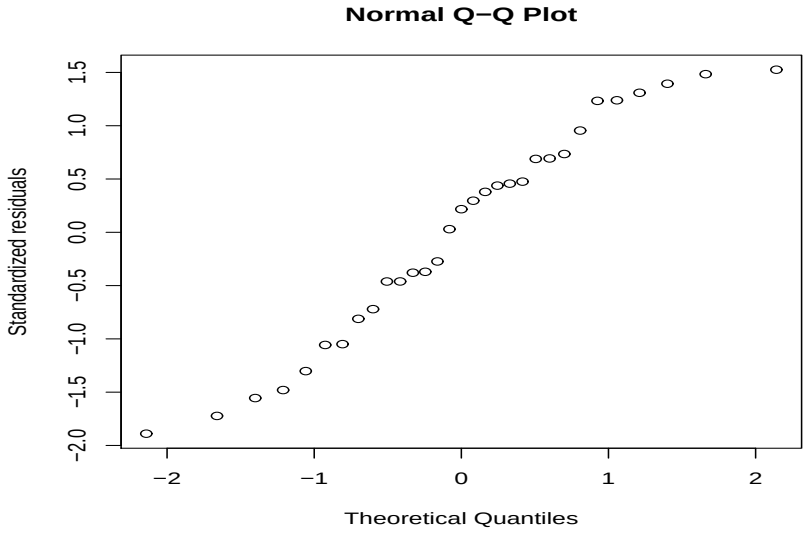

Fig. 3.6 Normal plot for quadratic model, $W' = 0.966$.

Finally, consider testing the quadratic model for lack of fit by comparing it to the quintic model (2). The F statistic is

$$F_{obs} = \frac{(0.70 - 0.674)/(28 - 25)}{0.027} = 0.321,$$

which is much smaller than 1 and makes no suggestion of lack of fit.

One thing we have not addressed is why we chose a fifth-degree polynomial rather than a fourth-degree or a sixth-degree or a twelfth-degree. The simplest answer is just to pick something that clearly turns out to be large enough to catch the important features of the data. If you start with too small a polynomial, go back and pick a bigger one. □

3.3 Overfitting Polynomial Regression

We now present a simple example that illustrates two points: that leverages depend on the model and that high-order polynomials can fit the data in very strange ways.

EXAMPLE 3.3.1. The data for the example follow. They were constructed to have most observations far from the middle.

Case	1	2	3	4	5	6	7
y	0.445	1.206	0.100	−2.198	0.536	0.329	−0.689
x	0.0	0.5	1.0	10.0	19.0	19.5	20.0

I selected the x values. The y values are a sample of size 7 from a $N(0,1)$ distribution. Note that with seven distinct x values, we can fit a polynomial of degree 6.

The data are plotted in Figure 3.7. Just by chance (honest, folks), I observed a very small y value at $x = 10$, so the data appear to follow a parabola that opens up. The small y value at $x = 10$ totally dominates the impression given by Figure 3.7. If the y value at $x = 10$ had been near 3 rather than near -2, the data would appear to be a parabola that opens down. If the y value had been between 0 and 1, the data would appear to fit a line with a slightly negative slope. When thinking about fitting a parabola, the case with $x = 10$ is an extremely high-leverage point.

Depending on the y value at $x = 10$, the data suggest a parabola opening up, a parabola opening down, or that we do not need a parabola to explain the data. Regardless of the y value observed at $x = 10$, the fitted parabola must go nearly through the point $(10, y)$. On the other hand, if we think only about fitting a line to these data, the small y value at $x = 10$ has much less effect. In fitting a line, the value $y = -2.198$ will look unusually small (it will have a very noticeable standardized residual), but it will not force the fitted line to go nearly through the point $(10, -2.198)$.

Table 3.2 gives the leverages for all of the polynomial models that can be fitted to these data. Note that there are no large leverages for the simple linear regression model (the linear polynomial). For the quadratic (parabolic) model, all of the

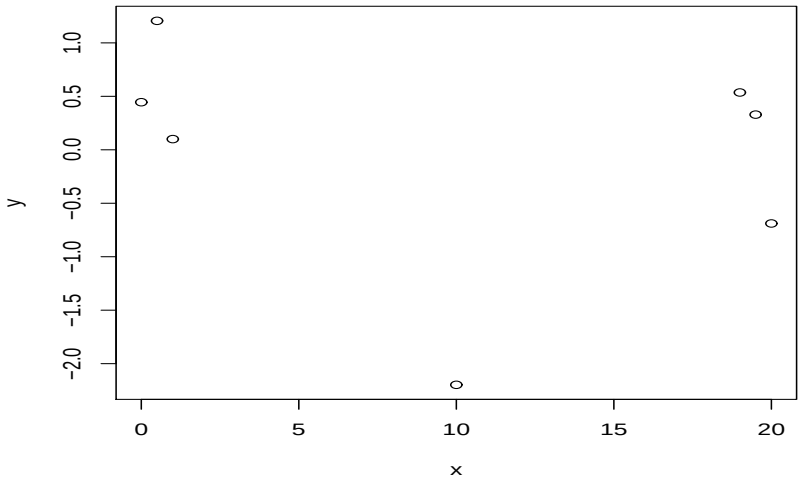

Fig. 3.7 Plot of y versus x.

leverages are reasonably small except the leverage of 0.96 at $x = 10$ that very nearly equals 1. Thus, in the quadratic model, the value of y at $x = 10$ dominates the fitted polynomial. The cubic model has extremely high leverage at $x = 10$, but the leverages are also beginning to get large at $x = 0, 1, 19, 20$. For the quartic model, the leverage at $x = 10$ is 1, to two decimal places; the leverages for $x = 0, 1, 19, 20$ are also nearly 1. The same pattern continues with the quintic model but the leverages at $x = 0.5, 19.5$ are also becoming large. Finally, with the sixth-degree (hexic) polynomial, all of the leverages are exactly one. This indicates that the sixth-degree polynomial has to go through every data point exactly and thus every data point is extremely influential on the estimate of the sixth-degree polynomial. (It is fortunate that there are only seven distinct x values. This discussion would really tank if we had to fit a seventh-degree polynomial. [Think about it: quartic, quintic, hexic, ... tank.])

As we fit larger polynomials, we get more high-leverage cases (and more numerical instability). Actually, as in our example, this occurs when the size of the polynomial nears one less than the number of distinct x values and nearly all data points have distinct x values. *The estimated polynomials must go very nearly through all high-leverage cases. To accomplish this the estimated polynomials may get very strange*. We now give all of the fitted polynomials for these data.

Table 3.2 Leverages.

	Model					
x	Linear	Quadratic	Cubic	Quartic	Quintic	Hexic
0.0	0.33	0.40	0.64	0.87	0.94	1.00
0.5	0.31	0.33	0.33	0.34	0.67	1.00
1.0	0.29	0.29	0.55	0.80	0.89	1.00
10.0	0.14	0.96	0.96	1.00	1.00	1.00
19.0	0.29	0.29	0.55	0.80	0.89	1.00
19.5	0.31	0.33	0.33	0.34	0.67	1.00
20.0	0.33	0.40	0.64	0.87	0.94	1.00

Model	Estimated polynomial
Linear	$\hat{y} = 0.252 - 0.029x$
Quadratic	$\hat{y} = 0.822 - 0.536x + 0.0253x^2$
Cubic	$\hat{y} = 1.188 - 1.395x + 0.1487x^2 - 0.0041x^3$
Quartic	$\hat{y} = 0.713 - 0.141x - 0.1540x^2 + 0.0199x^3 - 0.00060x^4$
Quintic	$\hat{y} = 0.623 + 1.144x - 1.7196x^2 + 0.3011x^3 - 0.01778x^4 + 0.000344x^5$
Hexic	$\hat{y} = 0.445 + 3.936x - 5.4316x^2 + 1.2626x^3 - 0.11735x^4 + 0.004876x^5 - 0.00007554x^6$

Figures 3.8 and 3.9 contain graphs of these estimated polynomials.

Figure 3.8 contains the estimated linear, quadratic, and cubic polynomials. The linear and quadratic curves fit about as one would expect from looking at the scatter plot Figure 3.7. For x values near the range 0 to 20, we could use these curves to predict y values and get reasonable, if not necessarily good, results. One could not say the same for the estimated cubic polynomial. The cubic curve takes on $\hat{y}$ values near -3 for some x values that are near 6. The y values in the data are between about -2 and 1.2; nothing in the data suggests that y values near -3 are likely to occur. Such predicted values are entirely the product of fitting a cubic polynomial. If we really knew that a cubic polynomial was correct for these data, the estimated polynomial would be perfectly appropriate. But most often we use polynomials to approximate the behavior of the data and for these data the cubic polynomial gives a poor approximation.

Figure 3.9 gives the estimated quartic, quintic, and hexic polynomials. Note that the scale on the y axis has changed drastically from Figure 3.8. Qualitatively, the fitted polynomials behave like the cubic except their behavior is even worse. These polynomials do very strange things everywhere except near the observed data.

It is a theoretical fact that when the degrees of freedom for error get small, the *MSE* should be an erratic estimate of σ^2. In my experience, another phenomenon that sometimes occurs when fitting large models to data is that the mean squared error gets unnaturally *small*, cf. Section 5.2. Table 3.3 gives the analysis of variance

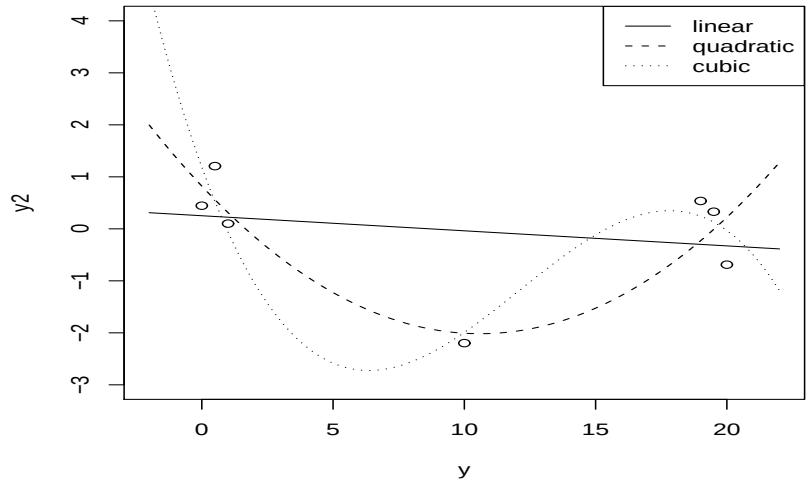

Fig. 3.8 Plots of linear (solid), quadratic (dashes), and cubic (dots) regression curves.

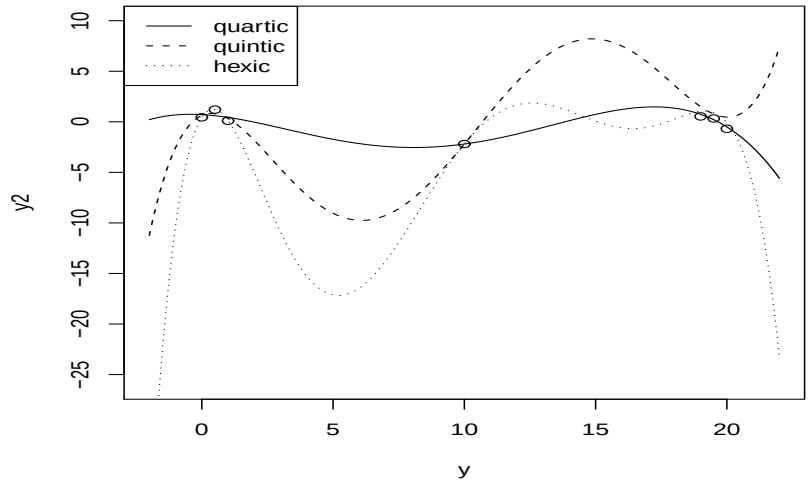

Fig. 3.9 Plots of quartic (solid), quintic (dashes), and hexic (dots) regression curves.

tables for all of the polynomial models. Our original data were a sample from a $N(0,1)$ distribution. The data were constructed with no regression structure so the best estimate of the variance comes from the total line and is $7.353/6 = 1.2255$. This value is a reasonable estimate of the true value 1. The *MSE* from the simple linear regression model also provides a reasonable estimate of $\sigma^2 = 1$. The larger models do not work as well. Most have variance estimates near 0.5, while the hexic model does not even allow an estimate of σ^2 because it fits every data point perfectly. By fitting models that are too large it seems that one can often make the *MSE* artificially small. For example, the quartic model has a *MSE* of 0.306 and an F statistic of 5.51; if it were not for the small value of dfE, such an F value would be highly significant. *If you find a large model that has an unnaturally small MSE with a reasonable number of degrees of freedom, everything can appear to be significant even though nothing you look at is really significant.*

Table 3.3 Analysis of variance tables.

Simple linear regression

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	1	0.457	0.457	0.33	0.59
Error	5	6.896	1.379		
Total	6	7.353			

Quadratic model

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	2	5.185	2.593	4.78	0.09
Error	4	2.168	0.542		
Total	6	7.353			

Cubic model

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	3	5.735	1.912	3.55	0.16
Error	3	1.618	0.539		
Total	6	7.353			

Quartic model

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	4	6.741	1.685	5.51	0.16
Error	2	0.612	0.306		
Total	6	7.353			

Quintic model

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	5	6.856	1.371	2.76	0.43
Error	1	0.497	0.497		
Total	6	7.353			

Hexic model

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	6	7.353	1.2255	—	—
Error	0	0.000	—		
Total	6	7.353			

Just as the mean squared error often gets unnaturally small when fitting large models, R^2 gets unnaturally large. As we have seen, there can be no possible reason to use a larger model than the quadratic with its R^2 of 0.71 for these 7 data points, but the cubic, quartic, quintic, and hexic models have R^2s of 0.78, 0.92, 0.93, and 1, respectively. □

3.4 Additional Spanning Functions

In a SLR, one method for testing lack of fit was to fit a larger polynomial model. In particular, for the Hooker data we fit a fifth-degree polynomial,

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \beta_5 x_i^5 + \varepsilon_i.$$

There was no particularly good reason to fit a fifth-degree, rather than a third-degree or seventh-degree polynomial. We just picked a polynomial that we hoped would be larger than we needed.

Rather than expanding the SLR model by adding polynomial terms, we can add other functions of x to the model. Commonly used functions are often simplified if we rescale x into a new variable taking values between 0 and 1, say, $\tilde{x}$. Commonly used functions are trig. functions, so we might fit a full model consisting of

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 \cos(\pi \tilde{x}_i) + \beta_3 \sin(\pi \tilde{x}_i) + \beta_4 \cos(\pi 2 \tilde{x}_i) + \beta_5 \sin(\pi 2 \tilde{x}_i) + \varepsilon_i \quad (1)$$

or a full model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 \cos(\pi \tilde{x}_i) + \beta_3 \cos(\pi 2 \tilde{x}_i) + \beta_4 \cos(\pi 3 \tilde{x}_i) + \beta_5 \cos(\pi 4 \tilde{x}_i) + \varepsilon_i. \quad (2)$$

As with the polynomial models, the number of additional predictors to add depends on how complicated the data are. For the purpose of testing lack of fit, we simply need the number to be large enough to find any salient aspects of the data that are not fitted well by the SLR model.

Another approach is to add a number of indicator functions. An *indicator function* of a set A is defined as

$$\mathscr{I}_A(\theta) = \begin{cases} 1 & \text{if } \theta \in A \\ 0 & \text{if } \theta \notin A \end{cases}. \quad (3)$$

We can fit models like

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 \mathscr{I}_{[0,.25)}(\tilde{x}_i) + \beta_3 \mathscr{I}_{[.25,.5)}(\tilde{x}_i) + \beta_4 \mathscr{I}_{[.5,.75)}(\tilde{x}_i) + \beta_5 \mathscr{I}_{[.75,1]}(\tilde{x}_i) + \varepsilon_i.$$

Adding indicator functions of length 2^{-j} defined on $\tilde{x}$ is equivalent to adding *Haar wavelets* to the model, cf. Chapter 7. Unfortunately, no regression programs will fit this model because it is no longer a regression model. It is no longer a regression model because there is a redundancy in the predictor variables. The model includes an intercept, which corresponds to a predictor variable that always takes on the value 1. However, if we add together our four indicator functions, their sum is also a variable that always takes on the value 1. To evade this problem, we need either to delete one of the indicator functions (doesn't matter which one) or remove the intercept from the model. Dropping the last indicator is convenient, so we fit

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 \mathscr{I}_{[0,.25)}(\tilde{x}_i) + \beta_3 \mathscr{I}_{[.25,.5)}(\tilde{x}_i) + \beta_4 \mathscr{I}_{[.5,.75)}(\tilde{x}_i) + \varepsilon_i. \quad (4)$$

Any continuous function defined on an interval $[a,b]$ can be approximated arbitrarily well by a sufficiently large polynomial. Similar statements can be made about the other classes of functions introduced here. Because of this, these classes of functions are known as *basis functions*. (Although they should really be known as *spanning functions*.)

EXAMPLE 3.4.1. We illustrate the methods on the Hooker data. With x the temperature, we defined $\tilde{x} = (x - 180.5)/30.5$. Fitting model (1) gives

Analysis of Variance: Sines and Cosines.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	5	447.185	89.437	3364.82	0.000
Residual Error	25	0.665	0.0266		
Total	30	447.850			

A test of whether model (1) fits significantly better than SLR has statistic

$$F_{obs} = \frac{(3.68 - 0.665)/(29 - 25)}{0.0266} = 28.4.$$

Clearly the reduced model of a simple linear regression fits worse than the model with two additional sine and cosine terms.

Fitting model (2) gives

Analysis of Variance: Cosines.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	5	447.208	89.442	3486.60	0.000
Residual Error	25	0.641	0.0257		
Total	30	447.850			

A test of whether the cosine model fits significantly better than SLR has statistic

$$F_{obs} = \frac{(3.68 - 0.641)/(29 - 25)}{0.0257} = 29.6.$$

Clearly the reduced model of a simple linear regression fits worse than the model with four additional cosine terms.

Fitting model (4) gives

Analysis of Variance: Haar Wavelets.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	4	446.77	111.69	2678.37	0.000
Residual Error	26	1.08	0.0417		
Total	30	447.85			

A test of whether this Haar wavelet model fits significantly better than SLR has statistic

$$F_{obs} = \frac{(3.68 - 1.08)/(29 - 26)}{0.0417} = 20.8.$$

Clearly the reduced model of a simple linear regression fits worse than the model with three additional indicator functions. □

3.4.1 High-order models

For spanning functions that are continuous, like the trig functions, high-order models can behave as strangely between the data points as polynomials. For example, Figure 3.10 contains a plot of the 7 data points discussed in Section 3.3 and, using $\tilde{x} = x/20$, a fitted cosine model with 5 terms and an intercept,

$$y_i = \beta_0 + \beta_1 \cos(\pi \tilde{x}_i) + \beta_2 \cos(\pi 2 \tilde{x}_i) + \beta_3 \cos(\pi 3 \tilde{x}_i) + \beta_4 \cos(\pi 4 \tilde{x}_i) + \beta_5 \cos(\pi 5 \tilde{x}_i) + \varepsilon_i.$$

The fit away from the data is even worse than for fifth- and sixth-order polynomials.

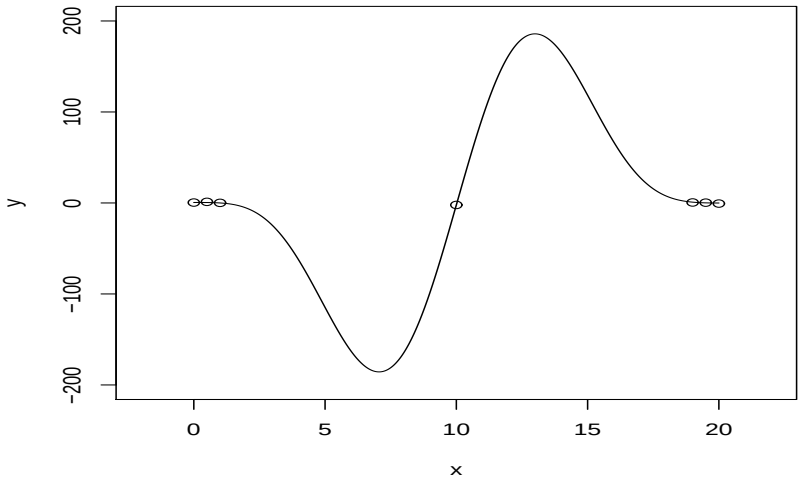

Fig. 3.10 Plot of fifth-order cosine model.

3.5 Partitioning

The basic idea of the partitioning method is quite simple. Suppose we are fitting a simple linear regression but that the actual relationship between x and y is a quadratic. If you can split the x values into two parts near the maximum or minimum of the quadratic, you can get a much better approximate fit using two lines instead of one. More generally, the idea is that an approximate model should work better on a smaller set of data that has predictor variables that are more similar. Thus,

if the original model is wrong, we should get a better approximation to the truth by fitting the original model on a series of smaller subsets of the data. Of course if the original model is correct, it should work about the same on each subset as it does on the complete data. The statistician partitions the data into disjoint subsets, fits the original model on each subset, and compares the overall fit of the subsets to the fit of the original model on the entire data. The statistician is free to select the partitions, including the number of distinct sets, but the subsets need to be chosen based on the predictor variable(s) alone.

EXAMPLE 3.5.1. We illustrate the partitioning method by splitting the Hooker data into two parts. Our partition sets are the data with the 16 smallest temperatures and the data with the 15 largest temperatures. We then fit a separate regression line to each partition. The two fitted lines are given in Figure 3.11. The ANOVA table is

Analysis of Variance: Partitioned Hooker data.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	3	446.66	148.89	3385.73	0.000
Error	27	1.19	0.04		
Total	30	447.85			

A test of whether this partitioning fits significantly better than SLR has statistic

$$F_{obs} = \frac{(3.68 - 1.19)/(29 - 27)}{0.04} = 31.125.$$

Clearly the reduced model of a simple linear regression fits worse than the model with two SLRs. Note that this is a simultaneous test of whether the slopes and intercepts are the same in each partition. □

3.5.1 Fitting the partitioned model

We now consider three different ways to fit this partitioned model. Our computations will be subject to some round-off error. One way to fit this model is simply to divide the data into two parts and fit a simple linear regression to each one. Fitting the lowest 16 x (temperature) values gives

Table of Coefficients: Low x values.

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	−50.725	2.596	−19.54	0.000
x-low	0.36670	0.01404	26.13	0.0001

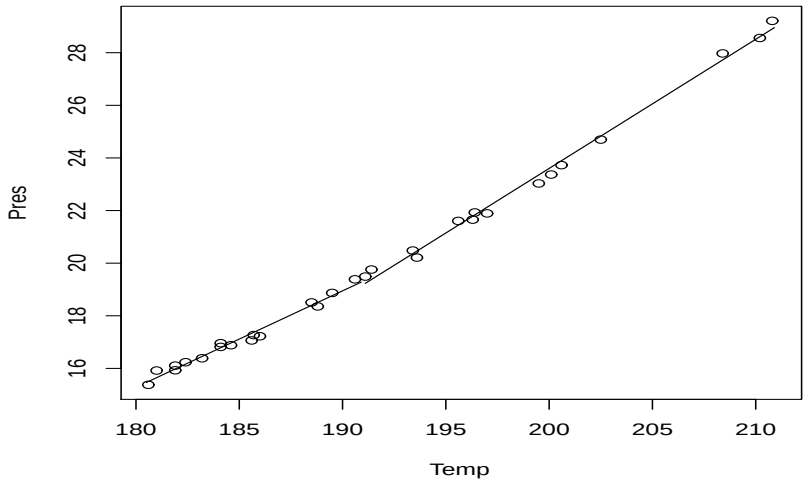

Fig. 3.11 Hooker data, partition method.

Analysis of Variance: Low x values.

Source	df	SS	MS	F	P
Regression	2	4687.1	2342.5	81269.77	0.000
Error	14	0.4	0.0		
Total	16	4687.5			

To get some extra numerical accuracy, from the F statistic we can compute $MSE = 2342.5/81269.77 = 0.028836$ so $SSE = 0.4037$. All we really care about in this ANOVA table is the error line but the table itself is unusual. In order to get this three-line anova table from most software you would need to tell the software to fit the model with no intercept but then manually enter an intercept by incorporating a predictor variable that always takes the value 1. In this anova table the 2 df for regression are for both the intercept and slope and the 16 degrees of freedom total are for the 16 observations. The sum of squares total is just the sum of the squares of the 16 low y_i values.

Similarly fitting the highest 15 x values gives

Table of Coefficients: High x values.

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	−74.574	2.032	−36.70	0.000
x-high	0.49088	0.01020	48.12	0.000

Analysis of Variance: High x values.

Source	df	SS	MS	F	P
Regression	2	8193.9	4096.9	67967.66	0.000
Error	13	0.8	0.1		
Total	15	8194.7			

Again, from the F statistic $MSE = 4096.9/67967.66 = 0.060277$, so $SSE = 0.7836$. The variance estimate for the overall model is obtained by pooling the two Error terms to give $dfE(Full) = 14+13 = 27$, $SSE(Full) = 0.4037+0.7836 = 1.1873$, with $MSE(Full) = 0.044$.

A more efficient way to proceed is to fit both simple linear regressions at once. Construct a variable h that identifies the 15 high values of x. In other words, h is 1 for the 15 highest temperature values and 0 for the 16 lowest values. Define $x_1 = h \times x$, $h_2 = 1-h$, and $x_2 = h_2 \times x$. Fitting these four variables in a *regression through the origin*, i.e., fitting

$$y_i = \beta_1 h_{i2} + \beta_2 x_{i2} + \beta_3 h_i + \beta_4 x_{i1} + \varepsilon_i,$$

gives

Table of Coefficients: Separate lines.

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
h_2	−50.725	3.205	−15.82	0.000
x_2	0.36670	0.01733	21.16	0.000
h	−74.574	1.736	−42.97	0.000
x_1	0.490875	0.008712	56.34	0.000

Analysis of Variance: Separate lines.

Source	df	SS	MS	F	P
Regression	4	12881.0	3220.2	73229.01	0.000
Error	27	1.2	0.0		
Total	31	12882.2			

Note that these regression estimates agree with those obtained from fitting each set of data separately. The standard errors differ because here we are pooling the information in the error rather than using separate estimates of σ^2 from each subset of data. Although the ANOVA table reports $MSE = 0.0$, we can see that it actually agrees with earlier calculations by noting that $MSE = MSReg/F = 0.04397$.

The way the model was originally fitted for our discussion was regressing on x, h, and x_1, i.e., fitting

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 h_i + \beta_3 x_{i1} + \varepsilon_i. \tag{1}$$

This is a model that has the low group of temperature values as a baseline and for the high group incorporates deviations from the baseline. The ANOVA table gives the same Error as the previous table and the table of regression coefficients is

Table of Coefficients: Low group baseline.

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	−50.725	3.205	−15.82	0.000
x	0.36670	0.01733	21.16	0.000
h	−23.849	3.645	−6.54	0.000
x_1	0.12418	0.01940	6.40	0.000

The slope for the low group is 0.36670 and for the high group it is $0.36670 + 0.12418 = 0.49088$. The t test for whether the slopes are different, in a model that retains separate intercepts, is based on the x_1 row of this table and has $t = 6.40$. The intercepts also look different. The estimated intercept for the low group is -50.725 and for the high group it is $-50.725 + (-23.849) = -74.574$. The t test for whether the intercepts are different, in a model that retains separate slopes, is based on the h row and has $t = -6.54$.

*3.5.2 Output for categorical predictors**

At the beginning of Chapter 1 we discussed the fact that predictor variables can be of two types: continuous or categorical. Regression analysis and computer programs for regression analysis consider only continuous variables. Various programs for fitting *linear models* (as distinct from fitting regression) handle both types of variables. Of the packages discussed on my website, R's command lm and SAS's PROC GENMOD treat all (numerical) variables as continuous unless otherwise specified. In particular, if no variables are specified as categorical, both lm and GENMOD act as regression programs. Minitab's glm, on the other hand, treats all variables as categorical (factors) unless otherwise specified. Not only are the defaults different, but how the programs deal with categorical variables differs. Since partitioning the data defines categories, we have cause to introduce these issues here. Categorical variables are ubiquitous when discussing ANOVA.

In our partitioning example, x is continuous but h is really a categorical variable indicating which points are in the high group. When a categorical variable has only two groups, or more specifically, if it is a 0-1 indicator variable like h (or h_2), it can be treated the same way that continuous variables are treated in regression software. Indeed, we have exploited that fact up to this point. The remainder of this subsection discusses how various software treat variables that are identified as factors.

As indicated earlier, R's lm command and SAS's PROC GENMOD both have x defaulting to a continuous variable but h can be specified as a factor. Minitab's glm output has h defaulting to a factor but x must be specified as a covariate. In all of them we fit a model that specifies effects for each variable plus we fit an "interaction" between the two variables. To mimic these procedures using regression, we need to construct and use variables h_2, x_1, x_2 and two new variables h_3, x_3. One advantage of specifying h as a factor variable is that you do not have to construct any new variables.

R's lm program with h as a factor, essentially, fits model (1), i.e., a model that uses the low temperatures as a baseline. The output is the same as the regression output that we already examined.

SAS's PROC GENMOD with h as a classification variable (factor), essentially, fits a model that uses the high group as the baseline, that is, it fits

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 h_{i2} + \beta_3 x_{i2} + \varepsilon_i.$$

For the low group, the model incorporates deviations from the baseline. The three-line ANOVA table does not change from model (1) but the table of regression coefficients is

Table of Coefficients: High group baseline.

Predictor	$\hat{\beta}_k$	$\mathrm{SE}(\hat{\beta}_k)$	t	P
Constant	-74.574	1.736	-42.97	0.000
x	0.49088	0.00871	56.34	0.000
h_2	23.849	3.645	6.54	0.000
x_2	-0.12418	0.01940	-6.40	0.000

The estimated slope for the high group is 0.49088 and for the low group it is $0.49088 + (-0.12418) = 0.36670$. The t test for whether the slopes are different, in a model that retains separate intercepts, is based on the x_2 row of this table and has $t = -6.40$. The intercepts also look different. The estimated intercept for the high group is -74.574 and for the low group it is $-74.574 + 23.849 = -50.725$. The t test for whether the intercepts are different, in a model that retains separate slopes, is based on the h_2 row and has $t = 6.54$.

The following table is how PROC GENMOD reports these results.

Table of Coefficients: SAS PROC GENMOD.

					95%			
Predictor		df	$\hat{\beta}_k$	$\mathrm{SE}_m(\hat{\beta}_k)$	Conf.	Limits	t^2	P
Intercept		1	-74.5741	1.6198	-77.7489	-71.3993	2119.55	$< .0001$
h	0	1	23.8490	3.4019	17.1815	30.5166	49.15	$< .0001$
h	1	0	0.0000	0.0000	0.0000	0.0000	.	
x		1	0.4909	0.0081	0.4749	0.5068	3644.94	$< .0001$
x *h	0	1	-0.1242	0.0181	-0.1597	-0.0887	47.04	$< .0001$
x *h	1	0	0.0000	0.0000	0.0000	0.0000	.	
Scale		1	0.1957	0.0249	0.1526	0.2510		

While the parameter estimates agree in obvious ways, the standard errors are different from the regression output. The coefficients for the highest level of the factor h are forced to be zero (R does this for the lowest level of h) and the corresponding standard errors are 0 because estimates that have been forced to be zero have no variability. The nonzero standard errors are also different in GENMOD because they are not based on the MSE but rather the maximum likelihood estimate of the variance,

$$\hat{\sigma}^2 \equiv \frac{SSE}{n}.$$

We used the notation $\mathrm{SE}_m(\hat{\beta}_k)$ with a subscript of m to indicate this difference. The relationship between the standard errors is

$$\mathrm{SE}(\hat{\beta}_k) = \frac{\sqrt{n}}{\sqrt{dfE}}\mathrm{SE}_m(\hat{\beta}_k).$$

Note also that GENMOD gives t^2 rather than t, provides 95% confidence intervals, and reports very small P values in a more appropriate fashion than merely reporting 0.0000. SAS also has a PROC GLM procedure that will fit the model, but it does not readily report parameter estimates.

R and SAS use variations on a theme, i.e., fix a baseline group. Minitab takes a different course. Minitab, essentially, defines variables $h_3 = h_2 - h$ and $x_3 = x \times h_3$ and fits

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 h_{i3} + \beta_3 x_{i3} + \varepsilon_i.$$

This gives the regression coefficients

Table of Coefficients.

Predictor	$\hat{\beta}_k$	$\mathrm{SE}(\hat{\beta}_k)$	t	P
Constant	−62.64962	1.82259	−34.374	0.000
x	0.42879	0.00970	44.206	0.000
h3	11.92452	1.82259	6.543	0.000
x3	−0.06209	0.00970	−6.401	0.000

Minitab's glm yields the following output for coefficients.

Table of Coefficients: Minitab glm.

Predictor	$\hat{\beta}_k$	$\mathrm{SE}(\hat{\beta}_k)$	t	P
Constant	−62.650	1.823	−34.37	0.000
h				
0	11.925	1.823	6.54	0.000
x	0.428787	0.009700	44.21	0.000
x*h				
0	−0.062089	0.009700	−6.40	0.000

provided you ask Minitab to provide coefficients for all terms. (The default does not give coefficients associated with h.) The "constant" value of −62.650 is the average of the two intercept estimates that were reported earlier for the separate lines. The intercept for the low group ($h = 0$) is $-62.650 + 11.925$ and the intercept for the high group is $-62.650 - 11.925$. Note that the t test for "h 0" is the same 6.54 that was reported earlier for testing whether the intercepts were different. Minitab is fitting effects for both $h = 0$ and $h = 1$ but forcing them to sum to zero, rather than what R and SAS do, which is picking a level of h and forcing the effect of that level to be zero (hence making it the baseline). Similarly, the "x" value 0.428787 is

the average of the two slope estimates reported earlier. The slope for the low group ($h = 0$) is $0.428787 + (-0.062089)$ and the slope for the high group is $0.428787 - (-0.062089)$. The t test for "x*h 0" is the same -6.40 as that reported earlier for testing whether the slopes were different. Minitab provides coefficient output that is more traditional than either R or SAS, but is often more difficult to interpret. However, given the wide variety of software and output that one may be confronted with, it is important to be able to cope with all of it.

Our discussion used the variable h that partitions the data into the smallest 16 observations and the largest 15 observations. Minitab's regression program provides a lack-of-fit test that partitions the data into the 18 observations below $\bar{x}_{\cdot} = 191.79$ and the 13 observations larger than the mean. Their test gets considerably more complicated when there is more than one predictor variable. They perform both this test (in more complicated situations, these tests) and a version of the test described in the next subsection, and combine the results from the various tests.

3.5.3 Utts' method

Utts (1982) proposed a lack-of-fit test based on comparing the original (reduced) model to a full model that consists of fitting the original model on a subset of the original data. In other words, you fit the model on all the data and test that against a full model that consists of fitting the model on a subset of the data. The subset is chosen to contain the points closest to $\bar{x}_{\cdot}$. Although it seems like fitting the model to a reduced set of points should create a reduced model, just the opposite is true. To fit a model to a reduced set of points, we can think of fitting the original model and then adding a separate parameter for every data point that we want to exclude from the fitting procedure. In fact, that is what makes this a partitioning method. There is one subset that consists of the central data and the rest of the partition has every data point in a separate set.

The central subset is chosen to be a group of points close to $\bar{x}_{\cdot}$. With only one predictor variable, it is easy to determine a group of central points. As mentioned earlier, for models with an intercept the leverages are really measures of distance from $\bar{x}_{\cdot}$; cf. *PA*, so even with more predictor variables, one could choose a group of points that have the lowest leverages in the original model.

EXAMPLE 3.5.1. We consider first the use of 15 central points with leverages below 0.05; about half the data. We then consider a group of 6 central points; about a fifth of the data.

The ANOVA table when fitting a simple linear regression to 15 central points is

Analysis of Variance: 15 central points.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	1	40.658	40.658	1762.20	0.000
Error	13	0.300	0.023		
Total	14	40.958			

The lack-of-fit test against a reduced model of simple linear regression on the entire data has

$$F_{obs} = \frac{(3.68 - 0.300)/(29 - 13)}{0.023} = 9.18,$$

which is highly significant. Figure 3.12 illustrates the fitting method.

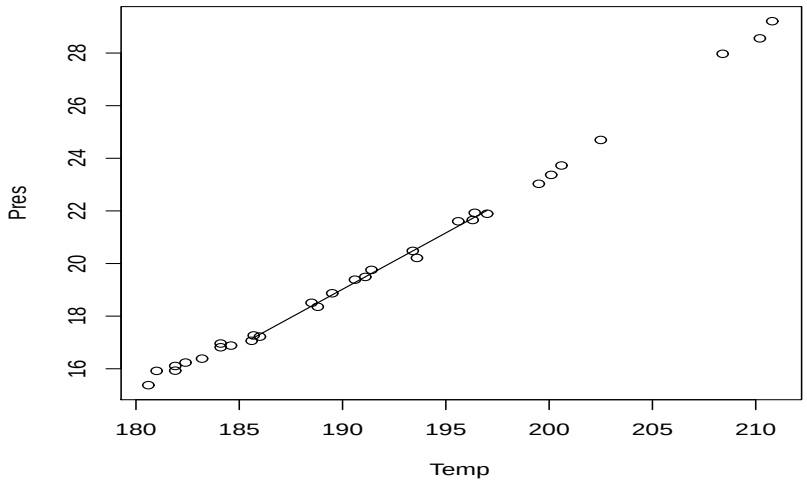

Fig. 3.12 Hooker data, Utts' method with 15 points.

When using 6 central points having leverages below 0.035, the ANOVA table is

Analysis of Variance: 6 central points.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	1	1.6214	1.6214	75.63	0.001
Error	4	0.0858	0.0214		
Total	5	1.7072			

and the F statistic is

$$F_{obs} = \frac{(3.68 - 0.0858)/(29 - 4)}{0.0214} = 6.72.$$

This is much bigger than 1 and easily significant at the 0.05 level. Both tests suggest lack of fit. Figure 3.13 illustrates the fitting method. □

My experience is that Utt's test tends to work better with relatively small groups of central points. (Even though the F statistic here was smaller for the smaller

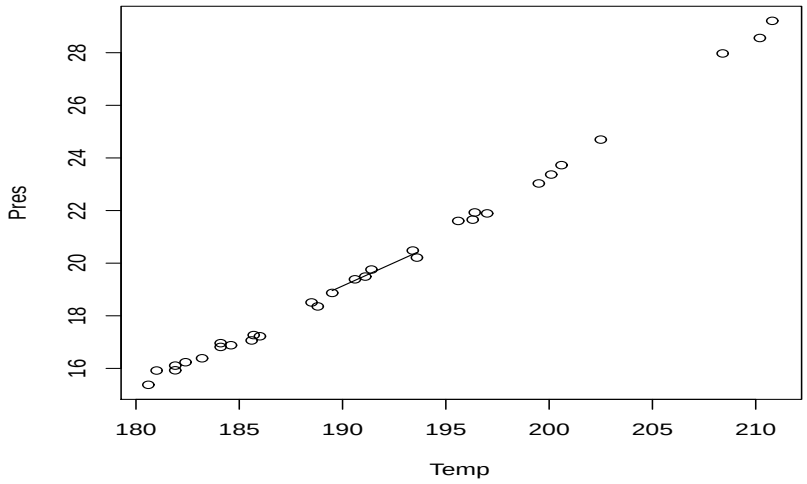

Fig. 3.13 Hooker data, Utts' method with 6 points.

group.) Minitab's regression program incorporates a version of Utt's test that defines the central region as those points with leverages less than $1.1p/n$ where p is the number of regression coefficients in the model, so for a simple linear regression $p = 2$. For these data, their central region consists of the 22 observations with temperature between 183.2 and 200.6.

3.6 Splines

When fitting a polynomial to a single predictor variable, the partitioning method is extremely similar to the nonparametric regression method known as fitting *splines*. When using partitioning to test for lack of fit, our fitting of the model on each subset was merely a device to see whether the original fitted model gave better approximations on smaller subsets of the data than it did overall. The only difference when fitting splines is that we take the results obtained from fitting on the partition sets seriously as a model for the regression function. As such, we typically do not want to allow discontinuities in the regression function at the partition points (known as "knots" in spline theory), so we include conditions that force continuity. Typically when fitting splines one uses a large number of partition sets, so there are a large number of conditions to force continuity. We illustrate the ideas on the Hooker data with only two partition sets and go into more detail in Chapter 7. Generalizations are available for more than one predictor variable; see Chapter 7 and Wahba (1990).

EXAMPLE 3.6.1. *Hooker data.*
Again, our partition sets are the data with the 16 smallest temperatures and the data with the 15 largest temperatures. Referring back to Table 3.1 we see that the partition point must be somewhere between 190.6 and 191.1. For convenience, let's set the partition point at 191. We model a separate regression line for each partition,

$$m(x) = \begin{cases} \beta_1 + \beta_2 x & \text{if } x \leq 191 \\ \beta_3 + \beta_4 x & \text{if } x > 191 \, . \end{cases}$$

Fitting two regression lines was discussed in Subsection 3.5.1 where we found the estimated lines

$$\hat{m}(x) = \begin{cases} -50.725 + 0.36670x & \text{if } x \leq 191 \\ -74.574 + 0.490875x & \text{if } x > 191 \, . \end{cases}$$

The two fitted lines were displayed in Figure 3.11.

To change this into a linear spline model, we need the two lines to match up at the knot, that is, we need to impose the continuity condition that

$$\beta_1 + \beta_2 191 = \beta_3 + \beta_4 191.$$

The condition can be rewritten in many ways but we will use

$$\beta_3 = \beta_1 + \beta_2 191 - \beta_4 191.$$

You can see from Figure 3.11 that the two separate fitted lines are already pretty close to matching up at the knot.

In Subsection 3.5.1 we fitted the partitioned model as a single linear model in two ways. The first was more transparent but the second had advantages. The same is true about the modifications needed to generate linear spline models. To begin, we constructed a variable h that identifies the 15 high values of x. In other words, h is 1 for the 15 highest temperature values and 0 for the 16 lowest values. We might now write

$$h(x) \equiv \mathscr{I}_{(191,\infty)}(x),$$

where we again use the indicator function introduced in Section 4. With slightly different notation for the predictor variables, we first fitted the two separate lines model as

$$y_i = \beta_1[1 - h(x_i)] + \beta_2 x_i[1 - h(x_i)] + \beta_3 h(x_i) + \beta_4 x_i h(x_i) + \varepsilon_i.$$

Imposing the continuity condition by substituting for β_3, the model becomes

$$y_i = \beta_1[1 - h(x_i)] + \beta_2 x_i[1 - h(x_i)] + \{\beta_1 + \beta_2 191 - \beta_4 191\} h(x_i) + \beta_4 x_i h(x_i) + \varepsilon_i$$

or

$$y_i = \beta_1 \{[1 - h(x_i)] + h(x_i)\} + \beta_2 \{x_i[1 - h(x_i)] + 191 h(x_i)\}$$

$$+\beta_4\left[x_i h(x_i) - 191h(x_i)\right] + \varepsilon_i$$

or

$$y_i = \beta_1 + \beta_2\left\{x_i[1-h(x_i)] + 191h(x_i)\right\} + \beta_4\left(x_i - 191\right)h(x_i) + \varepsilon_i, \tag{1}$$

where now β_1 is an overall intercept for the model.

As mentioned earlier, the two-lines model was originally fitted (with different symbols for the unknown parameters) as

$$y_i = \beta_1 + \beta_2 x_i + \gamma_1 h(x_i) + \gamma_2 x_i h(x_i) + \varepsilon_i.$$

This is a model that has the low group of temperature values as a baseline and for the high group incorporates deviations from the baseline, e.g., the slope above 191 is $\beta_2 + \gamma_2$. For this model the continuity condition is that

$$\beta_1 + \beta_2 191 = \beta_1 + \beta_2 191 + \gamma_1 + \gamma_2 191$$

or that

$$0 = \gamma_1 + \gamma_2 191$$

or that

$$\gamma_1 = -\gamma_2 191.$$

Imposing this continuity condition, the model becomes

$$y_i = \beta_1 + \beta_2 x_i - \gamma_2 191 h(x_i) + \gamma_2 x_i h(x_i) + \varepsilon_i$$

or

$$y_i = \beta_1 + \beta_2 x_i + \gamma_2\left(x_i - 191\right)h(x_i) + \varepsilon_i. \tag{2}$$

In discussions of splines, the function $(x_i - 191)\,h(x_i)$ is typically written $(x_i - 191)_+$ where for any scalar a,

$$(x-a)_+ \equiv \begin{cases} x-a & \text{if } x > a \\ 0 & \text{if } x \le a. \end{cases}$$

Fitting models (1) and (2) to the Hooker data gives

Table of Coefficients: Model (1).

Predictor	*Est*	SE(*Est*)	*t*	*P*
Constant	−48.70931	2.252956	−21.62	0.000
$x[1-h(x)] + 191h(x)$	0.35571	0.012080	29.45	0.000
$(x-191)_+$	0.48717	0.007619	63.95	0.000

and

Table of Coefficients: Model (2).

Predictor	Est	SE(Est)	t	P
Constant	−48.70931	2.25296	−21.620	0.000
x	0.35571	0.01208	29.447	0.000
$(x-191)_+$	0.13147	0.01751	7.509	0.000

Notice that the slope for x values above 191, $\hat{\beta}_4 = 0.48717$, equals the slope below 191 plus the change in slopes, $\hat{\beta}_2 + \hat{\gamma}_2 = 0.35571 + 0.13147$, there being round-off error in the last digit.

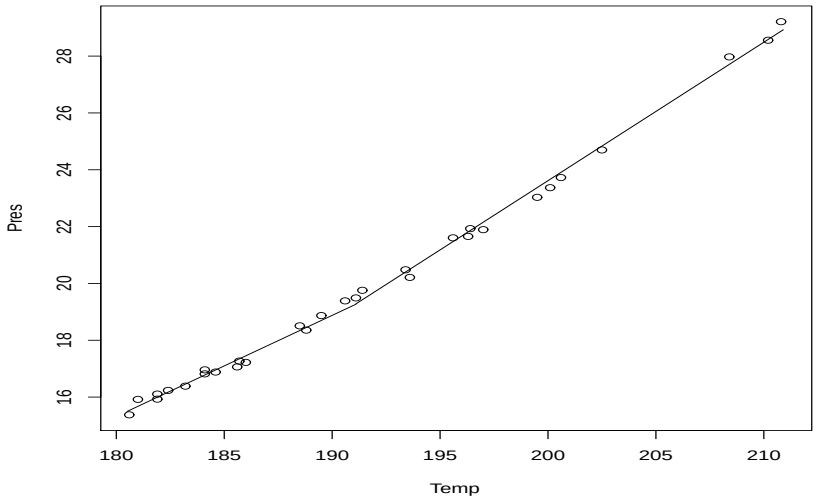

Fig. 3.14 Hooker data, linear spline with one knot at 191.

Both models give $dfE = 28$, $SSE = 1.2220$, and $MSE = 0.04364$. We can even use the linear spline model as the basis for a lack-of-fit test of the simple linear regression on the Hooker data,

$$F_{obs} = \frac{(3.6825 - 1.2220)/(29-28)}{0.04364} = 56.38.$$

Obviously, fitting different lines on each partition set is a more general model than fitting the same line on each partition set. But since fitting a single line to all the data gives continuity at each knot, fitting different lines on each partition set and forcing them to be continuous is still a more general model than fitting the same line on all the data. □

In general, to fit a linear spline model, you need to decide on a group of knots at which the slope will change. Call these $\tilde{x}_j$, $j = 1, \ldots, r$. The linear spline model then becomes

$$y_i = \beta_0 + \beta_1 x_i + \sum_{j=1}^{r} \gamma_j (x_i - \tilde{x}_j)_+ + \varepsilon_i.$$

Similar ideas work with higher-degree polynomials. The most popular polynomial to use is cubic; see Exercise 3.7.8. The general cubic spline model is

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x^3 + \sum_{j=1}^{r} \gamma_j [(x_i - \tilde{x}_j)_+]^3 + \varepsilon_i.$$

See Chapter 7 for details.

3.7 Fisher's Lack-of-Fit Test

We now introduce Fisher's lack-of-fit test for the Hooker data. For now, notice that the predictor variable includes two replicate temperatures: $x = 181.9$ with y values 15.106 and 15.928, and $x = 184.1$ with y values 16.959 and 16.817. In this case, the computation for Fisher's lack-of-fit test is quite simple. We use the replicated x values to obtain a measure of pure error. First, compute the sample variance of the y_is at each replicated x value. There are 2 observations at each replicated x, so the sample variance computed at each x has 1 degree of freedom. Since there are two replicated xs each with one degree of freedom for the variance estimate, the pure error has $1 + 1 = 2$ degrees of freedom. To compute the sum of squares for pure error, observe that when $x = 181.9$, the mean y is 15.517. The contribution to the sum of squares pure error from this x value is $(15.106 - 15.517)^2 + (15.928 - 15.517)^2$. A similar contribution is computed for $x = 184.1$ and they are added to get the sum of squares pure error. The degrees of freedom and sum of squares for lack of fit are found by taking the values from the original error and subtracting the values for the pure error. The F test for lack of fit examines the mean square lack of fit divided by the mean square pure error.

Analysis of Variance.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	1	444.17	444.17	3497.89	0.000
Error	29	3.68	0.13		
(Lack of Fit)	27	3.66	0.14	10.45	0.091
(Pure Error)	2	0.03	0.01		
Total	30	447.85			

The F statistic for lack of fit, 10.45, seems substantially larger than 1, but because there are only 2 degrees of freedom in the denominator, the P value is a relatively large 0.09. This method is closely related to one-way analysis of variance and is discussed in more detain in both *ANREG* and *PA*.

3.8 Additive Effects Versus Interaction

For the *Coleman Report* data, one of the viable models had two predictors: x_3, socioeconomic status, and x_4, teacher's verbal score. If the model displayed lack of fit, there are a number of ways that we could expand the model.

In general, the simplest multiple regression model for $\mathrm{E}(y)$ based on two predictors is

$$m(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2. \tag{1}$$

This model displays *additive effects*. The relative effect of changing the value of variable x_1 into, say, $\tilde{x}_1$ is the same, regardless of the value of x_2. Specifically,

$$[\beta_0 + \beta_1 \tilde{x}_1 + \beta_2 x_2] - [\beta_0 + \beta_1 x_1 + \beta_2 x_2] = \beta_2(\tilde{x}_1 - x_1).$$

This effect does not depend on x_2, which allows us to speak about an effect for x_1. If the effect of x_1 depends on x_2, no single effect for x_1 exists and we would always need to specify the value of x_2 before discussing the effect of x_1. An exactly similar argument shows that in model (1) the effect of changing x_2 does not depend on the value of x_1.

Generally, for any two predictors x_1 and x_2, an *additive effects (no-interaction) model* takes the form

$$m(x) = h_1(x_1) + h_2(x_2) \tag{2}$$

where $x = (x_1, x_2)$ and $h_1(\cdot)$ and $h_2(\cdot)$ are arbitrary functions. In this case, the relative effect of changing x_1 to $\tilde{x}_1$ is the same for any value of x_2 because

$$m(\tilde{x}_1, x_2) - m(x_1, x_2) = [h_1(\tilde{x}_1) + h_2(x_2)] - [h_1(x_1) + h_2(x_2)] = h_1(\tilde{x}_1) - h_1(x_1),$$

which does not depend on x_2. An exactly similar argument shows that the effect of changing x_2 does not depend on the value of x_1. In an additive model, the effect as x_1 changes can be anything at all; it can be any function h_1, and similarly for x_2. However, the combined effect must be the sum of the two individual effects. Other than model (1), the most common no-interaction models for two measurement predictors are probably a polynomial in x_1 plus a polynomial in x_2, say,

$$m(x) = \beta_0 + \sum_{r=1}^{R} \beta_{r0} x_1^r + \sum_{s=1}^{S} \beta_{0s} x_2^s. \tag{3}$$

An *interaction model* is literally any model that does not display the additive effects structure of (2). When generalizing no-interaction polynomial models, cross-product terms are often added to model interaction. For example, model (1) might be expanded to

$$m(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2.$$

This is an interaction model because the relative effect of changing x_1 to $\tilde{x}_1$ depends on the value of x_2. Specifically,

$$[\beta_0+\beta_1\tilde{x}_1+\beta_2x_2+\beta_3\tilde{x}_1x_2]-[\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_1x_2]=\beta_2(\tilde{x}_1-x_1)+\beta_3(\tilde{x}_1-x_1)x_2,$$

where the second term depends on the value of x_2. To include interaction, the no-interaction polynomial model (3) might be extended to an interaction polynomial model

$$m(x)=\sum_{r=0}^{R}\sum_{s=0}^{S}\beta_{rs}x_1^r x_2^s. \tag{4}$$

These devices are easily extended to more than two predictor variables.

EXAMPLE 3.8.1. Using the *Coleman Report* data, we begin by considering

$$y_h=\beta_0+\beta_3x_{h3}+\beta_4x_{h4}+\varepsilon_h,$$

which was fitted in Chapter 1. First we fit a simple quadratic additive model

$$y_h=\beta_0+\beta_{10}x_{h3}+\beta_{20}x_{h3}^2+\beta_{01}x_{h4}+\beta_{02}x_{h4}^2+\varepsilon_h.$$

From the table of coefficients

Table of Coefficients

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	38.0	106.5	0.36	0.726
x_3	0.54142	0.05295	10.22	0.000
x_3^2	−0.001892	0.006411	−0.30	0.772
x_4	−1.124	8.602	−0.13	0.898
x_4^2	0.0377	0.1732	0.22	0.831

we see that neither quadratic term is adding anything after the other terms because both quadratic terms have large P values. To make a simultaneous test of dropping the quadratic terms, we need to compare the error in the ANOVA table

Analysis of Variance

Source	df	SS	MS	F	P
Regression	4	571.47	142.87	29.99	0.000
Residual Error	15	71.46	4.76		
Total	19	642.92			

to the error given in Chapter 1. The F statistic becomes

$$F_{obs}=\frac{[72.43-71.46]/[17-15]}{71.46/15}=\frac{0.485}{4.76}=0.102,$$

so together the quadratic terms are contributing virtually nothing.

The simplest interaction model is

$$y_h=\beta_0+\beta_3x_{h3}+\beta_4x_{h4}+\beta_{34}x_{h3}x_{h4}+\varepsilon_h.$$

Fitting gives the table of coefficients.

Table of Coefficients

Predictor	$\hat{\beta}_k$	$\text{SE}(\hat{\beta}_k)$	t	P
Constant	10.31	10.48	0.98	0.340
x_3	1.900	1.569	1.21	0.244
x_4	0.9264	0.4219	2.20	0.043
x_3x_4	−0.05458	0.06304	−0.87	0.399

This shows no effect for adding the $\beta_{34}x_{h3}x_{h4}$ interaction ($P = 0.399$). Alternatively, we could compare the error from the ANOVA table

Analysis of Variance

Source	df	SS	MS	F	P
Regression	3	573.74	191.25	44.23	0.000
Residual Error	16	69.18	4.32		
Total	19	642.92			

to that given in Chapter 1 to get the F statistic

$$F_{obs} = \frac{[72.43 - 69.18]/[17 - 16]}{69.18/16} = \frac{3.25}{4.32} = 0.753 = (-0.87)^2,$$

which also gives the P value 0.399. □

3.9 Generalized Additive Models

Suppose we wanted to fit a cubic interaction model to the *Coleman Report* data. With five predictor variables, the model is

$$m(x) = \sum_{r=0}^{3}\sum_{s=0}^{3}\sum_{t=0}^{3}\sum_{u=0}^{3}\sum_{v=0}^{3}\beta_{rstuv}x_1^r x_2^s x_3^t x_4^u x_5^v \tag{1}$$

and includes $4^5 = 1024$ mean parameters β_{rstuv}. We might want to think twice about trying to estimate 1024 parameters from just 20 schools.

This is a common problem with fitting polynomial interaction models. When we have even a moderate number of predictor variables, the number of parameters quickly becomes completely unwieldy. And it is not only a problem for polynomial interaction models. In Section 4 we discussed replacing polynomials with other basis functions $\phi_r(x)$. The polynomial models happen to have $\phi_r(x) = x^r$. Other choices of ϕ_r include cosines, or both cosines and sines, or indicator functions, or wavelets. Typically, $\phi_0(x) \equiv 1$. In the basis function approach, the additive polynomial model (3.8.3) generalizes to

$$m(x) = \beta_0 + \sum_{r=1}^{R}\beta_{r0}\phi_r(x_1) + \sum_{s=1}^{S}\beta_{0s}\phi_s(x_2) \tag{2}$$

and the polynomial interaction model (3.8.4) generalizes to

$$m(x) = \sum_{r=0}^{R} \sum_{s=0}^{S} \beta_{rs} \phi_r(x_1) \phi_s(x_2). \tag{3}$$

When expanding model (3) to include more predictors, the generalized interaction model has exactly the same problem as the polynomial interaction model (1) in that it requires fitting too many parameters.

Generalized additive models provide a means for circumventing the problem. They do so by restricting the orders of the interactions. In model (1) we have five variables, all of which can interact with one another. Instead, suppose variables x_1 and x_4 can interact with one another but with no other variables and that variables x_2, x_3, and x_5 can interact with one another but with no other variables. We can then write a generalized additive model

$$m(x) \equiv m(x_1, x_2, x_3, x_4, x_5) = h_1(x_1, x_4) + h_2(x_2, x_3, x_5). \tag{4}$$

Using the basis function approach to model each of the two terms on the right gives

$$m(x) = \sum_{r=0}^{R} \sum_{u=0}^{U} \beta_{ru} \phi_r(x_1) \phi_u(x_4) + \sum_{s=0}^{S} \sum_{t=0}^{T} \sum_{v=0}^{V} \gamma_{stu} \phi_s(x_2) \phi_t(x_3) \phi_v(x_5) - \gamma_{000}.$$

We subtracted γ_{000} from the model because both β_{00} and γ_{000} serve as intercept terms, hence they are redundant parameters. This section started by considering the cubic interaction model (1) for the *Coleman Report* data. The model has $3 = R = S = T = U = V$ and involves 1024 mean parameters. Using similar cubic polynomials to model the generalized additive model (4) we need only $4^2 + 4^3 - 1 = 79$ parameters. While that is still far too many parameters to fit to the *Coleman Report* data, you can see that fitting generalized additive models are much more feasible than fitting full interaction models.

Another generalized additive model that we could propose for five variables is

$$m(x) = h_1(x_1, x_2) + h_2(x_2, x_3) + h_3(x_4, x_5).$$

A polynomial version of the model is

$$m(x) = \sum_{r=0}^{R} \sum_{s=0}^{S} \beta_{rs} x_1^r x_2^s + \sum_{s=0}^{S} \sum_{t=0}^{T} \gamma_{st} x_2^s x_3^t + \sum_{u=0}^{U} \sum_{v=0}^{V} \delta_{uv} x_4^u x_5^v. \tag{5}$$

In this case, not only are β_{00}, γ_{00}, and δ_{00} all redundant intercept parameters, but $\sum_{s=0}^{S} \beta_{0s} x_1^0 x_2^s$ and $\sum_{s=0}^{S} \gamma_{s0} x_2^s x_3^0$ are redundant simple polynomials in x_2. In this case it is more convenient to write model (5) as

$$m(x) = \sum_{r=0}^{R} \sum_{s=0}^{S} \beta_{rs} x_1^r x_2^s + \sum_{s=0}^{S} \sum_{t=1}^{T} \gamma_{st} x_2^s x_3^t + \sum_{u=0}^{U} \sum_{v=0}^{V} \delta_{uv} x_4^u x_5^v - \delta_{00}.$$

Of course, the catch with generalized additive models is that you need to have some idea of what variables may interact with one another. And the only obvious way to check that assumption is to test the assumed generalized additive model against the full interaction model. But this whole discussion started with the fact that fitting the full interaction model is frequently infeasible.

3.10 Exercises

EXERCISE 3.10.1. Dixon and Massey (1969) presented data on the relationship between IQ scores and results on an achievement test in a general science course. Table 3.4 contains a subset of the data. Fit the simple linear regression model of achievement on IQ and the quadratic model of achievement on IQ and IQ squared. Evaluate both models and decide which is the best.

Table 3.4 IQs and achievement scores.

IQ	Achiev.	IQ	Achiev.	IQ	Achiev.	IQ	Achiev.	IQ	Achiev.
100	49	105	50	134	78	107	43	122	66
117	47	89	72	125	39	121	75	130	63
98	69	96	45	140	66	90	40	116	43
87	47	105	47	137	69	132	80	101	44
106	45	95	46	142	68	116	55	92	50
134	55	126	67	130	71	137	73	120	60
77	72	111	66	92	31	113	48	80	31
107	59	121	59	125	53	110	41	117	55
125	27	106	49	120	64	114	29	93	50

EXERCISE 3.10.2. Use two methods other than fitting polynomial models to test for lack of fit in Exercise 3.10.1

EXERCISE 3.10.3. Based on the height and weight data given in Table 3.5, fit a simple linear regression of weight on height for these data and check the assumptions. Give a 99% confidence interval for the mean weight of people with a 72-inch height. Test for lack of fit of the simple linear regression model.

EXERCISE 3.10.4. Jensen (1977) and Weisberg (1985, p. 101) considered data on the outside diameter of crank pins that were produced in an industrial process. The diameters of batches of crank pins were measured on various days; if the industrial process is "under control" the diameters should not depend on the day they were measured. A subset of the data is given in Table 3.6 in a format consistent with performing a regression analysis on the data. The diameters of the crank pins are

Table 3.5 Weights for various heights.

Ht.	Wt.	Ht.	Wt.
65	120	63	110
65	140	63	135
65	130	63	120
65	135	72	170
66	150	72	185
66	135	72	160

actually $.742 + y_{ij}10^{-5}$ inches, where the y_{ij}s are reported in Table 3.6. Perform polynomial regressions on the data. Give two lack-of-fit tests for the simple linear regression not based on polynomial regression.

Table 3.6 Jensen's crank pin data.

Day	Diameter	Day	Diameter	Day	Diameter	Day	Diameter
4	93	10	93	16	82	22	90
4	100	10	88	16	72	22	92
4	88	10	87	16	80	22	82
4	85	10	87	16	72	22	77
4	89	10	87	16	89	22	89

EXERCISE 3.10.5. Beineke and Suddarth (1979) and Devore (1991, p. 380) consider data on roof supports involving trusses that use light-gauge metal connector plates. Their dependent variable is an axial stiffness index (ASI) measured in kips per inch. The predictor variable is the length of the light-gauge metal connector plates. The data are given in Table 3.7.

Table 3.7 Axial stiffness index data.

Plate	ASI	Plate	ASI	Plate	ASI	Plate	ASI	Plate	ASI
4	309.2	6	402.1	8	392.4	10	346.7	12	407.4
4	409.5	6	347.2	8	366.2	10	452.9	12	441.8
4	311.0	6	361.0	8	351.0	10	461.4	12	419.9
4	326.5	6	404.5	8	357.1	10	433.1	12	410.7
4	316.8	6	331.0	8	409.9	10	410.6	12	473.4
4	349.8	6	348.9	8	367.3	10	384.2	12	441.2
4	309.7	6	381.7	8	382.0	10	362.6	12	465.8

Fit linear, quadratic, cubic, and quartic polynomial regression models using powers of x, the plate length, and using powers of $x - \bar{x}.$, the plate length minus the

average plate length. Compare the results of the two procedures. If your computer program will not fit some of the models, report on that in addition to comparing results for the models you could fit.

EXERCISE 3.10.6. Consider fitting quadratic models $y_i = \gamma_0 + \gamma_1 x_i + \gamma_2 x_i^2 + \varepsilon_i$ and $y_i = \beta_0 + \beta_1(x_i - \bar{x}_\cdot) + \beta_2(x_i - \bar{x}_\cdot)^2 + \varepsilon_i$. Show that $\gamma_2 = \beta_2$, $\gamma_1 = \beta_1 - 2\beta_2\bar{x}_\cdot$, and $\gamma_0 = \beta_0 - \beta_1\bar{x}_\cdot + \beta_2\bar{x}_\cdot^2$.

EXERCISE 3.10.7. *Cubic Splines.*
To fit two cubic polynomials on the Hooker partition sets, we can fit the regression function

$$\begin{aligned} m(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \gamma_0 h(x) + \gamma_1 x h(x) + \gamma_2 x^2 h(x) + \gamma_3 x^3 h(x) \\ &= \left(\beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3\right) + h(x)\left(\gamma_0 + \gamma_1 x + \gamma_2 x^2 + \gamma_3 x^3\right), \end{aligned}$$

where the polynomial coefficients below the knot are the β_js and above the knot are the $(\beta_j + \gamma_j)$s. Define the change polynomial as

$$C(x) \equiv \gamma_0 + \gamma_1 x + \gamma_2 x^2 + \gamma_3 x^3.$$

To turn the two polynomials into cubic splines, we require that the two cubic polynomials be equal at the knot but also that their first and second derivatives be equal at the knot. It is not hard to see that this is equivalent to requiring that the change polynomial have

$$0 = C(191) = \frac{dC(x)}{dx}\Big|_{x=191} = \frac{d^2C(x)}{dx^2}\Big|_{x=191},$$

where our one knot for the Hooker data is at $x = 191$. Show that imposing these three conditions leads to the model

$$\begin{aligned} m(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \gamma_3 (x - 191)^3 h(x) \\ &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \gamma_3 [(x - 191)_+]^3. \end{aligned}$$

(It is easy to show that $C(x) = \gamma_3(x-191)^3$ satisfies the three conditions. It is a little harder to show that satisfying the three conditions implies that $C(x) = \gamma_3(x-191)^3$.)

Chapter 4
Alternative Estimates I

Abstract In this chapter we introduce three commonly used alternatives to least squares estimation. The first two of them, principal component regression and classical ridge regression, have been around for a very long time and were originally developed to deal with issues of collinearity. Today they are often used to deal with issues of overfitting. (But then overfitting tends to create collinearity.) The third commonly used alternative is lasso regression, which was specifically developed to deal with overfitting. The last section takes a broader view of alternatives to least squares. Classical ridge regression and the lasso are special cases of penalized estimation (regularization) a subject that is treated in more depth in Chapter 8.

4.1 Principal Component Regression

In Section 1.7 we dealt with the issue of collinearity. Four points were emphasized as the effects of collinearity.

1. The estimate of any parameter, say $\hat{\beta}_2$, depends on *all* the variables that are included in the model.
2. The sum of squares for any variable, say x_2, depends on *all* the other variables that are included in the model. For example, none of $SSR(x_2)$, $SSR(x_2|x_1)$, and $SSR(x_2|x_3,x_4)$ would typically be equal.
3. In a model such as $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \varepsilon_i$, small t statistics for both $H_0 : \beta_1 = 0$ and $H_0 : \beta_2 = 0$ are not sufficient to conclude that an appropriate model is $y_i = \beta_0 + \beta_3 x_{i3} + \varepsilon_i$. To arrive at a reduced model, one must compare the reduced model to the full model.
4. A moderate amount of collinearity has little effect on predictions and therefore little effect on SSE, R^2, and the explanatory power of the model. Collinearity increases the variance of the $\hat{\beta}_j$s, making the estimates of the parameters less reliable. Depending on circumstances, sometimes a large amount of collinearity

can have an effect on predictions. Just by chance one may get a better fit to the data than can be justified scientifically.

At its worst, collinearity involves near redundancies among the predictor variables. An exact redundancy among the predictor variables occurs when we can find a $p \times 1$ vector $d \neq 0$ so that $Xd = 0$. When this happens the rank of X is not p, so we cannot find $(X'X)^{-1}$ and we cannot find the estimates of β in Proposition 2.3.1. *Near* redundancies occur when we can find a vector d that is not too small, say with $d'd = 1$, having $Xd \doteq 0$. Principal components (PC) regression is a method designed to identify near redundancies among the predictor variables. Having identified near redundancies, they can be eliminated if we so choose. In Section 1.7 we mentioned that having small collinearity requires more than having small correlations among all the predictor variables, it requires all partial correlations among the predictor variables to be small as well. For this reason, eliminating near redundancies cannot always be accomplished by simply dropping well-chosen predictor variables from the model.

The basic idea of principal components is to find new variables that are linear combinations of the x_js and that are *best able to (linearly) predict the entire set of* x_j*s*; see *ALM*. Thus the first principal component variable is the one linear combination of the x_js that is best able to predict all of the x_js. The second principal component variable is the linear combination of the x_js that is best able to predict all the x_js among those linear combinations having a sample correlation of 0 with the first principal component variable. The third principal component variable is the best predictor that has sample correlations of 0 with the first two principal component variables. The remaining principal components are defined similarly. With $p-1$ predictor variables, there are $p-1$ principal component variables. The full collection of principal component variables always predicts the full collection of x_js perfectly. The last few principal component variables are least able to predict the original x_j variables, so they are the least useful. They are also the aspects of the predictor variables that are most redundant; see *PA*. The best (linear) predictors used in defining principal components can be based on either the covariances between the x_js or the correlations between the x_js. Unless the x_js are measured on the same scale (with similarly sized measurements), it is generally best to use principal components defined using the correlations.

For *The Coleman Report* data, a matrix of sample correlations between the x_js was given in Example 1.8.1. Principal components are derived from the eigenvalues and eigenvectors of this matrix, cf. Section A.8. An eigenvector corresponding to the largest eigenvalue determines the first principal component variable.

The eigenvalues are given in Table 4.1 along with proportions and cumulative proportions. The proportions in Table 4.1 are simply the eigenvalues divided by the sum of the eigenvalues. The cumulative proportions are the sum of the first group of eigenvalues divided by the sum of all the eigenvalues. In this example, the sum of the eigenvalues is

$$5 = 2.8368 + 1.3951 + 0.4966 + 0.2025 + 0.0689.$$

The sum of the eigenvalues must equal the sum of the diagonal elements of the original matrix. The sum of the diagonal elements of a correlation matrix is the number of variables in the matrix. The third eigenvalue in Table 4.1 is 0.4966. The proportion is $0.4966/5 = 0.099$. The cumulative proportion is $(2.8368 + 1.3951 + 0.4966)/5 = 0.946$. With an eigenvalue proportion of 9.9%, the third principal component variable accounts for 9.9% of the variance associated with predicting the x_js. Taken together, the first three principal components account for 94.6% of the variance associated with predicting the x_js because the third cumulative eigenvalue proportion is 0.946.

Table 4.1 Eigen analysis of the correlation matrix.

Eigenvalue	2.8368	1.3951	0.4966	0.2025	0.0689
Proportion	0.567	0.279	0.099	0.041	0.014
Cumulative	0.567	0.846	0.946	0.986	1.000

For the school data, the principal component (PC) variables are determined by the coefficients in Table 4.2. The first principal component variable is

$$\begin{aligned}\text{PC1}_i = &-0.229(x_{i1}-\bar{x}_{\cdot 1})/s_1 - 0.555(x_{i2}-\bar{x}_{\cdot 2})/s_2 \\ &- 0.545(x_{i3}-\bar{x}_{\cdot 3})/s_3 - 0.170(x_{i4}-\bar{x}_{\cdot 5})/s_4 - 0.559(x_{i5}-\bar{x}_{\cdot 5})/s_5 \quad (1)\end{aligned}$$

for $i = 1,\ldots,20$ where s_1 is the sample standard deviation of the x_{i1}s, etc. The columns of coefficients given in Table 4.2 are actually eigenvectors for the correlation matrix of the x_js. The PC1 coefficients are an eigenvector corresponding to the largest eigenvalue, the PC2 coefficients are an eigenvector corresponding to the second largest eigenvalue, etc.

Table 4.2 Principal component variable coefficients.

Variable	PC1	PC2	PC3	PC4	PC5
x_1	−0.229	−0.651	0.723	0.018	−0.024
x_2	−0.555	0.216	0.051	−0.334	0.729
x_3	−0.545	0.099	−0.106	0.823	−0.060
x_4	−0.170	−0.701	−0.680	−0.110	0.075
x_5	−0.559	0.169	−0.037	−0.445	−0.678

We can now perform a regression on the new principal component variables. The table of coefficients is given in Table 4.3. The analysis of variance is given in Table 4.4. The value of R^2 is 0.906. The analysis of variance table and R^2 are identical to those for the original predictor variables given in Section 1.1. The plot of standardized residuals versus predicted values from the principal component regression is given in Figure 4.1. This is identical to the plot given in Figure 1.6 for the original variables. All of the predicted values and all of the standardized residuals are identical.

Table 4.3 Table of Coefficients: Principal component regression.

Predictor	$\hat{\gamma}$	SE($\hat{\gamma}$)	t	P
Constant	35.0825	0.4638	75.64	0.000
PC1	−2.9419	0.2825	−10.41	0.000
PC2	0.0827	0.4029	0.21	0.840
PC3	−2.0457	0.6753	−3.03	0.009
PC4	4.380	1.057	4.14	0.001
PC5	1.433	1.812	0.79	0.442

Since Table 4.4 and Figure 4.1 are unchanged, any usefulness associated with principal component regression must come from Table 4.3. The principal component variables display no collinearity. Thus, contrary to the warnings given earlier about the effects of collinearity, we can make final conclusions about the importance of variables directly from Table 4.3. We do not have to worry about fitting one

Table 4.4 Analysis of Variance: Principal component regression.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	5	582.69	116.54	27.08	0.000
Error	14	60.24	4.30		
Total	19	642.92			

model after another or about which variables are included in which models. From examining Table 4.3, it is clear that the important variables are PC1, PC3, and PC4. We can construct a reduced model with these three; the estimated regression surface is simply

$$\hat{y} = 35.0825 - 2.9419(\text{PC1}) - 2.0457(\text{PC3}) + 4.380(\text{PC4}), \tag{2}$$

where we merely used the estimated regression coefficients from Table 4.3. Refitting the reduced model is unnecessary because there is no collinearity.

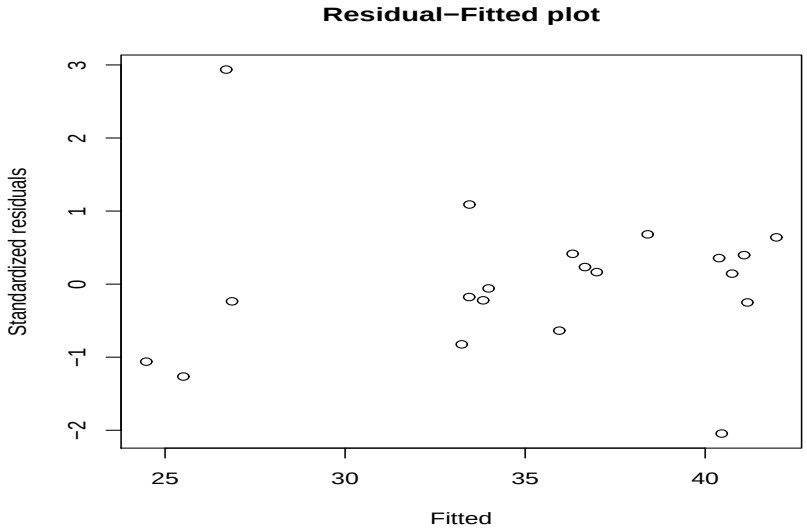

Fig. 4.1 Standardized residuals versus predicted values for principal component regression.

To get predictions for a new set of x_js, just compute the corresponding PC1, PC3, and PC4 variables using formulae similar to those in Equation (1) and make the predictions using the fitted model in Equation (2). When using equations like (1) to obtain new values of the principal component variables, continue to use the $\bar{x}_{\cdot j}$s and s_js computed from only the original observations.

As an alternative to this prediction procedure, we could use the definitions of the principal component variables, e.g., Equation (1), and substitute for PC1, PC3, and PC4 in Equation (2) to obtain estimated coefficients on the original x_j variables.

$$\begin{aligned}
\hat{y} &= 35.0825 + [-2.9419, -2.0457, 4.380] \begin{bmatrix} \text{PC1} \\ \text{PC3} \\ \text{PC4} \end{bmatrix} \\
&= 35.0825 + [-2.9419, -2.0457, 4.380] \times \\
&\quad \begin{bmatrix} -0.229 & -0.555 & -0.545 & -0.170 & -0.559 \\ 0.723 & 0.051 & -0.106 & -0.680 & -0.037 \\ 0.018 & -0.334 & 0.823 & -0.110 & -0.445 \end{bmatrix} \begin{bmatrix} (x_1 - \bar{x}_{\cdot 1})/s_1 \\ (x_2 - \bar{x}_{\cdot 2})/s_2 \\ (x_3 - \bar{x}_{\cdot 3})/s_3 \\ (x_4 - \bar{x}_{\cdot 4})/s_4 \\ (x_5 - \bar{x}_{\cdot 5})/s_5 \end{bmatrix} \\
&= 35.0825 + [-0.72651, 0.06550, 5.42492, 1.40940, -0.22889] \times \\
&\quad \begin{bmatrix} (x_1 - 2.731)/0.454 \\ (x_2 - 40.91)/25.90 \\ (x_3 - 3.14)/9.63 \\ (x_4 - 25.069)/1.314 \\ (x_5 - 6.255)/0.654 \end{bmatrix}.
\end{aligned}$$

Obviously this can be simplified into a form $\hat{y} = \tilde{\beta}_0 + \tilde{\beta}_1 x_1 + \tilde{\beta}_2 x_2 + \tilde{\beta}_3 x_3 + \tilde{\beta}_4 x_4 + \tilde{\beta}_5 x_5$, which in turn simplifies the process of making predictions and provides new estimated regression coefficients for the x_js that correspond to the fitted principal component model. In this case they become $\hat{y} = 12.866 - 1.598x_1 + 0.002588x_2 + 0.5639x_3 + 1.0724x_4 - 0.3484x_5$. These PC regression estimates of the original β_js can be compared to the least squares estimates. Many computer programs for performing PC regression report these estimates of the β_js and their corresponding standard errors. *A similar method is used to obtain lasso estimates when the lasso procedure is performed on standardized predictor variables, cf. Section 3.*

It was mentioned earlier that collinearity tends to increase the variance of regression coefficients. The fact that the later principal component variables are more nearly redundant is reflected in Table 4.3 by the fact that the standard errors for their estimated regression coefficients increase (excluding the intercept).

One rationale for using PC regression is that you just don't believe in using nearly redundant variables. The exact nature of such variables can be changed radically by small errors in the x_js. For this reason, one might choose to ignore PC5 because of its small eigenvalue proportion, regardless of any importance it may display in Table 4.3. If the t statistic for PC5 appeared to be significant, it could be written off as a chance occurrence or, perhaps more to the point, as something that is unlikely to be reproducible. If you don't believe redundant variables, i.e., if you don't believe that they are themselves reproducible, any predictive ability due to such variables will not be reproducible either.

When considering PC5, the case is pretty clear. PC5 accounts for only about 1.5% of the variability involved in predicting the x_js. It is a very poorly defined as-

pect of the predictor variables x_j and, anyway, it is not a significant predictor of y. The case is less clear when considering PC4. This variable has a significant effect for explaining y, but it accounts for only 4% of the variability in predicting the x_js, so PC4 is reasonably redundant within the x_js. If this variable is measuring some reproducible aspect of the original x_j data, it should be included in the regression. If it is not reproducible, it should not be included. From examining the PC4 coefficients in Table 4.2, we see that PC4 is roughly the average of the percent white-collar fathers x_2 and the mothers' education x_5 contrasted with the socio- economic variable x_3. (Actually, this comparison is between the variables after they have been adjusted for their means and standard deviation as in Equation (1).) If PC4 strikes the investigator as a meaningful, reproducible variable, it should be included in the regression.

In our discussion, we have used PC regression both to eliminate questionable aspects of the predictor variables and as a method for selecting a reduced model. We dropped PC5 primarily because it was poorly defined. We dropped PC2 solely because it was not a significant predictor. Some people might argue against this second use of PC regression and choose to take a model based on PC1, PC2, PC3, and possibly PC4.

On occasion, PC regression is based on the sample covariance matrix of the x_js rather than the sample correlation matrix. Again, eigenvalues and eigenvectors are used, but in using relationships like Equation (1), the s_js are deleted. The eigenvalues and eigenvectors for the covariance matrix typically differ from those for the correlation matrix. The relationship between estimated principal component regression coefficients and original least squares regression coefficient estimates is somewhat simpler when using the covariance matrix.

It should be noted that PC regression is just as sensitive to violations of the assumptions as regular multiple regression. Outliers and high-leverage points can be very influential in determining the results of the procedure. Tests and confidence intervals rely on the independence, homoscedasticity, and normality assumptions. Recall that in the full principal components regression model, the residuals and predicted values are identical to those from the regression on the original predictor variables. Moreover, highly influential points in the original predictor variables typically have a large influence on the coefficients in the principal component variables.

Mohammad Hattab and Gabriel Huerta have brought to my attention that principal component regression is also a viable method for fitting models with $n < p$. In that case it is simpler to use the eigenvalues and vectors of XX' rather than $X'X$. These matrices have the same *positive* eigenvalues and if you know the eigenvectors of one, it is easy to find the eigenvectors of the other.

4.2 Classical Ridge Regression

Ridge regression was originally proposed by Hoerl and Kennard (1970) as a method to deal with collinearity. Now it is more commonly viewed as a form of penalized

likelihood estimation, which makes it a form of Bayesian estimation. In this section, we consider the traditional view of ridge regression. Chapter 8 relates ridge regression to the more general issue of penalized estimation.

Hoerl and Kennard (1970) looked at the mean squared error, $\mathrm{E}[(\hat{\beta}-\beta)'(\hat{\beta}-\beta)]$, for estimating β with least squares. This is the expected value of a quadratic form in $(\hat{\beta}-\beta)$. $\mathrm{E}(\hat{\beta}-\beta)=0$ and $\mathrm{Cov}(\hat{\beta}-\beta)=\sigma^2(X'X)^{-1}$; so by Theorem 2.1.1,

$$\mathrm{E}[(\hat{\beta}-\beta)'(\hat{\beta}-\beta)]=\mathrm{tr}[\sigma^2(X'X)^{-1}].$$

If $\lambda_1^2,\ldots,\lambda_p^2$ are the eigenvalues of $(X'X)$, we have $\mathrm{tr}[(X'X)^{-1}]=\sum_{j=1}^p \lambda_j^{-2}$; so

$$\mathrm{E}[(\hat{\beta}-\beta)'(\hat{\beta}-\beta)]=\sigma^2\sum_{j=1}^p \lambda_j^{-2}.$$

If some of the values λ_j^2 are small (which indicates high collinearity), the mean squared error will be large. To alleviate this problem, Hoerl and Kennard suggested using the estimate

$$\tilde{\beta}\equiv(X'X+kI)^{-1}X'Y, \tag{1}$$

where $k\geq 0$ is some fixed scalar. The choice of k will be discussed briefly later but note that $k=0$ gives least squares estimates.

There exists (cf. *PA*) something called a canonical regression model which transforms $Y=X\beta+e$ into

$$Y_*=\begin{bmatrix}L\\0\end{bmatrix}\gamma+e_*,$$

where L is a diagonal matrix. The consequences of using ridge regression are easily studied in the canonical regression model. The least squares estimate is

$$\hat{\gamma}=(L'L)^{-1}[L',0]Y_*.$$

The ridge regression estimate is

$$\tilde{\gamma}=(L'L+kI)^{-1}[L',0]Y_*=(L^2+kI)^{-1}L^2\hat{\gamma}. \tag{2}$$

In particular,

$$\tilde{\gamma}_j=\frac{\lambda_j^2}{\lambda_j^2+k}\hat{\gamma}_j.$$

If λ_j is small, $\tilde{\gamma}_j$ will be shrunk toward zero. If λ_j is large, $\tilde{\gamma}_j$ will change relatively little from $\hat{\gamma}_j$.

The ridge estimate $\tilde{\beta}$ has expected mean square

$$\begin{aligned}\mathrm{E}[(\tilde{\beta}-\beta)'(\tilde{\beta}-\beta)]&=\sigma^2\mathrm{tr}[(X'X+kI)^{-1}X'X(X'X+kI)^{-1}]\\&\quad+\beta'\left\{(X'X+kI)^{-1}X'X-I\right\}'\left\{(X'X+kI)^{-1}X'X-I\right\}\beta.\end{aligned}$$

As in *PA*, this can be simplified to something for which the derivative with respect to k at $k = 0$ can be shown to be negative. Since $k = 0$ constitutes least squares estimation, in terms of mean squared error there exists $k > 0$ that gives better estimates of β than least squares. Unfortunately, the particular values of such k are not known.

Hoerl and Kennard suggested a *ridge trace* to determine k. A ridge trace is a simultaneous plot of the estimated regression coefficients (which are functions of k) against k. The value of k is chosen so that the regression coefficients change little for any larger values of k. In addition to providing a good overall review of ridge regression, Draper and van Nostrand (1979) provide references to criticisms that have been raised against the ridge trace.

Because the mean squared error, $\mathrm{E}[(\tilde{\beta} - \beta)'(\tilde{\beta} - \beta)]$, puts equal weight on each regression coefficient, it is often suggested that ridge regression be used after the predictor variables have been rescaled to have mean zero and a common length (variance).

4.3 Lasso Regression

An alternative to least squares estimation that has become quite popular is *lasso regression*, which was proposed by Tibshirani (1996). "Lasso" stands for *least absolute shrinkage and selection operator*. The interesting thing about lasso is that it automatically performs variable selection, i.e. it excludes "unimportant" variables, while it is estimating the regression parameters.

As discussed in Section 2.3, the least squares estimates $\hat{\beta}_j$ satisfy

$$\sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots - \hat{\beta}_{p-1} x_{i,p-1} \right)^2 = \min_{\beta_0,\ldots,\beta_{p-1}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \cdots - \beta_{p-1} x_{i,p-1} \right)^2 .$$

There are various ways that one can present the lasso criterion for estimation. One of them is to minimize the least squares criterion

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \cdots - \beta_{p-1} x_{i,p-1} \right)^2$$

subject to an upper bound on the sum of the absolute values of the regression coefficients. We define the upper bound in terms of the least squares estimates so that the lasso estimates must satisfy

$$\sum_{j=1}^{p-1} |\beta_j| \le \lambda \sum_{j=1}^{p-1} |\hat{\beta}_j| \tag{1}$$

for some λ with $0 \le \lambda \le 1$. The lasso estimates depend on the choice of λ. The least squares estimates obviously satisfy the inequality when $\lambda = 1$, so $\lambda = 1$ gives least squares estimates. When $\lambda = 0$, all the regression coefficients in the inequality must be zero, but notice that the intercept is not subject to the upper bound in (1). Thus, $\lambda = 0$ gives the least squares estimates for the intercept-only model, i.e., it zeros out all the regression coefficients except the intercept, which it estimates with $\bar{y}_\cdot$.

EXAMPLE 4.3.1. We examine the effect of lasso regression on *The Coleman Report* data. Table 4.5 contains results for five values of λ and least squares estimates for two reduced models. For $\lambda = 1$, the estimates are identical to the least squares estimates for the full model.

Table 4.5 Lasso and least squares estimates: *The Coleman Report* data.

	Lasso λ					Reduced Model	
Predictor	1	0.6	0.56348	0.5	0	Least Squares	
Constant	19.95	18.79306	20.39486	26.51564	35.0825	12.1195	14.58327
x_1	−1.793	−0.33591	0.00000	0.00000	0.0000	−1.7358	0.00000
x_2	0.04360	0.00000	0.00000	0.00000	0.0000	0.00000	0.00000
x_3	0.55576	0.51872	0.51045	0.47768	0.0000	0.5532	0.54156
x_4	1.1102	0.62140	0.52194	0.28189	0.0000	1.0358	0.74989
x_5	−1.811	0.00000	0.00000	0.00000	0.0000	0.00000	0.00000

R's *lasso2* package has a default value of $\lambda = 0.5$, which zeros out the coefficients for x_1, x_2, and x_5. The reduced model that only includes x_3 and x_4 is the model that we liked in Section 1.5. The lasso estimates of β_3 and β_4 are noticeably smaller than the least squares estimates from the reduced model given in the last column of Table 4.5. I also found the largest value of λ that zeros out the coefficients for x_1, x_2, and x_5. That value is $\lambda = 0.56348$. With this larger value of λ, the lasso estimates are closer to the reduced model least squares estimates but still noticeably different.

For $\lambda \ge 0.56349$, lasso produces a nonzero coefficient for x_1. From Section 1.5, if we were going to add another variable to the model containing only x_3 and x_4, the best choice is to add x_1. Table 4.5 includes results for $\lambda = 0.6$ and least squares on the three-variable model. $\lambda = 0.6$ still has the coefficients for x_2 and x_5 zeroed out. Again, the nonzero lasso estimates for β_1, β_3, and β_4 are all closer to zero than the least squares estimates from the model with just x_1, x_3, and x_4. □

Lasso seems to do a good job of identifying the important variables and it does it pretty automatically. That can be both a blessing and a curse. It is far less obvious how well lasso is estimating the regression coefficients. The least squares estimates seem more stable across reduced models than do the lasso estimates. And there is the whole issue of choosing λ.

Notice that the inequality (1) uses the same weight λ on all of the regression coefficients. That is not an obviously reasonable thing to do when the predictor

variables are measured in different units, so lasso is often applied to standardized predictor variables, i.e., variables that have their sample mean subtracted and are then divided by their standard deviation. (This is the default in R's lasso2 package.) The regression estimates can then be transformed back to their original scales to be comparable to the least squares estimates. Section 1 illustrated this standardization procedure for principal components regression. Lasso applied to the unstandardized *Coleman Report* data gives very different, and less appealing, results.

4.4 Robust Estimation and Alternative Distances

Robust estimates of β have less sensitivity to outlying y_i values than least squares estimates, see, for example, Huber and Ronchetti (2009). Robust estimates work better when the distribution of the y_is has fatter tails than the normal distribution, e.g. Laplace, logistic, $t(df)$. Optimal estimates for such distributions tend to be nonlinear. In standard linear models least squares estimates are BLUEs, so they should be reasonable, if not optimal, for most errors that are i.i.d. (independent identically distributed). The robust estimates discussed here are still sensitive to high leverage x_i vectors.

Write the $n \times p$ model matrix as

$$X = \begin{bmatrix} x_1' \\ \vdots \\ x_n' \end{bmatrix}$$

so that the linear model $Y = X\beta + e$ also takes the form

$$y_i = x_i'\beta + \varepsilon_i, \qquad i = 1, \ldots, n.$$

As in Section 2.3 (and reusing notation from Section 2.6), least squares estimates minimize

$$\|Y - X\beta\|^2 \equiv \sum_{i=1}^{n} (y_i - x_i'\beta)^2. \tag{1}$$

Least squares is a geometric estimation criterion, not a statistical one, but *PA* shows that for a standard linear model with $\mathrm{Cov}(Y) = \sigma^2 I$ the least squares estimates are BLUEs and if Y also has a multivariate normal distribution the least squares estimates have other optimal statistical properties. Because the squared distance in (1) is always nonnegative, it is equivalent (but less convenient) to minimize

$$\|Y - X\beta\| = \sqrt{\sum_{i=1}^{n} (y_i - x_i'\beta)^2}.$$

In Section 2.7 we discussed the simplest form of generalized least squares estimation, weighted least squares estimation, in which the weights are defined by a

diagonal matrix $D(w)$ for a vector $w = (w_1, \ldots, w_n)'$ with $w_i > 0$. Weighted least squares estimates minimize

$$\|Y - X\beta\|^2_{D(w)} \equiv (Y - X\beta)'D(w)(Y - X\beta) = \sum_{i=1}^{n} w_i(y_i - x_i'\beta)^2. \tag{2}$$

When $\mathrm{Cov}(Y) = \sigma^2 D(v_i)$, the optimal weights in (2) are $w_i = 1/v_i$.

For an arbitrary n vector v, the measures $\|v\|$ and $\|v\|_{D(W)}$ provide alternative definitions for the length of a vector. A wide variety of estimates for β can be obtained by defining yet other concepts of the length of a vector. One of the most common concepts of length used in mathematics is $\mathbf{L}^p$ length in which, for $p \geq 1$,

$$\|v\|_p \equiv \left[\sum_{i=1}^{n} |v_i|^p\right]^{1/p}.$$

There is also

$$\|v\|_\infty \equiv \max_i\{|v_1|, \ldots, |v_n|\}.$$

Relative to the notation defined in Section 2.6 we have

$$\|\cdot\| \equiv \|\cdot\|_2.$$

A minimum $\mathbf{L}^p$ estimate of β minimizes the distance $\|Y - X\beta\|_p$ or, equivalently, minimizes $(\|Y - X\beta\|_p)^p$. Not that I have ever seen anyone do it, but one could even estimate β by minimizing $\|Y - X\beta\|_\infty$. When $1 \leq p < 2$, minimum $\mathbf{L}^p$ estimates are robust to unusual y_i values. Taking $p > 2$ makes the estimates *more* sensitive to unusual y_i values, something statisticians rarely want. (Hence my never seeing anyone use minimum $\mathbf{L}^\infty$ estimation. However, according to Nievergelt (2000), Laplace used $p = 1, 2, \infty$ around 1800.) Recall that when estimating the mean of a thin tailed distribution, like the uniform, it is the most extreme observations that provide the most information. Minimum $\mathbf{L}^p$ estimation provides an immediate analogy to finding weighted least squares estimates: just minimize $\sum_{i=1}^n w_i|y_i - x_i'\beta|^p$ for positive w_is.

In the search for good robust estimates, people have gone well past the use of minimum weighted $\mathbf{L}^p$ estimation with $1 \leq p < 2$. *M-estimates* involve choosing a nonnegative loss function $\mathscr{L}(y, u)$ and weights and then picking $\tilde{\beta}$ to minimize the weighted sum of the losses, i.e.,

$$\sum_{i=1}^{n} w_i\mathscr{L}(y_i, x_i'\tilde{\beta}) = \min_\beta \sum_{i=1}^{n} w_i\mathscr{L}(y_i, x_i'\beta). \tag{3}$$

Whether this gives robust estimation or not depends on the choice of loss function.

The M in M-estimation is an allusion to maximum likelihood type estimates. The loss function $\mathscr{L}(y, u) = (y - u)^2$, with equal weights, leads to least squares and thus to maximum likelihood estimates for standard linear models with multivariate normal data. In generalized linear models for binomial data, wherein y_i denotes the

proportion of successes, maximum likelihood estimates of β can typically be cast as minimizing a weighted sum of losses where the weights equal the binomial sample sizes, cf. Section 9.1. Even Support Vector Machines can be cast as estimating β in $x'\beta$ by minimizing a sum of losses, cf. Section 9.5.

In linear models, loss functions typically take the form

$$\mathscr{L}(y,u) = \mathscr{L}(y-u).$$

If $\mathscr{L}(\xi)$ is differentiable everywhere, Newton-Raphson can be used to find the minimizing value. One of the more famous families of robust loss functions is *Tukey's biweight* which is typically defined by its derivative:

$$\mathbf{d}_{\xi}\mathscr{L}_c(\xi) \equiv \begin{cases} \xi\left(1-\frac{\xi^2}{c^2}\right) & \text{if } |\xi| < c \\ 0 & \text{if } |\xi| \geq c \end{cases}$$

for some scale factor c.

Another popular loss function leads to *quantile regression*. For $0 < \tau < 1$ define the loss function,

$$\mathscr{L}_{\tau}(y-u) \equiv \begin{cases} \tau(y-u) & \text{if } y-u > 0 \\ (\tau-1)(y-u) & \text{if } y-u \leq 0 \end{cases}.$$

If $\tau = 0.5$ this is median regression and is equivalent to using $\mathbf{L}^1$ loss. In general, the value of u that will minimize this loss function is the τ quantile of the random variable y. This loss function can be minimized using *linear programming*.

Chapter 5
Variable Selection

Abstract This chapter addresses the question of which predictor variables should be included in a linear model. The easiest version of the problem is, given a linear model, which variables should be excluded. To that end we examine the question of selecting the best subset of predictor variables from amongst the original variables. To do this requires us to define a *best* model and we examine several competing measures. We also examine a *greedy* algorithm for this problem known as backward elimination. The more difficult problem of deciding which variables to place into a linear model is addressed by the greedy algorithm of forward selection. (These algorithms are greedy in the sense of always wanting the best thing right now, rather than seeking a global sense of what is best.) We examine traditional forward selection as well as the modern adaptations of forward selection known as boosting, bagging, and random forests. We continue to illustrate techniques on the data from the *Coleman Report* given in Section 1.1 (Table 1.1)

In general suppose we have a set of variables $y, x_1, \ldots, x_s$ and observations on these variables $y_i, x_{i1}, \ldots, x_{is}$, $i = 1, \ldots, n$. We want to identify which of the predictor variables x_j are important for a regression on y. There are several methods available for doing this. Recall from Section 2.8 that models with fewer predictors, sometimes even when the models are incorrect, can provide better estimates than a full model. Reduced models are also of interest because, when a good reduced model provides an adequate explanation of the current data, the reduced model is typically more understandable because it is more succinct.

Tests for the adequacy of various reduced models can be performed, assuming that the full model

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_s x_{is} + e_i \tag{1}$$

is an adequate model for the data. This largest model will be written $Y = X\beta + e$. (In this chapter X is $n \times (s+1)$ rather than $n \times p$.) In this chapter a candidate (reduced) model with $p < s$ predictor variables will be written

$$y_i = \gamma_0 + \gamma_1 x_{i1} + \cdots + \gamma_p x_{ip} + e_i, \tag{2}$$

or as $Y = X_0\gamma + e$. In another notational change for this chapter, to eliminate any possible confusion about the term "mean squared error" we will refer to SSE/dfE as the *residual mean square (RMS)*.

Difficulties with predictions can arise when a good reduced model is used with new cases that are not similar to those on which the reduced model was fitted and evaluated. In particular, a good fitted reduced model should not be used for prediction of a new case unless *all* of the predictor variables in the new case are similar to those in the original data. *It is not enough that new cases be similar on just the variables in the reduced model*. In fact it is not even sufficient that they be similar on all of the variables in the full model because some important variable may not have been measured for the full model, yet a new case with a very different value of this unmeasured variable can act very differently. The new cases really need to be a sample from the same population as the original data.

In the era of big data, we conveniently cause new cases to be sampled from the same population when evaluating models. The gold standard for evaluating predictive models seems to be randomly dividing the data into a set of training data and a set of test data. A predictive model is constructed on the training data and its predictive accuracy is evaluated on the test data. In this scenario, the full data comprise the population, the training data constitutes one random sample from the population, and the test data constitutes another. Any relationships that actually exist in this population should exist in both the training and test data. But in truth, the complete data are unlikely to be our real target population since the real target of our predictions is likely to be behavior conducted in the future. By construction the test data has the same structure as the training data, so predicting the test data is an easier problem than predicting actual future behavior, which is not guaranteed to have the same structure as the current data.

More recently it seems to have become standard to break the data into three parts: *training data*, *developmental* (or *validation*) *data*, and *test data*. Training data is used to estimate the parameters of the various statistical models under consideration. The various models are then used to predict the developmental data as a means to select one of the various models under consideration. This could be used to choose between using polynomials, trig functions, wavelets, or splines. It could also be used to choose between a ridge penalty function or a LASSO penalty function and in choosing the tuning parameter k in penalized regression. Finally, the test data are used only for assessing the accuracy of predictions made by the models determined at the developmental stage. If the data are not big enough to allow construction of a developmental data set in addition to the training and test sets, cross-validation is often used in its place.

If the training error is small relative to the developmental and test errors, you are *overfitting* the model and creating excess variability in your predictions. If you have poor developmental and test errors, you may be *underfitting* the model and creating excess bias. Of course large prediction errors can also occur with the perfect predictive model when the underlying process simply involves a lot of error.

Section 1 discusses best subset selection and introduces six approaches for ranking (identifying the best) candidate models. Section 2 considers three methods for

making sequential selections of variables: backward elimination, forward selection, and stepwise methods. Obviously, it is better to consider all reduced models, whenever feasible, rather than making sequential selections. Sequential methods are flawed but they are cheap and easy. Section 3 illustrates how model selection results can depend on outliers. (If you change anything, you change everything.) Section 4 contains a discussion of the ideas presented to that point. Section 5 presents some modern alternatives to of forward selection.

5.1 Best Subset Selection

The most sure way to find the best reduced model is to look at all of them. If you have a criterion for deciding on the best model, fit all of the possible candidate models and select the best ones. For regression equations involving $x_1, \ldots, x_s$, there are 2^s candidate models. (Last time I checked, both R and Minitab required $s \leq 31$.) Even if one has the time and money to compute all of the models, it may be difficult to assimilate that much information, hence the need to specify a criterion for ranking the best models.

The efficient computation of all possible regressions is due to Schatzoff, Tsao, and Fienberg (1968). Their algorithm was a major advance. Further advances have made this method obsolete. It is a waste of money to compute all possible regressions. One should only compute those regressions that consist of the best subsets of the predictor variables. The efficient computation of the best regressions is due to Furnival and Wilson (1974).

"Best" is defined by ranking models on the basis of some measure of how well they fit or predict. The most commonly used of these measures were R^2, adjusted R^2, and Mallows's C_p. In recent years *AIC* and *BIC* have become increasingly popular measures. Except for R^2, all of these criteria introduce a penalty for fitting more parameters. *Cost complexity pruning* determines the best model by using cross-validation to determine the most appropriate penalty. All of these criteria are discussed in the subsections that follow. Although nominally discussed for regression models, *all of these measures are trivially adapted to general linear models by replacing the number of columns in model matrices by their ranks.*

Although the criteria for identifying best models are traditionally used in the context of finding the best subsets among all possible regression models, they can be used to identify the best within any collection of linear or generalized linear models. For example, Christensen (2015) uses the C_p statistic to identify best unbalanced ANOVA models, Christensen (1997) used AIC to identify best ANOVA-like log-linear models for categorical data, and in the next section we mention using them on the sequences of models created by stepwise regression procedures.

5.1.1 R^2 statistic

The fundamental statistic in comparing all possible reduced models is the R^2 statistic. This is appropriate but we should recall some of the weaknesses of R^2. The numerical size of R^2 is more related to predictive ability than to model adequacy. The perfect model can have small predictive ability and thus a small R^2, while demonstrably inadequate models can still have substantial predictive ability and thus a high R^2. Fortunately, we are typically more interested in prediction than in finding the perfect model, especially since our models are typically empirical approximations for which no perfect model exists. Moreover, although the absolute size of R^2 does not address model fit, the relative sizes of R^2 does. The model with the higher R^2 typically fits better (has a smaller SSE), regardless of whether it is actually a good fit. Finally, when considering transformations of the dependent variable, the R^2 values for different models are not comparable (unless predictions are back transformed to the original scale and correlated with the original data to obtain R^2).

In the present context, the most serious drawback of R^2 is that it typically goes up when more predictor variables are added to a model. (It cannot go down.) Thus it is not really appropriate to compare the R^2 values of two models with different numbers of predictors. However, we can use R^2 to compare models with *the same* number of predictor variables. In fact, for models with the same number of predictors, we can use R^2 to order them from best to worse; the largest R^2 value then corresponds to the best model. R^2 is the fundamental model comparison statistic for best subset methods in that, *for comparing models with the same number of predictors*, the other methods considered give the same relative orderings for models as R^2. The essence of the other methods is to develop a criterion for comparing models that have *different* numbers of predictors, i.e., the methods incorporate penalties for adding more regression parameters.

Table 5.1 contains the two best models for the *Coleman Report* data based on the R^2 statistic for each number of predictor variables. The best single variable is x_3; the second best is x_2. This information could be obtained from the correlations between y and the predictor variables given in Section 1.1. Note the drastic difference between the R^2 for using x_3 and that for x_2. The best pair of variables for predicting y is x_3 and x_4, while the second best pair is x_3 and x_5. The best three-variable model contains x_1, x_3, and x_4. Note that the largest R^2 values go up very little when a forth or fifth variable is added. Moreover, all the models in Table 5.1 that contain three or more variables include x_3 and x_4. We could conduct F tests to compare models with different numbers of predictor variables, as long as the smaller models are contained in the larger ones.

Any models that we think are good candidates should be examined for influential and outlying observations, consistency with assumptions, and subject matter implications. Any model that makes particularly good sense to a subject matter specialist warrants special consideration. Models that make particularly poor sense to subject matter specialists may be dumb luck but they may also be the springboard for new insights into the process generating the data. We will examine the role of observations that are influential or outlying in the original (full) model in more detail later.

Table 5.1 Best subset regression: R^2 statistic.

			Included variables				
Vars.	R^2	$\sqrt{RMS}$	x_1	x_2	x_3	x_4	x_5
1	86.0	2.2392			X		
1	56.8	3.9299		X			
2	88.7	2.0641			X	X	
2	86.2	2.2866			X		X
3	90.1	1.9974	X		X	X	
3	88.9	2.1137			X	X	X
4	90.2	2.0514	X		X	X	X
4	90.1	2.0603	X	X	X	X	
5	90.6	2.0743	X	X	X	X	X

Finally, recall that when making predictions based on reduced models, the point at which we are making the prediction generally needs to be consistent with the original data on all variables, not just the variables in the reduced model. When we drop a variable, we do not conclude that the variable is not important, we conclude that it is not important *for this set of data*. For different data, a dropped variable may become important. We cannot presume to make predictions from a reduced model for new cases that are substantially different from the original data.

5.1.2 Adjusted R^2 statistic

The adjusted R^2 statistic is simply an adjustment of R^2 that allows comparisons to be made between models with different numbers of predictor variables. Let p be the number of predictor variables in a candidate model (excluding the intercept), then the adjusted R^2 is defined to be

$$\text{Adj } R^2 \equiv 1 - \frac{n-1}{n-p-1}\left(1-R^2\right).$$

For the *Coleman Report* example with all predictor variables, this becomes

$$0.873 = 1 - \frac{20-1}{20-6}\left(1-0.9063\right),$$

or, as it is commonly written, 87.3%.

It is not too difficult to see that

$$\text{Adj } R^2 = 1 - \frac{RMS}{s_y^2}$$

where RMS is the residual mean square of the candidate model and s_y^2 is the sample variance of the y_is, i.e., $s_y^2 = SSTot/(n-1)$. This is a much simpler statement

than the defining relationship. For the *Coleman Report* example with all predictor variables, this is

$$0.873 = 1 - \frac{4.30}{(642.92)/19}.$$

Note that *when comparing two models, the model with the smaller RMS has the larger adjusted* R^2.

R^2 is always between 0 and 1, but while the adjusted R^2 cannot get above 1, it can get below 0. It is possible to find models that have $RMS > s_y^2$. In these cases, the adjusted R^2 is actually less than 0.

Models with large adjusted R^2s are precisely models with small residual mean squares. At first glance, this seems like a reasonable way to choose models, but upon closer inspection the idea seems flawed. The problem is that when comparing some model with a reduced model, the adjusted R^2 is greater for the larger model whenever the residual mean square of the larger model is less than the numerator mean square for testing the adequacy of the smaller model. In other words, the adjusted R^2 is greater for the larger model whenever the F statistic for comparing the models is greater than 1. Typically, we want the F statistic to be substantially larger than 1 before concluding that the extra variables in the larger model are important.

To see that the adjusted R^2 is larger for the larger model whenever $F > 1$, consider the simplest example, that of comparing the full model to the model that contains just an intercept. For the *Coleman Report* data, the residual mean square for the intercept model is

$$\begin{aligned} SSTot/19 = 642.92/19 &= (SSReg + SSE)/19 \\ &= (5MSReg + 14RMS)/19 = \frac{5}{19}116.54 + \frac{14}{19}4.30. \end{aligned}$$

Thus $SSTot/19$ is a weighted average of $MSReg$ and RMS. The $MSReg$ is greater than the RMS ($F > 1$), so the weighted average of the terms must be greater than the smaller term, RMS. The weighted average is $SSTot/19$, which is the residual mean square for the intercept model, while RMS is the residual mean square for the full model. Thus $F > 1$ implies that the residual mean square for the smaller model is greater than the residual mean square for the larger model and the model with the smaller residual mean square has the higher adjusted R^2.

In general, the residual mean square for the smaller model is a weighted average of the mean square for the variables being added and the residual mean square of the larger model. If the mean square for the variables being added is greater than the residual mean square of the larger model, i.e., if $F > 1$, the residual mean square for the smaller model must be greater than that for the larger model. If we add variables to a model whenever the F statistic is greater than 1, we will include a lot of unnecessary variables.

Table 5.2 contains the six best-fitting models as judged by the adjusted R^2 criterion. As advertised, the ordering of the models from best to worst is consistent whether one maximizes the adjusted R^2 or minimizes the RMS (or equivalently, $\sqrt{RMS}$). The best model based on the adjusted R^2 is the model with variables x_1,

x_3, and x_4, but a number of the best models are given. Presenting a number of the best models reinforces the idea that selection of one or more final models should be based on many more considerations than just the value of one model selection statistic. Moreover, the *best* model as determined by the adjusted R^2 often contains too many variables.

Table 5.2 Best subset regression: Adjusted R^2 statistic.

	Adj.		Included variables				
Vars.	R^2	$\sqrt{RMS}$	x_1	x_2	x_3	x_4	x_5
3	88.2	1.9974	X		X	X	
4	87.6	2.0514	X		X	X	X
4	87.5	2.0603	X	X	X	X	
2	87.4	2.0641			X	X	
5	87.3	2.0743	X	X	X	X	X
3	86.8	2.1137			X	X	X

Note also that the two models in Table 5.2 with three variables are precisely the two three-variable models with the highest R^2 values from Table 5.1. The same is true about the two four-variable models that made this list. As indicated earlier, when the number of variables is fixed, ordering models by their R^2s is equivalent to ordering models by their adjusted R^2s. The comments about model checking and prediction made in the previous subsection continue to apply.

5.1.3 Mallows's C_p statistic

Mallows's C_p statistic estimates a measure of the difference between the fitted regression surface from a reduced model and the actual regression surface. The idea is to compare the points

$$z_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \ldots + \beta_s x_{i,s}$$

on the actual regression surface of the full model (Full) to the corresponding predictions $\hat{y}_{iR}$ from some candidate model (Red.) that has, say, p predictor variables (excluding the constant). The comparisons are made at the locations of the original data. The model comparison is based on the sum of standardized squared differences,

$$\kappa \equiv \sum_{i=1}^{n} (\hat{y}_{iR} - z_i)^2 \big/ \sigma^2.$$

The term σ^2 serves only to provide some standardization. Small values of κ indicate good reduced models. Note that κ is not directly useful because it is unknown. It depends on the z_i values and they depend on the unknown full model regression

parameters. However, if we think of the $\hat{y}_{iR}$s as functions of the random variables y_i, the comparison value κ is a function of the y_is and thus is a random variable with an expected value. Mallows's C_p statistic is an estimate of the expected value of κ. In particular, Mallows's C_p statistic is

$$C_p = \frac{SSE(Red.)}{RMS(Full)} - (n - 2p - 2).$$

A derivation of this statistic is given later. The smaller the C_p value, the better the model (up to the variability of the estimation). If the C_p statistic is computed for the full model, the result is always $s+1$, the number of predictor variables including the intercept. For general linear models the C_p for a candidate model $Y = X_0\gamma + e$ replaces $n-2p-2$ with $n-2r(X_0)$.

In multiple regression, estimated regression surfaces are identical to prediction surfaces, so models with Mallows's C_p statistics that are substantially less than $s+1$ can be viewed as reduced models that are estimated to be better at prediction than the full model. Of course this comparison between predictions from the full and reduced models is restricted to the actual combinations of predictor variables in the observed data.

For the *Coleman Report* data, Table 5.3 contains the best six models based on the C_p statistic. The best model is the one with variables x_3 and x_4, but the model including x_1, x_3, and x_4 has essentially the same value of C_p. There is a substantial increase in C_p for any of the other four models. Clearly, we would focus attention on the two best models to see if they are adequate in terms of outliers, influential observations, agreement with assumptions, and subject matter implications. As always, predictions can only be made with safety from the reduced models when the new cases are to be obtained in a similar fashion to the original data. In particular, new cases must have similar values to those in the original data for all of the predictor variables, not just those in the reduced model. Note that the ranking of the best models is different here than for the adjusted R^2. The full model is not included here, while it was in the adjusted R^2 table. Conversely, the model with x_2, x_3, and x_4 is included here but was not included in the adjusted R^2 table. Note also that among models with three variables, the C_p rankings agree with the R^2 rankings and the same holds for four-variable models.

Table 5.3 Best subset regression: C_p statistic.

			Included variables				
Vars	C_p	$\sqrt{RMS}$	x_1	x_2	x_3	x_4	x_5
2	2.8	2.0641			X	X	
3	2.8	1.9974	X		X	X	
3	4.6	2.1137			X	X	X
4	4.7	2.0514	X		X	X	X
3	4.8	2.1272		X	X	X	
4	4.8	2.0603	X	X	X	X	

It has been my impression that Mallows's C_p statistic is the most popular method for selecting a best subset of the predictor variables. It is certainly my favorite. Mallows's C_p statistic is closely related to Akaike's information criterion (AIC), which is a general criterion for model selection and the model selection criterion that R seems to default to most often.

It is an exercise in PA-V *to show that when you can test two candidate models, the C_p is smaller for the larger model if and only if an appropriate F test statistic is larger than 2.*

5.1.3.1 Derivation

Assume a correct full model $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_s x_{is} + e_i$, i.e.,

$$Y = X\beta + e, \quad \mathrm{E}(e) = 0, \quad \mathrm{Cov}(e) = \sigma^2 I.$$

In variable selection the problem nominally being addressed is that some of the β_js may be zero. But it follows from Section 2.8 that when some of the β_js are merely small, we may get better fitted values by eliminating those small β_js altogether. Rather than trying to identify which β_js are zero, Mallows suggested that the appropriate criterion for evaluating a reduced candidate model $Y = X_0\gamma + e$ is via its *mean squared error for estimating $X\beta$*, i.e.,

$$\mathrm{E}\left[(X_0\hat{\gamma} - X\beta)'(X_0\hat{\gamma} - X\beta)\right].$$

This equals $\sigma^2\mathrm{E}(\kappa)$ and is the same criterion used in Section 2.8. To distinguish between this use of the term "mean squared error" and the estimate of the variance in the full model we write $RSS(\beta) \equiv Y'(I-M)Y$ for the residual sum of squares and $RMS(\beta) \equiv Y'(I-M)Y/r(I-M)$ for the residual mean square. The statistics $RSS(\gamma)$ and $RMS(\gamma)$ are the corresponding quantities for the model $Y = X_0\gamma + e$.

The quantity

$$(X_0\hat{\gamma} - X\beta)'(X_0\hat{\gamma} - X\beta)$$

is a quadratic form in the vector $(X_0\hat{\gamma} - X\beta)$. Writing the perpendicular projection operator onto $C(X_0)$ as

$$M_0 = X_0(X_0'X_0)^{-}X_0'$$

so that $X_0\hat{\gamma} = M_0Y$, recalling that M_0 must be idempotent and symmetric, and applying the first two parts of Proposition 2.1.1 gives

$$(X_0\hat{\gamma} - X\beta) = M_0Y - X\beta,$$

$$\mathrm{E}(X_0\hat{\gamma} - X\beta) = M_0X\beta - X\beta = -(I-M_0)X\beta,$$

$$\mathrm{Cov}(X_0\hat{\gamma} - X\beta) = \sigma^2 M_0.$$

From Proposition 2.1.1 part 3,

$$\mathrm{E}\left[(X_0\hat{\gamma}-X\beta)'(X_0\hat{\gamma}-X\beta)\right] = \sigma^2\mathrm{tr}(M_0)+\beta'X'(I-M_0)X\beta. \tag{1}$$

We do not know σ^2 or β but we can estimate the right-hand side of equation (1). We know that $\mathrm{E}[RMS(\beta)] = \sigma^2$ and note that

$$\mathrm{E}[RSS(\gamma)] = \mathrm{E}\left[Y'(I-M_0)Y\right] = \sigma^2\mathrm{tr}(I-M_0)+\beta'X'(I-M_0)X\beta,$$

so

$$\begin{aligned}\mathrm{E}\left[(X_0\hat{\gamma}-X\beta)'(X_0\hat{\gamma}-X\beta)\right] &= \sigma^2\mathrm{tr}(M_0)+\mathrm{E}\left[Y'(I-M_0)Y\right]-\sigma^2\mathrm{tr}(I-M_0)\\ &= \sigma^2\left[2\mathrm{tr}(M_0)-n\right]+\mathrm{E}\left[Y'(I-M_0)Y\right].\end{aligned}$$

With $p+1 = \mathrm{tr}(M_0) = r(X_0)$, an unbiased estimate of the mean squared error is

$$RMS(\beta)[2(p+1)-n]+RSS(\gamma).$$

Mallows's C_p statistic simply rescales the estimated mean squared error,

$$C_p \equiv \frac{RSS(\gamma)}{RMS(\beta)} - [n-2(p+1)].$$

The models with the smallest values of C_p have the smallest estimated mean squared error and should be among the best models for the data.

5.1.4 A combined subset selection table

For the Coleman Report data, Table 5.4 lists the three best models based on R^2 for each number of predictor variables. In addition, the adjusted R^2 and C_p values for each model are listed in the table. It is easy to identify the best models based on any of the model selection criteria. The output is extensive enough to include a few notably bad models. Rather than asking for the best 3, one might ask for the best 4, or 5, or 6 models for each number of predictor variables but it is difficult to imagine a need for any more extensive summary of the models when beginning a search for good reduced models.

Note that the model with x_1, x_3, and x_4 is the best model as judged by adjusted R^2 and is nearly the best model as judged by the C_p statistic. (The model with x_3 and x_4 has a slightly smaller C_p value.) The model with x_2, x_3, x_4 has essentially the same C_p statistic as the model with x_1, x_2, x_3, x_4 but the latter model has a larger adjusted R^2.

Table 5.4 Best subset regression.

		Adj.			Included variables				
Vars.	R^2	R^2	C_p	$\sqrt{RMS}$	x_1	x_2	x_3	x_4	x_5
1	86.0	85.2	5.0	2.2392			X		
1	56.8	54.4	48.6	3.9299		X			
1	53.7	51.2	53.1	4.0654					X
2	88.7	87.4	2.8	2.0641			X	X	
2	86.2	84.5	6.7	2.2866			X		X
2	86.0	84.4	6.9	2.2993		X	X		
3	90.1	88.2	2.8	1.9974	X		X	X	
3	88.9	86.8	4.6	2.1137			X	X	X
3	88.7	86.6	4.8	2.1272		X	X	X	
4	90.2	87.6	4.7	2.0514	X		X	X	X
4	90.1	87.5	4.8	2.0603	X	X	X	X	
4	89.2	86.3	6.1	2.1499		X	X	X	X
5	90.6	87.3	6.0	2.0743	X	X	X	X	X

5.1.5 Information Criteria: AIC, BIC

Unlike the previous criteria that depend only on second moment assumptions (means, variances, and covariances), information criteria depend on the likelihood of the data. The log-likelihood for a standard linear model under normal theory is

$$\ell(\beta,\sigma^2) = \frac{-n}{2}\log(2\pi) - \frac{n}{2}\log[\sigma^2] - (Y - X\beta)'(Y - X\beta)/2\sigma^2,$$

e.g. *PA*, Chapter 2 or the nonlinear regression chapter of *ANREG*. The log-likelihood is just the log of the probability density function for a random Y given the fixed parameters β and σ^2. What makes the density function into a likelihood function is thinking of Y as a fixed observation and viewing β and σ^2 as unknown variables. The *maximum likelihood estimates (MLEs)* of β and σ^2 are the values that maximize the likelihood. Maximizing the log-likelihood is equivalent and easier. It is not too hard to see that the MLEs are the least squares $\hat{\beta}$ and $\hat{\sigma}^2 \equiv RSS/n$. After simplification, the maximum value of -2 times the log-likelihood function is

$$-2\ell(\hat{\beta},\hat{\sigma}^2) = n\log(2\pi) + n\log(RSS) - n\log(n) + n.$$

The *Akaike Information Criterion (AIC)* is -2 times the maximum of the log-likelihood plus 2 times the number of parameters in the model. Better models have smaller AIC values. For a candidate regression model with p predictors plus an intercept plus an unknown variance, the MLEs are $\hat{\gamma}$, $\hat{\sigma}^2_\gamma = RSS(\gamma)/n$, so

$$\begin{aligned} AIC &= -2\ell(\hat{\gamma},\hat{\sigma}^2_\gamma) + 2(p+2) \\ &= n\log(2\pi) + n\log[RSS(\gamma)] - n\log(n) + n + 2(p+2). \\ &= \{n\log(2\pi) - n\log(n) + n + 4\} + n\log[RSS(\gamma)] + 2p. \end{aligned}$$

Everything in the first term of the last line is a constant that does not depend on the particular model, so, for comparing candidate models with intercepts, effectively

$$AIC = n\log[RSS(\gamma)] + 2p.$$

If you want to use AIC to compare models with different data distributions, the constant term needs to be included.

It is an exercise in *PA-V* to find the AIC for a linear model in which the variance is known. It turns out to be a monotone function of

$$\frac{RSS(\gamma)}{\sigma^2} - [n - 2(p+1)],$$

which is C_p except that $RMS(\beta)$ has been replaced with the known value σ^2.

While it is somewhat advantageous that AIC can be computed without worrying about the existence of a largest model, it is of interest to compare how AIC and C_p work when both are applicable. Using the notation introduced for deriving C_p statistics, AIC picks candidate models with small values of

$$RSS(\gamma)e^{2p/n} = \exp(AIC/n),$$

whereas C_p picks models with small values of

$$RSS(\gamma) + 2pRMS(\beta).$$

In particular, a full model is preferred to a reduced model if $AIC(\beta) < AIC(\gamma)$ but *PA-V* shows that for large samples n, this is approximately the same condition as

$$F > 2,$$

where F is the standard model testing statistic. This is similar to the condition mentioned earlier for C_p but there are some differences related to how the denominators of the F statistics are defined.

For small to moderate samples, many prefer to use a bias corrected form of AIC for evaluating candidate models, namely,

$$AICc \equiv AIC + \frac{2(p+2)(p+3)}{n-p-3}.$$

cf. Sugiura (1978), Hurvich and Tsai (1989), Bedrick and Tsai (1994), and Cavanaugh (1997). It is commonly suggested to use AICc when any of the candidate models have $n/(p+2) < 40$.

Schwarz (1978) presented an asymptotic *Bayesian information criterion (BIC)* which is -2 times the maximum of the log-likelihood plus $\log(n)$ times the number of model parameters. For a standard normal theory regression candidate model with an intercept,

$$BIC = -2\ell(\hat{\gamma}, \hat{\sigma}^2_\gamma) + (p+2)\log(n)$$

$$= \{n\log(2\pi) - (n-2)\log(n) + n\} + n\log RSS + p\log(n),$$

or effectively,

$$BIC = n\log RSS + p\log(n).$$

The derivation of BIC from Bayesian principals is described in Christensen et al. (2010).

BIC places much greater demands on variables to be included. When comparing nested models with p and s predictors, it chooses the larger model when, approximately,

$$\log(n) < F.$$

This approximation requires much larger sample sizes to be effective than the AIC approximation requires, cf. *PA-V*.

It is also an exercise in PA-V to show that, for a given value of p, the R^2, Adj R^2, C_p, AIC, AICc, and BIC criteria all induce the same rankings of candidate models.

5.1.6 Cost complexity pruning

Cost complexity pruning is a related, but more complicated, way of determining the best model within a collection of candidate models. The collection of models can be all possible models or just the sequences of models determined by a stepwise regression method as discussed in the next section. For a given value of α, pick the model from the collection that minimizes $RSS + \alpha p$. The complexity comes because the value of α is chosen by cross-validation. Find the best model for a large number of α values and then choose a final α, and thus a final model, by cross-validation.

Specifically, randomly divide the data into K equal sized subgroups of observations, leave out one subgroup and apply the procedure to the other $K-1$. Using the $K-1$ subgroups as data, find the model within the collection that minimizes $RSS + \alpha p$, fit the model, predict the results in the omitted subgroup, and find the mean of the squared prediction errors, i.e., $MSPE$. (Mean Squared *Prediction* Error; *not* the Mean Squared Pure Error associated with Fisher's lack of fit test.) In cost complexity pruning you do this for a large number of different α values to get a $MSPE$ for each α, i.e., $MSPE(\alpha)$. Cycle through, leaving out a different subgroup each time, to get K different means of squared prediction errors, i.e., $MSPE_k(\alpha)$, $k = 1, \ldots, K$. Pick $\hat{\alpha}$ to minimize $\sum_{k=1}^{K} MSPE_k(\alpha)$. The best model is the model that minimizes $RSS + \hat{\alpha} p$ when fitted to all the data.

James et al. (2013) discuss cost complexity pruning in the context of fitting the regression trees introduced in Chapter 7. The term "pruning" originates from lopping off tree limbs, not from devoring desiccated plums.

5.2 Stepwise Variable Selection

Best subset selection methods evaluate all the possible subsets of variables from a full model and identify the best reduced regression models based on some criterion. Evaluating all possible models is the most reasonable way to proceed in variable selection but the computational demands of evaluating every model can be staggering. Every additional variable in a model doubles the number of reduced models that can be constructed. In our example with five variables, there are $2^5 = 32$ reduced models to be considered; in an example with 8 variables there are $2^8 = 256$ reduced models to be fitted. Years ago, when computation was slow and expensive, fitting large numbers of models was not practical, and even now, when one has a very large number of predictor variables, fitting all models can easily overwhelm a computer. (Actually, improved computer algorithms allow us to avoid fitting all models, but even with the improved algorithms, computational limits can be exceeded.) As alluded to earlier, R and Minitab currently seem willing to consider up to $2^{31} \doteq 200,000$ reduced models.

An alternative to fitting all models is to evaluate the variables one at a time and look at a sequence of models. Stepwise variable selection methods do this. The best of these methods begin with a full model and sequentially identify variables that can be eliminated. In some procedures, variables that have been eliminated may be put back into the model if they meet certain criteria. The virtue of starting with the full model is that if you start with an adequate model and only do reasonable things, you should end up with an adequate model. A less satisfactory procedure is to begin with no variables and see which ones can be added into the model. This begins with an inadequate model and there is no guarantee that an adequate model will ever be achieved. We consider three methods: backwards elimination in which variables are deleted from the full model, forward selection in which variables are added to a model (typically the model that includes only the intercept), and stepwise methods in which variables can be both added and deleted. Because these methods only consider the deletion or addition of one variable at a time, they may never find the best models as determined by best subset selection methods.

5.2.1 Forward selection

Forward selection begins with an initial model and adds variables to the model one at a time. Most often, the initial model contains only the intercept, but many computer programs have options for including other variables in the initial model. Another reasonable starting point is to include all variables with large t statistics when fitting a full model containing all predictors. Logically, variables that are important in the full model should never lose their importance in reduced models.

To determine which variable to add at any step in the process, a candidate variable is added to the current model and the t statistic is computed for the candidate variable. This is done for each candidate variable and the candidate variable with

the largest $|t|$ statistic is added to the model. The procedure stops when none of the absolute t statistics is greater than a predetermined level. The predetermined level can be a fixed number for all steps or it can change with the step. When allowing it to change depending on the step, we could set the process so that it stops when none of the P values for the candidate variables is below a fixed level.

EXAMPLE 5.2.1. *Coleman Report.*
Table 5.5 gives an abbreviated summary of the procedure for the *Coleman Report* data using 2 as the predetermined $|t|$ level for stopping the process and starting with the intercept-only model. At the first step, the five models $y_i = \gamma_{0j} + \gamma_j x_{ij} + \varepsilon_i$, $j = 1, \ldots, 5$ are fitted to the data. The variable x_j with the largest absolute t statistic for testing $\gamma_j = 0$ is added to the model. Table 5.5 indicates that this was variable x_3. At step 2, the four models $y_i = \beta_{0j} + \beta_{3j} x_{i3} + \beta_j x_{ij} + \varepsilon_i$, $j = 1,2,4,5$ are fitted to the data and the variable x_j with the largest absolute t statistic for testing $\beta_j = 0$ is added to the model. In the example, the largest absolute t statistic belongs to x_4. At this point, the table stops, indicating that when the three models $y_i = \eta_{0j} + \eta_{3j} x_{i3} + \eta_{4j} x_{i4} + \eta_j x_{ij} + \varepsilon_i$, $j = 1,2,5$ were fitted to the model, none of the absolute t statistics for testing $\eta_j = 0$ were greater than 2.

Table 5.5 Forward selection on *Coleman Report* data.

Step		Const.	x_1	x_2	x_3	x_4	x_5	R^2	$\sqrt{RMS}$
1	$\hat{\beta}$	33.32			0.560			85.96	2.24
	t_{obs}				10.50				
2	$\hat{\beta}$	14.58			0.542	0.75		88.73	2.06
	t_{obs}				10.82	2.05			

The final model selected is the model with predictor variables x_3 and x_4. This is the same model that will be obtained from backwards elimination and the model that has the smallest C_p statistic. This is a fortuitous circumstance. There is no assurance that such agreement between methods will occur. □

Traditional forward selection sequentially adds variables to the model. Since this is a sequential procedure, the model in question is constantly changing. At any stage in the selection process, forward selection adds the variable that when added:

1. gives the largest absolute t statistic,
2. gives the largest F statistic,
3. gives the smallest P value,
4. increases R^2 the most,
5. decreases RSS the most,
6. gives the smallest C_p,
7. gives the smallest AIC,
8. gives the smallest BIC,

9. has the highest absolute partial correlation with y given the variables in the current model.

It is an exercise in *PA-V* to show that these criteria are equivalent when only considering the addition of one new variable.

Traditional forward selection stops adding variables when one of three things happens:

1. p^* variables have been added,
2. all absolute t statistics for adding variables not in the model are less than t^*,
3. the tolerance is too small for all variables not in the model.

The user (or programmer) picks the values of p^* and t^* and the tolerance limit. The tolerance condition is a limit on the collinearity allowed among of the predictors variables. Tolerance is discussed in the *PA* chapter on collinearity. No variable is ever added if its tolerance is too small, regardless of its absolute t statistic. Traditionally, one just uses the model that stopped the process. Alternatively, one could use a model selection criterion to pick the best among the sequence of models that forward selection has produced.

Although the nine criteria for adding variables are equivalent, stopping rules get tricky. For example, a stopping rule based on having all P values above some fixed number is not equivalent to any stopping rule based on having all $|t|$ statistics below some fixed number because the P values depend on the residual degrees of freedom which keep changing. Reasonable stopping rules based on AIC or BIC might be when these no longer decrease or one can stop when Adj. R^2 fails to increase. Unfortunately, such stopping rules remove the flexibility that a self-selected stopping rule has to control the extent to which forward selection explores the set of all models. A stopping rule for forward selection based on C_p is rarely appropriate because there is rarely a largest model under consideration. (I would never use forward selection if I could fit a reasonable full model.)

The author has a hard time imagining any situation where forward selection from the intercept-only model is a reasonable thing to do, except possibly as a screening device when there are more predictor variables than there are observations. In such a case, the full model cannot be fitted meaningfully, so best subset methods and backwards elimination do not work.

EXAMPLE 5.2.2. *Big Data.* The data involve 1000 observations on a dependent variable and 100 predictor variables. By big data standards, this is a small set of big data. I performed forward selection with a stopping rule that the P value to enter the model must be below $\alpha = 0.05$. The table of parameters follows.

Table of Coefficients: Forward Selection, $\alpha = 0.05$.

Predictor	$\hat{\gamma}_i$	SE($\hat{\gamma}_i$)	t	P
Constant	501.14	9.05	55.39	0.000
x_{24}	−21.84	9.03	−2.42	0.016
x_{56}	21.42	9.05	2.37	0.018
x_{72}	−26.34	8.91	−2.96	0.003
x_{75}	20.51	8.73	2.35	0.019
x_{82}	−18.08	8.79	−2.06	0.040

The model has a horrible R^2 of 0.0287 but it seems that there are several variables that help explain the data.

None of these variables has any actual effect! With two exceptions, all the 101,000 observations are i.i.d. standard normal. The first exception is that the y observations have a signal added to them. The other exception is that x_{49} consists of a small multiple of the signal buried within (the negative of) the white noise in x_{100}. The signal in x_{49} is almost hopelessly lost within that white noise, except for the fact that $x_{49} + x_{100}$ is precisely the small multiple of the signal. If you have both x_{49} and x_{100} in the model, you can extract the signal, but neither x_{49} nor x_{100} by itself helps. Any model that contains the two important variables will have $R^2 = 1.0000$.

I also ran the forward selection stopping when, to get a variable added, its P value must be below $\alpha = 0.25$ (the Minitab default). The idea of the higher cutoff value is to try to get important combinations of variables, like x_{49} and x_{100}, into the model so that their joint effect can be seen. It did not work here. The final model included the 5 predictors from $\alpha = 0.05$ plus 17 additional worthless predictors. Again, the model gives poor prediction with $R^2 = 0.0717$. When using a large α value, it seems inappropriate to blindly use the final model produced, since it will include a lot of variables with little predictive power. A far better procedure would be to select the best among the sequence of models produced or the best subset of the final model. While these are better procedures, in this example they are better procedures for producing garbage.

Even with $\alpha = 0.75$ and a final model that includes 79 of the 100 predictors, forward selection still did not manage to pick up both of the predictors necessary for getting a good model. Using a large α increases our chances of picking up the two good predictors, but it picks up a lot of junk predictors also. If we are lucky enough to pick up our two worthwhile predictors, clearly we would want to go back and eliminate the obvious junk. Of course $\alpha = 1$ will always find the two important variables because it will always fit the full model at the end of the sequence. But if you can fit the full model, you should be doing backward elimination or, better yet, best subset selection.

The way this example was constructed, α is pretty much the probability of finding a good prediction model using forward selection. I did not go looking for a large P value, but the P value associated with x_{100} was particularly large. (Recall that, by itself, x_{100} is unrelated to y and independent of all the predictors except x_{49}, so it's P remains pretty stable in any model that excludes x_{49}. Moreover, because the signal is *buried* in x_{49}, the same is true about x_{49} in any model that excludes x_{100}.) If

the P value for x_{100} is below α, forward selection should find a good model in this example. □

The problem of finding "important" effects that are actually meaningless is ubiquitous with big data. It is no accident that with $\alpha = 0.05$ we found about 5% of the meaningless predictors to be significant. It is no accident that with $\alpha = 0.25$ we found about 25% of the meaningless predictors in our model. The more tests you perform, the more meaningless things will look statistically significant. With any big data where nothing is related, you can always find something that looks related. In exploring big data, the usual standards of statistical significance do not apply, cf. Chapter 6.

As mentioned earlier, stopping rules based on AIC or BIC failing to decrease or Adj. R^2 failing to increase, lack the flexibility in exploring the space of possible models that one gets by selecting a P value or an absolute t statistic for inclusion.

5.2.2 Backwards elimination

Backwards elimination begins with the full model and sequentially eliminates from the model the least important variable. The importance of a variable is judged by the size of the t (or equivalent F) statistic for dropping the variable from the model, i.e., the t statistic for testing whether the corresponding regression coefficient is 0. After the variable with the smallest absolute t statistic is dropped, the model is refitted and the t statistics recalculated. Again, the variable with the smallest absolute t statistic is dropped. The process ends when all of the absolute values of the t statistics are greater than some predetermined level. The predetermined level can be a fixed number for all steps or it can change depending on the step. When allowing it to change depending on the step, we could set up the process so that it stops when all of the P values are below a fixed level.

EXAMPLE 5.2.3. *Coleman Report.*
Table 5.6 illustrates backwards elimination for the *Coleman Report* data. In this example, the predetermined level for stopping the procedure is 2. If all $|t|$ statistics are greater than 2, elimination of variables halts. Step 1 includes all 5 predictor variables. The table gives estimated regression coefficients, t statistics, the R^2 value, and the square root of the *RMS*. In step 1, the smallest absolute t statistic is 0.82, so variable x_2 is eliminated from the model. The statistics in step 2 are similar to those in step 1 but now the model includes only variables x_1, x_3, x_4, and x_5. In step 2, the smallest absolute t statistic is $|-0.41|$, so variable x_5 is eliminated from the model. Step 3 is based on the model with x_1, x_3, and x_4. The smallest absolute t statistic is the $|-1.47|$ for variable x_1, so x_1 is dropped. Step 4 uses the model with only x_3 and x_4. At this step, the t statistics are both greater than 2, so the process halts. Note that the intercept is not considered for elimination.

The final model given in Table 5.6 happens to be the best model as determined by the C_p statistic and the model at stage 3 is the second-best model as determined by

Table 5.6 Backwards elimination on *Coleman Report* data.

Step		Const.	x_1	x_2	x_3	x_4	x_5	R^2	$\sqrt{RMS}$
1	$\hat{\beta}$	19.95	−1.8	0.044	0.556	1.11	−1.8	90.63	2.07
	t_{obs}		−1.45	0.82	5.98	2.56	−0.89		
2	$\hat{\beta}$	15.47	−1.7		0.582	1.03	−0.5	90.18	2.05
	t_{obs}		−1.41		6.75	2.46	−0.41		
3	$\hat{\beta}$	12.12	−1.7		0.553	1.04		90.07	2.00
	t_{obs}		−1.47		11.27	2.56			
4	$\hat{\beta}$	14.58			0.542	0.75		88.73	2.06
					10.82	2.05			

the C_p statistic. This is a fortuitous event; there is no reason that this should happen other than these data being particularly clear about the most important variables. □

Backward elimination sequentially deletes variables from the model. At any stage in the selection process, it deletes the variable with the smallest absolute t statistic or F statistic or equivalent criterion. Backward elimination often stops deleting variables when:

1. p_* variables have been eliminated,
2. the smallest absolute t statistic for eliminating a variable is greater than t_*.

The user can usually specify p_* and t_* in a computer program. Often, the process is stopped when the P value associated with $|t|$ is too small. Stopping rules are also based on AIC or BIC failing to decrease or Adj. R^2 failing to increase. Traditionally, one just uses the model that stopped the process. Alternatively, one could use a model selection criterion to pick the best among the sequence of models that backward elimination produced.

The initial model in the backward elimination procedure is the model with all of the predictor variables included,

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_s x_{is} + e_i.$$

Backward elimination should give an adequate model. We assume that the process is started with an adequate model, and only variables that add nothing are eliminated. The model arrived at may, however, be far from the most succinct. On the other hand, *there is no reason to believe that forward selection gives even an adequate model*.

Since we are starting with an adequate model, unlike forward selection, there is little reason to choose a stopping rule that helps us to explore the space of possible models.

EXAMPLE 5.2.4. *Big Data.*
I applied backward selection to the data with a P value cutoff of (the Minitab default) $\alpha = 0.10$

Table of Coefficients: Backward Elimination, $\alpha = 0.10$.

Predictor	$\hat{\gamma}_i$	$\text{SE}(\hat{\gamma}_i)$	t	P
Constant	0.0458	0.0604	0.76	0.449
x_2	0.0669	0.0306	2.19	0.029
x_{34}	−0.0577	0.0302	−1.91	0.057
x_{48}	0.0632	0.0308	2.05	0.040
x_{49}	999879	104	9572.43	0.000
x_{59}	−0.0572	0.0311	−1.84	0.066
x_{66}	0.0680	0.0299	2.27	0.023
x_{95}	−0.0546	0.0293	−1.86	0.063
x_{98}	−0.0752	0.0303	−2.49	0.013
x_{100}	999879	104	9572.41	0.000

This is an extreme example so the $|t|$ statistics for the two important variables leap out. (Their P values do not!) But the point of this example is not that backward elimination found and kept the important variables. The point of this example is that backward elimination still finds 4 worthless variables that look significant by traditional standards and another 3 worthless variables that one would normally consider to be of marginal significance.

Unlike forward selection, backward elimination gives a good predictive model, one with $R^2 = 1.00$. The data were simulated with a variance of 1 but the residual mean square is $RMS = 0.906$, which is disturbingly below the correct value despite having many degrees of freedom in the estimate ($dfE = 990$). (Based on the asymptotic normal approximation to the χ^2, RMS is more than two standard deviations below the true value.) Fitting the full model there are 899 degrees of freedom for error. From the 98 worthless predictor variables there are another 98 sums of squares with 1 degree of freedom that could all go into the error. The seven largest of those 98 sums of squares have been assigned to the model and the 91 smallest have been assigned to the error. Even when starting with 899 degrees of freedom for error from the full model that truly estimate the error, adding in the 91 smallest and leaving out the 7 biggest sums for squares biases the estimated error downward in the fitted backward elimination model. □

If the final model from backward elimination is sufficiently small, one could apply best subset selection to the variables in the final model. But that is unlikely to accomplish anything that changing the α level could not accomplish and it is highly unlikely to eliminate all of the worthless predictor variables. The bigger the data set, the more stringent our requirements for significance should be.

5.2.3 Other Methods

Stepwise methods alternate between forward selection and backwards elimination. Suppose you have just arrived at a model by dropping a variable. A stepwise method will then check to see if any variable can be added to the model. If you have just arrived at a model by adding a variable, a stepwise method then checks to see if any variable can be dropped. The value of the absolute t statistic required for dropping a variable is allowed to be different from the value required for adding a variable. Stepwise methods often start with an initial model that contains only an intercept, but many computer programs allow starting the process with the full model. In the *Coleman Report* example, the stepwise method beginning with the intercept model gives the same results as forward selection and the stepwise method beginning with the full model gives the same results as backwards elimination. (The absolute t statistics for both entering and removing were set at 2.) Other initial models can also be used.

Four alternative rules for adding, deleting, and exchanging variables follow.

1. Add the variable with the largest absolute t value if that value is greater than t^*.
2. Delete the variable with the smallest absolute t value if that value is less than t_*.
3. A variable not in the model is exchanged for a variable in the model if the exchange increases R^2.
4. The largest R^2 for each size model considered so far is saved. Delete a variable if the deletion gives a model with R^2 larger than any other model of the same size.

These rules can be used in combination. For example, 1 then 2, 1 then 2 then 3, 1 then 4, or 1 then 4 then 3. Again, no variable is ever added if its tolerance is too small.

Basically, these rules are just an attempt to get forward selection to look at a broader collection of models and it would be wise to select the best among the sequence of models generated.

EXAMPLE 5.2.5. *Big Data.*
Minitab's default stepwise procedure began with the intercept model, concluded with 17 variables, none of which were the important two. □

5.3 Variable Selection and Case Deletion

In this section we examine how the results of the previous two sections change when influential cases are deleted. Before beginning, we make a crucial point. *Both variable selection and the elimination of outliers cause the resulting model to appear better than it probably should. Both tend to give RMSs that are unrealistically small. It follows that confidence and prediction intervals are unrealistically narrow and test statistics are unrealistically large*. Outliers tend to be cases with large residuals; any policy of eliminating the largest residuals obviously makes the *SSE*, which is

the sum of the squared residuals, and the *RMS* smaller. Some large residuals occur by chance even when the model is correct. Systematically eliminating these large residuals makes the estimate of the variance too small. Variable selection methods tend to identify as good reduced models those with small *RMS*s. The most extreme case is that of using the adjusted R^2 criterion, which identifies as the best model the one with the smallest *RMS*. Confidence and prediction intervals based on models that are arrived at after variable selection or outlier deletion should be viewed as the smallest reasonable intervals available, with the understanding that more appropriate intervals would probably be wider. Tests performed after variable selection or outlier deletion should be viewed as giving the greatest reasonable evidence against the null hypothesis, with the understanding that more appropriate tests would probably display a lower level of significance.

Recall that in Section 1.10, case 18 was identified as an influential point in the *Coleman Report* data and then case 3 was identified as highly influential. Table 5.7 gives the results of a best subset selection when case 18 has been eliminated. The full model is the best model as measured by either the C_p statistic or the adjusted R^2 value. This is a far cry from the full data analysis in which the models with x_3, x_4 and with x_1, x_3, x_4 had the smallest C_p statistics. These two models are only the seventh and fifth best models in Table 5.7. The two closest competitors to the full model in Table 5.7 involve dropping one of variables x_1 and x_2. The fourth and fifth best models involve dropping x_2 and one of variables x_1 and x_5. In this case, the adjusted R^2 ordering of the five best models agrees with the C_p ordering.

Table 5.7 Best subset regression: Case 18 deleted.

		Adj.			Included variables				
Vars	R^2	R^2	C_p	$\sqrt{RMS}$	x_1	x_2	x_3	x_4	x_5
1	89.6	89.0	21.9	1.9653			X		
1	56.0	53.4	140.8	4.0397		X			
1	53.4	50.6	150.2	4.1595					X
2	92.3	91.3	14.3	1.7414			X	X	
2	91.2	90.1	18.2	1.8635			X		X
2	89.8	88.6	23.0	2.0020		X	X		
3	93.7	92.4	11.4	1.6293			X	X	X
3	93.5	92.2	12.1	1.6573	X		X	X	
3	92.3	90.8	16.1	1.7942		X	X	X	
4	95.2	93.8	8.1	1.4766		X	X	X	X
4	94.7	93.2	9.8	1.5464	X		X	X	X
4	93.5	91.6	14.1	1.7143	X	X	X	X	
5	96.3	94.9	6.0	1.3343	X	X	X	X	X

Table 5.8 gives the best subset summary when cases 3 and 18 have both been eliminated. Once again, the best model as judged by either C_p or adjusted R^2 is the full model. The second best model drops x_1 and the third best model drops x_2. However, the subsequent ordering changes substantially.

Table 5.8 Best subset regression: Cases 3 and 18 deleted.

	Adj.				Included variables				
Vars	R^2	R^2	C_p	$\sqrt{RMS}$	x_1	x_2	x_3	x_4	x_5
1	92.2	91.7	66.5	1.7548			X		
1	57.9	55.3	418.8	4.0688		X			
1	55.8	53.0	440.4	4.1693					X
2	95.3	94.7	36.1	1.4004			X	X	
2	93.2	92.2	58.3	1.6939			X		X
2	92.3	91.2	67.6	1.8023		X	X		
3	96.6	95.8	25.2	1.2412	X		X	X	
3	96.1	95.2	30.3	1.3269			X	X	X
3	95.3	94.3	38.0	1.4490		X	X	X	
4	97.5	96.8	17.3	1.0911		X	X	X	X
4	97.2	96.3	20.8	1.1636	X		X	X	X
4	96.6	95.6	27.0	1.2830	X	X	X	X	
5	98.8	98.3	6.0	0.78236	X	X	X	X	X

Now consider backwards elimination and forward selection with influential observations deleted. In both cases, we continue to use the $|t|$ value 2 as the cutoff to stop addition and removal of variables.

Table 5.9 gives the results of a backwards elimination when case 18 is deleted and when cases 3 and 18 are deleted. In both situations, all five of the variables remain in the model. The regression coefficients are similar in the two models with the largest difference being in the coefficients for x_5. Recall that when all of the cases were included, the backwards elimination model included only variables x_3 and x_4, so we see a substantial difference due to the deletion of one or two cases.

Table 5.9 Backwards elimination.

Case 18 deleted									
Step		Const.	x_1	x_2	x_3	x_4	x_5	R^2	$\sqrt{RMS}$
1	$\hat{\beta}$	34.29	−1.62	0.085	0.674	1.11	−4.6	96.33	1.33
	t_{obs}		−2.04	2.41	10.34	3.98	−3.18		
Cases 18 and 3 deleted									
Step		Const.	x_1	x_2	x_3	x_4	x_5	R^2	$\sqrt{RMS}$
1	$\hat{\beta}$	29.76	−1.70	0.085	0.666	1.18	−4.07	98.83	0.782
	t_{obs}		−3.64	4.09	17.42	7.21	−4.79		

The results of forward selection are given in Table 5.10. With case 18 deleted, the process stops with a model that includes x_3 and x_4. With case 3 also deleted, the model includes x_1, x_3, and x_4. While these happen to agree quite well with the results from the complete data, they agree poorly with the results from best subset selection and from backwards elimination, both of which indicate that all variables

are important. Forward selection gets hung up after a few variables and cannot deal with the fact that adding several variables (rather than one at a time) improves the fit of the model substantially.

Table 5.10 Forward selection.

Case 18 deleted										
Step		Const.	x_1	x_2	x_3	x_4	x_5	R^2	$\sqrt{RMS}$
1	$\hat{\beta}$	32.92			0.604			89.59	1.97
	t_{obs}				12.10				
2	$\hat{\beta}$	14.54			0.585	0.74		92.31	1.74
	t_{obs}				13.01	2.38			
Cases 18 and 3 deleted									
Step		Const.	x_1	x_2	x_3	x_4	x_5	R^2	$\sqrt{RMS}$
1	$\hat{\beta}$	33.05			0.627			92.17	1.75
	t_{obs}				13.72				
2	$\hat{\beta}$	13.23			0.608	0.79		95.32	1.40
	t_{obs}				16.48	3.18			
3	$\hat{\beta}$	10.86	−1.66		0.619	1.07		96.57	1.24
	t_{obs}		−2.26		18.72	4.23			

What is really going on here is that deleting these odd cases makes the *RMS* much smaller, making *everything* look more significant.

5.4 Discussion of Traditional Variable Selection Techniques

Stepwise regression methods are fast, easy, cheap, and readily available. When the number of observations, n, is less than the number of variables, $s+1$, forward selection or a modification of it is the only available method for variable selection. Backward elimination and best subset regression assume that one can fit the model that includes all the predictor variables. This is not possible when $n < s+1$. In fact the use of t statistics, or anything equivalent to them, is probably unwise unless they are associated with a reasonable number of residual degrees of freedom.

There are serious problems with stepwise methods. They do not give the best model (based on any of the criteria we have discussed). In fact, stepwise methods can give models that contain none of the variables that are in the best regressions. That is because, as mentioned earlier, they handle variables one at a time. Another problem is nontechnical. The user of a stepwise regression program will end up with one model. The user may be inclined to think that this is *the* model. It probably is not. In fact, *the* model probably does not exist. Even though Adjusted R^2, Mallows's C_p, AIC, AICc, and BIC all define a unique best model, and could be subject to the

same problem, best subset regression programs generally present several of the best models.

A problem with variable selection methods is that they tend to give models that appear to be better than they really are. For example, the Adjusted R^2 criterion chooses the model with the smallest *RMS*. Because one has selected the smallest *RMS*, the *RMS* for that model is biased toward being too small. Almost any measure of the fit of a model is related to the *RMS*, so the fit of the model will appear to be better than it is. If one could sample the data over again and fit the same model, the *RMS* would almost certainly be larger, perhaps substantially so.

When using Mallows's C_p statistic, if one wants to exploit the virtues of biased estimation, one often picks models with the smallest value of C_p. This can be justified by the fact that the model with the smallest C_p is the model with the smallest estimated expected squared error. However, if you are looking for a correct model the target value of C_p is the number of predictors, so it seems to make little sense to pick the model with the smallest C_p. It seems that one should pick models for which C_p is close to the number of predictors. (I pick models with small C_p.)

The result that, for a fixed number of predictor variables. the best regression criteria are equivalent, is interesting because the various criteria can be viewed as simply different methods of penalizing models that include more variables. The penalty is needed because models with more variables necessarily explain as much or more variation (have as high or higher R^2s).

5.4.1 R^2

R^2 is a good statistic for measuring the predictive ability of a model. R^2 is also a good statistic for comparing models. That is what we used it for here. But the actual value of R^2 should not be overemphasized when it is being used to identify correct models (rather than models that are merely useful for prediction). If you have data with a lot of variability, it is possible to have a very good fit to the underlying regression model without having a high R^2. For example, if the *RSS* admits a decomposition into pure error and lack of fit, it is possible to have very little lack of fit while having a substantial pure error so that R^2 is small while the fit is good.

If transformations of the dependent variable y are considered, it is inappropriate to compare R^2 for models based on different transformations. For example, it is possible for a transformation to increase R^2 without really increasing the predictive ability of the model. One way to check whether this is happening is to compare the width of confidence intervals for predicted values after transforming them to a common scale.

To compare models based on different transformations of y, say $y_1 = f_1(y)$ and $y_2 = f_2(y)$, fit models to the transformed data to obtained predicted values $\hat{y}_1$ and $\hat{y}_2$. Return these to the original scale with $\tilde{y}_1 = f_1^{-1}(\hat{y}_1)$ and $\tilde{y}_2 = f_2^{-1}(\hat{y}_2)$. Finally, define R_1^2 as the squared sample correlation between the ys and the $\tilde{y}_1$s and define R_2^2 as the squared sample correlation between the ys and the $\tilde{y}_2$s. These R^2 values

are comparable (and particularly so when the number of parameters in the two fitted models are comparable).

5.4.2 Influential Observations

Influential observations are a problem in any regression analysis. Variable selection techniques involve fitting lots of models, so the problem of influential observations is multiplied. Recall that an influential observation in one model is not necessarily influential in a different model.

Some statisticians think that the magnitude of the problem of influential observations is so great as to reject all variable selection techniques. They argue that the models arrived at from variable selection techniques depend almost exclusively on the influential observations and have little to do with any real world effects. Most statisticians, however, approve of the judicious use of variable selection techniques. (But then, by definition, everyone will approve of the *judicious* use of anything.)

5.4.3 Exploritory Data Analysis

John W. Tukey, among others, has emphasized the difference between exploratory and confirmatory data analysis. Briefly, *exploratory data analysis (EDA)* deals with situations in which you are trying to find out what is going on in a set of data. Confirmatory data analysis is for proving what you already think is going on. EDA frequently involves looking at lots of graphs. *Confirmatory data analysis* looks at things like tests and confidence intervals. Strictly speaking, you cannot do both exploratory data analysis and confirmatory data analysis on the same set of data.

Variable selection is an exploratory technique. If you know what variables are important, you do not need variable selection and should not use it. When you do use variable selection, if the model is fitted with the same set of data that determined the variable selection, then the model you eventually decide on will give biased estimates and invalid tests and confidence intervals. The biased estimates may very well be better point estimates than a full or correct model gives but tests and confidence intervals are usually over optimistic. Because you typically pick a candidate model partially because it has $RMS(\gamma) < RMS(\beta)$, confidence intervals are too narrow and tests are too significant.

If you can fit the model with all predictor variables and still have a reasonable dfE, it might be reasonable to perform tests and confidence intervals using least squares on the full model but use biased methods for point estimation and prediction. (With many predictors, you still need to use multiple comparison methods.) The alternative seems to be to use asymptotic or ad hoc methods for inference based directly on biased estimates. (Bayesians have the best of both worlds in that a proper

Bayesian analysis both uses biased estimation and has exact small sample inference methods.)

One solution to this problem of selecting variables and fitting parameters with the same data is to divide the data into two parts. Do an exploratory analysis on one part and then a confirmatory analysis on the other. To do this well requires a lot of data. It also demonstrates the problem of influential observations. Depending on where the influential observations are, you can get pretty strange results.

5.4.4 Multiplicities

Methods of statistical inference were originally developed for situations where data were collected to investigate one thing. The methods work well on the one thing. In reality, even the best studies are designed to look at multiple questions, so the original methods need adjustment. Hence the need for the multiple comparisons methods discussed in the next chapter.

In an awful lot of studies, people collect data and muck around with it to see what they can find that is interesting. To paraphrase Seymour Geisser, they ransack the data. When mucking around with a lot of data, if you see something interesting, there is a good chance it is just random variation. Even if there is something there, the true effect is probably smaller than it looks. In this context, if you require a statistical test to show that something is important, it probably isn't important. We saw this with the Big Data examples and those were examples with very clear structures. The general problem in mucking with the data is that to adjust for ransacking you need to keep track of *everything* you looked at that could *possibly* have been interesting. And we are just not psychologically equipped to do that.

5.4.5 Predictive models

Predictive statistical models are based on correlation rather than causation. They work just fine as long as *nothing (important) has changed* from when the data were collected. You wake up, hear the shower on, you know your dad is making breakfast. Hearing the shower is a good predictor of Dad making breakfast. If you wake up from a nap and hear the shower at 2 in the afternoon, do you think Dad will be making breakfast?

What is the causation behind this prediction? Mom showering? It being 7am? Mom having to be to work at 8?

You cannot figure out what a change does to a system without changing the system! Yet *everybody* wants to do just that. They want to solve problems by collecting more data on present conditions. The world doesn't work that way. Without changing the system, you only have (hopefully intelligent) guesswork. But guesswork has

limited value. Evaluating data from current conditions may provide ideas about what changes to try but it provides no assurance of what those changes will accomplish.

5.4.6 *Overfitting*

A big problem with having s large relative to n is the tendency to overfit. *Overfitting* is the phenomenon of fitting a model with so many parameters that the model looks like it fits the data well, e.g. has a high R^2, but does a poor job of predicting future data. Fitting any model with $r(X) = n$ gives $R^2 = 1$, so it is easy to overfit regression models just by taking an X with $r(X) \doteq n$. Our discussion of variable selection was about making X smaller (turning X into a well chosen X_0), so as not to overfit the data. If $r(X) \doteq n$, forward selection could be applied to try to avoid overfitting. When fitting regression trees and other multiple nonparametric regression models (cf. Chapter 7), the set of potential predictor variables is huge and forward selection is used to pick variables that seem appropriate.

Under normal theory for a standard linear model

$$\mathrm{E}(RMS) = \sigma^2, \qquad \mathrm{Var}(RMS) = 2\sigma^4/dfE,$$

so the coefficient of variation (CV) is $\sqrt{2/dfE}$. For RMS is to be a decent estimate, we need CV reasonably small. With $dfE = 2, 8, 18$, CV is 1, 1/2, 1/3, so I would like at least 18 dfE, 8 might be tolerable, and using 2 or less is fraught with danger. (I don't want to think about how often I have failed to live up to that prescription.)

Chapter 3 shows how bad predictions can be from overfitted models but it incidentally shows how bad the estimated variances are from those overfitted models. For everything except the simple linear regression, the estimated variances were well below the target value of 1. We saw the same thing with backward elimination on our big data example.

Some rules I have seen to avoid overfitting require $n \geq 10p$, $n \geq 15p$, or $n \geq 50 + 8p$. These seem like they should work but they seem awfully stringent.

5.5 Modern Forward Selection: Boosting, Bagging, and Random Forests

For many years, forward selection was dismissed as the poor sibling of variable selection. Forward selection provides no assurance that it will find anything like the best models. Backward elimination, since it begins with a presumably reasonable full model and only does reasonable things to that model, should arrive at a decent model. Looking at the "best" subsets of variables seems like the best thing to do. But backward elimination and best subset selection both require being able to fit a reasonable full model.

If the number of predictor variables s is big enough so that $r(X) = n$, we have a saturated full model. Least squares then gives $\hat{Y} = Y$, $SSE = 0$, $dfE = 0$, and the model will be over-fitted so that predictions of new observations typically are poor. Whenever dfE is small, we have probably over-fitted, making our full-model results dubious. In problems with $s \doteq n$ or $s > n$, forward selection, poor as it is, is about the only game in town. (At least in the town of Least Squares Estimation. Principal component regression is another. Penalizing the estimates gets you out of town.)

Boosting, *Bagging*, and *Random Forests* are more recently developed methods of forward selection by which one can use over-fitting of models to improve predictions. Despite all of the difficulties that arose in our *Big Data* examples given earlier, by the standards of this section those examples have extremely well behaved data because $s << n$. Nonetheless, I do not *believe* that the improvements presented here are capable of overcoming the specific foreward selection problem built into the Big Data examples, namely that the importance of the pair of variables is not detectable from either variable separately. (Randomly picking variables for a full model could solve the problem with that example.)

Boosting is a biased estimation technique associated with forward selection. Bagging (bootstrap aggregation) involves use of the *bootstrap* to get more broad based estimates. Random forests are a modification of bagging.

Forward selection starts with some relatively small model and defines a sequence of larger and larger models. The two key features are (1) how to decide which variable gets added next and (2) when to stop adding variables. The traditional method of forward selection ranks variables based on the absolute value of the t statistic for adding them to the current model (or some equivalent statistic) and chooses the highest ranked variable.

If the predictor variables happen to have equal sample variances, forward selection could use the regression coefficients themselves to rank variables, rather than their associated t statistics. In general, using regression coefficients rather than $|t|$ statistics does not seem like a great idea, but in my *quite limited* experience, the procedure works remarkably similar to Tibshirani's (1996) lasso for standardized predictors.

My primary references for this section were James et al. (2013), Hastie, Tibshirani, and Friedman (2016), and Efron and Hastie (2016). This section is different from any other in the book because it contains quite a few of my speculations (clearly marked as such) about how these or related methods *might* work. (The first such speculation occurred in the previous paragragh.)

5.5.1 Boosting

The forward selection method known as *boosting* involves a sequence of model matrices X_j with ppos M_j. The procedure depends on choices for integers d and B and a scalar k.

Perform a forward selection from among the predictor vectors in X to obtain a model with d predictors,

$$Y = X_1\beta_1 + e.$$

From this obtain fitted values and residuals,

$$\hat{Y}_1 = M_1 Y; \quad \hat{e}_1 = Y - \hat{Y}_1.$$

Perform another forward selection using $\hat{e}_1$ as the dependent variable to obtain another model with d predictors,

$$\hat{e}_1 = X_2\beta_2 + e.$$

From this obtain fitted values

$$\tilde{Y}_2 = M_2\hat{e}_1 = M_2(I - M_1)Y.$$

Define overall fitted values

$$\hat{Y}_2 = \hat{Y}_1 + k\tilde{Y}_2$$

and residuals,

$$\hat{e}_2 = Y - \hat{Y}_2 = (I - kM_2)(I - M_1)Y.$$

In general, given residuals $\hat{e}_j$ perform a forward selection using $\hat{e}_j$ as the dependent variable to obtain another model with d predictors,

$$\hat{e}_j = X_{j+1}\beta_{j+1} + e.$$

From this obtain fitted values

$$\tilde{Y}_{j+1} = M_{j+1}\hat{e}_j.$$

Define overall fitted values

$$\hat{Y}_{j+1} = \hat{Y}_j + k\tilde{Y}_{j+1}$$

and residuals,

$$\hat{e}_{j+1} = Y - \hat{Y}_{j+1} = (I - kM_{j+1})\cdots(I - kM_2)(I - M_1)Y.$$

The procedure stops when j reaches the predetermined value B. The accepted wisdom seems to be that picking a stopping point B that is too large can still result in overfitting the model.

If $k = 1$, so that no shrinkage of the estimates is involved, boosting seems like just a lousy way of fitting

$$Y = [X_1, \ldots, X_B]\delta + e.$$

Next we present two adjusted methods that give least squares estimates when $k = 1$.

5.5.1.1 Alternatives

Collect all of the possible predictors into the matrix X. We use ideas related to the sweep operator discussed in Chapter 9. Perform a forward selection from the columns of X to obtain a model with d predictors,

$$Y = X_1\beta_1 + e.$$

From this obtain fitted values and residuals,

$$\hat{Y}_1 = M_1 Y; \quad \hat{e}_1 = Y - \hat{Y}_1.$$

Adjust all the columns of X into $\tilde{X}_2 = (I - M_1)X$. Note that X contains the columns of X_1 but these are zeroed out in $\tilde{X}_2$ and should not be eligible for future selection. This adjustment is not a hideously expensive thing to do. The single expensive operation is computing $(X_1'X_1)^{-1}$. The other operations are numerous but individually inexpensive.

Perform a forward selection using $\hat{e}_1$ as the dependent variable and $\tilde{X}_2$ as the matrix of possible variables to obtain another model with d predictors,

$$\hat{e}_1 = X_2\beta_2 + e.$$

Notice that $C(X_1) \perp C(X_2)$ so $M_1M_2 = 0$. From this obtain overall fitted values

$$\hat{Y}_2 = \hat{Y}_1 + kM_2\hat{e}_1 = (M_1 + kM_2)Y$$

and residuals,

$$\hat{e}_2 = Y - \hat{Y}_2 = (I - M_1 - kM_2)Y.$$

In general, given fitted values $\hat{Y}_j$, residuals $\hat{e}_j$ and the matrices X_j and $\tilde{X}_j$, construct the possible additions $\tilde{X}_{j+1} \equiv (I - M_j)\tilde{X}_j$ in which all variables that are already in the model will have been zeroed out.

Perform a forward selection using $\hat{e}_j$ as the dependent variable with the columns of $\tilde{X}_{j+1}$ as potential predictors to obtain another model with d predictors,

$$\hat{e}_j = X_{j+1}\beta_{j+1} + e.$$

Again, all of the $C(X_k)$s are orthogonal. From this model obtain overall fitted values

$$\hat{Y}_{j+1} = \hat{Y}_j + kM_{j+1}\hat{e}_j = [M_1 + k(M_2 + \cdots + M_{j+1})]Y$$

and residuals

$$\hat{e}_{j+1} = Y - \hat{Y}_{j+1} = (I - kM_{j+1})\hat{e}_j.$$

The accepted wisdom that, picking a stopping point B that is too large can still result in overfitting the model, seems related to the fact that the penalty term k remains the same for every step after the first. An alternative method, akin to exponential smoothing, might do better by defining

$$\hat{Y}_{j+1} = \hat{Y}_j + k^j \tilde{Y}_{j+1} = (M_1 + kM_2 + \cdots + k^j M_{j+1})Y.$$

The alternatives mentioned here are based on the idea that it is the shrinkage of estimates that is valuable in boosting. Although I don't see how it could be true, it is possible that the very awkwardness of adding nonorthogonalized variables could be of some benefit. Boosting was originally developed for binomial regression problems and I *suspect* behaves quite differently there.

5.5.2 Bagging

Bagging is a technique described by Hastie et al. (2016, p.282) as "how to use the bootstrap to improve the estimate or prediction." We will see that bagging can be useful but cannot be a panacea.

The fundamental idea of *bagging* (as I see it) follows: Suppose you have an algorithm for fitting a model to a set of data with n observations. In bagging this should be an algorithm that tends to overfit the data. Take a random sample with replacement of size n from your data. Apply your algorithm on this sample of data and obtain your desired results: predictions or estimates. Do this repeatedly for many random samples and average your predictions/estimates over these samples. These averages are the result of bagging. The hope is that these averages will be better predictions and estimates than the results of the original algorithm applied just once to the original data. We will see, in a simple example, that the better the algorithm, the less likely this is to be true. But in situations where we do not know how to create a good algorithm for the particular data, bagging can provide valuable improvements.

With reasonably large collections of data, the gold standard for determining the quality of a predictive model seems to be: (1) randomly pull out a set of *test data*, (2) use the remaining *training data* to develop the predictive model, and (3) evaluate the predictive model by seeing how well it predicts the test data. This is frequently used to compare the quality of various methods of developing predictive models. In this context, overfitting consists of fitting a model that explains the training data very well but does a poor job of predicting the test data.

If we think of the collection of observed data as the entire population of possible data, and randomly select the test data, then the training data is also just a random sample from the population. It should display the same predictive relationships as the overall population. If the goal is to predict a random sample (the test sample) from the population, why not use random samples from the population to develop a predictor? Moreover, we want to use overfitting to help, rather than hinder, the predictive process.

The idea is to start with a model selection procedure that is capable of modeling the salient features in the data. In multiple nonparametric regression that oftn involves constructing additional predictor variables that make the number of predictor variables s very large indeed. Different nonparametric regression procedures have different modeling capabilities (they create different model matrices X), so the best

choice of a procedure depends on the nature of the data in ways that we will rarely understand beforehand. But merely having an X matrix with $s >> n$ is not enough. To develop a prediction model we must have some variable selection scheme, presumably a form of forward selection, that includes enough predictor variables to capture the salient features of the data, and to do this consistently seems to require some overfitting.

Take a random sample from the training data and overfit a model on it. This ensures that the salient features that are present in all samples are caught but overfitting will also include features that are unique to the particular sample being fitted. Do this for many random samples (say B) and average the results. The hope is that the salient features will appear in a similar fashion in every sample but that the unique (random) features that occur from overfitting particular samples will average themselves out over the process of repeated sampling.

Bagging is a very complicated procedure. In fact, it is notorious for providing (good) predictions that are uninterpretable. We now examine an extremely simple example of bagging to explore how it actually works.

5.5.2.1 A simple example

Consider a random variable y. With no potential predictor variables available, the best predictor (BP) under squared error prediction loss is $\mathrm{E}(y) = \mu_y \equiv \mu$. Now suppose we have a random sample from y, say $Y_{n\times 1}$. The best nonparametric estimate of μ is $\bar{y}_\cdot = J'Y/n$. We know it is the BLUE but for a class of distributions that includes nearly all continuous distributions that have a mean, $\bar{y}_\cdot$ is minimum variance unbiased, cf. Fraser (1957). If the distribution of y is normal, $\bar{y}_\cdot$ is again minimum variance unbiased but it is much easier to be unbiased for normal distributions than it is for all continuous distributions. For uniform distributions with unknown limits, the best unbiased estimate of the expected value is the midrange. For symmetric distributions with heavy tails, e.g. Laplace (double exponential), the median tends to be a good estimate. In this context, $\bar{y}_\cdot$ is always going to be a reasonably good estimate of the BP. Generally, for light-tailed symmetric distributions, like the uniform with unknown limits, the midrange can be a good estimate of the BP but the median will be less good. For heavy-tailed symmetric distributions, the median can be a good estimate of the BP but the midrange will be less good. Neither the median nor the midrange are linear functions of the data.

Bagging brings an element of averaging into the estimates that has virtually no effect on the linear estimate $\bar{y}_\cdot$ and cannot improve an optimal estimate, but bagging can substantially improve a poor estimate and can even improve good but suboptimal estimates.

EXAMPLE 5.5.1. In the spirit of fitting a number of parameters that is a large proportion of the number of observations (and just to be able to perform the computations), suppose we have a simple random sample of size $n = 3$ with order statistics $y_{(1)} < y_{(2)} < y_{(3)}$. The best predictor is the population mean, which we want to

estimate. The midrange $(y_{(1)} + y_{(3)})/2$ is optimal for uniform distributions with unknown end points. The median $y_{(2)}$ works well for heavy tailed distributions. The sample mean $(y_{(1)} + y_{(2)} + y_{(3)})/3$ is optimal for normal data or extremely broad (nonparametric) families of distributions.

Normally one would not bootstrap a sample this small but the small sample size allows us to examine what Hastie et al. (2016) refer to as the *"true" bagging estimate*. With only 3 observations, bootstrapping takes samples from a population that has 27 equally probable outcomes: Three of the outcomes are $\{y_{(j)}, y_{(j)}, y_{(j)}\}$ for $j = 1,2,3$. Six of the outcomes are reorderings of $\{y_{(1)}, y_{(2)}, y_{(3)}\}$. The other 18 outcomes involve the three reorderings one can get from samples of the form $\{y_{(j)}, y_{(j)}, y_{(k)}\}$, $j = 1,2,3$, $k \neq j$ there being six distinct outcomes of this form.

From each of the 27 outcomes in the bootstrap population we can compute the sample mean, median, and midrange statistics. The bootstrap procedure actually provides an estimate (one that we can make arbitrarily good by picking B large) of the expected value over the 27 equally probable outcomes of these statistics (sample mean, median and midrange). We want to know what function of the observed data the bootstrap is estimating, because that is the function of the data that the bootstrap uses to estimate the BP (as $B \to \infty$).

The expected value of the sample mean is easily seen to be $(9y_{(1)} + 9y_{(2)} + 9y_{(3)})/27$, so unsurprisingly the bootstrap of the sample mean is estimating the sample mean of the original data. The bootstrap expected value of the sample median is $(7y_{(1)} + 13y_{(2)} + 7y_{(3)})/27$, which is a symmetric weighted average of the original observations; one that puts more weight on the middle observation. The bootstrap expected value of the midrange is $(10y_{(1)} + 7y_{(2)} + 10y_{(3)})/27$, which is again a symmetric weighted average of the original observations but one that puts less weight on the middle observation.

Another way to think about this is that the bagged median is estimating

$$(14/27)\text{midrange} + (13/27)\text{median}$$

and the bagged midrange is estimating

$$(20/27)\text{midrange} + (7/27)\text{median}.$$

But perhaps more importantly, *both of them are closer to the sample mean than they were originally*.

For a uniform distribution, where the midrange is optimal, the bagged midrange estimate will be less good because it puts too much weight on the middle observation. However, the bagged median will be better than the median because the bagged median puts more weight on the midrange.

When the median is good, the bagged median will be less good because it puts more weight on the extreme observations. However the bagged midrange will be better than the midrange because the bagged midrange puts more weight on the median.

Bagging the sample mean is a waste of effort. Because the sample mean is the best nonparametric estimate, the sample mean is never going to be too bad.

If you don't know the distribution of the data, which you almost never do, you might as well use the sample mean and bagging is irrelevant. Bagging would be useful if for some reason you cannot use the sample mean.

The three estimates we examined were all unbiased for symmetric distributions. Lets look at a biased estimate of the mean. Consider the estimate $(y_{(2)} + y_{(3)})/2$ which is clearly biased above the mean for symmetric distributions. The bootstraped estimate has expected value $(4y_{(1)} + 10y_{(2)} + 13y_{(3)})/27$, which, while still heavily biased above the mean, is considerably less biased than the original estimate.

5.5.2.2 Discussion

The prediction problem is to estimate the best predictor, $\mathrm{E}(y|x)$. The more you know about the conditional distribution of y given x, the easier the problem becomes. By expanding our definition of x, e.g. incorporating polynomials, we can often ensure that $\mathrm{E}(y|x)$ is approximately linear in x. If we have enough data to fit a full model, we should. For homoscedastic, uncorrelated data, least squares estimates are BLUEs, so they are probably about as good as we can do to start, but then we may be able to get better point estimates by incorporating bias. If $s > n$, we cannot fit the full model in any meaningful way, so we need some way of constructing a predictive model and typically that involves some form of forward selection. We know forward selection does not work well, so it is unlikely to give good estimates of the best predictor. When you have poor estimates, bagging seems to be good at improving them by averaging them over more of the data. (My *hope* is that the bagging estimates will move them closer to the least squares estimates from the [unfitable] full model.) While it is relatively easy to see that bagging has no systematic effect on estimates that are linear functions of the data, forward selection is a profoundly nonlinear estimation process.

5.5.3 Random Forests

The *random forest* idea modifies the forward selection procedure in conjunction with bagging. The name derives from applying the idea to regression trees.

Divide the predictor variables into G groups. The modification to forward selection is that instead of considering all of the variables as candidates for selection, one randomly chooses m of the G groups as candidates for forward selection. If you were only fitting one model, that would be a disastrous idea, but in the context of bagging, all of the important variables should show up often. Typically one takes $m \doteq G/3$ or $m \doteq \sqrt{G}$.

EXAMPLE 5.5.2. *Polynomial Regression.* Division into G groups occurs naturally in polynomial regression and many other nonparametric regression procedures that, like polynomial regression, begin with, say Q, measured predictor variables

and define functions of those measured variables. A full polynomial model on Q measured variables $x_1, \ldots, x_Q$ is

$$y_i = \sum_{j_1=0}^{d_1} \cdots \sum_{j_Q=0}^{d_Q} \beta_{j_1 \cdots j_s} x_{i1}^{j_1} \cdots x_{iQ}^{j_Q} + \varepsilon_i,$$

so the total number of predictor variables is $s = \prod_{j=1}^{Q} d_j$. The G groups can conveniently be taken as $x_k^{j_k} : j_k = 1, \ldots, d_k$ for $k = 1, \ldots, Q$ which makes $G = Q$. Instead of considering all of the variables $x_k^{j_k}$, $j_k = 1, \ldots, d_k$, $k = 1, \ldots, Q$ as candidates for selection, one randomly chooses m of the Q groups as candidates for forward selection. (Forward selection typically will not result in a hierarchical polynomial that contains all the lower order terms for every term in the polynomial. Although good arguments can be made for using hierarchical polynomials, they seem inconsistent with the spirit of forward selection.) Alternatively, one could pick $d_1 = \cdots = d_Q \equiv G$, and have the group be all of the linear terms, the second group the quadratics, etc.

EXAMPLE 5.5.3. *Big Data.* Divide the 100 predictors into 9 groups all but one having 11 predictor variables. Randomly pick 3 groups to be included in the variable selection. In this example, with almost everything being independent, it is hard to see how the random forest idea is going to help. Much of the time the two worthwhile predictors will not even be available for selection in the model. And even when the two good predictors are both available, nothing has happened that will increase their chances of being selected. Remember, we have to have an algorithm that randomly gets one of the two predictors into the model. Once one of them is in the model, forward selection will find the second variable (if it is available to find). So the random forest idea (or bagging alone) does not seem to help in this example, but then forward selection on these data is not a method well suited for finding the salient characteristics at the expense of some overfitting.

I would be tempted to define subgroups of variables for which a random selection would give something I consider a plausible full model and rather than averaging them all, actually look for good ones. But in the era of big data, there seems to be a premium on procedures you can run without having to supervise them.

5.6 Exercises

EXERCISE 5.6.1. Reconsider the advertising data of Exercise 1.12.1.

(a) Are there any high-leverage points? Why or why not?
(b) Test whether each case is an outlier using an overall significance level no greater than $\alpha = 0.05$. Completely state the appropriate reference distribution.
(c) Discuss the importance of Cook's distances in regard to these data.

(d) Using only analysis of variance tables, compute R^2, the adjusted R^2, and the C_p statistic for $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$. Show your work.
(e) In the three-variable model, which if any variable would be deleted by a backwards elimination method? Why?

EXERCISE 5.6.2. Consider the information given in Table 5.11 on diagnostic statistics for the wood data of Exercise 1.12.2.

(a) Are there any outliers in the predictor variables? Why are these considered outliers?
(b) Are there any outliers in the dependent variable? If so, why are these considered outliers?
(c) What are the most influential observations in terms of the predictive ability of the model?

Table 5.11 Diagnostics for wood data.

Obs.	Leverage	r	t	C	Obs.	Leverage	r	t	C
1	0.085	−0.25	−0.25	0.001	29	0.069	0.27	0.26	0.001
2	0.055	1.34	1.35	0.021	30	0.029	0.89	0.89	0.005
3	0.021	0.57	0.57	0.001	31	0.204	0.30	0.30	0.005
4	0.031	0.35	0.35	0.001	32	0.057	0.38	0.37	0.002
5	0.032	2.19	2.28	0.032	33	0.057	0.05	0.05	0.000
6	0.131	0.20	0.19	0.001	34	0.085	−2.43	−2.56	0.109
7	0.027	1.75	1.79	0.017	35	0.186	−2.17	−2.26	0.215
8	0.026	1.23	1.24	0.008	36	0.184	1.01	1.01	0.046
9	0.191	0.52	0.52	0.013	37	0.114	0.85	0.85	0.019
10	0.082	0.47	0.46	0.004	38	0.022	0.19	0.19	0.000
11	0.098	−3.39	−3.82	0.250	39	0.022	−0.45	−0.45	0.001
12	0.066	0.32	0.32	0.001	40	0.053	−1.15	−1.15	0.015
13	0.070	−0.09	−0.09	0.000	41	0.053	0.78	0.78	0.007
14	0.059	0.08	0.08	0.000	42	0.136	−0.77	−0.76	0.018
15	0.058	−0.91	−0.91	0.010	43	0.072	−0.78	−0.77	0.009
16	0.085	−0.09	−0.09	0.000	44	0.072	−0.27	−0.26	0.001
17	0.113	1.28	1.29	0.042	45	0.072	−0.40	−0.40	0.002
18	0.077	−1.05	−1.05	0.018	46	0.063	−0.62	−0.62	0.005
19	0.167	0.38	0.38	0.006	47	0.025	0.46	0.46	0.001
20	0.042	0.24	0.23	0.000	48	0.021	0.18	0.18	0.000
21	0.314	−0.19	−0.19	0.003	49	0.050	−0.44	−0.44	0.002
22	0.099	0.56	0.55	0.007	50	0.161	−0.66	−0.66	0.017
23	0.093	0.47	0.46	0.004	51	0.042	−0.44	−0.43	0.002
24	0.039	−0.60	−0.60	0.003	52	0.123	−0.26	−0.26	0.002
25	0.098	−1.07	−1.07	0.025	53	0.460	1.81	1.86	0.558
26	0.033	0.14	0.13	0.000	54	0.055	0.50	0.50	0.003
27	0.042	1.19	1.19	0.012	55	0.093	−1.03	−1.03	0.022
28	0.185	−1.41	−1.42	0.090					

EXERCISE 5.6.3. Consider the information in Table 5.12 on best subset regression for the wood data of Exercise 1.12.2.

Table 5.12 Best subset regression of wood data.

		Adj.			Included variables			
Vars	R^2	R^2	C_p	$\sqrt{RMS}$	x_1	x_2	x_3	x_4
1	97.9	97.9	12.9	18.881	X			
1	63.5	62.8	1064.9	78.889				X
1	32.7	31.5	2003.3	107.04			X	
2	98.3	98.2	3.5	17.278	X		X	
2	97.9	97.8	14.3	18.969	X	X		
2	97.9	97.8	14.9	19.061	X			X
3	98.3	98.2	5.3	17.419	X	X	X	
3	98.3	98.2	5.4	17.430	X		X	X
3	98.0	97.9	13.7	18.763	X	X		X
4	98.4	98.2	5.0	17.193	X	X	X	X

(a) In order, what are the three best models as measured by the C_p criterion?
(b) What is the residual mean square for the model with variables x_1, x_3, and x_4?
(c) In order, what are the three best models as measured by the adjusted R^2 criterion? (Yes, it is possible to distinguish between the best four!)
(d) What do you think are the best models and what would you do next?

EXERCISE 5.6.4. Consider the information in Table 5.12 on stepwise regression for the wood data of Exercise 1.12.2.

Table 5.13 Stepwise regression on wood data.

STEP	1	2	3
Constant	23.45	41.87	43.85
x_1	0.932	1.057	1.063
t	10.84	38.15	44.52
x_2	0.73	0.09	
t	1.56	0.40	
x_3	−0.50	−0.50	−0.51
t	−3.28	−3.27	−3.36
x_4	3.5		
t	1.53		
$\sqrt{RMS}$	17.2	17.4	17.3
R^2	98.36	98.29	98.28

(a) What is being given in the rows labeled x_1, x_2, x_3, and x_4? What is being given in the rows labeled t?

(b) Is this table for forward selection, backwards elimination, stepwise regression, or some other procedure?
(c) Describe the results of the procedure.

EXERCISE 5.6.5. Reanalyze the Prater data of Atkinson (1985) and Hader and Grandage (1958) from Exercise 1.12.3. Examine residuals and influential observations. Explore the use of the various model selection methods.

EXERCISE 5.6.6. Reanalyze the Chapman data of Exercise 1.12.4. Examine residuals and influential observations. Explore the use of the various model selection methods.

EXERCISE 5.6.7. Reanalyze the pollution data of Exercise 1.12.5. Examine residuals and influential observations. Explore the use of various model selection methods.

EXERCISE 5.6.8. Repeat Exercise 1.12.6 on the body fat data with special emphasis on diagnostics and model selection.

EXERCISE 5.6.9. Compare the results of using the modern forward selection techniques on the *Coleman Report* data.

Chapter 6
Multiple Comparison Methods

Abstract This chapter presents three methods of adjusting statistical tests to deal with the problem, illustrated in the previous chapter, of claiming that garbage is important.

There has been a great deal of controversy in the Statistics community about the proper use of P values. This has be exacerbated by at least two things: the prevalence of big data and widespread misunderstanding of what P values mean. (The second of these is by no means new.)

To compute an appropriate P value requires one to know an *appropriate* reference distribution for the test statistic. Finding an appropriate reference distribution can be difficult or even impossible. One thing that is difficult is finding an appropriate distribution when you are trying to test several distinct things simultaneously. This chapter deals with three widely applicable adjustments that can be used when an appropriate reference distribution is not readily available for multiple tests. (When dealing with balanced ANOVA, many more adjustment methods are available.)

As discussed in *ANREG*, Chapter 3 and *PA*, Appendix E, most statistical tests and confidence intervals are based on identifying

1. Par, a parameter of interest;
2. Est, an estimate of Par (that is usually easy to find);
3. $\mathrm{SE}(Est)$, a standard error for Est (that involves computing the variance of Est and often requires estimating a variance parameter σ^2);
4. an appropriate reference distribution for

$$\frac{Est - Par}{\mathrm{SE}(Est)}.$$

For a single test or confidence interval, the appropriate reference distribution is typically a $t(df)$ distribution where df denotes the degrees of freedom of the variance parameter estimate. (df can be thought of as the equivalent number of observations going into the variance estimate). *When making multiple tests, the appropriate reference distribution changes and may be unknowable.*

6.1 Bonferroni Corrections

The Bonferroni method is very simple. If you have r tests to perform and you want to keep an overall error rate of α for the entire collection of tests, perform each test at the α/r level. We already used this technique in Section 1.10 without explaining it. There we tested whether a standardized deleted (t) residual was far enough from zero to call in question the inclusion of that case in the data. Typically you only think about testing a particular case because you see that $|t_i|$ is large but the $t(dfE-1)$ reference distribution is no longer appropriate for the largest $|t_i|$ among $i = 1,\ldots,n$. Instead of looking at the P value, we looked at P/n.

Before addressing the justification for the Bonferroni method, we apply the idea to the big data examples of the previous chapter.

EXAMPLE 6.1.1. *Big Data.*
Our tiny little "big data" example in the previous chapter involved $n = 1000$ and $s = 100$. When looking for significant predictors in a model with 100 predictor variables, instead of looking at variables whose regression coefficient P values have, say, $P \leq 0.05$, perhaps we should be looking for $P < 0.0005 = 0.05/100$. Moreover, when performing variable selection, it may be a good idea to retain this requirement, i.e., base the Bonferroni adjustment on $s = 100$, the total number of variables considered, rather than p, the final number of variables in the selected candidate model. In Example 5.2.2 on forward selection, none of the variables achieved a P value below 0.0005. In Example 5.2.4 on backward elimination, the only candidates for achieving a level of significance of 0.0005 are the two important variables, but the rounding off of the P values to three digits leaves us unsure. (They *are* Bonferroni significant.) □

Unfortunately, we will see in the next chapter that in many problems the choice of s is often pretty arbitrary. Using $r = n$ for a Bonferroni adjustment might not be unreasonable if you get into the business of creating predictor variables, e.g., if your total number of possible predictors has $s > n$.

The justification for Bonferroni's method relies on a very simple result from probability: for two events, the probability that one or the other event occurs is no more than the sum of the probabilities for the individual events. Thus with two tests, say A and B, the probability that we reject A or reject B is less than or equal to the probability of rejecting A plus the probability of rejecting B. In particular, if we fix the probability of rejecting A at $\alpha/2$ and the probability of rejecting B at $\alpha/2$, then the probability of rejecting A or B is no more than $\alpha/2+\alpha/2 = \alpha$. More generally, if we have r tests and control the probability of type I error for each test at α/r, then the probability of rejecting any of the tests when all r null hypotheses are true is no more than $\alpha/r+\cdots+\alpha/r = \alpha$.

Bonferroni adjustments can also be used to obtain confidence intervals that have a simultaneous confidence of $(1-\alpha)100\%$ for covering all parameters in some set of r parameters. The endpoints of these intervals are

$$Est \pm t\Big(1-\frac{\alpha}{2r}, dfE\Big)\,\mathrm{SE}(Est)\,.$$

In particular, for testing the nonintercept regression parameters this becomes,

$$\hat{\beta}_j \pm t\Big(1-\frac{\alpha}{2s}, dfE\Big)\,\mathrm{SE}\Big(\hat{\beta}_j\Big)\,.$$

6.2 Scheffé's method

This is the most conservative method known to Mankind. (Yes, I am being bombastic. No, I am not referring to the knowledge of former professional wrestler Mick Foley.) Instead of adjusting the P or α levels as Bonferroni does, Scheffé's method adjusts the test statistics. When looking for significant predictors in a model with s predictor variables, instead of looking at the t statistics for the regression variables, say t_j, $j = 1,\ldots,s$, Scheffé's method has us look at $t_j/\sqrt{s}$. Strictly speaking, for an α level test, $t_j/\sqrt{s}$ should be compared to $\sqrt{F(1-\alpha,s,dfE)}$ but for $dfE > 2$, $t\big(1-\frac{\alpha}{2r},dfE\big) \geq \sqrt{F(1-\alpha,s,dfE)}$, so it is permissible to think about comparing the modified test statistics to the regular t percentiles.

EXAMPLE 6.2.1. *Big Data.*
With $s = 100$, the square root is 10, so before thinking about what variables look important, divide the t statistics by 10. Again, when performing variable selection, it may be a good idea to retain this requirement, i.e., base significance judgements for regression coefficients for a selected candidate model, not on the original t statistic, or on $t/\sqrt{p}$, but on $t/\sqrt{s}$. In Example 5.2.2 on forward selection, the regression coefficient t_j statistics divided by 10 lose all hope of being interpreted as significant. In Example 5.2.4 on backward elimination, after dividing by 10 the only variables whose t_j statistics still look important are the two that actually are important. □

We now run through the justification for Scheffé's method. The justification is based on F rather than t statistics. The earlier discussion was based on the fact that $t(df)^2 \sim F(1,df)$.

Suppose we have some hierarchy of models that includes a biggest model (Big.), some full model (Full), a reduced model (Red.), and a smallest model (Sml.). In most hierarchies of models, there are many choices for Full and Red. but Big. and Sml. are fixed. Scheffé's method can be used to perform tests on a fixed set of choices for Full and Red., or on all possible choices for Full and Red., or on a few choices determined by the data.

In Chapter 1, we introduced model testing for a full and reduced model using the F statistic

$$F = \frac{[SSE(Red.) - SSE(Full)]/[dfE(Red.) - dfE(Full)]}{MSE(Full)}$$

with reference distribution $F[dfE(Red.) - dfE(Full), dfE(Full)]$. As we got into hierarchies of models, we preferred the statistic

$$F = \frac{[SSE(Red.) - SSE(Full)]/[dfE(Red.) - dfE(Full)]}{MSE(Big.)}$$

with reference distribution $F[dfE(Red.) - dfE(Full), dfE(Big.)]$. Scheffé's method requires a further modification of the test statistic.

If the smallest model is true, then all of the other models are also true. The experimentwise error rate is the probability of rejecting any reduced model Red. (relative to a full model Full) when model Sml. is true. Scheffé's method allows us to compare any and all full and reduced models, those we even pick by looking at the data, and controls the experimentwise error rate at α by rejecting the reduced model only when

$$F = \frac{[SSE(Red.) - SSE(Full)]/[dfE(Sml.) - dfE(Big.)]}{MSE(Big.)} > F[1-\alpha, dfE(Sml.) - dfE(Big.), dfE(Big.)].$$

To justify this procedure, note that the test of the smallest model versus the biggest model rejects when

$$F = \frac{[SSE(Sml.) - SSE(Big.)]/[dfE(Sml.) - dfE(Big.)]}{MSE(Big.)} > F[1-\alpha, dfE(Sml.) - dfE(Big.), dfE(Big.)]$$

and when the smallest model is true, this has only an α chance of occurring. Because

$$SSE(Sml.) \geq SSE(Red.) \geq SSE(Full) \geq SSE(Big.),$$

we have

$$[SSE(Sml.) - SSE(Big.)] \geq [SSE(Red.) - SSE(Full)]$$

and

$$\frac{[SSE(Sml.) - SSE(Big.)]/[dfE(Sml.) - dfE(Big.)]}{MSE(Big.)} \geq \frac{[SSE(Red.) - SSE(Full)]/[dfE(Sml.) - dfE(Big.)]}{MSE(Big.)}.$$

It follows that you cannot reject Red. relative to Full unless you have already rejected Sml. relative to Big., and rejecting Sml. relative to Big. occurs only with probability α when Sml. is true. In other words, there is no more than an α chance of rejecting any of the reduced models when they are true.

Scheffé's method can be extended to examining any and all linear combinations of the regression coefficients simultaneously. *This method is primarily used with linear combinations that were suggested by the data*. In particular, Scheffé's method

can be adapted to provide simultaneous $(1-\alpha)100\%$ confidence intervals. These have the endpoints

$$\sum_{j=0}^{s} \lambda_i \hat{\beta}_j \pm \sqrt{(s+1)F(1-\alpha, s+1, dfE)}\,\mathrm{SE}\left(\sum_{j=0}^{s} \hat{\beta}_j\right).$$

In this case the Sml. model is being taken as $y_i = \varepsilon_i$. If you exclude the intercept from the linear combinations, the intervals become

$$\sum_{j=1}^{s} \lambda_i \hat{\beta}_j \pm \sqrt{sF(1-\alpha, s, dfE)}\,\mathrm{SE}\left(\sum_{j=1}^{s} \hat{\beta}_j\right).$$

In particular, for individual regression coefficients they become

$$\hat{\beta}_j \pm \sqrt{sF(1-\alpha, s, dfE)}\,\mathrm{SE}\left(\hat{\beta}_j\right).$$

Here the Sml. model is $y_i = \beta_0 + \varepsilon_i$.

6.3 Least Significant Differences

To apply the Least Significant Difference method, you fit the full model and perform an α level test for any effect over and above the intercept. If this test is significant, you perform the rest of your tests in the standard way at the α level. When dealing with big data, this method is essentially worthless; it provides no useful solution to the problem of finding far too many things to be significant.

EXAMPLE 6.3.1. *Big Data.*
Chapter 5 does not report the fit for the full model but, if it did, it would give a significant result at $\alpha = 0.05$ for the $F(100, 899)$ test of $SSReg/RMS$. The least significant difference method applied to the big data example would then claim that any variable significant at the 0.05 level is important, which we know is not true.

Chapter 7
Nonparametric Regression II

Abstract This chapter deepens our discussion of nonparametric regression. It details methods for a single predictor variable, discusses the curse of dimensionality that plagues nonparametric regression with multiple predictor variables, and discusses the kernel trick and related ideas as methods for overcoming the curse of dimensionality. The methods considered are all extensions of linear regression.

Suppose we have a dependent variable y and a vector of predictor variables x. Regression is about estimating $\mathrm{E}(y|x)$. In linear regression, we assume that $\mathrm{E}(y|x) = x'\beta$ for some unknown parameter vector β. Recall that this includes fitting indicator variables and polynomials as special cases. In nonlinear regression we assume that $\mathrm{E}(y|x) = f(x;\beta)$, where the function f is known but the vector β is unknown; see Christensen (1996, Chapter 18 or 2015, Chapter 23). A special case of nonlinear regression involves linearizable models, including generalized linear models, that assume $\mathrm{E}(y|x) = f(x'\beta)$ for f known, cf. Christensen (1997, Chapter 9). The key idea in nonlinear regression is using calculus to linearize the model. In *nonparametric regression*, we assume that $\mathrm{E}(y|x) = f(x)$, where the function f is unknown. Note the absence of a vector of parameters β, hence the name nonparametric. Often, f is assumed to be continuous or to have some specified number of derivatives. In reality, nonparametric regression is exactly the opposite of what its name suggests. Nonparametric regression involves fitting far more parameters than either standard linear or nonlinear regression.

EXAMPLE 7.0.1. Table 7.1 presents data from Montgomery and Peck (1982) and Eubank (1988) on voltage drops y over time t displayed by an electrical battery used in a guided missile. The 41 times go from 0 to 20. The variable x results from dividing t by 20, thus standardizing the times into the $[0,1]$ interval. The data comprise a time series (as discussed in *ALM-III*, Chapters 6 and 7), but the idea here is that the behavior over time is not a stationary stochastic process but rather a complicated regression function. An unusual feature of these data is that the t_i values are equally spaced (i.e., the t_is are ordered and $t_{i+1} - t_i$ is a constant). This typically occurs only when the data collection process is very well-controlled. However, when equal spacing does occur, it considerably simplifies data analysis. □

Table 7.1 Battery voltage drops versus time.

Case	y	t	x	Case	y	t	x
1	8.33	0.0	0.000	22	14.92	10.5	0.525
2	8.23	0.5	0.025	23	14.37	11.0	0.550
3	7.17	1.0	0.050	24	14.63	11.5	0.575
4	7.14	1.5	0.075	25	15.18	12.0	0.600
5	7.31	2.0	0.100	26	14.51	12.5	0.625
6	7.60	2.5	0.125	27	14.34	13.0	0.650
7	7.94	3.0	0.150	28	13.81	13.5	0.675
8	8.30	3.5	0.175	29	13.79	14.0	0.700
9	8.76	4.0	0.200	30	13.05	14.5	0.725
10	8.71	4.5	0.225	31	13.04	15.0	0.750
11	9.71	5.0	0.250	32	12.06	15.5	0.775
12	10.26	5.5	0.275	33	12.05	16.0	0.800
13	10.91	6.0	0.300	34	11.15	16.5	0.825
14	11.67	6.5	0.325	35	11.15	17.0	0.850
15	11.76	7.0	0.350	36	10.14	17.5	0.875
16	12.81	7.5	0.375	37	10.08	18.0	0.900
17	13.30	8.0	0.400	38	9.78	18.5	0.925
18	13.88	8.5	0.425	39	9.80	19.0	0.950
19	14.59	9.0	0.450	40	9.95	19.5	0.975
20	14.05	9.5	0.475	41	9.51	20.0	1.000
21	14.48	10.0	0.500				

Section 1 examines the basics of the linear-aproximation approach. This involves approximating general continuous functions by linear combinations of *basis functions*, or more properly, *spanning functions*. Function series whose elements have small support, i.e. are zero most but not all of the time, seem particularly useful. These include splines and wavelets. In Section 2 we examine the fact that these approaches to nonparametric regression involve fitting linear regression models. In Section 3, we discuss and illustrate least squares estimation. In Section 4, we discuss variable selection as applied to linear-approximation models. Section 5 discusses details of splines and introduce kernel estimation and other local polynomial regression techniques. Section 6 introduces nonparametric multiple regression. Section 7 examines testing lack of fit. Section 8 looks at regression trees. Section 9 introduces the use of functional predictors. Section 10 provides exercises. (This is pretty much just a watered down version of *ALM-III*, Chapter 1.)

7.1 Linear Approximations

The key idea behind linear approximations is that a finite linear combination of some known functions can approximate a wide variety of functions on a closed bounded set, cf. the famous Stone-Weierstrass theorem. For convenience, we initially assume that f is defined on the interval $[0,1]$ and is continuous. There are many ways to

approximate f including polynomials, sines and cosines, step functions, and also by things similar to step functions called *wavelets*. Most often we assume that for some predictor variable x

$$f(x) = \sum_{j=0}^{\infty} \beta_j \phi_j(x),$$

where the ϕ_js are known functions that can be defined in many ways. Later we will use this characterization with x being a p vector instead of a scalar. In particular, with $p = 1$ and functions defined on the unit interval, we can take for $j = 0, 1, 2, \ldots$

$$\phi_j(x) = x^j, \tag{1}$$

or

$$\phi_j(x) = \cos(\pi j x), \tag{2}$$

or

$$\phi_{2j}(x) = \cos(\pi j x) \quad \phi_{2j+1}(x) = \sin(\pi j x). \tag{3}$$

When using (2), it should be noted that the derivative of every $\cos(\pi j x)$ function is 0 at $x = 0$, so the derivative of $f(x)$ should be 0 at $x = 0$.

In practice we approximate f with a finite number of terms which determines a linear model in which only the β_js are unknown. We need to determine an appropriate finite approximation and estimate the corresponding β_js

With a single predictor, another obvious approximation uses step functions but some care must be used. Let $\mathscr{I}_A$ be the *indicator function* for the set A, namely

$$\mathscr{I}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{otherwise.} \end{cases}$$

Obviously, if we define

$$\phi_j(x) = \mathscr{I}_{(\frac{j-1}{m}, \frac{j}{m}]}(x), \quad j = 0, 1, \ldots, m,$$

we can approximate any continuous function f, and as $m \to \infty$ we can approximate f arbitrarily well. Note that $\phi_0(x)$ is essentially $\mathscr{I}_{\{0\}}(x)$. Technically, rather than the infinite sum characterization, we are defining a *triangular array* of functions ϕ_{jm}, $j = 1, \ldots, m$; $m = 1, 2, 3, \ldots$ and assuming that

$$f(x) = \lim_{m \to \infty} \sum_{j=0}^{m} \beta_{jm} \phi_{jm}(x). \tag{4}$$

More generally, we could define the indicator functions using intervals between *knots*, $\tilde{x}_{-1,m} < 0 = \tilde{x}_{0,m} < \tilde{x}_{1,m} < \tilde{x}_{2,m} < \cdots < \tilde{x}_{m,m} = 1$ with the property that $\max_i \{\tilde{x}_{i+1,m} - \tilde{x}_{i,m}\}$ goes to zero as m goes to infinity.

Splines are more complicated than indicator functions. Choosing $m - 1$ knots in the interior of $[0, 1]$ is fundamental to the use of splines. Rather than indicators, we can fit some low dimensional polynomial between the knots. In this context, indicator functions are 0 degree polynomials. For polynomials of degree greater

than 0, traditional splines force the polynomials above and below each knot in (0,1) to take the same value at the knot, thus forcing the splines to give a continuous function on [0,1]. *B-splines* use functions ϕ_{jm} that are nonzero only on small but overlapping subintervals with locations determined by (often centered around) a collection of knots. As with indicator functions, to get good approximations to an arbitrary regression function, the distances between consecutive knots must all get (asymptotically) small. As a practical matter, one tries to find one appropriate set of knots for the problem at hand. Technically, methods based on knots are not basis function methods because they do not provide a countable set of functions that are linearly independent and span the space of continuous functions. (B-spline is short for "basis spline" but that is something of a misnomer.)

As with basis function approaches based on an infinite sum, any triangular array satisfying equation (4) allows us to approximate f with a finite linear model in which only $\beta_{1m},\ldots,\beta_{mm}$ are unknown. Triangular array approximations can also be used with vector inputs.

Rather than defining a triangular array of indicator functions, we can use the following device to define a single infinite series :

$$\phi_0(x)=1,\quad \phi_1(x)=\mathscr{I}_{(0,.5]}(x),\quad \phi_2(x)=\mathscr{I}_{(.5,1]}(x),$$

$$\phi_3(x)=\mathscr{I}_{(0,.25]}(x),\quad \phi_4(x)=\mathscr{I}_{(.25,.5]}(x),$$
$$\phi_5(x)=\mathscr{I}_{(.5,.75]}(x),\quad \phi_6(x)=\mathscr{I}_{(.75,1]}(x),$$
$$\phi_7(x)=\mathscr{I}_{(0,2^{-3}]}(x),\ldots,\phi_{14}(x)=\mathscr{I}_{(\{2^3-1\}2^{-3},1]}(x),$$
$$\phi_{15}(x)=\mathscr{I}_{(0,2^{-4}]}(x),\ldots .$$

Technically, these ϕ_js constitute a spanning set of functions but are not basis functions. Except for approximating the point $f(0)$, including the function $\phi_0(x)$ is irrelevant once we include $\phi_1(x)$ and $\phi_2(x)$. Similarly, $\phi_1(x)$ and $\phi_2(x)$ are made irrelevant by $\phi_3(x),\ldots,\phi_6(x)$.

A sequence of basis functions, one that is equivalent to this spanning set of step functions, is the *Haar wavelet* collection

$$\phi_0(x)=1,\quad \phi_1(x)=\mathscr{I}_{(0,.5]}(x)-\mathscr{I}_{(.5,1]}(x),$$

$$\phi_2(x)=\mathscr{I}_{(0,.25]}(x)-\mathscr{I}_{(.25,.5]}(x),\quad \phi_3(x)=\mathscr{I}_{(.5,.75]}(x)-\mathscr{I}_{(.75,1]}(x),$$
$$\phi_4(x)=\mathscr{I}_{(0,1/8]}(x)-\mathscr{I}_{(1/8,2/8]}(x),\ldots,\phi_7(x)=\mathscr{I}_{(6/8,7/8]}(x)-\mathscr{I}_{(7/8,1]}(x),$$
$$\phi_8(x)=\mathscr{I}_{(0,1/16]}(x)-\mathscr{I}_{(1/16,2/16]}(x),\ldots .$$

It is customary to call $\phi_0(x)$ the *father* wavelet function and $\phi_1(x)$ the *mother* function. Note that all of the subsequent functions are obtained from the mother function by changing the location and scale, for example, $\phi_3(x)=\phi_1(2x-1)$, $\phi_7(x)=\phi_1(4x-3)$, and, in general, if $j=2^r+k$ for $k=0,1,\ldots,2^r-1$, then $\phi_j(x)=\phi_1(2^r x-k)$.

Actually, *this idea of changing location and scale can be applied to any mother function ϕ_1 that is 0 outside the unit interval and integrates to 0 over the unit interval*, hence generating different families of wavelets to be used as a basis series. (Rather than integrating to 0, theoretical developments often impose a stronger admissability condition on ϕ_1.) For simplicity we restrict ourselves to looking at Haar wavelets but my impression is that they are rarely used in practice. The *Mexican hat (Ricker) wavelet* seems to be quite popular.

7.2 Simple Nonparametric Regression

The simple nonparametric regression model is

$$y_i = f(x_i) + \varepsilon_i, \quad \mathrm{E}(\varepsilon_i) = 0,$$

$i = 1, \ldots, n$, where y_i is a random variable, x_i is a known (scalar) constant, f is an unknown continuous function, and the ε_is are unobservable independent errors with $\mathrm{Var}(\varepsilon_i) = \sigma^2$. Traditionally, the errors are assumed independent, rather than just uncorrelated, to facilitate asymptotic results. In matrix form, write

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

or

$$Y = F(X) + e, \quad \mathrm{E}(e) = 0, \quad \mathrm{Cov}(e) = \sigma^2 I,$$

where $X \equiv (x_1, \ldots, x_n)'$ and $F(X) \equiv [f(x_1), \ldots, f(x_n)]'$. Again, for ease of exposition, we assume that $x_i \in [0,1]$ for all i.

Using the infinite basis representation

$$f(x) = \sum_{j=0}^{\infty} \beta_j \phi_j(x),$$

the nonparametric regression model becomes an infinite linear model,

$$y_i = \sum_{j=0}^{\infty} \beta_j \phi_j(x_i) + \varepsilon_i.$$

This is not useful because it involves an infinite sum, so we use a finite linear model approximation,

$$y_i = \sum_{j=0}^{s-1} \beta_j \phi_j(x_i) + \varepsilon_i. \tag{1}$$

Essentially the same approximation results from a triangular array representation of f. If we define $\Phi_j \equiv [\phi_j(x_1), \ldots, \phi_j(x_n)]'$, in matrix terms model (1) becomes

$$Y = [\Phi_0, \Phi_1, \ldots, \Phi_{s-1}] \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{s-1} \end{bmatrix} + e,$$

or, defining $\Phi \equiv [\Phi_0, \Phi_1, \ldots, \Phi_{s-1}]$, we get

$$Y = \Phi\beta + e.$$

The linear model (1) is only an approximation, so in reality the errors will be biased. For basis functions $\mathrm{E}(\varepsilon_i) = \sum_{j=s}^{\infty} \beta_j \phi_j(x_i)$. It is important to know that for s large, these bias terms are small; see Efromovich (1999, Section 2.2).

Perhaps the two most important statistical questions are how to estimate the β_js and how to choose an appropriate value of s. These issues are addressed in the next two sections.

7.3 Estimation

Choose s so that, for all practical purposes,

$$Y = \Phi\beta + e, \quad \mathrm{E}(e) = 0, \quad \mathrm{Cov}(e) = \sigma^2 I. \tag{1}$$

Clearly, in this model, least squares estimates are BLUEs, so

$$\hat{\beta} = (\Phi'\Phi)^{-1}\Phi' Y.$$

To construct tests or confidence intervals, we would need to assume independent normal errors. The regression function is estimated by

$$\hat{f}(x) = \sum_{j=0}^{s-1} \hat{\beta}_j \phi_j(x).$$

This methodology requires $r(\Phi) \leq n$. Often the model will fit the data perfectly when $s = n$, but this would not occur if the Φ_js are linearly dependent (i.e., if $r(\Phi) < s$). In the next chapter we consider alternatives to least squares estimation.

For the voltage drop data we now examine the use of several methods of nonparametric regression: fitting polynomials, cosines, Haar wavelets, and cubic splines. We begin with the most familiar of these methodologies, fitting polynomials.

7.3.1 Polynomials

Fitting high-order polynomials becomes difficult numerically unless we do something toward orthogonalizing them. We will only fit a sixth degree polynomial, so for the battery data we can get by with simply subtracting the mean before defining the polynomials. The fitted sixth degree regression is

$$\hat{y} = 14.6 + 7.84(x-0.5) - 66.3(x-0.5)^2 - 28.7(x-0.5)^3 + 199(x-0.5)^4 + 10.2(x-0.5)^5 - 92(x-0.5)^6$$

with $R^2 = 0.991$. The regression coefficients, ANOVA table, and sequential sums of squares are:

Table of Coefficients: 6th Degree Polynomial.

Predictor	$\hat{\beta}_k$	SE($\hat{\beta}_k$)	t	P
Constant	14.6156	0.0901	162.24	0.000
$(x-0.5)$	7.8385	0.6107	12.83	0.000
$(x-0.5)^2$	−66.259	4.182	−15.84	0.000
$(x-0.5)^3$	−28.692	9.190	−3.12	0.004
$(x-0.5)^4$	199.03	43.87	4.54	0.000
$(x-0.5)^5$	10.17	30.84	0.33	0.744
$(x-0.5)^6$	−91.6	121.2	−0.76	0.455

Analysis of Variance: 6th Degree Polynomial

Source	df	SS	MS	F	P
Regression	6	259.256	43.209	624.77	0.000
Error	34	2.351	0.069		
Total	40	261.608			

Source	df	Seq. SS
$(x-0.5)$	1	47.081
$(x-0.5)^2$	1	170.159
$(x-0.5)^3$	1	11.155
$(x-0.5)^4$	1	30.815
$(x-0.5)^5$	1	0.008
$(x-0.5)^6$	1	0.039

From the sequential sums of squares, the F test for dropping to a fourth degree polynomial is

$$F = \frac{[0.039 + 0.008]/2}{0.069} < 1,$$

so, refitting, we can get by with the regression equation

$$\hat{y} = 14.6 + 7.67(x-0.5) - 63.4(x-0.5)^2 - 25.7(x-0.5)^3 + 166(x-0.5)^4,$$

which still has $R^2 = 0.991$. The regression coefficients and ANOVA table are

Table of Coefficients: 4th Degree Polynomial.

Predictor	$\hat{\beta}_k$	$\text{SE}(\hat{\beta}_k)$	t	P
Constant	14.5804	192.64	0.0757	0.000
$(x-0.5)$	7.6730	22.47	0.3414	0.000
$(x-0.5)^2$	−63.424	−34.99	1.812	0.000
$(x-0.5)^3$	−25.737	−12.94	1.989	0.000
$(x-0.5)^4$	166.418	21.51	7.738	0.000

Analysis of Variance: 4th Degree Polynomial.

Source	df	SS	MS	F	P
Regression	4	259.209	64.802	972.66	0.000
Error	36	2.398	0.0676		
Total	40	261.608			

Note that the estimated regression coefficients have changed with the dropping of the fifth and sixth degree terms. Figure 7.1 displays the data and the fitted curve.

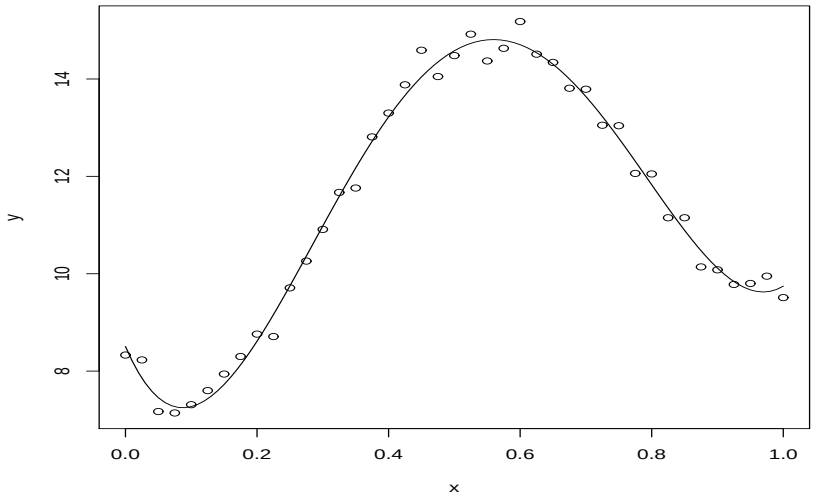

Fig. 7.1 Fourth-degree polynomial fit to battery data.

Polynomials fit these data very well. Other linear approximations may fit the data better or worse. What fits well depends on the particular data being analyzed.

7.3.2 Cosines

For fitting cosines, define the variable $c_j \equiv \cos(\pi jx)$. I arbitrarily decided to fit cosines up to $j = 30$. The fitted regression equation is

$$\begin{aligned}\hat{y} = {} & 11.4 - 1.63c_1 - 3.11c_2 + 0.457c_3 + 0.216c_4 + 0.185c_5 \\ & + 0.150c_6 + 0.0055c_7 + 0.0734c_8 + 0.0726c_9 + 0.141c_{10} \\ & + 0.0077c_{11} + 0.0603c_{12} + 0.125c_{13} + 0.120c_{14} + 0.0413c_{15} \\ & + 0.0184c_{16} + 0.0223c_{17} - 0.0320c_{18} + 0.0823c_{19} + 0.0409c_{20} \\ & - 0.0005c_{21} + 0.0017c_{22} + 0.0908c_{23} + 0.0036c_{24} - 0.0660c_{25} \\ & + 0.0104c_{26} + 0.0592c_{27} - 0.0726c_{28} - 0.0760c_{29} + 0.0134c_{30}\end{aligned}$$

with $R^2 = 0.997$ and ANOVA table

Analysis of Variance: 30 Cosines.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	30	260.7275	8.6909	98.75	0.000
Error	10	0.8801	0.0880		
Total	40	261.6076			

The table of regression coefficients is Table 7.2 and Figure 7.2 displays the data and the fitted model. Note that most of the action in Table 7.2 takes place from $j = 0, \ldots, 6$ with no other terms having P values less than 0.05. However, these all are tests of effects fitted last and are not generally appropriate for deciding on the smallest level of j. In this case, the xs are equally spaced, so the c_js are very nearly uncorrelated, so a model based on $j = 0, \ldots, 6$ will probably work well.

The regression equation based on only $j = 0, \ldots, 6$ is

$$\hat{y} = 11.4 - 1.61c_1 - 3.10c_2 + 0.473c_3 + 0.232c_4 + 0.201c_5 + 0.166c_6$$

with $MSE = 0.094 = 3.195/34$ and $R^2 = 98.8\%$. Notice the slight changes in the regression coefficients relative to the first 7 terms in Table 7.2 due to collinearity. The correlation matrix of c_1 to c_6 is not quite the identity:

	Correlations					
	c_1	c_2	c_3	c_4	c_5	c_6
c_1	1.00	0.00	0.05	0.00	0.05	0.00
c_2	0.00	1.00	0.00	0.05	0.00	0.05
c_3	0.05	0.00	1.00	0.00	0.05	0.00
c_4	0.00	0.05	0.00	1.00	0.00	0.05
c_5	0.05	0.00	0.05	0.00	0.00	0.00
c_6	0.00	0.05	0.00	0.05	0.00	1.00,

although the real issue is the correlations between these 6 and the other 24 variables. Figure 7.3 displays the data along with the fitted cosine curve for $s - 1 = 6$. Both the

Table 7.2 Regression coefficients for fitting cosines with $s-1=30$.

Table of Coefficients									
j	$\hat{\beta}_k$	SE	t	P	j	$\hat{\beta}_k$	SE	t	P
0	11.3802	0.0466	244.34	0.000	16	0.01844	0.06539	0.28	0.784
1	−1.62549	0.06538	−24.86	0.000	17	0.02225	0.06538	0.34	0.741
2	−3.11216	0.06539	−47.59	0.000	18	−0.03197	0.06539	−0.49	0.635
3	0.45701	0.06538	6.99	0.000	19	0.08235	0.06538	1.26	0.236
4	0.21605	0.06539	3.30	0.008	20	0.04087	0.06539	0.62	0.546
5	0.18491	0.06538	2.83	0.018	21	−0.00048	0.06538	−0.01	0.994
6	0.14984	0.06539	2.29	0.045	22	0.00165	0.06539	0.03	0.980
7	0.00553	0.06538	0.08	0.934	23	0.09076	0.06538	1.39	0.195
8	0.07343	0.06539	1.12	0.288	24	0.00356	0.06539	0.05	0.958
9	0.07262	0.06538	1.11	0.293	25	−0.06597	0.06538	−1.01	0.337
10	0.14136	0.06539	2.16	0.056	26	0.01038	0.06539	0.16	0.877
11	0.00765	0.06538	0.12	0.909	27	0.05924	0.06538	0.91	0.386
12	0.06032	0.06539	0.92	0.378	28	−0.07257	0.06539	−1.11	0.293
13	0.12514	0.06538	1.91	0.085	29	−0.07600	0.06538	−1.16	0.272
14	0.11983	0.06539	1.83	0.097	30	0.01338	0.06539	0.20	0.842
15	0.04128	0.06538	0.63	0.542					

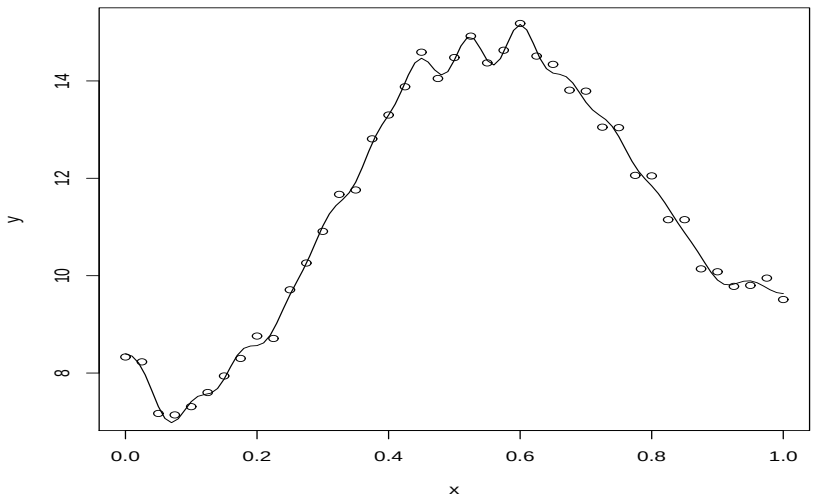

Fig. 7.2 Cosine fit with $s-1=30$ for the battery data.

figures and the R^2 values establish that fitting the 5 parameters in a 4th degree polynomial fits these data better than an intercept and 6 cosine terms. That a polynomial fits better than cosines is a peculiarity of these data.

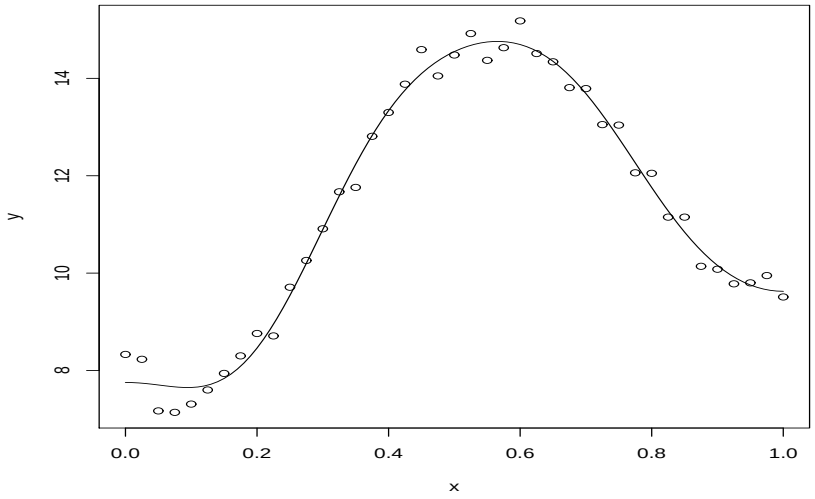

Fig. 7.3 Cosine fit with $s-1=6$ for the battery data.

7.3.3 Haar wavelets

Consider fitting Haar wavelets. We fit 32 functions, the father wavelet $\phi_0(x) \equiv p_0(x) \equiv 1$, the mother wavelet $\phi_1(x) \equiv m_0$, and then transformations of the mother wavelet, $\phi_j(x) \equiv m_{rk}$ where $j = 2^r + k - 1$, $k = 1, \ldots, 2^r$. The fitted regression equation is

$$\begin{aligned}\hat{y} = {} & 11.3 - 1.05m_0 - 2.31m_{11} + 1.78m_{12} \\ & - 0.527m_{21} - 1.36m_{22} + 0.472m_{23} + 0.814m_{24} \\ & + 0.190m_{31} - 0.444m_{32} - 0.708m_{33} - 0.430m_{34} \\ & - 0.058m_{35} + 0.317m_{36} + 0.567m_{37} + 0.071m_{38} \\ & + 0.530m_{4,1} - 0.181m_{4,2} - 0.180m_{4,3} - 0.248m_{4,4} \\ & - 0.325m_{4,5} - 0.331m_{4,6} - 0.290m_{4,7} + 0.139m_{4,8} \\ & + 0.275m_{4,9} - 0.131m_{4,10} + 0.265m_{4,11} + 0.349m_{4,12} \\ & + 0.005m_{4,13} + 0.229m_{4,14} + 0.150m_{4,15} + 0.012m_{4,16}\end{aligned}$$

with $R^2 = 0.957$ and ANOVA table

Analysis of Variance: 32 Haar Wavelets.

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	31	250.489	8.080	6.54	0.003
Error	9	11.118	1.235		
Total	40	261.6076			

Based on R^2, this fits the data much worse than either the fourth degree polynomial regression or the cosine regression model with $s-1=6$. Table 7.3 gives the estimated regression coefficients. This table gives little indication that either the third- or fourth-order wavelets are contributing to the fit of the model, but again, the predictor variables are not uncorrelated, so definite conclusions cannot be reached. For example, the $m_{4,k}$s are defined so that they are uncorrelated with each other, but they are not uncorrelated with all of the m_{3k}s. In particular, $m_{4,2}$ is not uncorrelated with m_{31}. To see this, note that the first six entries of the 41 dimensional vector $m_{4,2}$ are $(0,0,0,1,-1,-1)$, with the rest being 0's and the first six entries of m_{31} are $(0,1,1,-1,-1,-1)$, with the rest being 0's. Clearly, the two vectors are not orthogonal. The problem is that, even though the observations are equally spaced, the wavelets are based on powers of $1/2$, whereas the 41 observations occur at intervals of $1/40$.

Table 7.3 Regression coefficients for fitting 32 Haar wavelets.

Table of Coefficients									
Var.	$\hat{\beta}_k$	SE	*t*	*P*	Var.	$\hat{\beta}_k$	SE	*t*	*P*
p_0	11.3064	0.1813	62.35	0.000	$m_{4,1}$	0.5300	0.7859	0.67	0.517
m_0	−1.0525	0.1838	−5.73	0.000	$m_{4,2}$	−0.1808	0.6806	−0.27	0.797
m_{11}	−2.3097	0.2599	−8.89	0.000	$m_{4,3}$	−0.1800	0.7859	−0.23	0.824
m_{12}	1.7784	0.2599	6.84	0.000	$m_{4,4}$	−0.2483	0.6806	−0.36	0.724
m_{21}	−0.5269	0.3676	−1.43	0.186	$m_{4,5}$	−0.3250	0.7859	−0.41	0.689
m_{22}	−1.3637	0.3676	−3.71	0.005	$m_{4,6}$	−0.3308	0.6806	−0.49	0.639
m_{23}	0.4725	0.3676	1.29	0.231	$m_{4,7}$	−0.2900	0.7859	−0.37	0.721
m_{24}	0.8144	0.3676	2.22	0.054	$m_{4,8}$	0.1392	0.6806	0.20	0.842
m_{31}	0.1896	0.5198	0.36	0.724	$m_{4,9}$	0.2750	0.7859	0.35	0.734
m_{32}	−0.4441	0.5198	−0.85	0.415	$m_{4,10}$	−0.1308	0.6806	−0.19	0.852
m_{33}	−0.7079	0.5198	−1.36	0.206	$m_{4,11}$	0.2650	0.7859	0.34	0.744
m_{34}	−0.4304	0.5198	−0.83	0.429	$m_{4,12}$	0.3492	0.6806	0.51	0.620
m_{35}	−0.0579	0.5198	−0.11	0.914	$m_{4,13}$	0.0050	0.7859	0.01	0.995
m_{36}	0.3171	0.5198	0.61	0.557	$m_{4,14}$	0.2292	0.6806	0.34	0.744
m_{37}	0.5671	0.5198	1.09	0.304	$m_{4,15}$	0.1500	0.7859	0.19	0.853
m_{38}	0.0709	0.5198	0.14	0.895	$m_{4,16}$	0.0117	0.6806	0.02	0.987

Figure 7.4 displays the data along with the fitted Haar wavelets for $s = 32$. The figure displays a worse fit than the 4th degree polynomial and the 7 parameter cosine model despite fitting many more parameters. This is consistent with the relatively poor Haar wavelet R^2. Some kind of curved mother wavelet function would probably

fit better than the Haar wavelets. Notice the curious behavior of the plot at $x = 0$. The way the Haar wavelets have been defined here, if the columns of Φ are uncorrelated, the estimate of $f(0)$ will always be $\bar{y}$. Here, the columns of Φ are not orthogonal, but they are not ridiculously far from orthogonality, so the estimate of $f(0)$ is close to $\bar{y}$. If we had standardized the data in Table 7.1 so that $x_1 \neq 0$, this would not have been a problem. In particular, if we had defined $x_i = 2(t_i + .5)/42 = i/(n+1)$, we would not have $x_1 = 0$.

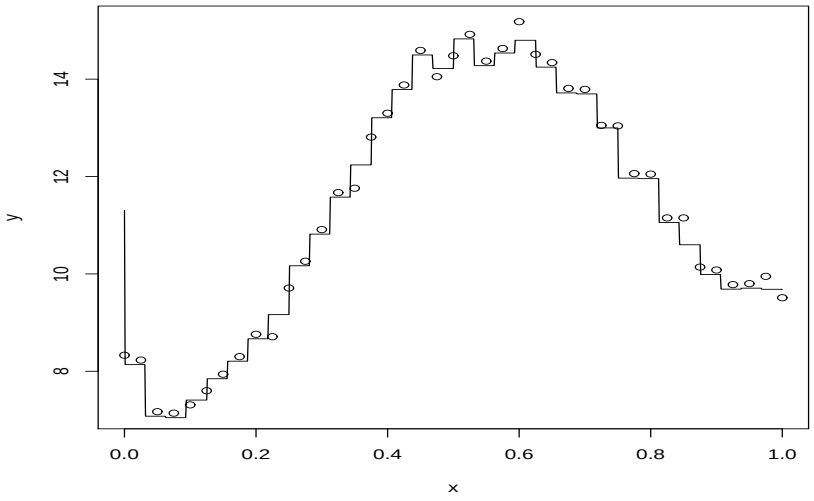

Fig. 7.4 Haar wavelet fit with $s = 32$ for the battery data.

7.3.4 Cubic splines

Splines are discussed in detail in *ALM-III*, Section 1.6. Splines are fundamentally related to fitting a collection of polynomials to disjoint subsets of the data, polynomials that are forced to connect nicely at a collections of knots where the subsets meet. They can also be related to things called *b-splines* that are reminiscent of fitting wavelets. But it turns out that a general cubic spline model with $m-1$ interior knots $\tilde{x}_j$ can be written

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x^3 + \sum_{j=1}^{m-1} \beta_{j+3}[(x_i - \tilde{x}_j)_+]^3 + \varepsilon_i$$

where for any scalar a

$$(x-a)_+ \equiv \begin{cases} x-a & \text{if } x > a \\ 0 & \text{if } x \leq a. \end{cases}$$

We began by fitting 30 equally spaced interior knots, to get

Analysis of Variance: Splines with 30 knots

Source	*df*	*SS*	*MS*	*F*	*P*
Regression	33	261.064	7.9110	101.8	0.0000
Error	7	0.544	0.0777		
Total	40	261.6076			

with $R^2 = 0.998$ and regression coefficients in Table 7.4. The fitted curve is displayed in Figure 7.5. Most of the regression coefficients are quite large because they are multiplying numbers that are quite small, i.e., the cube of a number in [0,1]. There is little useful information in the table of coefficients for the purpose of picking a smaller number of knots. We also fitted a model with only 4 equally spaced knots that fits surprisingly well, having $R^2 = 0.991$, see Figure 7.6

Table 7.4 Regression coefficients for fitting splines with 30 interior knots.

Table of Coefficients									
Var.	$\hat{\beta}_k$	SE	t	P	Var.	$\hat{\beta}_k$	SE	t	P
Const.	8.330	0.2788	29.881	0.0000	ϕ_{17}	78360	34500	2.271	0.0574
x	123.6	12.84	0.962	0.3680	ϕ_{18}	−91100	34620	−2.631	0.0338
x^2	−7530	7422	−1.015	0.3441	ϕ_{19}	85420	34620	2.467	0.0430
x^3	97080	103000	0.943	0.3773	ϕ_{20}	−62290	34500	−1.806	0.1140
ϕ_4	−116300	142700	−0.815	0.4419	ϕ_{21}	25890	34570	0.749	0.4783
ϕ_5	21860	64770	0.338	0.7456	ϕ_{22}	5164	34670	0.149	0.8858
ϕ_6	−7743	43780	−0.177	0.8646	ϕ_{23}	−13920	34560	−0.403	0.6991
ϕ_7	12430	37310	0.333	0.7487	ϕ_{24}	10190	34520	0.295	0.7765
ϕ_8	−21280	35460	−0.600	0.5674	ϕ_{25}	−9532	34640	−0.275	0.7911
ϕ_9	36210	34810	1.040	0.3329	ϕ_{26}	11840	34630	0.342	0.7425
ϕ_{10}	−45710	34560	−1.323	0.2275	ϕ_{27}	−9615	34560	−0.278	0.7889
ϕ_{11}	42350	34630	1.223	0.2608	ϕ_{28}	1079	34810	0.031	0.9761
ϕ_{12}	−39120	34640	−1.129	0.2960	ϕ_{29}	8318	35460	0.235	0.8213
ϕ_{13}	45090	34520	1.306	0.2328	ϕ_{30}	−10490	37310	−0.281	0.7868
ϕ_{14}	−51080	34560	−1.478	0.1829	ϕ_{31}	8146	43780	0.186	0.8577
ϕ_{15}	51290	34670	1.479	0.1826	ϕ_{32}	−8246	64770	−0.127	0.9023
ϕ_{16}	−58680	34570	−1.697	0.1334	ϕ_{33}	−10490	142700	−0.074	0.9434

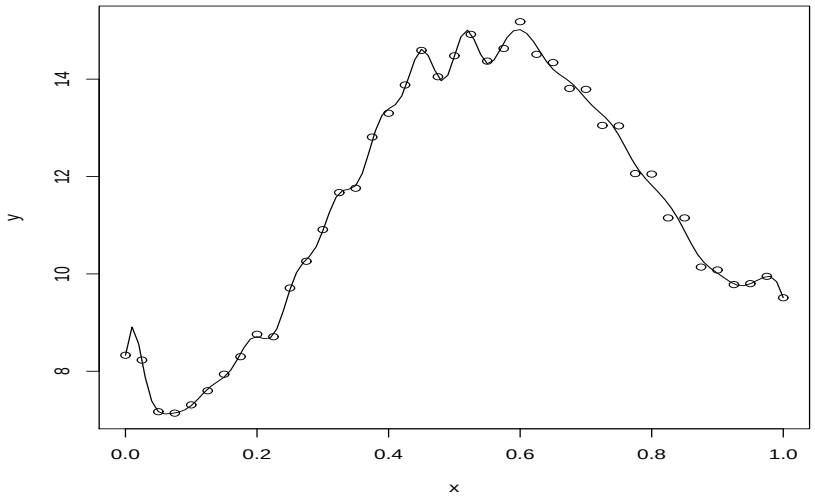

Fig. 7.5 Cubic spline fit with 30 interior knots for the battery data.

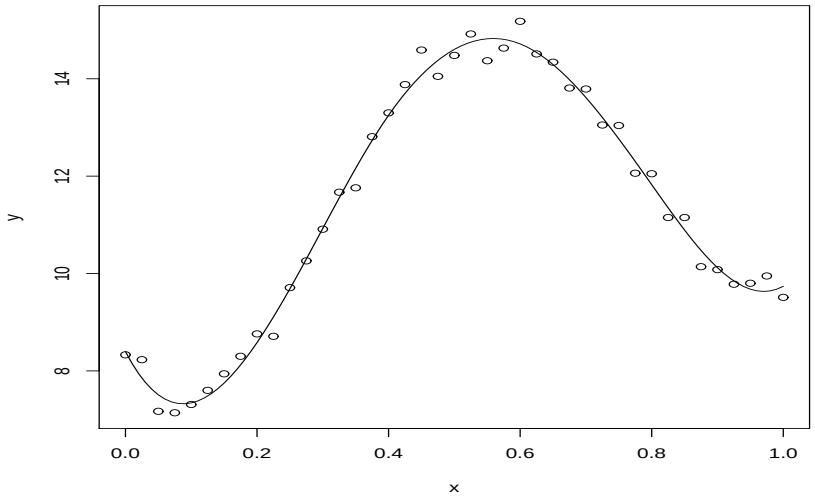

Fig. 7.6 Cubic spline fit with 4 interior knots for the battery data.

7.4 Variable Selection

Variable selection is of key importance in these problems because the linear model is only an approximation. The problem is to select an appropriate value of s in model (7.2.1).

For fitting polynomials, or cosines, or any ϕ_js with an obvious ordering, the situation is analogous to identifying the important features in a 2^n factorial design in that we could take $s = n$ and construct a $\chi^2(1)$ plot from the sequential sums of squares to identify the largest important terms; see Christensen (1996, Sections 17.3 and 17.4) or http://www.stat.unm.edu/~fletcher/TopicsInDesign. If the $\frac{1}{\sqrt{n}}\Phi_j$ vectors are orthonormal, the analogy to a factorial is even closer and we could construct a normal or half-normal plot of the $\hat{\beta}_j$s to identify any important ϕ_j functions. For sines and cosines the *pairs* of variables are ordered, so $\chi^2(2)$ plots are more appropriate. Such a plot is closely related to the periodogram, cf. the *ALM* Frequency Domain chapter. For the case against using such graphical methods, see Lenth (2015).

Another method of choosing s is by cross-validation. For example, one can minimize the PRESS statistic; see *PA-V* Section 12.5 (Christensen, 2011, Section 13.5) or Hart (1997, Section 4.2.1).

An alternative to variable selection is using a penalized estimation procedure as discussed in the next chapter.

In the remainder of this section we will assume that we have fitted a model with s predictors where s was chosen to be so large that it clearly gives a reasonable approximation. We want to find a reduced model with p predictors that does not over fit the data.

Hart (1997, Section 4.2.2) and Efromovich (1999, p. 125) implicity suggest selecting p to minimize C_p on the sequence of models defined using the first p cosine ϕs for $p = 1, \ldots, s$. (Actually they suggest maximizing a statistic A_p that is equivalent to minimizing C_p.) The same idea applies whenever the ϕ_js are ordered, e.g., polynomials. When using sines and cosines, by pairing terms with equal frequencies, you can cut the length of the sequence in half. Wavelets have a nesting structure that determines a sequence of models. For splines you would have to determine a sequence of models determined by adding additional knots, e.g., you could start with 1 knot, add two more, add four more, etc. much like wavelets.

Selecting p by minimizing C_p on a sequence of models does not allow dropping lower-order terms if higher ones are included (i.e., it is similar, in polynomial regression, to not allowing x^2 to be eliminated if x^3 remains in the model). For cosines Efromovich suggested picking $s = 6\tilde{p}$, where $\tilde{p}$ is the smallest value of p for which

$$MSE < 2[1.48\,\text{median}|y_i - \hat{y}_i|]^2.$$

Based on Hart's discussion of Hurvich and Tsai (1995), another crude upper bound might be $s = \sqrt{n}$, although in practice this seems to give too small values of s. Subsection 5.4.6 suggests that $n - s$ should be at least 8 with values of 18 or more preferable. In Example 7.4.1, s is chosen by the seat of my pants.

EXAMPLE 7.4.1. Using the battery data and fitting cosines with $s-1=30$ gives $MSE=0.0880$ on $dfE=10$. Table 7.5 gives sequential sums of squares and values of $-C_p(j+1)+C_p(1)$. Here we denote the C_p statistic based on r parameters $C_p(r)$. Because $C_p(r)=(s-r)(F-2)+s$ where F is the statistic for comparing the r parameter and s parameter models, the $-C_p(j+1)+C_p(1)$s are easily computed from the sequential sums of squares as partial sums of the (F_k-2) statistics where

$$F_k \equiv \frac{SSR(\Phi_k|\Phi_0,\ldots,\Phi_{k-1})}{MSE}.$$

For example, $F_5=0.8427/0.0880$ and $-C_p(6)+C_p(1)=(F_1-2)+\cdots+(F_5-2)$. The C_p statistic is minimized when $-C_p(r)+C_p(1)$ is maximized, so the best models from the sequence have $p-1=6,10,13,14$. If one were willing to consider models that do not include a contiguous set of j values, the problem becomes a traditional variable selection problem. Given the near orthogonality of the predictors in this example, it is fairly obvious from the sequential sums of squares alone that the most important predictors are $j=1,\ldots,6,10,13,14$. With more collinear data, such a conclusion could not be made from the sequential sums of squares.

Table 7.5 Selection of s based on the C_p statistic.

j	Seq SS	$-C_p(j+1)+C_p(1)$	j	Seq SS	$-C_p(j+1)+C_p(1)$
1	52.2633	591.90	16	0.0061	2922.20
2	198.6634	2847.44	17	0.0133	2920.35
3	4.8674	2900.75	18	0.0213	2918.59
4	1.2009	2912.40	19	0.1412	2918.19
5	0.8427	2919.97	20	0.0322	2916.56
6	0.5753	2924.51	21	0.0000	2914.56
7	0.0088	2922.61	22	0.0000	2912.56
8	0.1538	2922.36	23	0.1605	2912.38
9	0.1472	2922.03	24	0.0001	2910.39
10	0.4547	2925.20	25	0.0911	2909.42
11	0.0070	2923.28	26	0.0015	2907.44
12	0.0857	2922.25	27	0.0669	2906.20
13	0.3554	2924.29	28	0.1073	2905.42
14	0.2951	2925.64	29	0.1189	2904.77
15	0.0425	2924.13	30	0.0037	2902.81

Figure 7.7 gives fitted cosine curves for $s-1=6,10,14,30$. I suspect that, visually, $s-1=14$ in the bottom left is the one that would most appeal to practitioners of nonparametric regression. □

EXAMPLE 7.4.2. For fitting the Haar wavelets to the battery data, we have obvious groups of variables that occur in powers of 2. We can consider the highest-order group that we need, or we could consider including individual terms from any order group. In the first case, we would consider tests based on the ANOVA tables reported in Table 7.6.

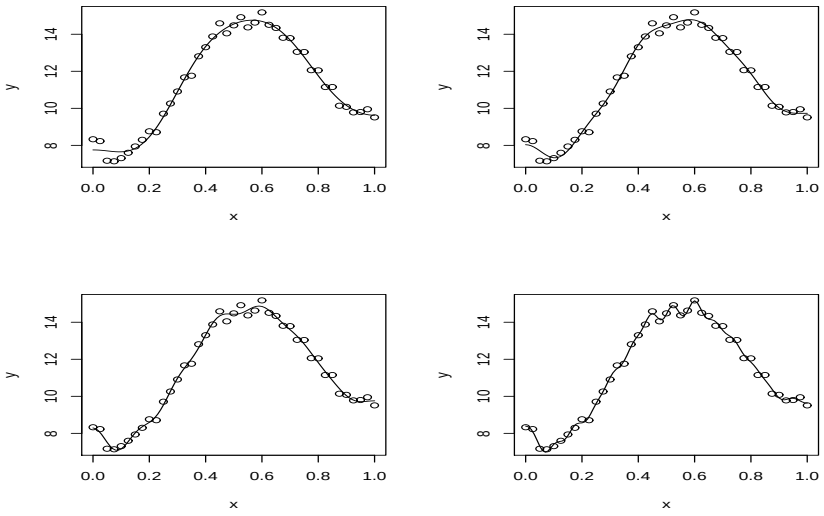

Fig. 7.7 Cosine fit with $s-1=6,10,14,30$ for the battery data. Read across and down.

Table 7.6 ANOVA tables for Haar wavelets.

Analysis of Variance: Fitting p_0 to $m_{4,16}$.					
Source	*df*	*SS*	*MS*	*F*	*P*
Regression	31	250.489	8.080	6.54	0.003
Error	9	11.118	1.235		
Total	40	261.6076			
Analysis of Variance: Fitting p_0 to $m_{3,8}$.					
Source	*df*	*SS*	*MS*	*F*	*P*
Regression	15	248.040	16.536	30.47	0.000
Residual Error	25	13.568	0.543		
Total	40	261.608			
Analysis of Variance: Fitting p_0 to $m_{2,4}$.					
Source	*df*	*SS*	*MS*	*F*	*P*
Regression	7	240.705	34.386	54.29	0.000
Residual Error	33	20.902	0.633		
Total	40	261.608			

To test whether we can drop the $m_{4,k}$s, the test statistic is

$$F = \frac{[13.568 - 11.118]/16}{1.235} < 1.$$

To test whether we can drop the m_{3k}s, the test statistic is

$$F = \frac{[20.902 - 13.568]/8}{0.543} \doteq 2$$

or, using the *MSE* from the largest model fitted,

$$F = \frac{[20.902 - 13.568]/8}{1.235} < 1$$

If we allow elimination of individual variables from any group, the problem becomes a traditional variable selection problem. The number of wavelets needed is related to the smoothness of f, and the smoothness can change on different subsets of [0,1]. Figure 7.8 gives the fitted Haar wavelet curves for $s = 8$ and $s = 16$. Relative to the $s = 16$ fit, the $s = 8$ wavelets work pretty well from 0.5 to 0.625 and also from 0.875 to 1 but not very well anywhere else. □

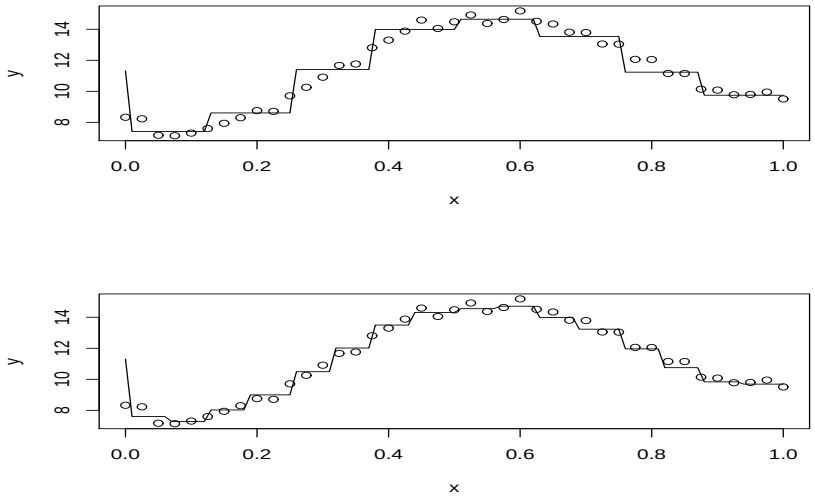

Fig. 7.8 Haar wavelet fit with $s = 8, 16$ for the battery data.

7.5 Approximating-Functions with Small Support

The *support* of a function is where it takes nonzero values. Except for a finite number of 0s, the support of the polynomials, sines, and cosines is the entire interval [0,1]. As seen in Chapter 3, this can cause strange behavior when s gets close to n, especially when the x_is are unevenly spaced. As regards polynomials, this strange behavior is a well-known fact, but it is, perhaps, less well known as regards sines and cosines. The fundamental idea of polynomial splines, b-splines, and wavelets is to fit functions that are nonzero only on small subsets of the domain space [0,1].

As discussed in Section 1, splines and b-splines both involve knots. Earlier we needed a subscript m on the knots to indicate how many knots were being used. In this section, we will drop m from the subscript when m is not subject to change.

7.5.1 Polynomial Splines

The basic idea behind using polynomial splines is to connect the dots. Suppose we have simple regression data (x_i, y_i), $i = 1, \ldots, n$ in which the x_is are ordered from smallest to largest. Linear splines quite simply give, as a regression function, the function that fits a line segment between the consecutive pairs of points. Cubic splines fit a cubic polynomial between every pair of points rather than a line. Note that all of the action here has nothing to do with fitting a model to the data. The data are being fitted perfectly (at least in this simplest form of spline fitting). The key issue is how to model what goes on between data points.

In practice, splines often do not actually connect the dots, they create smooth functions between knots. Section 3.6 illustrates the use of two linear splines, whereas here we have focused on several cubic splines. The reason for using cubic splines is to make the curve look smooth. With cubic splines we require the fitted regression function to have continuous second derivatives.

Fitting polynomials between knots with continuity and continuous derivative constraints at each interior knot is equivalent to the kind of model we fitted for cubic splines in Section 3 and to the b-spline method that we are about to formally introduce, cf. *ALM-III*, Subsection 1.6.1. For fitting dth order polynomial splines with $m-1$ interior knots $\tilde{x}_j$, it suffices to fit

$$y_i = \sum_{k=0}^{d} \beta_k x^k + \sum_{j=1}^{m-1} \beta_{j+d} [(x_i - \tilde{x}_j)_+]^d + \varepsilon_i,$$

which involves $m+d$ parameters. For equally spaced knots, the equivalent b-spline model is

$$y_i = \sum_{j=0}^{m+d-1} \gamma_j \phi_j(x_i) + \varepsilon_i$$

where a dth order mother spline Ψ_d is transformed into

$$\phi_j(x) = \Psi_d(mx - j + d). \tag{1}$$

7.5.1.1 B-splines

B-splines provide the same fit as regular splines but do so by defining a particular mother spline and then defining ϕ_j functions by rescaling and relocating the mother. B-splines are supposed to be *basis splines* but they do not actually define a meaningful basis in the space of functions. To get good approximations the number of interior knots $m-1$ must get large but the $\phi_j(x)$ functions all change with m. For fixed m the $\phi_j(x) \equiv \phi_{jm}(x)$ functions define a basis for their spanning space but not for an interesting function space.

The mother spline is itself a low order polynomial spline. The mother spline of degree 2 is nonzero over $(0,3)$ and defined as

$$\Psi_2(x) = \frac{x^2}{2}\mathscr{I}_{[0,1]}(x) - \left\{[x-1.5]^2 - 0.75\right\}\mathscr{I}_{(1,2]}(x) + \frac{[3-x]^2}{2}\mathscr{I}_{(2,3]}(x).$$

This is a bell-shaped curve, similar to a normal density centered at 1.5, but it is 0 outside the interval $[0,3]$ while still being smooth in that it is differentiable everywhere. Ψ_2 is itself a quadratic spline, i.e., quadratics have been pasted together as a smooth function.

A mother spline Ψ_d of degree d has support on the interval $(0,d+1)$. It splices together $(d+1)$ different d-degree polynomials, each defined on a length 1 interval, so that the whole function is differentiable $d-1$ times and looks like a mean-shifted Gaussian density. Commonly d is either 2 or 3. For $d=3$, the cubic mother spline on [0,4] is

$$\begin{aligned}\Psi_3(x) = {} & \frac{x^3}{3}\mathscr{I}_{[0,1]}(x) + \left\{-x^3 + 4x^2 - 4x + \frac{4}{3}\right\}\mathscr{I}_{(1,2]}(x) \\ & + \left\{-[4-x]^3 + 4[4-x]^2 - 4[4-x] + \frac{4}{3}\right\}\mathscr{I}_{(2,3]}(x) + \frac{[4-x]^3}{3}\mathscr{I}_{(3,4]}(x).\end{aligned}$$

Figure 7.9 shows these b-spline mother functions. Other than the domain on which they are defined, they look quite unremarkable. There is a body of theory associated with b-splines that includes defining the $d+1$ order mother spline recursively from the d order mother.

The approximating functions $\phi_j(x)$ are defined by rescaling and relocating the mother splines. For simplicity, consider $d=2$ with $m-1$ equally spaced interior knots. If the knots are equally spaced, the same rescaling of Ψ_2 works for all ϕ_j. Ψ_2 is defined on [0,3] and pastes together 3 polynomials on three intervals of length one. To define ϕ_0 we rescale Ψ_2 to live on $[0,3/m]$ and then shift it to the left $2/m$ units so that only the polynomial originally defined on [2,3] now overlaps the interval $[0,1/m]$ and ϕ_0 is 0 elsewhere in [0,1]. To define ϕ_1, again rescale Ψ_2 to live on $[0,3/m]$ but now shift it to the left only $1/m$ units so that the polynomial originally

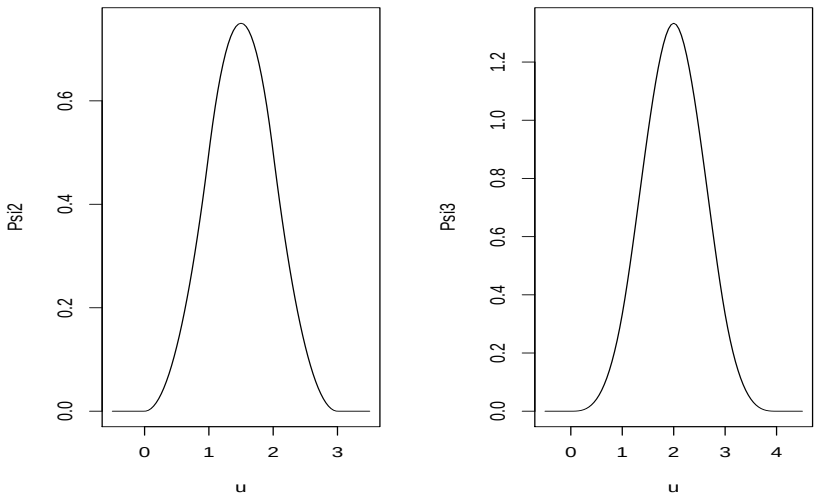

Fig. 7.9 B-spline mother functions for $d = 2,3$.

defined on [2,3] now overlaps the interval $[1/m, 2/m]$ and the polynomial originally defined on [1,2] now overlaps the interval $[0/m, 1/m]$. ϕ_1 is 0 elsewhere in [0,1]. ϕ_2 is just the rescaled version of Ψ_2. ϕ_3 is the rescaled Ψ_2 shifted to the *right* by $1/m$. More generally, ϕ_{2+j} is the rescaled Ψ_2 shifted to the *right* by j/m.

For arbitrary d, Ψ_d is rescaled so that its support is $(0, \{d+1\}/m)$ and ϕ_0 is the rescaled Ψ_d shifted to the left d/m units. Each successive ϕ_j is shifted to the right by an additional $1/m$ so that ϕ_d is the rescaled version of Ψ_d and ϕ_{d+j} is the rescaled Ψ_d shifted to the right by j/m. The general formula for this was given in equation (1). A proof that fitting b-splines is equivalent to fitting regular splines is given in *ALM-III*, Subsection 1.6.1.

7.5.2 Fitting local functions

In discussing b-splines for equally spaced knots we carefully defined $\phi_j \equiv \phi_{jm}$ functions so that they were equivalent to fitting polynomial splines. But in another sense, fitting b-splines is just fitting a bunch of bell shaped curves that were rescaled to have small support and shifted so that the functions had centers that were spread over the entire unit interval.

Just about any function Ψ that goes to 0 as $|x| \to \infty$ can be used as a "mother" to define a triangular array ϕ_{jm} of approximating-functions with small (practical) support. These include indicator functions, mother splines, mother wavelets, normal

densities, etc. Given a set of knots $\tilde{x}_{j,m}$, take $s-1=m$ with ϕ_{jm} a rescaled mother function with a location tied to (often centered at) $\tilde{x}_{j,m}$. The success of this enterprize will depend on the number and placement of the knots and how the mother function is rescaled. The process becomes a method based on approximating functions with small support when the mother function is rescaled in such a way that it becomes, for all practical purposes, 0 outside of a small interval. For example, normal densities have a support that is the entire real line, but if the variance is small enough the density is *practically* zero except for a small region around the mean.

7.5.3 Local Regression

Local (polynomial) regression, often referred to as *loess* or *lowess* (*lo*cal *we*ighted *s*catterplot *s*moothing) provides fitted values by fitting a separate low order polynomial for every prediction. It provides a collection of $(x,\hat{y})$ values that can be plotted, but it does not provide a formula for the estimated regression curve. As with splines, we assume that in the data (x_i,y_i), the x_is are ordered.

The key to local regression is that it uses weighted regression with weights determined by the distance between the actual data x_i and the location being fitted x. What makes this "local" regression is that the weights are either zero, or close to zero, outside a small region around x. The weights are determined by a *kernel* function (not to be confused with the reproducing kernels introduced later in Subsection 6.2).

Originally, this procedure was performed using 0 order polynomials and is known as *kernel smoothing*, see Green and Silverman (1994) or Efromovich (1999). The idea of kernel smoothing is to base estimation on the continuity of $f(x)$. The estimate $\hat{f}(x)$ is a weighted average of the y_i values in a small neighborhood of x. Less weight is given to a y_i for which the corresponding x_i is far from x. The weights are defined by a nonnegative kernel function $K(z)$ that gets small rapidly as z gets away from 0. The *Nadaraya–Watson kernel estimate* is

$$\hat{f}(x)=\sum_{i=1}^{n} y_i K[(x-x_i)] \Big/ \sum_{i=1}^{n} K[(x-x_i)],$$

which is just a weighted average, as advertised.

More generally, take a low order polynomial model, say,

$$Y=X\beta+0, \quad \mathrm{E}(e)=0$$

on which we perform weighted least squares with a diagonal matrix $D(w)$ having some vector of weights w. The definition of the weights is all-important. The ith element of w is

$$w_i \equiv K\left[(x_i-x)/h\right]$$

for some scalar tuning parameter h and kernel function K. From fitting this model we obtain only one thing, the fitted value $\hat{y}$ for the new data point x. You do this for

a lot of xs and plot the result. Obviously, fitting a separate linear model for every fitted value requires modern computing power.

In loess the most commonly used weighting seems to be the *tri-weight* where the kernel function is

$$K(z) = \begin{cases} (1-|z|^3)^3 & \text{if } |z| < 1 \\ 0 & \text{if } |z| \geq 1. \end{cases}$$

In R the default is to fit a quadratic polynomial.

For the battery data the default `loess` fit in R seems to me to oversmooth the data. It gives $R^2 = 0.962$.

7.6 Nonparametric Multiple Regression

Nonparametric multiple regression involves using a p vector x as the argument for $\phi_j(\cdot)$ in an infinite sum or $\phi_{jm}(\cdot)$ in a triangular array. The difficulty is in choosing which ϕ functions to use. There are two common approaches. One is to construct the vector functions ϕ from the scalar ϕ functions already discussed. The other method uses the *kernel trick* to replace explicit consideration of the ϕ functions with evaluation of a *reproducing kernel* function.

7.6.1 Redefining ϕ and the Curse of Dimensionality

In nonparametric multiple regression, the scalars x_i are replaced by vectors $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})'$. Theoretically, the only real complication is that the ϕ_j functions have to be redefined as functions of vectors rather than scalars.

In practice, we often construct vector ϕ functions from scalar ϕ functions. The ideas become clear in the case of $p = 2$. For variables x_1 and x_2, define

$$\phi_{jk}(x_1, x_2) \equiv \phi_j(x_1)\phi_k(x_2),$$

and the regression function approximation is

$$f(x_1, x_2) \doteq \sum_{j=0}^{s_1-1} \sum_{k=0}^{s_2-1} \beta_{jk}\phi_{jk}(x_1, x_2). \tag{1}$$

In general, for $x = (x_1, \ldots, x_p)'$,

$$f(x) \doteq \sum_{k_1=0}^{s_1-1} \cdots \sum_{k_p=0}^{s_p-1} \beta_{k_1 \ldots k_p}\phi_{k_1}(x_1) \cdots \phi_{k_p}(x_p), \tag{2}$$

where most often $\phi_0 \equiv 1$. One practical issue with fitting

$$\phi_{k_1 \ldots k_p}(x) \equiv \phi_{k_1}(x_1) \cdots \phi_{k_p}(x_p)$$

functions when using component functions $\phi_{k_j}(x_j)$ having small support is that when evaluated at the data vectors x_i, you may often get $\phi_{k_1 \ldots k_p}(x_i) = 0$, $i = 1, \ldots, n$, so many of the predictors may be extraneous.

There are a lot of ϕ functions involved in this process! For example, if we needed $s_1 = 10$ functions to approximate a function in x_1 and $s_2 = 8$ functions to approximate a function in x_2, it takes 80 functions to approximate a function in (x_1, x_2), and this is a very simple case. It is not uncommon to have $p = 5$ or more. If we need $s_* = 8$ for each dimension, we are talking about fitting $s = 8^5 = 32,768$ parameters for a very moderately sized problem. Clearly, this approach to nonparametric multiple regression is only practical for very large data sets if $p > 2$. However, nonparametric multiple regression seems to be a reasonable approach for $p = 2$ with moderately large amounts of data, such as are often found in problems such as two-dimensional image reconstruction and smoothing two-dimensional spatial data. Another way to think of the dimensionality problems is that, roughly, if we need n observations to do a good job of estimation with one predictor, we might expect to need n^2 observations to do a good job with two predictors and n^p observations to do a good job with p predictors. For example, if we needed 40 observations to get a good fit in one dimension, and we have $p = 5$ predictors, we need about 100 million observations. (An intercept can be included as either $\phi_0 \equiv 1$ or $x_{i1} \equiv 1$. In the latter case, s_*^{p-1} or n^{p-1} would be more appropriate.) This *curse of dimensionality* can easily make it impractical to fit nonparametric regression models.

One way to deal with having too many parameters is to use *generalized additive models*. Sections 3.8 and 3.9 contain some additional details about writing out generalized additive models but the fundamental idea is an analogy to multifactor analysis of variance, cf. Christensen (1996 or 2015). Fitting the full model with $p = 5$ and, say, $s = 8^5$ parameters is analogous to fitting a 5 factor interaction term. If we fit the model with only the 10 three-factor interaction terms, we could get by with $10(8^3) = 5120$ parameters. If we fit the model with only the 10 two-factor interaction terms, we could get by with $10(8^2) = 640$ parameters. In particular, with the two-factor interactions, $f(x_1, \ldots, x_5)$ is modeled as the sum of 10 terms each looking like equation (1) but with each involving a different pair of predictor variables.

The all three-factor and all two-factor models still seem like a lot of parameters but the decrease is enormous compared to the five-factor model. The price for this decrease is the simplifying assumptions being made. And if we cannot fit the 5-factor interaction model, we cannot test the validity of those simplifying assumptions, e.g., whether it is alright to drop, say, all of the 4-factor interactions. Of course we don't have to restrict ourselves to the all 4-factor, all three-factor, all two-factor, and main-effects only models. We can create models with some three-factor interactions, some two-factors, and some main effects. Like ANOVA, we need to be concerned about creating linear dependencies in the model matrix Φ.

The most difficult part of computing least squares estimates is that they generally involve finding the inverse or generalized inverse of the $p \times p$ matrix $X'X$ (or some similarly sized computation). When p is large, the computation is difficult. When

applying linear-approximation nonparametric methods the problem is finding the generalized inverse of the $s \times s$ matrix $\Phi'\Phi$, which typically has s much larger than p. This becomes particularly awkward when $s > n$. We now consider a device that gives us a model matrix that is always $n \times n$.

7.6.2 Reproducing Kernel Hilbert Space Regression

We now introduce a simple way to use *reproducing kernel Hilbert spaces (RKHSs)* in nonparametric regression. Before doing that we need to discuss some additional background on general linear models. In Subsection 2.2.3 we introduced the idea of equivalent linear models. If $Y = X_1\beta_1 + e_1$ and $Y = X_2\beta_2 + e_2$ are two models for the same dependent variable vector Y, the models are equivalent if $C(X_1) = C(X_2)$. The result that for any X,

$$C(X) = C(XX') \tag{3}$$

is proven in *PA*, Appendix B.4. In particular, this implies that the linear models $Y = X\beta_1 + e_1$ and $Y = XX'\beta_2 + e_2$ are equivalent. For reasons that I hope will become obvious, I have jokingly referred to the result in (3) as *The Fundamental Theorem (for Statistics) of Reproducing Kernel Hilbert Spaces*.

An RKHS transforms a p vector x_i into an s vector $\phi_i = [\phi_0(x_i), \ldots, \phi_{s-1}(x_i)]'$, where not infrequently $s = \infty$. Just as X has rows made up of the x_i's, Φ has rows made up of the ϕ_i's. Just as $XX' = [x_i'x_j]$ is an $n \times n$ matrix of inner products of the x_is, the whole point of RKHSs is that there exists a *reproducing kernel (r.k.)* function $R(\cdot,\cdot)$ with the property that

$$\tilde{R} \equiv [R(x_i,x_j)] = [\phi_i' D(\eta)\phi_j] = \Phi D(\eta)\Phi'$$

is an $n \times n$ inner product matrix of the ϕ_is where $D(\eta)$ is a positive definite diagonal matrix. Moreover, for s finite, $C[\Phi D(\eta)\Phi'] = C(\Phi)$ (see *PA* Section B.4), so fitting the r.k. model

$$Y = \tilde{R}\gamma + e$$

is equivalent to fitting the nonparametric model

$$Y = \Phi\beta + e.$$

The r.k. model is just a reparameterization with $\beta = D(\eta)\Phi'\gamma$. In particular, predictions are easy using the r.k. model,

$$\hat{y}(x) = [R(x,x_1), \ldots, R(x,x_n)]\hat{\gamma}.$$

This equivalence between fitting a linear structure with Φ and fitting one with the $n \times n$ matrix $\tilde{R}$ is sometimes known as the *kernel trick*.

A primary advantage of the kernel trick is simply that, for a known function $R(\cdot,\cdot)$, it is very easy to construct the matrix $\tilde{R}$. (It is time consuming to specify

s different $\phi_j(\cdot)$ functions, as opposed to one $R(\cdot,\cdot)$ function.) Moreover, the $n \times s$ matrix Φ is awkward to use when s is large. $\tilde{R}$ is always $n \times n$, which limits how awkward it can become to use, but also prevents the simplifications that arise when $s < n$.

When $s \geq n$ and the x_is are distinct, it is to be expected that $\tilde{R}$ will be an $n \times n$ matrix of rank n, so it defines a saturated model. Least squares estimates will give fitted values that equal the observations and zero degrees of freedom for error. Nothing interesting will come of fitting a saturated model. We need to deal with this overfitting. Indeed, the kernel trick is typically used together with a penalized (regularized) estimation method such as those discussed in Chapter 8.

If the x_is are not all distinct, as in the discussion of Fisher's Lack-of-Fit Test from Chapter 6 of *PA*, the row structures of X, Φ, and $\tilde{R}$ (no longer nonsingular) are the same. Fitting any of $X\xi$, $\Phi\beta$, and $\tilde{R}\gamma$ by least squares would give exactly the same *pure error* sum of squares ($SSPE$) and degrees of freedom ($dfPE$). Moreover, fitting $\Phi\beta$ and $\tilde{R}\gamma$ would give exactly the same *lack-of-fit* sum of squares and degrees of freedom but, depending on the size of s, there is a good chance that fitting $\Phi\beta$ and $\tilde{R}\gamma$ would give $SSLF = 0$ on 0 $dfLF$. (This is the equivalent of fitting a saturated model when the x_is are not all distinct.)

Different choices of $R(\cdot,\cdot)$, if they have $s \geq n$, typically all give the same $C(\tilde{R})$, which defines either a saturated model or a model with no lack of fit. Thus different choices of $R(\cdot,\cdot)$ typically all give the same model, but they typically are reparameterizations of each other. They give the same least squares fits. But we will see in the next chapter that if you have two different parameterizations of the same model, and obtain estimates by penalizing parameters in the same way (i.e. use the same penalty function for every parameterization), that having the same penalties applied to different parameters leads to different fitted models. So, even though different $R(\cdot,\cdot)$ functions define essentially the same model, applying any standard penalty like ridge regression or lasso, will lead to different fitted values because the equivalent linear models have different parameters that are being shrunk in the same way. The process of shrinking is the same but the parameters are different, thus the end results are different. We saw that different ϕ_js work better or worse on the battery data and there is no way to tell ahead of time which collection will work best. Similarly, different $R(\cdot,\cdot)$s (with the same penalty) work better or worse on different data and there is no way to tell, ahead of time, which will work best.

If you know what ϕ_j functions you want to use, there is not much mathematical advantage to using r.k.s. But you can use R functions that are known to be r.k.s for which it is difficult or, in the case of $s = \infty$, impossible to write down all the ϕ_js. *ALM-III*, Chapter 3 examines r.k.s that correspond to finite polynomial regression and to fitting splines. But there are a wide variety of potential r.k.s, many that correspond to $s = \infty$.

Table 7.7 gives some commonly used r.k.s. Any r.k. that depends only on $\|u - v\|$ is a *radial basis function* r.k.

The hyperbolic tangent in Table 7.7 is not really an r.k. because it can give $\tilde{R}$ matrices that are not nonnegative definite. But any function $R(u,v)$ that is continuous in u can give plausible answers because it leads to fitting models of the form

Table 7.7 Some common r.k. functions. b and c are scalars.

Names	$R(u,v)$
Polynomial of degree d	$(1+u'v)^d$
Polynomial of degree d	$b(c+u'v)^d$
Gaussian (Radial Basis)	$\exp(-b\|u-v\|^2)$
Sigmoid (Hyperbolic Tangent)	$\tanh(bu'v+c)$
Linear Spline (u, v scalars)	$\min(u,v)$
Cubic Spline (u, v scalars)	$\max(u,v)\min^2(u,v)/2-\min^3(u,v)/6$
Thin Plate Spline (2 dimensions)	$\|u-v\|^2\log(\|u-v\|)$

$$f(x)=\sum_{j=1}^{n}\gamma_j R(x,x_j). \tag{4}$$

This idea can be viewed as extending the use of approximating functions with small support, cf. Subsection 7.6.2, from one to higher dimensions in a way that limits the curse of dimensionality. With local support methods, in one dimension you partition the line into say s_* sets and fit a separate one-dimensional wavelet, B spline, or other function for each partition set. The problem is that in p dimensions the number of partition sets (obtained by Cartesian products) quickly gets out of control, s_*^p. The key idea behind kernel methods is to fit a p-dimensional function, not for each partition set but for each observed data point. The number of functions being fitted is n, which is large but manageable, rather than s_*^p which rapidly becomes unmanageably large. The p-dimensional functions used in fitting can be defined as a a product of p one-dimensional wavelet, spline, or other functions or they can be defined directly as p-dimensional functions via some kernel function. The tuning values b and c in Table 7.7 can be viewed as tools for getting the functions centered and scaled appropriately. Fitting n functions to n data points would typically result in overfitting, so penalizing the coefficients, as discussed in the next chapter, is appropriate. As mentioned earlier, when $\tilde{R}$ is a nonsingular matrix (or more generally has the column space associated with finding pure error), it does not matter what function you used to define $\tilde{R}$ because all such matrices are reparameterizations of each other and give the same least squares fitted values. But if you penalize the parameters in a fixed way, the parameterization penalty will have different effects on different parameterizations.

EXAMPLE 7.6.1. I fitted the battery data with the R language's `lm` command using the polynomial functions $R(u,v)=(u'v)^4$, $R(u,v)=(1+u'v)^4$, $R(u,v)=5(7+u'v)^4$ and the Gaussian functions $R(u,v)=\exp(-\|u-v\|^2)$ and $R(u,v)=\exp(-1000\|u-v\|^2)$. I defined x_i to include the intercept. The three polynomial functions gave fitted values $\hat{y}_i$ identical to those from fitting a fourth degree polynomial. (I fitted the fourth degree polynomial several ways including using $\Phi\Phi'$ as the model matrix.) The Gaussian r.k.s have $s=\infty$. The first Gaussian function gave an $\tilde{R}$ matrix that was computationally singular and gave fitted values that were (to me) unexplainable except as a convenient fitting device similar to the hyperbolic tangent discussed later.

The last function gave an $\tilde{R}$ that was computationally invertible and hence gave fitted values with $\hat{y}_i = y_i$. This has overfit the model so penalizing the regression coefficients, as discussed in the next chapter, is advisable.

Figure 7.10 contains the fit of the hyperbolic tangent "kernel" to the battery data using $b = 1$ and $c = 0$. It turns out that (at least computationally) $\tilde{R}$ is a rank 8 matrix with R's `lm` command including only the 1st, 2nd, 3rd, 4th, 11th, 16th, 29th, and 41st columns of $\tilde{R}$. For an 8 parameter model this has a remarkably high value of $R^2 = 0.9996$. Incidentally, this $\tilde{R}$ has negative eigenvalues so is not nonnegative definite. With $b = 5$ and $c = 0$ `lm` uses columns 1, 2, 3, 6, 12, 22, 37 of $\tilde{R}$ and again has $R^2 = 0.9996$. With $b = 10$ and $c = 10$ `lm` fits only the first column of $\tilde{R}$ yet has $R^2 = 0.9526$. In none of these cases has the hyperbolic tangent led to serious overfitting (although it is quite clear from inspecting the R output that we could drop at least one of the columns used in each $c = 0$ example). □

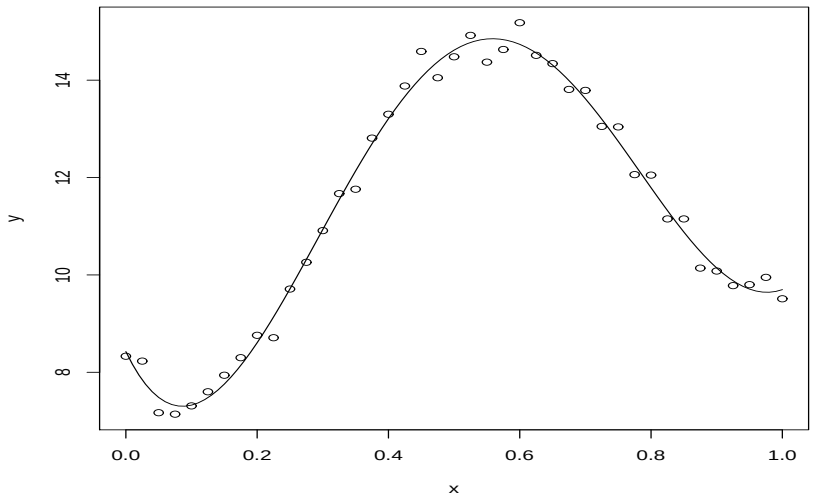

Fig. 7.10 Hyperbolic tangent fit to battery data.

7.7 Testing Lack of Fit in Linear Models

Given a linear model

$$Y = X\beta + e, \qquad \mathrm{E}(e) = 0, \tag{1}$$

any form of fitting a nonparametric regression determines a potential lack-of-fit test procedure. When fitting nonparametric regression via the linear-approximation

models discussed in this chapter, lack-of-fit tests are easy to specify. Because the procedure is based on having a linear-approximation model, essentially the same procedure works regardless of whether one is fitting polynomials, trigonometric functions, wavelets, or splines. (Local polynomials [lowess], because they do not fit a single linear model, do not seem to fit into this procedure.)

Suppose we have the linear model (1) based on predictor variables $x_1, \ldots, x_p$. Given enough data, it may be feasible to produce a nonparametric multiple regression model, say $Y = \Phi\gamma + e$. In practice, this may need to be some generalized additive model. If $C(X) \subset C(\Phi)$, we could just test the reduced model against the full model. Unless Φ is based on polynomials (including polynomial splines), more often than not $C(X) \not\subset C(\Phi)$. In that case we can test the reduced model (1) against the partitioned (analysis of covariance) full model

$$Y = X\beta + \Phi\gamma + e. \tag{2}$$

The F statistic is the standard

$$F = \frac{[SSE(1) - SSE(2)][dfE(1) - dfE(2)]}{MSE(2)}$$

and should be about 1 if the original model is correct. If $e \sim N(0, \sigma^2 I)$ is a good approximation, the test will have an approximate F distribution.

This procedure does not define a single lack-of-fit test. Every different method of picking Φ defines a different test. Which test is best? It seems pretty clear that no best test can possibly exist. If the lack of fit is due to the true model involving cosine curves that were not included in model (1), picking a Φ based on cosine curves should work better than picking a Φ based on polynomials or wavelets. If the lack of fit is due to the true model involving polynomial terms not included in model (1), picking Φ based on polynomials should work better than picking a Φ based on sines, cosines, or wavelets.

What works best will depend on the true nature of the lack of fit. Unfortunately, we do not know the true model. We won't know which of these tests will work best unless we have a very good idea about the true lack of fit and how to model it. If we had those good ideas, we probably wouldn't be thinking about doing a lack-of-fit test.

Incidentally, it is clearly impossible for such tests to be sensitive only to lack of fit in the mean structure. As discussed in *ALM-III* Chapter 5, it is perfectly reasonable to assume that γ in (2) is a random vector with mean 0. In such a case, $\mathrm{E}(Y) = X\beta$, so there is no lack of fit. However the F test will still be sensitive to seeing random values of γ that are very different from 0. Such values of γ will be the result of some combination of heteroscedasticity or dependence among the observations in Y. There is no way to tell from the test itself whether lack of fit or heteroscedasticity or dependence or some combination is causing a large F statistic.

The traditional lack-of-fit tests in Section 3.7 can be viewed through a lens of performing some kind of nonparametric regression on subsets of the data. Most traditional lack-of-fit tests rely on partitioning the data and fitting some kind of

linear model within the partition sets. Fitting models on partitions is nothing more than fitting approximating-functions with small support. Atwood and Ryan's idea for testing lack of fit is just fitting the original model on subsets of the data, so it is essentially fitting multivariate linear splines *without* the requirement that the fitted splines be continuous. Utts' method relies on fitting the original model on only a central group of points, so it implicitly puts each point not in the central group into a separate partition set and fits a separate parameter to each of those noncentral data points. (As alluded to earlier, the irony is that the more parameters you fit the more "nonparametric" your procedure.) Fisher's test fits the biggest model possible that maintains the row structure of the data, cf. *PA-V* Subsection 6.7.2, i.e., the data are partitioned into sets where the predictor variables are identical and a separate parameter if fitted to each set.

Clearly, this model based approach to performing lack-of-fit tests can be extended to testing lack of fit in logistic regression and other generalized linear models.

7.8 Regression Trees

Regression trees can be viewed as a form of linear modeling. In fact, they can be thought of as using forward selection to deal with the dimensionality problems of nonparametric multiple regression. But, unlike standard forward selection, the variables considered for inclusion in the model change with each step of the process. There are a number of different algorithms available for constructing regression trees, cf. Loh (2011). We merely discuss their general motivation. Constructing trees is also known as *recursive partitioning*.

A simple approach to nonparametric regression is to turn the problem into a multifactor ANOVA. (Appendix B contains an illustration of a 3-factor ANOVA.) With p predictor variables, partition each predictor variable into s_* groups. In other words, define s_* indicator functions to partition each variable. Construct the predictor functions as in (7.6.2) by multiplying the indicator fuctions. This amounts to partitioning p dimensional space into s_*^p subsets. Fitting the regression function (7.6.2) amounts to fitting an ANOVA with p factors each at s_* levels, i.e., an s_*^p ANOVA. Fitting the ANOVA model that includes the p-factor interaction is equivalent to fitting a one-way ANOVA with s_*^p groups. If you want to make a prediction for a new point $x = (x_1, \ldots, x_p)'$, just figure out which of the s_*^p partition sets includes x and the prediction is the sample mean of the y observations that fell into that set. Of course this does nothing to help with the curse of dimensionality, but fitting generalized additive models is clearly nothing more than fitting a model that eliminates many of the interactions in the s_*^p ANOVA.

How do you pick the partition sets? The more partition sets you have, the more "nonparametric" the model will be. A reasonable rule of thumb might be to require that if a partition set includes any data at all, it has to include, say, 5 observations. The sets with no data we will ignore and never make predictions there. Five observations in a partition set is not a crazy small number of observations on which

to base a prediction. Once the partition has be determined, we could use backward elimination to find partition sets that can be pooled together. (It would probably be wise to require that partition sets to be pooled must be contiguous in p dimensional space.)

Fitting regression trees is basically the same idea except that they are based on forward selection rather than backward elimination. By using forward selection, the procedure avoids the curse of dimensionality. Usually forward selection can easily miss important features. A nice feature of regression trees is that they pick the partition sets as well as deciding which partition sets need further dividing. In other words, they search through far more than a single set of s_*^p partition sets.

In practice, regression trees are often used with bagging or random forests as discussed in Section 5.5. Regression trees tend to overfit the data, causing low bias but high variability. Bagging is designed to reduce variability.

We now consider two examples. A very simple one to illustrate the ideas and a slightly more complicated one that examines the process.

EXAMPLE 7.8.1. Consider a simple example with $n = 7$ observations and two predictor variables x_1, x_2, specifically

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} \qquad [X_1, X_2] = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \\ 4 & 1 \\ 5 & 5 \\ 6 & 7 \\ 7 & 3 \end{bmatrix}.$$

The first step is to split the data into two parts based on the size of X_1 or X_2. For instance, we can consider a split that consists of the smallest x_1 value and the six largest; or the two smallest x_1 values and the five largest; or the smallest three x_2 values and the largest four. We consider all such splits and posit an initial regression tree model $Y = \Phi^{(1)}\beta + e$, where

$$\Phi^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

The last 12 columns identify all of the possible splits. Columns 2 through 7 are the splits based on x_1 and columns 8 through 13 are the splits based on x_2, with, for example, the tenth column identifying the smallest three x_2 values and, by default since a column of 1's is included, the largest four. Obviously, this initial model is overparameterized; it has 13 predictor variables to explain 7 observations. The

first (intercept) column is forced into the model and one other column is chosen by forward selection. Suppose that column is the fifth, so at the second stage we have the columns

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{or equivalently} \quad \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

forced into the second-stage model matrix. We now consider possible splits *within* the two groups that we have already identified. The first four observations can be split based on the sizes of either x_1 or x_2 and similarly for the last three. The second stage model is $Y = \Phi^{(2)}\beta + e$, where

$$\Phi^{(2)} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Here, columns 3, 4, and 5 are splits of the first group based on the size of x_1 and columns 6, 7, and 8 are splits of the first group based on the size of x_2. Columns 9 and 10 are splits of the second group based on x_1 and columns 11 and 12 are based on x_2. Again, the model is grossly overparameterized. Columns 1 and 2 are forced into the model, and one more column is chosen by forward selection. Suppose it is column 7, so at the third stage we have

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{or equivalently} \quad \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

forced into the model. We now have three groups, and again we consider splitting within groups. At the third stage, we have $Y = \Phi^{(3)}\beta + e$, where

$$\Phi^{(3)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Again, we add a column by forward selection. If no column can be added, we return to the model with the three forced variables,

$$Y = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \beta + e.$$

Note that this is just a one-way ANOVA model, so the parameter estimates are group means. We can identify the groups as (1) $x_1 < 4.5$, $x_2 < 2.5$; (2) $x_1 > 4.5$; and (3) $x_1 < 4.5$, $x_2 > 2.5$. Predictions are based on identifying the appropriate group and use the group mean as a point prediction. Note that this is essentially fitting a step function to the data.

Going back to the original parameterization of the model (i.e., the original choices of columns), the model is

$$Y = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \beta + e.$$

With these choices of the columns, the columns are ordered from left to right, and dropping columns successively from the right still gives a regression tree. □

As discussed in Chapter 5, forward selection defines a sequence of larger and larger models with various ways to determine which variable is added next and various ways to determine when to stop adding variables. Regression trees typically add variables based on minimizing the SSE, which is the traditional method employed in forward selection. Regression trees often employ an unusual stopping rule. Breiman et al. (1984, Section 8.5) suggest continuing the forward selection until each group has five or fewer observations. At that point, one can either accept the final model or pick a best model from the sequence using something like the C_p statistic (assuming that the final model gives a reasonable MSE).

EXAMPLE 7.8.2. In Chapter 1 we considered *The Coleman Report* data. We examine a partitioning created using only two predictor variables, x_3 and x_5 and the R package `rpart`. Details are given in http://www.stat.unm.edu/~fletcher/R-SL.pdf. For illustrative purposes, I required that there could be no fewer than two data points in any partition set. Typically one would do this on a bigger set of data and perhaps require more observations in every partition set. Figures 7.11 and 7.12 illustrate the recursive partitioning process. Figure 7.12 is the tree diagram produced by `rpart`. The algorithm begins by partitioning x_3 four times before it involves x_5. I set up `rpart` to keep running until it did a partition based on x_5.

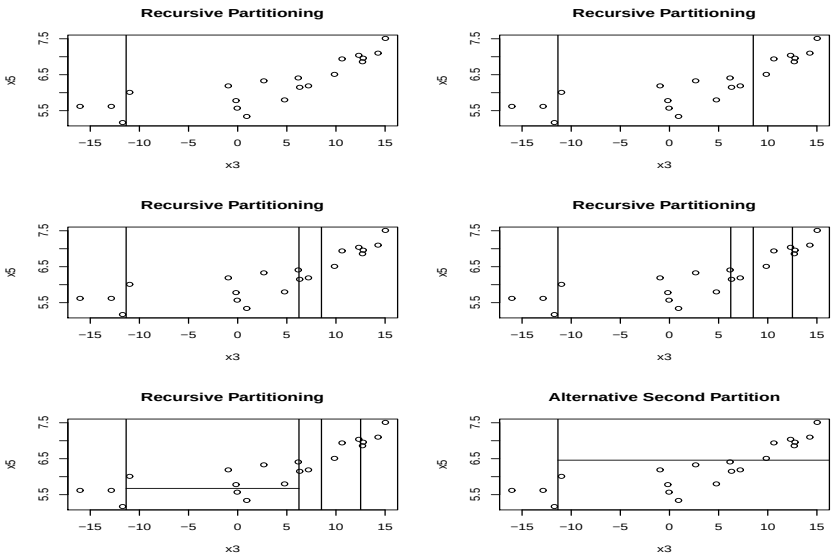

Fig. 7.11 Partition sets for x_3, x_5.

Table 7.9 contains the statistics that determine the first partitioning of the data. The values $x_{3(i)}$ are the ordered values x_3 with $y_{3(i)}$ the corresponding y values (order statistics and induced order statistics). $x_{5(j)}$ and $y_{5(j)}$ are the corresponding values for x_5. For $i = 3$, the partition sets consist of the 3 smallest x_3 observations, and the 17 largest. For $i = 18$ the partition sets are the 18 smallest x_3 observations, and the 2 largest. For $j = 18$ the partition sets are the 18 smallest x_5 observations, and the 2 largest. Built in is the requirement that each partition set contain at least 2 observations. The *SSE*s are from fitting a one-way ANOVA on the two groups. Note that the smallest *SSE* corresponds to $i = 3$, so that is the first partition used. The first split illustrated in Figures 7.11 and 7.12 is at $-11.35 = [x_{3(3)} + x_{3(4)}]/2$ so that the two partition sets include the 3 smallest x_3 observations, and the 17 largest.

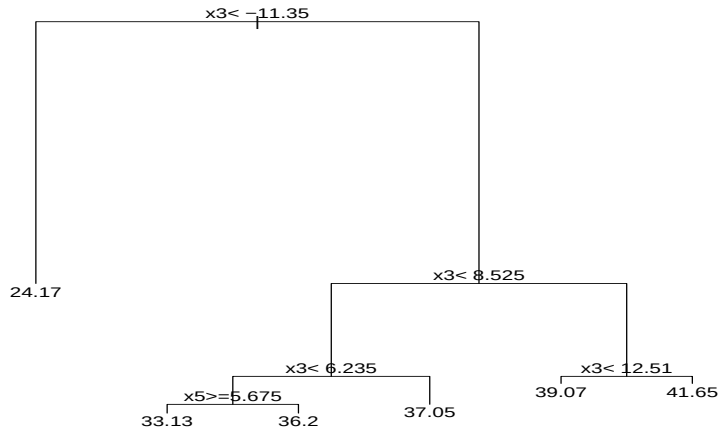

Fig. 7.12 Regression tree for x_3, x_5.

Table 7.8 First tree split.

i	$x_{3(i)}$	$y_{3(i)}$	SSE	j	$x_{5(j)}$	$y_{5(j)}$	SSE
1	−16.04	22.70		1	5.17	26.51	
2	−12.86	23.30	318.5	2	5.34	35.20	603.2
3	−11.71	26.51	222.6	3	5.57	37.20	627.2
4	−10.99	31.70	235.2	4	5.62	22.70	533.4
5	−0.96	33.10	255.8	5	5.62	23.30	394.8
6	−0.17	33.40	266.1	6	5.78	33.40	396.3
7	−0.05	37.20	331.1	7	5.80	34.90	412.7
8	0.92	35.20	349.2	8	6.01	31.70	376.7
9	2.66	31.80	306.1	9	6.15	37.10	413.0
10	4.77	34.90	306.5	10	6.19	33.10	387.8
11	6.16	33.90	283.2	11	6.19	37.01	412.3
12	6.31	37.10	306.6	12	6.33	31.80	356.6
13	7.20	37.01	321.3	13	6.41	33.90	321.3
14	9.85	41.01	394.1	14	6.51	41.01	394.1
15	10.62	39.70	438.2	15	6.86	41.80	468.1
16	12.32	36.51	427.1	16	6.94	39.70	505.2
17	12.70	41.80	492.8	17	6.96	41.01	554.0
18	12.77	41.01	539.6	18	7.04	36.51	539.6
19	14.28	40.70		19	7.10	40.70	
20	15.03	43.10		20	7.51	43.10	

For the second split we consider all the splits of each of the two partition sets from the first stage. Fortunately for our illustration, the partition set $x_3 < -11.35$ has only 3 observations, so we are not allowed to split it further because splitting it has to create a partition set with less than 2 observations. Thus we only need to consider all splits of the set $x_3 \geq -11.35$. In Table 7.10 I have blanked out the observations with $x_3 < -11.35$ but remember that these three observations are still included in the *SSE*s. The minimum *SSE* occurs when $i = 13$ so the next partition set goes from -11.35 to $8.525 = [x_{3(13)} + x_{3(14)}]/2$, as illustrated in Figures 7.11 and 7.12.

Table 7.9 Second tree split.

i	$x_{3(i)}$	$y_{3(i)}$	*SSE*	j	$x_{5(j)}$	$y_{5(j)}$	*SSE*
1				1			
2				2	5.34	35.20	
3				3	5.57	37.20	221.2
4	−10.99	31.70		4			
5	−0.96	33.10	174.5	5			
6	−0.17	33.40	156.1	6	5.78	33.40	211.6
7	−0.05	37.20	170.5	7	5.80	34.90	205.1
8	0.92	35.20	163.5	8	6.01	31.70	177.4
9	2.66	31.80	123.2	9	6.15	37.10	182.1
10	4.77	34.90	107.7	10	6.19	33.10	156.8
11	6.16	33.90	76.6	11	6.19	37.01	158.7
12	6.31	37.10	77.7	12	6.33	31.80	111.8
13	7.20	37.01	73.6	13	6.41	33.90	73.6
14	9.85	41.01	111.6	14	6.51	41.01	111.5
15	10.62	39.70	130.0	15	6.86	41.80	150.3
16	12.32	36.51	109.8	16	6.94	39.70	164.9
17	12.70	41.80	145.7	17	6.96	41.01	187.7
18	12.77	41.01	168.4	18	7.04	36.51	168.4
19	14.28	40.70		19	7.10	40.70	
20	15.03	43.10		20	7.51	43.10	

While `rpart` created the partition just mentioned, in Table 7.10 the value $j = 13$ gives the same *SSE* as $i = 13$. The alternative partition of the (x_3, x_5) plane with $x_3 \geq -11.35$ and x_5 divided at $6.46 = [x_{5(13)} + x_{5(14)}]/2$ is given in the bottom right of Figure 7.11. *It separates the data into exactly the same three groups* as the `rpart` partition. I have no idea why `rpart` chose the partition based on x_3 rather than the alternative partition based on x_5. It looks like, after incorporating the alternative partition, the subsequent partitions would continue to divide *the data* in the same way. However, the final partition sets would be different, which means that predictions could be different. There are 6 final partition sets, so there are only 6 distinct values that will be used to predict, and they will be the same 6 numbers for either partitioning. But the ranges of (x_3, x_5) values over which those 6 predictions are applied change with the different partitions.

The set $x_3 < -11.35$ cannot be split further because of our requirement that all partition sets include two data points. But $-11.35 \le x_3 < 8.525$ has 10 data points, so it can be split $14 = 2 \times (10-3)$ ways, and $x_3 \ge 8.525$ has 7 points, so can be split $8 = 2 \times (7-3)$ ways. That is another 22 ANOVAs to run from which we pick the one with the smallest *SSE*. The minimum occurs when splitting x_3 between $x_{3(11)}$ and $x_{3(12)}$, cf. Figures 7.11 and 7.12. We discontinue the detailed illustration.

An advantage of doing one full s_*^p partition is that you can easily identify empty cells and cells with little data. Prediction variances will reflect that. With a forward selection partition, the algorithms typically create partition sets that restrict the minimum number of observations in a partition. However, looking a Figure 7.11, an observation with, for example, a small value of x_3 and a large value of x_5 is far from the other data in its partition set, so it is unlikely to be predicted well by the mean of the observations in that set. It is not clear to me how one could identify that troublesome phenomenon when fitting a regression tree in higher dimensions □

Exercise 7.1. Without a computer, find the predictions for the point (15,6) from the two partitions. Hint: the `rpart` prediction is based on the average of four y values and the alternative partition prediction is based on two.

There is a strong tendency for regression trees to *overfit* the data, causing poor predictive performance from tree models. Random forests, bagging, and boosting may improve the predictive performance of tree models, cf. Section 5.5. Bagging is aimed at reducing variability while boosting seems to be aimed at reducing bias.

Regression trees are quite good a modeling interactions between the underlying variables. (They create categories similar to multifactor ANOVA.) What trees are not very good at is reproducing linear relationships, or quadratic relationships, or cyclic relationships, etc.

Exercise 7.2. Reanalyze the data in Appendix B by coding the three categorical predictor variables numerically as indicated in Table B.1 and applying a regression tree to the data.

7.9 Regression on Functional Predictors

For each dependent variable y_i, $i = 1, \ldots, n$, suppose we observe a function of predictor variables, say $\mathscr{X}_i(t)$, $t \in \mathscr{T} \subset \mathbf{R}^d$. The predictor function might be observed over time or might result from some sort of medical imaging. For some unknown function $\gamma(t)$ of regression coefficients, we assume the model

$$y_i = \alpha + \int_{\mathscr{T}} \mathscr{X}_i(t)\gamma(t)dt + e_i, \qquad \mathrm{E}(e_i) = 0.$$

As a practical matter, we incorporate a nonparametric characterization of the regression coefficient function using a standard spanning set of functions ϕ_j to get

$$\gamma(t) \doteq \sum_{j=0}^{s-1} \beta_j \phi_j(t) = \phi(t)'\beta.$$

where

$$\phi(t)' \equiv [\phi_0(t), \ldots, \phi_{s-1}(t)]', \qquad \beta \equiv (\beta_0, \ldots, \beta_{s-1})'.$$

This leads to a standard linear model for the observations

$$\begin{aligned} y_i &= \alpha + \int_{\mathscr{T}} \mathscr{X}_i(t)\gamma(t)dt + e_i \\ &\doteq \alpha + \int_{\mathscr{T}} \mathscr{X}_i(t)\phi(t)'\beta dt + e_i \\ &= \alpha + \left[\int_{\mathscr{T}} \mathscr{X}_i(t)\phi(t)'dt\right]\beta + e_i \\ &= \alpha + x_i'\beta + e_i \end{aligned}$$

where

$$x_i' \equiv (x_{i0}, \ldots, x_{i,s-1}), \quad \text{and} \quad x_{ij} \equiv \int_{\mathscr{T}} \mathscr{X}_i(t)\phi_j(t)dt.$$

See Reiss et al. (2017) for additional discussion.

In reality, it is impossible to observe $\mathscr{X}_i(t)$ for every t in an infinite set of points $\mathscr{T}$. At most we can observe $\mathscr{X}_i(t)$ at t_{ik}, $k = 1, \ldots, N_i$ where we would expect N_i to be a very large number. In this case, we would want to use numerical approximations to the integrals, perhaps even something as simple as

$$x_{ij} = \frac{1}{N_i} \sum_{k=1}^{N_i} \mathscr{X}_i(t_{ik})\phi_j(t_{ik}).$$

7.10 Exercises

The first six exercises reexamine the *Coleman Report* data. The first 5 consider only two variables: y, the mean verbal test score for sixth graders, and x_3, the composite measure of socioeconomic status.

Exercise 7.10.1. Rescale x_3 to make its values lie between 0 and 1. Plot the data. Using least squares, fit models with $s = 10$ using polynomials and cosines. Plot the regression lines along with the data. Which family works better on these data, cosines or polynomials?

Exercise 7.10.2. Using $s = 8$, fit the *Coleman Report* data using Haar wavelets. How well does this do compared to the cosine and polynomial fits?

Exercise 7.10.3. Based on the $s = 10$ polynomial and cosine models fitted in Exercise 7.10.1, use C_p to determine a best submodel for each fit. Plot the regression lines for the best submodels. Which family works better on these data, cosines or polynomials? Use C_p to determine the highest frequency needed when fitting sines and cosines.

Exercise 7.10.4. Investigate whether there is a need to consider heteroscedastic variances with the *Coleman Report* data. If appropriate, refit the data.

Exercise 7.10.5. Fit a cubic spline nonparametric regression to the *Coleman Report* data.

Exercise 7.10.6. Fit a regression tree to the *Coleman Report* data using just variable x_4.

Exercise 7.10.7. In Section 6 we set up the interpolating cubic spline problem as one of fitting a saturated linear model that is forced be continuous, have continuous first and second derivatives, and have 0 as the second derivative on the boundaries. In our discussion, the dots are only connected implicitly because a saturated model must fit every data point perfectly. Show that you can find the parameters of the fitted polynomials without using least squares by setting up a system of linear equations requiring the polynomials to connect the (x_i, y_i) dots along with satisfying the derivative conditions.

Exercise 7.10.8. Fit a tree model to the battery data and compare the results to fitting Haar wavelets.

Chapter 8
Alternative Estimates II

Abstract Nonparametric methods are really highly parametric methods. They suffer from fitting so many parameters to the data that the models lose their ability to make effective predictions, i.e. they suffer from overfitting. One way to stop overfitting is by using penalized estimation (regularization) methods. Penalized estimation provides an automated method of keeping the estimates from tracking the data more closely than is justified.

8.1 Introduction

In applications of linear model theory to situations where the number of model parameters is large relative to the sample size n, it is not uncommon to replace least squares estimates with estimates that incorporate a penalty on (some of) the regression coefficients. Nonparametric regression models are germane examples. Penalty functions are often used to avoid *overfitting* a model. (Chapters 3 and 7 contain plots of overfitted models.) Penalty functions can make it possible to use numbers of parameters that are similar to the number of observations without overfitting the model. As a tool to avoid overfitting, penalized estimation constitutes an alternative to variable selection. Penalized estimation generally results in biased estimates (but not necessarily so biased as to be a bad thing).

Penalized estimates are determined by adding some multiple of a nonnegative *penalty function* to the least squares criterion function and minimizing this new criterion function. Incorporating a penalty function is sometimes referred to as *regularization*. For simplicity we will discuss fitting the standard regression model

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots \beta_{p-1} x_{i,p-1} + \varepsilon_i = x_i'\beta + \varepsilon_i, \qquad i = 1, \ldots, n$$

or

$$Y = X\beta + e, \quad \mathrm{E}(e) = 0, \quad \mathrm{Cov}(e) = \sigma^2 I, \tag{1}$$

One might then minimize

$$(Y - X\beta)'(Y - X\beta) + k\mathscr{P}(\beta), \tag{2}$$

where $\mathscr{P}(\beta)$ is a nonnegative penalty function and $k \geq 0$ is a *tuning parameter*. If $k = 0$, the estimates are obviously least squares estimates. Many penalty functions have minimums at the vector $\beta = 0$, so as k gets large, the penalty function dominates the minimization and the procedure, in some fashion, shrinks the least squares estimate of β towards 0. In reality, the intercept term is almost never penalized, so some form of partitioned linear model is almost always used.

Many of the commonly used $\mathscr{P}$s penalize each regression coefficient the same amount. As a result, it is often suggested that the predictors in the model matrix X should be standardized onto a common scale. If the height of my doghouse is a predictor variable, the appropriate regression coefficient depends a great deal on whether the height is measured in miles or microns. For a penalty function to be meaningful, it needs to be defined on an appropriate scale for each predictor variable.

Typically we assume that the $X\beta$ portion of the model contains an intercept or its equivalent. Mathematically that means J, the vector of 1s, satisfies $J \in C(X)$. We put the predictor variables $x_1, \ldots, x_{p-1}$, into an $n \times (p-1)$ matrix $Z = [x_{ij}]$ so that the overall model matrix is $X = [J, Z]$. The predictor variables are then often recentered by their sample means. In other words, the observations on the jth predictor, the x_{ij}s $i = 1, \ldots, n$, are replaced by $x_{ij} - \bar{x}_{\cdot j}$. In matrix terms this means that Z is replaced by $[I - (1/n)JJ']Z = Z - J\bar{x}_{\cdot}'$ where $\bar{x}_{\cdot}' \equiv (\bar{x}_{\cdot 1}, \ldots, \bar{x}_{\cdot p-1})$. Most often the predictors are also rescaled so that the x_{ij}s get replaced by the values $(x_{ij} - \bar{x}_{\cdot j})/s_j$ where s_j is the sample standard deviation of the x_{ij}s, $i = 1, \ldots, n$. In matrix terms this means that Z is replaced by $[I - (1/n)JJ']ZD(s_j)^{-1}$. Note that

$$C(X) \equiv C(J, Z) = C\left\{J, [I - (1/n)JJ']Z\right\} = C\left\{J, [I - (1/n)JJ']ZD(s_j)^{-1}\right\},$$

so the recentered model and the recentered, rescaled model are both equivalent to the original model. The recentered model typically has a different intercept parameter from the original model and the recentered, rescaled model typically has all of its parameters different from the original model. This matters because we are penalizing the parameters, so changing what the parameters mean, changes the result we get. Rather than modifying the penalty function $\mathscr{P}$ to be appropriate to our original model, recentering and rescaling the predictor variables allows us to use an "off the shelf" penalty function to obtain reasonable results.

Rarely is there an obvious choice for the tuning parameter k. Extending the idea of Hoerl and Kennard (1970), we can use a *trace plot* to pick k. Denote the penalized estimate for given k and the jth predictor variable as $\tilde{\beta}_{kj}$. The trace plot is, for all j, a simultaneous plot of the curves defined by $(k, \tilde{\beta}_{kj})$ as k varies. As mentioned, for $k = 0$ the $\tilde{\beta}_{0j}$s are the least squares estimates and as k increases they typically all shrink towards 0. For the purpose of dealing with collinearity issues, Hoerl and Kennard suggested picking a small k for which the estimates settle down, i.e., stop

varying wildly. Draper and van Nostrand (1979) conclude that for ridge regression the problems with picking k using trace plots outweigh their benefits. More modern methods of picking k include cross-validation and generalized cross-validation, cf. Hastie, Tibshirani, and Friedman (2016) or any number of other sources.

Least squares is a geometric estimation criterion, not a statistical criterion. It minimizes squared distance and is achieved by projecting Y into $C(X)$. But least squares has many nice statistical properties. Penalized least squares is also a geometric criterion, not a statistical one. Unfortunately, it is harder to establish nice statistical properties for penalized estimates. That is *not* to say that they don't have any. Section 5 illustrates some geometry related to penalized least squares.

If model (1) has multivariate normal errors, the least squares residuals will be independent of the penalized fitted values, so if (1) provides an adequate number of degrees of freedom for error, it may be advantageous to use the least squares residuals, rather than residuals based on the penalized estimates, to estimate the variance and to check model assumptions. To establish this, we merely need to show that the penalized estimates are a function of the least squares estimates. Using ideas similar to the proof that least squares estimates satisfy $X\hat{\beta} = MY$, it is possible to write

$$\begin{aligned} \|Y - X\beta\|^2 &= (Y - X\beta)'(Y - X\beta) \\ &= (Y - X\hat{\beta})'(Y - X\hat{\beta}) + (\beta - \hat{\beta})'X'X(\beta - \hat{\beta}). \end{aligned} \tag{3}$$

The estimation criterion (2) becomes

$$(Y - X\hat{\beta})'(Y - X\hat{\beta}) + (\beta - \hat{\beta})'X'X(\beta - \hat{\beta}) + k\mathscr{P}(\beta)$$

in which the first term $(Y - X\hat{\beta})'(Y - X\hat{\beta})$ does not involve β, so is irrelevant to the minimization, and the other terms depend on Y only through $\hat{\beta}$. Since the residuals $\hat{e} = (I - M)Y$ are independent of $\hat{\beta}$, the residuals are independent of the penalized estimate which must be a function of $\hat{\beta}$. Moreover, if the penalized estimate is a *linear* function of $\hat{\beta}$ (e.g. ridge regression), $\hat{e}$ is uncorrelated with the penalized estimate even without the assumption of multivariate normality. The decomposition of the squared distance in (3) will be used again in Subsection 5.1 to facilitate geometric interpretations.

8.1.1 Reparameterization and RKHS Regression: It's All About the Penalty

In traditional linear models it is well know that reparameterizations are irrelevant, i.e., two models for the same data, say, $Y = X_1\beta_1 + e$ and $Y = X_2\beta_2 + e$ are equivalent if $C(X_1) = C(X_2)$. In particular, least squares gives the same fitted values $\hat{Y}$ and residuals $\hat{e}$ for each model. Moreover, the least squares estimates for either β_1 or β_2 may not be uniquely defined, but we don't much care. In penalized least squares,

if you use the same penalty function $\mathscr{P}(\cdot)$ for each of two equivalent models, you typically get different results, i.e., minimizing $\|Y - X_1\beta_1\|^2 + \mathscr{P}(\beta_1)$ does not give the same fitted values and residuals as minimizing $\|Y - X_2\beta_2\|^2 + \mathscr{P}(\beta_2)$. Moreover, incorporating the penalty function typically generates unique estimates, even when ordinary least squares does not. This reparameterization issue is a substantial one when using software that provides a default penalty function or even a menu of penalty functions.

To get equivalent results from equivalent models you need appropriate penalty functions. In particular, if X_1 and X_2 are both regression models so that $X_1 = X_2T$ for some invertible matrix T, then $\beta_2 = T\beta_1$ and minimizing $\|Y - X_1\beta_1\|^2 + \mathscr{P}(\beta_1)$ clearly gives the same fitted values and residuals as minimizing $\|Y - X_2\beta_2\|^2 + \mathscr{P}(T^{-1}\beta_2)$.

This discussion is particularly germane when applying the kernel trick as in Subsection 1.7.2. With two different reproducing kernels $R_1(\cdot,\cdot)$ and $R_2(\cdot,\cdot)$, for which $C(\tilde{R}_1) = C(\tilde{R}_2)$, the models $Y = \tilde{R}_1\gamma_1 + e$ and $Y = \tilde{R}_2\gamma_2 + e$ are reparameterizations of each other. Their least squares fits will be identical and with many kernels $Y = \hat{Y}_1 = \hat{Y}_2$ (when the x_i vectors are distinct). If, to avoid overfitting, we use an off the shelf penalty function $\mathscr{P}(\cdot)$ to estimate the γ_is, that common penalty function will be entirely responsible for the differences between the fitted values $\tilde{Y}_1 \equiv \tilde{R}_1\tilde{\gamma}_1$ and $\tilde{Y}_2 \equiv \tilde{R}_2\tilde{\gamma}_2$ as well as any differences in other predictions made with the two models.

8.1.2 Nonparametric Regression

As discussed in the previous chapter, one approach to nonparametric regression of y on a scalar predictor x is to fit a linear model, say,

$$y_i = \beta_0 + \gamma_1\phi_1(x_i) + \cdots + \gamma_s\phi_s(x_i) + \varepsilon_i$$

for known functions ϕ_j, e.g., polynomial regression. Penalized estimates are used in nonparametric regression to ensure smoothness. Penalized regression typically shrinks all regression estimates towards 0, some more than others when applied to nonparametric regression. Variable selection differs in that it shrinks the estimates of the eliminated variables to (not towards) 0 but lets least squares decide what happens to the estimates of the remaining variables.

The functions $\phi_j(x)$ in nonparametric regression are frequently subjected to some form of standardization when they are defined, thus obviating a strong need for further standardization of the vectors $\Phi_j \equiv [\phi_j(x_1), \cdots, \phi_j(x_n)]'$, especially when the x_is are equally spaced. For example, with x_is equally spaced from 0 to 1 and $\phi_j(x) = \cos(\pi jx)$, there is little need to standardize $Z \equiv [\Phi_1, \ldots, \Phi_s]$ further. When using simple polynomials $\phi_j(x) = x^j$, the model matrix *should* be standardized. When using the corresponding *Legendre polynomials* on equally spaced data, Z need not be.

In the context of nonparametric regression, not overfitting the model typically means ensuring appropriate smoothness. For example, with $\phi_j(x) = \cos(\pi jx)$, when j is large the cosine functions oscillate very rapidly, leading to *nonsmooth* or *noisy* behavior. Often, with linear-approximation approaches to nonparametric regression, large j is indicative of more noisy behavior. We want to allow noisy behavior if the data require it, but we prefer smooth functions if they seem reasonable. It therefore makes sense to place larger penalties on the regression coefficients for large j. In other words, for large values of j we often shrink the least squares estimate $\hat{\beta}_j$ towards 0 more than when j is small.

8.2 Ridge Regression

Classical ridge regression provides one application of penalty functions. For the model

$$Y = X\beta + e, \quad \mathrm{E}(e) = 0,$$

the simplest version of ridge regression is obtained by minimizing

$$(Y - X\beta)'(Y - X\beta) + k\beta'\beta. \tag{1}$$

This involves the penalty function $\mathscr{P}(\beta) = \beta'\beta$. To find the ridge regression estimates we augment the regression model with p artificial observations in which $0 = \sqrt{k}\beta_j + \tilde{\varepsilon}_{n+j}$. This leads to the augmented model

$$\begin{bmatrix} Y \\ 0 \end{bmatrix} = \begin{bmatrix} X \\ \sqrt{k}I_p \end{bmatrix} \beta + \begin{bmatrix} e \\ \tilde{e} \end{bmatrix}. \tag{2}$$

The extra observations are basically saying that we have gotten a look at β_j, subject to some error, and we saw it to be 0. (The tuning parameter k affects the amount of error involved.) Clearly the extra observations force the estimated regression coefficients closer to 0. It is not difficult to see that the least squares criterion for fitting model (2) is identical to the ridge regression criterion (1). The ridge regression estimates are just the least squares estimates from model (2), which simplify to

$$\tilde{\beta}_R = (X'X + kI)^{-1}X'Y. \tag{3}$$

This is the simplest form of ridge regression, but nobody actually does this, because nobody penalizes the intercept term.

A more realistic version is to write the multiple linear regression model with $X = [J, Z]$ as

$$Y = J\beta_0 + Z\beta_* + e = [J, Z] \begin{bmatrix} \beta_0 \\ \beta_* \end{bmatrix} + e = X\beta + e.$$

Here $\beta_* = (\beta_1, \ldots, \beta_{p-1})'$ and $Z = [x_{ij}]$, $i = 1, \ldots, n$, $j = 1, \ldots, p-1$. Classical ridge regression obtains estimates β_0 and β_* by minimizing

$$(Y - J\beta_0 - Z\beta_*)'(Y - J\beta_0 - Z\beta_*) + k\beta_*'\beta_* \tag{4}$$

which amounts to using the penalty function

$$\mathscr{P}_R(\beta) \equiv \beta_*'\beta_* = \sum_{j=1}^{p-1} \beta_j^2.$$

Other than β_0, this penalizes each β_j the same amount, so it is important that the columns of Z be standardized to a common length or that they be defined in such a way that they are already nearly standardized. It is easy to see that the function (4) is the least squares criterion function for the model

$$\begin{bmatrix} Y \\ 0 \end{bmatrix} = \begin{bmatrix} J \\ 0 \end{bmatrix} \beta_0 + \begin{bmatrix} Z \\ \sqrt{k}I \end{bmatrix} \beta_* + \begin{bmatrix} e \\ \tilde{e} \end{bmatrix}. \tag{5}$$

Fitting model (5) is a little more complicated than fitting model (2).

To fit model (5) it helps to think about replacing the original regression with the mean adjusted equivalent version, $y_i = \alpha + \sum_{j=1}^{p-1} \beta_j (x_{ij} - \bar{x}_{\cdot j}) + \varepsilon_i$ or

$$Y = J\alpha + [Z - J\bar{x}_\cdot']\beta_* + e = J\alpha + [I - (1/n)JJ']Z\beta_* + e.$$

The β_* vector is unchanged but

$$\beta_0 = \alpha - \bar{x}_\cdot'\beta_*.$$

The augmented version of this model becomes

$$\begin{bmatrix} Y \\ 0 \end{bmatrix} = \begin{bmatrix} J \\ 0 \end{bmatrix} \alpha + \begin{bmatrix} [I - (1/n)JJ']Z \\ \sqrt{k}I \end{bmatrix} \beta_* + \begin{bmatrix} e \\ \tilde{e} \end{bmatrix}.$$

Having corrected the predictor variables for their means, the least squares estimates are now easier to find for this augmented model and become the classical ridge regression estimates

$$\tilde{\beta}_{R*} = \left\{ Z'[I - (1/n)JJ']Z + kI \right\}^{-1} Z'[I - (1/n)JJ']Y$$

and

$$\hat{\alpha} = \bar{y}_\cdot, \quad \text{so} \quad \tilde{\beta}_{R0} = \bar{y}_\cdot - \bar{x}_\cdot'\tilde{\beta}_{R*}.$$

Again, the augmented regression model shows quite clearly that ridge regression is shrinking the regression parameters toward 0. The bottom part of the augmented model specifies

$$0 = \sqrt{k}\beta_* + \tilde{e},$$

so we are acting like 0 is an observation with mean vector $\sqrt{k}\beta_*$, which will shrink the estimate of β_* toward the 0 vector. Note that if $\sqrt{k}$ is already a very small number, then one expects $\sqrt{k}\beta_*$ to be small, so the shrinking effect of the artificial ob-

servations 0 will be small. If $\sqrt{k}$ is large, say 1, then we are acting like we have seen that β_* is near 0 and the shrinkage will be larger.

8.2.1 Generalized Ridge Regression

Returning to the unpartitioned model (8.1.1), *generalized ridge regression* takes the form of a penalty

$$\mathscr{P}_{GR}(\beta) \equiv \beta' Q \beta,$$

where for simplicity we focus on $Q \equiv D(q)$, a diagonal matrix with nonnegative entries q_j. This makes the penalty function $\mathscr{P}_{GR}(\beta) = \sum_{j=1}^{p-1} q_j \beta_j^2$. (In general Q can be a nonnegative definite matrix.)

We can minimize

$$(Y - X\beta)'(Y - X\beta) + k\beta' D(q)\beta \tag{6}$$

using the least squares fit to the augmented linear model

$$\begin{bmatrix} Y \\ 0 \end{bmatrix} = \begin{bmatrix} X \\ \sqrt{k} D(\sqrt{q}) \end{bmatrix} \beta + \begin{bmatrix} e \\ \tilde{e} \end{bmatrix} \tag{7}$$

where $D(\sqrt{q})$ has diagonal elements $\sqrt{q_j}$. Note that if q_j is 0, β_j is not penalized. The least squares estimates for model (7) minimize the quantity

$$\begin{bmatrix} Y - X\beta \\ -\sqrt{k} D(\sqrt{q})\beta \end{bmatrix}' \begin{bmatrix} Y - X\beta \\ -\sqrt{k} D(\sqrt{q})\beta \end{bmatrix} = (Y - X\beta)(Y - X\beta) + k\beta' D(q)\beta,$$

which is (6). It is not difficult to show that

$$\tilde{\beta} = [X'X + kD(q)]^{-1} X'Y \tag{8}$$

is the least squares estimate for a regression model (7) and thus is the generalized ridge estimate. Of course the generalized ridge augmented model (7) becomes the classical ridge augmented model (2) when $D(q) = I_p$.

Alternatively, when $D(q)$ is nonsingular, we can minimize (6) using the weighted least squares fit to the augmented linear model

$$\begin{bmatrix} Y \\ 0 \end{bmatrix} = \begin{bmatrix} X \\ I_p \end{bmatrix} \beta + \begin{bmatrix} e \\ \tilde{e} \end{bmatrix}, \mathrm{E} \begin{bmatrix} e \\ \tilde{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathrm{Cov} \begin{bmatrix} e \\ \tilde{e} \end{bmatrix} = \sigma^2 \begin{bmatrix} I_n & 0 \\ 0 & (1/k) D(q)^{-1} \end{bmatrix}. \tag{9}$$

To obtain BLUEs in model (9), the weighted least squares estimates minimize the quantity

$$\begin{bmatrix} Y - X\beta \\ -\beta \end{bmatrix}' \begin{bmatrix} I_n & 0 \\ 0 & kD(q) \end{bmatrix} \begin{bmatrix} Y - X\beta \\ -\beta \end{bmatrix} = (Y - X\beta)'(Y - X\beta) + k\beta' D(q)\beta,$$

which again is the generalized ridge regression criterion (6). The weighted least squares estimates for the weights vector is $w' = [J_n', k\,q']$ also devolve to (8). If $q = J_p$ we get classical ridge regression. Almost all regression software fits weighted least squares. The weighted least squares idea will again be useful in Chapter 9 when we generalize ridge regression to binomial regression models.

In generalized ridge regression choices for weights generally follow a pattern of more shrinkage for β_js that incorporate noisier behavior into the model. This is particularly apt in nonparametric regression and is discussed in Subsection 3.

8.2.2 Picking k

As alluded to in Section 8.1, to pick k in (1) for classical ridge regression, Hoerl and Kennard (1970) suggested plotting a *ridge trace*. If the classical ridge regression estimates are denoted $\tilde{\beta}_{kj}$ for the jth predictor variable and tuning parameter k, the ridge trace is, for all j, a simultaneous plot of the curves defined by $(k, \tilde{\beta}_{kj})$ as k varies. For $k = 0$ the $\tilde{\beta}_{0j}$s are the least squares estimates and as k increases they all shrink towards 0 (except β_{k0} which is not penalized). The idea is to pick k just big enough to stabilize the regression coefficients. Hoerl and Kennard's original idea was using ridge to deal with high collinearity in $[I - (1/n)JJ']Z$, rather than using shrinkage as an alternative to variable selection. As mentioned earlier, the trace plot idea applies to all penalized regression but Draper and van Nostrand (1979) found it lacking for ridge regression.

More recently, cross-validation and generalized cross-validation have been used to pick k, for example see Green and Silverman (1994, Sections 3.1 and 3.2).

8.2.3 Nonparametric Regression

We use the notation from Chapter 7 for simple nonparametric regression but the ideas extend immediately to multiple nonparametric regression.

Assuming that $f(x) = \sum_{j=0}^{s-1} \beta_j \phi_j(x)$, the generalized ridge regression estimate minimizes

$$(Y - \Phi\beta)'(Y - \Phi\beta) + k\beta' D(q)\beta,$$

where $D(q)$ is a nonnegative definite matrix of penalties for unsmoothness and k is a tuning parameter which for many purposes is considered fixed but which ultimately is estimated. Again, it seems most common not to penalize the intercept parameter when one exists.

In the special case in which $\frac{1}{\sqrt{n}}\Phi$ has orthonormal columns, it is not difficult to see that the generalized ridge estimate is

$$\tilde{\beta} = [nI + k\,D(q)]^{-1}\Phi' Y$$

$$= [D(nJ_s + kq)]^{-1}\Phi' Y$$
$$= D\left(\frac{n}{n+kq_j}\right)\hat{\beta},$$

where $\hat{\beta}$ is the least squares estimate. By letting $\alpha = k/n$, we get

$$\tilde{\beta}_j = \frac{1}{1+\alpha q_j}\hat{\beta}_j,$$

which shows quite clearly the nature of the shrinkage.

A frequently used nondiagonal penalty matrix Q does not seem to require Φ to have columns of near equal length. It takes

$$Q = [q_{rs}], \qquad q_{rs} = \int_0^1 \phi_r^{(2)}(x)\phi_s^{(2)}(x)dx$$

with $\phi_r^{(2)}(x) \equiv \mathbf{d}^2\phi_r(x)$ is the second derivative of $\phi_r(x)$. This penalty function does not depend on the data (X,Y). Whenever $\phi_0 \equiv 1$, $\phi_0^{(2)} \equiv 0$, so the first row and column of the second derivative Q will be 0. This places no penalty on the intercept and we could choose to think of penalizing a partitioned model.

Clearly, any constant multiple of the matrix Q works equivalently to Q if we make a corresponding change to k. If we use the cosine basis of (7.1.2), the second derivative matrix Q is proportional to $D(q)$ with

$$q = [0, 1^4, 2^4, \ldots, (s-1)^4]'.$$

For the sines and cosines of (7.1.3), Q is proportional to $D(q)$ with

$$q = \left\{0, 1^4, 1^4, 2^4, 2^4, \ldots, [(s-1)/2]^4, [(s-1)/2]^4\right\}'.$$

It is clear that the terms getting the greatest shrinkage are the terms with the largest values of j in (1.1.2) and (1.1.3), i.e., the highest frequency terms.

EXAMPLE 8.2.1. For the voltage drop data of Chapter 1, using cosines with $j = 0, \ldots, 10$ and least squares, the estimated regression equation is

$$y = 11.4 - 1.61c_1 - 3.11c_2 + 0.468c_3 + 0.222c_4 + 0.196c_5 + 0.156c_6 + 0.0170c_7 + 0.0799c_8 + 0.0841c_9 + 0.148c_{10}.$$

Using the generalized ridge regression augmented model

$$\begin{bmatrix} Y \\ 0 \end{bmatrix} = \begin{bmatrix} \Phi \\ \sqrt{k}D(\sqrt{q}) \end{bmatrix}\beta + \begin{bmatrix} e \\ \tilde{e} \end{bmatrix}$$

with $\sqrt{k} = 0.2$ and $D(\sqrt{q}) = \text{Diag}(0, 1, 4, 9, \ldots, 100)$, the estimated regression equation becomes

$$y = 11.4 - 1.60c_1 - 3.00c_2 + 0.413c_3 + 0.156c_4 + 0.0925c_5 \\ +0.0473c_6 + 0.0049c_7 + 0.0102c_8 + 0.0068c_9 + 0.0077c_{10}.$$

Note the shrinkage towards 0 of the coefficients relative to least squares, with more shrinkage for higher values of j.

Figure 8.1 gives the data along with the generalized ridge regression fitted cosine curve using $j = 0, \ldots, 10$. With $k = 0.04$, the plot is very similar to the unpenalized cosine curve using $j = 0, \ldots, 6$ which is also plotted. Defining R^2 as the squared correlation between the observations and the fitted values, the ridge regression gives $R^2 = 0.985$ which is less than the value 0.988 from the least squares fit with 6 cosines. □

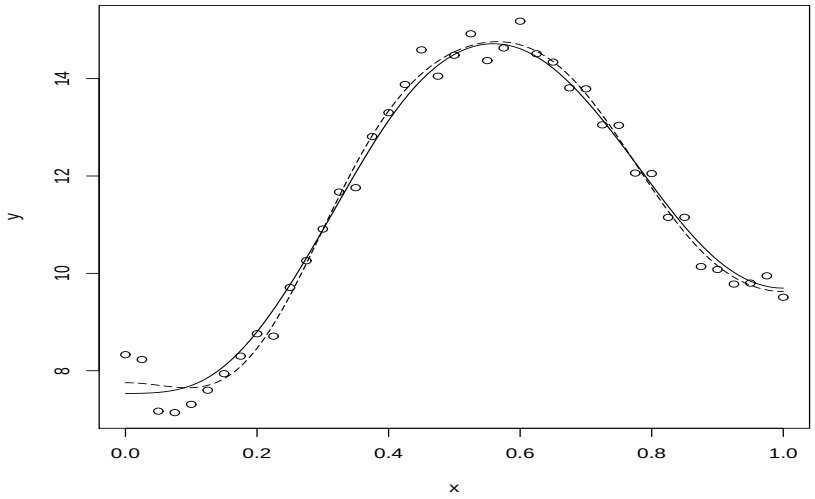

Fig. 8.1 Solid: Generalized ridge regression cosine fit with $k = .04$, $s - 1 = 10$ and second derivative weights for the battery data. Dashed: Least squares cosine fit with $s - 1 = 6$.

The second derivative penalty approach is worthless for Haar wavelets because it gives $Q = 0$ and thus the least squares estimates. Theoretically, one could use penalized least squares with other wavelets. The integral of the product of the second derivatives would be difficult to find for many wavelets. Fitting functions with small support inherently makes the fitted functions less smooth. Choosing how small to make the supports, e.g. choosing how many wavelets to fit, is already a choice of how smooth to make the fitted function. In such cases a smoothing penalty associated with ϕ_j should increase as the size (length, area, volume) of the support of ϕ_j gets smaller.

If x is a vector, the second derivative of $\phi_j(x)$ is a square matrix as is the product of the second derivatives for different j. One might use something like the determinant of the integral of the matrix product to define Q.

8.3 Lasso Regression

Currently, a very popular method for fitting model (8.1.1) is Tibshirani's (1996) *lasso (least absolute shrinkage and selection operator)* which uses the penalty function

$$\mathscr{P}_L(\beta) \equiv \sum_{j=1}^{p-1} |\beta_j| \equiv \|\beta\|_1. \tag{1}$$

The book by Hastie, Tibshirani, and Wainwright (2015) provides a wealth of information on this procedure. For applications with $p > n$ see Bühlmann and van de Geer (2011) or Bühlmann, Kalisch, and Meier (2014).

Because the lasso penalty function is not a quadratic form in β, unlike ridge regression the estimate cannot be obtained by fitting an augmented linear model using (weighted) least squares. Lasso estimates can be computed efficiently for a variety of values k using a modification of the *LARS* algorithm of Efron et al. (2004).

Less computationally efficient than LARS, but easier to understand, is an algorithm that involves obtaining weighted least squares estimates $\tilde{\beta}^{h+1}$ for a partitioned version of the augmented model (8.2.9), namely,

$$\begin{bmatrix} Y \\ 0 \end{bmatrix} = \begin{bmatrix} J_n & Z \\ 0 & I_{p-1} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_* \end{bmatrix} + \begin{bmatrix} e \\ \tilde{e} \end{bmatrix}, \tag{2}$$

$$\mathrm{E}\begin{bmatrix} e \\ \tilde{e} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \mathrm{Cov}\begin{bmatrix} e \\ \tilde{e} \end{bmatrix} = \sigma^2 \begin{bmatrix} I_n & 0 \\ 0 & (1/k)D(q)^{-1} \end{bmatrix}.$$

The method repeatedly finds BLUEs wherein $D(q)^{-1} = D(|\tilde{\beta}_*^h|)$; taking $|\tilde{\beta}_*^h|$ to be the vector of absolute values from a previous set of weighted least squares estimates for β_*. This means that the weights on the actual observations are 1 but the $(n+j)$th weight is $w_{n+j} = k/|\tilde{\beta}_j^h|$ and the penalty function is

$$\mathscr{P}(\beta) = \sum_{j=1}^{p-1} \frac{\beta_j^2}{|\tilde{\beta}_j^h|} \doteq \sum_{j=1}^{p-1} |\beta_j|.$$

When $|\tilde{\beta}_j^h|$ gets small, β_j becomes very highly penalized, thus forcing $\tilde{\beta}_*^{h+1}$ even closer to 0.

As the actual name (not the acronym) suggests, one of the benefits of the lasso penalty is that it automates variable selection. Rather than gradually shrinking all regression coefficients towards 0 like ridge regression, lasso can make some of the regression coefficients collapse to 0.

The lasso penalty (1) treats every coefficient the same, so it would typically be applied to standardized predictors. An obvious modification of lasso that penalizes coefficients at different rates has

$$\mathscr{P}_{GL}(\beta) = \sum_{j=0}^{s} q_j |\beta_j|$$

with $q_j \geq 0$ often increasing in j when x_j (or more commonly $\phi_j(x)$) becomes more noisy as j increases.

Section 4.3 contains an example of the lasso applied to a 5 predictor regression problem. (The *Coleman Report* data.) Here we illustrate its use in nonparametric regression.

EXAMPLE 8.3.1. For the battery data of Chapter 7, Figure 8.2 shows the least squares cosine fits for $s-1 = 6$, 30 and the R package `lasso2`'s default fit except that, with equally spaced cosine predictors, the predictors were not standardized. (I also looked at the standardized version and it made little difference.) The default lasso fit has $k = 12.2133$, which is a lot of shrinkage. (The default is actually $\delta = 0.5\|\hat{\beta}_*\|_1$ where $\hat{\beta}_*$ is the $p-1 = 30$ least squares estimate vector without the intercept, $\|\hat{\beta}_*\|_1 \equiv \sum_{j=1}^{30} |\hat{\beta}_j|$, and δ is defined in Section 4.)

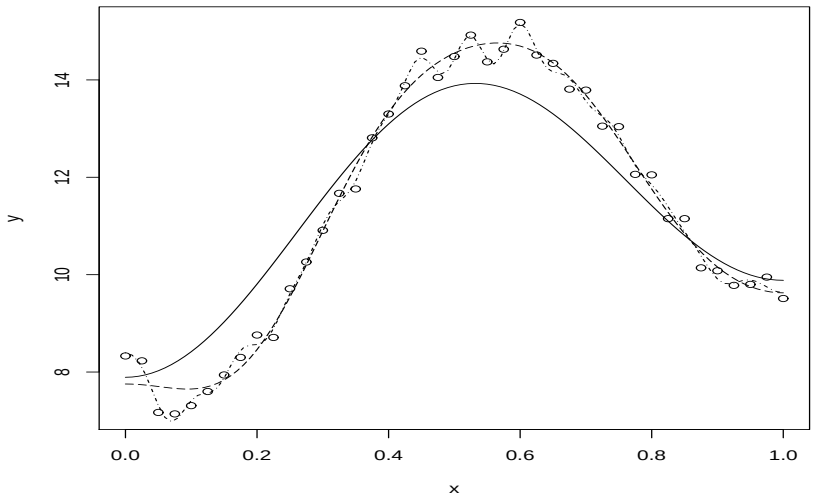

Fig. 8.2 Solid: Lasso cosine fit with $k = 12.2133$ ($\delta = 0.5\|\hat{\beta}_*\|_1$), $p-1 = 30$. Dot-dash: Least squares cosine fit with $p-1 = 30$. Dashed: Least squares cosine fit with $p-1 = 6$.

The default is a shockingly bad fit. It gives $R^2 = 0.951$, which is poor for this problem. It has zeroed out too many of the cosine terms. A more reasonable lasso

fit is given in Figure 8.3. The fit in Figure 8.3 has nonzero coefficients on precisely the first six cosine terms (and the constant) and it gives $R^2 = 0.981$, which cannot be greater than than the R^2 provided by the least squares fit on the six cosine terms. Unlike our ridge regression example for these data, in neither of the lasso fits have we put larger penalties on more noisy variables. □

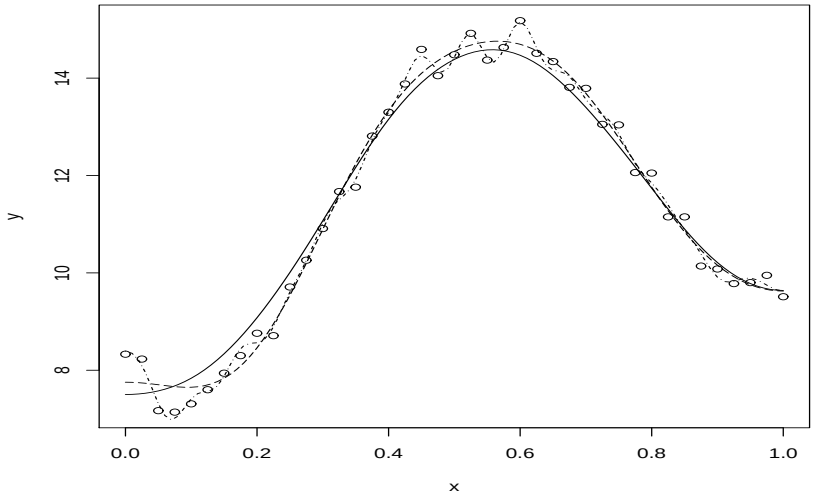

Fig. 8.3 Solid: Lasso cosine fit with $k = 3.045046$ ($\delta = 0.7\|\hat{\beta}_*\|_1$), $p-1 = 30$. Dot-dash: Least squares cosine fit with $p-1 = 30$. Dashed: Cosine least squares fit with $p-1 = 6$.

8.4 Geometric Approach

For model (8.1.1) the penalized least squares estimate minimizes

$$\|Y - X\beta\|^2 + k\mathscr{P}(\beta) \tag{1}$$

for some tuning parameter $k \geq 0$. Alternatively, the procedure can be defined as choosing β to minimize the least squares criterion

$$\|Y - X\beta\|^2, \tag{2}$$

subject to a restriction on the regression coefficients,

$$\mathscr{P}(\beta) \leq \delta. \tag{3}$$

ALM-III establishes the equivalence of these two procedures. We explore the geometry of the alternative procedure. In penalized regression, we do not have a good reason for choosing any particular δ in (3), so we look at all possible values of δ or, more often and equivalently, all possible values of k in (1).

The restricted least squares problem of minimizing (2) subject to the inequality constraint (3) lends itself to a geometric interpretation. Our discussion is reasonably general but most illustrations are of the lasso in two dimensions. For simplicity, we examine a standard linear model $Y = X\beta + e$ but in practice penalized regression is always applied to some version of the partitioned model $Y = J\beta_0 + Z\beta_* + e$ where β_0 is not penalized and the predictors in Z have often been adjusted for their means or rescaled.

To explore the geometry, we want to facilitate our ability to create contour maps of the least squares criterion surface as a function of β. Using the decomposition of $\|Y - X\beta\|^2$ given in (8.1.3), we rewrite (2) in terms of a quadratic function of β minimized at the least squares estimate $\hat{\beta}$ plus a constant. The first term of the last line in (8.1.3) is the constant that does not depend on β and the second term, since it is a quadratic function, has contours that are ellipsoids in β centered at $\hat{\beta}$. The function minimum is at $\hat{\beta}$, for which the function value is *SSE*. The contours are ellipsoids in β for which

$$SSE + (\beta - \hat{\beta})'X'X(\beta - \hat{\beta}) = D$$

for some D. As D gets larger, the contours get farther from $\hat{\beta}$. The geometry of ellipsoids is discussed more in *ALM-III*, Subsection 14.1.3. The shape of an ellipsoid is determined by the eigenvalues and eigenvectors of $X'X$ – the major axis is in the direction of the eigenvector with the largest eigenvalue and is proportional in length to the square root of the eigenvalue. Note that, with multivariate normal data, each ellipsoid is also the confidence region for β corresponding to some confidence level. The least squares estimate subject to the constraint (3) is a β vector on the smallest elliptical contour that intersects the region defined by (3). Of course if the least squares estimates already satisfy (3), there is nothing more to find.

If the least squares estimate does not already satisfy (3), a multiple regression lasso that penalizes the intercept minimizes

$$SSE + (\beta - \hat{\beta})'X'X(\beta - \hat{\beta})$$

subject to

$$\sum_{j=0}^{p-1} |\beta_j| = \delta.$$

We need a β vector on the smallest elliptical contour that intersects the region $\sum_{j=0}^{p-1} |\beta_j| = \delta$. Where that intersection occurs depends on the value of $\hat{\beta}$, the orientation of the ellipsoid, and the size of δ.

For $y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \varepsilon_i$, the lasso penalty constraint $|\beta_1| + |\beta_2| \leq \delta$ is a square (diamond) centered at (0,0) with diameter 2δ. To find the lasso estimate, grow the

ellipses centered at $\hat{\beta}$ until they just touch the edge of the square. The point of contact is the lasso estimate, i.e., the point that has the minimum value of the least squares criterion (2) subject to the penalty constraint (3). The point of contact can either be on the face of the square, as illustrated in Figure 8.4, or it can be a corner of the square as in Figure 8.5. When the contact is on a corner, one of the regression estimates has been zeroed out. In Figure 8.5, $\delta = 1$, the lasso estimate of β_1 is 0 and the lasso estimate of β_2 is 1. For classical ridge regression, the diamonds in the two figures are replaced by circles of radius 1. Using a circle would definitely change the point of contact in Figure 8.4 and almost certainly change the point of contact in Figure 8.5.

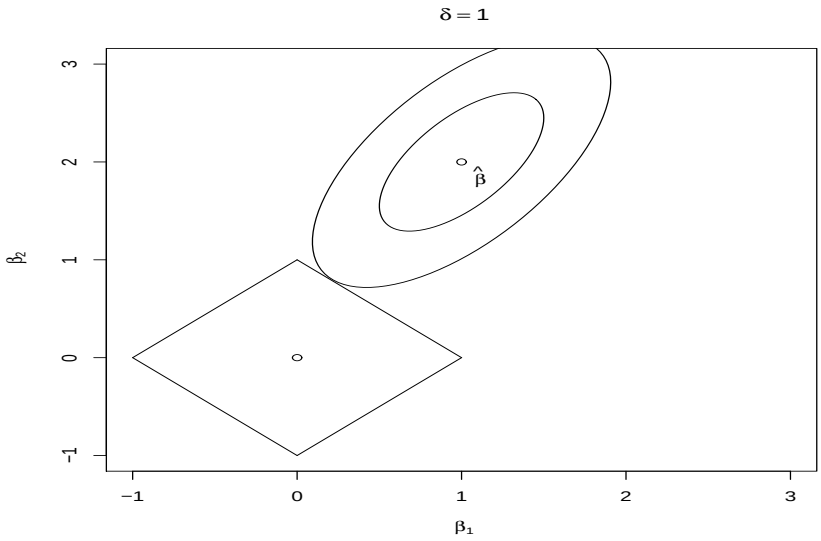

Fig. 8.4 Lasso shrinkage without variable selection.

8.4.0.1 More lasso geometry

In two dimensions the lasso estimate feels easy to find. With $\delta = 1$ and $\hat{\beta}$ in the first quadrant like it is in the two figures, the lasso estimate feels like it should be $(1,0)'$ or $(0,1)'$ or it should be the least squares estimate subject to the linear constraint $\beta_1 + \beta_2 = 1$. Finding the least squares estimate subject to a linear equality constraint is straightforward (although beyond the scope of this book). Figure 8.6 shows that it is possible for the lasso estimate of β_1 to be negative even when the least squares estimate is positive. And things get much more complicated in higher dimensions.

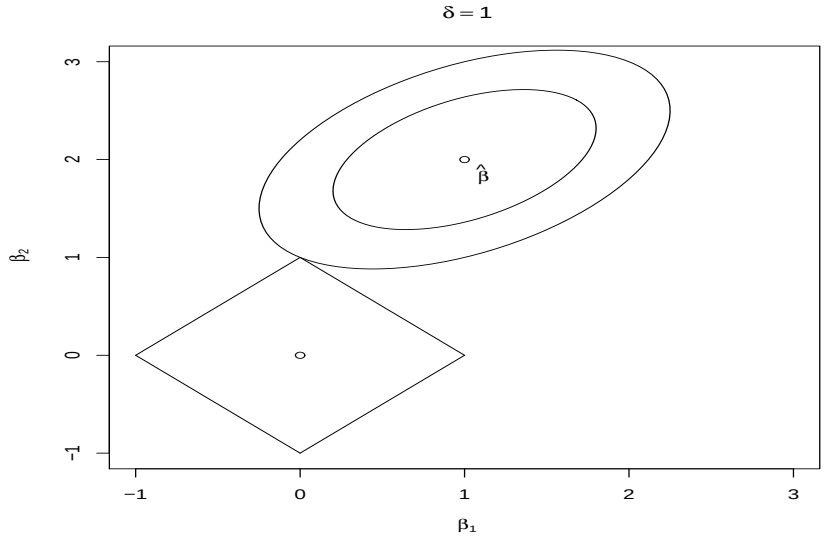

Fig. 8.5 Lasso shrinkage and variable selection.

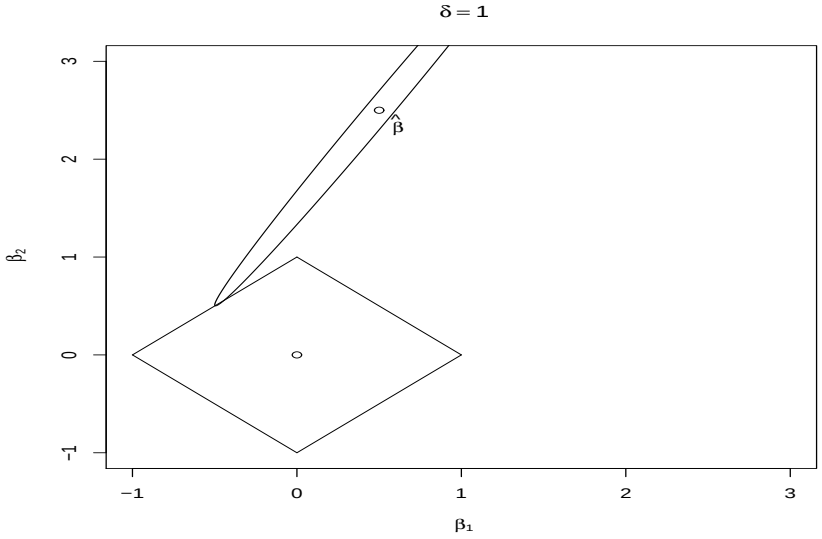

Fig. 8.6 Lasso sign change.

Figures 8.7 through 8.9 illustrate the geometry behind a trace plot. Remember that varying δ is equivalent to varying k, although the exact relationship is not simple. With both least squares $\hat{\beta}_j$s positive as in Figure 8.7 and with δ large enough, the lasso estimate is just the least squares estimate constrained to be on $\beta_1 + \beta_2 = \delta$, unless δ is big enough that $\hat{\beta}_1 + \hat{\beta}_2 \leq \delta$ in which case least squares is lasso. Figure 8.8 has smaller δs than Figure 8.7 but both plots in its top row have the same δ with the second plot being a closeup. The bottom row of Figure 8.8 shows that as δ decreases, the penalized estimates remain on $\hat{\beta}_1 + \hat{\beta}_2 = \delta$ until β_1 becomes 0. The top row of Figure 8.9 has the lasso estimate of β_1 zeroed out for two additional δs. The bottom row shows the lasso estimate becoming negative.

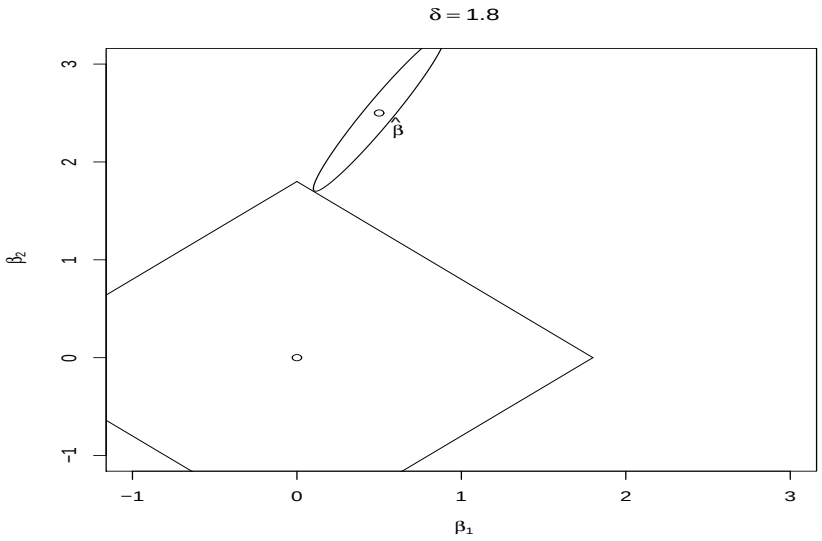

Fig. 8.7 Lasso trace $\delta = 1.8$.

In three dimensions the lasso geometry is that of throwing an American football (or a rugby ball) at an octohedron. Technically, the football should have someone sitting on it and, instead of throwing the football, we should blow it up until it hits the octohedron. The squashed football denotes the ellipsoids of the least squares criterion. The octohedron, see Figure 8.10, is the lasso penalty region and should be centered at 0. The octohedron has 8 sides consisting of isosceles triangles, 12 edges between the sides, and 6 corners. The football can hit any of these 26 features. If we knew which of the 26 features the ellipsoid was hitting, it would be easy to find the restricted least squares estimate because it would be the least squares estimate subject to a set of linear constraints. The problem is knowing which of the 26 features the ellipsoid is hitting. If the ellipsoid hits a corner, two regression estimates are zeroed out and the third takes a value $\pm\delta$. If it hits an edge, one

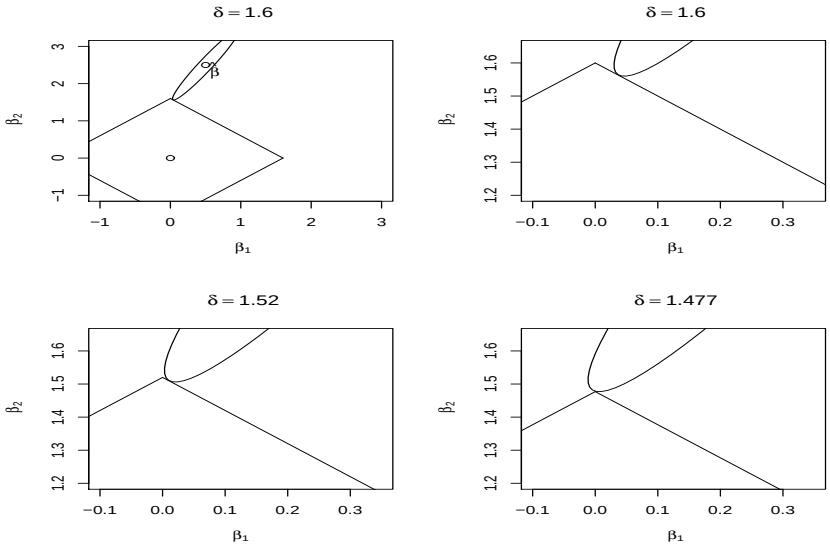

Fig. 8.8 Lasso trace.

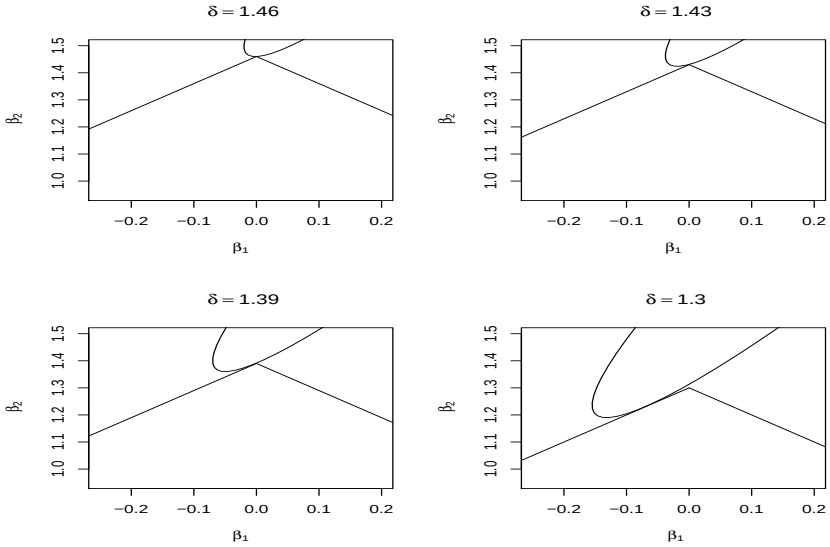

Fig. 8.9 Lasso trace.

estimate is zeroed out and the other two are shrunk towards but not to 0. If it hits a surface, no estimates are zeroed but all are shrunk.

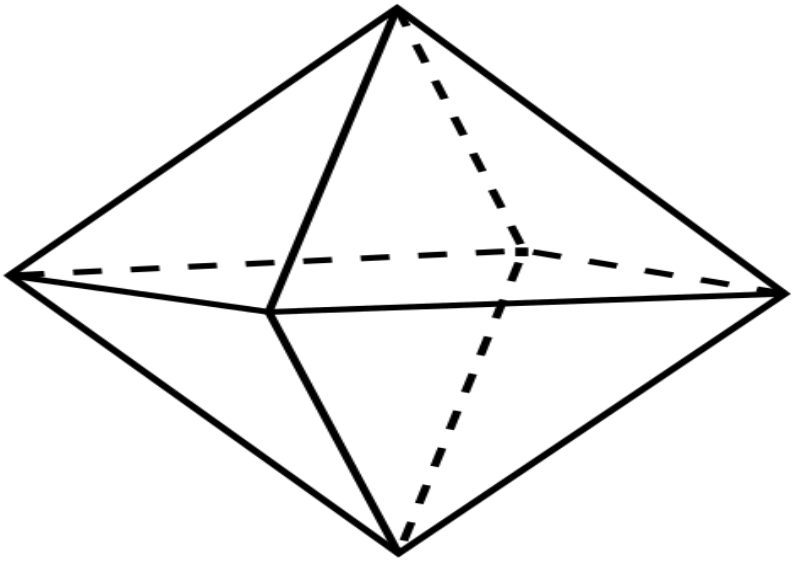

Fig. 8.10 Octohedron.

Typically, if you make δ big enough $\mathscr{P}(\hat{\beta}) \leq \delta$, so the penalty function has no effect on the least squares estimates. As soon as $\delta < \mathscr{P}(\hat{\beta})$, penalized least squares should be different from least squares. For the lasso, if all the elements of $\hat{\beta}$ are positive and δ is below, but sufficiently close to, $\sum_{j=0}^{p-1} |\hat{\beta}_j|$, the lasso estimate equals the least squares estimate subject to the linear constraint $\sum_{j=0}^{p-1} \beta_j = \delta$. More generally, the pattern of positive and negative values in $\hat{\beta}$ determines the pattern of positive and negative values in the linear constraint, i.e., $\sum_{j=0}^{p-1} \text{sign}(\hat{\beta}_j)\beta_j = \delta$. As you continue to decrease δ, the penalized least squares estimates $\tilde{\beta}$ gradually change, continuing to satisfy the constraint $\sum_{j=0}^{p-1} \text{sign}(\hat{\beta}_j)\beta_j = \delta$ until for some δ the estimates satisfies an additional linear constraint associated with some $\sum_{j=0}^{p-1} \pm\tilde{\beta}_j = \delta$, a constraint that changes only precisely one coefficient sign from the original constraint, so that together they cause the coefficient with the sign change to be zero. As δ further decreases, typically both linear constraints continue to hold for a while and then it is possible that the first linear constraint is supplanted by the second one.

It is convenient to think about the estimates moving along the surface of the penalty region (diamond, octahedron, etc.) as δ changes but that is not quite true because the surface itself changes with δ. Yet it is clear that features of the surface (diamond:4 edges and 4 corners, octohedron: 26 features) are comparable for all δ.

8.5 Two Other Penalty Functions

Other approaches to regularization replace k with more flexible options, minimizing

$$(Y - X\beta)'(Y - X\beta) + \mathscr{P}_\theta(\beta),$$

where a tuning parameter vector θ helps define the penalty function $\mathscr{P}_\theta(\beta)$.

The *elastic net* penalty combines the ridge and lasso penalties but incorporates another tuning parameter $\alpha \in [0,1]$,

$$\mathscr{P}_{EN}(\beta) \equiv \alpha\mathscr{P}_R(\beta) + (1-\alpha)\mathscr{P}_L(\beta) = \alpha\sum_{j=1}^{p-1}\beta_j^2 + (1-\alpha)\sum_{j=1}^{p-1}|\beta_j|.$$

Thus $\mathscr{P}_\theta(\beta) = k\mathscr{P}_{EN}(\beta)$ with $\theta = (k,\alpha)'$.

Fan and Li (2001) suggest the *scad (smoothly clipped absolute deviation)* penalty. The advantage of scad is that, like the lasso, it shrinks small estimates to zero but unlike lasso, it does not shrink large estimates at all. The penalty function is

$$\mathscr{P}_\theta(\beta) = k\mathscr{P}_S(\beta), \qquad \mathscr{P}_S(\beta) \equiv \sum_{j=1}^{p-1} P_S(\beta_j),$$

where for $a > 2$,

$$P_S(\beta_j) \equiv \begin{cases} |\beta_j| & \text{if } |\beta_j| \le k, \\ -\left(\frac{|\beta_j|^2 - 2ak|\beta_j| + k^2}{2(a-1)k}\right) & \text{if } k < |\beta_j| \le ak, \\ \frac{(a+1)k}{2} & |\beta_j| > ak. \end{cases}$$

The scad penalty function depends on $\theta = (k,a)'$. Fan and Li suggest that $a = 3.7$ often works well.

When the columns of X are orthonormal, it can be shown that scad results in the following modifications to the least squares estimates $\hat\beta_j$,

$$\tilde\beta_{Sj} = \begin{cases} 0, & \text{if } |\hat\beta_j| \le k, \\ \hat\beta_j - \text{sign}(\hat\beta_j)\,k, & \text{if } k < |\hat\beta_j| \le 2k, \\ \frac{(a-1)\hat\beta_j - \text{sign}(\hat\beta_j)ak}{a-2}, & \text{if } 2k < |\hat\beta_j| \le ak, \\ \hat\beta_j, & \text{if } |\hat\beta_j| > ak. \end{cases}$$

A similar result for lasso can be obtained by doubling the lasso tuning parameter,

$$\tilde\beta_{Lj} = \begin{cases} 0, & \text{if } |\hat\beta_j| < k, \\ \hat\beta_j - \text{sign}(\hat\beta_j)\,k, & \text{if } |\hat\beta_j| \ge k. \end{cases}$$

The estimates agree for $|\hat\beta_j| \le 2k$ but scad does less shrinkage on larger least squares estimates.

Chapter 9
Classification

Abstract Classification seems to be the statistical learning/machine learning/data science term for regression when the dependent variable is a 0-1 indicator variable that denotes inclusion in a group. Traditionally, classification was used as an alternative name for problems that are still referred to as discriminant analysis. The data collection scheme for discriminant analysis is very different from that of regression analysis and the data collection induces important differences in how the data should be analyzed. This chapter examines regression on 0-1 data; in particular binomial/binary regression and support vector machines (*SVM*s). The next chapter examines discriminant analysis and how it differs from regression on 0-1 data.

These days introductions to regression typically include not only standard regression but logistic regression as well, e.g., Christensen (2015). It is my expectation that logistic regression would be covered in a first course in regression, so its details need not be addressed here. As this book relates to a second course in regression, this chapter discusses the wider topic of binomial regression, introducing topics such as probit and complimentary log-log regression. But the real reason for the existence of this chapter is to examine support vector machines.

The general prediction/regression problem (as discussed in *PA* Section 6.3) considers the problem of predicting a random variable y from, say, a $d-1$ dimensional random vector $\mathbf{x}$. Let $f(\mathbf{x})$ be a predictor. Its expected prediction loss is $\mathrm{E}\{\mathscr{L}[y, f(\mathbf{x})]\}$, where $\mathscr{L}[\cdot,\cdot]$ is some predictive loss function and the expected value is taken with respect to both y and $\mathbf{x}$. Good predictor functions f have small expected loss values.

We now focus on the special case in which the dependent variable y takes only the values 0 and 1. (We will use p to denote the probability of a 1, which is why we now use d to label the number of predictor variables.) 0-1 random variables are *Bernoulli random variables*, which suggests that this be called Bernoulli prediction. Another reasonable name seems like Boolean prediction. But my friends seem to like "binary" so I will stick with binary prediction/regression (rather than classification).

As shown in *PA* Section 6.3, when using the squared error prediction loss function $\mathscr{L}[y,u] = [y-u]^2$, the best predictor is

$$p(\mathbf{x}) \equiv \mathrm{E}(y|\mathbf{x}),$$

which is also the probability of getting a 1 (success) conditional on $\mathbf{x}$. The Hamming loss function is 0 if the prediction equals y and 1 otherwise. Under *Hamming prediction loss* the best predictor is 0 when $p(\mathbf{x}) < 0.5$ and 1 when $p(\mathbf{x}) > 0.5$. In both cases it is incumbent upon us to obtain a good estimate of $p(\mathbf{x})$.

Regression analysis and discrimination involve different data collection schemes. Regression collects independent observations from the joint distribution of $(y,\mathbf{x}')$. Aldrich (2005) suggests that it was Fisher who first argued that regression estimation should condition on the predictor variables $\mathbf{x}$. This chapter examines how to estimate $p(\mathbf{x})$ from the conditional distribution of y given $\mathbf{x}$. For this we need only assume that the observations are conditionally independent and that $y \sim \mathrm{Bin}[1, p(\mathbf{x})] \equiv \mathrm{Bern}[p(\mathbf{x})]$. The last section of the next chapter considers estimates of $p(\mathbf{x})$ derived from sampling the conditional distribution of $\mathbf{x}$ given y. The end of this chapter examines support vector machines. These perform Hamming prediction without explicitly estimating $p(\mathbf{x})$.

Throughout we will explicitly incorporate ideas on penalized estimates. Implicit throughout is that the models can exploit nonparametric linear structures.

We begin with the binomial regression problem that most generalized linear model computer programs are written to handle.

9.1 Binomial Regression

Suppose there are a number of independent observations with $y_h \sim \mathrm{Bin}[1, p(\mathbf{x}_h)]$. Often such data get reported only as the total number of successes for each vector of predictor variables. In such cases, we implicitly reindex the original data as

$$(y_{ij}, \mathbf{x}_i'), \quad i = 1,\ldots,n,\ j = 1,\ldots,N_i$$

so that the reported data are

$$(y_{i\cdot}, \mathbf{x}_i'), \quad i = 1,\ldots,n, \quad \text{where } y_{i\cdot} \equiv \sum_{j=1}^{N_i} y_{ij}\ .$$

We now have independent binomial random variables

$$N_i \bar{y}_{i\cdot} \equiv y_{i\cdot} \sim \mathrm{Bin}[N_i, p(\mathbf{x}_i)]; \qquad i = 1,\ldots,n,$$

where the binomial proportions $\bar{y}_{i\cdot}$ are between 0 and 1. It is common practice to write binomial generalized linear model computer programs using the binomial proportions as the input data and specifying the N_is as weights. Obviously such

programs can also handle the original binary data $(y_h, \mathbf{x}_h')$ by writing $h = 1, \ldots, n$ but with $N_h = 1$ for all h. *In conformance with such programs, we write*

$$y_i \equiv \bar{y}_{i\cdot}$$

for the rest of this section.

The likelihood function for independent data with $N_i y_i \sim \text{Bin}[N_i, p(\mathbf{x}_i)]$ is

$$L[p(\cdot)] \equiv \prod_{i=1}^{n} \binom{N_i}{N_i y_i} [p(\mathbf{x}_i)]^{N_i y_i} [1 - p(\mathbf{x}_i)]^{N_i - N_i y_i}.$$

The *deviance* is defined as -2 times the log-likelihood so

$$\begin{aligned}
&\mathscr{D}[p(\cdot)] \\
&\equiv -2 \sum_{i=1}^{n} \{N_i y_i \log[p(\mathbf{x}_i)] + (N_i - N_i y_i) \log[1 - p(\mathbf{x}_i)]\} - 2 \sum_{i=1}^{n} \log\left[\binom{N_i}{N_i y_i}\right] \\
&= \sum_{i=1}^{n} -2N_i \{y_i \log[p(\mathbf{x}_i)] + (1 - y_i) \log[1 - p(\mathbf{x}_i)]\} - 2 \sum_{i=1}^{n} \log\left[\binom{N_i}{N_i y_i}\right].
\end{aligned}$$

A maximum likelihood estimate of $p(\cdot)$ maximizes the likelihood or, equivalently, minimizes the deviance. To simplify notation denote the constant term in the deviance

$$K \equiv -2 \sum_{i=1}^{n} \log\left[\binom{N_i}{N_i y_i}\right].$$

The constant term has no effect on estimation. For binary regression models in which $N_i \equiv 1$ so that y_i is 1 or 0, the constant term in the deviance vanishes and only one of the two terms in the braces actually applies. Either y_i or $1 - y_i$ has to be zero, so one of the terms in the braces always gets multiplied by 0.

If the function $p(\mathbf{x})$ is known except for some unknown parameter vector θ, write $p(\mathbf{x}; \theta)$. The maximum likelihood estimate of θ maximizes the parameterized likelihood

$$L(\theta) \equiv \prod_{i=1}^{n} \binom{N_i}{N_i y_i} [p(\mathbf{x}_i; \theta)]^{N_i y_i} [1 - p(\mathbf{x}_i; \theta)]^{N_i - N_i y_i}$$

or minimizes the parameterized deviance

$$\mathscr{D}(\theta) \equiv \sum_{i=1}^{n} -2N_i \{y_i \log[p(\mathbf{x}_i; \theta)] + (1 - y_i) \log[1 - p(\mathbf{x}_i; \theta)]\} + K.$$

Henceforth, we take x to be a d vector that includes all explanatory variables. In most cases $x' = (1, \mathbf{x}')$.

Binomial generalized linear models typically specify that the conditional probability is a known function of $x'\beta$. In particular,

$$p(\mathbf{x}) \equiv p(x) = F(x'\beta)$$

for some known cumulative distribution function (cdf) F for which the inverse function F^{-1} exists. F is a cdf so that the real valued term $x'\beta$ is transformed into a number between 0 and 1. The inverse function is called a *link* function and is used to isolate the linear structure $x'\beta$ of the model, i.e.,

$$F^{-1}[p(x)] = x'\beta.$$

The most common choices for F are the standard versions of the *logistic*, normal, and *Gumbel (minimum)* distributions. With $\Phi(\cdot)$ denoting the cdf for a $N(0,1)$ random variable,

$$p(x) = F(x'\beta) = \begin{cases} e^{x'\beta} \Big/ \left[1 + e^{x'\beta}\right] & \text{Logistic} \\ \Phi(x'\beta) & \text{Normal} \\ 1 - \exp\left[-e^{x'\beta}\right] & \text{Gumbel.} \end{cases}$$

Most often the procedures are referred to by the names of the inverse functions rather than the names of the original cdfs:

$$x'\beta = F^{-1}[p(x)] = \begin{cases} \log\left\{p(x) \big/ [1 - p(x)]\right\} & \text{Logit} \\ \Phi^{-1}[p(x)] & \text{Probit} \\ \log\left\{-\log[1 - p(x)]\right\} & \text{Complementary log-log.} \end{cases}$$

In the case of logit/logistic models, logit often refers to ANOVA type models and logistic is often used for regression models. I use the terms interchangeably but prefer calling them logit models. (In this chapter, all references to "generalized linear models" refer to the subclass of binomial generalized linear models defined using an inverse cdf link.)

In any case, the likelihood function for such data is

$$L_F(\beta) \equiv \prod_{i=1}^{n} \binom{N_i}{N_i y_i} [F(x_i'\beta)]^{N_i y_i} [1 - F(x_i'\beta)]^{N_i - N_i y_i}$$

and the deviance is

$$\mathscr{D}_F(\beta) \equiv \sum_{i=1}^{n} -2N_i \left\{ y_i \log\left[F(x_i'\beta)\right] + (1 - y_i) \log\left[1 - F(x_i'\beta)\right] \right\} + K. \qquad (1)$$

As always, the constant term K in the deviance is irrelevant to the estimation of β.

Note that minimum deviance (maximum likelihood) estimation fits into the pattern discussed in Section 4.4 of estimating β by defining weights $w_i > 0$ and a loss function $\mathscr{L}(y,u)$ and minimizing

$$\sum_{i=1}^{n} w_i \mathscr{L}(y_i, x_i'\beta).$$

Here the weights are $w_i = N_i$ and the loss function is

$$\mathscr{L}(y,u) = -2\{y\log[F(u)] + (1-y)\log[1-F(u)]\}.$$

In later sections we will see that the loss function is easier to interpret for binary data and that *support vector machines* use a very similar procedure to estimate β.

In analogy to penalized least squares estimates, we can form *penalized minimum deviance (penalized maximum likelihood)* estimates that minimize

$$\mathscr{D}_F(\beta) + k\,\mathscr{P}(\beta).$$

Typically, we use the same penalty functions $\mathscr{P}(\beta)$ as discussed in Chapter 8 for penalized least squares. For a multiple regression with $x_i'\beta \equiv \beta_0 + \sum_{j=1}^{d-1}\beta_j x_{ij}$, write $\beta_* \equiv (\beta_1,\ldots,\beta_{d-1})'$. As with standard regression, we typically would not penalize the intercept. By choosing

$$\mathscr{P}_L(\beta) \equiv \sum_{j=1}^{d-1}|\beta_j| = \|\beta_*\|_1,$$

we get lasso binomial regression. By choosing

$$\mathscr{P}_R(\beta) \equiv \beta_*'\beta_* = \sum_{j=1}^{d-1}\beta_j^2$$

we get one form of ridge binomial regression. Elastic net binomial regression is obtained by using

$$\mathscr{P}_{EN}(\beta) \equiv \alpha\mathscr{P}_R(\beta) + (1-\alpha)\mathscr{P}_L(\beta)$$

with an additional tuning parameter $\alpha \in [0,1]$. As mentioned in Chapter 8, these penalty functions penalize each coefficient the same amount, so typically one would *standardize the predictor variables to a common length* before applying such a penalty. (The penalization ideas apply to all generalized linear models, not just these binomial generalized linear models, and are also fundamental to support vector machines.)

It is a simple matter to generalize the penalized estimation ideas to a partitioned model,

$$F^{-1}[p(x_i,z_i)] = x_i'\beta + z_i'\gamma$$

where x_i and z_i are known vectors of predictors, β and γ are unknown parameters, and where we only penalize γ. (*ALM-III*, Chapter 2 focuses on penalized estimation for partitioned linear models.)

9.1.1 Data Augmentation Regression

For linear ridge regression we established in Section 8.2 that the ridge estimates could be obtained by fitting augmented linear models. We now define an analogous augmented binomial regression model and infer the penalty function that it implicitly incorporates. (The penalty is not the traditional ridge penalty.) Although ridge regression requires no assumption of normality, the analogies between standard regression and binomial regression will be clearer making it.

Model (8.3.2) is an augmented, partitioned linear model that provides generalized ridge estimates. With d predictor variables (rather than the notation p used in Chapter 8) and $q = J_{d-1}$, model (8.3.2) provides standard ridge estimates. In particular, it treats the augmented observations 0 as observations on independent random variables $\tilde{y}_j$, $j = 1,\ldots,d-1$ with the distribution $\tilde{y}_j \sim N(\gamma_j, \sigma^2/k)$. The model involves finding weighted least squares estimates. The vector of weights becomes $w \equiv [J_n', kJ_{d-1}']'$.

Data augmentation binomial regression takes $d-1$ augmenting observations as $\tilde{y}_j = F(0)$ and treats them as independent with $k\tilde{y}_j \sim \text{Bin}[k, F(\beta_j)]$. For logit and probit models $\tilde{y}_j = 0.5$. To analyze such data you need software that is coded in enough generality that it permits analysis on binomials with non-integer numbers of trials. An augmented observation $\tilde{y}_j = F(0)$ comes from a case with probability $F(\beta_j)$ so it forces β_j towards 0. The parameter k determines how many Bernoulli trials $\tilde{y}_j$ corresponds to, so it determines how much β_j gets forced towards 0. These augmented data define the same augmented model matrix as in (8.3.2). Model (8.3.2) augments the data Y with a string of 0s but instead we augment Y into $[Y', F(0)J_{d-1}']'$. The weight vector we need for the augmented binomial model is exactly the same as the weight vector for model (8.3.2).

The penalty associated with this procedure is defined by what the augmenting observations add to the deviance function. Ignoring the constant term that the augmented data add to the deviance, it is not hard to see that the penalty function, say, $\mathscr{P}_{R2}(\beta)$ is defined via

$$k\mathscr{P}_{R2}(\beta) \equiv k\sum_{j=1}^{d-1} -2\left\{F(0)\log[F(\beta_j)] + [1-F(0)]\log[1-F(\beta_j)]\right\}.$$

9.2 Binary Prediction

Henceforth we use binary data ($N_i = 1$) to make binary predictions. Again, in the machine learning community, binary prediction is called *classification*, cf. Hastie et al. (2016). I cannot overemphasize that when performing this activity, there is an important distinction to be made over how the data were collected. In regression problems, a sample is taken from a population and individuals randomly fall into a group: 0 or 1. In the discrimination problems considered in Chapter 10, data are ran-

domly sampled from each group separately. The term "classification" has traditionally been associated with discrimination problems, but maintaining that distinction is surely a losing battle.

One of the beauties of using the binomial/binary generalized linear models in Section 1 for binary prediction of regression data is that they provide estimated probabilities for belonging in the two groups. Having good estimates of $p(x) \equiv \mathrm{E}(y|x)$ helps in making good predictions for any prediction loss: squared error, absolute error, even Hamming. However, under Hamming prediction loss, what that matters is estimating whether $p(x) > 0.5$ or $p(x) < 0.5$ for all x. Hamming loss only cares whether cases get assigned to the correct group.

Our binomial regression estimates of $p(x)$ leads to linear prediction rules. A *linear prediction rule* amounts to defining a hyperplane of x vectors and predicting that points on one side of the hyperplane will be a 1 and points on the other side will be a 0. In Section 4 we consider a wider class of linear prediction rules that merely assign cases to groups without actually estimating the probability function. The motivation for this will be by analogy to the estimation methods for generalized linear models discussed in Section 3. These linear prediction rules include the support vector machines considered in Section 5 and they seem to implicitly assume that the data are regression data rather than discrimination data. Section 6 looks at how to estimate $p(x)$ from the best predictor associated with a specific loss function.

EXAMPLE 9.2.1. Aitchison and Dunsmore (1975) present data on Cushing's Syndrome, a medical condition characterized by overproduction of cortisol by the adrenal cortex. Individuals were identified as belonging to one of three types: *adenoma*, *bilateral hyperplasia*, or *carcinoma*. The amounts of tetrahydrocortisone and pregnanetriol excreted in their urine were measured. Table 9.1, presents the data for twenty-one cases. To illustrate binary prediction, we restrict our attention to the 15 cases that are bilateral hyperplasia or carcinoma. Following Aitchison and Dunsmore (1975), the analysis is performed on the logarithms of the predictor variables. Figure 9.1 plots the 15 points and includes three linear prediction rules: logistic regression, probit regression and a support vector machine. Points above a line are classified as carcinoma and points below a line are identified as bilateral hyperplasia. To anthropomorphize, the generalized linear models seem to care more about not getting any point too badly wrong. The SVM almost seems like if it cannot get that one bilateral point correctly classified, it doesn't care how far that point is from the line. (Indeed, even if the SVM could get that bilateral point correctly classified, if ignoring it will get the fitted line far enough away from all the other points, the SVM would still largely ignore the misclassified point. For more on this, see the artificially simple example in `http://www.stat.unm.edu/~fletcher/R-SL.pdf`.) Christensen (1997, Section 4.7) and Ripley (1996, Section 2.4) discuss the complete three group data.

It is not clear whether the Cushing Syndrome data are regression data or discrimination data. If regression data, someone would have sampled 21 Cushing's Syndrome patients who fell into the categories: 6 adenoma, 10 bilateral hyperplasia, 5 carcinoma. If discrimination data, someone decided to sample 6 adenoma patients,

Table 9.1 Cushing's Syndrome data.

Case	Type	TETRA	PREG	Case	Type	TETRA	PREG
1	A	3.1	11.70	12	B	15.4	3.60
2	A	3.0	1.30	13	B	7.7	1.60
3	A	1.9	0.10	14	B	6.5	0.40
4	A	3.8	0.04	15	B	5.7	0.40
5	A	4.1	1.10	16	B	13.6	1.60
6	A	1.9	0.40	17	C	10.2	6.40
7	B	8.3	1.00	18	C	9.2	7.90
8	B	3.8	0.20	19	C	9.6	3.10
9	B	3.9	0.60	20	C	53.8	2.50
10	B	7.8	1.20	21	C	15.8	7.60
11	B	9.1	0.60				

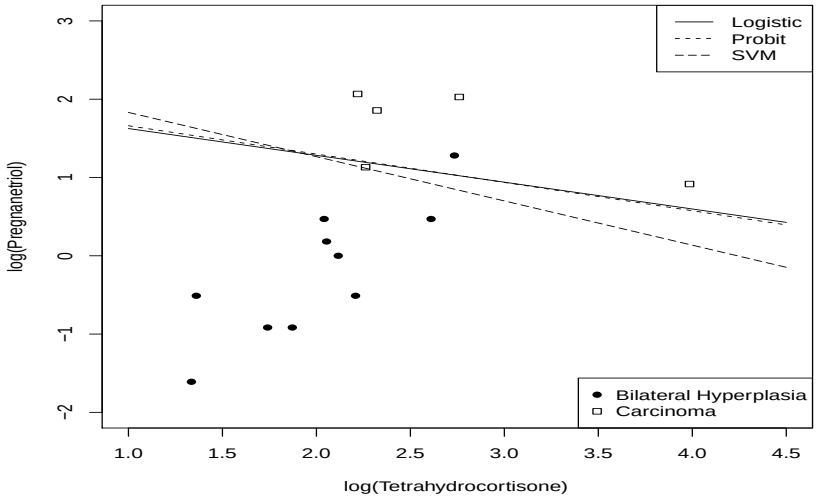

Fig. 9.1 Logistic regression, probit regression, and an SVM: Cushing's Syndrome data (subset).

10 bilateral hyperplasia patients, and 5 carcinoma patients. We assume the former in this chapter. We assume the latter in Chapter 10. For discrimination data, the generalized linear model methods of the next section require the adjustments discussed at the end Chapter 10 before they will make proper predictions. Linear prediction methods that are not closely associated with estimating $p(x)$ have shakier justifications when used for discrimination data because they do not lend themselves to the adjustments that are clearly needed for generalized linear models. The discrimination methods of Chapter 10 *are* closely associated with estimating $p(x)$ but they do it indirectly by estimating the density of the predictor variables given the group. □

9.3 Binary Generalized Linear Model Estimation

For binary data the deviance in (9.1.1) reduces to

$$\mathscr{D}_F(\beta) \equiv \sum_{i=1}^{n} -2\left\{y_i \log\left[F(x_i'\beta)\right] + (1-y_i)\log\left[1-F(x_i'\beta)\right]\right\}. \tag{1}$$

Again, minimum deviance (maximum likelihood) estimation fits into the pattern discussed in Section 4.4 of estimating β by defining a loss function $\mathscr{L}(y,u)$ and weights $w_i > 0$ and then minimizing

$$\sum_{i=1}^{n} w_i \mathscr{L}(y_i, x_i'\beta).$$

For binary generalized linear models the weights are all 1 and, as before, the loss function is

$$\mathscr{L}_F(y,u) = -2\{y\log[F(u)] + (1-y)\log[1-F(u)]\}$$

but now, because of the binary nature of the data, we can write

$$\mathscr{L}_F(y,u) = \begin{cases} -2\log[F(u)] & \text{if } y = 1 \\ -2\log[1-F(u)] & \text{if } y = 0\,. \end{cases} \tag{2}$$

The logit and probit loss functions are plotted in Figure 9.2. The loss functions are quite similar, as were the prediction lines in Figure 9.1.

A penalized minimum deviance (penalized maximum likelihood) estimate is defined as in Section 1. It can be viewed as minimizing

$$\sum_{i=1}^{n} \mathscr{L}_F(y_i, x_i'\beta) + k\mathscr{P}(\beta).$$

(The artificial example in the R code document includes a data augmentation fit that is reasonably similar to the default SVM.)

9.4 Linear Prediction Rules

We now examine linear prediction rules in detail. First, that generalized linear models lead to linear prediction rules and then, that similar ideas can produce linear prediction rules without an explicit probability model. In Section 6 we will try to relate such rules back to probability models.

For the generalized linear models, the optimal Hamming loss predictor is 1 when $F(x'\beta) > 0.5$ and 0 when $F(x'\beta) < 0.5$. These conditions are equivalent to predicting 1 when $x'\beta > F^{-1}(0.5)$ and 0 when $x'\beta < F^{-1}(0.5)$. Thus the hyperplane of x

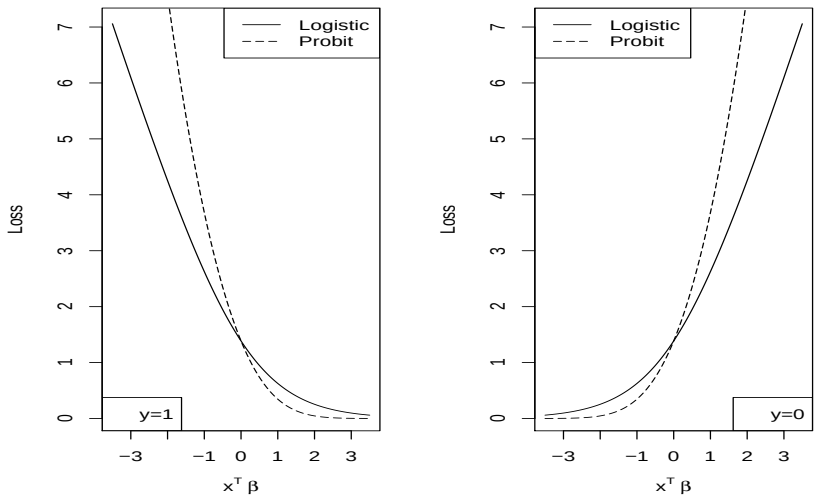

Fig. 9.2 Binary logistic regression and probit regression loss functions.

vectors that satisfy $x'\beta = F^{-1}(0.5)$ implicitly defines a linear prediction rule which is the optimal Hamming rule for the generalized linear model.

With $\hat{\beta}$ the minimum deviance estimate and $F(x'\hat{\beta})$ the estimated probability for group 1, the logistic and probit lines in Figure 9.1 were constructed by setting $F(x'\hat{\beta}) = 0.5$, i.e., $x'\hat{\beta} = F^{-1}(0.5) = 0$. (The last equality only holds when 0 is a median of F and always holds when F is symmetric about 0.) When $x'\hat{\beta} > 0$, the logistic and probit models have $F(x'\hat{\beta}) > 0.5$. When $x'\hat{\beta} < 0$, they have $F(x'\hat{\beta}) < 0.5$.

In a regression setting we typically have

$$x'\beta \equiv \beta_0 + \sum_{j=1}^{d-1} \beta_j x_j = \beta_0 + \mathbf{x}'\beta_*$$

where

$$\mathbf{x}' \equiv (x_1, \ldots, x_{d-1}).$$

The hyperplane $x'\hat{\beta} = F^{-1}(0.5)$ is the same creature as $\left[\beta_0 - F^{-1}(0.5)\right] + \mathbf{x}'\beta_* = 0$. As a function of the predictor variables in $\mathbf{x}$, the orientation of the hyperplane is determined by β_*. Hyperplanes with β_* vectors that are multiples of one another are parallel in $\mathbf{R}^{d-1}$.

Any hyperplane $x'\beta = 0$ can be used to predict binary outcomes using the rule: *if* $x'\beta > 0$ *the case is predicted as group 1 and if* $x'\beta < 0$ *the case is predicted as group 0.* Using the regression notation that means: group 1 if $-\beta_0 < \mathbf{x}'\beta_*$ and group 0 if $-\beta_0 > \mathbf{x}'\beta_*$. To use a hyperplane $x'\beta = C$ is simply to redefine β_0 as $\beta_0 - C$. If η is

a nonzero scalar multiple of β, the vectors x with $x'\beta = 0$ are precisely the same as the vectors x with $x'\eta = 0$, so β and η define the same linear predictor. (Although if the constant of proportionality is negative, the codes for the two groups will be reversed.)

Since any β determines a binary predictor, we can certainly pick one by minimizing

$$\sum_{i=1}^{n} \mathscr{L}(y_i, x_i'\beta) + k\mathscr{P}(\beta),$$

for any loss function $\mathscr{L}$, for any penalty function $\mathscr{P}$, and any tuning parameter k. The question is, "Will it be any good?" Certainly if we use the loss function associated with minimizing the deviance of a generalized linear model having $F(0) = 0.5$ and either take $k = 0$ or any reasonable penalty function with k small, the linear predictor will be reasonable.

It is an exercise in *ALM-III* to show that $x'\beta$ essentially measures the distance between x and the hyperplane defined by $x'\beta = 0$. In particular, one can show that $|x'\beta|$ is $\|\beta\|$ times the perpendicular distance from x to the prediction hyperplane (subspace) $\{x|x'\beta = 0\}$ by finding the perpendicular distance using $M_\beta \equiv \beta(\beta'\beta)^{-1}\beta'$.

Since the numerical value of $x'\beta$ essentially measures the distance of x from the hyperplane defined by $x'\beta = 0$, it should provide a measure of how clearly a case belongs to a group. Ideally, we would like to know $p(x)$. For the generalized linear model loss functions that is easy, $p(x) = F(x'\beta)$. For a general differentiable loss function, i.e. one not associated with a generalized linear model, we will probably need to rely on equation (9.6.2) to estimate probabilities. In the next section we will see that SVMs use a loss function that looks reasonable, but not one that is consistent with a generalized linear model nor is it differentiable, so there is no obvious method of turning an estimated SVM into group probabilities.

Like all linear models, the linearity of a linear prediction rule is linearity in the unknown regression coefficients, not in the originally measured predictor variables.

EXAMPLE 9.4.2. In Figure 9.1, the linear structure used for determining the logit and probit linear prediction rules was

$$x'\beta = \beta_0 + \beta_1 TL + \beta_2 PL.$$

where TL and PL are the logs of the tetrahydrocortisone and pregnanetriol scores. Figure 9.3 illustrates the use of the quadratic model

$$x'\beta = \beta_{00} + \beta_{10} TL + \beta_{01} PL + \beta_{20} TL^2 + \beta_{02} PL^2 + \beta_{11} TL \times PL.$$

The logistic and probit linear predictors $0 = x'\hat{\beta}$ are hyperplanes in 5 dimensions but take the form of parabolas when plotted in the original two dimensions. (In 5 dimensions a hyperplane would be ignoring the fact that, say, "PL^2" is not just the name of a variable but is actually the square of the "PL" variable.)

Even more than in Figure 9.1, the logistic and probit linear predictors in Figure 9.3 plot almost on top of one another. Unlike Figure 9.1, the parabolas com-

pletely separate the carcinoma cases from the bilateral hyperplacia cases. (More on this later.) Figure 9.3 also illustrates a quadratic support vector machine. Only one of the two SVM parabolic curves appears on this plot and the one that appears, over the range of this plot, is almost a straight line. The SVM fails to separate the two groups of observations. More details on the SVM linear predictor are given in the next section. □

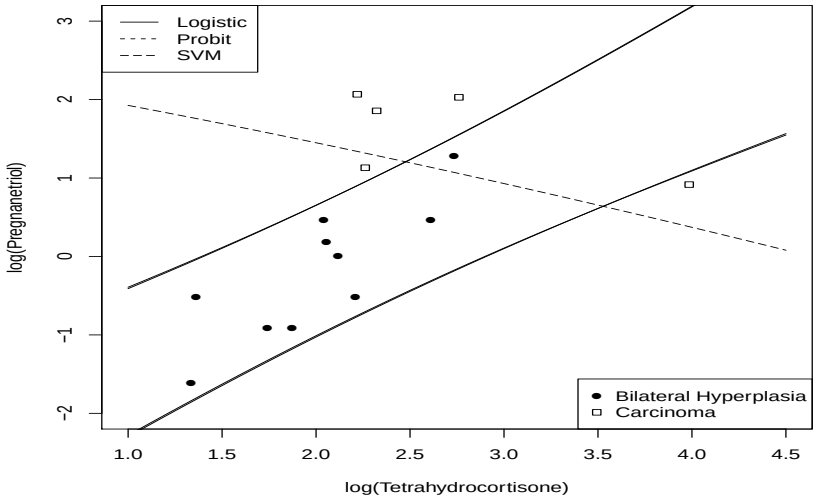

Fig. 9.3 Quadratic model logistic regression, probit regression (indistinguishable from logistic), and an SVM: Cushing's Syndrome data (subset).

9.4.1 Loss Functions

We have discussed the loss functions associated with binomial/binary generalized linear models. The loss function for support vector machines is discussed in the next section. The use of squared error loss is related to the normal theory discriminant analysis of Chapter 10 and is also discussed in the next subsection. In the machine learning community the use of squared error loss together with a penalty function is sometimes called the *proximal support vector machine*. Another loss function that gets used as an approximation to AdaBoost is

$$\mathscr{L}_{Ada}(y,u) = \begin{cases} e^{-u} & \text{if } y = 1 \\ e^{u} & \text{if } y = 0\,. \end{cases}$$

9.4.2 Least Squares Binary Prediction

Both *linear discriminant analysis (LDA)* and *quadratic discriminant analysis (QDA)*, as defined in the next chapter, satisfy the definition of an estimated linear predictor given here. In particular, there is a strong relationship between fitting a least squares regression model to binary y data and LDA, cf. Williams (1959). The algebra involved in the demonstration is tedious. (I have about four pages of formulae with no explanations of what I am doing.) Suffice it to say that if the number of successes equals the number of failures, LDA agrees with least squares regression on 0-1 data where a case is assigned to group 1 if and only if its predicted (fitted) value is greater than 0.5. Moreover, if the number of successes and failures are not equal, there exists a cut-off point for the least squares predicted values (typically different from 0.5) that will give the same predictions as LDA.

9.5 Support Vector Machines

Support vector machines are linear predictors that pick $\beta = (\beta_0, \beta_*')'$ by minimizing

$$\sum_{i=1}^{n} \mathscr{L}_S(y_i, x_i'\beta) + k\mathscr{P}_R(\beta).$$

The penalty function is the standard ridge regression penalty, $\mathscr{P}_R(\beta) \equiv \beta_*'\beta_*$, but most importantly the loss function is

$$\mathscr{L}_S(y,u) = \begin{cases} (1-u)_+ & \text{if } y = 1 \\ (1+u)_+ & \text{if } y = 0\,. \end{cases}$$

(Recall that $a_+ = a$ if a is positive and $a_+ = 0$ is a is not positive.) Figure 9.4 plots the logit and SVM loss functions. The SVM loss function is certainly reasonable.

EXAMPLE 9.5.2. In Figures 9.1 and 9.3, the SVM curves presented were the default linear and quadratic fits for unscaled predictor variables from the R library `e1071`'s program `svm`. Figure 9.5 is similar to 9.3 but plots the SVM when the tuning parameter k associated with the penalty function has been reduced by a factor of 100. The new fitted SVM is much more like the maximimum likelihood fits and it separates the two classes. □

As discussed in Section 4.4, least squares is all about minimizing a squared error loss function. Similarly, ridge and lasso regression problems are dominated by the problem of minimizing the squared error loss function subject to quadratic and linear inequality constraints, cf. Section 8.4. Somewhat ironically, programs for finding SVM estimates seem to focus on minimizing the quadratic penalty function subject to the constraints imposed by needing to minimize the loss function. The issue is less about whether the loss is more important than the penalty function and more about

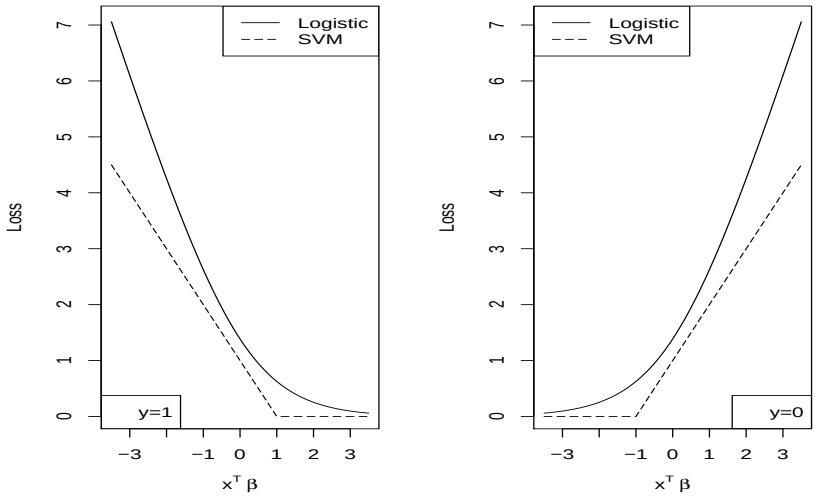

Fig. 9.4 Logistic regression and SVM loss functions.

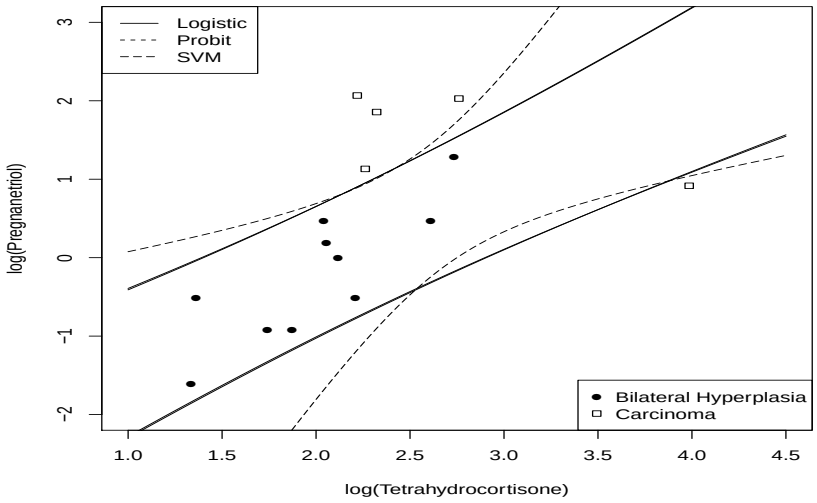

Fig. 9.5 Quadratic model logistic regression, probit regression, and an SVM with reduced tuning parameter: Cushing's Syndrome data (subset).

the highest order polynomial involved in the minimization. *ALM-III*, Appendix A.3 discusses the general problem of minimizing quadratic functions subject to linear inequality constraints and a subsection applies the general results to the SVM problem. In this chapter, we merely cite the most important of those results. Hastie et al. (2016), Zhu (2008), and Moguerza and Muñoz (2006) all present introductions to SVMs.

9.5.1 Probability Estimation

The value $|x'\beta|$, which is $\|\beta\|$ times the perpendicular distance from x to the prediction hyperplane, should measure the assuredness of a classification. The bigger the value, the more sure we should be of the classification. Unfortunately, for SVMs this does not obviously convert to a classification probability. First, the loss function associated with SVMs is similar to the logit and probit losses, so SVMs might be generalized linear models for some F. They are not, cf. *ALM-III*, so we cannot get probabilities associated with SVMs by appealing to generalized linear models. Second, the general equation for determining probabilities from a best predictor given in equation (9.6.2) does not apply because the SVM loss function is not differentiable everywhere. I am not aware of any way to get classification probabilities from SVMs.

9.5.2 Parameter Estimation

Finding the SVM parameter estimates is generally performed by turning the estimation problem into a quadratic optimization problem, cf. *ALM-III*, Appendix A.3.

Write our binary data in vector form as

$$Y \equiv \begin{bmatrix} Y_1 \\ Y_0 \end{bmatrix}$$

where N_1 successes are in $Y_1 \equiv J_{N_1}$ and $N_0 = n - N_1$ failures are in $Y_0 \equiv 0^1_{N_0}$. For this discussion only

$$J_1 \equiv J_{N_1}; \qquad J_0 \equiv J_{N_0}.$$

Similarly write the model matrix, which includes an intercept predictor, as

$$X \equiv \begin{bmatrix} X_1 \\ X_0 \end{bmatrix} \equiv \begin{bmatrix} J_1 & \mathbf{X}_1 \\ J_0 & \mathbf{X}_0 \end{bmatrix}.$$

Any n vector v may be written

$$v = \begin{bmatrix} v_1 \\ v_0 \end{bmatrix}$$

in conformance with Y_1 and Y_0.

Support vector machines pick $\beta = (\beta_0, \beta_*')'$ by minimizing

$$\sum_{h=1}^{n} \mathscr{L}_S(y_h, x_h'\beta) + k\beta_*'\beta_* \tag{1}$$

where

$$\mathscr{L}_S(y,u) = \begin{cases} (1-u)_+ & \text{if } y = 1 \\ (1+u)_+ & \text{if } y = 0\,. \end{cases}$$

We will leave the details of minimizing this to *ALM-III* but the innovative idea is to introduce slack variables $\xi = (\Xi_1, \ldots, \Xi_n)'$ that serve as upper bounds for the contributions to the loss function. It turns out that minimizing (1) is equivalent to finding

$$\inf_{\beta,\xi} \left(k\beta_*'\beta_* + \xi' J\right) \tag{2}$$

subject to

$$\mathscr{L}_S(y_h, x_h'\beta) \le \Xi_h, \quad h = 1, \ldots, n. \tag{3}$$

To establish this as a quadratic optimization problem, we need to replace the loss function constraints (3) with linear constraints. In matrix form these constraints turn out to be, for cases associated with Y_1

$$0_{N_1}^1 \le \xi_1; \qquad J_1 - X_1\beta \le \xi_1 \tag{3a}$$

where *an inequality applied to a matrix is understood to apply elementwise*. Similarly for Y_0 cases,

$$0_{N_0}^1 \le \xi_0; \qquad J_0 + X_0\beta \le \xi_0. \tag{3b}$$

In total there are $2n$ linear inequality constraints being imposed on the criterion function (2).

In matrix notation, rewrite the penalized loss function in standard form for quadratic optimization as

$$k\beta_*'\beta_* + \xi' J = \frac{1}{2} \begin{bmatrix} \beta_0 \\ \beta_* \\ \xi \end{bmatrix}' \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2kI & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_* \\ \xi \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ J_n \end{bmatrix}' \begin{bmatrix} \beta_0 \\ \beta_* \\ \xi \end{bmatrix}, \tag{4}$$

which is to be minimized subject to the constraints (3) rewritten in standard form as

$$\begin{bmatrix} -X_1 & -I & 0 \\ X_0 & 0 & -I \\ 0 & -I & 0 \\ 0 & 0 & -I \end{bmatrix} \begin{bmatrix} \beta \\ \xi_1 \\ \xi_0 \end{bmatrix} \le \begin{bmatrix} -J_1 \\ -J_0 \\ 0 \\ 0 \end{bmatrix}$$

or

$$\begin{bmatrix} -J_1 & -\mathbf{X}_1 & -I & 0 \\ J_0 & \mathbf{X}_0 & 0 & -I \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & -I \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_* \\ \xi_1 \\ \xi_0 \end{bmatrix} \leq \begin{bmatrix} -J_1 \\ -J_0 \\ 0 \\ 0 \end{bmatrix}.$$

In the SVM literature the criterion function (4) often gets multiplied by $\tilde{C} \equiv 1/2k$ which results in minor changes to the results.

As discussed in *ALM-III*, Appendix A.3, typically one finds an n vector λ_1 (*not* an N_1 vector like v_1 in $v = (v_1', v_0')'$) that *maximizes* the dual criterion

$$\frac{-1}{2k}\lambda_1' \begin{bmatrix} \mathbf{X}_1\mathbf{X}_1' & -\mathbf{X}_1\mathbf{X}_0' \\ -\mathbf{X}_0\mathbf{X}_1' & \mathbf{X}_0\mathbf{X}_0' \end{bmatrix} \lambda_1 + \lambda_1' J_n.$$

(*ALM-III*, Appendix A.3 involves another n vector λ_2 because there are $2n$ linear inequality constraints.) The dual criterion has λ_1 subject to the constraints

$$-J_1'\lambda_{11} + J_0'\lambda_{10} = 0$$

and

$$0_n \leq \lambda_1 \leq J_n.$$

Actual solutions β and ξ need to incorporate the well-known KKT conditions.

ALM-III, Appendix A.3 establishes that

$$\hat{\beta}_* = \frac{1}{2k}\left(\mathbf{X}_1'\lambda_{11} - \mathbf{X}_0'\lambda_{10}\right).$$

Often many of the λ_1 values are zero, so it makes sense to report only the values of λ_1 that are nonzero and report the corresponding rows of $\mathbf{X}_1$ and $-\mathbf{X}_0$. The computer programs I have seen do something equivalent; they report the nonzero coefficients of $\frac{1}{2k}(\lambda_{11}', -\lambda_{10}')$ and report the corresponding rows of $\mathbf{X}_1$ and $\mathbf{X}_0$ as *support vectors*. Typically, they make you figure out $\hat{\beta}_*$.

As discussed in *ALM-III*, Appendix A.3, if $0 < \lambda_{1h} < 1$, depending on whether y_h is 1 or 0, we must have $\hat{\beta}_0 + \mathbf{x}_h'\hat{\beta}_* = 1$ or $\hat{\beta}_0 + \mathbf{x}_h'\hat{\beta}_* = -1$, respectively. Changing notation a bit, think about $\lambda_1' = (\lambda_{11}', \lambda_{10}')$. If y_{1j} denotes an element of Y_1 with $0 < \lambda_{11j} < 1$, then $\hat{\beta}_0 = 1 - \mathbf{x}_{1j}'\hat{\beta}_*$ and similarly, when y_{0j} has $0 < \lambda_{10j} < 1$, $\hat{\beta}_0 = -1 - \mathbf{x}_{0j}'\hat{\beta}_*$. If you have the correct λ_1, and thus the correct $\hat{\beta}_*$, all of these cases should give the same $\hat{\beta}_0$.

It may seem curious that finding $\hat{\beta}_0$ is so directly tied to cases with $x'\hat{\beta} = \pm 1$, but remember that any multiple of $\hat{\beta}$ defines the same hyperplane, so we have merely chosen a multiple that defines $\hat{\beta}_0$ in terms of being 1 unit away from an appropriate value of $\mathbf{x}_h'\hat{\beta}_*$.

Computer programs often report $-\hat{\beta}_0$ rather than $\hat{\beta}_0$. Surprisingly, computer programs, and even published works, often make some fuss about how to obtain $\hat{\beta}_0$ from the various cases that have $0 < \lambda_{1h} < 1$. Indeed, when fitting the linear (as opposed to quadratic) model to the Cushing's Syndrome data, the `svm`

program for R reports three vectors with $0 < \lambda_{1h} < 1$. These imply the values $\hat{\beta}_0 = 3.101790, 3.102093, 3.101790$. Your guess is as good as mine for why the middle one is slightly different. The program reports $-\hat{\beta}_0 = -3.101891$, which is the average of the three.

9.5.2.1 The Kernel Trick

As discussed in Chapter 1, you could just replace X with $\tilde{R}$ and proceed exactly as before. However, the SVM methodology admits a more particular approach to using kernels. Every time you evaluate $\mathbf{x}'\mathbf{x}_h$ in the discussion, you could replace it with an evaluation of $R(\mathbf{x}, \mathbf{x}_h)$. The primary change that ensues is that instead of evaluating

$$\begin{aligned}
\mathbf{x}'\hat{\beta}_* &= \mathbf{x}'\left[\frac{1}{2k}\left(\mathbf{X}_1'\lambda_{11} - \mathbf{X}_0'\lambda_{10}\right)\right] \\
&= \mathbf{x}'\left[\frac{1}{2k}\left(\sum_{j=1}^{N_1}\mathbf{x}_{1j}\lambda_{11j} - \sum_{j=1}^{N_0}\mathbf{x}_{0j}\lambda_{10j}\right)\right] \\
&= \frac{1}{2k}\left(\sum_{j=1}^{N_1}\mathbf{x}'\mathbf{x}_{1j}\lambda_{11j} - \sum_{j=1}^{N_0}\mathbf{x}'\mathbf{x}_{0j}\lambda_{10j}\right),
\end{aligned}$$

you evaluate

$$\frac{1}{2k}\left(\sum_{j=1}^{N_1}R(\mathbf{x}, \mathbf{x}_{1j})\lambda_{11j} - \sum_{j=1}^{N_0}R(\mathbf{x}, \mathbf{x}_{0j})\lambda_{10j}\right).$$

Again, this computation is simplified by dropping terms with $\lambda_{1rj} = 0$, $r = 0, 1$.

9.5.3 Advantages of SVMs

To be honest, the whole point of this chapter is to address support vector machines. I had planned a subsection listing the advantages of SVMs but, after studying SVMs, I no longer see any advantages. I once thought that their ability to involve the kernel trick was an advantage. But we established in Subsection 1.7.2 that $C(\Phi) = C(\tilde{R})$, so the kernel trick applies to any linear structure $X\beta$, whether it is applied to regular linear models, generalized linear models, proportional hazard models, or anything else. The other advantage I imagined for SVMs was computational, because the vector λ_1 is often nonzero on only a relatively small subset of the data involving "support vectors." But in my review of the literature (which was far from complete but more extensive than the references I have given here or in *ALM-III*) I did not notice any such claims being made for SVMs; no more does the theory in *ALM-III*, Appendix A.3 suggest to me any such advantage. In fact, I found some discussion of the need to deal with the computational problems that SVMs have with big data (something that would be unlikely to arise if the computational complexity was be-

ing driven by a relatively small number of support vectors). This is *not* to say that SVMs don't give reasonable answers; they do. I am just not aware of any advantages they have over using logistic regression with the kernel trick and penalized estimation. (One advantage that SVM programs have is that, unlike most logistic regression programs, SVM programs generally have the kernel trick and the ridge penalty built into them.)

9.5.4 Separating Hyper-Hogwash

SVMs are often sold as finding the optimal hyperplane that has all the data from one group above the hyperplane and all the data from the other group below the hyperplane. The "optimal" hyperplane is defined as the hyperplane that maximizes the distance from the plane to the points on either side that are closest to the hyperplane. While this technical argument is correct, as a reason for using SVMs I think it is quite simply hogwash. I am *not* saying that SVMs are hogwash, only this argument for using them. The optimal separating hyperplane phenomenon is based almost entirely on the fact that SVMs involve minimizing the ridge regression penalty.

- The whole point of binomial generalized linear models is to find good ways of estimating probabilities for the cases that are not obviously from one group or the other. If a separating hyperplane exists, the problem is trivial! All of the MLE probabilities can be pushed arbitrarily close to 1 or 0. *The important question for SVMs (like for all linear predictors) is not how to pick a separating hyperplane but how to pick a hyperplane when separation is not possible!*

In Figure 9.1, using a linear function of TL and PL, separation was not possible. In Figure 9.3, using quadratic functions of TL and PL, separation is possible, so the *reported* maximum likelihood logistic and probit fits do that; they separate the cases. In fact, because it is possible to separate the cases, unique maximum likelihood fits to the linear predictors do not exist. The reported curves in Figure 9.3 for logistic and probit regression are merely those reported when R's `glm` function stopped iterating. Anderson (1972) argued that any logistic regression program will find you a separating hyperplane when they exist. Essentially, when the program finds a separating hyperplane, that fact establishes that no unique maximum likelihood estimate will exist. Anderson (1972) and Albert and Anderson (1984) show that there are no unique maximum likelihood estimators for separable logistic regression.

- *Finding the **optimal** separating hyperplane is **largely** a waste of time.* Figure 9.5 illustrates three separating hyperplanes in the form of parabolas. What basis is there for picking one separating hyperplane over another one? In terms of maximizing the likelihood, they are all equally good. Why should you think there would be a best separating hyperplane? Do you really need to impose some artificial optimality criterion to find a "best" separating hyperplane? (I admit that maximizing the distance from the separating hyperplane to the closest points on

either side is a nice choice, if you think it is worth the trouble to make a choice.)

- *If a separating hyperplane exists, and the procedure does not give you a separating hyperplane, then clearly the procedure is not about finding the optimal separating hyperplane.* Figure 9.3 shows that the default parabola fitted by `svm` does *not* separate the two groups, even though the logit and probit fitted parabolas do seperate the groups. I am not saying that the `svm` solution is bad, only that it is not finding a separating hyperplane when one clearly exists.

9.6 Best Prediction and Probability Estimation

We began by assuming independent binary data $y_h \sim \text{Bern}[p(x_h)]$ and showed that fitting generalized linear models leads us to minimizing certain loss functions. In the last two sections we have ignored the distributional assumptions and discussed linear predictors based on minimizing different loss functions. We now go back and relate minimization of arbitrary loss functions to best prediction and to estimation of probabilities. Earlier we made the case that good estimation of probabilities was vital to estimating the best predictors for standard predictive loss functions such as squared error and Hamming.

The best predictor $\hat{f}$ for an arbitrary predictive loss function $\mathscr{L}(y,u)$ satisfies

$$\mathrm{E}_{y,\mathbf{x}}\left\{\mathscr{L}[y,\hat{f}(\mathbf{x})]\right\} = \inf_{f} \mathrm{E}_{y,\mathbf{x}}\left\{\mathscr{L}[y,f(\mathbf{x})]\right\}. \tag{1}$$

The best predictor, if it can be found, is found by conditioning on $\mathbf{x}$ and is the number $\hat{u} \equiv \hat{f}(\mathbf{x})$ that achieves

$$\mathrm{E}_{y|\mathbf{x}}\left[\mathscr{L}(y,\hat{u})\right] = \inf_{u} \mathrm{E}_{y|\mathbf{x}}\left[\mathscr{L}(y,u)\right].$$

If $\mathscr{L}(y,u)$ is differentiable in u for all y and if the derivative can be taken under the integral of the conditional expectation, cf. Cramér (1946), the best prediction for a fixed $\mathbf{x}$ should occur when

$$0 = \mathbf{d}_u \mathrm{E}_{y|\mathbf{x}}\left[\mathscr{L}(y,u)\right] = \mathrm{E}_{y|\mathbf{x}}\left[\mathbf{d}_u \mathscr{L}(y,u)\right].$$

In the special case of binary prediction, this easily becomes

$$0 = p(\mathbf{x})\left[\mathbf{d}_u \mathscr{L}(1,u)\right] + \left[1-p(\mathbf{x})\right]\left[\mathbf{d}_u \mathscr{L}(0,u)\right].$$

Typically, for known $p(\mathbf{x})$, we would solve for $\hat{u} \equiv \hat{f}(\mathbf{x})$ to find the best predictor for the loss function. As alluded to earlier, we can find the best predictor for square error, Hamming, and even absolute error loss functions.

In binary regression, sometimes people solve the equation for $p(\mathbf{x})$,

$$p(\mathbf{x}) = \frac{-\mathbf{d}_u \mathscr{L}[0,u]}{[\mathbf{d}_u \mathscr{L}(1,u)] - [\mathbf{d}_u \mathscr{L}(0,u)]}. \tag{2}$$

The original idea was to use the conditional distribution of y to find the best predictor (BP) under the loss function. Solving for $p(\mathbf{x})$ is using the BP to find the conditional distribution. It presumes that you know the BP without knowing the conditional distribution. In practice, an estimated predictor $u = \tilde{f}(\mathbf{x})$ is sometimes plugged into (2) to obtain an estimate of $p(\mathbf{x})$.

Predictive estimators $\tilde{f}$ are often chosen to achieve

$$\inf_{f \in \mathscr{F}} \left\{ \sum_{h=1}^{n} \mathscr{L}[y_h, f(x_h)] + k\mathscr{P}(f) \right\}.$$

The fact that the sum puts the same weight on each observation pair (y_h, x_h') is something that is (only?) appropriate when the data come from a simple random sample of some population, e.g., not discrimination data.

Standard limit theorems assure that $(1/n)\sum_{h=1}^{n} \mathscr{L}[y_h, f(x_h)]$ will be a reasonable estimate of $\mathrm{E}_{y,\mathbf{x}}\{\mathscr{L}[y, f(\mathbf{x})]\}$ and if $k\mathscr{P}(f)/n \to 0$, we should be able to evaluate the effectiveness of f for large samples. But the estimated predictor $\tilde{f}$ is not generally an estimate of the best predictor $\hat{f}$ as defined by (1), it is an estimate of the best predictor in $\mathscr{F}$. Only if you are willing to assume that the best predictor is in $\mathscr{F}$ does it make sense to use equation (2) to estimate the conditional probabilities, but that is an assumption that we often make. Zhang, Liu, and Wu (2013) discuss these issues and argue that regularization, i.e., incorporating a penalty function when estimating the predictor $\tilde{f}$, can have deleterious effects on using (2) for probability estimation.

It seems to be the case that people often define best prediction with one loss function, e.g., squared error or Hamming, but are willing to use a completely different predictive loss function to obtain an estimated predictor $\tilde{f}$ and an estimate of $p(x)$.

An exercise in ALM-III establishes the following:

- For linear models with squared error loss, equation (2) gives $p(\mathbf{x}) = x'\beta$. This is unsatisfactory since the probabilities are not required to be between 0 and 1.
- For a generalized linear model based on a cdf F that is symmetric about 0, equation (2) returns the standard answer $p(\mathbf{x}) = F(x'\beta)$.
- If a loss function has the properties $\mathscr{L}[0,-u] = \mathscr{L}[1,u]$ and $\mathbf{d}_u \mathscr{L}[1,u] < 0$, then equation (2) gives a number between 0 and 1 with $p(0) = 0.5$.

Equation (2) can work very well as a method for estimating probabilities but it can also work very poorly. It depends on the loss function being used.

Incidentally, equation (2) and fitting binomial generalized linear models are not the only ways to associate a linear predictor with group probabilities given the predictor variables. In Chapter 10 we used LDA and QDA (which are both linear predictors) to estimate group probabilities via estimation of the sampling distribution (density) of the predictor variables. Doing that requires knowledge of the prevalences (marginal group probabilities).

Chapter 10
Discrimination and Allocation

Abstract This chapter discusses discrimination and allocation. Regression data are commonly the result of sampling a population, taking two or more measurements on each individual sampled, and then examining how those variables relate to one another. Discrimination problems have a very different sampling scheme. In discrimination problems data are obtained from multiple groups and we seek efficient means of telling the groups apart, i.e., discriminating between them. Discrimination is closely related to one-way multivariate analysis of variance in that we seek to tell groups apart. One-way multivariate analysis of variance addresses the question of whether the groups are different whereas discriminant analysis seeks to specify how the groups are different. Allocation is the problem of assigning new individuals to their appropriate group. Allocation procedures have immediate application to diagnosing medical conditions.

Consider the eight populations of people determined by all combinations of sex (male, female) and age (adult, adolescent, child, infant). These are commonly used distinctions, but the populations are not clearly defined. It is not obvious when infants become children, when children become adolescents, nor when adolescents become adults. On the other hand, most people can clearly be identified as members of one of these eight groups. It might be of interest to see whether one can *discriminate* among these populations on the basis of, say, various aspects of their blood chemistry. The discrimination problem is sometimes referred to as the problem of *separation*. Another potentially interesting problem is trying to predict the population of a new individual given only the information on their blood chemistry. The problem of predicting the population of a new case is referred to as the problem of *allocation*. Other names for this problem are *identification* and *classification*.

We will illustrate discrimination techniques on Aitchison and Dunsmore's (1975) Cushing's Syndrome data, cf. Example 9.2.1, and Table 9.1. In this chapter we examine all three types: *adenoma*, *bilateral hyperplasia*, and *carcinoma* (except in the last section). Figure 10.1 plots the logs of the amounts of tetrahydrocortisone and pregnanetriol excreted in the subjects' urine for all three groups. In Chapter 9 we treated the data as though it resulted from a sample of 21 Cushing's Syndrome pa-

tients. A key difference here is that we now treat the data as though it resulted from sampling 6 adenoma patients, 10 bilateral hyperplasia patients, and 5 carcinoma patients. The key difference in the sampling schemes is that in the regression sampling we have reason to believe that bilateral hyperplasia is roughly twice as common within the population as the other groups; whereas in the discrimination sampling scheme the mere fact that there are nearly twice as many bilateral hyperplasia patients tells us nothing about the prevalence of bilateral hyperplasia among Cushing's Syndrome patients.

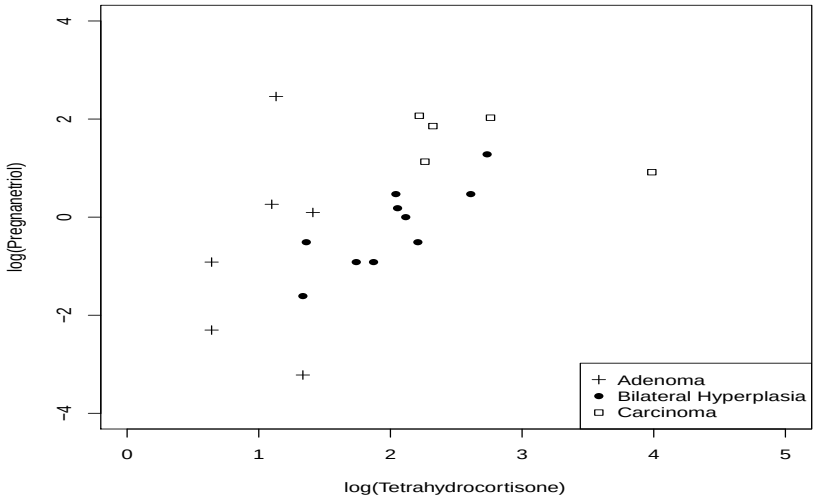

Fig. 10.1 Cushing's Syndrome data.

Most books on multivariate analysis contain extensive discussions of discrimination and allocation. The author can particularly recommend the treatments in Anderson (2003), Johnson and Wichern (2007), and Seber (1984). In addition, Hand (1981) and Lachenbruch (1975) have written monographs on the subject. As is so often the case in statistics, the first modern treatment of these problems was by Sir Ronald A. Fisher; see Fisher (1936, 1938). The discussion in this chapter is closely related to methods associated with the multivariate normal distribution. There are also a variety of nonparametric discrimination methods available.

Another common approach to discrimination is based on logistic regression (or its generalization, log-linear models). Christensen (1997, 2015) and others treat logistic discrimination in more detail. Our interest is restricted to addressing how the binary regression methods of the previous chapter must/can be adjusted for the discriminant analysis sampling scheme.

Discrimination seems to be a purely descriptive endeavor. The observations are vectors $y = (y_1, \ldots, y_q)'$ in $\mathbf{R}^q$. All observations come from known populations. Discriminate analysis uses the observations to partition $\mathbf{R}^q$ into regions, each uniquely associated with a particular population. Given a partition, it is easy to allocate future observations. An observation y is allocated to population r if y falls into the region of $\mathbf{R}^q$ associated with the rth population. The difficulty lies in developing a rational approach to partitioning $\mathbf{R}^q$.

Just as a solution to the discrimination problem implicitly determines an allocation rule, a solution to the allocation problem implicitly solves the discrimination problem. The set of all y values to be allocated to population r determines the region associated with population r.

Our discussion will be centered on the allocation problem. We present allocation rules based on Mahalanobis's distance, maximum likelihood, and Bayes theorem. An advantage of the Mahalanobis distance method is that it is based solely on the means and covariances of the population distributions. The other methods require knowledge of the entire distribution in the form of a density. Not surprisingly, the Mahalanobis, the maximum likelihood, and the Bayes rules are similar for normally distributed populations.

In general, consider the situation in which there are t populations and q variables $y_1, \ldots, y_q$ with which to discriminate among them. In particular, if $y = (y_1, \ldots, y_q)'$ is an observation from the ith population, we assume that either the mean and covariance matrix of the population are known, say

$$\mathrm{E}(y) = \mu_i$$

and

$$\mathrm{Cov}(y) = \Sigma_i,$$

or that the density of the population distribution, say

$$f(y|i),$$

is known. In practice, neither the density, the mean, nor the covariance matrix will be known for any population. These must be estimated using data from the various populations.

An important special case is where the covariance matrix is the same for all populations, say

$$\Sigma \equiv \Sigma_1 = \cdots = \Sigma_t\,.$$

In this case, samples from the populations constitute data for a standard one-way *multivariate analysis of variance (MANOVA)*. A complete discussion of MANOVA is beyond the scope of this book, see *ANREG* for a basic introduction and *ALM* for a more theortical one. Appendix C contains an illustration of a three-factor MANOVA.

Section 1 deals with the general allocation problem. Section 2 examines quadratic discriminant analysis (QDA). It applies the general ideas by estimating parameters and densities. Section 3 examines linear discriminant analysis (LDA). It is the spe-

cial case of equal covariance matrices. Section 4 introduces ideas of cross-validation for estimating error rates. Section 5 contains some general discussion. Sections 6 and 7 examine the relationship between MANOVA and LDA and in particular discuss a method for selecting variables in LDA and introduce discrimination coordinates that are useful in visualizing the discrimination procedure. Section 8 discusses a broader idea of linear discrimination that includes both LDA and QDA and relates it to the ideas of linear prediction rules examined in the previous chapter. Section 9 gets into how one adjusts binary regression results for the discrimination sampling scheme.

10.1 The General Allocation Problem

In this section, we discuss allocation rules based on Mahalanobis's distance, maximum likelihood, and Bayes theorem. These rules are based on populations with either known means and covariances or known densities.

10.1.1 Mahalanobis's distance

The Mahalanobis distance

$$D^2 \equiv (y-\mu)'\Sigma^{-1}(y-\mu)$$

is a frequently used measure of how far a random vector is from the center of its distribution, cf. *PA*. In the allocation problem, we have a random vector y and t possible distributions from which it could arise. A reasonable allocation procedure is to assign y to the population that minimizes the observed Mahalanobis distance. In other words, allocate y to population r if

$$(y-\mu_r)'\Sigma_r^{-1}(y-\mu_r) = \min_i (y-\mu_i)'\Sigma_i^{-1}(y-\mu_i)\,. \tag{1}$$

10.1.2 Maximum likelihood

If the densities $f(y|i)$ are known for each population, the population index i is the only unknown parameter. Given an observation y, the likelihood function is

$$L(i) \equiv f(y|i),$$

which is defined for $i = 1,\ldots,t$. The maximum likelihood allocation rule assigns y to population r if

$$L(r) = \max_i L(i),$$

or equivalently if

$$f(y|r) = \max_i f(y|i).$$

If the observations have a multivariate normal distribution, the maximum likelihood rule is very similar to the Mahalanobis distance rule. From *PA* Section 1.2, the likelihoods (densities) are

$$L(i) = f(y|i) = (2\pi)^{-q/2}|\Sigma_i|^{-1/2}\exp[-(y-\mu_i)'\Sigma_i^{-1}(y-\mu_i)/2],$$

$i = 1,\ldots,t$, where $|\Sigma_i|$ is the determinant of the covariance matrix. The logarithm is a monotone increasing function, so maximizing the log-likelihood is equivalent to maximizing the likelihood. The log-likelihood is

$$\ell(i) \equiv \log[L(i)] = -\frac{q}{2}\log(2\pi) - \frac{1}{2}\log(|\Sigma_i|) - \frac{1}{2}(y-\mu_i)'\Sigma_i^{-1}(y-\mu_i).$$

If we drop the constant term $-\frac{q}{2}\log(2\pi)$ and minimize twice the negative of the log-likelihood rather than maximizing the log-likelihood, we see that the maximum likelihood rule for normally distributed populations is: assign y to population r if

$$\log(|\Sigma_r|) + (y-\mu_r)'\Sigma_r^{-1}(y-\mu_r) = \min_i \left\{\log(|\Sigma_i|) + (y-\mu_i)'\Sigma_i^{-1}(y-\mu_i)\right\}. \quad (2)$$

The only difference between the maximum likelihood rule and the Mahalanobis rule is the inclusion of the term $\log(|\Sigma_i|)$ which does not depend on y, the case to be allocated. Both the Mahalanobis rule and the maximum likelihood rule involve quadratic functions of y. Methods related to these rules are often referred to as *quadratic discrimination* methods. (As discussed in the previous chapter, quadratic functions of the predictor variables define linear functions of an expanded set of predictor variables.)

EXAMPLE 10.1.1. The simplest case of allocation is assigning a new observation y to one of two normal populations with the same variance. The top panel of Figure 10.2 contains normal densities with variance 1. The one on the left has mean 2; the one on the right has mean 5. The solid dot is the point at which the two densities are equal. For y to the right of the dot, the maximum likelihood allocation is to the mean 5 population. To the left of the dot, the maximum likelihood allocation is to the mean 2 population. The Mahalanobis distance in this one dimensional problem is $|y-\mu_i|/1$. The black dot is also the solution to $y-2 = y-5$, so y values to the left of the dot are closer to the mean 2 population and those to the right are closer to the mean 5 population.

The bottom panel of Figure 10.2 is far more complicated because it involves unequal variances. The population on the left is $N(2,1)$ whereas the population on the right is now $N(5,9)$. The two squares are where the densities from the two distributions are equal. To the left of the left square and to the right of the right square, the $N(5,9)$ has a higher density, so a y in those regions would be assigned

to that population. Between the two squares, the $N(2,1)$ has a higher density, so a y between the squares is assigned to the mean 2 population. The squared Mahalanobis distances are $(y-2)^2/1$ and $(y-5)^2/9$. Setting these equal gives the two black dots. Again, y is assigned to the mean 2 population if and only if y is between the black dots. The normal theory maximum likelihood and Mahalanobis methods are similar but distinct. □

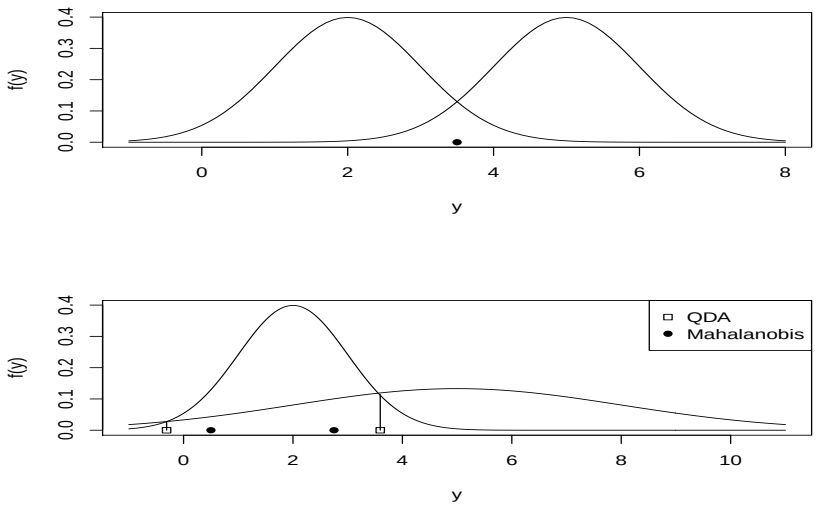

Fig. 10.2 One dimensional normal discrimination.

10.1.3 Bayesian methods

We will discuss an intuitive rule for Bayesian allocation. *ALM-III* also discusses a formal procedure based on costs of misclassification and shows that the intuitive rule can be arrived at by the formal procedure.

Bayesian allocation methods presuppose that for each population i there exists a prior probability, say $\pi(i)$, that the new observation y comes from that population. Typically, these prior probabilities are arrived at either from previous knowledge of the problem or through the use of the maximum entropy principle (see Berger 1985, Section 3.4). The maximum entropy principle dictates that the $\pi(i)$s should be chosen to minimize the amount of information contained in them. This is achieved by selecting

$$\pi(i) = 1/t\,, \tag{3}$$

$i = 1, \ldots, t$.

Given the prior probabilities and the data y, the posterior probability that y came from population i can be computed using Bayes theorem (see Berger, 1985, Section 4.2 or Christensen et al., 2010, Chapter 2). The posterior probability is

$$\pi(i|y) = f(y|i)\pi(i) \Big/ \sum_{j=1}^{t} f(y|j)\pi(j)\,. \tag{4}$$

A simple intuitive allocation rule is to assign y to the population with the highest posterior probability. In other words, assign y to population r if

$$\pi(r|y) = \max_i \pi(i|y)\,.$$

The denominator in (4) does not depend on i, so the allocation rule is equivalent to choosing r such that

$$f(y|r)\pi(r) = \max_i \{f(y|i)\pi(i)\}.$$

In the important special case in which the $\pi(i)$ are all equal, this corresponds to maximizing $f(y|i)$; that is, choosing r so that

$$f(y|r) = \max_i f(y|i)\,.$$

Thus, for equal initial probabilities, the intuitive Bayes allocation rule is the same as the maximum likelihood allocation rule. In particular, if the populations are normal, the Bayes rule with equal prior probabilities is based on (2).

Most methods of discrimination are based on estimating the density functions $f(y|i)$. These include LDA, QDA, and such nonparametric methods as nearest neighbors and kernels (in the sense of Subsection 7.5.3.). As discussed in Section 9, logistic regression and some other binary regression methods give direct estimates of $\pi(i|y)$, but those estimates are based on having implicitly estimated $\pi(i)$ with $N_i/\sum_j N_j$, where the N_js are the numbers of observations in each group. This is rarely appropriate for discrimination data. Appropriate discrimination procedures must correct for this.

10.2 Estimated Allocation and QDA

One serious problem with the allocation rules of the previous section is that typically the moments and the densities are unknown. In (10.1.1) and (10.1.2) typically the values μ_i and Σ_i are unknown. In practice, allocation is often based on estimated means and covariances or estimated densities.

We assume that a random sample of $N_i \equiv N(i)$ observations is available from the ith population. The jth observation from the ith population is a q vector denoted $y_{ij} \equiv (y_{ij,1}, \ldots, y_{ij,q})'$. Note that

$$\mathrm{E}(y_{ij}) = \mu_i$$

and

$$\mathrm{Cov}(y_{ij}) = \Sigma_i \,.$$

It is important to recognize that in the special case of equal covariance matrices, the data follow a multivariate one-way ANOVA model with t groups, cf. Appendix C. It will be convenient to write a matrix that contains the ith sample,

$$Y_i \equiv \begin{bmatrix} y'_{i1} \\ \vdots \\ y'_{iN(i)} \end{bmatrix} .$$

The estimated Mahalanobis distance rule is that an observation y is allocated to population r if

$$(y-\bar{y}_{r\cdot})' S_r^{-1} (y-\bar{y}_{r\cdot}) = \min_i (y-\bar{y}_{i\cdot})' S_i^{-1} (y-\bar{y}_{i\cdot}) \,,$$

where

$$S_i = \sum_{j=1}^{N_i} (y_{ij}-\bar{y}_{i\cdot})(y_{ij}-\bar{y}_{i\cdot})' \Big/ (N_i-1) = Y_i' \left[I - \frac{1}{N_i} J_{N(i)}^{N(i)} \right] Y_i \Big/ (N_i-1) \,.$$

An estimated maximum likelihood allocation rule is to assign y to population r if

$$\hat{f}(y|r) = \max_i \hat{f}(y|i).$$

If q is not too large, the estimate $\hat{f}(y|i)$ can be estimated nonparametrically using nearest neighbors or kernels (Subsection 7.5.3) or, if $\hat{f}(y|i)$ depends on parameters θ_i, $\hat{f}(y|i)$ can be obtained by estimating the parameters, which is what we do for multivariate normals.

For multivariate normal densities, an estimated maximum likelihood allocation rule is to assign y to population r if

$$\log(|S_r|) + (y-\bar{y}_{r\cdot})' S_r^{-1} (y-\bar{y}_{r\cdot}) = \min_i \{ \log(|S_i|) + (y-\bar{y}_{i\cdot})' S_i^{-1} (y-\bar{y}_{i\cdot}) \} \,.$$

Application of this estimated normal theory allocation rule is often referred to as *Quadratic Discriminant Analysis (QDA)*.

Although the QDA allocation decision is based a quadratic function of the predictor variables in y, the quadratic function in y can also be written as a linear combination of the y predictor variables, their squares, and their crossproducts. In one sense, QDA is not linear discrimination but in another sense it is. (In terms of the "training" data y_{ij}, it is neither quadratic nor linear but rather a very complicated function.) This issue is discussed further in Section 8.

With estimated parameters, the Bayes allocation rule is really only a quasi-Bayesian rule. The allocation is Bayesian, but the estimation of $f(y|i)$ is not.

Geisser's (1971) suggestion of using the Bayesian predictive distribution as an estimate of $f(y|i)$ has been shown to be optimal under frequentist criteria by Aitchison (1975), Murray (1977), and Levy and Perng (1986). It also provides the optimal Bayesian allocation. In particular, for normal data, treating the maximum likelihood estimates as the mean and covariance matrix of the normal gives an inferior estimate for the distribution of new observations. The appropriate distribution (for "noninformative" priors) is a multivariate t with the same location vector and a covariance matrix that is a multiple of the MLE. See Geisser (1977) for a discussion of these issues in relation to discrimination.

In any case, plugging in estimates of μ_i and Σ_i requires that good estimates be available. Friedman (1989) has proposed an alternative estimation technique for use with small samples.

Finally, in examining the data, it is generally not enough to look just at the results of the allocation. Typically, one is interested not only in the population to which a case is allocated but also in the clarity of the allocation. It is desirable to know whether a case y is clearly in one population or whether it could have come from two or more populations. The posterior probabilities from the Bayesian method address these questions in the simplest fashion. Similar information can be gathered from examining the entire likelihood function or the entire set of Mahalanobis distances.

We now illustrate QDA on the Cushing's Syndrome data. Clarity of allocation is addressed later in Example 10.5.1. An alternative analysis of the data based on assuming equal covariance matrices for the groups is presented in the next section.

EXAMPLE 10.2.1. *Cushing' Syndrome: Quadratic Discrimination Analysis.*
We wish to discriminate among the three types of Cushing's Syndrome based on the urinary excretion data. The data were given in Table 9.1. The pregnanetriol value for case 4 looks out of place because it is the only value that is nonzero in the hundredths place but, from looking at similar data, it is apparently a valid observation. Ripley (1996, Section 2.4) discusses these data in the context of predictive allocation.

As before we analyze logs of the data. The log data were plotted in Figure 10.1. A quick glance at the figure establishes that none of the data groups seems clearly to be from a bivariate normal distribution, but with small sample sizes it is always hard to tell. Performing a QDA on the Cushing's Syndrome data begins with estimating the means and covariance matrices for the three populations. These are as follows.

Variable	Group Means			Grand
	a	*b*	*c*	Mean
log(Tet)	1.0433	2.0073	2.7097	1.8991
log(Preg)	−0.60342	−0.20604	1.5998	0.11038

Covariance Matrix for Adenoma

	log(Tet)	log(Preg)
log(Tet)	0.1107	0.1239
log(Preg)	0.1239	4.0891

Covariance Matrix for Bilateral Hyperplasia

	log(Tet)	log(Preg)
log(Tet)	0.2119	0.3241
log(Preg)	0.3241	0.7203

Covariance Matrix for Carcinoma

	log(Tet)	log(Preg)
log(Tet)	0.5552	−0.2422
log(Preg)	−0.2422	0.2885

The results of QDA are given in Table 10.1. Group probabilities are computed in two ways. First, the data are used to estimate the parameters of the normal densities and the estimated densities are plugged into Bayes theorem. Second, the probabilities are estimated from cross-validation. The details of cross-validation will be discussed in Section 4. The analyses are based on *equal prior probabilities*; hence, they are also maximum likelihood allocations. Using the resubstitution method, only two cases are misallocated: 9 and 12. Under cross-validation, cases 1, 8, 19, and 20 are also misclassified. □

10.3 Linear Discrimination Analysis: LDA

If the populations are all multivariate normal with identical covariance matrices, say $\Sigma \equiv \Sigma_1 = \cdots = \Sigma_t$, then the Mahalanobis distance rule is identical to the maximum-likelihood/Bayes allocation rule (10.1.2). The maximum-likelihood/Bayes allocation rule assigns y to the population r that satisfies

$$\log(|\Sigma|) + (y-\mu_r)'\Sigma^{-1}(y-\mu_r) = \min_i \{\log(|\Sigma|) + (y-\mu_i)'\Sigma^{-1}(y-\mu_i)\}\,.$$

However, the term $\log(|\Sigma|)$ is the same for all populations, so the rule is equivalent to choosing the population r that satisfies

$$(y-\mu_r)'\Sigma^{-1}(y-\mu_r) = \min_i (y-\mu_i)'\Sigma^{-1}(y-\mu_i)\,.$$

This is precisely the Mahalanobis distance rule (10.1.1).

In practice, estimates must be substituted for Σ and the μ_is. With equal covariance matrices the y_{ij} data fit the multivariate one-way ANOVA model of Appendix C, so the standard estimates $\bar{y}_{i\cdot}$, $i = 1,\ldots,t$ and $S = E/(n-t)$ are reasonable. The estimated allocation rule is: assign y to population r if

$$(y-\bar{y}_{r\cdot})'S^{-1}(y-\bar{y}_{r\cdot}) = \min_i (y-\bar{y}_{i\cdot})'S^{-1}(y-\bar{y}_{i\cdot})\,.$$

Table 10.1 Quadratic Discrimination Analysis.

Allocated to Group	Resubstitution True Group			Cross-Validation True Group		
	a	*b*	*c*	*a*	*b*	*c*
a	6	1	0	5	2	0
b	0	8	0	0	7	2
c	0	1	5	1	1	3

Case		Group	Resubstitution Probability			Cross-Validation Probability		
			a	*b*	*c*	*a*	*b*	*c*
1	**	*a*	0.61	0.00	0.39	0.24	0.00	0.76
2		*a*	1.00	0.00	0.00	1.00	0.00	0.00
3		*a*	0.91	0.09	0.00	0.81	0.19	0.00
4		*a*	1.00	0.00	0.00	0.96	0.04	0.00
5		*a*	0.92	0.08	0.00	0.87	0.13	0.00
6		*a*	1.00	0.00	0.00	1.00	0.00	0.00
7		*b*	0.00	1.00	0.00	0.00	1.00	0.00
8	**	*b*	0.42	0.58	0.00	0.62	0.38	0.00
9	**	*b*	0.61	0.39	0.00	0.93	0.07	0.00
10		*b*	0.00	1.00	0.00	0.00	1.00	0.00
11		*b*	0.00	1.00	0.00	0.00	1.00	0.00
12	**	*b*	0.00	0.28	0.72	0.00	0.12	0.88
13		*b*	0.01	0.99	0.00	0.01	0.98	0.01
14		*b*	0.02	0.98	0.00	0.03	0.97	0.00
15		*b*	0.04	0.96	0.00	0.05	0.95	0.00
16		*b*	0.00	0.96	0.04	0.00	0.94	0.06
17		*c*	0.00	0.01	0.99	0.00	0.01	0.99
18		*c*	0.00	0.00	1.00	0.00	0.00	1.00
19	**	*c*	0.00	0.40	0.60	0.00	1.00	0.00
20	**	*c*	0.00	0.00	1.00	0.00	1.00	0.00
21		*c*	0.00	0.05	0.95	0.00	0.08	0.92

Recall that in a one-way ANOVA, the estimated covariance matrix is a weighted average of the individual estimates, namely

$$S = \sum_{i=1}^{t} (N_i - 1) S_i / (n - t) \, .$$

Although $(y - \bar{y}_{i\cdot})' S^{-1} (y - \bar{y}_{i\cdot})$ is a quadratic function of y, the allocation only depends on a linear function of y. Note that

$$(y - \bar{y}_{i\cdot})' S^{-1} (y - \bar{y}_{i\cdot}) = y' S^{-1} y - 2 \bar{y}_{i\cdot}' S^{-1} y + \bar{y}_{i\cdot}' S^{-1} \bar{y}_{i\cdot} \, .$$

The term $y' S^{-1} y$ is the same for all populations. Subtracting this constant and dividing by -2, the allocation rule can be rewritten as: assign y to population r if

$$y'S^{-1}\bar{y}_{r\cdot} - \frac{1}{2}\bar{y}'_{r\cdot}S^{-1}\bar{y}_{r\cdot} = \max_i \left\{ y'S^{-1}\bar{y}_{i\cdot} - \frac{1}{2}\bar{y}_{i\cdot}S^{-1}\bar{y}_{i\cdot} \right\}.$$

This is based on a linear function of y, so this allocation rule is often referred to as (traditional) *Linear Discriminant Analysis (LDA)*.

EXAMPLE 10.3.1. *Cushing's Syndrome data.*
LDA results from estimating normal densities with equal covariance matrices. In this example we also *assume equal prior probabilities* so the Bayesian allocation corresponds to a maximum likelihood allocation. All results are based on pooling the covariance matrices from Example 10.2.1 into

Pooled Covariance Matrix

	log(Tet)	log(Preg)
log(Tet)	0.2601	0.1427
log(Preg)	0.1427	1.5601.

The results of the linear discriminant analysis are summarized in Table 10.2. Based on resubstitution, four cases are misallocated, all from group b. Based on leave-one-out cross-validation, three additional cases are misallocated. Cases 8 and 9 are consistently classified as belonging to group a, and cases 12 and 16 are classified as c. In addition, when they are left out of the fitting process, cases 1 and 4 are allocated to groups c and b, respectively, while case 19 is misallocated as b. It is interesting to note that linear discrimination has a hard time deciding whether case 19 belongs to group b or c. A more detailed discussion of these cases is given later. □

If $q = 2$ we can easily plot the data from each population as we did in Figure 10.1. Consider $t = 2$ populations. As demonstrated in the next paragraph, linear discrimination estimates a line that best separates the two clouds of data points. This is similar to what we did in Figure 9.1 to predict bilateral hyperplasia and carcinoma but the criteria for choosing the line has now changed because the nature of the data is different. (Or at least we are treating it differently.) In Section 9 we will consider the differences between the LDA line, the SVM line from the previous chapter, and a logistic regression line modified to deal with discrimination data. If $q = 3$, while it is harder to do, we can still plot the data from each population. LDA now estimates the plane in 3 dimensions that best classifies the two clouds of data points. For $q > 3$, visualization becomes problematic but LDA uses hyperplanes, i.e., translated (shifted) vector spaces (affine spaces) of dimension $q - 1$, to classify the populations. Under different assumptions, QDA estimates a quadratic, rather than linear, function of y that classifies best. Again, QDA uses a different quadratic function than those illustrated in Figures 9.3 and 9.5, see Section 9 for comparisons. Remember, quadratics are linear functions in higher dimensional spaces than $\mathbf{R}^q$ and LDA and QDA only estimate the best classifiers when classifying multivariate normal data y.

We already know that LDA involves maximizing t linear functions of the data. If there are just $t = 2$ groups, we now show that LDA classifies by evaluating whether

Table 10.2 Linear discrimination.

Allocated to Group	Resubstitution True Group			Cross-Validation True Group		
	a	*b*	*c*	*a*	*b*	*c*
a	6	2	0	4	2	0
b	0	6	0	1	6	1
c	0	2	5	1	2	4

Case		Group	Resubstitution Probability			Cross-Validation Probability		
			a	*b*	*c*	*a*	*b*	*c*
1	**	*a*	0.80	0.14	0.05	0.17	0.30	0.54
2		*a*	0.83	0.16	0.01	0.80	0.19	0.01
3		*a*	0.96	0.04	0.00	0.94	0.06	0.00
4	**	*a*	0.61	0.39	0.00	0.12	0.88	0.00
5		*a*	0.60	0.37	0.03	0.56	0.41	0.03
6		*a*	0.96	0.04	0.00	0.96	0.04	0.00
7		*b*	0.08	0.71	0.21	0.09	0.69	0.22
8	**	*b*	0.64	0.35	0.00	0.75	0.25	0.00
9	**	*b*	0.64	0.35	0.01	0.69	0.30	0.01
10		*b*	0.10	0.69	0.22	0.11	0.67	0.23
11		*b*	0.06	0.77	0.17	0.07	0.75	0.19
12	**	*b*	0.00	0.20	0.80	0.00	0.12	0.88
13		*b*	0.10	0.64	0.26	0.11	0.62	0.27
14		*b*	0.19	0.75	0.06	0.21	0.73	0.06
15		*b*	0.29	0.68	0.04	0.31	0.65	0.04
16	**	*b*	0.01	0.42	0.58	0.01	0.36	0.63
17		*c*	0.02	0.26	0.73	0.02	0.31	0.67
18		*c*	0.02	0.26	0.72	0.04	0.36	0.61
19	**	*c*	0.03	0.43	0.53	0.03	0.49	0.47
20		*c*	0.00	0.02	0.98	0.00	0.14	0.86
21		*c*	0.00	0.10	0.89	0.00	0.12	0.88

a linear function of y is greater or less than (above or below) a particular hyperplane in q dimensions. We have seen that the traditional LDA rule assigns y to group 1 if

$$y'S^{-1}\bar{y}_{1\cdot} - \frac{1}{2}\bar{y}'_{1\cdot}S^{-1}\bar{y}_{1\cdot} > y'S^{-1}\bar{y}_{2\cdot} - \frac{1}{2}\bar{y}'_{2\cdot}S^{-1}\bar{y}_{2\cdot},$$

which occurs if and only if

$$y'S^{-1}(\bar{y}_{1\cdot} - \bar{y}_{2\cdot}) > \frac{1}{2}\bar{y}'_{1\cdot}S^{-1}\bar{y}_{1\cdot} - \frac{1}{2}\bar{y}'_{2\cdot}S^{-1}\bar{y}_{2\cdot},$$

iff

$$y'S^{-1}(\bar{y}_{1\cdot} - \bar{y}_{2\cdot}) > \frac{1}{2}(\bar{y}_{1\cdot} + \bar{y}_{2\cdot})'S^{-1}(\bar{y}_{1\cdot} - \bar{y}_{2\cdot}),$$

iff, with $\hat{\mu} \equiv (\bar{y}_{1\cdot} + \bar{y}_{2\cdot})/2$,

$$(y-\hat{\mu})'S^{-1}(\bar{y}_{1\cdot}-\bar{y}_{2\cdot})>0. \tag{5}$$

The classification line (or plane or hyperplane) in a plot consists of the y values with $(y-\hat{\mu})'S^{-1}(\bar{y}_{1\cdot}-\bar{y}_{2\cdot})=0$ or $y'S^{-1}(\bar{y}_{1\cdot}-\bar{y}_{2\cdot})=\hat{\mu}'S^{-1}(\bar{y}_{1\cdot}-\bar{y}_{2\cdot})$.

Similar to the discussion in Section 9.4, any linear function of y can be used to form a two group discrimination rule, say, assign y to a group based on whether $y'\beta_*>\beta_0$. The trick is to find a good vector β_* and a good constant β_0. LDA uses obviously good estimates of what are the optimal choices for β_* and β_0 when the different populations are multivariate normal with equal covariance matrices.

10.4 Cross-Validation

It is of considerable interest to be able to evaluate the performance of allocation rules. Depending on the populations involved, there is generally some level of misclassification that is unavoidable. (If the populations are easy to tell apart, why are you worrying about them? In particular, cf. Subsection 9.5.4.) If the distributions that determine the allocation rules are known, one can simply classify random samples from the various populations to see how often the data are misclassified. This provides simple yet valid estimates of the error rates. Unfortunately, things are rarely that straightforward. In practice, the data available are used to estimate the distributions of the populations. If the same data are also used to estimate the error rates, a bias is introduced. Typically, this double dipping in the data overestimates the performance of the allocation rules. The method of estimating error rates by reclassifying the data used to construct the classification rules is often called the *resubstitution method*.

To avoid the bias of the resubstitution method, *cross-validation* is often used, cf. Geisser (1977) and Lachenbruch (1975). Cross-validation often involves leaving out one data point, estimating the allocation rule from the remaining data, and then classifying the deleted case using the estimated rule. Every data point is left out in turn. Error rates are estimated by the proportions of misclassified cases. This version of cross-validation is also known as the *jackknife*. (The jackknife was originally a tool for reducing bias in location estimates.) The computation of the cross-validation error rates can be simplified by the use of updating formulae similar to those discussed in *PA-V* Section 12.5 (Christensen, 2011, Section 13.5).

While resubstitution underestimates error rates, cross-validation may tend to overestimate them. In standard linear models, if one thinks of the variance σ^2 as the error rate, resubstitution is analogous to estimating the variance with the *naive estimate* SSE/n (also the normal theory MLE) whereas leave-one-out cross-validation is analogous to estimating the variance with PRESS/n where PRESS is the *predicted residual sum of squares* discussed in *PA-V* Section 12.5 (Christensen, 2011, Section 13.5). Using Jensen's inequality it is not too difficult to show that

$$\mathrm{E}\left[\frac{SSE}{n}\right] < \mathrm{E}\left[\frac{SSE}{n-r(X)}\right] = \sigma^2 \leq \mathrm{E}\left[\frac{PRESS}{n}\right],$$

and, indeed, that on average the cross-validation estimate PRESS$/n$ over estimates σ^2 by at least as much as the naive (resubstitution) method underestimates σ^2. While this is not really an issue in linear models, because we know how to find an unbiased estimate of the error in linear models, this result calls in question the idea of blindly using leave-one-out cross-validation to estimate error rates in allocation problems and logistic regression. In fact, since the over-estimation and the under-estimation seem to have similar orders of magnitude in standard linear models, one might consider averaging the two estimates.

For large data sets, K group cross-validation seems to be more popular than leave-one-out cross-validation. This involves (1) randomly dividing the data into K groups and (2) fitting the model on $K-1$ of those groups. Evaluate the error by (3) using this fitted model to allocate the data for the one omitted group, and (4) comparing these allocations to the true group memberships for the omitted group. This is done K times, where each group is omitted one time. The overall estimates of error are averages from the K different estimates. This approach requires quite a bit of data to be effective. Leave-one-out cross-validation uses $K = n$ but it seems popular to pick K considerably smaller than n and, indeed, this seems likely to reduce the bias problem of the previous paragraph.

When I first wrote this book (and contrary to the previous discussion), leave-one-out cross-validation was considered to have less bias than the resubstitution method but typically a considerably larger variance, cf. more recently Hastie et al. (2016, Section 7.10). Hastie et al. also suggest that smaller values of K like $K = 5$ should have less variability but possibly more bias.

If the number of observations is much larger than the number of parameters to be estimated, resubstitution is often adequate for estimating the error rates. When the number of parameters is large relative to the number of observations, the bias becomes unacceptably high. Under normal theory, the parameters involved are simply the means and covariances for the populations. Thus, the key issue is the number of variables used in the discrimination relative to the number of observations. (While we have not discussed nonparametric discrimination, in the context of this discussion nonparametric methods should be considered as highly parametric methods.)

The bootstrap (which was introduced in Section 5.5) has also been suggested as a tool for estimating error rates. It often has both small bias and small variance, but it is computationally intensive and handles large biases poorly. The interested reader is referred to Efron (1983) and the report of the Panel on Discriminant Analysis, Classification, and Clustering in *Statistical Science* (1989).

10.5 Discussion

EXAMPLE 10.5.1. *Cushing's Syndrome Data.*
Careful inspection of Table 9.1 and Figure 10.1 sheds light on both the QDA and LDA procedures summarized in Tables 10.1 and 10.2. From Figure 10.1, there seems to be almost no evidence that the covariance matrices of the three groups are equal. (The spatial orientation of the three clusters of points look very different.) Adenoma displays large variability in log(pregnanetriol), very small variability in log(tetrahydrocortisone), and almost no correlation between the variables. Carcinoma is almost the opposite. It has large variability in log(Tet) and small variability in log(Preg). Carcinoma seems to have a negative correlation. Bilateral hyperplasia displays substantial variability in both variables, with a positive correlation. These conclusions are also visible from the estimated covariance matrices. Given that the covariance structure seems to differ from group to group, linear discrimination does surprisingly well when evaluated by resubstitution. As per references given later, Linear Discriminant Analysis has been found to be rather robust. Of course, QDA does a much better job for these data.

The fact that the assessments based on cross-validation are much worse than those based on resubstitution is due largely to the existence of influential observations. The mean of group c and especially the covariance structure of group c are dominated by the large value of log(Tet) for case 20. Case 20 is not misclassified by the LDA because its effect on the covariance structure is minimized by the pooling of covariance estimates over groups. In cross-validated QDA, its effect on the covariance of group c is eliminated, so case 20 seems more consistent with group b. The large log(Preg) value of case 1 is also highly influential. With case 1 dropped out and case 20 included, case 1 is more consistent with carcinoma than with adenoma. The reason that cases 8 and 9 are misclassified is simply that they tend to be consistent with group a, see Figure 10.1. In examining Table 9.1, a certain symmetry can be seen involving cases 12 and 19. Because of case 19, when case 12 is unassigned it looks more like group c than its original group. Similarly, because of case 12, when case 19 is unassigned it looks more like group b than group c under quadratic discrimination. Case 19 is essentially a toss-up under LDA. Cases 4 and 16 are misclassified under LDA because they involve very unusual data. Case 4 has an extremely small pregnanetriol value, and case 16 has a very large tetrahydrocortisone value for being part of the bilateral hyperplasia group.

In a data set this small, it seems unreasonable to drop influential observations. If we cannot believe the data, there is little hope of being able to arrive at a reasonable analysis. If further data bear out the covariance tendencies visible in Figure 10.1, the better analysis is provided by quadratic discrimination. It must be acknowledged that the error rates obtained by resubstitution are unreliable. They are generally biased toward underestimating the true error rates and may be particularly bad for these data. QDA simply provides a good description of the data. There is probably insufficient data to produce really good predictions. □

To this point the methods explicitly discussed in this chapter both relate to the normal distribution. If the true distributions $f(y|i)$ are elliptically symmetric, both the quadratic and linear methods work well. Moreover, the LDA method is generally quite robust; it even seems to work quite well for discrete data. See Lachenbruch, Sneeringeer, and Revo (1973), Lachenbruch (1975), and Hand (1983) for details.

The gold standard for discrimination seems to be, depending on one's philosophical bent, maximum likelihood or Bayesian discrimination. But they are only the gold standard if you know what the distributions are. If you know the densities, those are the only functions of the data that need concern you.

Linear and quadratic discrimination for nonnormal data can be based on Mahalanobis distances rather than on densities. Since they are not based on densities, they are ad hoc methods. Many of the binary regression methods discussed in the previous chapter provide direct estimates of $\pi(i|y)$ that are (typically) inappropriate for discrimination data but from which appropriate density estimates can be inferred, cf. Section 9. Often the regression methods implicitly or explicitly perform discrimination in higher dimensions. Instead of linear or quadratic discrimination on the basis of, say, $y = (y_1, y_2, y_3)'$, they discriminate on the basis of some extended vector, for example, $\tilde{y} = (y_1, y_2, y_3, y_1^2, y_2^2, y_3^2, y_1y_2, y_1y_3, y_2y_3)'$. *If you know the densities, there is little point in expanding the dimensionality, because the density is the only relevant function of the data.* But if you do not know the densities, expanding the dimensionality can be very useful. In particular, support vector machines typically use expanded data. Of course, one could also perform traditional linear or quadratic discrimination on the new $\tilde{y}$ and I suspect that, when practical, LDA and QDA discrimination on $\tilde{y}$ will often be competitive with the newer methods. Personally, I am more comfortable using expanded data in logistic (or log-linear) discrimination than in LDA or QDA. (Clearly, the data in $\tilde{y}$ will not be multivariate normal!)

For the specific $\tilde{y}$ given above, any linear discrimination method based on $\tilde{y}'\beta_*$ is equivalent to a quadratic discrimination based on y. This is not to say that LDA applied to $\tilde{y}$ is QDA, but merely that this $\tilde{y}'\beta_*$ is always a quadratic function of y. If you know that the data y are normal, QDA on y is pretty nearly optimal. (For known mean vectors and covariance matrices it is optimal.) And if the data are normal with equal covariance matrices, those optimal quadratic discriminate functions reduce to linear functions of y. But if y is normal, $\tilde{y}$ is certainly *not* normal and applying traditional LDA methods to $\tilde{y}$ is unlikely to agree with QDA. Nonetheless, LDA on $\tilde{y}$ is some form of quadratic discrimination.

10.6 Stepwise LDA

One interesting problem in allocation is the choice of variables. Including variables that have no ability to discriminate among populations can only muddy the issues involved. By analogy with multiple regression, one might expect to find advantages to allocation procedures based solely on variables with high discriminatory power. In multiple regression, methods for eliminating predictor variables are either di-

rectly based on, or closely related to, testing whether exclusion of the variables hurts the regression model. In other words, a test is performed of whether, given the included variables, the excluded variables contain any additional information for prediction. In LDA, methods for choosing discriminatory variables such as *stepwise discrimination* are based on testing whether, given the included variables, the excluded variables contain any additional information for discrimination among the populations. We have noted that LDA is closely related to the multivariate one-way ANOVA model. The background for tests of additional information in multivaratiate linear models is beyond the scope of this book but is examined in *ALM*. Suffice it to say that in this context they can be performed by testing a one-way ACOVA model against the corresponding no-group-effects regression model. A one-way ACOVA model can be viewed as simply a regression model that allows different intercepts for different groups (*not* entirely different regressions for different groups), whereas the corresponding no-group-effects regression model is the ACOVA model but with only a single intercept.

EXAMPLE 10.6.1. We now illustrate the process of stepwise LDA using data given by Lubischew (1962). He considered the problem of discriminating among three populations of flea-beetles within the genus *Chaetocnema*. Six variables were given: y_1, the width, in microns, of the first joint of the first tarsus, y_2, the same measurement for the second joint, y_3, the maximum width, in microns, of the aedeagus in the fore part, y_4, the front angle, in units of 7.5 degrees, of the aedeagus, y_5, the maximum width of the head, in .01 millimeter units, between the external edges of the eyes, and y_6, the width of the aedeagus from the side, in microns. In addition, Lubischew mentions that $r_{12} \equiv y_1/y_2$ is very good for discriminating between one of the species and the other two. The vector of dependent variables is taken as $y' = (y_1, y_2, y_3, y_4, y_5, y_6, r_{12})$. stepwise LDA is carried out by testing for additional information in the one-way MANOVA.

Evaluating the assumptions of a one-way MANOVA with three groups and seven dependent variables is a daunting task. There are three 7×7 covariance matrices that should be roughly similar. To wit, there are $\binom{7}{2} = 21$ bivariate scatter plots to check for elliptical patterns. If the capability exists for the user, there are $\binom{7}{3} = 35$ three-dimensional plots to check. There are $3(7) = 21$ normal plots to evaluate the marginal distributions and at least some linear combinations of the variables should be evaluated for normality. Of course, if y_1 and y_2 are multivariate normal, the constructed variable r_{12} cannot be. However, it may be close enough for our purposes.

If the assumptions break down, it is difficult to know how to proceed. After any transformation, everything needs to be reevaluated, with no guarantee that things will have improved. It seems like the best bet for a transformation is some model-based system similar to the Box and Cox (1964) method (see Andrews, Gnanadesikan, and Warner, 1971 or *ANREG*).

For the most part, in this example, we will cross our fingers and hope for the best. In other words, we will rely on the robustness of the procedure. While it is certainly true that the P values used in stepwise LDA should typically not be taken at face

value (this is true for almost any variable selection technique), the P values can be viewed as simply a one-to-one transformation of the test statistics. Thus, decisions based on P values are based on the relative sizes of corresponding test statistics. The test statistics are reasonable even without the assumption of multivariate normality so, from this point of view, multivariate normality is not a crucial issue.

Although the properties of formal tests can be greatly affected by the invalidity of the MANOVA assumptions, crude but valid evaluations can still be made based on the test statistics. This is often the most that we have any right to expect from multivariate procedures. For univariate models, Scheffé (1959, Chapter 10) gives an excellent discussion of the effects of invalid assumptions on formal tests.

The three species of flea-beetles considered will be referred to as simply A, B, and C and indexed as 1, 2, and 3, respectively. There are 21 observations on species A with

$$\bar{y}'_{1\cdot} = (183.1, 129.6, 51.2, 146.2, 14.1, 104.9, 1.41)$$

and

$$S_1 = \begin{bmatrix} 147.5 & 66.64 & 18.53 & 15.08 & -5.21 & 14.21 & 0.406 \\ 66.64 & 51.25 & 11.55 & 2.48 & -1.81 & 3.09 & -0.044 \\ 18.53 & 11.55 & 4.99 & 5.85 & -0.524 & 5.49 & 0.017 \\ 15.08 & 2.48 & 5.85 & 31.66 & -0.969 & 15.63 & 0.090 \\ -5.21 & -1.81 & -0.524 & -0.969 & 0.791 & -1.99 & -0.021 \\ 14.21 & 3.09 & 5.49 & 15.63 & -1.99 & 38.23 & 0.078 \\ 0.406 & -0.044 & 0.017 & 0.090 & -0.021 & 0.078 & 0.0036 \end{bmatrix}.$$

Species B has 31 observations with

$$\bar{y}'_{2\cdot} = (201.0, 119.3, 48.9, 124.6, 14.3, 81.0, 1.69)$$

and

$$S_2 = \begin{bmatrix} 222.1 & 63.40 & 22.60 & 30.37 & 4.37 & 29.47 & 0.926 \\ 63.40 & 44.16 & 7.91 & 11.82 & 0.337 & 11.47 & -0.100 \\ 22.60 & 7.91 & 5.52 & 5.69 & 0.005 & 4.23 & 0.075 \\ 30.37 & 11.82 & 5.69 & 21.37 & -0.327 & 11.70 & 0.088 \\ 4.37 & 0.337 & 0.005 & -0.327 & 1.21 & 1.27 & 0.029 \\ 29.47 & 11.47 & 4.23 & 11.70 & 1.27 & 79.73 & 0.085 \\ 0.926 & -0.100 & 0.075 & 0.088 & 0.029 & 0.085 & 0.009 \end{bmatrix}.$$

For species C, there are 22 observations with

$$\bar{y}'_{3\cdot} = (138.2, 125.1, 51.6, 138.3, 10.1, 106.6, 1.11)$$

and

$$S_3 = \begin{bmatrix} 87.33 & 44.55 & 20.53 & 19.17 & -0.736 & 15.29 & 0.301 \\ 44.55 & 73.04 & 15.71 & 14.02 & -0.390 & 21.23 & -0.267 \\ 20.53 & 15.71 & 8.06 & 8.21 & -0.294 & 4.97 & 0.027 \\ 19.17 & 14.02 & 8.21 & 2.16 & -0.502 & 7.93 & 0.027 \\ 0.736 & -0.390 & -0.294 & -0.502 & 0.944 & 0.277 & -0.002 \\ 15.29 & 21.23 & 4.97 & 7.93 & 0.277 & 34.25 & -0.061 \\ 0.301 & -0.267 & 0.027 & 0.027 & -0.002 & -0.061 & 0.0046 \end{bmatrix}.$$

The pooled estimate of the covariance is a weighted average of S_1, S_2, and S_3, with approximately 50% more weight on S_2 than on the other estimates.

Although, typically, backward elimination is to be preferred to forward selection in stepwise procedures, it is illustrative to demonstrate forward selection on these data. We will begin by making a very rigorous requirement for inclusion: variables will be included if the P value for adding them is 0.01 or less.

The first step in forward selection consists of performing the univariate one-way ANOVA F tests for each variable.

Step 1: Statistics for entry, $df = 2,71$.

Variable	F_{obs}	$\Pr[F > F_{obs}]$
y_1	160.339	0.0001
y_2	12.499	0.0001
y_3	9.659	0.0002
y_4	134.353	0.0001
y_5	129.633	0.0001
y_6	101.314	0.0001
r_{12}	351.292	0.0001

The P values are all sufficiently small to warrant inclusion of the variables. By far the largest F statistic, and thus the smallest P value, is for r_{12}, so this is the first variable included for use in discrimination. Note that r_{12} is the variable constructed by Lubischew.

The second and all subsequent steps of the procedure involve performing a one-way analysis of covariance for each variable not yet included. For the second step, the sole covariate is r_{12}, and a test is made for treatment effects in the analysis of covariance model. For the dependent variables y_1 through y_6, the results are as follows.

Step 2: Statistics for entry, $df = 2,70$.

Variable	F_{obs}	$\Pr[F > F_{obs}]$
y_1	9.904	0.0002
y_2	8.642	0.0004
y_3	6.386	0.0028
y_4	87.926	0.0001
y_5	30.549	0.0001
y_6	28.679	0.0001

The largest F statistic is for y_4, and the corresponding P value is less than 0.01, so y_4 is included for discrimination.

At the third step, both r_{12} and y_4 are used as covariates in a one-way analysis of covariance. Again, the F tests for treatment differences are performed.

Step 3: Statistics for entry, $df = 2, 69$.

Variable	F_{obs}	$\Pr[F > F_{obs}]$
y_1	2.773	0.0694
y_2	3.281	0.0436
y_3	6.962	0.0018
y_5	24.779	0.0001
y_6	3.340	0.0412

Variable y_5 is included for discrimination. Note the large difference between the F statistic for y_5 and that for the other variables. There is an order-of-magnitude difference between the abilities of the r_{12}, y_4, and y_5 to discriminate and the abilities of the other variables. Considering the questionable validity of formal tests, this is an important point. It should also be mentioned that this conclusion is based on one sequence of models. There is a possibility that other sequences would lead to different conclusions about the relative importance of the variables. In fact, it would be desirable to check all models or, better yet, have an algorithm to identify the best models.

Step 4 simply adds weight to our conclusions of the previous paragraph. In performing the analysis of covariance with three covariates, none of the variables considered have the very large F statistics seen earlier.

Step 4: Statistics for entry, $df = 2, 68$.

Variable	F_{obs}	$\Pr[F > F_{obs}]$
y_1	1.985	0.1453
y_2	2.567	0.0842
y_3	3.455	0.0372
y_6	3.359	0.0406

Any rule that terminates forward selection when all P values exceed .0371 will stop the selection process at Step 4. In particular, our stringent stopping rule based on P values of 0.01 terminates here.

In practice, it is much more common to use a stopping rule based on P values of 0.05, 0.10, or 0.15. By any of these rules, we would add variable y_3 and continue checking variables. This leads to Step 5 and the corresponding F statistics.

Step 5: Statistics for entry, $df = 2, 67$.

Variable	F_{obs}	$\Pr[F > F_{obs}]$
y_1	7.040	0.0017
y_2	8.836	0.0004
y_6	3.392	0.0395

Surprisingly, adding y_3 has changed things dramatically. While the F statistic for y_6 is essentially unchanged, the F values for y_1 and y_2 have more than tripled. Of course, we are still not seeing the huge F statistics that were encountered earlier, but

apparently one can discriminate much better with y_3 and either y_1 or y_2 than would be expected from the performance of any of these variables individually. This is precisely the sort of thing that is very easily missed by forward selection procedures and one of the main reasons why they are considered to be poor methods for model selection. Forward selection does have advantages. In particular, it is cheap and it is able to accommodate huge numbers of variables.

The stepwise procedure finishes off with two final steps. Variable y_2 was added in the previous step. The results from Step 6 are as follows.

Step 6: Statistics for entry, $df = 2, 66$.

Variable	F_{obs}	$\Pr[F > F_{obs}]$
y_1	0.827	0.4418
y_6	3.758	0.0285

Variable y_6 is added if our stopping rule is not extremely stringent. This leaves just y_1 to be evaluated.

Step 7: Statistics for entry, $df = 2, 65$.

Variable	F_{obs}	$\Pr[F > F_{obs}]$
y_1	0.907	0.4088

By any standard y_1 would not be included. Of course, r_{12} is the ratio of y_1 and y_2, so it is not surprising that there is no need for all three variables. A forward selection procedure that does not include r_{12} would simply include all of the variables.

We have learned that r_{12}, by itself, is a powerful discriminator. The variables r_{12}, y_4, and y_5, when taken together, have major discriminatory powers. Variable y_3, taken together with either y_1 or y_2 and the previous three variables, may provide substantial help in discrimination.

Finally, y_6 may also contribute to distinguishing among the populations. Most of these conclusions are visible from the Table 10.3 that summarizes the results of the forward selection.

Table 10.3 Summary of Forward Selection

Step	Variable Entered	F_{obs}	$\Pr[F > F_{obs}]$
1	r_{12}	351.292	0.0001
2	y_4	87.926	0.0001
3	y_5	24.779	0.0001
4	y_3	3.455	0.0372
5	y_2	8.836	0.0004
6	y_6	3.758	0.0285

It is also of interest to see the results of a multivariate analysis of variance for all of the variables included at each step. For example, after Step 3, variables r_{12}, y_4, and y_5 were included for discrimination. The likelihood ratio test statistic for no

group effects in the one-way MANOVA is $\Lambda = 0.0152$ which is a very small, hence very significant, number. (The likelihood ratio test statistic in this context is usually referred to as *Wilks' Lambda*.) Table 10.4 lists the results of such tests for each step in the process. Based on their P values, all of the variables added had substantial discriminatory power. Thus, it is not surprising that the Λ statistics in Table 10.4 decrease as each variable is added.

Table 10.4 Forward Stepwise Discrimination: MANOVA Tests

Step	Variable Entered	LRTS Λ_{obs}	$P =$ $\Pr[\Lambda < \Lambda_{obs}]$
1	r_{12}	0.09178070	0.0001
2	y_4	0.02613227	0.0001
3	y_5	0.01520881	0.0001
4	y_3	0.01380601	0.0001
5	y_2	0.01092445	0.0001
6	y_6	0.00980745	0.0001

To perform backward elimination you begin with the seven ACOVA models that use one of the 7 variables as the dependent variable with the other 6 acting as predictors. Drop the variable that gives the smallest F statistic for group differences. Using the remaining 6 variables, fit six ACOVA models each with 5 predictors variables wherein the first dropped variable is not considered for inclusion in any way. Proceed until all the F statistics for group differences are large.

In practice, decisions about the practical discriminatory power of variables should not rest solely on the P values. After all, the P values are often unreliable. Other methods, such as the graphical methods presented in the next section, should be used in determining the practical usefulness of results based on multivariate normal distribution theory. □

10.7 Linear Discrimination Coordinates

As mentioned earlier, one is typically interested in the clarity of classification. This can be investigated by examining the posterior probabilities, the entire likelihood function, or the entire set of Mahalanobis distances. It is done by computing the allocation measures for each element of the data set. The allocation measure can be estimated either by the entire data set or the data set having deleted the case currently being allocated. To many people, the second, cross-validatory, approach is more appealing.

An alternative approach to examining the clarity of discrimination is through the use of *linear discrimination coordinates*. This approach derives from the work of Fisher (1938) and Rao (1948, 1952). It consists of redefining the coordinate system in $\mathbf{R}^q$ in such a way that the different treatment groups in the one-way ANOVA have,

in some sense, maximum separation in each coordinate. The clarity of discrimination can then be examined visually by inspecting one-, two-, or three-dimensional plots of the data. In these plots, cases are identified by their populations. If the new coordinate system is effective, observations from the same population should be clustered together and distinct populations should be well-separated.

It is standard practice to redefine the coordinate system by taking linear combinations of the original variables. It is also standard practice to define the new coordinate system sequentially. In particular, the first coordinate is chosen to maximize the separation between the groups. The second coordinate maximizes the separation between the groups given that the second linear combination is uncorrelated with the first. The third maximizes the separation given that the linear combination is uncorrelated with the first two. Subsequent coordinates are defined similarly. In the following discussion, we assume a constant covariance matrix for the t groups. It remains to define what precisely is meant by "maximum separation of the groups." Details are given in ALM. Here we just hit the highlights.

Recall that with equal covariance matrices, the data available in a discriminant analysis fit a multivariate one-way ANOVA,

$$y'_{ij} = \mu'_i + \varepsilon'_{ij},$$

thus

$$E = \sum_{i=1}^{t} \sum_{j=1}^{N_i} (y_{ij} - \bar{y}_{i\cdot})(y_{ij} - \bar{y}_{i\cdot})'$$

and

$$H = \sum_{i=1}^{t} N_i (\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})(\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})'.$$

Also, define

$$H_* = \sum_{i=1}^{t} (\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})(\bar{y}_{i\cdot} - \bar{y}_{\cdot\cdot})'.$$

The linear discrimination coordinates are based on eigenvectors associated with E and either H or H_*. We will examine the use of H in detail. For reasons discussed in *ALM*, some people prefer to use H_*.

For any vector $y = (y_1, \ldots, y_q)'$, the first linear discrimination coordinate is defined by

$$y' a_1,$$

where the vector a_1 is chosen so that the univariate one-way ANOVA model

$$y'_{ij} a_1 = \mu_i + \varepsilon_{ij}.$$

has the largest possible F statistic for testing equality of group effects. Intuitively, the linear combination of the variables that maximizes the F statistic must have the greatest separation between groups. A one-dimensional plot of the n elements

of $y'_{ij}a_1$ shows the maximum separation between groups that can be achieved in a one-dimensional plot.

The second linear discrimination coordinate maximizes the F statistic for testing groups in

$$y'_{ij}a_2 = \mu_i + \varepsilon_{ij}.$$

subject to the requirement that the estimated covariance between the $y'_{ij}a_1$s and $y'_{ij}a_2$s is zero. The covariance condition is specifically,

$$a'_1 S a_2 = 0.$$

A one-dimensional plot of the $y'_{ij}a_2$s illustrates visually the separation in the groups. Even more productively, the n ordered pairs that are $y'_{ij}(a_1, a_2)$ can be plotted to illustrate the discrimination achieved by the first two linear discrimination coordinates.

For $h = 3, \ldots, r(H)$ the hth linear discriminant coordinate is

$$y' a_h,$$

where a_h maximizes the F statistic for testing groups in

$$y'_{ij}a_h = \mu_i + \varepsilon_{ij}$$

but is maximized subject to the covariance condition

$$a'_h S a_i = 0 \qquad i = 1, 2, \ldots, h-1.$$

Unfortunately, the discrimination coordinates are not uniquely defined. Given a vector a_h, any scalar multiple of a_h also satisfies the requirements listed earlier. One way to avoid the nonuniqueness is to impose another condition. The most commonly used extra condition is that $a'_h S a_h = 1$. Alas, even this does not quite solve the uniqueness problem because $-a_h$ has the same properties as a_h. *ALM* shows that the linear discrimination coordinate vectors a_i, $i = 1, \ldots, q$ are eigenvectors of $E^{-1}H$ and that simple visual inspection of the transformed data is appropriate.

EXAMPLE 10.7.1. *One-Way Analysis of Variance with Repeated Measures*
A study was conducted to examine the effects of two drugs on heart rates. Thirty women were randomly divided into three groups of ten. An injection was given to each person. Depending on their group, women received either a placebo, drug A, or drug B indexed as $i = 1, 2, 3$, respectively. Repeated measurements of their heart rates were taken beginning at two minutes after the injection and at five minute intervals thereafter. Four measurements were taken on each individual, thus we have $t = 3$ and $q = 4$. The data are given in Table 10.5. They are from *ALM* where they were examined for multivariate normality and equal covariance matrices. The data seem to satisfy the assumptions.

The linear discrimination coordinates are defined by a matrix of eigenvectors of $E^{-1}H$. One choice is

Table 10.5 Heart rate data.

	DRUG											
	Placebo				A				B			
TIME	1	2	3	4	1	2	3	4	1	2	3	4
SUBJECT												
1	80	77	73	69	81	81	82	82	76	83	85	79
2	64	66	68	71	82	83	80	81	75	81	85	73
3	75	73	73	69	81	77	80	80	75	82	80	77
4	72	70	74	73	84	86	85	85	68	73	72	69
5	74	74	71	67	88	90	88	86	78	87	86	77
6	71	71	72	70	83	82	86	85	81	85	81	74
7	76	78	74	71	85	83	87	86	67	73	75	66
8	73	68	64	64	81	85	86	85	68	73	73	66
9	76	73	74	76	87	89	87	82	68	75	79	69
10	77	78	77	73	77	75	73	77	73	78	80	70

$$A = \begin{bmatrix} 0.739 & 0.382 & 0.581 & 0.158 \\ -0.586 & -0.323 & -0.741 & 0.543 \\ -0.353 & -0.234 & 0.792 & -0.375 \\ 0.627 & -0.184 & -0.531 & -0.218 \end{bmatrix}.$$

The columns of A define four new data sets $y_{ij}'a_1$, $y_{ij}'a_2$, $y_{ij}'a_3$, and $y_{ij}'a_4$ but remember that eigenvectors are not uniquely defined. Different software often give different eigenvectors but (when the eigenvalues are unique) they only vary by a scale factor, so the differences typically do not matter (unless you are trying to reproduce existing results). If we perform an analysis of variance on each variable, we get F statistics for discriminating between groups. All have 2 degrees of freedom in the numerator and 27 in the denominator.

Variable	F
Ya_1	74.52
Ya_2	19.47
Ya_3	0.0
Ya_4	0.0

As advertised, the F statistics are nonincreasing. The first two F statistics clearly establish that there are group differences in the first two coordinates. The last two F statistics are zero because with three groups there are 2 degrees of freedom for treatments, so H is a 4×4 matrix of rank 2. Only two of the linear discrimination coordinates can have positive F statistics. This issue is discussed in more detail in *ALM*.

The big advantage of linear discrimination coordinates is that they allow us to plot the data in ways that let us visualize the separation in the groups. Figure 10.3 shows two plots that display the first discrimination coordinate values for each population. The R plotting software placed the populations in different positions. Note that the degree of separation is substantial and about the same for all three groups.

The edges of the middle group are close to the edges of the other groups. The placebo has one observation that is consistent with drug A.

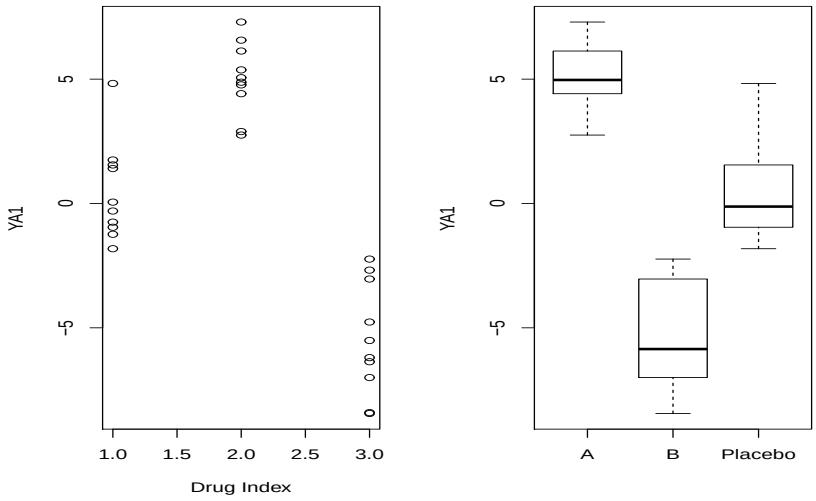

Fig. 10.3 Plots of the heart rate data in the first linear discrimination coordinate.

Figure 10.4 is similar to Figure 10.3 except that it plots the data in the second discrimination coordinate. Note that in the second coordinate it is very difficult to distinguish between drugs A and B. The placebo is separated from the other groups, but there is more overlap around the edges than was present in the first coordinate.

Figure 10.5 is a scatter plot of the data in the first two discrimination coordinates. Together, the separation is much clearer than in either of the individual coordinates. There is still one observation from drug A that is difficult to distinguish from the placebo group but, other than that, the groups are very well-separated. That the one observation from drug A is similar to the placebo is a conclusion based on the Euclidean distance of the point from the centers of the groups for drug A and the placebo. It is not clear that Euclidean distances are appropriate, but that is shown in *ALM*. □

10.8 Linear Discrimination

In linear models we consider a vector of predictor variables x and linear models $x'\beta$. The key feature of a linear model is that $x'\beta$ is a linear function of the predictor variables, whatever the predictor variables may be. The predictor variables are *not*

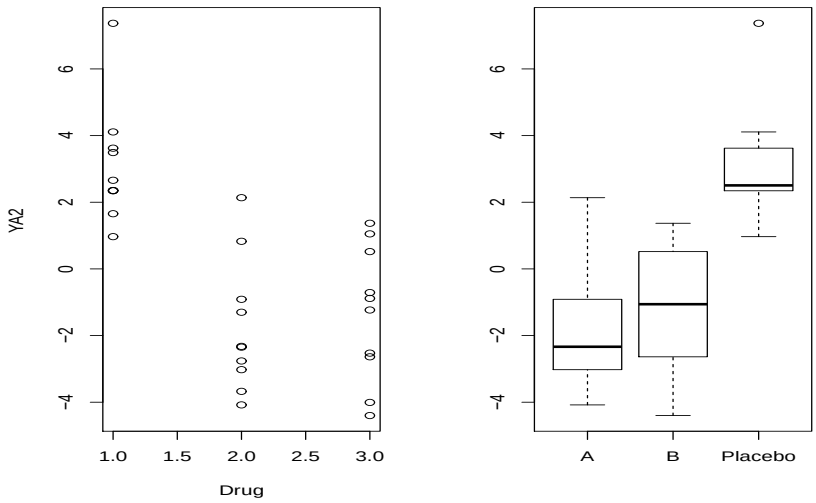

Fig. 10.4 Plots of the heart rate data in the second discrimination coordinate.

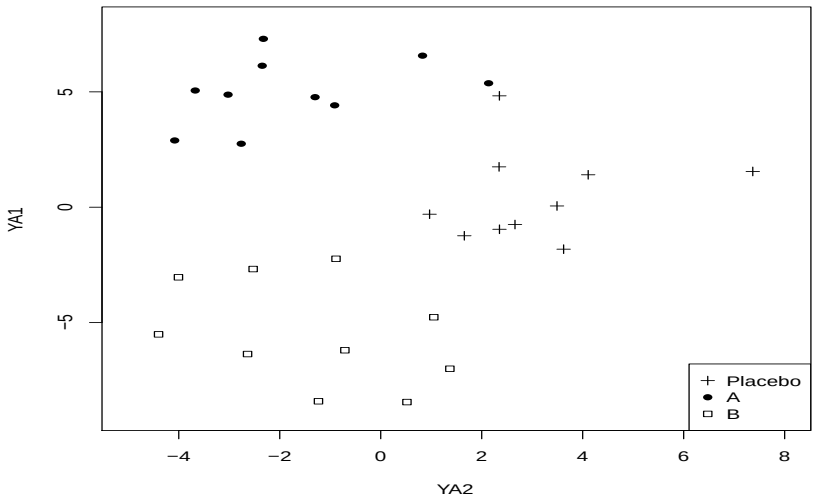

Fig. 10.5 Scatter plot of the heart rate data in the first two linear discrimination coordinates.

restricted to be a set of measurements originally taken on a collection of observational units. Chapters 4 and 8 examined the large variety of transformations that can be applied to the original measurements that make linear models far more flexible.

In this chapter, our predictor variables have been denoted y, rather than x. We have examined the traditional linear and quadratic discrimination methods LDA and QDA. We have pointed out that both of these methods are linear in the sense that they involve linear combinations of predictor variables, it is just that quadratic discrimination includes squares and cross-products of the original measurements as additional predictors. *The key aspect of LDA and QDA is not that they involve linear or quadratic functions of the original measurements but that the methods assume that the original data have a multivariate normal distribution and involve estimating appropriate normal densities.* If the data really are multivariate normal, no other procedure will give much of an improvement on LDA or QDA. If the data are not multivariate normal, nor easily transformed to multivariate normal, alternative discrimination procedures should be able to improve on them.

Obviously one *could* apply LDA to a y vector that includes not only the original measurements but also squares and cross-products (or other transformations) and LDA would probably give reasonable results, even though such a y vector could not possibly have a multivariate normal distribution. In the next section we focus on linear discrimination methods that do not assume multivariate normality. These methods include both logistic discrimination and support vector machines (SVMs). All such methods admit as predictor variables, transformations of the original measurements.

EXAMPLE 10.8.1. Figure 10.6 contains data of a form that have often been used to sell support vector machines because neither LDA nor QDA can distinguish the two populations whereas SVMs separate them easily. *Such a claim is comparing apples with oranges.* It is true that the most naive forms of LDA and QDA cannot separate them. Nor can the most naive forms of logistic or probit regression separate them. But the most naive form of SVMs cannot separate them either. If you transform the data into polar coordinates, you get the data representation in Figure 10.7. It is trivial to separate the data in Figure 10.7 with a (nearly) vertical line and pretty much any standard method will do it. The differences among the methods are that they may pick different nearly vertical lines to do the separation. And in this case, how much do you really care which line you use?

Obviously, no line is going to separate the data in Figure 10.6. To separate the groups with a line, you have to transform the data. The main difference is that computer programs for SVMs have a selection of transformation methods built into them by allowing the specification of an appropriate reproducing kernel. Logistic regression programs could also allow the specification of an appropriate reproducing kernel, but typically they do not. For the more traditional methods, like LDA and QDA, it seems that the transformations need to be specified explicitly. (Even though you *can* apply LDA and QDA to transformed data, it is rarely a good idea unless the transformation is designed to make the data more multivariate normal.) □

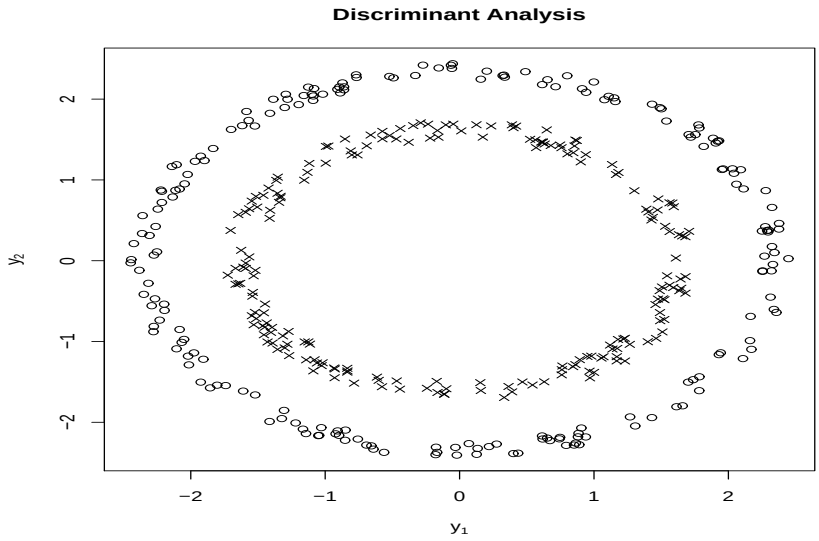

Fig. 10.6 Doughnut data.

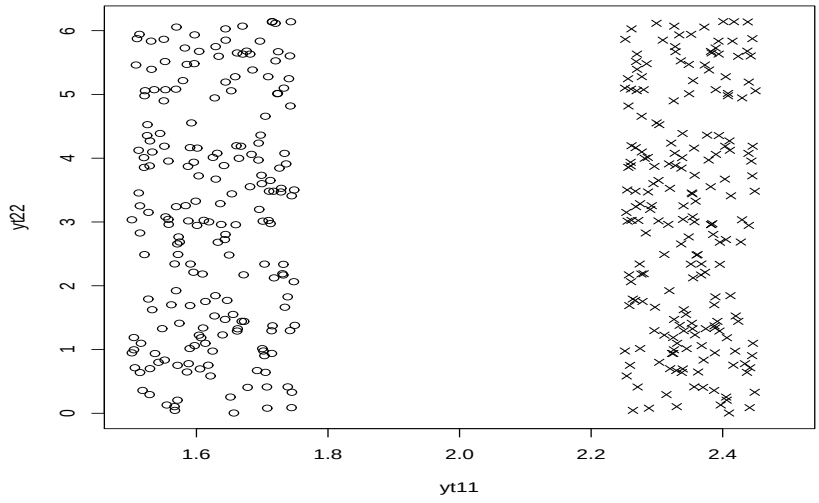

Fig. 10.7 Doughnut data in polar coordinates.

10.9 Modified Binary Regression

Binary discrimination shares the same predictive goal as binary regression but it involves using a different type of data and therefore requires outside information about the prevalences of the groups within the overall population. Instead of sampling from the joint distribution of the dependent and predictor variables (or the conditional distribution of the dependent variable given the predictor variables), discrimination involves sampling from the conditional distribution of the predictor variables given the "dependent" variable. With a different data collection scheme, binary regression methods need to be modified before they are appropriate for discrimination data. Binary regression and discrimination are restricted to $t = 2$ populations. ANREG-II, Section 21.9 uses log-linear models to illustrate the extension of logistic discrimination to the full $t = 3$ Cushing's Syndrome data.

To be consistent with the notation of this chapter, we need to change the binary regression notation used in Chapter 9. Here, y is a q dimensional vector of predictor variables (rather than the $d-1$ vector $\mathbf{x}$ of Chapter 9) and we will now use z (rather than the y of Chapter 9) to denote group membership. To predict z from y we need to envision a joint distribution for (z, y'). Call the density $f(z,y)$. There are some easily resolved mathematical issues related to z being discrete and y typically being continuous. Denote the marginal density (*prevalence*) of z as $\pi(z)$, the conditional density of z given y as $\pi(z|y)$, the marginal density of y as $f(y)$, and the conditional density of y given z as $f(y|z)$. Thus far we have treated z as fixed, not random, and we wrote i in place of z. In binary regression and discrimination we denote the two z groups as 0 and 1. (Thus far the two groups have been labeled 1 and 2.)

Since we are focused on predicting z, in both regression [sampling from either $f(z,y)$ or $\pi(z|y)$] and discrimination [sampling from $f(y|z)$], our goal is to estimate $\pi(z|y)$. (A sample from the joint distribution $f(z,y)$ can be viewed as a sample from either conditional scheme.) Regression data gives direct information on $\pi(z|y)$. Discrimination data gives direct information on $f(y|z)$ but only indirect information on $\pi(z|y)$. Using Bayes' Theorem with only two groups, the posterior probabilities in (10.1.4) become

$$\pi(1|y) = \frac{f(y|1)\pi(1)}{f(y|1)\pi(1) + f(y|0)\pi(0)}; \qquad \pi(0|y) = 1 - \pi(1|y).$$

If we know the prevalence distribution $\pi(z)$, discrimination data allow us to estimate $f(y|z)$ and, indirectly, $\pi(z|y)$.

Bayes theorem also determines the posterior odds for seeing $z = 1$,

$$O(1|y) \equiv \frac{\pi(1|y)}{\pi(0|y)} = \frac{f(y|1)}{f(y|0)} \frac{\pi(1)}{\pi(0)}. \tag{1}$$

Note that

$$\pi(1|y) = \frac{O(1|y)}{1 + O(1|y)}.$$

We will see that binomial regression methods applied to discrimination data are easily adjusted to give appropriate posterior odds. The formulae for logistic regression is particularly nice. (There is no obvious way of making the necessary adjustment for SVMs.)

Define the $q+1$ dimensional row vectors $x' \equiv (1, y')$ and $\beta' = (\beta_0, \beta_*')$ so that $x'\beta = \beta_0 + y'\beta_*$. Further combining the notation of this chapter with the binary regression notation, the probability of seeing an observation in group 1 given the predictors is defined interchangeably as

$$\pi(1|y) \equiv p(y) \equiv p(x).$$

Binary generalized linear models further assume $p(y) = F(x'\beta)$ for a known, invertible cdf F. The densities $f(z,y)$, $f(y|z)$, and $f(y)$ bear *no relationship* to the cdf F used in specifying binary generalized linear models.

Binary regression assumes (conditionally) independent observations

$$z_h \sim \text{Bin}[1, p(y_h)]; \qquad h = 1, \ldots, n$$

with the associated likelihood function, cf. Chapter 9.1. The likelihood function for discrimination data is

$$\prod_{h=1}^{n} f(y_h|z_h) \equiv \prod_{i=0}^{1} \prod_{j=1}^{N_i} f(y_{ij}|i),$$

where there are N_i observations y_{ij} on group i. Christensen (1997) argues that the logistic regression likelihood function can be viewed as a *partial likelihood* function for discrimination data. (The argument is actually for the log-linear model that is equivalent to the logistic model.) This allows one to use binary regression methods to estimate the densities $f(y|i)$ associated with discrimination data but it requires a correction for the prevalences implicitly assumed when treating discrimination data as if it were regression data.

Unconditional data constitute a random sample of (z, y') values. Clearly, one can estimate the marginal probabilities (prevalences) of the groups from unconditional data. The obvious estimate of $\pi(1)$ is the number of observed values $z = 1$ divided by the sample size, $N_1/(N_1 + N_0)$.

If the ys are preselected and one samples from $z|y$, i.e. the usual regression sampling scheme, there is no statistical basis for estimating $\pi(1)$ without knowing the marginal density $f(y)$. If the y_h were sampled from the appropriate distribution for y, it would make (z_h, y_h') a random sample from the appropriate joint distribution, and z_h a random sample with the prevalence probabilities, which makes $N_1/(N_1 + N_0)$ the obvious estimate of $\pi(1)$. Any estimation scheme that puts equal weight on the losses associated with each observation is implicitly treating the y_hs as a random sample and using $N_1/(N_1 + N_0)$ as an estimate of $\pi(1)$. All of our binary estimation schemes in Chapter 9 used equal weights, so in particular using $N_1/(N_1 + N_0)$ as an estimate of $\pi(1)$ is implicit in SVMs. (In Chapter 9 we had little interest in estimating $\pi(1)$.)

There is no possible way to estimate $\pi(1)$ from discrimination data. A value for $\pi(1)$ has to be obtained from some source outside the data before it becomes possible to estimate $\pi(1|y)$. If we want to use regression estimates computed from discriminant data, we need to correct for the implicit use of $N_1/(N_1+N_0)$ as an estimate of $\pi(1)$.

In discriminating between two groups, the maximum likelihood allocation can be based on the relative densities (likelihood ratio) $f(y|1)/f(y|0)$. We now derive the likelihood ratio estimate, say, $\hat{f}(y|1)/\hat{f}(y|0)$ from some arbitrary fitted binary regression estimates $\tilde{\pi}(1|y) \equiv \tilde{p}(y) \equiv \tilde{p}(x)$ that incorporate the inappropriate prevalence $\tilde{\pi}(1) = N_1/(N_1+N_0)$. From these we further obtain an estimate $\hat{\pi}(1|y)$ of $\pi(1|y)$ for an appropriate prior $\pi(1)$.

The binary regression estimated posterior odds are

$$\tilde{O}(y) = \frac{\tilde{\pi}(1|y)}{\tilde{\pi}(0|y)} = \frac{\hat{f}(y|1)}{\hat{f}(y|0)} \frac{\tilde{\pi}(1)}{\tilde{\pi}(0)} = \frac{\hat{f}(y|1)}{\hat{f}(y|0)} \frac{N_1}{N_0}.$$

These induce the estimated relative likelihoods

$$\frac{\hat{f}(y|1)}{\hat{f}(y|0)} = \frac{\tilde{\pi}(1|y)}{\tilde{\pi}(0|y)} \frac{N_0}{N_1}.$$

To obtain the actual estimated posterior probabilities $\hat{\pi}(i|y)$ for discrimination using the actual prevalences $\pi(i)$, use the estimated odds

$$\hat{O}(y) \equiv \frac{\hat{\pi}(1|y)}{\hat{\pi}(0|y)} = \frac{\hat{f}(y|1)}{\hat{f}(y|0)} \frac{\pi(1)}{\pi(0)} = \frac{\tilde{\pi}(1|y)}{\tilde{\pi}(0|y)} \frac{N_0}{N_1} \frac{\pi(1)}{\pi(0)}.$$

The discrimination odds $\hat{O}$ give discrimination probabilities through $\hat{\pi} = \hat{O}/(1+\hat{O})$.

When fitting a logistic discrimination from a logistic regression estimate $\tilde{\beta} = (\tilde{\beta}_0, \tilde{\beta}_*')'$, the odds take a particularly nice form. Since

$$\log\left[\tilde{O}(y)\right] \equiv x'\tilde{\beta},$$

we get

$$\begin{aligned}
\log\left[\hat{O}(y)\right] &= x'\tilde{\beta} - \log\left(\frac{N_1}{N_0}\right) + \log\left(\frac{\pi(1)}{\pi(0)}\right) \\
&= y'\tilde{\beta}_* + \left[\tilde{\beta}_0 - \log\left(\frac{N_1}{N_0}\right) + \log\left(\frac{\pi(1)}{\pi(0)}\right)\right] \\
&= x'\hat{\beta}
\end{aligned}$$

where

$$\hat{\beta}' \equiv \left(\tilde{\beta}_0 - \log\left(\frac{N_1}{N_0}\right) + \log\left(\frac{\pi(1)}{\pi(0)}\right),\ \tilde{\beta}_*'\right).$$

This leads to

$$\hat{\pi}(1|y) = e^{x'\hat{\beta}} \Big/ \left[1 + e^{x'\hat{\beta}}\right].$$

Logistic discrimination defines a hyperplane of y values by $x'\hat{\beta} \equiv \hat{\beta}_0 + y'\hat{\beta}_* = 0$ which corresponds to $0.5 = \hat{\pi}(1|y)$. If $x'\hat{\beta} > 0$, $\hat{\pi}(1|y) > 0.5$. If $x'\hat{\beta} < 0$, $\hat{\pi}(1|y) < 0.5$. The logistic regression model also defines a hyperplane of y values defined by $0.5 = \tilde{p}(x'\tilde{\beta})$ which is equivalent to $0 = x'\tilde{\beta}$ or $-\tilde{\beta}_0 = y'\tilde{\beta}_*$. Because $\hat{\beta}_* = \tilde{\beta}_*$, these hyperplanes are parallel in q dimensions, but unless the prevalences $\pi(i)$ are proportional to the sample sizes N_i, the regression and discrimination hyperplanes are distinct.

There are several ways of generalizing logistic discrimination to handle $t > 2$. Christensen (1997, 2015) focuses on the fact that logit/logistic models are actually log-linear models and that the appropriate log-linear model can easily be generalized to handle more than two populations. In particular, he illustrates for $t = 3$ how to turn estimated odds into allocations. Without probability estimates, SVMs often rely on performing all of the $\binom{t}{2}$ binary discrimination problems and "voting" for a winning allocation. Voting could also be used when probability estimates exist, not that I would do that.

EXAMPLE 10.9.1. Figures 10.8, 10.9, and 10.10 are discrimination versions of the regression Figures 9.1, 9.3, and 9.5. Figures 10.8, 10.9, and 10.10 have dropped the probit regression curves and replaced them with LDA and QDA as appropriate. They have also replaced the logistic regression curves with logistic discrimination curves. The discrimination curves are all based on $\pi(1) = 0.5$. The SVM curves are unchanged from the previous plots because I do not know how to make the necessary adjustments.

In Figure 9.1 the logistic and probit regression lines were almost on top of each other. The logistic discrimination line in Figure 10.8 is parallel but lower than the regression line. This is consistent with the fact that putting equal prior probabilities (prevalences) on the groups makes the carcinoma group more probable relative to the prior probabilities proportional to sample sizes built into the logistic regression line which makes bilateral twice as probable as carcinoma. The LDA line turns out to be nearly parallel to the logistic discrimination line but is even lower. From the limited amount of data, there is no reason to think that the covariance matrices of the two groups are equal; the spreads of the points are not at all similar. Despite this, the LDA does not do a bad job on these data. The SVM is unchanged.

In Figure 9.3 the logistic and probit regression parabolas were almost on top of each other. The logistic discrimination parabolas in Figure 10.9 should have the same shape as the logistic regression parabolas but move closer to the bilateral group. In fact, I cannot *see* any difference between the logistic regression and logistic discrimination parabolas for these data. The QDA parabolas have a radically different shape than the logistic parabolas but, although they do not completely separate the two groups, they do not do a bad job. The SVM is unchanged and the curve that is visible on the plot has a different shape from both other methods. Figure 10.10 is the same as Figure 10.9 except that it replaces the default SVM with the one from Figure 9.5 that has a reduced tuning parameter (increased cost). □

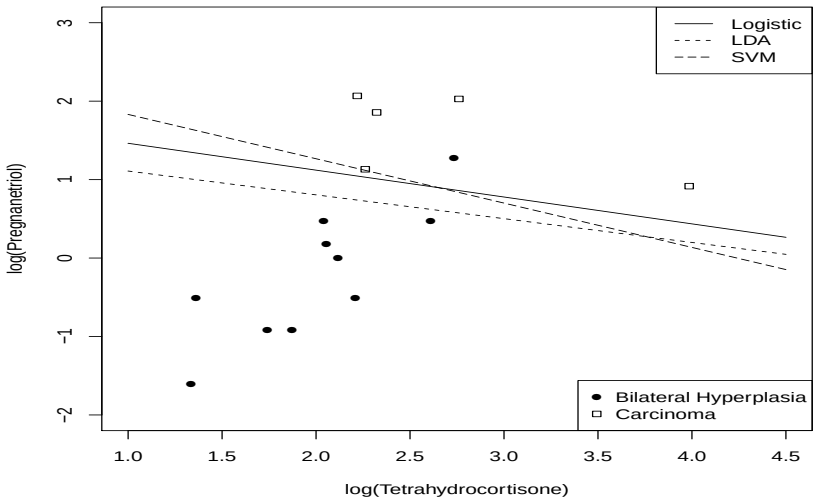

Fig. 10.8 Logistic discrimination, LDA, and an SVM: Cushing's Syndrome data (subset).

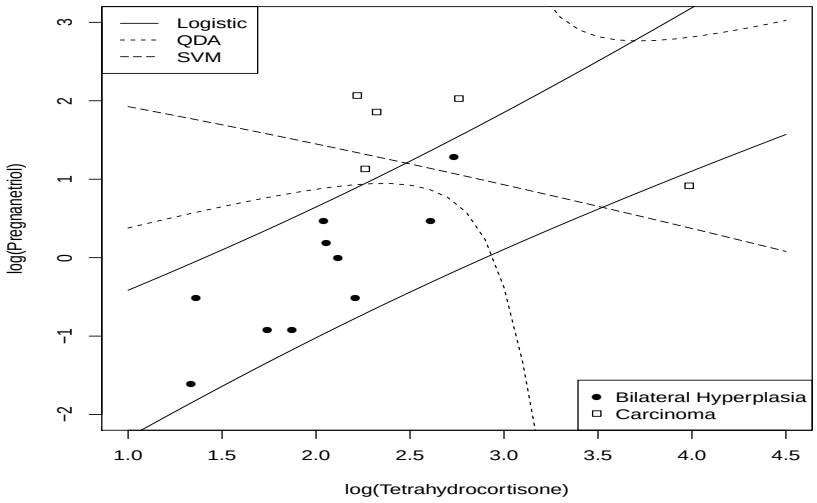

Fig. 10.9 Quadratic model logistic discrimination, QDA, and an SVM: Cushing's Syndrome data (subset).

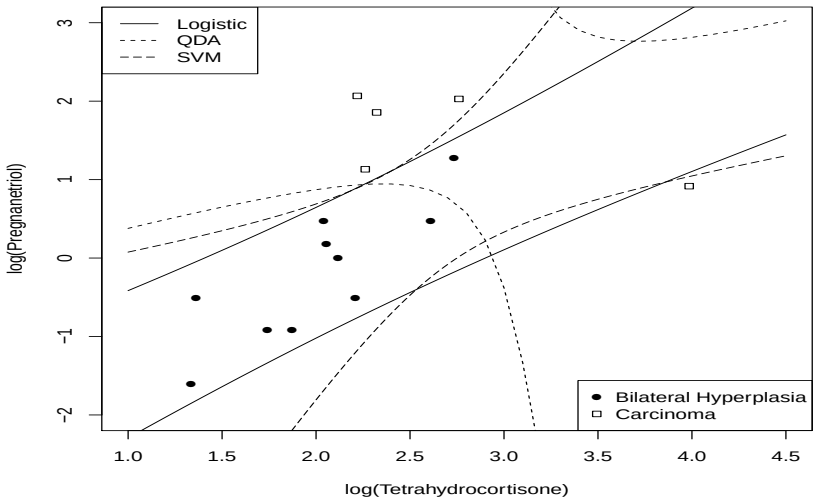

Fig. 10.10 Quadratic model logistic discrimination, QDA, and an SVM with reduced tuning parameter: Cushing's Syndrome data (subset).

My code for constructing these figures also contains code for producing tables of estimated posterior probabilities for logistic discrimination that are similar to those presented in Chapter 9 and in Christensen (1997, 2015). I personally find the lack of probability estimates a substantial disadvantage to SVMs. For binary regression data, although I do not see any advantages to SVMs over binomial regression (other than software advantages), SVMs clearly give reasonable answers. But for discrimination data, unless you think that using prevalences proportional to sample sizes is reasonable, SVM will not give reasonable answers.

10.10 Exercises

EXERCISE 10.10.1. Consider the data of Example 10.3.1. Suppose a person has heart rate measurements of $y = (84, 82, 80, 69)'$.
(a) Using normal theory linear discrimination, what is the estimated maximum likelihood allocation for this person?
(b) Using normal theory quadratic discrimination, what is the estimated maximum likelihood allocation for this person?
(c) If the two drugs have equal prior probabilities but the placebo is twice as probable as the drugs, what is the estimated maximum posterior probability allocation?

(d) What is the optimal allocation using only the first two linear discrimination coordinates?

EXERCISE 10.10.2. In the motion picture *Diary of a Mad Turtle* the main character, played by Richard Benjamin Kingsley, claims to be able to tell a female turtle by a quick glance at her carapace. Based on the data of Exercise 10.6.1, do you believe that it is possible to accurately identify a turtle's sex based on its shell? Explain. Include graphical evaluation of the linear discrimination coordinates.

EXERCISE 10.10.3. Using the data of Exercise 10.6.3, do a stepwise LDA to distinguish among the thyroxin, thiouracil, and control rat populations based on their weights at various times. To which group is a rat with the following series of weights most likely to belong: $(56, 75, 104, 114, 138)$?

EXERCISE 10.10.4. Lachenbruch (1975) presents information on four groups of junior technical college students from greater London. The information consists of summary statistics for the performance of the groups on arithmetic, English, and form relations tests that were given in the last year of secondary school. The four groups are Engineering, Building, Art, and Commerce students. The sample means are:

	Engineering	Building	Art	Commerce
Arithmetic (y_1)	27.88	20.65	15.01	24.38
English (y_2)	98.36	85.43	80.31	94.94
Form Relations (y_3)	33.60	31.51	32.01	26.69
Sample Size	404	400	258	286

The pooled estimate of the covariance matrix is

$$S_p = \begin{bmatrix} 55.58 & 33.77 & 11.66 \\ 33.77 & 360.04 & 14.53 \\ 11.66 & 14.53 & 69.21 \end{bmatrix}.$$

What advice could you give to a student planning to go to a junior technical college who just achieved scores of $(22, 90, 31)'$?

EXERCISE 10.10.5. Suppose the concern in Exercise 10.10.4 is minimizing the cost to society of allocating students to the various programs of study. The great bureaucrat in the sky, who works on the top floor of the tallest building in Whitehall, has determined that the costs of classification are as follows:

Cost		Optimal Study Program			
		Engineering	Building	Art	Commerce
Allocated	Engineering	1	2	8	2
Study	Building	4	2	7	3
Program	Art	8	7	4	4
	Commerce	4	3	5	2

Evaluate the program of study that the bureaucrat thinks is appropriate for the student from Exercise 10.10.4.

EXERCISE 10.10.6. Show that the Mahalanobis distance is invariant under affine transformations $z = Ay + b$ of the random vector y when A is nonsingular.

EXERCISE 10.10.7. Let y be an observation from one of two normal populations that have means of μ_1 and μ_2 and common covariance matrix Σ. Define $\lambda' = (\mu_1 - \mu_2)'\Sigma^{-1}$.
(a) Show that, under linear discrimination, y is allocated to population 1 if and only if

$$\lambda' y - \lambda' \frac{1}{2}(\mu_1 + \mu_2) > 0.$$

(b) Show that if y is from population 1,

$$\mathrm{E}(\lambda' y) - \lambda' \frac{1}{2}(\mu_1 + \mu_2) > 0$$

and if y is from population 2,

$$\mathrm{E}(\lambda' y) - \lambda' \frac{1}{2}(\mu_1 + \mu_2) < 0.$$

EXERCISE 10.10.8. Consider a two group allocation problem in which the prior probabilities are $\pi(1) = \pi(2) = 0.5$ and the sampling distributions are exponential, namely

$$f(y|i) = \theta_i e^{-\theta_i y}, \quad y \geq 0.$$

Find the optimal allocation rule. Assume a cost structure where $c(i|j)$ is zero for $i = j$ and one otherwise. The *total probability of misclassification* for an allocation rule is precisely the Bayes risk of the allocation rule under this cost structure. Let $\delta(y)$ be an allocation rule. The frequentist risk for the true population j is $R(j,\delta) = \int c(\delta(y)|j) f(y|j) dy$ and the Bayes risk is $r(p,\delta) = \sum_{j=1}^{t} R(j,\delta)\pi(j)$. See Berger (1985, Section 1.3) for more on risk functions. Find the total probability of misclassification for the optimal rule.

EXERCISE 10.10.9. Suppose that the distributions for two populations are bivariate normal with the same covariance matrix. For $\pi(1) = \pi(2) = 0.5$, find the value of the correlation coefficient that minimizes the total probability of misclassification. The total probability of misclassification is defined in Exercise 10.10.8.

Chapter 11
Dimension Reduction

Abstract This chapter introduces the theory and application of principal components, classical multidimensional scaling, and factor analysis. Principal components seek to effectively summarize high dimensional data as lower dimensional scores. Multidimensional scaling gives a visual representation of points when all we know about the points are the distances separating them. Classical multidimensional scaling is seen to be an application of principal components when the distances are standard Euclidean distances. Ideas similar to principal components, i.e. the singular value decomposition for a non-square matrix, can but directly applied to a data matrix to compress the data. Principal components and factor analysis are often used for similar purposes but their theoretical background is quite different. The linear discrimination coordinates of the previous chapter are sometimes considered a form of dimension reduction.

Suppose that observations are available on q variables. When q is quite large it can be very difficult to grasp the relationships among the many variables. It might be convenient if the variables could be reduced to a more manageable number. Clearly, it is easier to work with 4 or 5 variables than with, say, 25. (In the era of big data, perhaps I should be arguing that 400 or 500 variables are easier to work with than 2500.) Of course, one cannot reasonably expect to get a substantial reduction in dimensionality without some loss of information. We want to minimize that loss. Assuming that a reduction in dimensionality is desirable, how can it be performed efficiently? One reasonable method is to choose a small number of linear combinations of the variables based on their ability to reproduce the entire set of variables. In effect, we want to create a few new variables that are best able to predict the original variables. *Principal component analysis (PCA)* finds linear combinations of the original variables that are best linear predictors of the full set of variables. This predictive approach to *dimensionality reduction* seems intuitively reasonable. We emphasize this interpretation of principal component analysis rather than the traditional motivation of finding linear combinations that account for most of the variability in the data. We have previously used principal components as a tool for finding alternative estimates in Section 4.1.

More recently, *independent component analysis (ICA)* has become a popular method of data reduction. Hyvärinen, Karhunen, and Oja (2001) introduce the subject as both a generalization of principal components and as a generalization of factor analysis. (The latter seems more appropriate to me.) The R (package and) program `fastICA` begins by computing the principal components and obtains the "independent components" from them.

Principal components are similar in spirit to the linear discrimination coordinates discussed in Chapter 10. Principal components form a new coordinate system for $\mathbf{R}^q$. These coordinates are defined sequentially so that they are uncorrelated but, subject to being uncorrelated, they maximize the ability to predict the original dependent variables. In practice, only the first few coordinates are used to represent the entire vector of dependent variables.

Section 1 presents several alternative derivations for theoretical principal components including both predictive and nonpredictive motivations. Section 2 examines the use of sample principal components. Section 3 introduces *classical multidimensional scaling (CMDS)*, which seeks to plot the locations of cases when one only knows the distances between the cases. The reason for examining CMDS here is their close relation to PCA. Sections 4 and 5 introduce some *nonstatistical* methods used for *compressing data matrices*. The final section examines *factor analysis*. Although many people consider principal component analysis a special case of factor analysis, in fact their theoretical bases are quite different.

11.1 The Theory of Principal Components

There are several equivalent definitions of principal components. We begin with the predictive definition. Principal components are a sequence of linear combinations of the variable vector y. Each linear combination has maximum capability to predict the full set of variables subject to the condition that each combination is uncorrelated with the previous linear combinations.

What does it mean to be a linear combination that has maximum capability to predict the full set of variables? Details of best linear prediction can be found in *PA* Chapter 6 and, more extensively, in *ALM*. Briefly, when predicting a random vector $y = (y_1, \ldots, y_q)'$ from another random vector $x = (x_1, \ldots, x_{p-1})'$, the *best linear predictor (BLP)* is a linear function of x that minimizes $\mathrm{E}\{[y - f(x)]'[y - f(x)]\}$. This is the expected squared distance between the vector we want to predict and what we are using to predict it with. In particular if we let

$$\begin{aligned} \mathrm{E}(y) &= \mu_y & \qquad \mathrm{E}(x) &= \mu_x \\ \mathrm{Cov}(y) &= V_{yy} & \qquad \mathrm{Cov}(x) &= V_{xx} \end{aligned}$$

and

$$\mathrm{Cov}(y,x) = V_{yx} = V_{xy}',$$

then the best linear predictor, also called the linear expectation, is

$$\hat{E}(y|x) \equiv \mu_y + B'(x - \mu_x),$$

where $B_{(p-1)\times q}$ is a solution

$$V_{xx}B = V_{xy}.$$

When V_{xx} is nonsingular, $B = V_{xx}^{-1}V_{xy}$ and $\hat{E}(y|x) \equiv \mu_y + V_{yx}V_{xx}^{-1}(x - \mu_x)$.

In this chapter there is little need for consideration of predictor variables x and we change notation to

$$\mu \equiv \mathrm{E}(y); \qquad \Sigma \equiv \mathrm{Cov}(y).$$

For derivations of the following results see *ALM*. For a given scalar random variable $a'y$, the best linear predictor of y is

$$\begin{aligned}\hat{E}(y|a'y) &= \mu + \mathrm{Cov}(y, a'y)[\mathrm{Var}(a'y)]^{-1}(a'y - a'\mu) \\ &= \mu + \Sigma a[a'\Sigma a]^{-1}a'(y - \mu),\end{aligned}$$

but we want to find the best choice of a, the one whose BLP does the best job of predicting y. Call this best choice a_1. It is some work to show that the best choice a_1 is any eigenvector corresponding to the largest eigenvalue of Σ.

Next we want to find a linear combination $a_2'y$ so that $a_2'y$ does the best job of predicting y subject to the condition that this second linear combination is uncorrelated with the first, i.e.,

$$0 = \mathrm{Cov}(a_2'y, a_1'y) = a_2'\Sigma a_1.$$

It turns out that the best choice of a_2 is any eigenvector corresponding to the second largest eigenvalue of Σ. Here I have assumed that the largest and second largest eigenvalues are different, but there is little problem if they are not.

We continue in this way. For $h = 1, \ldots, q$ we find a linear combination $a_h'y$ so that $a_h'y$ does the best job of predicting y subject to the condition that this linear combination is uncorrelated with the previous ones, i.e.,

$$0 = \mathrm{Cov}(a_h'y, a_j'y) = a_h'\Sigma a_j, \qquad j = 1, \ldots, h-1.$$

(I have described this process with an implicit assumption that the q eigenvalues of Σ are distinct. If the largest eigenvalue has multiplicity greater than 1, the second principal component is determined by any eigenvector for the largest eigenvalue that is orthogonal to [uncorrelated with] the first chosen eigenvector. If the multiplicity is greater than 2, the third is determined by any eigenvector orthogonal to the first 2. Similar ideas continue to apply for any eigenvalues with multiplicities greater than 1.)

Now suppose we consider the first r of these *principal component variables*, $a_1'y, \ldots, a_r'y$ and consider a different set of linear combinations $b_1'y, \ldots, b_r'y$. We can also show that the joint prediction of y based on $a_1'y, \ldots, a_r'y$ is at least as good as the joint prediction of y based on $b_1'y, \ldots, b_r'y$. In other words, $\hat{E}(y|a_1'y, \ldots, a_r'y)$ does at least as good a job of predicting y as $\hat{E}(y|b_1'y, \ldots, b_r'y)$ does. The second approach to principal components consists of maximizing the joint predictive ability. There are

lots of good choices for selecting good joint predictors, but you cannot do any better than the sequentially defined principal components.

A third approach to principal components is based on trying to maximize the variability in linear combinations. In other words, pick a_1 to maximize

$$\text{Var}(a_1'y) = a_1'\Sigma a_1.$$

It has never been intuitive to me why you would want to maximize the variance. It is somewhat intuitive that maximizing the variance is needed for getting good prediction. In any case, as it stands maximizing the variance does not make a lot of sense because $\text{Var}(10a_1'y) = 100\text{Var}(a_1'y)$, so without some other conditions you can never maximize the variance. Suffice it to say that after specifying appropriate conditions, this approach gives the same linear combinations $a_j'y$ as the sequential prediction approach.

Finally, principal components can be related to ellipsoids. If $y \sim N(\mu, \Sigma)$, the set of points that have constant likelihood (the same value of the density) fall on ellipsoids defined by Σ^{-1}. Figure 11.1 illustrates a density isobar for

$$y \sim N\left(\begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}1.0 & 0.9\\0.9 & 2.0\end{bmatrix}\right).$$

The major and minor axes are denoted a_1 and a_2, respectively. One can show that the axes of the ellipse are determined by the eigenvectors of Σ, so they also determine the sequential best predictors.

11.2 Sample Principal Components

In practice, the covariance matrix Σ is unknown, so the principal components cannot be computed. However, if a sample $y_1, \ldots, y_n$ of observations on y is available, sample principal components can be computed from either the sample covariance matrix

$$S = \sum_{i=1}^{n} (y_i - \bar{y}_\cdot)(y_i - \bar{y}_\cdot)' / (n-1)$$

or the sample correlation matrix

$$R = D^{-1/2} S D^{-1/2},$$

where $S = [s_{ij}]$ and $D = \text{Diag}(s_{11}, \ldots, s_{qq})$. Most often, the correlation matrix seems to be the appropriate choice.

Choose $a_1, \ldots, a_q$ as an orthonormal set of eigenvectors corresponding to the eigenvalues $\phi_1 \geq \cdots \geq \phi_q$ of S or R, respectively. Write $A = [a_1, \ldots, a_q]$ and, for $r \leq q$, $A_r = [a_1, \ldots, a_r]$. A vector w, rewritten in the principal component coordinate system, is $A'w$. Write the entire data set as

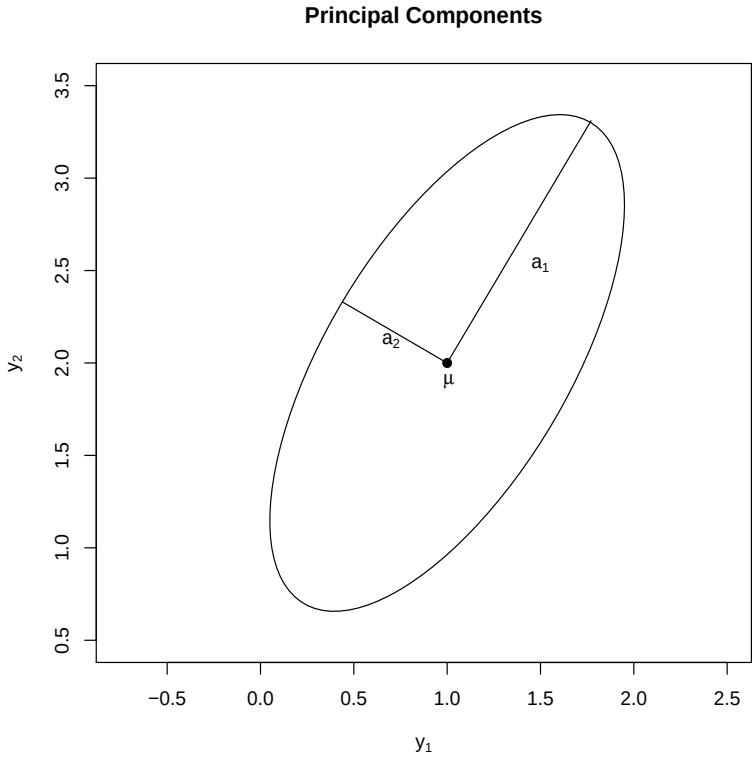

Fig. 11.1 Two-dimensional normal density isobar ($\mu = (1,2)'$, $\sigma_{11} = 1.0$, $\sigma_{12} = 0.9$, $\sigma_{22} = 2.0$) with major and minor axes.

$$Y = \begin{bmatrix} y_1' \\ \vdots \\ y_n' \end{bmatrix}.$$

Using S, the data in the principal component coordinate system are

$$YA.$$

If principal components are based on the correlation matrix R, the rescaled data are

$$\begin{bmatrix} z_1' \\ \vdots \\ z_n' \end{bmatrix} = Z = YD^{-1/2}$$

and the data in the principal component coordinate system are

$$ZA\,.$$

The point of principal component analysis is to reduce dimensionality. If the smallest ordered eigenvalues of S, $\phi_{r+1},\ldots,\phi_q$ are small, a random vector y with covariance matrix S can be predicted well by $A_r'y$. If the entire data set is transformed in this way, a principal component observation matrix is obtained,

$$YA_r = [Ya_1,\ldots,Ya_r], \tag{1}$$

where

$$Ya_i = \begin{bmatrix} a_i'y_1 \\ \vdots \\ a_i'y_n \end{bmatrix}.$$

The elements of the vector Ya_i consist of the ith principal component applied to each of the n observation vectors. The principal component observation matrix combines these vectors for each of the first r principal components.

The analysis of the data can be performed on the principal component observations with a minimal loss of information. This includes various plots and formal statistical techniques for the analysis of a sample from one population.

If you want to predict back the actual data you need to estimate the BLP. The estimate of $\hat{E}(y_i|A_r'y_i)$ is

$$\begin{aligned} \hat{y}_i &= \bar{y}_\cdot + SA_r[A_r'SA_r]^{-1}A_r'(y_i-\bar{y}_\cdot) \\ &= \bar{y}_\cdot + A_rD(\lambda_{(r)})[D(\lambda_{(r)})]^{-1}A_r'(y_i-\bar{y}_\cdot) \\ &= \bar{y}_\cdot + A_rA_r'(y_i-\bar{y}_\cdot), \end{aligned}$$

where $\lambda_{(r)}$ is the vector of the largest r eigenvalues of S. So for the entire data matrix,

$$\hat{Y} = J\bar{y}_\cdot' + [I-(1/n)J_n^n]YA_rA_r' = (1/n)J_n^nY + [I-(1/n)J_n^n]YA_rA_r'. \tag{2}$$

We will see in Section 4 that this involves performing a singular value decomposition on the $n\times q$ mean adjusted data matrix $[I-(1/n)J_n^n]Y$.

11.2.1 The Sample Prediction Error

A question that arises immediately is just how much information is lost by using r principal components rather than the entire data set. The value

$$100\sum_{j=r+1}^{q}\phi_j \Big/ \sum_{j=1}^{q}\phi_j$$

is the percentage of the maximum prediction error left unexplained by $\hat{E}(y_i|A_r'y_i)$, $i=1,\ldots,n$. Alternatively,

$$100 \sum_{j=1}^{r} \phi_j \Big/ \sum_{j=1}^{q} \phi_j$$

is the percentage of the maximum prediction error accounted for by $A_r' y$.

11.2.2 Using Principal Components

Principal components are designed to reduce dimensionality. They provide a number $r < q$ of linear combinations $a_i' y$ that maximize the ability to linearly predict the original random q-vector y. Thus they are appropriate to use when you are taking a random sample of ys.

Although the analysis of data can be performed on the principal component observations with a minimal loss of information, why accept any loss of information? Two possibilities come to mind. First, if q is very large, an analysis of all q variables may be untenable. If one must reduce the dimensionality before any work can proceed, principal components are a reasonable place to begin. However, it should be kept in mind that principal components are based on linear combinations of y and linear predictors of y. If the important structure in the data is nonlinear, principal components can totally miss that structure.

A second reason for giving up information is when you do not trust all of the information. In prediction theory the underlying idea is that a vector (y, x') would be randomly sampled from some population and we would seek to predict y based on x. Principal component regression (cf. *ANREG* or *PA*) can be used to reduce the dimensionality of x. But *PA* argues that using the principal components are an effective way to treat collinearity, even if you only have a sample from $y|x$. The main idea was that, with errors in the model matrix, directions corresponding to small eigenvalues are untrustworthy. In the present context, we might say that any statistical relationships depending on linear combinations that do not provide substantial power of prediction are questionable.

As a general principle, to reduce the dimensionality of data successfully you need to know ahead of time that it can be reduced. Whether it can be reduced depends on the goal of the analysis. The work involved in figuring out whether data reduction can be accomplished often negates the value of doing it. The situation is similar to that associated with Simpson's paradox in contingency table analysis (see Christensen, 1997, Section 3.1; Christensen, 2014). Valid inferences cannot always be obtained from a collapsed contingency table. To know whether valid inferences can be obtained, one needs to analyze the full table first. Having analyzed the full table, there may be little point in collapsing to a smaller-dimensional table. Often, it would be convenient to reduce a data set using principal components and then do a MANOVA on the reduced data. Unfortunately, about the only way to find out if that approach is reasonable is to examine the results of a MANOVA on the entire data set.

Principal components are well designed for data reduction within a given population. If there are samples available from several populations with the same covariance matrix, then the optimal data reduction will be the same for every group and can be estimated using the pooled covariance matrix. Note that this essentially requires doing a one-way MANOVA prior to the principal component analysis. If an initial MANOVA is required, you may wonder why one would bother to reduce the data having already done a significant analysis on the unreduced set.

In particular, my friend Ed Bedrick has pointed out that if y is sampled from more than one population, reducing the dimensionality, without having first accounted for the different populations, can cause you to loose the ability to distinguish the populations. With two normal populations having means μ_1 and μ_2, if the vector $\mu_1 - \mu_2$ is orthogonal to the eigenvectors that you are using to define your principal components, then there may be no information in your principal components capable of distinguishing the populations. For a two-dimensional illustration using one principal component see Figure 11.2. In fact, even if $\mu_1 - \mu_2$ is merely close to perpendicular with $a_1, \ldots, a_r$ you may lose most of the information for distinguishing the populations. (These ideas are explored in Exercise 14.3 of *ALM-III*.) Basically, the only way to tell that this is not happening is to do the one-way MANOVA prior to doing the principal component analysis. Again, there may be no point in doing principal components after doing MANOVA. Jolliffe (2002, Section 9.1) discusses this problem in more detail.

Data reduction is also closely related to a more nebulous idea, the identification of underlying factors that determine the observed data. For example, the vector y may consist of a battery of tests on a variety of subjects. One may seek to explain scores on the entire set of tests using a few key factors such as general intelligence, quantitative reasoning, verbal reasoning, and so forth. It is common practice to examine the principal components and try to interpret them as measuring some sort of underlying factor. Such interpretations are based on examination of the relative sizes of the elements of a_i. Although factor identification is commonly performed, it is, at least in some circles, quite controversial.

EXAMPLE 11.2.1. One of the well-traveled data sets in multivariate analysis is from Jolicoeur and Mosimann (1960) on the shell (carapace) sizes of painted turtles, cf. Table 11.1. Aspects of these data have been examined by Morrison (2004) and Johnson and Wichern (2007). The analysis is based on $10^{3/2}$ times the natural logs of the height, width, and length of the shells. Because all of the measurements are taken on a common scale, it may be reasonable to examine the sample covariance matrix rather than the sample correlation matrix. The point of this example is to illustrate the type of analysis commonly used in identifying factors. No claim is made that these procedures are reasonable.

For 24 males, the covariance matrix is

$$S = \begin{bmatrix} 6.773 & 6.005 & 8.160 \\ 6.005 & 6.417 & 8.019 \\ 8.160 & 8.019 & 11.072 \end{bmatrix}.$$

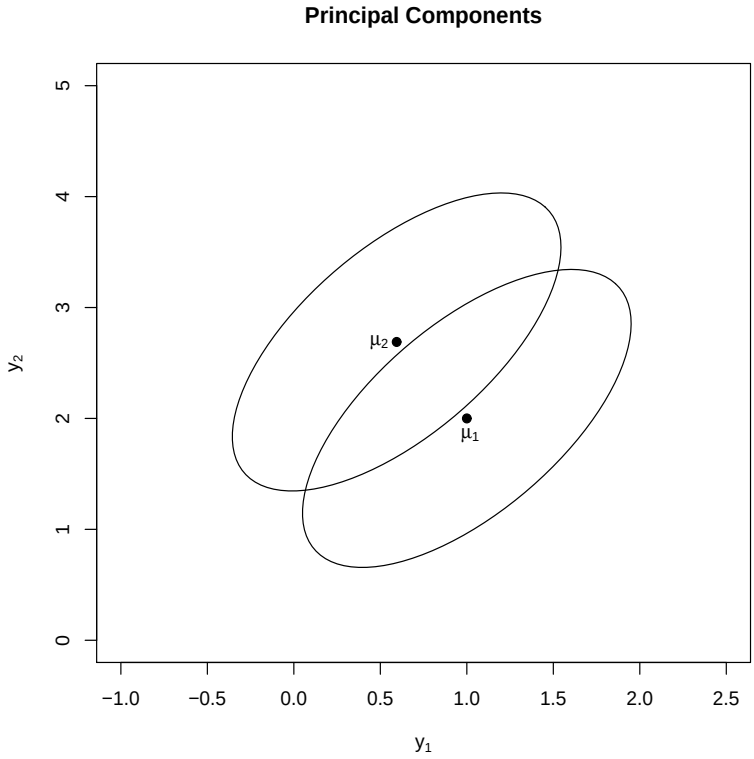

Fig. 11.2 Two populations indistinguishable in the first principal component.

Table 11.1 Carapace dimensions.

Female			Female			Male			Male		
Length	Width	Height	Length	Width	Height	Length	Width	Height	Length	Width	Height
98	81	38	138	98	51	93	74	37	116	90	43
103	84	38	138	99	51	94	78	35	117	90	41
103	86	42	141	105	53	96	80	35	117	91	41
105	86	42	147	108	57	101	84	39	119	93	41
109	88	44	149	107	55	102	85	38	120	89	40
123	92	50	153	107	56	103	81	37	120	93	44
123	95	46	155	115	63	104	83	39	121	95	42
133	99	51	155	117	60	106	83	39	125	93	45
133	102	51	158	115	62	107	82	38	127	96	45
133	102	51	159	118	63	112	89	40	128	95	45
134	100	48	162	124	61	113	88	40	131	95	46
136	102	49	177	132	67	114	86	40	135	106	47

The eigenvalues and corresponding eigenvectors for S are as follows.

ϕ_i	23.303	0.598	0.360
	a_1	a_2	a_3
$10^{3/2}\ln$(height)	0.523	0.788	−0.324
$10^{3/2}\ln$(width)	0.510	−0.594	−0.622
$10^{3/2}\ln$(length)	0.683	−0.159	0.713

Recall that eigenvectors are not uniquely defined. Eigenvectors of a matrix B corresponding to ϕ (along with the zero vector) constitute the null space of $B - \phi I$. Often, the null space has rank 1, in which case every eigenvector is a multiple of every other eigenvector. If we standardize the eigenvectors of S so that each has a maximum element of 1, we get the following eigenvectors.

	a_1	a_2	a_3
$10^{3/2}\ln$(height)	0.764	1	−0.451
$10^{3/2}\ln$(width)	0.747	−0.748	−0.876
$10^{3/2}\ln$(length)	1	−0.205	1
ϕ	23.30	0.60	0.36

The first principal component accounts for 100(23.30)/(23.30 + 0.60 + 0.36) = 96% of the predictive capability (variance) of the variables. The first two components account for 100(23.30 + 0.60)/(24.26) = 98.5% of the predictive capability (variance) of the variables. All the elements of a_1 are positive and approximately equal, so $a_1'y$ can be interpreted as a measure of overall size. The elements of a_2 are a large positive value for $10^{3/2}\ln$(height), a large negative value for $10^{3/2}\ln$(width), and a small value for $10^{3/2}\ln$(length). The component $a_2'y$ can be interpreted as a comparison of the ln(height) and the ln(width). Finally, if one considers the value $a_{31} = -0.451$ small relative to $a_{32} = -0.876$ and $a_{33} = 1$, one can interpret $a_3'y$ as a comparison of width versus length.

Interpretations such as these necessarily involve rounding values to make them more interpretable. The interpretations just given are actually appropriate for the three linear combinations of y, $b_1'y$, $b_2'y$, and $b_3'y$ that follow.

	b_1	b_2	b_3
$10^{3/2}\ln$(height)	1	1	0
$10^{3/2}\ln$(width)	1	−1	−1
$10^{3/2}\ln$(length)	1	0	1

The first interpreted component is

$$\begin{aligned} b_1'y &= 10^{3/2}\ln[(\text{height})(\text{width})(\text{length})] \\ &= 10^{3/2}\ln[\text{volume}], \end{aligned}$$

where the volume is that of a box. It is interesting to note that in this particular example, the first principal component can be interpreted without changing the coefficients of a_1.

$$\begin{aligned} a_1'y &= 10^{3/2}[0.764\ln(\text{height}) + 0.747\ln(\text{width}) + \ln(\text{length})] \\ &= 10^{3/2}\ln[(\text{height})^{0.764}(\text{width})^{0.747}(\text{length})]\,. \end{aligned}$$

The component $a_1'y$ can be thought of as measuring the log volume with adjustments made for the fact that painted turtle shells are somewhat curved and thus not a perfect box. Because the first principal component accounts for 96% of the predictive capability, to a very large extent, if you know this pseudovolume measurement, you know the height, length, and width.

In this example, we have sought to interpret the elements of the vectors a_i. Alternatively, one could base interpretations on estimates of the correlations $\text{Corr}(y_h, a_i'y)$. The estimates of $\text{Corr}(y_h, a_1'y)$ are very uniform, so they also suggest that a_1 is an overall size factor. □

Linear combinations $b_i'y$ that are determined by the effort to interpret principal components will be called *interpreted components*. Although it does not seem to be common practice, it is interesting to examine how well interpreted components predict the original data and compare that to how well the corresponding principal components predict the original data. As long as the interpreted components are linearly independent, a full set of q components will predict the original data perfectly. *Any* nonsingular transformation of y will predict y perfectly because it amounts to simply changing the coordinate system. If we restrict attention to r components, we know from the theoretical results on joint prediction that the interpreted components can predict no better than the actual principal components.

In general, to evaluate the predictive capability of r interpreted components, write $B_r = [b_1, \ldots, b_r]$ and compute

$$\sum_{i=1}^{n}[y_i - \hat{E}(y|B_r'y_i)]'[y_i - \hat{E}(y|B_r'y_i)] = (n-1)\text{tr}\{S - SB_r(B_r'SB_r)^{-1}B_r'S\}.$$

One hundred times this value divided by $(n-1)\sum_{i=1}^{q}\phi_i = (n-1)\text{tr}(S)$ gives If this is not much greater than the corresponding percentage for the first r principal components, the interpretations are to some extent validated.

EXAMPLE 11.2.2. As explained in *ALM*, the percentage of the predictive error unaccounted for by using the first two interpreted components in Example 11.2.1 is

$$\begin{aligned} \frac{100\text{tr}[S - SB_2(B_2'SB_2)^{-1}B_2'S]}{\text{tr}[S]} &= \frac{100(24.26 - 23.88)}{24.26} \\ &= \frac{100(0.38)}{24.26} \\ &= 1.6. \end{aligned}$$

Using the first two principal components,

$$\frac{100\text{tr}[S - SA_2(A_2'SA_2)^{-1}A_2'S]}{\text{tr}[S]} = \frac{100\sum_{j=1}^{3}\phi_j - \sum_{j=1}^{2}\phi_j}{\sum_{j=1}^{3}\phi_j}$$
$$= \frac{100(0.36)}{24.26}$$
$$= 1.5.$$

Thus, in this example, there is almost no loss of predictive capability by using the two interpreted components rather than the first two principal components. □

There is one aspect of principal component analysis that is often overlooked. It is possible that the most interesting components are those that have the least predictive power. Such components are taking on very similar values for all cases in the sample. It may be that these components can be used to characterize the population. Jolliffe (2002) has a fairly extensive discussion of uses for the *last few* principal components.

EXAMPLE 11.2.3. The smallest eigenvalue of S is 0.36 and corresponds to the linear combination

$$a_3'y = 10^{3/2}\left[-0.451\ln(\text{height}) - 0.876\ln(\text{width}) + \ln(\text{length})\right].$$

This linear combination accounts for only 1.5% of the variability in the data. It is essentially a constant. All male painted turtles in the sample have about the same value for this combination. The linear combination is a comparison of the ln-length with the ln-width and ln-height. This might be considered as a measurement of the general shape of the carapace. One would certainly be very suspicious of any new data that were supposedly the shell dimensions of a male painted turtle but which had a substantially different value of $a_3'y$. On the other hand, this should not be thought of as a discrimination tool except in the sense of identifying whether data are or are not consistent with the male painted turtle data. We have no evidence that other species of turtles will produce substantially different values of $a_3'y$. □

11.3 Classical Multidimensional Scaling

Multidimensional Scaling starts with a matrix containing the squared distances between a set of objects and produces a plot of the objects that reflects those distances. There are a number of methods for doing this but we restrict our attention to *Classical Multidimensional Scaling (CMDS)* because it reproduces the (mean corrected) sample principal component scores from only the squared distance matrix.

Consider a matrix $\mathscr{D}$ that consists of the squared distances between n objects. To obtain an r dimensional graphical representation of the objects, find eigenvectors

$a_1, \ldots, a_r$ of $[I-(1/n)JJ']\mathscr{D}[I-(1/n)JJ']$ corresponding to its r largest eigenvalues. Create the matrix

$$A_r \equiv [a_1, \cdots, a_r] \equiv \begin{bmatrix} \mathbf{a}_1' \\ \vdots \\ \mathbf{a}_n' \end{bmatrix}$$

and in r dimensions, plot the n vectors $\mathbf{a}_i$ to represent the n objects.

EXAMPLE 11.3.1. I computed the distances between the 21 observations in the Cushing's Syndrome data of Table 9.1 and applied CMDS to the squared distances. The result appears in Figure 11.3. The plot is just a recentering and rotation of the data appearing in Figure 10.1. (The data are rotated about 45 degrees counterclockwise.) This occurs because, as we will show, the 2-dimensional CMDS method is essentially just plotting the first two principal components of the data. Because the original data were two dimensional, the first two principal components contain all the information in the data, so we just get a recentered, rotated plot of the data. □

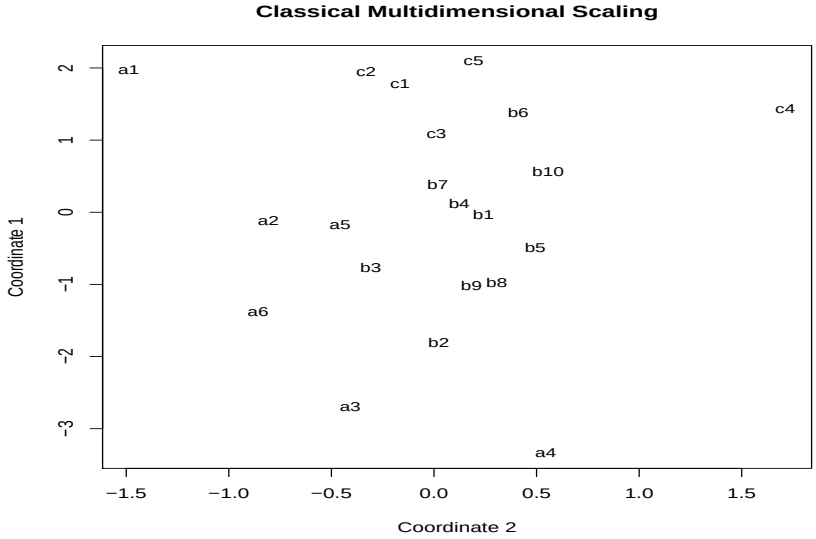

Fig. 11.3 Classical multidimensional scaling: Cushing syndrome data.

Starting with a data matrix

$$Y = \begin{bmatrix} y_1' \\ \vdots \\ y_n' \end{bmatrix}$$

that contains observations on n objects, we construct the squared Euclidian distance matrix $\mathscr{D}$ for which the ij element d_{ij} is the squared distance between y_i and y_j. In particular,

$$d_{ij} = (y_i - y_j)'(y_i - y_j).$$

We then establish that we can find the mean corrected sample principal components directly from $\mathscr{D}$. CMDS consists of plotting the mean corrected sample principal components.

Multiplying out the squared distances gives

$$d_{ij} \equiv (y_i - y_j)'(y_i - y_j) = y_i'y_i + y_j'y_j - 2y_i'y_j.$$

The squared distances are functions of the inner products and all of the inner products are given by

$$YY' = \begin{bmatrix} y_1'y_1 & y_1'y_2 & \cdots & y_1'y_n \\ y_2'y_1 & y_2'y_2 & \cdots & y_2'y_n \\ \vdots & \vdots & \ddots & \vdots \\ y_n'y_1 & y_n'y_2 & \cdots & y_n'y_n \end{bmatrix}.$$

Create a vector consisting of the squared lengths of the vectors of observations,

$$\mathbf{d} \equiv (y_1'y_1, y_2'y_2, \cdots, y_n'y_n)'.$$

It is not hard to see that the squared Euclidean distance matrix is

$$\mathscr{D} = \mathbf{d}J' + J\mathbf{d}' - 2YY'. \tag{1}$$

To relate this to principal components, we need to relate $\mathscr{D}$ to the sample covariance matrix

$$S \equiv \frac{1}{n-1} Y'[I - (1/n)JJ']Y = \frac{1}{n-1} \left\{[I - (1/n)JJ']Y\right\}' \left\{[I - (1/n)JJ']Y\right\}.$$

The key mathematical fact in relating squared distances to principal components is that if λ and b are an eigenvalue and eigenvector for $B'B$, then λ and Bb are an eigenvalue and eigenvector for BB'. In particular, if ϕ and a are an eigenvalue and eigenvector of $(n-1)S$, then ϕ and

$$\left\{[I - (1/n)JJ']Y\right\} a \tag{2}$$

are an eigenvalue and eigenvector of

$$\left\{[I - (1/n)JJ']Y\right\} \left\{[I - (1/n)JJ']Y\right\}' = [I - (1/n)JJ']YY'[I - (1/n)JJ']. \tag{3}$$

The formula in (2) is just a mean corrected principal component.

We can show

$$[I - (1/n)JJ'] \left(YY'\right) [I - (1/n)JJ'] = \frac{-1}{2}[I - (1/n)JJ']\mathscr{D}[I - (1/n)JJ'].$$

Since S and $(n-1)S$ have the same eigenvectors and the same ordering of the eigenvalues, we can compute the mean corrected principal components from the squared distance matrix.

While Example 11.3.1 is informative about what CMDS is doing, in practice multidimensional scaling is often used in situations where only a measure of distance between objects is available; not the raw data from which the distances were computed.

EXAMPLE 11.3.2. Lawley and Maxwell (1971) and Johnson and Wichern (1988) examine data on the examination scores of 220 male students. The dependent variable vector consists of test scores on (Gaelic, English, history, arithmetic, algebra, geometry). The correlation matrix is

$$R = \begin{bmatrix} 1.000 & 0.439 & 0.410 & 0.288 & 0.329 & 0.248 \\ 0.439 & 1.000 & 0.351 & 0.354 & 0.320 & 0.329 \\ 0.410 & 0.351 & 1.000 & 0.164 & 0.190 & 0.181 \\ 0.288 & 0.354 & 0.164 & 1.000 & 0.595 & 0.470 \\ 0.329 & 0.320 & 0.190 & 0.595 & 1.000 & 0.464 \\ 0.248 & 0.329 & 0.181 & 0.470 & 0.464 & 1.000 \end{bmatrix}.$$

We are going to treat the 6 tests as objects and use the correlation matrix as a measure of similarity between the objects. In particular, we used

$$\mathscr{D} = 1 - R$$

as our squared distance measure with results displayed in the top panel of Figure 11.4. The bottom panel of Figure 11.4 contains the CMDS representation when the distance is measured as 1 minus the squared correlation between the variables. (This does *not* involve multiplying the matrix R times itself.) □

11.4 Data Compression

A black and white photograph can be digitized as a matrix of pixels where the elements of the matrix are gray scale intensities. Such a matrix, say Y, might be 1000×800 requiring us to store 800,000 numerical values. The object of this section is simply to find an approximation

$$Y_{n\times q} \doteq W_{n\times r} H_{r\times q}; \qquad r \leq \min\{n, q\}$$

that allows us to store fewer values while still retaining enough of the original information to reconstruct Y in a useful fashion. For the black and white photo with 800,000 pixel intensities it might be enough just to store, say, $(1000 \times 50) + (50 \times 800) = 90,000$ values. Such decompositions are not unique. If you take an r dimensional orthonormal matrix O, it is clear that

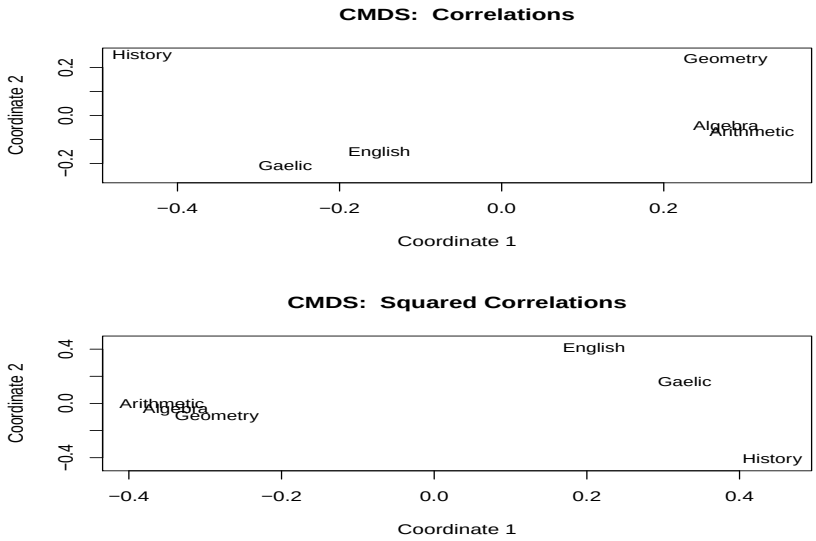

Fig. 11.4 Classical multidimensional scaling: Examination data.

$$WH = (WO)(O'H) \equiv \tilde{W}\tilde{H}.$$

There is nothing inherently statistical about the approximation in equation (1) or, indeed, in this entire section or the next (except for pointing out a relationship to the statistical procedure of principal components). The statistical version of this problem is know as Factor Analysis and is treated in the last section of this chapter. The following subsection presents an exact decomposition that can be used as the basis for compression (approximate decompositions). The second subsection presents an iterative procedure based on least squares.

11.4.1 The Singular Value Decomposition

The generalized Singular Value Decomposition, Proposition A.8.7, can be applied to an $n \times q$ data matrix Y. Let Y be an $n \times q$ matrix with rank s. Then Y can be written as

$$Y = ULV',$$

where U is $n \times s$, L is $s \times s$, V' is $s \times q$, and

$$L \equiv \text{Diag}(\lambda_j).$$

The λ_js are the positive square roots of the positive eigenvalues (singular values) of $Y'Y$ and YY'. The columns of V are s orthonormal eigenvectors of $Y'Y$ corresponding to the positive eigenvalues with

$$Y'YV = VL^2,$$

and the columns of U are s orthonormal eigenvectors of YY' with

$$YY'U = UL^2.$$

When computing these things it is important to remember that you should compute the eigenvalues and eigenvectors of either $Y'Y$ or YY', which ever has smaller dimensions, but not compute both. The eigenvalues and eigenvectors for the larger matrix can be determined from those of the smaller matrix. If you know V, take $U = YVL^{-1}$. If you know U, take $V = Y'UL^{-1}$

To solve the 1000×800 data matrix compression problem, typically $s = q$ so consider the 800 λ_j values from the Theorem. Many of them will be small, some number of them will be larger. If there are, say, $r = 50$ larger ones, then use the 50 corresponding columns of U and rows of V' to create new matrices $\tilde{U}$, $\tilde{V}'$, $\tilde{L}$ and $\tilde{Y} = \tilde{U}\tilde{L}\tilde{V}'$. If $\tilde{Y}$ does a good job of approximating Y you can save $\tilde{U}$, $\tilde{L}$, and $\tilde{V}'$ rather than the original Y using fewer resources. The number r of λ_j values needed for a good approximation depends on the particular application. Note that

$$Y\tilde{V} = ULV'\tilde{V} = UL\begin{bmatrix} I_r \\ 0 \end{bmatrix} = U\begin{bmatrix} \tilde{L} \\ 0 \end{bmatrix} = \tilde{U}\tilde{L},$$

which implies that

$$\tilde{Y} = \tilde{U}\tilde{L}\tilde{V}' = Y\tilde{V}\tilde{V}'.$$

Obviously, once you have the matrices $\tilde{U}$, $\tilde{L}$, and $\tilde{V}$, you have wide latitude in using them to define matrices W and H for approximating Y.

In equation (11.2.2) we pointed out that the estimated BLP of the data matrix Y based on r principal components is

$$\hat{Y} = J\bar{y}_\cdot' + [I - (1/n)J_n^n]YA_rA_r' = (1/n)J_n^nY + [I - (1/n)J_n^n]YA_rA_r'.$$

The second term of $\hat{Y}$ is just the singular value decomposition data compression of $[I - (1/n)J_n^n]Y$ based on the r largest eigenvalues.

11.4.2 Iterative Least Squares

The *iterative least squares* algorithm is based on alternating least squares estimates. Using notation similar to the previous subsection, think about fitting a multivariate linear model $Y = WH + e$ by first fixing W to estimate H and then by using that H as fixed to estimate a new W. Using the subscript j to indicate the jth column of

matrices, compute each $H_j^{(n+1)}$ to be the least squares estimate of H_j from minimizing $[Y_j - W^{(n)}H_j]'[Y_j - W^{(n)}H_j]$, $j = 1,\ldots,q$. Similarly, looking at the ith rows of matrixes, now take $w_i^{(n+1)}$ to be the least squares estimate of w_i from minimizing $[y_i - H^{(n+1)\prime}w_i]'[y_i - H^{(n+1)\prime}w_i]$.

For pure data compression problems, like the black and white photograph, this seems like a perfectly reasonable method although no advantages over the singular value decomposition leap out at me. However, the method seems less appropriate for many statistical applications. In statistical applications the rows of Y are typically taken as observations on independent individuals whereas the entries of a particular row, say, y_i' are taken to have some correlation matrix, say, $\Sigma_{q\times q}$ that is the same for every individual. Using least squares to find approximations to random vectors y_i with correlated components has little statistical justification.

For $r = 1$ this **may** be the same as partial least squares regression where W_0 is a vector of measurements that you want to relate to Y. Partial least squares was developed as an alternative to principal components regression and a generalization of the NIPALS algorithm.

11.4.3 NIPALS

NIPALS – *Nonlinear iterative partial least squares* turns an orthogonal basis for $X'X$ into an orthonormal basis of eigenvectors.

Take $\tilde{b}_0$. $b_0 = \tilde{b}_0\|\tilde{b}_0\|^{-1}$, $\tilde{b}_{n+1} = (X'X)b_n$, $b_{n+1} = \tilde{b}_n\|\tilde{b}_n\|^{-1}$. If $b_n \to b$, then $\|Xb\|^2 b = X'Xb$ and $\|b\| = 1$. Define $\lambda_b \equiv \|Xb\|^2$. In the descriptions I have seen they add another step $\hat{b}_{n+1} = Xb_n$ and $\tilde{b}_{n+1} = X'\hat{b}_{n+1}\|\hat{b}_n\|^{-1}$ that can obviously be consolidated.

Now take $\tilde{c}_0 \perp b$, $c_0 = \tilde{c}_0\|\tilde{c}_0\|^{-1}$, $\tilde{c}_{n+1} = (X'X - \lambda_b bb')c_n$, $c_{n+1} = \tilde{c}_n\|\tilde{c}_n\|^{-1}$. Note that an easy inductive proof gives that if $c_n \perp b$, then $c_{n+1} \perp b$. If $c_n \to c$, then $c \perp b$, so $\|Xc\|^2 c = (X'X - \lambda_b bb')c = X'Xc$ and $\|c\| = 1$ and define $\lambda_c \equiv \|Xc\|^2$. Although in theory the c_js are all orthogonal to b, in computational practice one might need to use Gram-Schmidt on them to deal with accumulating round off errors.

To find the next one, use the matrix $X'X - \lambda_b bb' - \lambda_c cc'$.

11.4.4 Partial Least Squares

Partial Least Squares (PLS) is an alternative to principle component regression. Consider a linear model

$$Y = X\beta + e.$$

Principal component regression replaces this with a reduced model

$$Y = T\gamma + e; \qquad C(T) \subset C(X),$$

in which the columns of T consist of principal components. PLS gives another way of finding a T matrix.

As I understand it the process begins as follows. Here we will find just the first column of T, say t and worry about subsequent columns afterwards.

Fit the multivariate model

$$X = Yw' + e$$

to obtain

$$\hat{w}' = (Y'Y)^{-1}Y'X$$

and

$$\hat{X} = Y(Y'Y)^{-1}Y'X.$$

Note that $\hat{X} = M_Y X$ where M_Y is the perpendicular projection operator onto $C(Y)$.

The second step is a key one. Fit

$$X = t\hat{w}' + e.$$

This can be written in multivariate linear model form as

$$X' = \hat{w}t' + e'$$

so we see that the least squares estimates are

$$\begin{aligned} \hat{t}' &= (\hat{w}'\hat{w})^{-1}\hat{w}'X' \\ &= \frac{Y'Y}{Y'XX'Y}Y'XX' \end{aligned}$$

or equivalently

$$\hat{t} = XX'Y\frac{Y'Y}{Y'XX'Y}.$$

We also obtain

$$\hat{X}' = X'YY'XX'(Y'XX'Y)^{-1}.$$

Note that $\hat{X}' = \hat{w}\hat{t} = M_{X'Y}X'$ where $M_{X'Y}$ is the perpendicular projection operator onto $C(X'Y)$.

The reason I consider the second step key is because I would argue that least squares estimation is not appropriate for the model $X' = \hat{w}t' + e'$. The rows of X' are not uncorrelated as in a regular multivariate linear model; it is the columns of X' that are uncorrelated. Thus each column of X can be fitting separately from the other columns but each column should be fitted using generalized (weighted) least squares based on the covariance matrix Σ. Of course, Σ will have to be estimated. More on this later.

The final step that I will address is fitting

$$X = \hat{t}b' + e.$$

The least squares estimates are

$$\begin{aligned}\hat{b}' &= (\hat{t}'\hat{t})^{-1}\hat{t}'X' \\ &= \frac{Y'XX'Y}{Y'Y}\frac{1}{Y'XX'XX'Y}Y'XX'X\end{aligned}$$

with

$$\hat{X} = XX'YY'XX'X\frac{1}{Y'XX'XX'Y}.$$

Note that $\hat{X} = M_{XX'Y}X$ where $M_{XX'Y}$ is the perpendicular projection operator onto $C(XX'Y)$.

To find a second column t in the matrix T we repeat the process using $(I - M_{XX'Y})X$ in place of X and $(I - M_{XX'Y})Y$ in place of Y. From properties of projection operators it is easily seen that the second column will end up being orthogonal to the first.

Now let us get back to the real issue: using $\hat{T}$ in the inverse regression model. I want to consider what happens if we decide to use only this first column $\hat{t}$. The full model based on only one column of C is

$$Y = Xp + e.$$

This yields the sum of squares for the model

$$Y'X(X'X)^{-1}X'Y$$

and the estimate

$$\hat{\xi}_0 = a_0'\hat{p} = a_0'(X'X)^{-1}X'Y$$

The model based on only $\hat{t}$ is

$$Y = \hat{t}v_1 + e. \tag{1}$$

This yields the sum of squares for the model

$$Y'\hat{t}(\hat{t}'\hat{t})^{-1}\hat{t}'Y = Y'XX'Y(Y'XX'XX'Y)^{-1}Y'XX'Y$$

and the estimate

$$\hat{\xi}_0 = a_0'X'Y\hat{v}_1 = a_0'X'Y(Y'XX'XX'Y)^{-1}Y'XX'Y.$$

Interestingly, there is a special case in which the full model and the Partial Least Squares answers are identical. If the is no collinearity in the data i.e., $X'X = I$ then both the sum of squares and the estimate are the same.

I will now argue that this special case should occur all the time i.e., the first component should always capture everything in the full model. The basis of the argument is the idea alluded to above that (estimated) generalized least squares rather than ordinary least squares should be used in step 2. In using generalized least squares we use $X'X$ as an estimate of Σ. (The constant multiplier needed to make this an appropriate estimate cancels out in the calculations.) The estimate of $\hat{t}'$ becomes

$$\begin{aligned}\hat{t}' &= (\hat{w}'(X'X)^{-1}\hat{w})^{-1}\hat{w}'(X'X)^{-1}X' \\ &= \frac{Y'Y}{Y'X(X'X)^{-1}X'Y}Y'X(X'X)^{-1}X'\end{aligned}$$

or equivalently

$$\begin{aligned}\hat{t} &= \frac{Y'Y}{Y'X(X'X)^{-1}X'Y}X(X'X)^{-1}X'Y \\ &= M_X Y\frac{Y'Y}{Y'M_X Y}\end{aligned}$$

If we now fit equation (1.1) we get the sum of squares for the model identical to that of the full model, also

$$\hat{v}_1 = \frac{Y'M_X Y}{Y'Y},$$

and the calibration estimate is

$$\hat{\hat{\xi}}_0 = a_0'(X'X)^{-1}X'Y\frac{Y'Y}{Y'M_X Y}\hat{v}_1 = a_0'(X'X)^{-1}X'Y$$

just as in the full model.

11.5 Nonnegative Data Compression

Sometimes when Y contains only nonnegative numbers (as in the black and white photo example) it is desired to write

$$Y_{n\times q} = W_{n\times r}H_{r\times q}, \tag{1}$$

where both W and H also only contain nonnegative numbers.

It is by no means clear that this can be done at all, and if it can be done, at least some times it will not be unique. If you take an r dimensional orthonormal matrix O, it is clear that

$$WH = (WO)(O'H) \equiv \tilde{W}\tilde{H}.$$

While there is no assurance that WO and $O'H$ will contain only nonnegative numbers, because WO rotates the rows of W and $O'H$ rotates the columns of H, if W and H contain only positive numbers, we can always choose a rotation that is small enough to keep all the numbers positive. Hence for positive Y, W, and H we know the decomposition is not unique.

Much like the previous section, our real interest is in finding matrices W and H that make equation (1) approximately correct, rather than finding a true solution.

11.5.1 Iterative Proportional Fitting

This method is often called *the multiplicative update rule* but *iterative proportional fitting* seems as apt a name and the algorithm is quite similar to the iterative proportional fitting sometimes used for fitting log-linear models, cf. Christensen (1997, Section 3.3).

If $Y = WH$, then we have equality of the $r \times q$ matrices

$$W'Y = W'WH. \tag{2}$$

Write the ij element of these matrices as

$$\{W'Y\}_{ij} = \{W'WH\}_{ij}.$$

Remembering that $W'Y$, $W'WH$, and H all have the same dimensions and writing $H = [h_{ij}]$ (and $W = [w_{ij}]$), it is immediately obvious that

$$h_{ij} = h_{ij} \frac{\{W'Y\}_{ij}}{\{W'WH\}_{ij}}$$

and that a similar argument based on the $n \times r$ matrix equality

$$YH' = WHH' \tag{3}$$

gives

$$w_{ij} = w_{ij} \frac{\{YH'\}_{ij}}{\{WHH'\}_{ij}}.$$

This is the basis for the algorithm. Start with any initial guesses for W and H, say $W^{(0)}$ and $H^{(0)}$ that contain nonnegative numbers. (Off the top of my head I would probably pick $W^{(0)} = J_n^r$ and $H^{(0)} = J_r^q$.) Update the initial guesses with the equations

$$h_{ij}^{(n+1)} = h_{ij} \frac{\{W^{(n)\prime}Y\}_{ij}}{\{W^{(n)\prime}W^{(n)}H^{(n)}\}_{ij}} \tag{4}$$

$$w_{ij}^{(n+1)} = w_{ij}^{(n)} \frac{\{YH^{(n+1)\prime}\}_{ij}}{\{W^{(n)}H^{(n+1)}H^{(n+1)\prime}\}_{ij}}. \tag{5}$$

If the current versions $W^{(n)}$ and $H^{(n)}$ have all nonnegative entries, the multipliers are nonnegative, so the updated versions of $W^{(n+1)}$ and $H^{(n+1)}$ also nonnegative. If the algorithm converges, it will converge to values of W and H that satisfy equations (2) and (3), not equation (1). Equation (2) ensures that

$$M_W Y = M_W WH = WH$$

and equation (3) ensures that

$$YM_{H'} = WHM_{H'} = WH,$$

so

$$M_W Y M_{H'} = WH.$$

The question then becomes whether

$$M_W Y M_{H'} \doteq Y = M_W Y M_{H'} + (I_n - M_W) Y M_{H'} + M_W Y (I_q - M_{H'}) + (I_n - M_W) Y (I_q - M_{H'})$$

is a sufficiently useful approximation.

Different starting matrices $W^{(0)}$ and $H^{(0)}$ (even with the same value of r) may converge to different choices of W and H giving different approximations to Y. I also know of no reason the algorithm has to converge.

11.5.2 Nonnegative Iterative Least Squares

The idea of this is simple enough but the execution is much less so. When executing the least squares estimation schemes for partial least squares in Subsection 11.4.2, simply add in a constraint that the least squares estimates must be nonnegative numbers. Easy to say; hard to do. The unrestricted least squares estimates have closed form solutions so the partial least squares algorithm is fairly straight forward. Finding least squares estimates subject to linear inequality constraints is a much more difficult thing to do, cf. ALM-III, Appendix A.3.

Using, as before, the subscript j to indicate the jth column of matrices, compute each $H_j^{(n+1)}$ to be the least squares estimate of H_j from minimizing $[Y_j - W^{(n)}H_j]'[Y_j - W^{(n)}H_j]$, $j = 1, \ldots, q$ but now the parameters and estimates are subject to the linear inequality constraints $h_{ij}^{(n+1)} \geq 0$. Similarly, looking at the ith rows of matrixes, now take $w_i^{(n+1)}$ to be the least squares estimate of w_i from minimizing $[y_i - H^{(n+1)}w_i]'[y_i - H^{(n+1)}w_i]$, $i = 1, \ldots, n$ subject to the linear inequality constraints $w_{ij}^{(n+1)} \geq 0$. Again, the linear inequality constraints make the problem of finding the least squares estimates much harder.

11.6 Factor Analysis

Principal components are often used in an attempt to identify factors underlying the observed data. There is also a formal modeling procedure called *factor analysis* that is used to address this issue. The model looks similar to the multivariate linear models of Appendix C.1 but several of the assumptions are changed and, most importantly, you don't get to see the matrix of predictor variables. It is assumed that the observation vectors y_i are uncorrelated and have $\mathrm{E}(y_i) = \mu$ and $\mathrm{Cov}(y_i) = \Sigma$. For n observations, the factor analysis model is

$$y_i' = \mu' + x_i' B + \varepsilon_i', \quad i = 1, \ldots, n$$

or

$$Y = J\mu' + XB + e, \tag{1}$$

where Y is $n \times q$ and X is $n \times r$. Most of the usual multivariate linear model assumptions about the rows of e are made,

$$\begin{aligned} \mathrm{E}(\varepsilon_i) &= 0, \\ \mathrm{Cov}(\varepsilon_i, \varepsilon_j) &= 0 \qquad i \neq j, \end{aligned}$$

and for Ψ nonnegative definite

$$\mathrm{Cov}(\varepsilon_i) \equiv \Psi,$$

except Ψ is now assumed to be *diagonal*. The matrix B remains a fixed but unknown matrix of parameters. The entries in B are now called *factor loadings*.

The primary change in assumptions relates to X which is now random and *unobservable*. We have assumed that the mean of the observation vector on individual i is $\mathrm{E}(y_i') = \mu'$, so the corresponding row $x_i' B$ better be random with mean zero. Each row of X is assumed to be an unobservable random vector with

$$\begin{aligned} \mathrm{E}(x_i) &= 0, \\ \mathrm{Cov}(x_i, x_j) &= 0, \quad i \neq j, \\ \mathrm{Cov}(x_i) &= I_r, \end{aligned}$$

and

$$\mathrm{Cov}(x_i, \varepsilon_j) = 0, \quad \text{any } i, j.$$

The idea behind the model is that the elements of X consist of r underlying *common factors*. For fixed individual i, the different observations, y_{ih}, $h = 1, \ldots, q$ are different linear combinations of the r (random) common factors for that individual x_{ik}, $k = 1, \ldots, r$ plus some (white noise) error ε_{ih}, $h = 1, \ldots, q$. The errors for the individual are allowed to have different variances but they are assumed to be uncorrelated.

Specifically, every individual i has observations y_{ih} that involve the same linear combinations (loadings) of the common factors, $\sum_{k=1}^{r} \beta_{hk} x_{ik}$, where the β_{hk}s do not depend on the individual i but change depending on h, so the linear combination depends on which variable y_{ih} is being considered for individual i. Of course different individuals i have different realizations of the random common factors x_{ik}. So, while all the common factors affect each column of Y, different linear combinations of the common factors apply to different columns. The kth column of X consists of n realizations of the kth common factor. The row k of B therefore tells how the kth factor is incorporated into all of observations. It serves as the basis for trying to interpret what the kth factor contributes. The hth column of B are the coefficients that generate the hth dependent variable y_{ih}.

For the model to be of interest, the number of factors r should be less than the number of variables q. Based on this model, $y_i = \mu + B'x_i + \varepsilon_i,\ i = 1,\ldots,n$, so

$$\begin{aligned}\mathrm{Cov}(y_i) &= \mathrm{Cov}(B'x_i + \varepsilon_i)\\ &= B'B + \Psi\,.\end{aligned}$$

In most of the discussion to follow, we will work directly with the matrix $B'B$. It is convenient to have a notation for this matrix. Write

$$\Lambda \equiv B'B\,.$$

The matrix Λ is characterized by two properties: (1) Λ is nonnegative definite and (2) $r(\Lambda) = r$. Recalling our initial assumption that $\mathrm{Cov}(y_i) = \Sigma$, the factor analysis model has imposed the restriction that

$$\Sigma = \Lambda + \Psi\,. \tag{2}$$

Clearly, one cannot have $r(\Lambda) = r > q$. It is equally clear that if $r = q$, one can always find matrices Λ and Ψ that satisfy (2). Just choose $\Lambda = \Sigma$ and $\Psi = 0$. If $r < q$, equation (2) may place a real restriction on Σ, see also a comment below about an exercise in *ALM-III*.

In practice, Σ is unknown and estimated by S, so one seeks matrices $\hat{B}$ and $\hat{\Psi}$ such that

$$S \doteq \hat{B}'\hat{B} + \hat{\Psi}\,.$$

The interesting questions now become (a) how many factors r are needed to get a good approximation and (b) which matrices $\hat{B}$ and $\hat{\Psi}$ give good approximations. The first question is certainly amenable to analysis. Clearly, $r = q$ will always work, so there must be ways to decide when $r < q$ is doing an adequate job. The second question ends up being tricky. The problem is that if U is any orthonormal matrix, $U\hat{B}$ works just as well as $\hat{B}$ because

$$\hat{B}'\hat{B} = (U\hat{B})'(U\hat{B})\,.$$

11.6.1 Additional Terminology and Applications

As may already be obvious, factor analysis uses quite a bit of unusual terminology.

The elements of a row of e are called *unique* or *specific factors*. These are uncorrelated random variables that are added to the linear combinations of common factors to generate the observations. They are distinct random variables for each distinct observation (i.e., they are specific to the observation). The ith diagonal element of $\Psi \equiv \mathrm{Cov}(\varepsilon_i)$ is called the *uniqueness*, *specificity*, or *specific variance* of the ith variable.

The diagonal elements of Λ are called *communalities*. Writing $\Lambda = [\lambda_{ij}]$, the communality of the ith variable is generally denoted

$$h_i^2 \equiv \lambda_{ii}\,.$$

Note that if $B = [\beta_{ij}]$,

$$h_i^2 = \sum_{k=1}^{r} \beta_{ki}^2\,.$$

The total variance is

$$\mathrm{tr}[\Sigma] = \mathrm{tr}[\Lambda] + \mathrm{tr}[\Psi]\,.$$

The *total communality* is

$$v \equiv \mathrm{tr}[\Lambda] = \sum_{i=1}^{q} h_i^2 = \sum_{i=1}^{q}\sum_{k=1}^{r} \beta_{ki}^2\,.$$

The matrix

$$\Lambda = \Sigma - \Psi = \mathrm{Cov}(B'x)$$

is called the *reduced covariance matrix*, for obvious reasons. Often the observations are standardized so that Σ is actually a correlation matrix. If this has been done, Λ is sometimes called the reduced correlation matrix (even though it need not be a correlation matrix).

In practice, factor analysis is used primarily to obtain estimates of B. One then tries to interpret the estimated factor loadings in some way that makes sense relative to the subject matter of the data. As is discussed later, this is a fairly controversial procedure. One of the reasons for the controversy is that B is not uniquely defined. Given any orthonormal $r \times r$ matrix U, write $X_0 = XU'$ and $B_0 = UB$; then,

$$XB = XU'UB = X_0B_0,$$

where X_0 again satisfies the assumptions made about X. Unlike standard linear models, X is not observed, so there is no way to tell X and X_0 apart. There is also no way to tell B and B_0 apart. Actually, this indeterminacy is used in factor analysis to increase the interpretability of B. This will be discussed again later. At the moment, we examine ways in which the matrix B is interpreted.

One of the key points in interpreting B is recognizing that it is the rows of B that are important and not the columns. A column of B is used to explain one dependent variable. A row of B consists of all of the coefficients that affect a single common factor. The q elements in the jth row of B represent the contributions made by the jth common factor to the q dependent variables. Traditionally, if a factor has all of its large loadings with the same sign, the subject matter specialist tries to identify some common attribute of the dependent variables that correspond to the high loadings. This common attribute is then considered to be the underlying factor. A *bipolar* factor involves high loadings that are both positive and negative; the user identifies common attributes for both the group of dependent variables with positive signs and

the group with negative signs. The underlying factor is taken to be one that causes individuals who are high on some scores to be low on other scores. The following example involves estimated factor loadings. Estimation is discussed in the following two subsections.

EXAMPLE 11.6.1. Again consider the correlation matrix

$$R = \begin{bmatrix} 1.000 & 0.439 & 0.410 & 0.288 & 0.329 & 0.248 \\ 0.439 & 1.000 & 0.351 & 0.354 & 0.320 & 0.329 \\ 0.410 & 0.351 & 1.000 & 0.164 & 0.190 & 0.181 \\ 0.288 & 0.354 & 0.164 & 1.000 & 0.595 & 0.470 \\ 0.329 & 0.320 & 0.190 & 0.595 & 1.000 & 0.464 \\ 0.248 & 0.329 & 0.181 & 0.470 & 0.464 & 1.000 \end{bmatrix}$$

from Lawley and Maxwell (1971) and Johnson and Wichern (2007) that was obtained from (Gaelic, English, history, arithmetic, algebra, geometry) examination scores on 220 male students.

For $r = 2$, maximum likelihood estimation gives one choice of estimates,

$$\hat{B} = \begin{bmatrix} 0.553 & 0.568 & 0.392 & 0.740 & 0.724 & 0.595 \\ 0.429 & 0.288 & 0.450 & -0.273 & -0.211 & -0.132 \end{bmatrix}$$

and

$$(\hat{\psi}_1, \ldots, \hat{\psi}_6) = (0.510, 0.594, 0.644, 0.377, 0.431, 0.628).$$

Factor interpretation involves looking at the rows of $\hat{B}$ and trying to interpret them. Write

$$\hat{B} = \begin{bmatrix} \hat{b}_1' \\ \hat{b}_2' \end{bmatrix}.$$

All of the elements of $\hat{b}_1$ are large and fairly substantial. This suggests that the first factor is a factor that indicates general intelligence. The second factor is bipolar, with positive scores on math subjects and negative scores on nonmath subjects. The second factor might be classified as some sort of math–nonmath factor. This example will be examined again later with a slightly different slant. □

Rather than taking the factor analysis model as a serious model for the behavior of data, it may be more appropriate to view factor analysis as a data analytic procedure that seeks to discover structure in the covariance matrix and may *suggest* the presence of underlying factors. My son Fletcher (not to be confused with my imaginary son Basil) has convinced me that if you have a previous idea of the important factors it may be a worthwhile exercise to see whether the data are capable of being contorted into consistency with those previous factors.

11.6.2 Maximum Likelihood Theory

Maximum likelihood theory can be used for both estimation and testing. Maximum likelihood factor analysis is based on assuming that the random vectors in the factor model have a joint multivariate normal distribution and rewriting the factor analysis model as a standard multivariate linear model. (By contrast, ICA allows at most one of the factors to have a normal distribution.) To do this, the random terms are pooled together as, say

$$\xi_i = B' x_i + \varepsilon_i$$

and

$$\xi = XB + e\,.$$

With $\Lambda = B'B$, the factor analysis model is a special case of the one-sample model of Section 10.1,

$$Y = J\mu' + \xi, \tag{3}$$

where

$$\begin{aligned} E(\xi_i) &= 0\,, \\ \mathrm{Cov}(\xi_i, \xi_j) &= 0 \qquad i \neq j\,, \end{aligned}$$

and

$$\mathrm{Cov}(\xi_i) = \Lambda + \Psi,$$

with Ψ diagonal, Λ nonnegative definite, and $r(\Lambda) = r$.

In Section 10.1, the assumption was simply that

$$\mathrm{Cov}(\xi_i) = \Sigma\,.$$

The new model places the restriction on Σ that

$$\Sigma = \Lambda + \Psi, \tag{4}$$

where $r(\Lambda) = r$ and Ψ is diagonal. For ξ_is with a joint multivariate normal distribution, the likelihood function for an arbitrary Σ was discussed in Chapter 9. Clearly, the likelihood can be maximized subject to the restrictions that $\Sigma = \Lambda + \Psi$, Λ is nonnegative definite, $r(\Lambda) = r$, and Ψ is diagonal. However Seber (1984, Exercise 5.4) argues that even these parameters are not identifiable without additional restrictions.

Because $\Lambda + \Psi$ is just a particular choice of Σ, as in Chapter 9 the maximum likelihood estimate of μ is always the least squares estimate, $\hat{\mu} = \bar{y}_\cdot$. This simplifies the maximization problem. Unfortunately, with the additional restrictions on Σ, closed-form estimates of the covariance matrix are no longer available. Computational methods for finding MLEs are discussed in Lawley and Maxwell (1971) and Jöreskog (1975). They can be quite difficult.

It is an exercise in *ALM-III* to show
(a) that the maximization problem reduces to finding a rank r matrix $\hat{\Lambda}$ and a diagonal matrix $\hat{\Psi}$ that minimize

$$\log(|\Lambda+\Psi|)+\mathrm{tr}\{(\Lambda+\Psi)^{-1}\hat{\Sigma}_q\},$$

where $\hat{\Sigma}_q=\dfrac{n-1}{n}S$ and $\hat{\Lambda}$ is nonnegative definite and
(b) that any positive definite Σ can be written with a factor analysis analysis structure having $r=q-1$. To do this it is well to remember that $\Sigma=PD(\phi_i)P'=P\left[D(\phi_i)-\phi_q I+\phi_q I\right]P'$

One advantage of the maximum likelihood method is that standard asymptotic results apply. Maximum likelihood estimates are asymptotically normal. Minus two times the likelihood ratio test statistic is asymptotically chi-squared under the null hypothesis. See Geweke and Singleton (1980) for a discussion of sample size requirements for the asymptotic test.

Of specific interest are tests for examining the rank of Λ. If $r<s$, the restriction $r(\Lambda)=r$ is more stringent than the restriction $r(\Lambda)=s$. To test $H_0: r(\Lambda)=r$ versus $H_A: r(\Lambda)=s$, one can use the likelihood ratio test statistic. This is just the maximum value of the likelihood under $r(\Lambda)=r$ divided by the maximum value of the likelihood under $r(\Lambda)=s$. Under H_0, -2 times the log of this ratio has an asymptotic chi-squared distribution. The degrees of freedom are the difference in the number of independent parameters for the models with $r(\Lambda)=s$ and $r(\Lambda)=r$. If we denote

$$\hat{\Sigma}_r=\hat{\Lambda}+\hat{\Psi}$$

when $r(\Lambda)=r$ with a similar notation for $r(\Lambda)=s$, -2 times the log of the likelihood ratio test statistic is easily shown to be

$$n\left[\ln\left(\frac{|\hat{\Sigma}_r|}{|\hat{\Sigma}_s|}\right)+\mathrm{tr}\{(\hat{\Sigma}_r^{-1}-\hat{\Sigma}_s^{-1})\hat{\Sigma}_q\}\right],$$

where again

$$\hat{\Sigma}_q=\frac{n-1}{n}S.$$

As will be seen later, the degrees of freedom for the test are usually

$$\begin{array}{ll}\text{if } s=q,q-1 & df=q(q+1)/2-q-\left[qr-r(r-1)/2\right],\\ \text{if } s<q-1 & df=\left[qs-s(s-1)/2\right]-\left[qr-r(r-1)/2\right].\end{array}$$

The formula for degrees of freedom is derived from the number of independent parameters in each model. If $r(\Lambda)\equiv r=q,q-1$, the covariance matrix Σ is unrestricted. The independent parameters are the q elements of μ and the $q(q+1)/2$ distinct elements of Σ. Recall that because Σ is symmetric, not all of its elements are distinct. Thus, for $r=q,q-1$, the model has

$$q+q(q+1)/2$$

degrees of freedom.

Counting the degrees of freedom when $r(\Lambda) = r < q-1$ is a bit more complicated. The model involves the restriction

$$\Sigma = \Lambda + \Psi,$$

where Ψ is diagonal and Λ is nonnegative definite with rank r. Clearly, Ψ has q independent parameters, the diagonal elements. Because Λ is of rank r, then $q-r$ columns out of the $q \times q$ matrix are linear combinations of the other r columns. Thus, the independent parameters are at most the elements of these r columns. There are qr of these parameters. However, Λ is also symmetric. All of the parameters above the diagonal are redundant. In the first r columns there are $1+2+\cdots+(r-1) = r(r-1)/2$ of these redundant values. Thus, Λ has at most $qr - r(r-1)/2$ parameters. Finally, μ again involves q independent parameters. Adding the number of independent parameters in μ, Ψ, and Λ gives the maximum model degrees of freedom as

$$q + q + [qr - r(r-1)/2].$$

Taking differences in model degrees of freedom gives the test degrees of freedom indicated earlier. However, if r and q are both large, the number of factor model parameters can exceed the number of parameters for the unrestricted covariance matrix model, so an unrestricted covariance matrix should be used. In particular, if $r = q-1$, the number of factor model parameters always exceeds the number of parameters in the unrestricted covariance matrix. Fortunately, r is usually taken to be small.

Thus far in the discussion, we have ignored B in favor of $\Lambda = B'B$. Given a function of Λ, say $B = f(\Lambda)$, and the maximum likelihood estimate $\hat{\Lambda}$, the MLE of B is $\hat{B} = f(\hat{\Lambda})$. The problem is in defining the function f. There are an uncountably infinite number of ways to define f. If f defines B and U is an orthonormal matrix, then

$$f_1(\Lambda) = U f(\Lambda)$$

is just as good a definition of B because $\Lambda = B'B = B'U'UB$. As mentioned, this indeterminacy is used to make the results more interpretable. The matrix B is redefined until the user gets a pleasing $\hat{B}$. The procedure starts with any $\hat{B}$ and then $\hat{B}$ is rotated (multiplied by an orthonormal matrix) until $\hat{B}$ seems to be interpretable to the user. In fact, there are some standard rotations (e.g., varimax and quartimax), that are often used to increase interpretability. For a more complete discussion of rotations see Williams (1979).

Often, in an effort to make B well-defined, it is taken to be $D\left(\sqrt{\phi_1}, \ldots, \sqrt{\phi_r}\right) A_r'$, where $A_r = [a_1, \ldots, a_r]$ with a_i an eigenvector of Λ with length one corresponding to a positive eigenvalue ϕ_i. To accomplish the goal one would need to address the issues of eigenvalues not being unique and the nonidentifiability of Λ.

EXAMPLE 11.6.2. For $r = 2$, the orthonormal matrices U used in rotations are 2×2 matrices. Thus, the effects of orthogonal rotations can be plotted. The plots consist of q points, one for each dependent variable. Each point consists of the two values in

each column of $\hat{B}$. Figure 11.5 gives a plot of the unrotated factor loadings presented in Example 11.6.1 for the examination score data. The points labeled 1 through 6 indicate the corresponding dependent variable $h = 1,\ldots,6$. Two commonly used rotations are the varimax rotation and the quartimax rotation (see Exercise 11.6.10). The *varimax* rotation for these data is

$$\hat{B}_V = \begin{bmatrix} 0.235 & 0.323 & 0.088 & 0.771 & 0.724 & 0.572 \\ 0.659 & 0.549 & 0.590 & 0.170 & 0.213 & 0.210 \end{bmatrix},$$

and the *quartimax* rotation is

$$\hat{B}_Q = \begin{bmatrix} 0.260 & 0.344 & 0.111 & 0.777 & 0.731 & 0.580 \\ 0.650 & 0.536 & 0.587 & 0.139 & 0.184 & 0.188 \end{bmatrix}.$$

A plot of the varimax factor loadings is presented in Figure 11.6. It is a substantial counterclockwise rotation about the origin (0,0) of Figure 11.5. A plot (not given) of the quartimax loadings is a very slight clockwise rotation of the varimax loadings. Rather than isolating a general intelligence factor and a bipolar factor as seen in the unrotated factors, these both identify factors that can be interpreted as one for mathematics ability and one for nonmathematics ability. I have had different software give me slightly different versions of $\hat{B}_V$. □

The factor analysis model for maximum likelihood assumes that the matrix of common factors X has rows consisting of independent observations from a multivariate normal distribution with mean zero and covariance matrix I_r. While X is not observable, it is possible to predict the rows of X. In It can be seen that $\hat{E}(x_i|Y) = B(\Lambda + \Psi)^{-1}(y_i - \mu)$. Thus, estimated best linear predictors of the x_is can be obtained. These *factor scores* are frequently used to check the assumption of multivariate normality. Bivariate plots can be examined for elliptical shapes and outliers. Univariate plots can be checked for normality.

11.6.3 Principal Factor Estimation

It would be nice to have a method for estimating the parameters of model (3) that did not depend on the assumption of normality. Thurstone (1931) and Thompson (1934) have proposed *principal (axes) factor estimation* as such a method. The parameters to be estimated are μ, Λ, and Ψ. As mentioned earlier, model (3) is just a standard multivariate linear model with a peculiar choice for Σ. The results of Section 9.2 imply that $\bar{y}_\cdot$ is the best linear unbiased estimate of μ.

It remains to estimate Λ and Ψ. If Ψ is known, estimation of Λ is easy. Using equation (4)

$$\Lambda = \Sigma - \Psi,$$

where Λ is assumed to be nonnegative definite of rank r. If it were not for the rank condition, a natural estimate would be

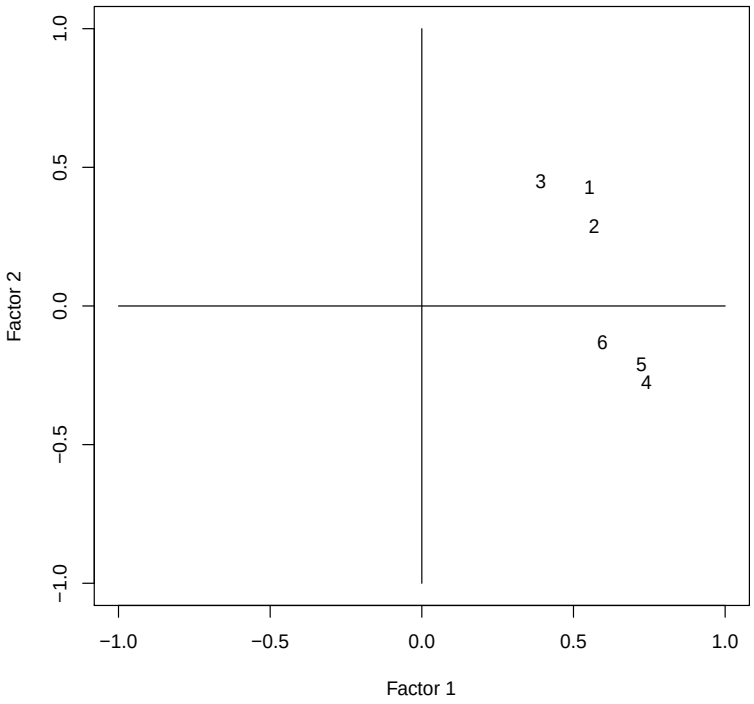

Fig. 11.5 Unrotated factor loadings.

$$\tilde{\Lambda} \equiv S - \Psi .$$

Incorporating the rank condition, one natural way to proceed is to choose a nonnegative definite matrix of rank r, say $\hat{\Lambda}$, that minimizes, say,

$$\mathrm{tr}\{(S-\Psi)-\Lambda\} .$$

Although other functions of $(S-\Psi)-\Lambda$ might be reasonable, the trace is a convenient choice because we have already solved a version of this problem.

Let $\phi_1 \geq \cdots \geq \phi_q$ be the eigenvalues of S, let $a_1, \ldots, a_q$ be the corresponding eigenvectors, let $A_r = [a_1, \ldots, a_r]$, and let G be a $q \times r$ matrix of rank r. In our discussion of principal components, we established that

$$\mathrm{tr}[S - SA_r(A_r'SA_r)^{-1}A_r'S] = \min_G \mathrm{tr}[S - SG(G'SG)^{-1}G'S] .$$

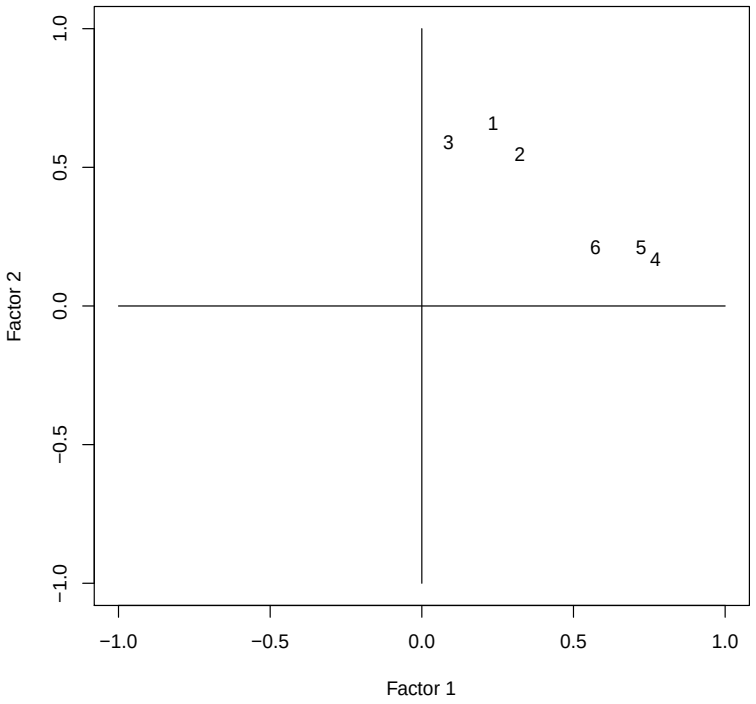

Fig. 11.6 Varimax factor loadings.

Clearly, $SG(G'SG)^{-1}G'S$ is nonnegative definite of rank r. If we consider the problem of estimating Σ when $r(\Sigma) = r$ and restrict ourselves to the class of estimates $SG(G'SG)^{-1}G'S$, then the matrix $SA_r(A_r'SA_r)^{-1}A_r'S$ is an optimal rank r estimate of S.

Applying this result in the factor analysis problem gives an optimal estimate

$$\hat{\Lambda} = \tilde{\Lambda}A_r(A_r'\tilde{\Lambda}A_r)^{-1}A_r'\tilde{\Lambda},$$

where A_r consists of eigenvectors of $\tilde{\Lambda} = S - \Psi$. If we choose the eigenvectors so that $A_r'A_r = I_r$, $\hat{\Lambda}$ simplifies to

$$\hat{\Lambda} = A_r D(\phi_1, \ldots, \phi_r)A_r',$$

where $\phi_1, \ldots, \phi_r$ are the r largest eigenvalues of $\tilde{\Lambda}$. An obvious estimate of B is

$$\hat{B} = D(\sqrt{\phi_1}, \ldots, \sqrt{\phi_r})A_r'.$$

Of course, any rotation of $\hat{B}$ is an equally appropriate estimate.

All of this assumes that Ψ is known. In practice, one makes an initial guess Ψ_0 that leads to initial estimates $\tilde{\Lambda}_0 = S - \Psi_0$ and $\hat{\Lambda}_0$. Having computed $\hat{\Lambda}_0$, compute Ψ_1 from the diagonal elements of $S - \hat{\Lambda}_0$ and repeat the process to obtain $\hat{\Lambda}_1$. This iterative procedure can be repeated until convergence. A common choice for $\Psi_0 = D(\psi_{i0})$ is

$$\psi_{i0} = 1/s^{ii},$$

where s^{ii} is the ith diagonal element of S^{-1}.

Another common choice for Ψ_0 is taking $\psi_{i0} = 0$ for all i. This choice yields $\tilde{\Lambda} = S$, and the rows of $\hat{B} = D(\sqrt{\phi_i})A_r'$ are eigenvectors of S. These are the same vectors as used to determine principal components. In fact, principal components are often used to address questions about underlying factors. The difference is that in a principal component analysis the elements of the eigenvector determine a linear combination of the dependent variable y. In the factor analysis model, the elements of an eigenvector, say a_1, are the q coefficients applied to the first hypothetical factor. Although factor interpretations are based on these q values, in the factor analysis model, data are generated using the r values $a_{1h}, \ldots, a_{rh}$ taken *across* eigenvectors.

Some of the problems with principal factor estimation are that r is assumed to be known, there are no tests available for the value of r, and the matrix $S - \Psi$ may not be nonnegative definite.

In our examination of principal components, we found that the eigenvectors of Σ provided solutions to several different problems: sequential prediction, joint prediction, sequential variance maximization, and geometrical interpretation. The principal factor estimation method can also be motivated by a sequential optimization problem, see Gnanadesikan (1977).

11.6.4 Computing

I performed the analysis in Minitab 18, in R's `factanal`, and in the R library `psych`'s program `fa`. R's `factanal` fits using only maximum likelihood, Minitab allows both methods, and `fa` allows these two and several more. Between these various programs, the third digit of the loadings often differed by 1.

As for rotations, R's `factanal` allows none and varimax as well as promax which is a transformation but not a rotation. Minitab allows none, varimax, quartimax, equimax, and a family of rotations called orthomax. `Psych`'s `fa` allows none, varimax, quartimax, equamax, and four others as well as promax and six more transformations that are not rotations.

11.6.5 Discussion

There is no question that model (3) is a reasonable model. There is considerable controversy about whether the factor analysis model (1) has any meaning beyond that of model (3).

Factor analysis is a frequently used methodology. Obviously, its users like it. Users like to rotate the estimated factor loadings $\hat{B}$ and interpret their results. On the other hand, many people, often of a more theoretical bent, are deeply disturbed by the indeterminacy of the factors and the factor loadings. Many people claim it is impossible to understand the nature of the underlying factors and the basis of their interpretation. Personally, I have always tried to straddle this particular fence. There are people on both sides that I respect. (OK, as I have gotten older, I've fallen on the "disturbed by the indeterminacy" side of the fence.)

An important criterion for evaluating models is that if a model is useful it should be useful for making predictions about future observables. The maximum likelihood model (3), like all linear models, satisfies this criterion. The prediction of a new case would be $\bar{y}_\cdot$. The peculiar covariance matrix of model (3) plays a key role in predicting the unobserved elements of a new case when some of the elements have been observed.

The factor analysis model (1) looks like it is more than the corresponding linear model. The interpretation of factor loadings depends on (1) being more than the linear model. If the factor analysis model really is more than the linear model, it should provide predictions that are distinct from the linear model. When the factor analysis model is correct, these predictions should be better than the linear model predictions.

Unfortunately, the factor analysis model does not seem to lend itself to prediction except through the corresponding linear model. One can predict the factor vectors x_i (assuming that μ, B, and Ψ are known), but this does not affect prediction of y_i. In particular, it is an exercise in *ALM* to show that

$$\hat{E}(x_i|Y) = \hat{E}(x_i|y_i) = B(\Lambda + \Psi)^{-1}(y_i - \mu).$$

Though the factor analysis model may not hold up to careful scrutiny, it does not follow that the data-analytic method known as factor analysis is a worthless endeavor. Rather than thinking of factor analysis as a theoretical method of estimating the loadings on some unspecified factors, it may be better to think of it as a data-analytic method for identifying structure in the covariance matrix. As a data-analytic method, it is neither surprising nor disconcerting that different people (using different rotations) obtain different results. It is more important whether, in practice, users working on similar problems often obtain similar results.

The factor analysis model is one motivation for this method of data analysis. We now will present a slightly different view. We begin by decomposing the covariance matrix into the sum of r different covariance matrices plus Ψ. In other words, write

$$B = \begin{bmatrix} b_1' \\ \vdots \\ b_r' \end{bmatrix}$$

and

$$\Lambda_i = b_i b_i'.$$

Thus,

$$\begin{aligned} \Sigma &= \Lambda + \Psi \\ &= B'B + \Psi \\ &= \sum_{i=1}^{r} b_i b_i' + \Psi \\ &= \sum_{i=1}^{r} \Lambda_i + \Psi. \end{aligned}$$

We can think of y as being a random observation vector and Λ_i as being the covariance matrix for some factor, say w_i, where $y = \mu + \sum_{i=1}^{r} w_i + \varepsilon$ with $\mathrm{Cov}(w_i, w_j) = 0$, $\mathrm{Cov}(\varepsilon) = \Psi$, and $\mathrm{Cov}(w_i, \varepsilon) = 0$. In the usual factor analysis model with factors $x = (x_1, \ldots, x_r)'$ and $B' = [b_1, \ldots, b_r]$, we have $w_i = x_i b_i$. The question then becomes what kind of underlying factor w_i would generate a covariance matrix such as Λ_i. The advantage to this point of view is that attention is directed towards explaining the observable correlations. In traditional factor analysis, attention is directed towards estimating the ill-defined factor loadings. Of course, the end result is the same.

Just as the matrix B is not unique, neither is the decomposition

$$\Lambda = \sum_{i=1}^{r} \Lambda_i.$$

In practice, one would rotate B to make the Λ_is more interpretable. Moreover, as will be seen later, one need not actually compute Λ_i to discover its important structure. The key features of Λ_i are obvious from examination of b_i.

EXAMPLE 11.6.3. Using $\hat{B}$ from Example 11.6.1

$$\hat{\Lambda}_1 = \hat{b}_1 \hat{b}_1' = \begin{bmatrix} 0.31 & 0.31 & 0.22 & 0.41 & 0.40 & 0.33 \\ 0.31 & 0.32 & 0.22 & 0.42 & 0.41 & 0.34 \\ 0.22 & 0.22 & 0.15 & 0.29 & 0.28 & 0.23 \\ 0.41 & 0.42 & 0.29 & 0.55 & 0.54 & 0.44 \\ 0.40 & 0.41 & 0.28 & 0.44 & 0.52 & 0.43 \\ 0.33 & 0.34 & 0.23 & 0.44 & 0.43 & 0.35 \end{bmatrix}.$$

All of the variances and covariances are uniformly high because all of the elements of b_1 are uniformly high. The factor w_1 must be some kind of overall measure — call it general intelligence.

The examination of the second covariance matrix

$$\hat{\Lambda}_2 = \begin{bmatrix} 0.18 & 0.12 & 0.19 & -0.12 & -0.09 & -0.07 \\ 0.12 & 0.08 & 0.13 & -0.08 & -0.06 & -0.04 \\ 0.19 & 0.13 & 0.20 & -0.12 & -0.09 & -0.06 \\ -0.12 & -0.08 & -0.12 & 0.07 & 0.06 & 0.04 \\ -0.09 & -0.06 & -0.09 & 0.06 & 0.04 & 0.03 \\ -0.07 & -0.04 & -0.06 & 0.04 & 0.03 & 0.02 \end{bmatrix}$$

is trickier. The factor w_2 has two parts; there is positive correlation among the first three variables: Gaelic, English, and history. There is positive correlation among the last three variables: arithmetic, algebra, and geometry. However, the first three variables are negatively correlated with the last three variables. Thus, w_2 can be interpreted as a math factor and a nonmath factor that are negatively correlated.

A totally different approach to dealing with Λ_2 is to decide that any variable with a *variance* less than, say, 0.09 is essentially constant. This leads to

$$\tilde{\Lambda}_2 = \begin{bmatrix} 0.18 & 0 & 0.19 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.19 & 0 & 0.20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Thus, the second factor puts weight on only Gaelic and history. The second factor would then be interpreted as some attribute that only Gaelic and history have in common.

Either analysis of Λ_2 can be arrived at by direct examination of

$$b_2' = (-0.429, -0.288, -0.450, 0.273, 0.211, 0.132).$$

The pattern of positives and negatives determines the corresponding pattern in Λ_2. Similarly, the requirement that a variance be greater than 0.09 to be considered nonzero corresponds to a variable having an absolute factor loading greater than 0.3. Only Gaelic and history have factor loadings with absolute values greater than 0.3. Both are negative, so Λ_i will display the positive correlation between them. □

The examination of underlying factors is, of necessity, a very slippery enterprize. The parts of factor analysis that are consistent with traditional ideas of modeling are estimation of Λ and Ψ and the determination of the rank of Λ. The rest is pure data analysis. It is impossible to prove that underlying factors actually exist. The argument in factor analysis is that if these factors existed they could help explain the data.

Factor analysis can only suggest that certain factors might exist. The appropriate question is not whether they really exist but whether their existence is a useful idea. For example, does the idea of a factor for general intelligence help people to understand the nature of test scores. A more stringent test of usefulness is whether the idea

of a general intelligence factor leads to accurate predictions about future observable events. Recall from Exercise 11.5 that one can predict factor scores, so those predictions can be used as a tool in making predictions about future observables for the individuals in the study.

An interesting if unrelated example of these criteria for usefulness involves the force of gravity. For most of us, it is impossible to prove that such a force exists. However, the idea of this force allows one to both explain and predict the behavior of physical objects. The fact that accurate predictions can be made does not prove that gravity exists. If an idea explains current data in an intelligible manner and/or allows accurate prediction, it is a useful idea. For example, the usefulness of Newton's laws of motion cannot be disregarded just because they break down for speeds approaching that of light.

11.7 Independent Component Analysis

The math in this section is at a higher level than most of the book.

Suppose we have several, say q, independent sources (people talking) along with q data collection points (microphones in a room). Our interest is in using the data to isolate/recover the independent sources. This is similar in spirit to factor analysis but does not actually involve reducing the dimensionality of the problem.

Denote the sources as x and the data as y, Eventually, we will collect n observations on y and look at their corresponding source information. The model assumes that the data are a linear transformation of the sources, so $y = Bx$, and with $W \equiv B^{-1}$, $x = Wy$. Write $W' = [w_1, \ldots, w_q]$, the probability density function of y as $f_y(v)$, and the density of x as $f_x(u) \equiv \prod_{j=1}^{q} f_j(u_j)$, where multiplying the individual marginal densities occurs because we assume the sources are independent. Often the densities f_j are taken as identical from either the logistic or double exponential distribution.

Using the standard change of variable formula (e.g., Section A.5 of INFER) with the determinant of W denoted $|W|$,

$$f_y(v) = f_x(Wv)|W| = |W| \prod_{j=1}^{q} f_j(w_j' v).$$

For a random sample y_i, $i = 1, \ldots, n$, the log-likelihood is

$$\ell(W) = \log \left\{ \prod_{i=1}^{n} \left[|W| \prod_{j=1}^{q} f_j(w_j' v) \right] \right\} = n \log |W| + \sum_{i=1}^{n} \sum_{j=1}^{q} \log \left[f_j(w_j' y_i) \right]$$

Find the derivative.

Stochastic gradient ascent is often used to try to find a maximum of the likelihood function using a random starting matrix W_0.

Once you have (an estimate of) W,

$$\begin{bmatrix} y_1' \\ \vdots \\ y_n' \end{bmatrix} W' = YW' = X$$

gives columns of X that are the independent components. By examining the sounds in the columns of X, one might be able to isolate what an individual person was saying in the room from the sound picked up by the array of microphones.

This method is not without problems.

- The transformation method requires having the same number of microphones as speakers. If there are fewer speakers, perhaps some speakers can be considered white noise. If there are more speakers than microphones, some speakers will get lumped together, perhaps even in multiple ways. I have heard that this is a subject of ongoing research.
- The distributional assumptions (e.g., tail behaviors) on f_j are probably very important in determining the end result. Also, the suggested distributions both have mean 0, so the analysis will probably be applied to $y_i - \bar{y}_\cdot$, an adjustment not incorporated into the theory.
- Who knows what this likelihood function looks like? How many local maxima and saddle points are there? (Gradient ascent can ascend to a saddlepoint as well as a local maxima.) It seems pretty clear that the rows of W are interchangeable, which means there cannot be a unique maximum likelihood estimate. On the other hand, if you are a spy trying to find out what a diplomat was saying at a party, if any one of the columns of X gives intelligible information, you probably don't care which it is.
- There are apparently theorems saying you cannot do this if the sources are Gaussian. But as in the previous point it seems quite clear that you can never do this uniquely. With sources that are independent standard normals, i.e., multivariate normal $z \sim N(0, I_q)$, then for *any* orthonormal matrix O, the corresponding rotation of z again has $Oz \sim N(0, I_q)$. Thus, much like factor analysis, we also have $ZO' = YW'$ and $Z = YW'O$. Any rotation of the rows of YW' would be just as valid a decomposition into independent sources. Hence people use nonnormal distributions like the logistic and double exponential for modeling the independent sources. Unlike independent normals, rotating independent logistics or independent double exponentials do not lead to vectors that again have independent components.

11.8 Additional Exercises

Exercise 11.8.1. (a) Find the vector b that minimizes

$$\sum_{i=1}^{q} \left[y_i - \mu_i - b'(x - \mu_x) \right]^2 .$$

(b) For given weights w_i, $i = 1, \ldots, q$, find the vector b that minimizes

$$\sum_{i=1}^{q} w_i^2 \left[y_i - \mu_i - b'(x - \mu_x) \right]^2 .$$

(c) Find the vectors b_i that minimize

$$\sum_{i=1}^{q} w_i^2 \left[y_i - \mu_i - b_i'(x - \mu_x) \right]^2 .$$

Exercise 11.8.2. In a population of large industrial corporations, the covariance matrix for $y_1 = assets/10^6$ and $y_2 = net\ income/10^6$ is

$$\Sigma = \begin{bmatrix} 75 & 5 \\ 5 & 1 \end{bmatrix}.$$

(a) Determine the principal components.
(b) What proportion of the total prediction variance is explained by $a_1'y$?
(c) Interpret $a_1'y$.
(d) Repeat (a), (b), and (c) for principal components based on the correlation matrix.

Exercise 11.8.3. What are the principal components associated with

$$\Sigma = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} ?$$

Discuss the problem of reducing the variables to a two-dimensional space.

Exercise 11.8.4. Let $v_1 = (2,1,1,0)'$, $v_2 = (0,1,-1,0)'$, $v_3 = (0,0,0,2)'$, and

$$\Sigma = \sum_{i=1}^{3} v_i v_i' .$$

(a) Find the principal components of Σ.
(b) What is the predictive variance of each principal component? What percentage of the maximum prediction error is accounted for by the first two principal components?
(c) Interpret the principal components.
(d) What are the correlations between the principal components and the original variables?

Exercise 11.8.5. Do a principal components analysis of the female turtle carapace data of Exercise 10.6.1.

Exercise 11.8.6. The data in Table 11.2 are a subset of the Chapman data reported by Dixon and Massey (1983). It contains the age, systolic blood pressure, diastolic blood pressure, cholesterol, height, and weight for a group of men in the Los Angeles Heart Study. Do a principal components analysis of the data.

Table 11.2 Chapman data.

age	sbp	dbp	chol	ht	wt	age	sbp	dbp	chol	ht	wt
44	124	80	254	70	190	37	110	70	312	71	170
35	110	70	240	73	216	33	132	90	302	69	161
41	114	80	279	68	178	41	112	80	394	69	167
31	100	80	284	68	149	38	114	70	358	69	198
61	190	110	315	68	182	52	100	78	336	70	162
61	130	88	250	70	185	31	114	80	251	71	150
44	130	94	298	68	161	44	110	80	322	68	196
58	110	74	384	67	175	31	108	70	281	67	130
52	120	80	310	66	144	40	110	74	336	68	166
52	120	80	337	67	130	36	110	80	314	73	178
52	130	80	367	69	162	42	136	82	383	69	187
40	120	90	273	68	175	28	124	82	360	67	148
49	130	75	273	66	155	40	120	85	369	71	180
34	120	80	314	74	156	40	150	100	333	70	172
37	115	70	243	65	151	35	100	70	253	68	141
63	140	90	341	74	168	32	120	80	268	68	176
28	138	80	245	70	185	31	110	80	257	71	154
40	115	82	302	69	225	52	130	90	474	69	145
51	148	110	302	69	247	45	110	80	391	69	159
33	120	70	386	66	146	39	106	80	248	67	181

Exercise 11.8.7. Assume a two-factor model with

$$\Sigma = \begin{bmatrix} 0.15 & 0.00 & 0.05 \\ 0.00 & 0.20 & -0.01 \\ 0.05 & -0.01 & 0.05 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0.3 & 0.2 & 0.1 \\ 0.2 & -0.3 & 0.1 \end{bmatrix}.$$

What is Ψ? What are the communalities?

Exercise 11.8.8. Using the vectors v_1 and v_2 from Exercise 11.8.4, let

$$\Lambda = v_1 v_1' + v_2 v_2'.$$

Give the eigenvector solution for B and another set of loadings that generates Λ.

Exercise 11.8.9. Given that

$$\Sigma = \begin{bmatrix} 1.00 & 0.30 & 0.09 \\ 0.30 & 1.00 & 0.30 \\ 0.09 & 0.30 & 1.00 \end{bmatrix}$$

and

$$\Psi = D(0.1, 0.2, 0.3),$$

find Λ and two choices of B.

Exercise 11.8.10. Find definitions for the well-known factor loading matrix rotations varimax, direct quartimin, quartimax, equamax, and orthoblique. What is each rotation specifically designed to accomplish? Apply each rotation to the covariance matrices of Exercise 11.8.9.

Exercise 11.8.11. Do a factor analysis of the female turtle carapace data of Exercise 10.6.1. Include tests for the numbers of factors and examine various factor-loading rotations.

Exercise 11.8.12. Do a factor analysis of the Chapman data discussed in Exercise 11.8.6.

Exercise 11.8.13. Show the following determinant equality.

$$|\Psi + BB'| = |I + B'\Psi^{-1}B||\Psi|.$$

Exercise 11.8.14. Find the likelihood ratio test for

$$H_0 : \Sigma = \sigma^2 \left[(1-\rho)I + \rho JJ'\right]$$

against the general alternative.

Chapter 12
Clustering

Abstract Cluster analysis takes an unstructured data matrix $Y_{n\times q}$ and turns it into a one-way MANOVA data structure that places every observation into a group. We will illustrate two methods for doing this and mention a third. Hierarchical cluster analysis starts by treating every data vector y_h as a separate group and then sequentially looks to combine groups into larger groups. Left to the bitter end, the process combines everything into one big group. The key issues are how to combine groups and how many groups is it appropriate to have. K-means clusterimg starts with K clusters, typically constructed randomly, and looks to improve the homogeneity of the K clusters. Spectral cluster analysis is little more than applying clustering methods to principal components.

12.1 Pointwise Distance Measures

The idea of clustering is to collect cases that are close together. That presupposes some idea of what it means for data vectors to be near one another.

In Section 4.4 we discussed various *norm*s for vectors. The norm of a vector is just its length. The distance between two vectors is the norm of their difference vector. For example, the squared Euclidean distance between a vector $y = (y_1, \ldots, y_q)'$ and another vector w is

$$\|y-w\|^2 \equiv (y-w)'(y-w) = (w-y)'(w-y) \equiv \|w-y\|^2.$$

For $p \geq 1$ the $\mathbf{L}^p$ norm is defined as

$$\|y\|_p \equiv \left(\sum_{h=1}^{q} |y_h|^p \right)^{1/p}.$$

The $\mathbf{L}^2$ norm gives Euclidian distance, so in our notation

$$\|y-w\| \equiv \|y-w\|_2.$$

The second most frequently used $\mathbf{L}^p$ norm in Statistics is probably $\mathbf{L}^1$,

$$\|y\|_1 \equiv \sum_{h=1}^{q} |y_h|.$$

Ridge regression and lasso regression both penalize estimates of the regression coefficient vector β_* based on the length of β_*. Ridge regression uses the squared $\mathbf{L}^2$ norm as a penalty function and lasso uses the $\mathbf{L}^1$ norm. The $\mathbf{L}^\infty$ norm is defined as

$$\|y\|_\infty \equiv \max\{|y_1|, |y_2|, \ldots, |y_q|\}.$$

Finally, for a positive definite matrix W one can define a norm via

$$\|y\|_W^2 \equiv y'Wy.$$

Ideas of distance are fundamentally geometric ideas and, indeed, cluster analysis has little to do with statistical ideas. (Although cluster analysis has been taught as a part of statistics for as long as I can remember.) A key feature in any cluster analysis is deciding on an appropriate measure of distance. Such a decision can, and should, be made based on the characteristics of the specific problem. In the examples given later, the measurements involved are comparable to one another and standard Euclidean distance is used. But if the measurements are not comparable, Euclidean distance becomes problematic. For example, if the length and width of turtle shells are measured in millimeters but the height is measured in kilometers, the use of Euclidean distance amounts to ignoring the height measurements since, relative to the other measurements, the heights are all essentially the same.

If no specific ideas about distance are forthcoming in an application, treating the data as a random sample from some population (even one defined as a mixture of subpopulations), suggests using the squared Mahalanobis distance. This was defined in Subsection 10.1.1 for the squared distance between a random vector and its mean. Letting Σ denote the covariance matrix of the population, use

$$y'\Sigma^{-1}y$$

as the squared norm of the vector y, so that the squared distance between y_i and y_j is

$$D^2(y_i, y_j) = (y_i - y_j)'\Sigma^{-1}(y_i - y_j).$$

This is equivalent to using Euclidean distance on a transformation of y that has an identity covariance matrix. In practice we estimate Σ from our sample of ys.

Rather than transforming y to have an identity covariance matrix, a far less appealing (to me) alternative is merely to rescale the components of y so that each has variance 1 (but retain their correlations) and then using Euclidean distances on those transformed variables. This transformation is equivalent to using $\|y\|_W^2$ with $W^{-1} = \text{Diag}(\sigma_{11}, \ldots, \sigma_{qq})$, or, in practice, the estimated variances.

12.2 Hierarchical Cluster Analysis

12.2.1 Background

Hierarchical clustering is hierarchical in that it starts (stage $s = n$) with every row of Y constituting a different group/cluster. At stage $s = n-1$, it combines the two closest observations into one cluster, to give $n-1$ clusters. At stage $s = n-2$, it combines the two clusters that are closest together to give $n-2$ clusters. It proceeds with each stage combining the two clusters that are closest together. When $s = 1$, everything is together in one cluster.

Hierarchal clustering defines a sequence of one-way MANOVA data structures. For each $s = n, \ldots, 1$ define the data structure

$$y_{sij}, \quad i = 1, \ldots, s, \; j = 1, \ldots, N_{si}. \tag{1}$$

At every stage s, $n = N_{s1} + \cdots + N_{ss}$.

There are two things that need to be specified: (1) the distance between two observation vectors and (2) the distance between two clusters of observations. There are several ways to define the distance between two observations and there are several ways to use observation distances to define cluster distances. The results of the algorithm typically depend on both specifications.

The clusters you get depend on how you measure distances between clusters and there exist various proposals for defining the distance between two clusters of vectors. All such proposals for cluster distances are based on the definition of the distance between two vectors, which was considered in the previous section.

12.2.2 Clusterwise "distance" measures

In this subsection we will use the notation for the Euclidean norm to indicate the distance between two vectors but remember that it can be replaced by any of the other norms that we discussed. (I strongly favor the Mahalanobis norm.)

Now that we can talk about the distance between two observations, $\|y - w\|$, we can discuss alternative definitions of the distance between two clusters of points. For some reason the different methods of measuring cluster distances are referred to as *linkage* methods. We specify linkage methods for a fixed number of clusters s, so we use the data notation in display (1) but suppress the subscript s in the notation. Cluster C_i consists of all the observations y_{ij}, $j = 1, \ldots, N_i$. None of the proposed measures of cluster "distance" satisfy the mathematical definition of a distance measure. (Perhaps why they are called linkages?) The measures are summarized in Table 12.1. At each stage of an hierarchical process, combine the two clusters that are closest to each other.

Table 12.1 Clustering Methods.

Linkage Method	Distance Formula
Simple (Nearest Neighbor)	$D_S(C_i,C_k) \equiv \min_{j,h} \|y_{ij} - y_{kh}\|$
Complete (Farthest Neighbor)	$D_C(C_i,C_k) \equiv \max_{j,h} \|y_{ij} - y_{kh}\|$
Average	$D_A(C_i,C_k) \equiv \frac{1}{N_i}\frac{1}{N_k}\sum_{j=1}^{N_i}\sum_{h=1}^{N_k} \|y_{ij} - y_{kh}\|$
Centroid	$D_{Ct}(C_i,C_k) \equiv \|\bar{y}_{i\cdot} - \bar{y}_{k\cdot}\|$
Ward	$D_{Wa}(C_i,C_k)$, cf. equation (2)

All but the last distance measure in Table 12.1 are pretty self explanatory. Simple linkage measures the minimum distance between two clusters. Complete linkage measures the maximum distance between two clusters. Average looks at the average of all the pointwise distances. Centroid looks at the distance between the centers of the clusters. The last measure, Ward's, is more complex.

Ward's method is to join the pair of clusters that minimizes the increase in sum of cluster variances. At the first step, all cluster variances are 0. Join the two points closest together in *Euclidean* distance. With S_i denoting the sample covariance matrix for the ith cluster, thereafter it seems that Ward's linkage joins the two clusters that generate the smallest value of

$$D_{Wa}(C_i,C_k) = \sum_{r\neq i,k} \mathrm{tr}(S_r)^2 + \frac{N_i-1}{N_i+N_k-1}\mathrm{tr}(S_i)^2 + \frac{N_k-1}{N_i+N_k-1}\mathrm{tr}(S_k)^2 + \frac{N_iN_k}{(N_i+N_k)(N_i+N_k-1)}(\bar{y}_{i\cdot} - \bar{y}_{k\cdot})'(\bar{y}_{i\cdot} - \bar{y}_{k\cdot}). \quad (2)$$

I merely think that (2) is correct.

According to Murtagh and Legendre (2014), there are two different algorithms popularly employed for implementing Ward's method and they differ by whether they require inputs that are *Euclidean* distances (R's `ward.D2`, SAS, JMP, Matlab — all as of December, 2012) or require inputs that are squared *Euclidean* distances (R's `ward.D`, SPSS, Systat, Statistica — same date). Moreover, it seems that most programs were not very good about telling users which input was needed.

12.2.3 An Illustration

It is interesting to apply hierarchical clustering to one-way MANOVA data so that we can look at how well the procedure reproduces the actual groups. I used the Cushing's Syndrome data because the small sample size makes it amenable to comparing methods. Because the two measurements are comparable, I used Euclidean distances for simplicity — rather than (my preferred) Mahalanobis distances. Figure 12.1 contains two tree diagrams (*dendograms*); the top is based on single linkage and the bottom is based on complete linkage. In each case I asked the program to identify 5 groups. Single linkage creates one large cluster and 4 singleton groups that you might call outliers. The singleton clusters are 3 of the 6 adenoma cases and the other is a carcinoma case. Complete linkage only generates 1 of the 5 clusters being a singleton. The other 4 clusters are pretty well mixed up relative to the true disease groups. Only one of the 4 clusters with multiple observations contains observations from a single disease category.

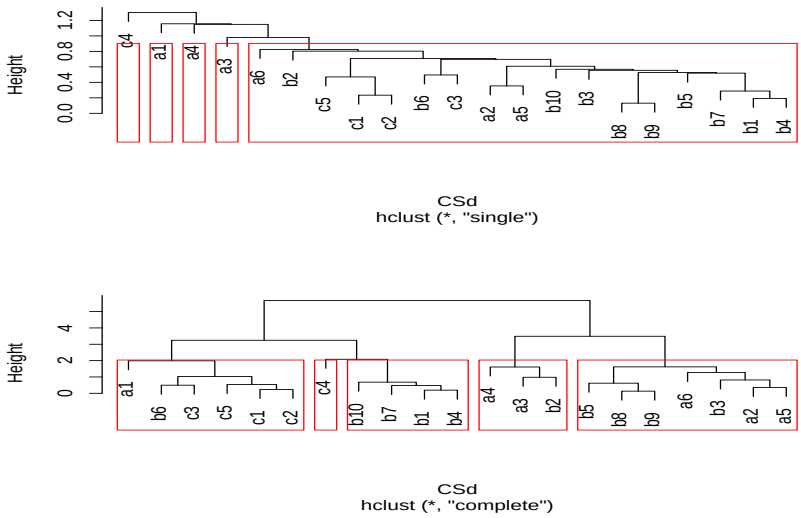

Fig. 12.1 Hierarchical clustering. Single and complete linkage. Cushing Syndrome Data. Five clusters identified.

Figure 12.2 also contains two dendograms, the top based on average linkage and the bottom based on centroid linkage. Again I asked the program to identify 5 groups. Average linkage generates 2 singletons, one adenoma and one carcinoma. The other three groups are pretty well mixed up. Centroid linkage generates 3 singletons but one of the other groups is, with one exception, carcinoma.

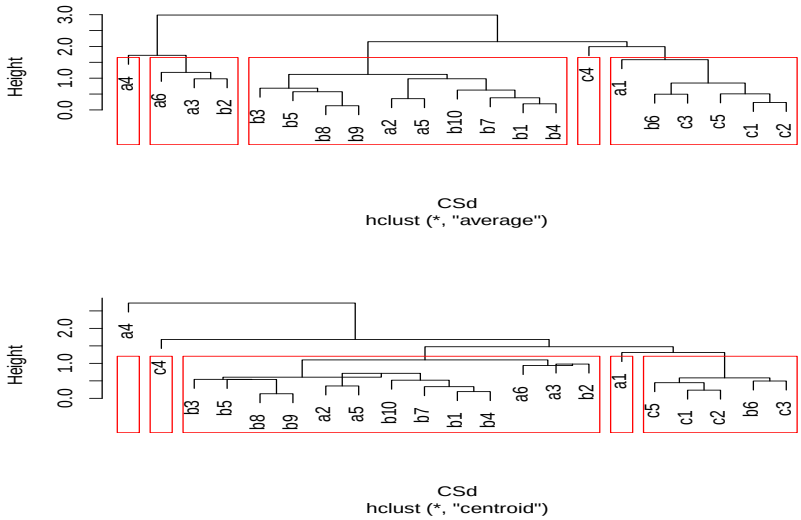

Fig. 12.2 Hierarchical clustering. Average and centroid linkage. Cushing Syndrome Data. Five clusters identified.

Finally, Figure 12.3 contains the dendograms from using the two programs available in R for Ward's method. Inputting distances to one and squared distances to the other, the two programs give the same results; results that do not seem strikingly different from the other methods. Ward's method identifies c4 as a singleton. The rest of the clusters pretty well mix up the disease categories but one of the 4 clusters with multiple observations contains observations from a single disease category.

None of these procedures seem very good at reproducing the 3 disease clusters that we know exist. I leave it as an exercise to reevaluate the clustering using the Mahalanobis norm based on the pooled estimate of the covariance matrix used in the LDA analysis of these data. Doing that is cheating because it presupposes that you know the group structure. A more legitimate alternative is to ignore the true group structure and use the estimated covariance matrix from all observations.

All of these methods identify c4 as a singleton cluster. Three of them identify a4 as a singleton. The cases b3, b5, b8, b9 always seem pretty close. Clustering seems to be based on the hope that points that are close to one another should behave similarly. To me, that seems like a good bet when trying to identify clusters of points to use in a near replicate lack-of-fit test, cf. *ANREG-II*, Section 15.4. In those applications, the only thing you care about is finding points that are very close together. It seems less clear to me what value these methods add to the Cushing's Syndrome data. But then, other than trying to reproduce the known group structure, it is not clear what we would want to accomplish by clustering those observations. That is an issue well worth considering before applying cluster analysis to any data. Often

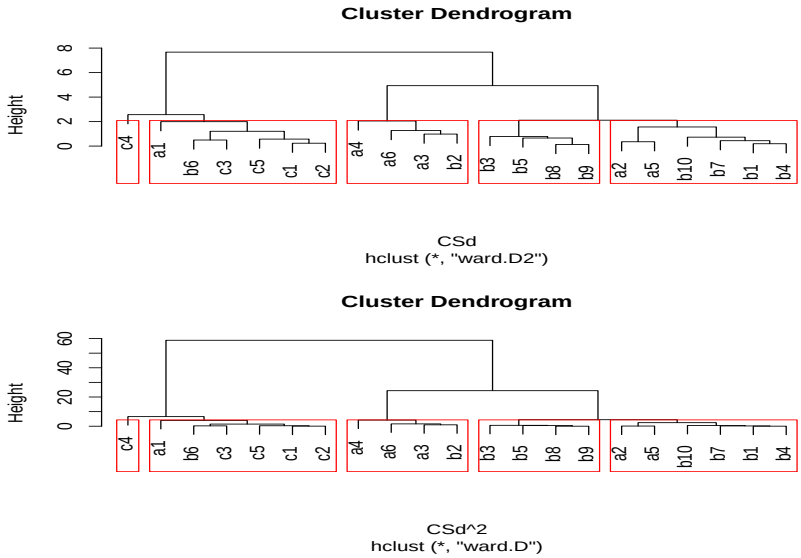

Fig. 12.3 Ward hierarchical clustering: Cushing Syndrome Data. Five clusters identified.

the idea is for cluster analysis to find meaningful groups that you did not know were there.

12.3 K-means Clustering

Again we use $\|y\|$ as generic notation for the length of y, cf. Section 12.1.

The $n \times q$ data matrix Y has rows y_h', $h = 1,\ldots,n$. Similar to hierarchical clustering, at each stage of the process, say s, we will create repeated versions of one-way MANOVA data, but unlike hierarchical clustering, the number of clusters always remains K, i.e.,

$$y_{sij}, \quad i = 1,\ldots,K,\ j = 1,\ldots,N_{si}.$$

At every stage s, $n = N_{s1} + \cdots + N_{sK}$. We will use both the y_h and y_{sij} notations simultaneously.

Pick K points, perhaps randomly, as cluster centers. Call these, $\bar{y}_{0i}$. $i = 1,\ldots,K$. Apparently, the process is very sensitive to how these initial points are selected. Assign y_h to the cluster that minimizes, $\|y_h - \bar{y}_{0i}\|$. This defines the first set of MANOVA data,

$$y_{1ij}, \quad i = 1,\ldots,K,\ j = 1,\ldots,N_{1i}.$$

Compute the cluster means $\bar{y}_{1i\cdot} = (1/N_{1i})\sum_{j=1}^{N_{1i}} y_{1ij}$. Reassign y_h to the cluster that minimizes, $\|y_h - \bar{y}_{1i\cdot}\|$ and use this to define the next set of MANOVA data

$$y_{2ij}, \quad i = 1,\ldots,K,\ j = 1,\ldots,N_{2i}.$$

Repeat this process until the clusters stop changing.

EXAMPLE 12.3.1. *Cushing's Syndrome Data.*
I ran the algorithm with $K = 5$. There does not seem to be much more to do than report the clusters.

$$C_1 = \{a2, a5, b1, b4, b5, b7\},\ C_2 = \{a6, b2, b3, b8, b9\},$$

$$C_3 = \{a3, a4\},\ C_4 = \{a1, c1, c2, c5\}, C_5 = \{b6, b10, c3, c4\}.$$

The comments from the end of Subsection 12.1.4 on the value added by clustering still apply.

12.4 Spectral Clustering

Although there are some twists that can be performed (that actually make the procedure cruder), this is basically about clustering the principal component scores as opposed to the raw data. The name devolves from the relationship of principal component scores to the singular value decomposition of the data matrix as discussed in Section 11.4 because the singular value decomposition is sometimes called the *spectral decomposition*.

move It really does not matter if you look at the raw data principal components YA_r or principal components based on the mean adjusted data $[Y - J\bar{y}_\cdot']A_r = [I - (1/n)J_n^n]YA_r$. You get the same result if you adjust for the mean before transforming the data or after transforming the data because $[(1/n)J_n^nY]A_r = (1/n)J_n^n[YA_r]$. In other words, if you apply the principal component transformation to the raw data, YA_r, and then adjust for the mean values of the transformed data you get $YA_r - (1/n)J_n^n[YA_r]$. You get the same result if you adjusting the raw data for the mean values $Y - (1/n)J_n^nY$, before applying the principal component transformation $[Y - (1/n)J_n^nY]A_r$

12.5 Exercises

EXAMPLE 12.5.1. *Heart Rate Data.*

EXAMPLE 12.5.2. *Other Data.*

Appendix A
Matrices and Derivatives

A matrix is a rectangular array of numbers. Such arrays have *rows* and *columns*. The numbers of rows and columns are referred to as the *dimensions* of a matrix. A matrix with, say, 5 rows and 3 columns is referred to as a 5×3 matrix.

EXAMPLE A.0.1. Three matrices are given below along with their dimensions.

$$\underset{3\times 2}{\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}}, \quad \underset{2\times 2}{\begin{bmatrix} 20 & 80 \\ 90 & 140 \end{bmatrix}}, \quad \underset{4\times 1}{\begin{bmatrix} 6 \\ 180 \\ -3 \\ 0 \end{bmatrix}}.$$

□

Let r be an arbitrary positive integer. A matrix with r rows and r columns, i.e., an $r \times r$ matrix, is called a *square matrix*. The second matrix in Example A.0.1 is square. A matrix with only one column, i.e., an $r \times 1$ matrix, is a *vector*, sometimes called a *column vector*. The third matrix in Example A.0.1 is a vector. A $1 \times r$ matrix is sometimes called a *row vector*.

An arbitrary matrix A is often written

$$A = [a_{ij}]$$

where a_{ij} denotes the element of A in the ith row and jth column. Two matrices are equal if they have the same dimensions and all of their elements (entries) are equal. Thus for $r \times c$ matrices $A = [a_{ij}]$ and $B = [b_{ij}]$, $A = B$ if and only if $a_{ij} = b_{ij}$ for every $i = 1, \ldots, r$ and $j = 1, \ldots, c$.

EXAMPLE A.0.2. Let

$$A = \begin{bmatrix} 20 & 80 \\ 90 & 140 \end{bmatrix} \text{ and } B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}.$$

If $B = A$, then $b_{11} = 20, b_{12} = 80, b_{21} = 90$, and $b_{22} = 140$. □

The *transpose* of a matrix A, denoted A', changes the rows of A into columns of a new matrix A'. If A is an $r \times c$ matrix, the transpose A' is a $c \times r$ matrix. In particular, if we write $A' = [\tilde{a}_{ij}]$, then the element in row i and column j of A' is defined to be $\tilde{a}_{ij} = a_{ji}$.

EXAMPLE A.0.3.

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}' = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

and

$$\begin{bmatrix} 20 & 80 \\ 90 & 140 \end{bmatrix}' = \begin{bmatrix} 20 & 90 \\ 80 & 140 \end{bmatrix}.$$

The transpose of a column vector is a row vector,

$$\begin{bmatrix} 6 \\ 180 \\ -3 \\ 0 \end{bmatrix}' = \begin{bmatrix} 6 & 180 & -3 & 0 \end{bmatrix}. \qquad \square$$

A.1 Matrix Addition

Two matrices can be added (or subtracted) if they have the same dimensions, that is, if they have the same number of rows and columns. Addition and subtraction is performed elementwise.

EXAMPLE A.1.1.

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} + \begin{bmatrix} 2 & 8 \\ 4 & 10 \\ 6 & 12 \end{bmatrix} = \begin{bmatrix} 1+2 & 4+8 \\ 2+4 & 5+10 \\ 3+6 & 6+12 \end{bmatrix} = \begin{bmatrix} 3 & 12 \\ 6 & 15 \\ 9 & 18 \end{bmatrix}.$$

$$\begin{bmatrix} 20 & 80 \\ 90 & 140 \end{bmatrix} - \begin{bmatrix} -15 & -75 \\ 80 & 130 \end{bmatrix} = \begin{bmatrix} 35 & 155 \\ 10 & 10 \end{bmatrix}. \qquad \square$$

In general, if A and B are $r \times c$ matrices with $A = [a_{ij}]$ and $B = [b_{ij}]$, then

$$A + B = [a_{ij} + b_{ij}] \text{ and } A - B = [a_{ij} - b_{ij}].$$

A.2 Scalar Multiplication

Any matrix can be multiplied by a scalar. Multiplication by a scalar (a *real number*) is elementwise.

EXAMPLE A.2.1. Scalar multiplication gives

$$\frac{1}{10}\begin{bmatrix}20 & 80\\ 90 & 140\end{bmatrix} = \begin{bmatrix}20/10 & 80/10\\ 90/10 & 140/10\end{bmatrix} = \begin{bmatrix}2 & 8\\ 9 & 14\end{bmatrix}.$$

$$2\begin{bmatrix}6 & 180 & -3 & 0\end{bmatrix} = \begin{bmatrix}12 & 360 & -6 & 0\end{bmatrix}. \qquad \square$$

In general, if λ is any number and $A = [a_{ij}]$, then

$$\lambda A = [\lambda a_{ij}].$$

A.3 Matrix Multiplication

Two matrices can be multiplied together if the number of columns in the first matrix is the same as the number of rows in the second matrix. In the process of multiplication, the rows of the first matrix are matched up with the columns of the second matrix.

EXAMPLE A.3.1.

$$\begin{bmatrix}1 & 4\\ 2 & 5\\ 3 & 6\end{bmatrix}\begin{bmatrix}20 & 80\\ 90 & 140\end{bmatrix} = \begin{bmatrix}(1)(20)+(4)(90) & (1)(80)+(4)(140)\\ (2)(20)+(5)(90) & (2)(80)+(5)(140)\\ (3)(20)+(6)(90) & (3)(80)+(6)(140)\end{bmatrix}$$
$$= \begin{bmatrix}380 & 640\\ 490 & 860\\ 600 & 1080\end{bmatrix}.$$

The entry in the first row and column of the product matrix, $(1)(20)+(4)(90)$, matches the elements in the first row of the first matrix, $(1\ 4)$, with the elements in the first column of the second matrix, $\begin{pmatrix}20\\ 90\end{pmatrix}$. The 1 in $(1\ 4)$ is matched up with the 20 in $\begin{pmatrix}20\\ 90\end{pmatrix}$ and these numbers are multiplied. Similarly, the 4 in $(1\ 4)$ is matched up with the 90 in $\begin{pmatrix}20\\ 90\end{pmatrix}$ and the numbers are multiplied. Finally, the two products are added to obtain the entry $(1)(20)+(4)(90)$. Similarly, the entry in the third row, second column of the product, $(3)(80)+(6)(140)$, matches the elements in the third row of the first matrix, $(3\ 6)$, with the elements in the second column of the second

matrix, $\begin{pmatrix} 80 \\ 140 \end{pmatrix}$. After multiplying and adding we get the entry $(3)(80)+(6)(140)$. To carry out this matching, the number of columns in the first matrix must equal the number of rows in the second matrix. The matrix product has the same number of rows as the first matrix and the same number of columns as the second because each row of the first matrix can be matched with each column of the second.

□

EXAMPLE A.3.2. We illustrate another matrix multiplication commonly performed in Statistics, multiplying a matrix on its left by the transpose of that matrix, i.e., computing $A'A$.

$$\begin{aligned}\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}' \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} &= \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \\ &= \begin{bmatrix} 1+4+9 & 4+10+18 \\ 4+10+18 & 16+25+36 \end{bmatrix} \\ &= \begin{bmatrix} 14 & 32 \\ 32 & 77 \end{bmatrix}.\end{aligned}$$

□

Notice that in matrix multiplication the roles of the first matrix and the second matrix are *not* interchangeable. In particular, if we reverse the order of the matrices in Example A.3.1, the matrix product

$$\begin{bmatrix} 20 & 80 \\ 90 & 140 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

is undefined because the first matrix has two columns while the second matrix has three rows. Even when the matrix products are defined for both AB and BA, the results of the multiplication typically differ. If A is $r \times s$ and B is $s \times r$, then AB is an $r \times r$ matrix and BA is and $s \times s$ matrix. When $r \neq s$, clearly $AB \neq BA$, but even when $r = s$ we still can not expect AB to equal BA.

EXAMPLE A.3.3. Consider two square matrices, say,

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 2 \\ 1 & 2 \end{bmatrix}.$$

Multiplication gives

$$AB = \begin{bmatrix} 2 & 6 \\ 4 & 14 \end{bmatrix}$$

and

$$BA = \begin{bmatrix} 6 & 8 \\ 7 & 10 \end{bmatrix},$$

so $AB \neq BA$. □

In general if $A = [a_{ij}]$ is an $r \times s$ matrix and $B = [b_{ij}]$ is a $s \times c$ matrix, then

$$AB = [d_{ij}]$$

is the $r \times c$ matrix with

$$d_{ij} = \sum_{\ell=1}^{s} a_{i\ell} b_{\ell j}.$$

A useful result is that the transpose of the product AB is the product, in reverse order, of the transposed matrices, i.e. $(AB)' = B'A'$.

EXAMPLE A.3.4. As seen in Example A.3.1,

$$AB \equiv \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 20 & 80 \\ 90 & 140 \end{bmatrix} = \begin{bmatrix} 380 & 640 \\ 490 & 860 \\ 600 & 1080 \end{bmatrix} \equiv C.$$

The transpose of this matrix is

$$C' = \begin{bmatrix} 380 & 490 & 600 \\ 640 & 860 & 1080 \end{bmatrix} = \begin{bmatrix} 20 & 90 \\ 80 & 140 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = B'A'.$$

□

Let $a = (a_1, \ldots, a_n)'$ be a vector. A very useful property of vectors is that

$$a'a = \sum_{i=1}^{n} a_i^2 \geq 0.$$

A.4 Special Matrices

If $A = A'$, then A is said to be *symmetric*. If $A = [a_{ij}]$ and $A = A'$, then $a_{ij} = a_{ji}$. The entry in row i and column j is the same as the entry in row j and column i. Only square matrices can be symmetric.

EXAMPLE A.4.1. The matrix

$$A = \begin{bmatrix} 4 & 3 & 1 \\ 3 & 2 & 6 \\ 1 & 6 & 5 \end{bmatrix}$$

has $A = A'$. A is symmetric about the diagonal that runs from the upper left to the lower right. □

For any $r \times c$ matrix A, the product $A'A$ is always symmetric. This was illustrated in Example A.3.2. More generally, write $A = [a_{ij}]$, $A' = [\tilde{a}_{ij}]$ with $\tilde{a}_{ij} = a_{ji}$, and

$$A'A = [d_{ij}] = \left[\sum_{\ell=1}^{c} \tilde{a}_{i\ell} a_{\ell j}\right].$$

Note that

$$d_{ij} = \sum_{\ell=1}^{c} \tilde{a}_{i\ell} a_{\ell j} = \sum_{\ell=1}^{c} a_{\ell i} a_{\ell j} = \sum_{\ell=1}^{c} \tilde{a}_{j\ell} a_{\ell i} = d_{ji}$$

so the matrix is symmetric.

Diagonal matrices are square matrices with all off-diagonal elements equal to zero.

EXAMPLE A.4.2. The matrices

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}, \quad \begin{bmatrix} 20 & 0 \\ 0 & -3 \end{bmatrix}, \text{ and } \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

are diagonal. □

In general, a diagonal matrix is a square matrix $A = [a_{ij}]$ with $a_{ij} = 0$ for $i \neq j$. Obviously, diagonally matrices are symmetric. When $v = (v_1, \ldots, v_p)'$ we denote the $p \times p$ diagonal matrix with the v_is on the diagonal as $D(v)$ or $D(v_i)$ (or possible replace D with *Diag*).

An *identity matrix* is a diagonal matrix with all 1s along the diagonal, i.e., $a_{ii} = 1$ for all i. The third matrix in Example A.4.2 above is a 3×3 identity matrix. The identity matrix gets it name because any matrix multiplied by an identity matrix remains unchanged.

EXAMPLE A.4.3.

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}.$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}.$$

□

An $r \times r$ identity matrix is denoted I_r with the subscript deleted if the dimension is clear.

A *zero matrix* is a matrix that consists entirely of zeros. Obviously, the product of any matrix multiplied by a zero matrix is zero.

EXAMPLE A.4.4.

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

□

Often a zero matrix is denoted by 0 where the dimension of the matrix, and the fact that it is a matrix rather than a scalar, must be inferred from the context.

A matrix M that has the property $MM = M$ is called *idempotent*. A symmetric idempotent matrix is a *perpendicular projection operator*.

EXAMPLE A.4.5. The following matrices are both symmetric and idempotent:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}, \quad \begin{bmatrix} .5 & .5 & 0 \\ .5 & .5 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

□

A.5 Linear Dependence and Rank

Consider the matrix

$$A = \begin{bmatrix} 1 & 2 & 5 & 1 \\ 2 & 2 & 10 & 6 \\ 3 & 4 & 15 & 1 \end{bmatrix}.$$

Note that each column of A can be viewed as a vector. The *column space* of A, denoted $C(A)$, is the collection of all vectors that can be written as *a linear combination of the columns of A*. In other words, $C(A)$ is the set of all vectors that can be written as

$$\lambda_1 \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \lambda_2 \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix} + \lambda_3 \begin{bmatrix} 5 \\ 10 \\ 15 \end{bmatrix} + \lambda_4 \begin{bmatrix} 1 \\ 6 \\ 1 \end{bmatrix} = A \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{bmatrix} = A\lambda$$

for some vector $\lambda = (\lambda_1, \lambda_2, \lambda_3, \lambda_4)'$.

The columns of any matrix A are *linearly dependent* if they contain redundant information. Specifically, let x be some vector in $C(A)$. The columns of A are linearly dependent if we can find two distinct vectors λ and γ such that $x = A\lambda$ and $x = A\gamma$. Thus two distinct linear combinations of the columns of A give rise to the same vector x. Note that $\lambda \neq \gamma$ because λ and γ are distinct. Note also that, using a distributive property of matrix multiplication, $A(\lambda - \gamma) = A\lambda - A\gamma = 0$, where $\lambda - \gamma \neq 0$. This condition is frequently used as an alternative definition for linear dependence, i.e., the columns of A are linearly dependent if there exists a vector

$\delta \neq 0$ such that $A\delta = 0$. If the columns of A are not linearly dependent, they are *linearly independent*.

EXAMPLE A.5.1. Observe that the example matrix A given at the beginning of the section has

$$\begin{bmatrix} 1 & 2 & 5 & 1 \\ 2 & 2 & 10 & 6 \\ 3 & 4 & 15 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

so the columns of A are linearly dependent. □

The *rank* of A is the smallest number of columns of A that can generate $C(A)$. It is also the maximum number of linearly independent columns in A.

EXAMPLE A.5.2. The matrix

$$A = \begin{bmatrix} 1 & 2 & 5 & 1 \\ 2 & 2 & 10 & 6 \\ 3 & 4 & 15 & 1 \end{bmatrix}$$

has rank 3 because the columns

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \\ 4 \end{bmatrix}, \begin{bmatrix} 1 \\ 6 \\ 1 \end{bmatrix}$$

generate $C(A)$. We saw in Example A.5.1 that the column $(5,10,15)'$ was redundant. None of the other three columns are redundant; they are linearly independent. In other words, the only way to get

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 6 \\ 3 & 4 & 1 \end{bmatrix} \delta = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

is to take $\delta = (0,0,0)'$. □

A.6 Inverse Matrices

The *inverse* of a square matrix A is the matrix A^{-1} such that

$$AA^{-1} = A^{-1}A = I.$$

The inverse of A exists only if the columns of A are linearly independent. Typically, it is difficult to find inverses without the aid of a computer. For a 2×2 matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

the inverse is given by

$$A^{-1} = \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}. \tag{1}$$

To confirm that this is correct, multiply AA^{-1} to see that it gives the identity matrix. Moderately complicated formulae exist for computing the inverse of 3×3 matrices. Inverses of larger matrices become very difficult to compute by hand. Of course computers are ideally suited for finding such things.

One use for inverse matrices is in solving systems of equations.

EXAMPLE A.6.1. Consider the system of equations

$$\begin{aligned} 2x + 4y &= 20 \\ 3x + 4y &= 10. \end{aligned}$$

We can write this in matrix form as

$$\begin{bmatrix} 2 & 4 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 20 \\ 10 \end{bmatrix}.$$

Multiplying on the left by the inverse of the coefficient matrix gives

$$\begin{bmatrix} 2 & 4 \\ 3 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 2 & 4 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 3 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 20 \\ 10 \end{bmatrix}.$$

Using the definition of the inverse on the left-hand side of the equality and the formula in (A.6.1) on the right-hand side gives

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 3/4 & -1/2 \end{bmatrix} \begin{bmatrix} 20 \\ 10 \end{bmatrix}$$

or

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -10 \\ 10 \end{bmatrix}.$$

Thus $(x,y) = (-10,10)$ is the solution for the two equations, i.e., $2(-10)+4(10) = 20$ and $3(-10)+4(10) = 10$. □

More generally, a system of equations, say,

$$\begin{aligned} a_{11}y_1 + a_{12}y_2 + a_{13}y_3 &= c_1 \\ a_{21}y_1 + a_{22}y_2 + a_{23}y_3 &= c_2 \\ a_{31}y_1 + a_{32}y_2 + a_{33}y_3 &= c_3 \end{aligned}$$

in which the a_{ij}s and c_is are known and the y_is are variables, can be written in matrix form as

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

or

$$AY = C.$$

To find Y simply observe that $AY = C$ implies $A^{-1}AY = A^{-1}C$ and $Y = A^{-1}C$. Of course this argument assumes that A^{-1} exists, which is not always the case. Moreover, the procedure obviously extends to larger sets of equations.

On a computer, there are better ways of finding solutions to systems of equations than finding the inverse of a matrix. In fact, inverses are often found by solving systems of equations. For example, in a 3×3 case the first column of A^{-1} can be found as the solution to

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

For a special type of square matrix, called an *orthonormal matrix*, the transpose is also the inverse. In other words, a square matrix P is an orthonormal matrix if

$$P'P = I = PP'.$$

To establish that P is orthonormal, it is enough to show either that $P'P = I$ or that $PP' = I$. Orthonormal matrices are particularly useful in discussions of eigenvalues and principal components. (Most people call these *orthogonal matrices* but orthonormal is clearly a better name for them.)

Orthogonal is a synonym for perpendicular. Two vectors x and y are orthogonal (written $x \perp y$) if $x'y = 0$. (You can check out the geometry of this in two and three dimensions by realizing that with a coordinate system a vector is the line segment that goes from the point with all zeros to the point specified by the coordinates in the vector.) If the columns of a matrix A are orthogonal it is easy to see that $A'A$ has to be a diagonal matrix. Requiring that $P'P = I$ is requiring more than that the columns of P be orthogonal. It also requires that the columns of P be normalized to have length one. The length of a vector x is defined to be $\sqrt{x'x}$. (Again, you can check out the geometry of this in three or fewer dimensions.)

A.7 Useful Properties

The following proposition summarizes many of the key properties of matrices and the operations performed on them.

Proposition A.7.1. Let A, B, and C be matrices of appropriate dimensions and let λ be a scalar.

$$\begin{aligned}
A+B &= B+A \\
(A+B)+C &= A+(B+C) \\
(AB)C &= A(BC) \\
C(A+B) &= CA+CB \\
\lambda(A+B) &= \lambda A+\lambda B \\
\left(A'\right)' &= A \\
(A+B)' &= A'+B' \\
(AB)' &= B'A' \\
\left(A^{-1}\right)^{-1} &= A \\
\left(A'\right)^{-1} &= \left(A^{-1}\right)' \\
(AB)^{-1} &= B^{-1}A^{-1}.
\end{aligned}$$

The last equality only holds when A and B both have inverses. The second-to-last property implies that the inverse of a symmetric matrix is symmetric because then $A^{-1} = (A')^{-1} = (A^{-1})'$. This is a very important property.

A.8 Eigenvalues; Eigenvectors

Let A be a symmetric matrix. A scalar ϕ is an eigenvalue of A and $x \neq 0$ is an eigenvector for A corresponding to ϕ if

$$Ax = \phi x.$$

This specification also works for any square matrix but then you have to deal with the possibility that the eigenvalues and eigenvectors may involve complex numbers.

EXAMPLE A.8.1. Consider the matrix

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 1 & 3 & -1 \\ -1 & -1 & 5 \end{bmatrix}.$$

The value 3 is an eigenvalue and any nonzero multiple of the vector $(1,1,1)'$ is a corresponding eigenvector. For example,

$$\begin{bmatrix} 3 & 1 & -1 \\ 1 & 3 & -1 \\ -1 & -1 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Similarly, if we consider a multiple, say, $4(1,1,1)'$,

$$\begin{bmatrix} 3 & 1 & -1 \\ 1 & 3 & -1 \\ -1 & -1 & 5 \end{bmatrix} \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 12 \\ 12 \\ 12 \end{bmatrix} = 3 \begin{bmatrix} 4 \\ 4 \\ 4 \end{bmatrix}.$$

The value 2 is also an eigenvalue with eigenvectors that are nonzero multiples of $(1,-1,0)'$.

$$\begin{bmatrix} 3 & 1 & -1 \\ 1 & 3 & -1 \\ -1 & -1 & 5 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \\ 0 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}.$$

Finally, 6 is an eigenvalue with eigenvectors that are nonzero multiples of $(1,1,-2)'$. □

Our uses for eigenvalues and eigenvectors are closely tied to the following result. Sometimes eigenvalues are called singular values or characteristic values (similarly for eigenvectors) but almost everyone seems to use the indicated name for the following result.

Proposition A.8.2. *The Singular Value Decomposition.*
Let A be a symmetric matrix, then for a diagonal matrix $D(\phi_i)$ consisting of eigenvalues there exists an orthonormal matrix P whose columns are corresponding eigenvectors such that

$$A = PD(\phi_i)P'.$$

EXAMPLE A.8.3. Consider again the matrix

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 1 & 3 & -1 \\ -1 & -1 & 5 \end{bmatrix}.$$

In writing $A = PD(\phi_i)P'$, the diagonal matrix is

$$D(\phi_i) = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 6 \end{bmatrix}.$$

The orthonormal matrix is

$$P = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{-2}{\sqrt{6}} \end{bmatrix}.$$

We leave it to the reader to verify that $PD(\phi_i)P' = A$ and that $P'P = I$.

Note that the columns of P are multiples of the vectors identified as eigenvectors in Example A.8.1; hence the columns of P are also eigenvectors. The multiples of the eigenvectors were chosen so that $P'P = I$ and $PP' = I$. Moreover, the first column of P is an eigenvector corresponding to 3, which is the first eigenvalue listed in $D(\phi_i)$. Similarly, the second column of P is an eigenvector corresponding to 2 and the third column corresponds to the third listed eigenvalue, 6.

With a 3×3 matrix A having three *distinct* eigenvalues, any matrix P with eigenvectors for columns would have $P'P$ a diagonal matrix, but the multiples of the eigenvectors must be chosen so that the diagonal entries of $P'P$ are all 1. □

EXAMPLE A.8.4. Consider the matrix

$$B = \begin{bmatrix} 5 & -1 & -1 \\ -1 & 5 & -1 \\ -1 & -1 & 5 \end{bmatrix}.$$

This matrix is closely related to the matrix in Example A.8.1. The matrix B has 3 as an eigenvalue with corresponding eigenvectors that are multiples of $(1,1,1)'$, just like the matrix A. Once again 6 is an eigenvalue with corresponding eigenvector $(1,1,-2)'$ and once again $(1,-1,0)'$ is an eigenvector, but now, unlike A, $(1,-1,0)$ also corresponds to the eigenvalue 6. We leave it to the reader to verify these facts. The point is that in this matrix, 6 is an eigenvalue that has two linearly independent eigenvectors. In such cases, any nonzero linear combination of the two eigenvectors is also an eigenvector. For example, it is easy to see that

$$3\begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix} + 2\begin{bmatrix} 1 \\ 1 \\ -2 \end{bmatrix} = \begin{bmatrix} 5 \\ -1 \\ -4 \end{bmatrix}$$

is an eigenvector corresponding to the eigenvalue 6.

To write $B = PD(\phi)P'$ as in Proposition A.8.2, $D(\phi)$ has 3, 6, and 6 down the diagonal and one choice of P is that given in Example A.8.3. However, because one of the eigenvalues occurs more than once in the diagonal matrix, there are many choices for P. □

Generally, if we need eigenvalues or eigenvectors we get a computer to find them for us. Since eigenvectors are not unique, every program for finding them has to decide which ones to report. Often they report eigenvectors that have length 1, but even if the eigenvalues are all distinct that is not enough to uniquely determine them.

Two frequently used functions of a square matrix are the determinant and the trace.

Definition A.8.5.

a) The determinant of a square matrix is the product of the eigenvalues of the matrix.

b) The trace of a square matrix is the sum of the diagonal elements of the matrix.

This is not the usual definition for a determinant but it works fine for our purposes. It turns out that the trace of a square matrix is also the sum of its eigenvalues but for a nonsymmetric matrix that may involve having eigenvalues that are complex conjugates so that their sum is a real number.

An extremely useful property of the trace, and one that is not at all difficult to show, is that

Proposition A.8.6. For matrices $A_{r\times n}$ and $B_{n\times r}$,

$$\operatorname{tr}(AB) = \operatorname{tr}(BA).$$

We close with a version of the singular value decomposition that applies to matrices that are not square.

Theorem A.8.7. *The Singular Value Decomposition.*
Let X be an $n \times p$ matrix with rank s. Then X can be written as

$$X = ULV',$$

where U is $n \times s$, L is $s \times s$, V' is $s \times p$, and

$$L \equiv \operatorname{Diag}(\lambda_j).$$

The λ_js are the positive square roots of the positive eigenvalues (singular values) of $X'X$ and XX' (i.e., $\lambda_j^2 = \delta_j$). The columns of V are s orthonormal eigenvectors of $X'X$ corresponding to the positive eigenvalues with

$$X'XV = VL^2,$$

and the columns of U are s orthonormal eigenvectors of XX' with

$$XX'U = UL^2.$$

This is proven in *PA*-*V*. In this result, because eigenvectors are not uniquely defined, if you find V with $X'XV = VL^2$, you need to take U as $U = XVL^{-1}$. Similarly, if you find U with $XX'U = UL^2$, you need to take V as $V = X'UL^{-1}$. Typically, one would find the eigenvectors from the smaller of the matrices $X'X$ and XX'.

A.9 Differentiation

Typically, to find maxima and minima one uses differential calculus. This involves finding the derivative of a function. For a function f taking real numbers into real numbers,

$$\mathbf{d}_x f(x) \equiv \lim_{\Delta \to 0} \frac{f(x+\Delta) - f(x)}{\Delta}.$$

Older software for doing minimizations often asked one for both the function and the derivative. Now, software is often either smart enough to figure out the derivative or it approximates the derivative by simply choosing a small Δ.

If f is a (well behaved) function from $\mathbf{R}^p$ into $\mathbf{R}^t$ with $f(x) = [f_1(x), \ldots, f_t(x)]'$, then the derivative of f at c is the $t \times p$ matrix of partial derivatives,

$$\mathbf{d}_x f(x)|_{x=c} \equiv [\partial f_i(x)/\partial x_j|_{x=c}].$$

When the context is clear, we often use simpler notations such as

$$\mathbf{d}_x f(x)|_{x=c} \equiv \mathbf{d}_x f(c) \equiv \mathbf{d} f(c).$$

Critical points are points c where $\mathbf{d} f(c) = 0$.

The chain rule can be written as a matrix product. If $f : \mathbf{R}^s \to \mathbf{R}^t$ and $g : \mathbf{R}^t \to \mathbf{R}^n$, then the composite function is defined by

$$(g \circ f)(x) \equiv g[f(x)]$$

and its derivative is an $n \times s$ matrix that satisfies

$$\mathbf{d}(g \circ f)(c) = [\mathbf{d}_v g(v)|_{v=f(c)}][\mathbf{d}_x f(x)|_{x=c}] \equiv \mathbf{d} g[f(c)]\mathbf{d} f(c).$$

Proposition F.1. Let A be a fixed $t \times s$ matrix with $t = s$ in part (b).
(a) $\mathbf{d}_x[Ax] = A$.
(b) $\mathbf{d}_x[x'Ax] = 2x'A$.

PROOF. (a) is proven by writing each element of Ax as a sum and taking partial derivatives. (b) is proven by writing $x'Ax$ as a double sum and taking partial derivatives. □

Proposition F.2 For (c) and (d), $t = s$.
(a) A form of the product rule holds for conformable matrices $A(u)$ and $B(u)$,

$$\mathbf{d}_u[A(u)B(u)] = [\mathbf{d}_u A(u)]B(u) + A(u)[\mathbf{d}_u B(u)].$$

(b) When B and C are fixed matrices of conformable sizes,

$$\mathbf{d}_u[CA(u)B] = C[\mathbf{d}_u A(u)]B.$$

(c) The derivative of an inverse is

$$\mathbf{d}_u A^{-1}(u) = -A^{-1}(u)[\mathbf{d}_u A(u)]A^{-1}(u).$$

(d) The derivative of a trace is

$$\mathbf{d}_u\{\mathrm{tr}[A(u)]\} = \mathrm{tr}[\mathbf{d}_u A(u)].$$

(e) If $V(u)$ is positive definite for all u,

$$\mathbf{d}_u \log\{\det[V(u)]\} = \mathrm{tr}\left\{V^{-1}(u)[\mathbf{d}_u V(u)]\right\}.$$

The notations $\det[V]$ and $|V|$ are used interchangeably to indicate the determinant.

PROOF. The proof is an exercise in *ALM-III*.

The following results are specific to applications in Appendix D. They use results on differentiation, Vec operators, and Kronecker products (denoted $\otimes$) from Appendix A in *ALM-III*. The Vec operator simply stacks the columns of a matrix and we need to vectorize any matrices prior to applying differentiation results to them. When g is a scalar function applied to any matrix $W = [w_{ij}]$, then $g(W) \equiv [g(w_{ij})]$, and in turn,

$$\mathrm{Vec}[g(W)] = g[\mathrm{Vec}(W)].$$

For a scalar function g with derivative $\dot{g}$ and v a vector, $\mathbf{d}_v g(v)$ is the diagonal matrix with entries $\dot{g}(v)$. In particular, $\mathbf{d}_v v = I$.

In Appendix D on neural nets, $g[x'B_1]$ is $1 \times r$ because B_1 is $p \times r$. Thus,

$$\begin{aligned} g[x'B_1]' &= \mathrm{Vec}\{g[x'B_1]\} \\ &= g[\mathrm{Vec}\{x'B_1\}] \\ &= g\{[I_r \otimes x']\mathrm{Vec}(B_1)\}. \end{aligned}$$

Now, using the chain rule,

$$\begin{aligned} \mathbf{d}_{\mathrm{Vec}(B_1)} g[x'B_1]' &= \mathbf{d}_{\mathrm{Vec}(B_1)} g\{[I_r \otimes x']\mathrm{Vec}(B_1)\} \\ &= \left\{\mathbf{d}_v g(v)|_{v=[I_r\otimes x']\mathrm{Vec}(B_1)}\right\}\{\mathbf{d}_{\mathrm{Vec}(B_1)}[I_r \otimes x']\mathrm{Vec}(B_1)\} \\ &= \left\{\mathbf{d}_v g(v)|_{v=[I_r\otimes x']\mathrm{Vec}(B_1)}\right\}\{[I_r \otimes x']\mathbf{d}_{\mathrm{Vec}(B_1)}\mathrm{Vec}(B_1)\} \\ &= \mathrm{Diag}(\dot{g}\{[I_r \otimes x']\mathrm{Vec}(B_1)\})[I_r \otimes x']I_{rp} \\ &= \mathrm{Diag}(\dot{g}[x'B_1])'[I_r \otimes x']. \end{aligned}$$

Appendix B
A Three-Factor ANOVA

Table B.1 is derived from Scheffé (1959) and gives the moisture content (in grams) for samples of a food product made with three kinds of salt (A), three amounts of salt (B), and two additives (C). The amounts of salt, as measured in moles, are equally spaced. This has $18 = 3 \times 3 \times 2$ treatment cells in the ANOVA. The cells are marked by vertical and horizontal lines. The two numbers listed for some treatment cells are replications. We wish to analyze these data.

Table B.1 Moisture content of a food product.

A (salt i)		1			2			3		
B (amount salt j)		1	2	3	1	2	3	1	2	3
C (additive k)	1	8	17	22	7	26	34	10	24	39
			13	20	10	24		9		36
	2	5	11	16	3	17	32	5	16	33
		4	10	15	5	19	29	4		34

We can consider these data as a one-way ANOVA with $18 = 3 \times 3 \times 2$ groups of observations where the groups are a unique combination of a salt, an amount of salt, and an additive. The model for such data can be taken as

$$y_{ijkm} = \mu_{ijk} + e_{ijkm}$$

where together ijk indicates one of the 18 groups. In this example, each group has either 1 or 2 observations in it. To test whether the 18 groups have different means, we fit the reduced model

$$y_{ijkm} = \mu + e_{ijkm}$$

The results from fitting these models and performing the test is usually summarized in a three-line ANOVA table

Analysis of Variance: Moisture content data.

Source	*df*	*SS*	*MS*	*F*	*P*
Groups	17	3643.22	214.307	92.32	0.000
Error	14	32.50	2.3214		
Total	31	3675.72			

With $F_{obs} = 92.32$ it is pretty clear that the 18 groups do not all have the same mean value.

B.1 Three-way ANOVA

We now consider these data as a three-factor ANOVA. From the structure of the replications the ANOVA has unequal numbers. The general model for a three-factor ANOVA with replications is

$$y_{ijkm} = G + A_i + B_j + C_k + [AB]_{ij} + [AC]_{ik} + [BC]_{jk} + [ABC]_{ijk} + e_{ijkm}.$$

For example, A_i indicates a main effect for the type of salt, $[AB]_{ij}$ indicates a two-factor interaction between type and amount of salt, while $[ABC]_{ijk}$ indicates a three-factor interaction between type of salt, amount of salt, and the additive. Mathematically, μ_{ijk} and $[ABC]_{ijk}$ are equivalent terms and including the three factor interaction makes all of the main effects and two-factor interactions redundant. Our first priority is to find out which interactions are important.

Table B.2 contains the sum of squares for error and the degrees of freedom for error for all the ANOVA models that include all of the main effects. Each model is identified in the table by the highest-order terms in the model. For example, $[AB][AC]$ indicates the model

$$y_{ijkm} = G + A_i + B_j + C_k + [AB]_{ij} + [AC]_{ik} + e_{ijkm}$$

with only the $[AB]$ and $[AC]$ interactions. In $[AB][AC]$, the grand mean and all of the main effects are redundant; it does not matter whether these terms are included in the model. Similarly, $[AB][C]$ indicates the model

$$y_{ijkm} = G + A_i + B_j + C_k + [AB]_{ij} + e_{ijkm}$$

with the $[AB]$ interaction and the C main effect. In $[AB][C]$, the grand mean and the A and B main effects are redundant. Readers familiar with methods for fitting log-linear models (cf. Christensen, 1997 or Fienberg, 1980) will notice a correspondence between Table B.2 and similar displays used in fitting three-dimensional contingency tables. The analogies between selecting log-linear models and selecting models for unbalanced ANOVA are pervasive.

All of the models have been compared to the full model using F statistics in Table B.2. It takes neither a genius nor an F table to see that the only models that

Table B.2 Statistics for fitting models to the data of Table B.1.

Model	SSE	dfE	F*	C_p
$[ABC]$	32.50	14		18.0
$[AB][AC][BC]$	39.40	18	0.743	13.0
$[AB][AC]$	45.18	20	0.910	11.5
$[AB][BC]$	40.46	20	0.572	9.4
$[AC][BC]$	333.2	22	16.19	131.5
$[AB][C]$	45.75	22	0.713	7.7
$[AC][B]$	346.8	24	13.54	133.4
$[BC][A]$	339.8	24	13.24	130.4
$[A][B][C]$	351.1	26	11.44	131.2

*The F statistics are for testing each model against the model with a three-factor interaction, i.e., $[ABC]$. The denominator of each F statistic is $MSE([ABC]) = 32.50/14 = 2.3214$.

fit the data are the models that include the $[AB]$ interaction. The C_p statistics tell the same story.

In addition to testing models against the three-factor interaction model, there are a number of other comparisons that can be made among models that include $[AB]$. These are $[AB][AC][BC]$ versus $[AB][AC]$, $[AB][AC][BC]$ versus $[AB][BC]$, $[AB][AC][BC]$ versus $[AB][C]$, $[AB][AC]$ versus $[AB][C]$, and $[AB][BC]$ versus $[AB][C]$. None of the comparisons show any lack of fit. The last two comparisons are illustrated below.

$$[AB][AC] \text{ versus } [AB][C]$$

$$R(AC|AB,C) = 45.75 - 45.18 = 0.57$$

$$F_{obs} = (0.57/2)/2.3214 = 0.123$$

$$[AB][BC] \text{ versus } [AB][C]$$

$$R(BC|AB,C) = 45.75 - 40.46 = 5.29$$

$$F_{obs} = (5.29/2)/2.3214 = 1.139.$$

Here we use the $R(\cdot|\cdot)$ notation that was introduced in Subsection 1.5.1 (along with the $SSR(\cdot|\cdot)$ notation). The denominator in each test is $MSE([ABC])$, i.e., the variance estimate from the biggest model under consideration.

The smallest model that seems to fit the data adequately is $[AB][C]$. This is indicated by the C_p statistic but also the F statistics for comparing $[AB][C]$ to the larger models are all extremely small. Writing out the model $[AB][C]$, it is

$$y_{ijkm} = G + A_i + B_j + C_k + [AB]_{ij} + e_{ijkm}.$$

We need to examine the $[AB]$ interaction. Since the levels of B are quantitative, a model that is equivalent to $[AB][C]$ is a model that includes the main effects for C, but, instead of fitting an interaction in A and B, fits a separate regression equation in the levels of B for each level of A. Let x_j, $j = 1,2,3$ denote the levels of B. There are three levels of B, so the most general polynomial we can fit is a second-degree polynomial in x_j. Since the amounts of salt were equally spaced, it does not matter much what we use for the x_js. The computations were performed using $x_1 = 1$, $x_2 = 2$, $x_3 = 3$. In particular, the model $[AB][C]$ was reparameterized as

$$y_{ijkm} = A_{i0} + A_{i1}x_j + A_{i2}x_j^2 + C_k + e_{ijkm}. \tag{1}$$

The nature of this model is that for a fixed additive, the three curves for the three salts can take any shapes at all. However, if you change to the other additive all three of the curves will shift, either up or down, exactly the same amount due to the change in additive. The shapes of the curves do not change.

With a notation similar to that used in Table B.2, the *SSE* and the *dfE* are reported in Table B.3 for Model (1) and three reduced models. Note that the *SSE* and *dfE* reported in Table B.3 for $[A_0][A_1][A_2][C]$ are identical to the values reported in Table B.2 for $[AB][C]$. This, of course, must be true if the models are merely reparameterizations of one another. First we want to establish whether the quadratic effects are necessary in the regressions. To do this we drop the A_{i2} terms from Model (1) and test

$$[A_0][A_1][A_2][C] \text{ versus } [A_0][A_1][C]$$

$$R(A_2|A_1,A_0,C) = 59.98 - 45.75 = 14.23$$

$$F_{obs} = (14.23/3)/2.3214 = 2.04.$$

Since $F(.95,3,14) = 3.34$, there is no evidence of any nonlinear effects.

Table B.3 Additional statistics for data of Table B.1.

Model	*SSE*	*dfE*
$[A_0][A_1][A_2][C]$	45.75	22
$[A_0][A_1][C]$	59.98	25
$[A_0][A_1]$	262.0	26
$[A_0][C]$	3130.	28

At this point it might be of interest to test whether there are any linear effects. This is done by testing $[A_0][A_1][C]$ against $[A_0][C]$. The statistics needed for this test are in Table B.3. Instead of actually doing the test, recall that no models in Table B.2 fit the data unless they included the $[AB]$ interaction. If we eliminated the linear effects we would have a model that involved none of the $[AB]$ interaction. (The model $[A_0][C]$ is identical to the ANOVA model $[A][C]$.) We already know that such models do not fit.

Finally, we have never explored the possibility that there is no main effect for C. This can be done by testing

$$[A_0][A_1][C] \text{ versus } [A_0][A_1]$$

$$R(C|A_1,A_0) = 262.0 - 59.98 = 202$$

$$F_{obs} = (202/1)/2.3214 = 87.$$

Obviously, there is a substantial main effect for C, the type of food additive.

Our conclusion is that the model $[A_0][A_1][C]$ is the smallest model that has been considered that adequately fits the data. This model indicates that there is an effect for the type of additive and a linear relationship between amount of salt and moisture content. The slope and intercept of the line may depend on the type of salt. (The intercept of the line also depends on the type of additive.) Table B.4 contains parameter estimates and standard errors for the model. All estimates in the example use the side condition $C_1 = 0$.

Table B.4 $y_{ijkm} = A_{i0} + A_{i1}x_j + C_k + e_{ijkm}$.

Table of Coefficients		
Parameter	Estimate	SE
A_{10}	3.35	1.375
A_{11}	5.85	0.5909
A_{20}	−3.789	1.237
A_{21}	13.24	0.5909
A_{30}	−4.967	1.231
A_{31}	14.25	0.5476
C_1	0.	none
C_2	−5.067	0.5522

Note that, in lieu of the F test given earlier, the test for the main effect C could be performed from Table B.4 by looking at $t = -5.067/.5522 = -9.176$. Moreover, we should have $t^2 = F$. The t statistic squared is 84, while the F statistic reported earlier is 87. The difference is due to the fact that the SE reported in Table B.4 uses the MSE for the model being fitted, while in performing the F test we used $MSE([ABC])$.

Are we done yet? No. The parameter estimates suggest some additional questions. Are the slopes for salts 2 and 3 the same, i.e., is $A_{21} = A_{31}$? In fact, are the entire lines for salts 2 and 3 the same, i.e., are $A_{21} = A_{31}$, $A_{20} = A_{30}$? We can fit models that incorporate these assumptions.

Model	SSE	dfE
$[A_0][A_1][C]$	59.98	25
$[A_0][A_1][C], A_{21} = A_{31}$	63.73	26
$[A_0][A_1][C], A_{21} = A_{31}, A_{20} = A_{30}$	66.97	27

It is a small matter to check that there is no lack of fit displayed by any of these models. The smallest model that fits the data is now $[A_0][A_1][C]$, $A_{21} = A_{31}$, $A_{20} = A_{30}$. Thus there seems to be no difference between salts 2 and 3, but salt 1 has a different regression than the other two salts. (We did not actually test whether salt 1 is different, but if salt 1 had the same slope as the other two then there would be no $[AB]$ interaction and we know that interaction exists.) There is also an effect for the food additives. The parameter estimates and standard errors for the final model are given in Table B.5.

Table B.5 $y_{ijkm} = A_{i0} + A_{i1}x_j + C_k + e_{ijkm}$, $A_{21} = A_{31}$, $A_{20} = A_{30}$.

Table of Coefficients		
Parameter	Estimate	SE
A_{10}	3.395	1.398
A_{11}	5.845	0.6008
A_{20}	−4.466	0.9030
A_{21}	13.81	0.4078
C_1	0.	none
C_2	−5.130	0.5602

Figure B.1 shows the fitted values as functions of the amount of salt for each combination of a salt (with salts 2 and 3 treated as the same) and the additive. The fact that the slope for salt 1 is different from the slope for salts 2 and 3 constitutes an AB interaction. The vertical distances between the two lines for each salt are the same due to the simple main effect for C (additive). The two lines are shockingly close at $x_1 = 1$, which makes one wonder if perhaps $j = 1$ is a condition of no salt being used.

If $j = 1$ really consists of not adding salt, then, when $j = 1$, the means should be identical for the three salts. The additives can still affect the moisture contents and positive salt amounts can affect the moisture contents. To incorporate these ideas, we subtract one from the salt amounts and eliminate the intercepts from the lines in the amount of salt. That makes the effects for the additive the de facto intercepts, and they are no longer overparameterized,

$$y_{ijkm} = C_k + A_{i1}(x_j - 1) + e_{ijkm}, \qquad A_{21} = A_{31}.$$

This model has $dfE = 28$ and $SSE = 67.0$ so it fits the data almost as well as the previous model but with one less parameter. The estimated coefficients are given in Table B.6 and the results are plotted in Figure B.2. The figure is almost identical to Figure B.1. Note that the vertical distances between the two lines with "the same" salt in Figure B.2 are $5.1347 = 9.3162 - 4.1815$, almost identical to the 5.130 in Figure B.1.

Are we done yet? Probably not. We have not even considered the validity of the assumptions. Are the errors normally distributed? Are the variances the same for every treatment combination? Technically, we need to ask whether $C_1 = C_2$ in this

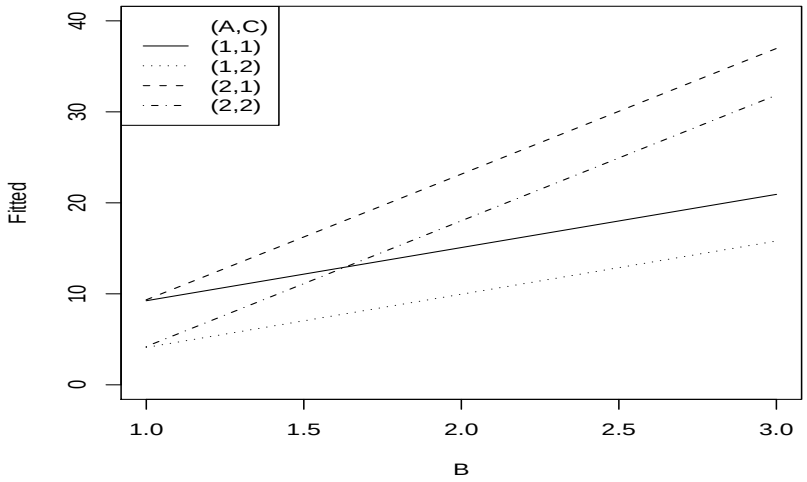

Fig. B.1 Fitted values for moisture content data treating salts 2 and 3 as the same.

Table B.6 $y_{ijkm} = C_k + A_{i1}(x_j - 1) + e_{ijkm}$, $A_{21} = A_{31}$.

Table of Coefficients			
Parameter	Estimate	SE	t_{obs}
C_1	9.3162	0.5182	17.978
C_2	4.1815	0.4995	8.371
A_{11}	5.8007	0.4311	13.456
A_{21}	13.8282	0.3660	37.786

new model. A quick look at the estimates and standard errors answers the question in the negative.

B.2 Computing

We now present and contrast R and SAS code for fitting $[AB][C]$ and discuss the fitting of other models from this section. Table B.7 illustrates the variables needed for a full analysis. The online data file contains only the y values and indices for the three groups. Creating X and X2 is generally easy. Creating the variable A2 that does not distinguish between salts 2 and 3 can be trickier. If we had a huge number of observations, we would want to write a program to modify A into A2. With the data we have, in Minitab it is easy to make a copy of A and modify it appropriately in the spreadsheet. Similarly, it is easy to create A2 in R using `A2=A` followed by

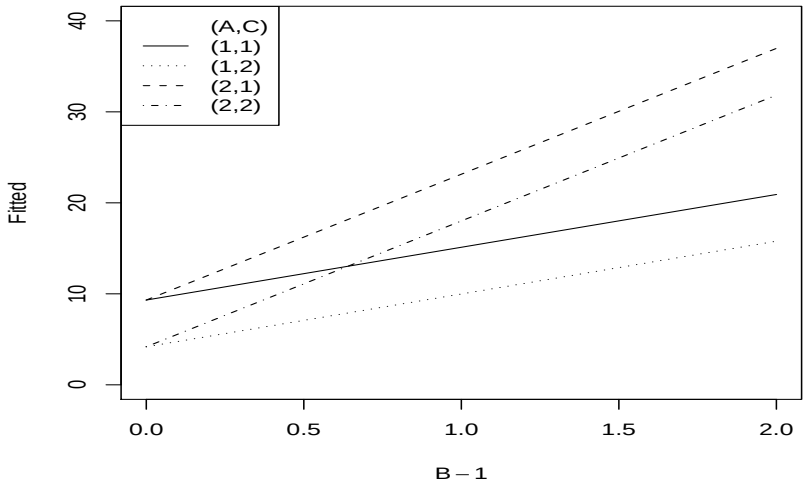

Fig. B.2 Fitted values for moisture content data treating salts 2 and 3 as the same and $B = 1$ as 0 salt.

`A2[(A2 == 3)] <- 2`. For SAS, I would probably modify the data file so that I could read A2 with the rest of the data.

Table B.7 Moisture data, indices, and predictors.

	A	B	C	X	X2	A2		A	B	C	X	X2	A2
y	i	j	k	x	x^2		y	i	j	k	x	x^2	
8	1	1	1	1	1	1	11	1	2	2	2	4	1
17	1	2	1	2	4	1	16	1	3	2	3	9	1
22	1	3	1	3	9	1	3	2	1	2	1	1	2
7	2	1	1	1	1	2	17	2	2	2	2	4	2
26	2	2	1	2	4	2	32	2	3	2	3	9	2
34	2	3	1	3	9	2	5	3	1	2	1	1	2
10	3	1	1	1	1	2	16	3	2	2	2	4	2
24	3	2	1	2	4	2	33	3	3	2	3	9	2
39	3	3	1	3	9	2	4	1	1	2	1	1	1
13	1	2	1	2	4	1	10	1	2	2	2	4	1
20	1	3	1	3	9	1	15	1	3	2	3	9	1
10	2	1	1	1	1	2	5	2	1	2	1	1	2
24	2	2	1	2	4	2	19	2	2	2	2	4	2
9	3	1	1	1	1	2	29	2	3	2	3	9	2
36	3	3	1	3	9	2	4	3	1	2	1	1	2
5	1	1	2	1	1	1	34	3	3	2	3	9	2

An R script for fitting $[AB][C]$ follows. R needs to locate the data file, which in this case is located at `E:\Books\ANREG2\DATA2\tab16-1.dat`.

```
scheffe <- read.table("E:\\Books\\ANREG2\\DATA2\\tab16-1.dat",
        sep="",col.names=c("y","a","b","c"))
attach(scheffe)
scheffe
summary(scheffe)

#Summary tables
A=factor(a)
B=factor(b)
C=factor(c)
X=b
X2=X*X
sabc <- lm(y ~ A:B + C)
coef=summary(sabc)
coef
anova(sabc)
```

SAS code for fitting $[AB][C]$ follows. The code assumes that the data file is the same directory (folder) as the SAS file.

```
options ps=60 ls=72 nodate;
data anova;
   infile 'tab16-1.dat';
   input y A B C;
   X = B;
   X2=X*X;
proc glm data=anova;
   class A B C ;
   model y = A*B C ;
   means C / lsd alpha=.01 ;
   output out=new r=ehat p=yhat cookd=c h=hi rstudent=tresid student=sr;
proc plot;
   plot ehat*yhat sr*R/  vpos=16 hpos=32;
proc rank data=new normal=blom;
   var sr;
   ranks nscores;
proc plot;
   plot sr*nscores/vpos=16 hpos=32;
run;
```

To fit the other models, one needs to modify the part of the code that specifies the model. In R this involves changes to "`sabc <- lm(y ~ A:B + C)`" and in SAS it involves changes to "`model y = A*B C;`". Alternative model specifications follow.

Model	Minitab	R	SAS
$[ABC]$	$A\|B\|C$	A:B:C	A*B*C
$[AB][BC]$	$A\|B \quad B\|C$	A:B+B:C	A*B B*C
$[AB][C]$	$A\|B \quad C$	A:B+C	A*B C
$[A_0][A_1][A_2][C]$	$A\|X \quad A\|X2 \quad C$	A+A:X+A:X2+C	A A*X A*X2 C
$[A_0][A_1][C], A_{21} = A_{31}$	$A \quad A2\|X \quad C$	A+A2:X+C-1	A A2*X C
$[A_0][A_1][C], A_{21} = A_{31}, A_{20} = A_{30}$	$A2 \quad A2\|X \quad C$	A2+A2:X+C-1	A2 A2*X C

B.3 Regression fitting

We start by creating 0-1 indicator variables for the factor variables A, B, and C. Call these, $A_1, A_2, A_3, B_1, B_2, B_3, C_1, C_2$, respectively. The values used to identify groups in factor variable B are measured quantities, so create a measurement variable $x \equiv B$ and another x^2. We can construct all of the models from these 10 predictor variables by multiplying them together judiciously. (For example, A_1x is the product of the A_1 variable and the x variable and A_1B_2 is a similar product.) Of course there are many equivalent ways of specifying these models; we present only one. None of the models contain an intercept.

Model	Variables
$[ABC]$	$A_1B_1C_1$, $A_1B_1C_2$, $A_1B_2C_1$, $A_1B_2C_2$, $A_1B_3C_1$, …, $A_3B_3C_1$, $A_3B_3C_2$
$[AB][AC][BC]$	A_1B_1, A_1B_2, …, A_3B_3, A_1C_2, A_2C_2, A_3C_2, B_2C_2, B_3C_2
$[AB][BC]$	A_1B_1, A_1B_2, …, A_3B_3, B_1C_2, B_2C_2, B_3C_2
$[AB][C]$	A_1B_1, A_1B_2, …, A_3B_3, C_2
$[A][B][C]$	A_1, A_2, A_3, B_2, B_3, C_2
$[A_0][A_1][A_2][C]$	A_1, A_2, A_3, A_1x, A_2x, A_3x, A_1x^2, A_2x^2, A_3x^2, C_2
$[A_0][A_1][C]$	A_1, A_2, A_3, A_1x, A_2x, A_3x, C_2
$[A_0][A_1]$	A_1, A_2, A_3, A_1x, A_2x, A_3x
$[A_0][C]$	A_1, A_2, A_3, C_2

Constructing the models in which salts 2 and 3 are treated alike requires some additional algebra.

Model	Variables
$[A_0][A_1][C], A_{21} = A_{31}$	A_1, A_2, A_3, A_1x, $(A_2+A_3)x$, C_2
$[A_0][A_1][C], A_{21} = A_{31}, A_{20} = A_{30}$	A_1, (A_2+A_3), A_1x, $(A_2+A_3)x$, C_2

Appendix C
MANOVA

Table C.1 gives data from Box (1950) on the abrasion resistance of a fabric. The data are weight loss of a fabric that occurs during the first 1000 revolutions of a machine designed to test abrasion resistance y_1, during the second 1000 revolutions y_2, and during the third 1000 revolutions y_3. A piece of fabric is weighed, put on the machine for 1000 revolutions, and weighed again. The measurement is the change in weight. This is done three times for each piece of fabric. Fabrics of several different types are compared. They differ by whether a surface treatment was applied, the type of filler used, and the proportion of filler used. Two pieces of fabric of each type are examined, giving two replications in the analysis of variance. Here we view measurements on different pieces of fabric as independent but the three measurements on each piece as possibly correlated.

Table C.1 Abrasion resistance data.

Surf.		Proportions								
		25%			50%			75%		
treat.	Fill	1000	2000	3000	1000	2000	3000	1000	2000	3000
Yes	A	194	192	141	233	217	171	265	252	207
	A	208	188	165	241	222	201	269	283	191
	B	239	127	90	224	123	79	243	117	100
	B	187	105	85	243	123	110	226	125	75
No	A	155	169	151	198	187	176	235	225	166
	A	173	152	141	177	196	167	229	270	183
	B	137	82	77	129	94	78	155	76	91
	B	160	82	83	98	89	48	132	105	67

The data involve three explanatory factors: Surface treatment (yes, no), Fill (A, B), and Proportion of fill (25%, 50%, 75%). These are referred to as **S**, **F**, and **P**, respectively. (We hope no confusion occurs between the factor **F** and the use of *F*

statistics or between the factor **P** and the use of P values!) In analyzing y_1, y_2, y_3, many aspects are just simple extensions of the analysis on a single y_r, cf. Appendix B.

The multivariate approach to analyzing data that contain multiple measurements on each subject involves using the multiple measures as separate dependent variables in a collection of standard analyses each involving a single dependent variable. The method of analysis known as *multivariate analysis of variance (MANOVA)*, or with more generality as *multivariate linear models*, then combines results from the several linear models. A detailed discussion of MANOVA is beyond the scope of this book, but we present a short introduction to some of the underlying ideas.

After introducing multivariate linear models in general, for simplicity we focus on a balanced analysis of variance. There is nothing in the general theory that requires balance except that there be no missing observations among the multiple measures on a subject. Entirely missing a subject causes few problems. The discussion in *ALM* is quite general but at a higher mathematical level. Almost all Statistics books on Multivariate Analysis deal with MANOVA. Johnson and Wichern (2007) or Johnson (1998) are reasonable places to look for more information on the subject.

C.1 Multivariate Linear Models

The distinction between standard univariate linear models and standard multivariate linear models is simply that multivariate linear models involve more than one dependent variable. For multivariate data, let the dependent variables be $y_1, \ldots, y_q$. The idea is that all q random variables will be observed on each of n individuals. The standard assumption is that the random variables have some unknown covariance matrix Σ that is the same for all individuals but different individuals are uncorrelated. If n observations are taken on each dependent variable, we have $y_{i1}, \ldots, y_{iq}$, $i = 1, \ldots, n$. Let $Y_1 = [y_{11}, \ldots, y_{n1}]'$ and, in general, $Y_h = [y_{1h}, \ldots, y_{nh}]'$, $h = 1, \ldots, q$. For each h, the vector Y_h is the vector of n responses on the variable y_h and can be used as the response vector for a linear model. For $h = 1, \ldots, q$, write the linear model

$$Y_h = X\beta_h + e_h, \quad \mathrm{E}(e_h) = 0, \quad \mathrm{Cov}(e_h) = \sigma_{hh} I, \tag{1}$$

where X is a known $n \times p$ matrix that is the same for all dependent variables (it depends on the individuals but not on the variable being measured), but β_h and the error vector $e_h = [e_{1h}, \ldots, e_{nh}]'$ are peculiar to the dependent variable. Here we are using σ_{hh} (rather than σ_h^2) to denote the variance associated with y_h.

The multivariate linear model consists of fitting the q linear models simultaneously. Write the matrices

$$Y_{n\times q} = [Y_1, \ldots, Y_q], \quad B_{p\times q} = [\beta_1, \ldots, \beta_q], \quad e_{n\times q} = [e_1, \ldots, e_q].$$

The multivariate linear model is

$$Y = XB + e\,. \tag{2}$$

The key to the analysis of the standard multivariate linear model is the random nature of the $n \times q$ error matrix $e = [e_{ih}]$. At a minimum, we assume that $\mathrm{E}(e) = 0$ and that different individuals i are uncorrelated,

$$\mathrm{Cov}(e_{ih}, e_{i'h'}) = \begin{cases} \sigma_{hh'} & \text{if } i = i' \\ 0 & \text{if } i \neq i' \end{cases}.$$

Let

$$\delta_{ii'} = \begin{cases} 1 & \text{if } i = i' \\ 0 & \text{if } i \neq i' \end{cases},$$

then the covariances can be written simply as

$$\mathrm{Cov}(e_{ih}, e_{i'h'}) = \sigma_{hh'}\delta_{ii'}\,.$$

To construct tests and confidence regions, we would assume that the e_{ij}s have a multivariate normal distribution with the previously indicated mean and covariances. Note that this covariance structure implies that the error vector in model (1) has $\mathrm{Cov}(e_h) = \sigma_{hh}I$, as indicated previously.

An alternative but equivalent way to state the standard multivariate linear model is by examining the rows of model (2). Write

$$Y = \begin{bmatrix} y_1' \\ \vdots \\ y_n' \end{bmatrix}, \quad X = \begin{bmatrix} x_1' \\ \vdots \\ x_n' \end{bmatrix}, \quad \text{and} \quad e = \begin{bmatrix} \varepsilon_1' \\ \vdots \\ \varepsilon_n' \end{bmatrix}.$$

The standard multivariate linear model is also

$$y_i' = x_i'B + \varepsilon_i', \qquad i = 1, \ldots, n. \tag{3}$$

The error vector ε_i has the properties

$$\mathrm{E}(\varepsilon_i) = 0, \qquad \mathrm{Cov}(\varepsilon_i) = [\sigma_{hh'}] \equiv \Sigma_{q\times q},$$

and, for $i \neq j$,

$$\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0.$$

To construct tests and confidence regions, the vectors ε_i are assumed to have independent multivariate normal distributions.

To reiterate, the multivariate model (2) holds if and only if the q univariate models (1) hold simultaneously and the multivariate model holds if and only if the n models in (3) hold simultaneously. All of these models have errors with mean zero and all models determine the same covariance structure. The unknown covariance parameters are the unique parameters in Σ. We assume throughout that Σ is positive definite.

C.2 MANOVA Example

Again consider the Box (1950) data. In the multivariate approach, we begin by fittin separate ANOVA models for the data from 1000 rotations, 2000 rotations, and 3000 rotations. The variables $y_{hijk,1}$, $y_{hijk,2}$, and $y_{hijk,3}$ denote the data from 1000, 2000, and 3000 rotations, respectively. We fit the models

$$\begin{aligned} y_{hijk,1} &= \mu_{hij,1} + \varepsilon_{hijk,1} \\ &= \mu_1 + s_{h,1} + f_{i,1} + p_{j,1} \\ &\quad + (sf)_{hi,1} + (sp)_{hj,1} + (fp)_{ij,1} + (sfp)_{hij,1} + \varepsilon_{hijk,1}, \end{aligned}$$

$$\begin{aligned} y_{hijk,2} &= \mu_{hij,2} + \varepsilon_{hijk,2} \\ &= \mu_2 + s_{h,2} + f_{i,2} + p_{j,2} \\ &\quad + (sf)_{hi,2} + (sp)_{hj,2} + (fp)_{ij,2} + (sfp)_{hij,2} + \varepsilon_{hijk,2}, \end{aligned}$$

and

$$\begin{aligned} y_{hijk,3} &= \mu_{hij,3} + \varepsilon_{hijk,3} \\ &= \mu_3 + s_{h,3} + f_{i,3} + p_{j,3} \\ &\quad + (sf)_{hi,3} + (sp)_{hj,3} + (fp)_{ij,3} + (sfp)_{hij,3} + \varepsilon_{hijk,3} \end{aligned}$$

$h = 1,2$, $i = 1,2$, $j = 1,2,3$, $k = 1,2$. The first versions of these models (the ones written with $\mu_{hij,m}$) are one-way ANOVA (regression) models with $t = 12$ groups. The three indices hij together identify the 12 groups. The second version of each model is equivalent to the first version but the second version exploits the particular (factorial) nature of how the 12 treatments are defined. The second version involves main effects for the three factors **S**, **F**, and **P**, two-factor interactions between pairs of factors, and a three-way interaction between the three factors, which is nothing more than a renaming of $\mu_{hij,m}$ as $(sfp)_{hij,m}$. The main effects and two-factor interactions are only of interest when the $(sfp)_{hij,m}$s have been dropped out of the models (and even then, unless you drop out at least 2 two-factor interactions, the main effects are all extraneous).

As in standard ANOVA models, we assume that the individuals (on which the repeated measures were taken) are independent. Thus, for fixed $m = 1,2,3$, the $\varepsilon_{hijk,m}$s are independent $N(0,\sigma_{mm})$ random variables. Again we are using a double subscript in σ_{mm} to denote a variance rather than writing σ_m^2. As usual, the errors on a common dependent variable, say $\varepsilon_{hijk,m}$ and $\varepsilon_{h'i'j'k',m}$, are independent for different individuals, i.e., when $(h,i,j,k) \neq (h',i',j',k')$, but we also assume that the errors on different dependent variables, say $\varepsilon_{hijk,m}$ and $\varepsilon_{h'i'j'k',m'}$, are independent when $(h,i,j,k) \neq (h',i',j',k')$. However, not all of the errors for all the variables are assumed independent. Two observations (or errors) on the same subject are *not* assumed to be independent. For fixed h, i, j, k the errors for any two variables are possibly correlated with, say, $\mathrm{Cov}(\varepsilon_{hijk,m}, \varepsilon_{hijk,m'}) = \sigma_{mm'}$.

The models for each variable m are of the same form but the parameters differ for the different dependent variables $y_{hijk,m}$. All the parameters have an additional subscript to indicate which dependent variable m they belong to. The essence of the procedure is simply to fit each of the models individually and then to combine results. Fitting individually gives three separate sets of residuals, $\hat{\varepsilon}_{hijk,m} = y_{hijk,m} - \bar{y}_{hij\cdot,m}$ for $m = 1,2,3$, so three separate sets of residual plots and three separate ANOVA tables. The three ANOVA tables are given as Tables C.2, C.3, and C.4. Residual plots for y_3 are given as Figures C.1 and C.2 but similar plots for y_1 and y_2 should also be examined. Each variable can be analyzed in detail using the ordinary methods for multifactor ANOVA models illustrated in Appendix B. (The analysis in Appendix B is more complicated because the ANOVA there is unbalanced.) If we ignored the factorial group structure of the treatments, the one-way ANOVA would provide a three line ANOVA table with the same Total and Error lines but the rest of the Source rows would be consolidated into a single row of the ANOVA table and labeled Groups (or Treatments) with $12-1 = 1+1+2+1+2+2+2$ degrees of freedom and provide an F test for whether any of the 12 groups were different from each other.

Table C.2 Analysis of variance for y_1.

Source	*df*	*SS*	*MS*	*F*	*P*
S	1	26268.2	26268.2	97.74	0.000
F	1	6800.7	6800.7	25.30	0.000
P	2	5967.6	2983.8	11.10	0.002
S∗**F**	1	3952.7	3952.7	14.71	0.002
S∗**P**	2	1186.1	593.0	2.21	0.153
F∗**P**	2	3529.1	1764.5	6.57	0.012
S∗**F**∗**P**	2	478.6	239.3	0.89	0.436
Error	12	3225.0	268.8		
Total	23	51407.8			

The key to multivariate analysis of variance is to combine results *across* the three variables y_1, y_2, and y_3. Recall that the mean squared errors are just the sums of the squared residuals divided by the error degrees of freedom, i.e.,

$$MSE_{mm} \equiv s_{mm} = \frac{1}{dfE}\sum_{hijk}\hat{\varepsilon}^2_{hijk,m}.$$

This provides an estimate of σ_{mm}. We can also use the residuals to estimate covariances between the three variables. The estimate of $\sigma_{mm'}$ is

$$MSE_{mm'} \equiv s_{mm'} = \frac{1}{dfE}\sum_{hijk}\hat{\varepsilon}_{hijk,m}\hat{\varepsilon}_{hijk,m'}.$$

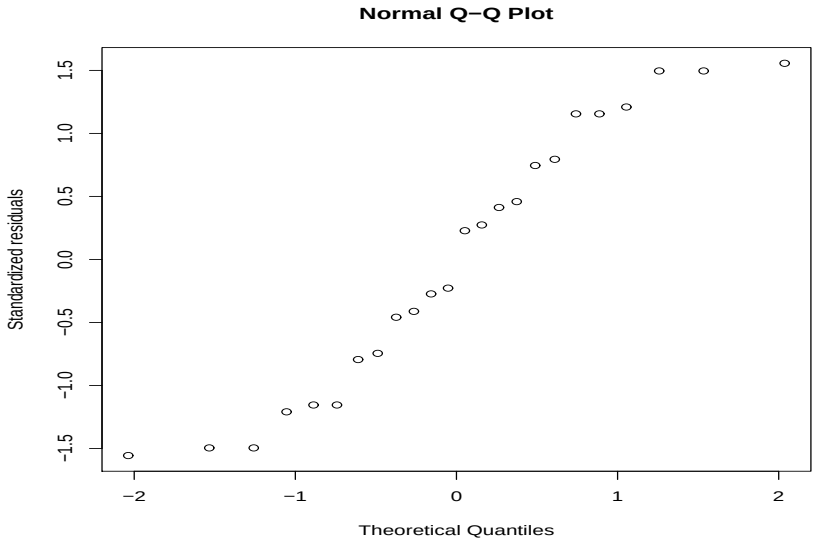

Fig. C.1 Normal plot for y_3, $W' = 0.94$.

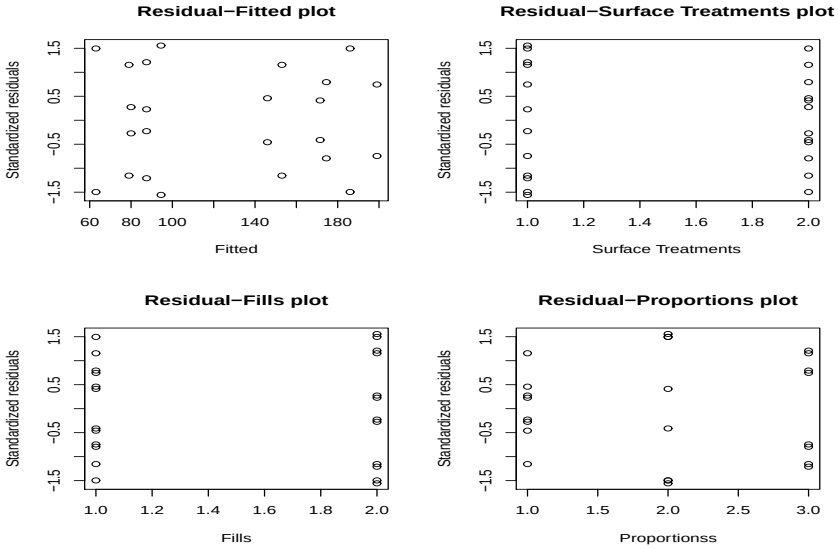

Fig. C.2 Residual plots for y_3.

Table C.3 Analysis of variance for y_2.

Source	*df*	*SS*	*MS*	*F*	*P*
S	1	5017.0	5017.0	25.03	0.000
F	1	70959.4	70959.4	353.99	0.000
P	2	7969.0	3984.5	19.88	0.000
S∗**F**	1	57.0	57.0	0.28	0.603
S∗**P**	2	44.3	22.2	0.11	0.896
F∗**P**	2	6031.0	3015.5	15.04	0.001
S∗**F**∗**P**	2	14.3	7.2	0.04	0.965
Error	12	2405.5	200.5		
Total	23	92497.6			

Table C.4 Analysis of variance for y_3.

Source	*df*	*SS*	*MS*	*F*	*P*
S	1	1457.0	1457.0	6.57	0.025
F	1	48330.4	48330.4	217.83	0.000
P	2	1396.6	698.3	3.15	0.080
S∗**F**	1	0.4	0.4	0.00	0.968
S∗**P**	2	250.6	125.3	0.56	0.583
F∗**P**	2	1740.3	870.1	3.92	0.049
S∗**F**∗**P**	2	272.2	136.1	0.61	0.558
Error	12	2662.5	221.9		
Total	23	56110.0			

We now form the estimates into a matrix of estimated covariances

$$\mathscr{S} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}.$$

Note that $s_{mm'} = s_{m'm}$, e.g., $s_{12} = s_{21}$. The matrix $\mathscr{S}$ provides an estimate of the covariance matrix

$$\Sigma \equiv \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}.$$

Similarly, we can construct a matrix that contains sums of squares error and sums of cross products error. Write

$$e_{mm'} \equiv \sum_{hijk} \hat{\varepsilon}_{hijk,m} \hat{\varepsilon}_{hijk,m'}$$

where $e_{mm} = SSE_{mm}$ and

$$E \equiv \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}.$$

Obviously, $E = (dfE)\mathscr{S}$. For Box's fabric data,

$$E = \begin{bmatrix} 3225.00 & -80.50 & 1656.50 \\ -80.50 & 2405.50 & -112.00 \\ 1656.50 & -112.00 & 2662.50 \end{bmatrix}.$$

The diagonal elements of this matrix are the error sums of squares from Tables C.2, C.3, and C.4.

We can use similar methods for every line in the three analysis of variance tables. For example, each variable $m = 1,2,3$ has a sum of squares for $\mathbf{S}*\mathbf{P}$, which is computed as

$$SS(\mathbf{S}*\mathbf{P})_{mm} \equiv h(\mathbf{S}*\mathbf{P})_{mm} = 4\sum_{h=1}^{2}\sum_{j=1}^{3}\left(\bar{y}_{h\cdot j\cdot,m} - \bar{y}_{h\cdots,m} - \bar{y}_{\cdot\cdot j\cdot,m} + \bar{y}_{\cdots,m}\right)^2,$$

and where the multiplier of 4 is because the term $\bar{y}_{h\cdot j\cdot,m}$ has been averaged over 4 observations. (The nice algebraic formula only exists because the entire model is balanced.) We can also include cross products using $SS(\mathbf{S}*\mathbf{P})_{mm'} \equiv h(\mathbf{S}*\mathbf{P})_{mm'}$, where

$$h(\mathbf{S}*\mathbf{P})_{mm'} = 4\sum_{h=1}^{2}\sum_{j=1}^{3}\left(\bar{y}_{h\cdot j\cdot,m} - \bar{y}_{h\cdots,m} - \bar{y}_{\cdot\cdot j\cdot,m} + \bar{y}_{\cdots,m}\right)\left(\bar{y}_{h\cdot j\cdot,m'} - \bar{y}_{h\cdots,m'} - \bar{y}_{\cdot\cdot j\cdot,m'} + \bar{y}_{\cdots,m'}\right)$$

and create a matrix

$$H(\mathbf{S}*\mathbf{P}) \equiv \begin{bmatrix} h(\mathbf{S}*\mathbf{P})_{11} & h(\mathbf{S}*\mathbf{P})_{12} & h(\mathbf{S}*\mathbf{P})_{13} \\ h(\mathbf{S}*\mathbf{P})_{21} & h(\mathbf{S}*\mathbf{P})_{22} & h(\mathbf{S}*\mathbf{P})_{23} \\ h(\mathbf{S}*\mathbf{P})_{31} & h(\mathbf{S}*\mathbf{P})_{32} & h(\mathbf{S}*\mathbf{P})_{33} \end{bmatrix}.$$

For the fabric data

$$H(\mathbf{S}*\mathbf{P}) = \begin{bmatrix} 1186.0833 & -33.166667 & 526.79167 \\ -33.166667 & 44.333333 & -41.583333 \\ 526.79167 & -41.583333 & 250.58333 \end{bmatrix}.$$

Note that the diagonal elements of $H(\mathbf{S}*\mathbf{P})$ are the $\mathbf{S}*\mathbf{P}$ interaction sums of squares from Tables C.2, C.3, and C.4. Table C.5 contains the H matrices for all of the sources in the analysis of variance.

If you want to do just a standard one-way MANOVA for getting a multivariate test of whether the 12 groups are different, define the 3×3 matrix

$$H(Grps) = [h_{mm'}],$$

where

$$h_{mm'} = 2\sum_{h=1}^{2}\sum_{i=1}^{2}\sum_{j=1}^{3}\left(\bar{y}_{hij\cdot,m} - \bar{y}_{\cdots,m}\right)\left(\bar{y}_{hij\cdot,m'} - \bar{y}_{\cdots,m'}\right)$$

and the multiplier of 2 is because the term $\bar{y}_{hij\cdot,m}$ has been averaged over 2 observations. It turns out that

$$H(Grps) = H(\mathbf{S}) + H(\mathbf{F}) + H(\mathbf{P}) + H(\mathbf{S}*\mathbf{F}) + H(\mathbf{S}*\mathbf{P}) + H(\mathbf{F}*\mathbf{P}) + H(\mathbf{S}*\mathbf{F}*\mathbf{P}).$$

Table C.5 MANOVA statistics.

$$H(GRANDMEAN) = \begin{bmatrix} 940104.17 & 752281.25 & 602260.42 \\ 752281.25 & 601983.37 & 481935.13 \\ 602260.42 & 481935.13 & 385827.04 \end{bmatrix}$$

$$H(\mathbf{S}) = \begin{bmatrix} 26268.167 & 11479.917 & 6186.5833 \\ 11479.917 & 5017.0417 & 2703.7083 \\ 6186.5833 & 2703.7083 & 1457.0417 \end{bmatrix}$$

$$H(\mathbf{F}) = \begin{bmatrix} 6800.6667 & 21967.500 & 18129.500 \\ 21967.500 & 70959.375 & 58561.875 \\ 18129.500 & 58561.875 & 48330.375 \end{bmatrix}$$

$$H(\mathbf{P}) = \begin{bmatrix} 5967.5833 & 6818.2500 & 2646.9583 \\ 6818.2500 & 7969.0000 & 3223.7500 \\ 2646.9583 & 3223.7500 & 1396.5833 \end{bmatrix}$$

$$H(\mathbf{S}*\mathbf{F}) = \begin{bmatrix} 3952.6667 & 474.83333 & 38.500000 \\ 474.83333 & 57.041667 & 4.6250000 \\ 38.500000 & 4.6250000 & 0.37500000 \end{bmatrix}$$

$$H(\mathbf{S}*\mathbf{P}) = \begin{bmatrix} 1186.0833 & -33.166667 & 526.79167 \\ -33.166667 & 44.333333 & -41.583333 \\ 526.79167 & -41.583333 & 250.58333 \end{bmatrix}$$

$$H(\mathbf{F}*\mathbf{P}) = \begin{bmatrix} 3529.0833 & 4275.5000 & 2374.1250 \\ 4275.5000 & 6031.0000 & 2527.2500 \\ 2374.1250 & 2527.2500 & 1740.2500 \end{bmatrix}$$

$$H(\mathbf{S}*\mathbf{F}*\mathbf{P}) = \begin{bmatrix} 478.58333 & 4.4166667 & 119.62500 \\ 4.4166667 & 14.333333 & -57.750000 \\ 119.62500 & -57.750000 & 272.25000 \end{bmatrix}$$

$$E = \begin{bmatrix} 3225.00 & -80.50 & 1656.50 \\ -80.50 & 2405.50 & -112.00 \\ 1656.50 & -112.00 & 2662.50 \end{bmatrix}$$

In the standard (univariate) analysis of y_1 that was given in Table C.2, the test for $\mathbf{S}*\mathbf{P}$ interactions was based on

$$F = \frac{MS(\mathbf{S}*\mathbf{P})_{11}}{MSE_{11}} = \frac{SS(\mathbf{S}*\mathbf{P})_{11}}{SSE_{11}} \frac{1/df(\mathbf{S}*\mathbf{P})}{1/dfE} = \frac{h(\mathbf{S}*\mathbf{P})_{11}}{e_{11}} \frac{dfE}{df(\mathbf{S}*\mathbf{P})}.$$

The last two equalities are given to emphasize that the test depends on the $y_{hijk,1}$s only through $h(\mathbf{S}*\mathbf{P})_{11}\,[e_{11}]^{-1}$. Similarly, a multivariate test of $\mathbf{S}*\mathbf{P}$ is a function of the matrix

$$H(\mathbf{S}*\mathbf{P})E^{-1},$$

where E^{-1} is the matrix inverse of E. A major difference between the univariate and multivariate procedures is that there is no uniform agreement on how to use $H(\mathbf{S}*\mathbf{P})E^{-1}$ to construct a test statistic. The *generalized likelihood ratio* test statistic, also known as *Wilks' lambda*, is

$$\Lambda(\mathbf{S}*\mathbf{P}) \equiv \frac{1}{|I + H(\mathbf{S}*\mathbf{P})E^{-1}|}$$

where I indicates a 3×3 identity matrix and $|A|$ denotes the determinant of a matrix A. *Roy's maximum root statistic* is the maximum eigenvalue of $H(\mathbf{S}*\mathbf{P})E^{-1}$, say, $\phi_{max}(\mathbf{S}*\mathbf{P})$. On occasion, Roy's statistic is taken as

$$\theta_{max}(\mathbf{S}*\mathbf{P}) \equiv \frac{\phi_{max}(\mathbf{S}*\mathbf{P})}{1 + \phi_{max}(\mathbf{S}*\mathbf{P})}.$$

A third statistic is the *Lawley–Hotelling trace*,

$$T^2(\mathbf{S}*\mathbf{P}) \equiv dfE\ \mathrm{tr}\left[H(\mathbf{S}*\mathbf{P})E^{-1}\right],$$

and a final statistic is *Pillai's trace*,

$$V(\mathbf{S}*\mathbf{P}) \equiv \mathrm{tr}\left[H(\mathbf{S}*\mathbf{P})\left(E + H(\mathbf{S}*\mathbf{P})\right)^{-1}\right].$$

Similar test statistics Λ, ϕ, θ, T^2 and V can be constructed for all of the other main effects and interactions and also for the one-way MANOVA. It can be shown that for H terms with only one degree of freedom, these test statistics are equivalent to each other and to an F statistic. In such cases, we only present T^2 and the F value.

Table C.6 presents the test statistics for each term. When the F statistic is exactly correct, it is given in the table. In other cases, the table presents F statistic approximations. The approximations are commonly used and discussed; see, for example, Rao (1973, chapter 8) or *ALM*. Degrees of freedom for the F approximations and P values are also given. □

To complete a multivariate analysis, additional modeling is needed (or MANOVA contrasts for balanced data). The MANOVA assumptions also suggest some alternative residual analysis. We will not discuss either of these subjects. Moreover, our

Table C.6 Multivariate statistics.

Effect	Statistics	F	df	P
GRAND MEAN	$T^2 = 6836.64$	1899.07	3, 10	0.000
S	$T^2 = 137.92488$	38.31	3, 10	0.000
F	$T^2 = 612.96228$	170.27	3, 10	0.000
P	$\Lambda = 0.13732$	5.66	6, 20	0.001
	$T^2 = 65.31504$	8.16	6, 18	0.000
	$V = 0.97796$	3.51	6, 22	0.014
	$\phi_{max} = 5.28405$			
S $*$ **F**	$T^2 = 21.66648$	6.02	3, 10	0.013
S $*$ **P**	$\Lambda = 0.71068$	0.62	6, 20	0.712
	$T^2 = 4.76808$	0.60	6, 18	0.730
	$V = 0.29626$	0.64	6, 22	0.699
	$\phi_{max} = 0.37102$			
F $*$ **P**	$\Lambda = 0.17843$	4.56	6, 20	0.005
	$T^2 = 46.03092$	5.75	6, 18	0.002
	$V = 0.95870$	3.38	6, 22	0.016
	$\phi_{max} = 3.62383$			
S $*$ **F** $*$ **P**	$\Lambda = 0.75452$	0.50	6, 20	0.798
	$T^2 = 3.65820$	0.46	6, 18	0.831
	$V = 0.26095$	0.55	6, 22	0.765
	$\phi_{max} = 0.20472$			

analysis has exploited the balance in **S**, **F**, and **P** so that we have not needed to examine various sequences of models that would, in general, determine different H matrices for the effects. (Balance in, i.e. seeing all of, the "rotations" is required for the MANOVA).

Finally, a personal warning. One should not underestimate how much one can learn from simply doing the analyses for the individual variables. Personally, I would look thoroughly at each individual variable (number of rotations in our example) before worrying about what a multivariate analysis can add.

Appendix D
Neural Networks and Deep Learning as Nonparametric/Nonlinear Regression

Neural Networks (NNs) use nonlinear regression to solve nonparametric regression problems. *Deep Learning* is merely a reference to how complicated one makes the Neural Network/nonlinear regression function. Christensen (1996, Chapter 18 or 2015, Chapter 23) provides a brief introduction to fitting nonlinear regression models. Seber and Wild (1989, 2003) give a more expansive treatment of the area. In this appendix we focus on discussing the nonlinear regression models defined by NNs. NNs provide a special case of nonlinear regression and specialized software is available for them.

At the beginning of Chapter 7 we categorized various types of (univariate) regression problems into a hierarchy:

Conditional Expectation	Regression Type
$\mathrm{E}(y\|x) = x'\beta$	Linear Regression
$\mathrm{E}(y\|x) = f(x'\beta)$	Generalized Linear Models
$\mathrm{E}(y\|x) = f(x;\beta)$	Nonlinear Regression
$\mathrm{E}(y\|x) = f(x)$	Nonparametric Regression.

When it appears, β is a vector of unknown parameters. In both generalized linear models and nonlinear regression the function f is known but in nonparametric regression f is unknown.

In regression methods involving β, we choose estimates of β based on minimizing some loss function, most often squared error loss. For observations (y_i, x_i'), $i = 1, \ldots, n$, least squares nonlinear regression minimizes

$$SSE(\beta) \equiv \sum_{i=1}^{n} [y_i - f(x_i;\beta)]^2 .$$

In the NN literature, people often minimize $SSE(\beta)/2$. Sometimes a penalty function is added to the least squares criterion or other loss function. For least squares *linear* regression (and ridge regression) we can find explicit formulae for the estimates. When fitting nonlinear regression, like fitting generalized linear models, we need iterative computer methods to find estimates that minimize the (penal-

ized/regularized) loss function. You basically have to start with a guess for the best values of the parameters β and explore the loss function surface until you find a local minimum. Finding the global minimum can be extremely difficult unless the loss, as a function of the model parameters, is convex. If you have more parameters than data points I would be very surprised if there were not a great number (probably an uncountable number) of global minima to choose from and a far greater number of local minima, cf. Cooper (2021). (Although penalty functions can solve this issue, they do so at the price of being arbitrarily chosen.) One convenient way to shrink regression coefficients towards 0, or more generally to any value β_0, is to add artificial observations $\tilde{y}_k$ with covariates $\tilde{x}_k$ that satisfy $\tilde{y}_k \equiv f(\tilde{x}_k; \beta_0)$.

Most of Chapter 7 was devoted to using linear regression to solve nonparametric regression problems. Chapter 9 addressed classification problems using special cases of generalized linear models and contrasted them with support vector machines. The models of Chapter 9 can incorporate the nonparametric regression ideas of Chapter 7. Appendix C considered multivariate responses. We consider NNs for nonparametric nonlinear univariate regression, classification, and multivariate regression.

A common pedagogical application of NNs is identifying the correct digit in photographs of the numbers 0 through 9. Typically, the predictor variables are a constant (for an intercept term) along with the gray scale value for each pixel in some rectangle of pixels. The response is a 10 dimensional vector y that is used to identify the true digit, i.e., it has 0s everywhere except a 1 for the dimension associated with the correct digit. The hope is to feed the NN new pictures of digits and have them correctly classified. We will examine both univariate and multivariate NNs with an ultimate goal of developing a NN general enough to address this classification problem. Predicting the correct digit is inherently multivariate in that it involves associating a probability with each of the 10 possible outcomes (digits).

A philosophical difference between Machine Learning and traditional Statistics seems to be that statisticians have looked for "best" estimates and been interested in drawing conclusions about the parameters of their models whereas machine learners look at much more complicated models and look merely for useful answers (estimates/predictions). It seems that much of the work in NNs is advice on how to find any workable/good answer, cf. Ng (2018).

D.1 Univariate Neural Networks

In Chapters 1 and 2 we considered (univariate) linear models

$$y_i = x_i'\beta + \varepsilon_i \qquad \text{or} \qquad \mathrm{E}(y_i|x_i) = x_i'\beta.$$

(We sometimes write $\mathrm{E}(y_i) = x_i'\beta$ wherein conditioning on x_i is implicit.) In Chapters 3 and, especially, 7 we used spanning functions so that we could fit a nonparametric regression using linear models. In particular we fitted models like

$$y_i = \phi_i'\beta + \varepsilon_i \qquad \text{or} \qquad \mathrm{E}(y_i|x_i) = \phi_i'\beta,$$

where $\phi_i' = [\phi_0(x_i), \ldots, \phi_{s-1}(x_i)]$ was constructed from the spanning functions. Typically, the s-vectors ϕ_i have much higher dimension than the p-vectors x_i.

Neural networks construct a known nonlinear regression function $f(x_i;\beta)$ that depends on a large number of parameters in the hope that the function is sufficiently general that it can serve as a method for fitting nonparametric regressions. The function $f(x_i;\beta)$ is constructed in stages and deep learning is merely a reference to having many stages in the construction. The ability of NNs to serve as a model for nonparametric regression depends on the number of stages, the complexity at each stage, and on the choice of a known nonlinear differentiable *activation function* g that maps the real numbers onto the real numbers. (We will use the same g function for every stage but one could easily use different g functions at different stages.) The most commonly used g functions currently seem to be the logistic transform and the function we used to help define splines, called the *rectified linear unit (ReLU)* in the NN literature, i.e.,

$$g(u) = \frac{e^u}{1+e^u} = \frac{1}{1+e^{-u}} \qquad \text{or} \qquad g(x) = (x)_+ \equiv \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \le 0 \end{cases}.$$

(The nondifferentiability of $(x)_+$ at 0 does not seem to worry anyone.) The hyperbolic tangent (`tanh`) is another popular choice but there are a variety of common options. For any such activation function g we may apply it elementwise to any matrix. Thus, if $W = [w_{ij}]$ is a matrix,

$$g(W) \equiv [g(w_{ij})].$$

If we have another matrix $Q = [q_{ij}]$ that is the same size as W, on occasion we will want to multiply all of the individual elements in them, so we define the operation $\odot$ by

$$W \odot Q \equiv [w_{ij}q_{ij}].$$

This will be useful in writing derivatives.

Beginning with a p vector of predictor variables x, the essence of NNs is using g to define intermediate $r+1$-dimensional vectors of constructed predictor variables z_{k+1} and to use as a final model $\mathrm{E}(y|x) = z_D'\beta$. In particular, the constructed predictor variables are defined recursively as $z_{k+1}' \equiv \left[1, g\left(z_k'B_k\right)\right]$ where B_k is a matrix of unknown parameters, $k = 1, \ldots, D-1$, where D is the depth of the NN, and where $z_1 \equiv x$ with the understanding that the first component of x is typically 1. It follows that B_1 must be $p \times r$ but all remaining B_k are $(r+1) \times r$. In these problems typically $r << p$. (One could easily use different dimensions r_k at different stages.) The model involves many linear pieces but is highly nonlinear.

The simplest NNs have $D = 2$ and the simplest individual NN also has $r = 1$. With $r = 1$, B_1 is $p \times 1$ and β is 2×1. Denote $x' = (x_1, \ldots, x_p)$ remembering that typically $x_1 \equiv 1$. It is not hard to see that the nonlinear regression function is

$$f(x;\beta,B_1) = \beta_1 + \beta_2\, g\left(b_{11} + \sum_{j=2}^{p} b_{j1}x_j\right) = \beta_1 + \beta_2 g(x'B_1) = \left[1, g(x'B_1)\right]\beta.$$

With $D = 2$ and $r = 2$, now $B_1 = [b_{1ij}]$ is $p \times 2$ which we can write as two column vectors $B_1 = [B_{11}, B_{12}]$ and β is now 3×1. The nonlinear regression function becomes

$$\begin{aligned} f(x;\beta,B_1) &= \beta_1 + \beta_2\, g\left(b_{111} + \sum_{j=2}^{p} b_{1j1}x_j\right) + \beta_3\, g\left(b_{112} + \sum_{j=2}^{p} b_{1j2}x_j\right) \\ &= \beta_1 + \beta_2\, g(x'B_{11}) + \beta_3\, g(x'B_{12}) \\ &= \left[1, g(x'B_1)\right]\beta. \end{aligned}$$

Note that the terms

$$\beta_2\, g\left(b_{111} + \sum_{j=2}^{p} b_{1j1}x_j\right) \quad \text{and} \quad \beta_3\, g\left(b_{112} + \sum_{j=2}^{p} b_{1j2}x_j\right)$$

are completely interchangeable because the βs and bs are unknown parameters. Thus the parameters of this model are not identifiable. There can be no unique best estimates of them. *This problem occurs whenever* $r \geq 2$.

For $D = 3$ the NN model is

$$\begin{aligned} \mathrm{E}(y|x) = z_3'\beta = [1, g\left(z_2'B_2\right)]\beta = [1, g\left([1, g\left(z_1'B_1\right)]B_2\right)]\beta \\ \equiv [1, g\left([1, g\left(x'B_1\right)]B_2\right)]\beta \equiv f(x;\beta,B_2,B_1). \end{aligned}$$

More generally,

$$\begin{aligned} \mathrm{E}(y|x) &= z_D'\beta \\ &= [1, g\left(z_{D-1}'B_{D-1}\right)]\beta \\ &= [1, g\left([1, g\left(z_{D-2}'B_{D-2}\right)]B_{D-1}\right)]\beta \\ &= [1, g\left([1, g\left(\cdots [1, g\left(z_1'B_1\right)]B_{D-2}\right)]B_{D-1}\right)]\beta \\ &\equiv [1, g\left([1, g\left(\cdots [1, g\left(x'B_1\right)]B_{D-2}\right)]B_{D-1}\right)]\beta \\ &\equiv f(x;\beta,B_{D-1},\ldots,B_1). \end{aligned}$$

The unknown parameter matrices B_k and β altogether involve $s \equiv (p \times r) + (D-2)[(r+1) \times r] + (r+1)$ unknown parameters. Nonlinear regressions are typically fitted by least squares but obviously there is no reason one could not use another loss function or add a differentiable penalty function, cf. Chapter 8. (Minimizing a function as complicated as $\sum_i [y_i - f(x_i;\beta,B_{D-1},\ldots,B_1)]^2$ is difficult enough without adding nondifferentiability issues into it but some programs allow incorporation of the LASSO penalty.)

EXAMPLE D.1.1. *Battery Data.*
Consider again the battery data of Chapter 7. Figure D.1 diagrams the model

and provides least squares estimated parameters obtained from one fit of R's `neuralnet` program specifying $D = 3$, $r = 2$, and a logistic activation function. The parameter estimates can be read off the figure and are,

$$\hat{B}_1 = \begin{bmatrix} -6.407815 & -4.337426 \\ 7.293236 & 13.458036 \end{bmatrix}, \quad \hat{B}_2 = \begin{bmatrix} -2.1456266 & -0.5137351 \\ 0.6429814 & -8.3132015 \\ 2.9668757 & 2.9781474 \end{bmatrix},$$

$$\hat{\beta} = \begin{bmatrix} 4.061124 \\ 6.843595 \\ 7.171289 \end{bmatrix}.$$

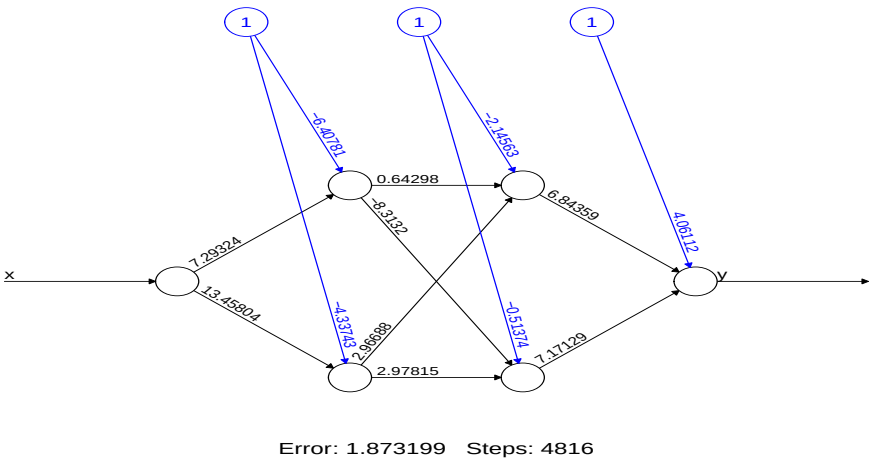

Fig. D.1 Battery data: $D = 3$, $r = 2$, g-logistic neural net. Diagram with one set of estimated parameters.

Figure D.2 gives the data and the NN fitted values along with the squared correlation between them. Comparing these results with Chapter 7, this model fits $s = 13$ parameters, yet gives an R^2 that is less than that of the 5 parameter (4th degree) polynomial model, the cosine model with $s = 7$ parameters, and the $s = 7$ cubic spline model having 4 interior knots.

These figures are from just one `neuralnet` fit to this model. Since $r \geq 2$, there are other fits, some of which are not as good. It is relatively simple to fit this model in `neuralnet`. Understanding how and why it gives these fitted values is not simple. There are many computational issues to be explored with NNs. □

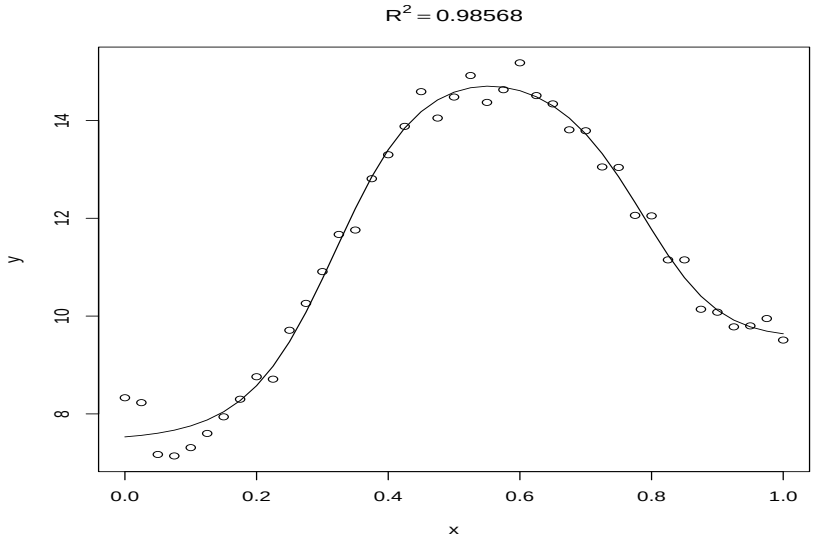

Fig. D.2 One `neuralnet` fit to the battery data: $D = 3$, $r = 2$, g-logistic.

EXERCISE D.1. Create a NN diagram similar to Figure D.1 for the following estimates.

$$\hat{B}_1 = \begin{bmatrix} -4.337426 & -6.407815 \\ 13.458036 & 7.293236 \end{bmatrix}, \quad \hat{B}_2 = \begin{bmatrix} -0.5137351 & -2.1456266 \\ 2.9781474 & 2.9668757 \\ -8.3132015 & 0.6429814 \end{bmatrix},$$

$$\hat{\beta} = \begin{bmatrix} 4.061124 \\ 7.171289 \\ 6.843595 \end{bmatrix}.$$

Show that these estimates give precisely the same fitted NN as in Example D.1.1. How are the numbers rearranged from the earlier estimates?

It is interesting to note that if you were to choose a linear activation function g, all of the NN structure would be a waste of time because such a NN is equivalent to fitting a linear model $\mathrm{E}(y|x) = x'\gamma$. In particular, if g is the identity function and you eliminate the intercept terms, the NN is quite literally $\mathrm{E}(y|x) = x'B_1B_2\cdots B_{D-1}\beta$ in which almost all of the parameters are completely unidentifiable and which is clearly equivalent to just fitting $\mathrm{E}(y|x) = x'\gamma$. (Retaining the intercept terms makes the expression far more complicated but amounts to the same thing.) This argument also highlights the importance of picking a nonlinear function g to define the NN.

It will be convenient to combine all of the NN parameters $\beta, B_{D-1}, \ldots, B_1$ into a single vector of parameters, say, $\tilde{\beta}$. (For those familiar with the Vec operator

that stacks the columns of a matrix, $\tilde{\beta} \equiv [\beta', \text{Vec}(B_{D-1})', \ldots, \text{Vec}(B_1)]'$.) We write $SSE(\tilde{\beta})$ for the criterion being minimized.

There may be advantages to standardizing the predictor variables or even replacing them with their principal components. These may help to avoid dealing with places where the derivative of the criterion function is close to zero or may help avoiding criterion functions that are similar to ellipsoids with radically different sized axes for which steepest descent tends not to work well.

D.1.1 Relation to Spanning Functions

We have seen that for $D = 2$ and $B_1 \equiv [B_{11}, \ldots, B_{1r}]$, the NN defines

$$f(x; \beta, B_1) = \beta_1 + \beta_2 g(x'B_{11}) + \cdots + \beta_{r+1} g(x'B_{1r}).$$

In the NN B_1 is a matrix of parameters but imagine now that B_1 is known. With B_1 known, the NN can be viewed as an application of the spanning function approach to nonparametric regression discussed in Chapter 7 by setting

$$\phi_0(x) \equiv 1, \qquad \phi_j(x) \equiv g(x'B_{1j}), \quad j = 1, \ldots, r.$$

For the spanning function approach to be effective, we typically need $r+1$ to be substantially larger than p. In NNs we typically have $r+1$ much smaller than p but we can do that because we are using the data to determine B_1, so we are using the data to determine the best choices of the spanning functions.

In this context it is immediately clear that, when $r > 1$, the parameters in B_1 are not identifiable because if you permute the columns of B_1 you get the same collection of spanning functions and therefore get the same fit to the data. Thus B_1 cannot possibly be uniquely determined by the (distribution of the) data.

When $D > 2$, we are allowing more flexibility in letting the data determine better spanning functions. It is also quite clear that none of the B_ks can be identifiable because any permutation of the r columns, if applied to *all* of the B_ks, will always give the same results. (As illustrated in Exercise D.1, if you permute the r columns of B_k you also need to apply that permutation to the last r rows of B_{k+1}.)

While the permutation argument clearly demonstrates that the parameters are not identifiable, permutations are not the only source for nonidentifiability in NNs.

D.2 Nonlinear Regression

As a statistics problem, nonlinear regression has been around a long time. For a nonlinear regression $f(x; \tilde{\beta})$ with independent, mean zero, homoskedastic, normally distributed errors, the least squares estimates of $\tilde{\beta}$ are maximum likelihood estimates. Traditional nonlinear regression is based on using Taylor's Theorem to get an ap-

proximate linear model. The calculus of Taylor's Theorem tells us that for $\tilde{\beta}$ values close to a known value $\tilde{\beta}_0$,

$$f(x;\tilde{\beta}) \doteq f(x;\tilde{\beta}_0) + [\mathbf{d}_{\tilde{\beta}} f(x;\tilde{\beta}_0)](\tilde{\beta} - \tilde{\beta}_0).$$

(Derivatives are discussed in Appendix A.9.) Taylor's Theorem suggests fitting the approximating model,

$$y_i \doteq f(x_i;\tilde{\beta}_0) + [\mathbf{d}_{\tilde{\beta}} f(x_i;\tilde{\beta}_0)](\tilde{\beta} - \tilde{\beta}_0) + \varepsilon_i, \quad i = 1,\ldots,n.$$

Because $\tilde{\beta}_0$ is known, this is actually a standard linear model that incorporates a pair of offsets. First, define our predictor variables as

$$w_i' \equiv \mathbf{d}_{\tilde{\beta}} f(x_i;\tilde{\beta}_0).$$

Now we can remove the known offset terms to the left-hand side and write the standard linear model

$$y_i - f(x_i;\tilde{\beta}_0) + w_i'\tilde{\beta}_0 \doteq w_i'\tilde{\beta} + \varepsilon_i, \quad i = 1,\ldots,n.$$

If $\tilde{\beta}_0$ minimizes $SSE(\tilde{\beta})$, this approximate linear model can be used in pretty much the standard way to obtain predictions and inferences about the parameters of the nonlinear model.

A version of this linearization procedure is also how we traditionally find the least squares estimates. Writing $\delta \equiv (\tilde{\beta} - \tilde{\beta}_0)$, an equivalent approximating model can we written

$$y_i - f(x_i;\tilde{\beta}_0) \doteq w_i'\delta + \varepsilon_i, \quad i = 1,\ldots,n. \tag{1}$$

In matrix terms we can write this as

$$Y - F(\tilde{\beta}_0) = W\delta + e.$$

Here W actually depends on $\tilde{\beta}_0$, so actually

$$Y - F(\tilde{\beta}_0) = W(\tilde{\beta}_0)\delta + e.$$

Estimating δ is the key to finding the least squares estimates. As justified in the earlier references on nonlinear regression, the least squares estimate of δ, i.e., $\hat{\delta} = (W'W)^{-1}W'[Y - F(\tilde{\beta}_0)]$, can be used to move $\tilde{\beta}_0$ closer to the least squares estimate via $\tilde{\beta}_1 \equiv \tilde{\beta}_0 + \hat{\delta}$. Repeating this process with $\tilde{\beta}_1$ replacing $\tilde{\beta}_0$, leads to values $\tilde{\beta}_2, \tilde{\beta}_3, \tilde{\beta}_4, \ldots$ that (we hope) will converge to the least squares estimate. This is the *Gauss-Newton algorithm* for finding the least squares estimates. If, instead of least squares estimation, we use ridge regression to estimate δ, we get $\tilde{\beta}_1 = \tilde{\beta}_0 + \hat{\delta}_R$ where $\hat{\delta}_R = (W'W + \lambda I)^{-1}W'[Y - F(\tilde{\beta}_0)]$. This method is known as *Marquart's compromise algorithm*. In what sense is this a compromise? The method of *gradient/steepest descent* uses $\hat{\delta}_S = (\lambda I)^{-1}W'[Y - F(\tilde{\beta}_0)] = (1/\lambda)W'[Y - F(\tilde{\beta}_0)]$, so Marquart's method is a combination of the other two. What these methods have in

common is that to get $\hat{\delta} = 0$ requires $W(\tilde{\beta}_0)'[Y - F(\tilde{\beta}_0)] = 0$, which makes $\tilde{\beta}_0$ a critical point of $SSE(\beta)$. A variety of other least squares algorithms have been proposed and are available in various software products.

If Gauss-Newton works, it tends to work very well. (For example, with a linear model, Gauss-Newton always gets the answer in just one step.) If W displays collinearity problems, Gauss-Newton frequently works poorly. We will see that even in a very simple NN for the battery data, one without identifiability issues, W has severe collinearity problems. In addition, NNs often have s, the number of parameters, very large and $W'W$ is an $s \times s$ matrix, so even without collinearity issues, it can be difficult to invert the matrix. Not surprisingly, NN programs tend to focus on the method of gradient/steepest descent, thus avoiding large matrix inversions.

How well a NN model works depends on the true structure of $\mathrm{E}(y|x)$, the choices of g, r, and D, and the fitting method. It reminds me of reproducing kernel regression in which success depends on how the choice of kernel meshes with the choice of penalty function and the true structure. It is impossible to know ahead of time what choices will work well.

EXAMPLE D.2.1. I fitted the simplest NN model to the battery data, a logistic $D = 2$, $r = 1$. This model does not fit the data very well, but that is irrelevant to the issues discussed here. The model has $r = 1$, so it has no identifiability issues. It was fitted using R's NN programs `neuralnet` and `nnet`, and R's nonlinear least squares program `nls`. The NN programs were run four times. Running `neuralnet`, based on random starting values $\tilde{\beta}_0$, gave estimates

$$\begin{aligned}
\tilde{\beta} &= (7.703679, 4.993587, -7.598368, 28.173712)' \\
\tilde{\beta} &= (7.703560, 4.993619, -7.599077, 28.177550)' \\
\tilde{\beta} &= (7.701728, 4.995634, -7.582689, 28.122157)' \\
\tilde{\beta} &= (7.702761, 4.994409, -7.594703, 28.163069)',
\end{aligned}$$

so it looks like `neuralnet` consistently converges to the unique least squares estimates. Running `nnet`, also based on random starting values, gave

$$\begin{aligned}
\tilde{\beta} &= (7.711252, 4.985046, -7.684437, 28.463444)' \\
\tilde{\beta} &= (7.711415, 4.984930, -7.687263, 28.472109)' \\
\tilde{\beta} &= (7.711415, 4.984809, -7.685779, 28.468946)' \\
\tilde{\beta} &= (7.711285, 4.985015, -7.684596, 28.463871)',
\end{aligned}$$

so it also looks like `nnet` consistently converges to the unique least squares estimates. Unfortunately, they are not quite the same values as those given by `neuralnet`. Finally, `nls`, for all the starting values I tried, converged to

$$\tilde{\beta} = (7.711, 4.985, -7.684, 28.463)',$$

which agrees well with the `nnet` results. □

EXAMPLE D.2.2. I attempted to fit the second simplest well-defined NN model to the battery data, a logistic $D = 3$, $r = 1$. (Again, this simple model seems to be incapable of actually fitting the data well, but that is again irrelevant.) Because $D > 2$, `nnet` will not fit this model. We will see later that `neuralnet` is incapable of fitting it in any consistant way. Alas, `nls` is also not capable of fitting it using the Gauss-Newton method. This seems to be due to collinearity problems. To illustrate this, I obtained an estimate from neuralnet and used it as a starting value $\tilde{\beta}_0$ for Gauss-Newton. In particular I computed the derivative matrix $W(\tilde{\beta}_0)$. Due to the definition of NNs, the first column of $W(\tilde{\beta}_0)$ is a column of 1s. After that, there is one column for β_2, two columns for B_2, and two columns for B_1. I treated all the columns after the first as standard predictor variables and looked at the eigenvalues of their covariance matrix to assess collinearity.

My starting value was

$$\tilde{\beta}_0 = (5.2801397, 7.8749274, -0.8069346, 3.5837052, -7.1122252, 24.2575694).$$

The eigenvalues for the covariance matrix of all the predictors were 0.5073504, 0.1494210, 0.0008043, 0.00003363, and 0.000001177. With such small eigenvalues, it is not surprising that using Gauss-Newton and fitting linear models is problematic. □

In traditional applications of nonlinear regression, one could frequently find the derivative of $f(x;\tilde{\beta})$ with respect to $\tilde{\beta}$ quite easily. NNs involve fitting a much more complicated function than in traditional statistical applications. Finding derivatives in NN's version of nonlinear regression, which involves recursive use of g, involves multiple applications of the chain rule. This does not appear difficult but seems ripe for specialized software. Moreover, like the spanning function approach to nonparametric regression, NNs seem rife with computational issues associated with the very large number of parameters. (Rather than finding the derivatives, modern software often approximates them numerically.)

We now find the derivatives of the nonlinear regression function for NNs with $D = 2$. Definitions and some additional comments appear in Appendix A.9. Denote

$$\dot{g}(u) \equiv \mathbf{d}_u g(u).$$

When g is the logistic transform,

$$\dot{g}(u) = e^u/(1+e^u)^2.$$

For $D = 2$ and $r = 1$, with $f(x;\beta,B_1) = \beta_1 + \beta_2 g\left(b_{11} + \sum_{j=2}^{p} b_{j1}x_j\right)$,

$$\mathbf{d}_\beta f(x;\beta,B_1) = \left[1, g(x'B_1)\right] = z_2',$$

$$\mathbf{d}_{b_{11}} f(x;\beta,B_1) = \beta_2\, \dot{g}\left(b_{11} + \sum_{j=2}^{p} b_{j1}x_j\right),$$

and for $j = 2, \ldots, p$

$$\mathbf{d}_{b_{1j}} f(x;\beta,B_1) = \beta_2\, \dot{g}\left(b_{11} + \sum_{j=2}^{p} b_{j1} x_j \right) x_j.$$

In vector form

$$\mathbf{d}_{B_1} f(x;\beta,B_1) = \beta_2\, \dot{g}(x'B_1)x'$$

For $D = 2$ and $r = 2$ with $f(x;\beta,B_1) = \beta_1 + \beta_2 g(x'B_{11}) + \beta_3 g(x'B_{12})$,

$$\mathbf{d}_{\beta} f(x;\beta,B_1) = [1, g(x'B_{11}), g(x'B_{12})] = [1, g(x'B_1)] = z_2',$$

$$\mathbf{d}_{B_{11}} f(x;\beta,B_1) = \beta_2\, \dot{g}(x'B_{11})x',$$
$$\mathbf{d}_{B_{12}} f(x;\beta,B_1) = \beta_3\, \dot{g}(x'B_{12})x'.$$

In matrix form,

$$\mathbf{d}_{\mathrm{Vec}(B_1)} f(x;\beta,B_1) = \left[\beta_2\, \dot{g}(x'B_{11})x', \beta_3\, \dot{g}(x'B_{12})x'\right].$$

For $D = 2$ and any r we should get

$$\mathbf{d}_{\beta} f(x;\beta,B_1) = [1, g(x'B_1)] = z_2'$$

and with $\beta' \equiv [\beta_1, \beta_*']$

$$\mathbf{d}_{\mathrm{Vec}(B_1)} f(x;\beta,B_1) = \beta_*' \odot \dot{g}(x'B_1) \otimes x'.$$

In addition, the derivatives for the battery data with logistic activation, $D = 3$, and $r = 1$ are programmed in the accompanying manual.

D.2.1 Back Propagation

In standard nonlinear regression, with a model $f(x;\tilde{\beta})$, Gauss-Newton, gradient/steepest Descent, and Marquart's Compromise all seek to find a sequence of $\tilde{\beta}$ values $\tilde{\beta}_0, \tilde{\beta}_1, \tilde{\beta}_2, \ldots$ that converge to some value and when the sequence converges that value must be the least squares estimate. Each method involves one computation to move from $\tilde{\beta}_k$ to $\tilde{\beta}_{k+1}$.

In NNs, $\tilde{\beta}$ is typically a large vector, but one that is naturally partitioned into smaller parts based on $\beta, B_{D-1}, ..., B_1$. It is convenient (and perhaps necessary) to break up the large computation from $\tilde{\beta}_k$ to $\tilde{\beta}_{k+1}$ into smaller more manageable parts. That is what back propagation does.

We always start with an initial guess for all the parameters, $\tilde{\beta}_0$ or equivalently $\beta^0, B^0_{D-1}, ..., B^0_1$. In back propagation we start by updating β^0 into β^1 leaving all the other B^0_j's fixed. This gives us $\beta^1, B^0_{D-1}, ..., B^0_1$. Then with the new β^1, and all the

other B_j^0s fixed, update B_{D-1} giving $\beta^1, B_{D-1}^1, B_{D-2}^0, ..., B_1^0$. Work your way through until you have $\beta^1, B_{D-1}^1, ..., B_1^1$ and start all over again.

We now get more explicit. For simplicity, consider a NN with $D = 3$, so

$$\tilde{\beta}_k' = (\beta^{k\prime}, \text{Vec}(B_2^k)', \text{Vec}(B_1^k)').$$

To simplify notation, rewrite this as

$$\tilde{\beta}_k' \equiv (\tilde{\beta}_{k3}', \tilde{\beta}_{k2}', \tilde{\beta}_{k1}'),$$

so a starting value is

$$\tilde{\beta}_0' = (\tilde{\beta}_{03}', \tilde{\beta}_{02}', \tilde{\beta}_{01}').$$

Also write

$$f(x; \tilde{\beta}_k) \equiv f(x; \tilde{\beta}_{k3}, \tilde{\beta}_{k2}, \tilde{\beta}_{k1}).$$

Instead of fitting the model (D.1) all at once, to obtain $\tilde{\beta}_1$ we fit $D = 3$ models. First, fit

$$y_i - f(x_i; \tilde{\beta}_{03}, \tilde{\beta}_{02}, \tilde{\beta}_{01}) \doteq [\mathbf{d}_{\tilde{\beta}_3} f(x_i; \tilde{\beta}_{03}, \tilde{\beta}_{02}, \tilde{\beta}_{01})]\delta_3 + \varepsilon_i, \quad i = 1, \ldots, n.$$

to obtain $\tilde{\beta}_{13} = \tilde{\beta}_{03} + \hat{\delta}_3$. Then, fit

$$y_i - f(x_i; \tilde{\beta}_{13}, \tilde{\beta}_{02}, \tilde{\beta}_{01}) \doteq [\mathbf{d}_{\tilde{\beta}_2} f(x_i; \tilde{\beta}_{13}, \tilde{\beta}_{02}, \tilde{\beta}_{01})]\delta_2 + \varepsilon_i, \quad i = 1, \ldots, n.$$

to obtain $\tilde{\beta}_{12} = \tilde{\beta}_{02} + \hat{\delta}_2$. Finally, fit

$$y_i - f(x_i; \tilde{\beta}_{13}, \tilde{\beta}_{12}, \tilde{\beta}_{01}) \doteq [\mathbf{d}_{\tilde{\beta}_1} f(x_i; \tilde{\beta}_{13}, \tilde{\beta}_{12}, \tilde{\beta}_{01})]\delta_1 + \varepsilon_i, \quad i = 1, \ldots, n.$$

to obtain $\tilde{\beta}_{11} = \tilde{\beta}_{01} + \hat{\delta}_1$. Having obtained $\tilde{\beta}_1 = (\tilde{\beta}_{13}', \tilde{\beta}_{12}', \tilde{\beta}_{11}')'$, repeat the process to get $\tilde{\beta}_2$ and so on until they converge (or convince you that they will not converge).

With NNs, $\hat{\delta}_k$ is generally obtained by gradient/steepest descent.

EXAMPLE D.2.2 CONTINUED. For the battery data with $D = 3$ and $r = 1$, Gauss-Newton continues to have collinearity problems even with back propagation. Rather than updating the predictor variables in turn, I just looked at the predictors associated with the original starting values, i.e., $\mathbf{d}_{\tilde{\beta}_k} f(x_i; \tilde{\beta}_{03}, \tilde{\beta}_{02}, \tilde{\beta}_{01})$. In other words, I partitioned the $W(\tilde{\beta}_0)$ matrix into three sets of two columns. Fitting a linear model to get δ_3 will not be a problem because the model involves just an intercept and one predictor. For the predictors associated with just B_2 and just B_1, I found the covariance matrices and their eigenvalues. For the B_2 predictors, the eigenvalues are 0.3845 and 0.0291, which are not a problem. For the B_1 predictors, the covariance eigenvalues are 0.1760 and 0.0002779, which could be a problem. Remember that collinearity problems can arise at any step of the Gauss-Newton iterative procedure and cause it to fail. (Note: The linear models to be fitted for updating B_2 and B_1 are regressions through the origin but in this case the ratio of eigenvectors for the

covariance matrices was roughly comparable to the ratio of eigenvectors for the uncentered matrices.) □

D.3 Computational Issues

EXAMPLE D.3.1. Earlier we attempted to fit the second simplest well-defined NN model to the battery data, a logistic $D = 3$, $r = 1$. With $r = 1$, this seems to be an identifiable model, so there should be a unique set of least squares estimates. Yet in looking at the `neuralnet` output, I was surprised to find a great deal of variability in the reported estimates. So much so that I computed 1000 sets of $\tilde{\beta}$ estimates, and then computed their mean vector and covariance matrix. The mean vector is not very interesting,

$$(5.7504, 6.9914, -0.5063, 17.9064, -4.7870, 11.7851)'.$$

At best this mean could be used as an improved estimate for $\tilde{\beta}$. The covariance matrix was

$$\begin{bmatrix} 0.5754 & -0.4894 & -0.7770 & 1.764 & -0.2247 & -0.3047 \\ -0.4894 & 2.3580 & -0.3858 & -3.043 & -1.3709 & 4.4923 \\ -0.7770 & -0.3858 & 4.9928 & -10.717 & 5.7829 & -6.3938 \\ 1.7640 & -3.0432 & -10.7167 & 36.457 & -12.4099 & 7.1190 \\ -0.2247 & -1.3709 & 5.7829 & -12.410 & 8.4314 & -10.5457 \\ -0.3047 & 4.4923 & -6.3938 & 7.119 & -10.5457 & 20.6283 \end{bmatrix}.$$

I find the variances to be stunningly large for a process that purports to be converging to unique estimates. There are even substantial differences in these numbers when repeating the computation with a different 1000 estimates. □

For $D = 2$, both `nnet` and `neuralnet` purport to give the least squares estimates. When $r > 1$, we know the estimates are not identifiable, so the least squares estimates are not unique. Moreover, there may be different local minima to the $SSE(\tilde{\beta})$ surface that the different programs may find, so there is no reason to expect `nnet` to give the same results as `neuralnet`. However, if we take the least squares estimates from `neuralnet` and feed them as starting values into `nnet`, we would expect `nnet` to recognize them quickly as least squares estimates. The same should happen when feeding `nnet` estimates into `neuralnet`. In Example D.2.1 we saw that even for the simplest NN model, $D = 2$, $r = 1$, where the estimates are identifiable, `neuralnet` and `nnet` could not agree on the least squares estimates for the battery data. It will not be too surprising then, that the two NN programs do not always agree on what is a least squares estimate in more complicated models.

EXAMPLE D.3.2. For the battery data, there is code in the accompanying manual for $D = 2$ and any r feeding `neuralnet` estimates into `nnet` and vice versa. Only a few results are presented here.

Figure D.3 gives the battery data and 4 `neuralnet` fits of the data on the left and then 4 `nnet` fits on the right all using $D = 2$ and $r = 5$. The 4 `neuralnet` fits come from using different random starting values and they differ radically in the quality of their fitted values in that one of the four does a horrible job of fitting the data for large x values. The difference in fits should be due to using different starting values, leading to finding different local minima of the $SSE(\tilde{\beta})$ surface, and thus to different parameter estimates. Figure D.3 by no means exhausts all the possible fits that can be obtained from either `neuralnet` or `nnet`.

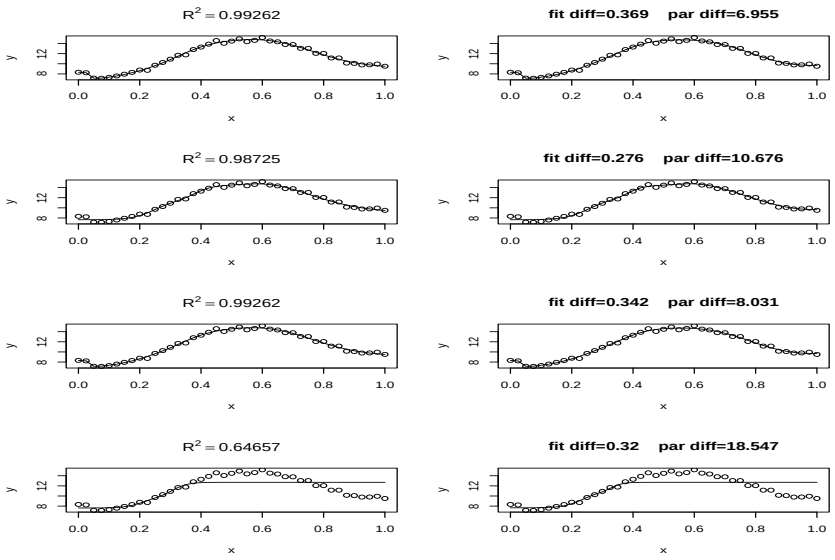

Fig. D.3 Battery data neural net fits: $D = 2$, $r = 5$, neuralnet feeding nnet.

Above each plot on the left of Figure D.3 is R^2, the squared correlation between the y_i values and the corresponding `neuralnet` fitted values. There are three distinct values and one is much lower than the other two. Even among well-fitting NNs, there typically are slight differences among the fitted values that cause slight differences in the R^2 values.

Above each plot on the right of Figure D.3 are two numbers. The first number is the square root of the sum of squared differences between the `neuralnet` fitted values and the `nnet` fitted values. This number should be 0 if the two programs have found the same local minimum for the $SSE(\tilde{\beta})$ surface. With $r > 1$ we expect an infinite number of $\tilde{\beta}$ values that give the same local minima. The second number above each plot on the right is the square root of the sum of squared differences

between the `neuralnet` parameter estimates and the `nnet` parameter estimates. This should be 0 if the second program has recognized the first program's estimates as being values that minimize $SSE(\tilde{\beta})$. Sometimes the values are small and sometimes they are not. In my experience it seems that `neuralnet` is more likely to accept `nnet` estimates as correct than vice versa. In the figure the fitted value differences are all distinct but are all reasonably small. The parameter differences vary a great deal.

The first and third `neuralnet` plots *might* have resulted from finding the same local minimum to $SSE(\tilde{\beta})$, since they have the same R^2 to five digits. They certainly did not have the same parameter estimates because if they did, the numbers above the corresponding `nnet` plots would have to be the same. (Identical starting values should lead to identical results.)

The differences in parameter estimates might be a result of the step sizes used when searching the $SSE(\tilde{\beta})$ surface for local minima. Due to nonidentifiability when $r > 1$, the $SSE(\tilde{\beta})$ surface should have troughs of values that give the same local minimum. If so, you can move around the trough, changing the estimates, but not changing the fitted values very much.

I have created a lot of these plots and in my experience, as r gets larger, you are less likely to get NN fitted values that fit the battery data poorly. (For $r = 5$, I chose a figure with one bad fit because that seemed about the correct proportion.) For $r = 1, 2, 3$, I have rarely seen a good fit to the battery data. □

D.4 Classification

If we are doing a classification problem with $y = 0, 1$, rather than a standard regression problem, then $\mathrm{E}(y|x)$ needs to be a probability between 0 and 1, so we want to specify a cumulative distribution function F and use

$$\begin{aligned}
\mathrm{E}(y|x) &= F(z_D'\beta) \\
&= F\big([1, g(z_{D-1}'B_{D-1})]\beta\big) \\
&= F\big([1, g\big([1, g(z_{D-2}'B_{D-2})]B_{D-1}\big)]\beta\big) \\
&= F\big([1, g\big([1, g\big(\cdots[1, g(z_1'B_1)]B_{D-2}\big)]B_{D-1}\big)]\beta\big) \\
&= F\big([1, g\big([1, g\big(\cdots[1, g(x_i'B_1)]B_{D-2}\big)]B_{D-1}\big)]\beta\big) \\
&\equiv f(x; \beta, B_{D-1}, \ldots, B_1).
\end{aligned}$$

One the most popular choices for F is the logistic transform which is also one of the most popular choices for g.

Often it makes sense to fit a classification NN by maximum likelihood with the y_is distributed as independent Bernoulli variables having probability $\mathrm{E}(y_i|x_i)$, but my impression is that classification NNs are often fitted using least squares.

Classification NNs are regression procedures, similar to logistic regression, so they require modification if being used on discrimination data as discussed in Chapter 10.

D.5 Generalized Weights

Weights is a term often used in the NN literature for the parameters other than the intercepts involved at each stage. The intercepts are often referred to as *biases*. For a standard nonlinear regression

$$\mathrm{E}(y|x_i) = f(x_i;\beta),$$

we fit linear models based on the Taylor's expansion of β around some fixed points β_0. We need to keep changing the fixed point β_0 until it becomes the least squares estimate $\hat{\beta}$.

Generalized weights involve Taylor's expansions of $f(x;\hat{\beta})$ around the fixed points x_i, $i = 1,\ldots,n$. Taylor's theorem says that for x in a neighborhood of x_i, an approximating linear model holds,

$$\mathrm{E}(y|x) \doteq f(x_i;\hat{\beta}) + [\mathbf{d}_x f(x_i;\hat{\beta})](x - x_i) \equiv \gamma_0 + x'\gamma_*.$$

Here the regression coefficients are $\gamma_* \equiv [\mathbf{d}_x f(x_i;\hat{\beta})]'$ and the intercept is $\gamma_0 = f(x_i;\hat{\beta}) - [\mathbf{d}_x f(x_i;\hat{\beta})]x_i$. The regression coefficient vector $[\mathbf{d}_x f(x_i;\hat{\beta})]'$ is often called the vector of *generalized weights* ($\tilde{w}_i$) at x_i.

For a binomial regression they do it a little differently. For a cdf F, a typical model is

$$\mathrm{E}(y|x_i) = f(x_i;\beta) \equiv F[h(x_i;\beta)],$$

where h is defined as

$$h(x_i;\beta) \equiv F^{-1}[f((x_i;\beta)].$$

Instead of getting a linear model approximation to the nonlinear regression, we do a generalized linear model approximation so that

$$f(x_i;\beta) = F[h(x_i;\beta)] \doteq F(\gamma_0 + x'\gamma)$$

The Taylor's approximation is applied to h rather than f, so the generalized weights are the regression coefficients of the approximating "linear predictor."

D.6 Multivariate Neural Networks

In Appendix C.1 we discussed multivariate linear models $Y = XB + e$ wherein Y is an $n \times q$ response matrix on n independent individuals with q (presumably) corre-

lated measurements per individual. B is a $p \times q$ matrix of unknown parameters. The model can also be written on an individual basis for $i = 1, \ldots, n$ as

$$y_i' = x_i' B + \varepsilon_i' \qquad \text{or} \qquad \mathrm{E}(y_i'|x_i) = x_i' B.$$

It is a trivial generalization to use spanning functions $\phi_j(\cdot)$ to turn this into a multivariate nonparametric regression model $Y = \Phi B + e$ which can also be written as

$$y_i' = \phi_i' B + \varepsilon_i' \qquad \text{or} \qquad \mathrm{E}(y_i'|x_i) = \phi_i' B.$$

Here again $\phi_i' \equiv [\phi_0(x_i), \ldots, \phi_{s-1}(x_i)]'$.

The multivariate generalization of the univariate NN is straightforward, with the vector β being replaced by a matrix, say B_D, having q columns,

$$\begin{aligned}
\mathrm{E}(y'|x) &= z_D' B_D \\
&= [1, g(z_{D-1}' B_{D-1})] B_D \\
&= [1, g([1, g(z_{D-2}' B_{D-2})] B_{D-1})] B_D \\
&= [1, g([1, g(\cdots [1, g(z_1' B_1)] B_{D-2})] B_{D-1})] B_D \\
&= [1, g([1, g(\cdots [1, g(x_i' B_1)] B_{D-2})] B_{D-1})] B_D \\
&\equiv f(x; B_D, B_{D-1}, \ldots, B_1).
\end{aligned}$$

Like multivariate linear models these are often fitted using least squares although the justification for using least squares seems less clear than it is in either multivariate linear models or univariate nonlinear regression. In both of those cases, least squares is known to provide maximum likelihood estimates for individuals having (multivariate) normal errors.

If y is a vector of 0-1 classification responses rather than a standard regression problem, the components of $\mathrm{E}(y|x)$ need to be probabilities between 0 and 1, so we again want to specify a (univariate) cdf F and use

$$\begin{aligned}
\mathrm{E}(y'|x) &= F(z_D' B_D) \\
&= F([1, g(z_{D-1}' B_{D-1})] B_D) \\
&= F([1, g([1, g(z_{D-2}' B_{D-2})] B_{D-1})] B_D) \\
&= F([1, g([1, g(\cdots [1, g(z_1' B_1)] B_{D-2})] B_{D-1})] B_D) \\
&\equiv F([1, g([1, g(\cdots [1, g(x_i' B_1)] B_{D-2})] B_{D-1})] B_D) \\
&\equiv f(x; B_D, B_{D-1}, \ldots, B_1).
\end{aligned}$$

Again, one the most popular choices for F is the logistic transform which is also one of the most popular choices for g. This form of a NN is one that I have seen (inappropriately) used for classifying photographs of digits.

Again, these classification NNs seem to get fitted often by least squares (rather than by maximum likelihood). As a classification procedure least squares seems rather crude.

This NN seems most applicable for analyzing, say, a collection of 0-1 tests on an individual (e.g. high cholesterol, high blood pressure, covid-19 positive, HIV positive). It seems less appropriate for identifying an individual's correct category from a single test with multiple outcomes. For example, an individual photograph of a digit belongs in only one of 10 categories. Ideally, the fitted probabilities for each of the 10 categories should not only be all between 0 and 1 but they should add up to 1. The multivariate NN model as described earlier gives an estimated $F(z_D' B_D)$ vector with entries between 0 and 1 but no reason for them to add up to 1. To this end, for such multinomial data the "*softmax*" generalized activation is often recommended, where for a q-vector v,

$$F(v) = g(v) = \frac{1}{\sum_{h=1}^{q} e^{v_h}} (e^{v_1}, \ldots, e^{v_q})'.$$

Recall that in our NNs g is typically being applied to vectors and in multivariate NNs F is also applied to vectors.

Appendix E
Function Minimization/Maximization

Statistical estimates are generally found by either maximizing a log-likelihood function or minimizing some loss function. We consider four algorithms for function minimization/maximization commonly used in Statistical Learning. These are gradiant (steepest) descent, Newton-Raphson, Gauss-Newton, and EM (Expectation-Maximization). These methods involve finding critical points, i.e., for a real valued function of a vector, say, $g(\beta)$ we find values $\hat{\beta}$ such that $0 = \mathbf{d}_\beta g(\beta)|_{\beta=\hat{\beta}}$, cf. Section A.9. Of course critical points can be local minima, or local maxima, or saddlepoints.

When maximizing penalized likelihood functions, the likelihoods are generally differentiable but some penalty functions, like that used in the LASSO, are not differentiable everywhere, in which case these methods cannot be applied directly.

E.1 Examples

We introduce three common examples and find the first and second derivatives of their criterion functions.

EXAMPLE E.1.1. *Standard Linear Models.*
The standard linear model is

$$Y = X\beta + e, \quad \mathrm{E}(e) = 0, \quad \mathrm{Cov}(e) = \sigma^2 I.$$

The least squares criterion is to choose an estimate of β that minimizes the squared Euclidean distance between Y and $X\beta$, namely

$$\|Y - X\beta\|^2 \equiv (Y - X\beta)'(Y - X\beta).$$

It is well-known that any $\hat{\beta}$ minimizing the squared distance criterion also maximizes the likelihood (and log-likelihood) associated with the elements of e having independent $N(0, \sigma^2)$ distributions, e.g., Christensen (2020, Section 2.4).

Using the chain rule, the first derivative of the squared error loss function is

$$\begin{aligned}\mathbf{d}_\beta (Y-X\beta)'(Y-X\beta) &= \left[\mathbf{d}_\mu \mu'\mu)|_{\mu=(Y-X\beta)}\right]\left[\mathbf{d}_\beta (Y-X\beta)\right] \\ &= [2(Y-X\beta)'][-X] = -2(Y-X\beta)'X. \quad (1)\end{aligned}$$

Note that setting the derivative equal to 0 leads to the well known *normal equations* $X'X\beta = X'Y$ for finding least squares estimates. In regression models, defined to be models for which $(X'X)^{-1}$ exists, the normal equations have the unique solution $\hat{\beta} = (X'X)^{-1}X'Y$. When $(X'X)^{-1}$ does not exist, it can be replaced by any generalized inverse $(X'X)^-$, defined to be any matrix that satisfies $(X'X)(X'X)^-(X'X) = (X'X)$.

The second derivative of $\|Y - X\beta\|^2$ is

$$\begin{aligned}\mathbf{d}^2_{\beta\beta}(Y-X\beta)'(Y-X\beta) &= \mathbf{d}_\beta\left[-2X'(Y-X\beta)\right] \\ &= \mathbf{d}_\beta\left[-2X'Y + 2X'X\beta\right] = 2X'X. \quad (2)\end{aligned}$$

Because the second derivative matrix is nonnegative definite ($u'Au \geq 0$ for any vector u) regardless of the value of β, all critical points are global minima but there may be an infinite number of them. When the second derivative matrix is positive definite ($u'Au > 0$ for any $u \neq 0$), there will be a unique minimum.

EXAMPLE E.1.2. *Log-Linear Models.*

Log-linear models involve observing a random q vector of counts, say, n and modeling their expectation, say $\mathrm{E}(n) \equiv m$, by specifying $\log(m) \equiv \mu = X\beta$, where the log function (like the exponent function used later) is applied elementwise to any vector, and it is merely convenient to have the name μ for $\log(m)$. The real modeling is in $X\beta$ which, just like standard linear models, has X as a known $n \times p$ model matrix and β as a p-vector of unknown parameters. The parameter vector β determines both μ and m. Sometimes m and μ uniquely determine β.

As discussed in Chapters 10 and 12 of Christensen (2024), the most commonly used probabilistic models for n are independent Poisson sampling, multinomial sampling, and product-multinomial sampling. Letting J denote a vector of 1s, all of these sampling schemes lead to a log-likelihood function that is some fixed additive constant plus

$$\ell(n,\mu) \equiv n'\mu - \sum_{i=1}^{q} e^{\mu_i} = n'\mu - J'm = n'X\beta - J'\exp(X\beta), \quad (3)$$

so we seek to maximize this as a function of β. Often there are many β vectors that give the same maximization.

The first derivative with respect to μ is

$$\mathbf{d}_\mu \ell(n,\mu) = \mathbf{d}_\mu\left(n'\mu - \sum_{i=1}^{q} e^{\mu_i}\right) = n' - (e^{\mu_1},\ldots,e^{\mu_q}) = n' - m'.$$

To get the first derivative with respect to β use the chain rule,

$$\mathbf{d}_\beta \ell[n,\mu(\beta)] = \left[\mathbf{d}_\mu \ell(n,\mu)\right]\left[\mathbf{d}_\beta X\beta\right] = (n-m)'X. \tag{4}$$

Here it is important to remember that m is a function of μ which is a function of β, so this is really $[n-m(\beta)]'X$. The *likelihood equations* are defined by $[n-m(\beta)]'X = 0$. Solving the likelihood equations provides a critical point.

For the second derivative,

$$\mathbf{d}^2_{\beta\beta} \ell[n,\mu(\beta)] = \mathbf{d}_\beta \left[X'n - X'm(\beta)\right] = -X'\mathbf{d}_\beta m(\beta). \tag{5}$$

Write

$$X = \begin{bmatrix} x_1' \\ \vdots \\ x_q' \end{bmatrix} = [x_{ij}],$$

then

$$m(\beta) = (m_1,\ldots,m_q)' = \left(e^{x_1'\beta},\ldots,e^{x_q'\beta}\right)'.$$

Therefore,

$$\begin{aligned} \mathbf{d}_\beta m(\beta) &= \begin{bmatrix} x_{11}e^{x_1'\beta} & \cdots & x_{1p}e^{x_1'\beta} \\ \vdots & & \vdots \\ x_{q1}e^{x_q'\beta} & \cdots & x_{qp}e^{x_q'\beta} \end{bmatrix} \\ &= \begin{bmatrix} x_1'e^{x_1'\beta} \\ \vdots \\ x_q'e^{x_q'\beta} \end{bmatrix} = \begin{bmatrix} x_1'm_1 \\ \vdots \\ x_q'm_q \end{bmatrix} = \begin{bmatrix} m_1 & & 0 \\ & \ddots & \\ 0 & & m_q \end{bmatrix} \begin{bmatrix} x_1' \\ \vdots \\ x_q' \end{bmatrix} \\ &= D(m)X, \end{aligned} \tag{6}$$

where $m = m(\beta)$. Substitution of (6) into (5) gives

$$\mathbf{d}^2_{\beta\beta} \ell[n,\mu(\beta)] = -X'D[m(\beta)]X. \tag{7}$$

It is implicit in our models that every element of $m(\beta)$ is positive, so the negative of the second derivative is nonnegative definite and critical points $\hat{\beta}$ maximize the likelihood function. When the negative of the second derivative is positive definite, the maxima will be unique.

EXAMPLE E.1.3. *Logistic Regression*

All logistic models are special cases of log-linear models, so one has two choices for developing the theory. One can either apply the log-linear model theory to the special case or one can develop the logistic theory from scratch. Chapter 11 of Christensen (2024) applies the special case, the details of which we spare the reader. The name "logistic regression" is commonly used even when the models do not qualify as "regression."

Logistic regression as a form of binomial regression was discussed in Chapter 9 where the data were taken as independent observations y_i with $N_i y_i \sim \text{Bin}(N_i, p_i)$,

$i = 1, \ldots, n$. Here y_i is the proportion of successes from the ith binomial. The data pair $[N_i y_i, (N_i - N_i y_i)]$ constitute a particular form of the product-multinomial data alluded to in the previous example.

Logistic regression uses the *logistic transformation* that takes real valued numbers into the open unit interval $(0,1)$ via,

$$F(u) \equiv \frac{e^u}{1+e^u} = \frac{1}{1+e^{-u}}.$$

Any logistic regression analysis also requires use of the inverse of the logistic transformation, known is the *logit transformation*. It transforms numbers p in $(0,1)$ into the real line via,

$$\text{logit}(p) \equiv \log[p/(1-p)].$$

Logits are often called what they are, *log odds*, and "logit model" is a viable alternative to the name "logistic regression."

To define a logistic regression model write a model matrix

$$X = \begin{bmatrix} x_1' \\ \vdots \\ x_n' \end{bmatrix} = [x_{ij}]$$

except that here X is an $n \times d$ matrix so that probabilities p are not confused with the number of columns in X. Now specify probabilities for the binomial observations as

$$p_i(\beta) = \frac{e^{x_i'\beta}}{1+e^{x_i'\beta}} = F(x_i'\beta), \qquad i = 1, \ldots, n. \tag{8}$$

Vectorize the binomial proportions y_i, sample sizes N_i, and probabilities p_i as

$$Y \equiv \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad N \equiv \begin{bmatrix} N_1 \\ \vdots \\ N_n \end{bmatrix}, \quad p \equiv \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix}.$$

The binomial counts can now be written $[D(N)]Y$.

In Chapter 9 the likelihood function for logistic regression was discussed and in (9.1.1) the deviance function was given. Maximizing the likelihood is equivalent to minimizing the deviance which can be written as

$$\mathscr{D}(\beta) = \sum_{i=1}^{n} N_i \mathscr{L}(y_i, x_i'\beta),$$

where

$$\mathscr{L}(y,u) \equiv -2\{y \log[F(u)] + (1-y)\log[1-F(u)]\}.$$

Finding $\mathbf{d}_\beta \mathscr{D}(\beta)$ and $\mathbf{d}^2_{\beta\beta} \mathscr{D}(\beta)$ are going to get somewhat involved with multiple applications of the chain rule.

To get the first derivative, begin by using the chain rule to obtain

$$\mathbf{d}_u \mathscr{L}(y,u) = -2\left[\mathbf{d}_v \{y\log(v) + (1-y)\log(1-v)\}|_{v=F(u)}\right]\left[\mathbf{d}_u F(u)\right].$$

Note that

$$\mathbf{d}_v \{y\log(v) + (1-y)\log(1-v)\} = \frac{y}{v} - \frac{1-y}{1-v} = \frac{y-v}{v(1-v)}$$

and, as discussed in Example D.2.2, for a scalar u,

$$\dot{F}(u) \equiv \mathbf{d}_u F(u) = e^u/(1+e^u)^2 = F(u)[1-F(u)], \tag{9}$$

so

$$\mathbf{d}_u \mathscr{L}(y,u) = -2\left\{\frac{y-F(u)}{F(u)[1-F(u)]}\right\}\{F(u)[1-F(u)]\} = -2[y-F(u)].$$

Finally

$$\mathbf{d}_\beta \mathscr{L}(y_i, x_i'\beta) = \left[\mathbf{d}_u \mathscr{L}(y,u)|_{u=x_i'\beta}\right]\left[\mathbf{d}_\beta x_i'\beta\right] = -2\left[y_i - F(x_i'\beta)\right]x_i'.$$

Now getting back to the deviance function

$$\begin{aligned}
\mathbf{d}_\beta \mathscr{D}(\beta) &= \sum_{i=1}^n N_i \mathbf{d}_\beta \mathscr{L}(y_i, x_i'\beta) \\
&= \sum_{i=1}^n N_i(-2)\left[y_i - F(x_i'\beta)\right]x_i' \\
&= -2\sum_{i=1}^n [y_i - p_i(\beta)]N_i x_i' \\
&= -2[Y - p(\beta)]'D(N)X.
\end{aligned} \tag{10}$$

The likelihood equations set

$$X'D(N)[Y - p(\beta)] = 0.$$

This agrees with the log-linear models approach to this problem.

We now consider the second derivative,

$$\begin{aligned}
\mathbf{d}_{\beta\beta}^2 \mathscr{D}(\beta) &= \mathbf{d}_\beta\left\{-2X'D(N)[Y - p(\beta)]\right\} \\
&= -2X'D(N)\mathbf{d}_\beta[Y - p(\beta)] = 2X'D(N)\mathbf{d}_\beta p(\beta),
\end{aligned} \tag{11}$$

so we need to compute derivatives for model (8). Applying the result (9) for scalar u to vectors u,

$$\mathbf{d}_\beta p(\beta) = \mathbf{d}_\beta F(X\beta) = \left[\mathbf{d}_u F(u)|_{u=X\beta}\right]\left[\mathbf{d}_\beta X\beta\right]$$

$$= D\{F(X\beta)[1-F(X\beta)]\}X = D\{p(\beta)[1-p(\beta)]\}X, \quad (12)$$

where we are treating $F(u)[1-F(u)]$ and $p(v)[1-p(v)]$ as scalar functions of scalars u and v and then applying them to the vectors $X\beta$ and β, respectively. Combining the results of (11) and (12) gives

$$\mathbf{d}^2_{\beta\beta}\mathscr{D}(\beta) = 2X'D(N)D\{p(\beta)[1-p(\beta)]\}X. \quad (13)$$

The model ensures that $0 < p_i(\beta) < 1$, so $0 < p_i(\beta)[1-p_i(\beta)] < 1$, and the second derivative matrix is nonnegative definite. Thus the deviance will be minimized. If the second derivative is positive definite, the minimization occurs for a unique value of β.

E.2 Gradient (Steepest) Descent

Computationally, gradient descent is very easy to perform but it is often inefficient. It seems to be the method of choice for most high dimensional problems. To minimize $g(\beta)$ from some starting value β_0 and for a given scalar jump size $\eta > 0$, recursively define a sequence

$$\beta_{t+1} \equiv \beta_t - \eta[\mathbf{d}_\beta g(\beta_t)]'$$

At convergence of the sequence, $\beta_{t+1} = \beta_t$, so it must be that $[\mathbf{d}_\beta g(\beta_t)]' = 0$, hence β_t is a critical point of g. To maximize the function use

$$\beta_{t+1} \equiv \beta_t + \eta[\mathbf{d}_\beta g(\beta_t)]'.$$

In the absence of better ideas, often one takes $\beta_0 = 0$, but if g is a complicated function, one should take a variety of initial values and see what happens.

When the target function $g(\cdot)$ has isobars similar to elongated ellipsoids, like those illustrated in Figures 8.5 and 8.6, steepest descent tends to move in the direction of the minor axis whereas the location of the extreme point is most rapidly approached by moving in the direction of the major axis. In fact, depending on step size, the iterations can zigzag back and forth, mostly in the direction of the minor axis while making only small gains along the major axis toward the extreme point. This illustrates the inefficiency often displayed by gradient descent. Certainly in linear regression one can solve this problem by replacing the original predictor variables by their equivalent principal components. This changes the elongated ellipsoid into a circular ball. **Or does it merely rotate the axes?** Using principal components may also be helpful for nonlinear regression problems like neural networks. Alternatively, if steepest descent is zigzagging, averaging the two steps will largely cancel the wasted movement and put more of the relative change in the direction of the major axis.

EXAMPLE E.2.1, *Standard Linear Models*.
To minimize $\|Y - X\beta\|^2$, using the derivative given in (E.1.1) define the sequence

$$\beta_{t+1} = \beta_t - \eta \left[-2X'(Y - X\beta_t)\right]$$

or, incorporating the constant 2 into η,

$$\beta_{t+1} = \beta_t + \eta \left[X'Y - X'X\beta_t\right].$$

If the dimension p is extremely large, so that finding the inverse (or even the generalized inverse) of $X'X$ is difficult, this may be an effective method of finding least squares estimates, c.f., Christensen (2024b).

EXAMPLE E.2.2. *Log-Linear Models*.
Because we want to maximize $\ell[n, \mu(\beta)]$ change the sign on the derivative term in (E.1.4) and define

$$\beta_{t+1} = \beta_t + \eta \left[X'n - X'm(\beta_t)\right]$$

or

$$\beta_{t+1} = \beta_t + \eta \left[X'n - X' \begin{pmatrix} e^{x_1'\beta_t} \\ \vdots \\ e^{x_q'\beta_t} \end{pmatrix}\right].$$

As in the previous example, when p is very large, this may be more effective than using Newton-Raphson which involves computing the inverse of a $p \times p$ matrix.

EXAMPLE E.2.3. *Logistic Regression*.
We want to minimize the deviance so using the derivative in (E.1.10) define

$$\beta_{t+1} = \beta_t - \eta \left\{-2X'D(N)[Y - p(\beta_t)]\right\}$$

or, incorporating the constant 2 into η,

$$\beta_{t+1} = \beta_t + \eta X'D(N)[Y - p(\beta_t)].$$

Here, when the dimension d is very large, this may be more effective than using Newton-Raphson which involves computing the inverse of a $d \times d$ matrix.

In our three examples, we have used all of the data and that is sometimes referred to as using *batch gradient descent*. The transposed derivatives all have the form $X'W = \sum_i x_i w_i'$ where we have written $W' = [w_1, \ldots, w_n]$ and W depends on β. An alternative procedure is to use individual data points in

$$\beta_{t,i+1} = \beta_{t,i} - \eta x_i w_i'(\beta_{t,i}), \quad i = 1, \ldots, n$$

with $\beta_{t+1,1} \equiv \beta_{t,n+1}$. Defining convergence as $\beta_{t,n+1} = \cdots = \beta_{t,1}$, we will have a critical point $X'W(\beta_{t,1}) = 0$. For reasons unfathomable to me, this is often referred to as *stochastic gradient descent*. Alternatively, rather than looking at each individual row of X, you could break the rows down into groups (minibatches) and do separate iterations for each minibatch. This is *minibatch gradient descent*.

E.3 Newton-Raphson

To find a sequence of points leading to a critical point, i.e., a solution to $0 = \mathbf{d}_\beta g(\beta)$, Newton-Raphson uses a first-order Taylor's approximation to the derivative function about the current value β_t, i.e.,

$$[\mathbf{d}_\beta g(\beta)]' \doteq [\mathbf{d}_\beta g(\beta_t)]' + [\mathbf{d}^2_{\beta\beta} g(\beta_t)](\beta - \beta_t).$$

Set $0 = \mathbf{d}_\beta g(\beta)$ and to define the next element in the sequence, β_{t+1}, substitute this for β and use the approximation to get

$$0 = [\mathbf{d}_\beta g(\beta_t)]' + [\mathbf{d}^2_{\beta\beta} g(\beta_t)](\beta_{t+1} - \beta_t).$$

Multiplying through by the inverse of the second derivative matrix gives,

$$0 = [\mathbf{d}^2_{\beta\beta} g(\beta_t)]^{-1}[\mathbf{d}_\beta g(\beta_t)]' + (\beta_{t+1} - \beta_t)$$

which, after rearranging terms, gives

$$\beta_{t+1} = \beta_t - [\mathbf{d}^2_{\beta\beta} g(\beta_t)]^{-1}[\mathbf{d}_\beta g(\beta_t)]'.$$

At convergence, $\beta_{t+1} = \beta_t$, so

$$0 = [\mathbf{d}^2_{\beta\beta} g(\beta_t)]^{-1}[\mathbf{d}_\beta g(\beta_t)]'. \tag{1}$$

Since we have assumed that $[\mathbf{d}^2_{\beta\beta} g(\beta_t)]$ is invertable, equation (1) holds if and only if

$$0 = [\mathbf{d}_\beta g(\beta_t)]',$$

so β_t is a critical point.

This algorithm tends to be very efficient but when the dimension p of β is large, finding the inverse of the $p \times p$ matrix $[\mathbf{d}^2_{\beta\beta} g(\beta_t)]$ can be very difficult in which case gradient descent might be preferred. When applied to generalized linear models, a class that contains all three of our examples, the Newton-Raphson algorithm can be recast as fitting a sequence of weighted regression models which is known as *iteratively reweighted least squares*. This is the method typically used in computer programs for fitting generalized linear models, e.g, R's `glm` command and SAS's `proc genmod`.

EXAMPLE E.3.1. *Standard Linear Models.*
To minimize $\|Y - X\beta\|^2$, using the first and second derivatives given in equations (E.1.1) and (E.1.2), define the sequence

$$\beta_{t+1} = \beta_t - (2X'X)^{-1}\left[-2X'(Y - X\beta_t)\right]$$

or,

$$\beta_{t+1} = \beta_t + (X'X)^{-1}\left[X'Y - X'X\beta_t\right] = (X'X)^{-1}X'Y \equiv \hat{\beta}$$

where the least squares estimate is well known to be $\hat{\beta} \equiv (X'X)^{-1}X'Y$. In particular, from any starting value, just one iteration of this procedure gives the least squares estimates of a linear model. You cannot get much more efficient than that.

For linear models, there is no point in using this algorithm because we already know the formula for $\hat{\beta}$. When $(X'X)^{-1}$ does not exist, one can use any generalized inverse of $(X'X)$ and the Moore-Penrose generalized inverse $(X'X)^{+}$ leads to the minimum norm least squares estimate.

EXAMPLE E.3.2. *Log-Linear Models*.
To maximize $\ell[n, \mu(\beta)]$ use the first and second derivatives given in equations (E.1.4) and (E.1.7) and define

$$\beta_{t+1} = \beta_t + \left\{X'D\left[m(\beta_t)\right]X\right\}^{-1}\left[X'n - X'm(\beta_t)\right].$$

EXAMPLE E.3.3. *Logistic Regression*.
Using the first and second derivatives in (E.1.10) and (E.1.13), define

$$\beta_{t+1} = \beta_t + \left[X'D(N)D\left\{p(\beta_t)[1-p(\beta_t)]\right\}X\right]^{-1}X'D(N)[Y - p(\beta_t)].$$

Exercise E.1 Show that any solution β_{t+1} to

$$0 = [\mathbf{d}_\beta g(\beta_t)]' + [\mathbf{d}^2_{\beta\beta} g(\beta_t)](\beta - \beta_t),$$

that is, any solution β_{t+1} to the system of linear equations

$$[\mathbf{d}^2_{\beta\beta} g(\beta_t)]\beta = \left\{[\mathbf{d}^2_{\beta\beta} g(\beta_t)]\beta_t - [\mathbf{d}_\beta g(\beta_t)]'\right\},$$

will, upon convergence, give a critical point. Thus we do not need to have $\mathbf{d}^2_{\beta\beta} g(\beta_t)$ invertible. (The point is that we can use solutions based on generalized inverses. The trick is to get convergence. Sticking to Moore-Penrose generalized inverses may help.)

E.4 Gauss-Newton

The Gauss-Newton method applies only to minimizing real valued functions of the form

$$g(\beta) \equiv \|Y - G(\beta)\|^2,$$

where Y is an observed n vector and $G(\cdot)$ is a function that maps p vectors β into n vectors. A major application of this method is to nonlinear regression, e.g., Christensen (1996, Chapter 18) or Christensen (2015, Chapter 23). The Neural Networks of Appendix D are an important special case of nonlinear regression.

The essence of the Gauss-Newton algorithm is to replace $G(\beta)$ with an approximating linear model and then to use standard linear model theory to find the least squares estimate for the approximate model. One does this repeatedly until finding a critical point to the least squares criterion.

Given a current value for β, say, β_t, use the first-order Taylor's approximation

$$G(\beta) \doteq G(\beta_t) + [\mathbf{d}_\beta G(\beta_t)](\beta - \beta_t).$$

To simplify notation write

$$\tilde{X}_t \equiv [\mathbf{d}_\beta G(\beta_t)]$$

so that

$$G(\beta) \doteq \left[G(\beta_t) - \tilde{X}_t\beta_t\right] + \tilde{X}_t\beta. \tag{1}$$

Now define

$$\tilde{Y}_t \equiv Y - \left[G(\beta_t) - \tilde{X}_t\beta_t\right],$$

with the idea of finding β_{t+1} as the least squares estimate from fitting the model

$$\tilde{Y}_t = \tilde{X}_t\beta + e.$$

In particular, using (1)

$$Y - G(\beta) \doteq Y - \left[G(\beta_t) - \tilde{X}_t\beta_t\right] - \tilde{X}_t\beta = \tilde{Y}_t - \tilde{X}_t\beta$$

and

$$\|Y - G(\beta)\|^2 \doteq \|\tilde{Y}_t - \tilde{X}_t\beta\|^2.$$

With β_{t+1} the least squares estimate from the approximation, from standard linear model theory

$$\begin{aligned}
\beta_{t+1} &= (\tilde{X}_t'\tilde{X}_t)^{-1}\tilde{X}_t'\tilde{Y}_t \\
&= (\tilde{X}_t'\tilde{X}_t)^{-1}\tilde{X}_t'\left[Y - G(\beta_t) + \tilde{X}_t\beta_t\right] \\
&= (\tilde{X}_t'\tilde{X}_t)^{-1}\tilde{X}_t'\tilde{X}_t\beta_t + (\tilde{X}_t'\tilde{X}_t)^{-1}\tilde{X}_t'[Y - G(\beta_t)] \\
&= \beta_t + \left([\mathbf{d}_\beta G(\beta_t)]'[\mathbf{d}_\beta G(\beta_t)]\right)^{-1}[\mathbf{d}_\beta G(\beta_t)]'[Y - G(\beta_t)].
\end{aligned}$$

To check that this works, note that similar to our discussion of standard linear models

$$\mathbf{d}_\beta \|Y - G(\beta)\|^2 = 2[Y - G(\beta)]'[-\mathbf{d}_\beta G(\beta)],$$

so a critical point β_t should have

$$0 = [\mathbf{d}_\beta G(\beta_t)]'[Y - G(\beta_t)].$$

At convergence $\beta_{t+1} = \beta_t$ so

$$0 = \left([\mathbf{d}_\beta G(\beta_t)]'[\mathbf{d}_\beta G(\beta_t)]\right)^{-1}[\mathbf{d}_\beta G(\beta_t)]'[Y - G(\beta_t)]$$

but with $\left([\mathbf{d}_\beta G(\beta_t)]'[\mathbf{d}_\beta G(\beta_t)]\right)$ nonsingular, that only happens when

$$0 = [\mathbf{d}_\beta G(\beta_t)]'\,[Y - G(\beta_t)] = \mathbf{d}_\beta \|Y - G(\beta_t)\|^2,$$

so at convergence β_t must be a critical point and one hopes provides least squares estimates.

In nonlinear regression, the ith component of the n-vector $G(\beta)$ typically has the form $\mathrm{E}(y_i) = f(x_i;\beta)$ for some fixed function $f(\cdot)$ and vector of predictors x_i. A generalized linear model is the special case $\mathrm{E}(y_i) = f(x_i'\beta)$ but typically involves additional specific assumptions about the distributions of the y_is, rarely leading to the use of least squares estimates.

E.5 EM (Expectation-Maximization)

Once again, we consider maximizing a real valued function $g(\beta)$ using a recursively defined sequence of values β_t, however the *Expectation-Maximization (EM) algorithm* was developed originally for finding maximum likelihood estimates and explicitly employs the probabilistic structure of likelihood functions. EM also incorporations ideas from *information theory* related to entropy and *Kullback-Leibler divergences* and particularly *Gibbs' inequality* that make it relatively easy to establish that the sequence β_t increases the value of the likelihood function with each iteration. Unfortunately, it is by no means obvious that the sequence will converge to a critical point of the likelihood function. Dempster, Laird, and Rubin (1977) popularized, generalized, and named an algorithm having several antecedents. Wu (1983) cleaned up the convergence issues.

The EM algorithm is designed specifically for *latent (hidden) variable models* in which one posits the existence of unobservable random variables. The factors in factor analysis, cf. Section 11.6, constitute latent variables. Latent variables seem to be particularly useful for missing data problems, for estimating the parameters of Gaussian (normal) mixture models, and are even part of to mixed linear models.

Dempster, A. P., Laird, N. M., and Rubin, D. R. (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39, 1-38.

C. F. Jeff Wu (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11, 95-103.

We begin with some examples

EXAMPLE E.5.1. *Linear Mixed Models.*
A standard mixed linear model adds to the standard linear model a fixed $n \times s$ matrix Z and a random unobservable vector γ by writing

$$Y = X\beta + Z\gamma + e, \quad e \sim N(0, \sigma^2 I), \quad \gamma \sim N[0, D(\theta)], \quad \gamma \perp\!\!\!\perp e.$$

Here θ is an unknown vector of fixed parameters involved in the covariance matrix of γ and $\gamma \perp\!\!\!\perp e$ indicates that γ and e are independent. The latent variable here is γ. (Note that the error vector e in any linear model is also a latent variable but one that is easy to work around.) The marginal distribution of the observable data vector is

$$Y \sim N[X\beta, ZD(\theta)Z' + \sigma^2 I].$$

Using $|\cdot|$ to denote the determinant of a matrix, the density of Y and the likelihood function is

$$\begin{aligned} &f(Y|\beta,\sigma^2,\theta) = L(\beta,\sigma^2,\theta|Y) \\ &= (2\pi)^{-n/2}|ZD(\theta)Z' + \sigma^2 I|^{-1/2} \exp\left\{ \frac{-1}{2}(Y - X\beta)\left[ZD(\theta)Z' + \sigma^2 I\right]^{-1}(Y - X\beta)\right\}, \end{aligned}$$

e.g., Christensen (2019, Chapter 5). However incorporating the unobservable latent variable γ, we can specify the joint distribution of Y and γ given β, σ^2, θ using

$$Y|\gamma,\beta,\sigma^2,\theta \sim N[X\beta + Z\gamma, \sigma^2 I], \tag{1}$$

which does not actually depend on θ, and

$$\gamma|\beta,\sigma^2,\theta \sim N[0, D(\theta)], \tag{2}$$

which does not actually depend on either β or σ^2. We can now write the joint density of Y and γ given β, σ^2, θ as the product of the densities associated with (1) and (2), which is what the EM algorithm will exploit.

EXAMPLE E.5.2. *Gaussian Mixture Models*.
Men are taller than women on average. If you pick a random student from the University of New Mexico, a reasonable model for their height might be $N(\mu_1, \sigma_1^2)$ if they are male and $N(\mu_0, \sigma_0^2)$ if they are female with some probability p that they will be male. If we collect a random sample of students heights, we would want to estimate $p, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2$. That is very easy to do if along with their heights we learn the sex of each individual, but if we get the heights without learning the sexes, the estimation problem is much harder. For this problem, each individual has an observable height y_i along with an unobservable sex γ_i.

$$y_i|\gamma_i \sim N(\mu_\gamma, \sigma_\gamma^2)$$

$$\gamma_i \sim \text{Bern}(p)$$

In this model, we know that $\gamma_i = 0, 1$ determine two populations but we don't know if $\gamma_i = 1$ is a male or a female. In particular, the observable height distribution given by $(p, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2) = (.48, 69, 16, 64, 9)$ is indistinguishable from that given by $(p, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2) = (.52, 64, 9, 69, 16)$. There are a number of ways to fix this problem. One could require $\mu_1 < \mu_2$ or alternatively require $p < 0.5$.

EXAMPLE E.5.3. *Missing Data Models.*

$$f(y|\beta) = \int f(y|\beta,\gamma)p(\gamma|\beta)$$

Expectation step: Define $Q(\beta|\beta_t) = \mathrm{E}_{\gamma|y,\beta_t}[\log f(y,\gamma|\beta)]$ Maximization step: β_{t+1} satisfies $Q(\beta_{t+1}|\beta_t) = \sup_\beta Q(\beta|\beta_t)$

References

Aitchison, J. (1975). Goodness of prediction fit. *Biometrika*, **62**, 547–554.

Aitchison, J. and Dunsmore, I.R. (1975). *Statistical Prediction Analysis*. Cambridge University Press, Cambridge.

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In *Proceedings of the 2nd International Symposium on Information*, edited by B.N. Petrov and F. Czaki. Akademiai Kiado, Budapest.

Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, **71**, 1–10.

Aldrich, John (2005). Fisher and regression. *Statistical Science*, 20, 401–417.

Anderson, J. A. (1972). Separate sample logistic discrimination. *Biometrika*, **59**, 19–35.

Anderson, T.W. (2003). *An Introduction to Multivariate Statistical Analysis*, Third Edition. John Wiley and Sons, New York.

Andrews, D.F., Gnanadesikan, R., and Warner, J.L. (1971). Transformations of multivariate data. *Biometrics*, **27**, 825–840.

Ansley, C.F. and Kohn, R. (1984). On the estimation of ARIMA models with missing values. In *Time Series Analysis of Irregularly Observed Data*, edited by E. Parzen. Springer-Verlag, New York.

Armstrong, M. (1984). Problems with universal kriging. *Journal of the International Association for Mathematical Geology*, **16**, 101–108.

Arnold, Stephen F. (1981). *The Theory of Linear Models and Multivariate Analysis*. John Wiley and Sons, New York.

Atkinson, A. C. (1985). *Plots, Transformations, and Regression: An Introduction to Graphical Methods of Diagnostic Regression Analysis*. Oxford University Press, Oxford.

Banerjee, Sudipto, Carlin, Bradley P., Gelfand, Alan E. (2015). *Hierarchical Modeling and Analysis for Spatial Data*, Second Edition. Chapman & Hall/CRC, Boca Raton, FL.

Bartlett, M.S. (1946). On the theoretical specification of sampling properties of autocorrelated time series. *Journal of the Royal Statistical Society, Supplement*, **8**, 27–41.

Bedrick, E. J. and Tsai, C.-L. (1994). Model selection for multivariate regression in small samples. *Biometrics*, **50**, 226-231.

Beineke, L. A. and Suddarth, S. K. (1979). Modeling joints made with light-gauge metal connector plates. *Forest Products Journal*, **29**, 39–44.

Berger, James O. (1985). *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York.

Berke, O. (1998). On spatio-temporal prediction for on-line monitoring data. *Communications in Statistics, Series A*, **27**, 2343–2369.

Berlinet, A. and Thomas-Agnan, C. (2004). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic Publishers, Norwell, MA.

Bivand, R., Pebesma, E. and Gómez-Rubio, V. (2013). *Applied Spatial Data Analysis with R*, Second Edition. Springer, New York.

Bloomfield, Peter (1976). *Fourier Analysis of Time Series: An Introduction*. John Wiley and Sons, New York.

Box, George E.P. (1950). Problems in the analysis of growth and wear curves. *Biometrics*, **6**, 362–389.

Box, G.E.P. and Cox, D.R. (1964). An analysis of transformations (with discussion). *Journal of the Royal Statistical Society, Series B*, **26**, 211–246.

Box, George E.P., Jenkins, Gwylem M., and Reinsel, Gregory C. (1994). *Time Series Analysis: Forecasting and Control*, Third Edition. John Wiley and Sons, New York.

Box, George E.P., Jenkins, Gwylem M., Reinsel, Gregory C., and Ljung, Greta M. (2015). *Time Series Analysis: Forecasting and Control*, Fifth Edition. John Wiley and Sons, New York.

Breiman, Leo (1968). *Probability*. Addison-Wesley, Reading, MA.

Breiman, Leo, Friedman, Jerome H., Olshen, Richard A., and Stone, Charles J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont, CA.

Brillinger, David R. (1981). *Time Series: Data Analysis and Theory*, Second Edition. Holden Day, San Francisco.

Brockwell, Peter J. and Davis, Richard A. (1991). *Time Series: Theory and Methods*, Second Edition. Springer-Verlag, New York.

Brockwell, Peter J. and Davis, Richard A. (2002). *Introduction to Time Series and Forecasting*, Second Edition. Springer-Verlag, New York.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, Heidelberg.

Bühlmann, Peter, Kalisch, Markus and Meier, Lukas (2014). High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Applications*, **1**, 255–78.

Carroll, R.J. and Ruppert, D. (1988). *Transformations and Weighting in Regression*. Chapman and Hall, New York.

Cavanaugh, J. E. (1997). Unifying the derivations of the Akaike and corrected Akaike information criteria. *Statistics and Probability Letters*, **31**, 201-208.

Chatfield, Christopher (2003).*The Analysis of Time Series: An Introduction*, Sixth Edition. Chapman and Hall, New York.

Christensen, R. (1984). A note on ordinary least squares methods for two-stage sampling. *Journal of the American Statistical Association*, **79**, 720-721.

Christensen, R. (1987). The analysis of two-stage sampling data by ordinary least squares. *Journal of the American Statistical Association*, **82**, 492–498.

Christensen, Ronald (1989). Lack of fit tests based on near or exact replicates. *Annals of Statistics*, **17**, 673–683.

Christensen, Ronald (1990). The equivalence of predictions from universal kriging and intrinsic random function kriging. *Mathematical Geology*, **22**, 655–664.

Christensen, Ronald (1991). Small sample characterizations of near replicate lack of fit tests. *Journal of the American Statistical Association*, **86**, 752–756.

Christensen, Ronald (1993). Quadratic covariance estimation and equivalence of predictions. *Mathematical Geology*, **25**, 541–558.

Christensen, R. (1996). *Analysis of Variance, Design, and Regression: Applied Statistical Methods*. Chapman and Hall, London.

Christensen, Ronald (1997). *Log-Linear Models and Logistic Regression*, Second Edition. Springer-Verlag, New York.

Christensen, Ronald (2011). *Plane Answers to Complex Questions: The Theory of Linear Models*, Fourth Edition. Springer-Verlag, New York.

Christensen, Ronald (2014). Comment. *The American Statistician*, **68**, 13–17

Christensen, R. (2015). *Analysis of Variance, Design, and Regression: Linear Modeling for Unbalanced Data*, Second Edition. Chapman and Hall/CRC Pres, Boca Raton, FL.

Christensen, Ronald (2019). *Advanced Linear Modeling: Statistical Learning and Dependent Data*, Fifth Edition. Springer-Verlag, New York.

Christensen, Ronald (2020). *Plane Answers to Complex Questions: The Theory of Linear Models*, Fifth Edition. Springer-Verlag, New York.

Christensen, R. and Bedrick, E. J. (1997). Testing the independence assumption in linear models. *Journal of the American Statistical Association*, **92**, 1006-1016.

Christensen, Ronald and Bedrick, Edward J. (1999). A survey of some new alternatives to Wald's variance component test. *Tatra Mountains Mathematical Publications*, **17**, 91–102.

Christensen, R., Johnson, W., Branscum, A. and Hanson, T. E. (2010). *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians*. Chapman and Hall/CRC Press, Boca Raton, FL.

Christensen, Ronald and Lin, Yong (2015). Lack-of-fit tests based on partial sums of residuals. *Communications in Statistics–Theory and Methods*, **44**, 2862–2880.

Christensen, Ronald and Sun, Siu Kei (2010). Alternative goodness-of-fit tests for linear models. *Journal of the American Statistical Association*, **105**, 291–301.

Clayton, Murray K., Geisser, Seymour, and Jennings, Dennis E. (1985). A comparison of several model selection procedures. In *Bayesian Inference and Decision Techniques*, edited by P. Goel and A. Zellner. Elsevier Science Publishers B.V., Amsterdam.

Cliff, A. and Ord, J.K. (1981). *Spatial Processes: Models and Applications*. Pion, London.

Cooper, Yaim (2021). Global minima of overparameterized neural networks. *SIAM Journal on Mathematics of Data Science*, **3**(2), 676-691. https://doi.org/10.1137/19M1308943 "We explore some mathematical features of the loss landscape of overparameterized neural networks. A priori, one might imagine that the loss function looks like a typical function from $\mathbf{R}^d$ to $\mathbf{R}$, in particular, that it has discrete global minima. In this paper, we prove that in at least one important way, the loss function of an overparameterized neural network does not look like a typical function. If a neural net has d parameters and is trained on n data points $(x_i, y_i) \in \mathbf{R}^s \times \mathbf{R}^r$, with $d > rn$, we show that the locus M of global minima of L is usually not discrete but rather an $(d - rn)$-dimensional submanifold of $\mathbf{R}^d$. In practice, neural nets commonly have orders of magnitude more parameters than data points, so this observation implies that M is typically a very-high-dimensional submanifold of $\mathbf{R}^d$."

Cox, D.R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society, Series B*, **34**, 187–220.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerical Mathematics*, **31**, 377–403.

David, M. (1977). *Geostatistical Ore Reserve Estimations*. Elsevier, New York.

de Barra, G. (1981). *Measure Theory and Integration*. Horwood Publishing, West Sussex.

Delfiner, P. (1976). Linear estimation of nonstationary spatial phenomena. In *Advanced Geostatistics in the Mining Industry*, edited by M. Guarascia, M. David, and C. Hüijbregts. Reidel, Dordrecht.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **39**, 1–38.

Devore, Jay L. (1991). *Probability and Statistics for Engineering and the Sciences*, Third Edition. Brooks/Cole, Pacific Grove, CA.

Diamond, P. and Armstrong, M. (1983). Robustness of variograms and conditioning of kriging matrices. *Journal of the International Association for Mathematical Geology*, **16**, 809–822.

Diderrich, George T. (1985). The Kalman filter from the perspective of Goldberger-Theil estimators. *The American Statistician*, **39**, 193–198.

Diggle, Peter J. (1990). *Time Series: A Biostatistical Introduction*. Oxford University Press, New York.

Diggle, Peter J., Heagerty, Patrick, Liang, Kung-Yee, and Zeger, Scott L. (2002). *Analysis of Longitudinal Data*, Second Edition. Oxford University Press, Oxford.

Diggle, P.J., Tawn, J.A., and Moyeed, R.A. (1998). Model-based geostatistics. *Applied Statistics*, **47**, 299–326.

Dillon, Wm. R. and Goldstein, Matthew (1984). *Multivariate Analysis: Methods and Applications*. John Wiley and Sons, New York.

Dixon, W. J. and Massey, F. J., Jr. (1969). *Introduction to Statistical Analysis*, Third Edition. McGraw-Hill, New York.

Dixon, Wilfrid J. and Massey, Frank J., Jr. (1983). *Introduction to Statistical Analysis*, Fourth Edition. McGraw-Hill, New York.

Doob, J.L. (1953). *Stochastic Processes*. John Wiley and Sons, New York.

Draper, Norman and Smith, Harry (1981). *Applied Regression Analysis*, Second Edition. John Wiley and Sons, New York.

Draper, Norman R. and van Nostrand, R. Craig (1979). Ridge regression and James-Stein estimation: Review and comments *Technometrics*, **21**, 451–466.

Durbin, J. and Koopman, S.J. (2012). *Time Series Analysis by State Space Methods*, Second Edition. Oxford University Press, Oxford.

Eaton, Morris L. (1983). *Multivariate Statistics: A Vector Space Approach*. John Wiley and Sons, New York.

Eaton, Morris L. (1985). The Gauss-Markov theorem in multivariate analysis. In *Multivariate Analysis - VI*, edited by P.R. Krishnaiah. Elsevier Science Publishers B.V., Amsterdam.

Edwards, David (2000). *Introduction to Graphical Modeling*, Second Edition. Springer-Verlag, Berlin.

Efromovich, Sam (1999). *Nonparametric Curve Estimation: Methods, Theory, and Applications*. Springer-Verlag, New York.

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

Efron, B., Hastie, T. (2016). *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science.* Cambridge University Press, Cambridge.

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

Eilers, P.H.C. and Marx, B.D. (1996). Flexible smoothing with b-splines and penalties (with discussion). *Statistical Science*, **11**, 89–121.

Eubank, Randall L. (1988). *Spline Smoothing and Nonparametric Regression*. Marcel Dekker, New York.

Eubank, R. (1999). *Nonparametric Regression and Spline Smoothing*. Marcel Dekker, New York.

Everitt, Brian and Hothorn, Torsten (2011). *An Introduction to Applied Multivariate Analysis with R.* Springer, New York.

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

Ferguson, Thomas S. (1967). *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, New York.

Ferguson, Thomas S. (1996). *A Course in Large Sample Theory*. Chapman & Hall/CRC, Boca Raton.

Fisher, Ronald A. (1936). The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, **7**, 179–188.

Fisher, Ronald A. (1938). The statistical utilization of multiple measurements. *Annals of Eugenics*, **8**, 376–386.

Fitzmaurice, Garrett M., Laird, Nan M., Ware, James H. (2011). *Applied Longitudinal Analysis*, 2nd Edition. Wiley, New York.

Forbes, J. D. (1857). Further experiments and remarks on the measurement of heights by the boiling point of water. *Transactions of the Royal Society of Edinburgh*, **21**, 135–143.

Friedman, Jerome H. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association*, **84**, 165–175.

Fuller, Wayne A. (1976). *Introduction to Statistical Time Series*. John Wiley and Sons, New York.

Fuller, Wayne A. (1996). *Introduction to Statistical Time Series*, Second Edition. John Wiley and Sons, New York.

Furnival, G. M. and Wilson, R. W. (1974). Regression by leaps and bounds. *Technometrics*, **16**, 499-511.

Geisser, Seymour (1970). Bayesian Analysis of Growth Curves, *Sankhya*, Series A, **32**, 53–64.

Geisser, Seymour (1971). The inferential use of predictive distributions. In *Foundations of Statistical Inference*, edited by V.P. Godambe and D.A. Sprott. Holt, Rinehart, and Winston, Toronto.

Geisser, Seymour (1977). Discrimination, allocatory and separatory, linear aspects. In *Classification and Clustering*, edited by J. Van Ryzin. Academic Press, New York.

Gelfand, Alan E., Peter Diggle, Peter, Guttorp, Peter, Fuentes, Montserrat (2010). *Handbook of Spatial Statistics* Chapman & Hall/CRC, Boca Raton, FL.

Geweke, J.F. and Singleton, K.J. (1980). Interpreting the likelihood ratio statistic in factor models when sample size is small. *Journal of the American Statistical Association*, **75**, 133–137.

Gnanadesikan, R. (1977). *Methods for Statistical Data Analysis of Multivariate Observations*. John Wiley and Sons, New York.

Goldberger, Arthur S. (1962). Best linear unbiased prediction in the generalized linear regression model. *Journal of the American Statistical Association*, **57**, 369–375.

Goldstein, M. and Smith, A.F.M. (1974). Ridge-type estimators for regression analysis. *Journal of the Royal Statistical Society, Series B*, **26**, 284–291.

Golub, G., Heath, M. and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, **21**, 215–223.

Green, P.J. and Silverman, B.W. (1994). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, Chapman and Hall, London.

Greenhouse, S.W. and Geisser, S. (1959). On methods in the analysis of profile data. *Psychometrika*, **24**, 95–112.

Gu, C. (2002). *Smoothing Spline ANOVA Models*. Springer, New York.

Gupta, N.K. and Mehra, R.K. (1974). Computational aspects of maximum likelihood estimation and reduction in sensitivity function calculations. *IEEE Transactions on Automatic Control*, **AC-19**, 774–783.

Hader, R. J. and Grandage, A. H. E. (1958). Simple and multiple regression analyses. In *Experimental Designs in Industry*, edited by V. Chew, pp. 108–137. John Wiley and Sons, New York.

Halmos, P. R. (1958). *Finite-Dimensional Vector Spaces*, Second Edition. Van Nostrand, Princeton, NJ.

Hand, D.J. (1981). *Discrimination and Classification*. John Wiley and Sons, New York.

Hand, D.J. (1983). A comparison of two methods of discriminant analysis applied to binary data. *Biometrics*, **39**, 683–694.

Hand, D.J. and Crowder, M.J. (1996). *Practical Longitudinal Data Analysis*. Chapman and Hall, London.

Handcock, Mark S. and Stein, Michael L. (1993). A Bayesian analysis of kriging. *Technometrics*, **35**, 403–410.

Handcock, Mark S. and Wallis, James R. (1994). An approach to statistical spatial-temporal modeling of meterological fields (with discussion). *Journal of the American Statistical Association*, **89**, 368–378.

Hannan, Edward James (1970). *Multiple Time Series*. John Wiley and Sons, New York.

Harrison, P.J. and Stevens, C.F. (1971). A Bayesian approach to short-term forecasting. *Operations Research Quarterly*, **22**, 341–362.

Harrison, P.J. and Stevens, C.F. (1976). Bayesian forecasting. *Journal of the Royal Statistical Society, Series B*, **38**, 205–247.

Hart, Jeffrey D. (1997). *Nonparametric Smoothing and Lack-of-Fit Tests*. Springer-Verlag, New York.

Harville, David A. (1997). *Matrix Algebra From a Statistician's Perspective*. Springer, New York.

Hastie, T., Tibshirani, R. and Friedman, J. (2016). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition. Springer, New York.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparcity: The Lasso and Generalizations*. Chapman and Hall, Boca Raton, FL.

Heck, D.L. (1960). Charts of some upper percentage points of the distribution of the largest characteristic root. *Annals of Mathematical Statistics*, **31**, 625–642.

Heckman, Nancy (2012). The theory and application of penalized methods or Reproducing Kernel Hilbert Spaces made easy. *Statistics Surveys*, **6**, 113–141.

Henderson, C. R. (1953). Estimation of variance and covariance components. *Biometrics*, **9**, 226–252.

Hodges, James S. (2014). *Richly Parameterized Linear Models: Additive, Time Series, and Spatial Models Using Random Effects*. Chapman and Hall/CRC, Boca Raton, FL.

Hoerl, A.E. and Kennard, R. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67.

Hotelling, Harold (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, **24**, 417-441, 498–520.

Huber, P. J. and Ronchetti, E. M. (2009). *Robust Statistics*, Second Edition. John Wiley and Sons, New York.

Hüijbregts, C.J. (1975). Regionalized variables and quantitative analysis of spatial data. In *Display and Analysis of Spatial Data*, edited by J.C. Davis and M.J. McCullagh. John Wiley and Sons, New York.

Hurvich, C.M. and Tsai, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika*, **76**, 297–308.

Hurvich, C.M. and Tsai, C-L. (1995). Relative rates of convergence for efficient model selection criteria in linear regression. *Biometrika*, **82**, 418–425.

Huynh, H. and Feldt, L.S. (1976). Estimation of the Box correction for degrees of freedom from sample data in randomized block and split-plot designs. *Journal of Educational Statistics*, **1**, 69–82.

Hyvärinen, Aapo, Karhunen, Juha and Oja, Erkki (2001). *Independent Component Analysis.* John Wiley and Sons, New York.

Isaaks, Edward H. and Srivastava, R. Mohan (1989). *An Introduction to Applied Geostatistics* Oxford University Press, Oxford.

James, Gareth; Witten, Daniela; Hastie, Trevor; and Tibshirani, Robert (2013). *An Introduction to Statistical Learning*. Springer, New York.

Jensen, R. J. (1977). Evinrude's computerized quality control productivity. *Quality Progress*, **X, 9**, 12–16.

Jiang, Jiming (2007). *Linear and Generalized Linear Mixed Models and Their Applications*. Springer, New York.

Johnson, Richard A. and Wichern, Dean W. (2007). *Applied Multivariate Statistical Analysis*, Sixth Edition. Prentice–Hall, Englewood Cliffs, NJ.

Jolicoeur, P. and Mosimann, J.E. (1960). Size and shape variation on the painted turtle: A principal component analysis. *Growth*, **24**, 339–354.

Jolliffe, I.T. (2002). *Principal Component Analysis*, Second Edition. Springer-Verlag, New York.

Jones, R.H. (1980). Maximum likelihood fitting of ARMA models to time series with missing observations. *Technometrics*, **22**, 389–396.

Jöreskog, K.G. (1975). Factor analysis by least squares and maximum likelihood. In *Statistical Methods for Digital Computers*, edited by K. Enslein, A. Ralston, and H.S. Wilf. John Wiley and Sons, New York.

Journel, A.G. and Hüijbregts, Ch.J. (1978). *Mining Geostatistics*. Academic Press, New York.

Kalbfleisch, John D. and Prentice, Ross L. (1980). *The Statistical Analysis of Failure Time Data*. John Wiley and Sons, New York.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, **82**, 34–45.

Kalman, R.E. and Bucy, R.S. (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering*, **83**, 95–108.

Kenward, M.G., Roger, J.H. (1997). Small sample inference for fixed effects from restricted maximum likelihood, *Biometrics*, **53**, 983–997.

Kenward, M.G., Roger, J.H. (2009). An improved approximation to the precision of fixed effects from restricted maximum likelihood. *Computational Statistics and Data Analysis*, **53**, 2583–2595.

Khatri, C.G. (1966). A note on a MANOVA model applied to problems in growth curves. *Annals of the Institute of Statistical Mathematics*, **18**, 75–86.

Kitanidis, Peter K. (1983). Statistical estimation of polynomial generalized covariance functions and hydrologic applications. *Water Resources Research*, **19**, 909–921.

Kitanidis, Peter K. (1985). Minimum-variance unbiased quadratic estimation of covariances of regionalized variables. *Journal of the International Association for Mathematical Geology*, **17**, 195–208.

Kitanidis, Peter K. (1986). Parameter uncertainty in estimation of spatial functions: Bayesian analysis. *Water Resources Research*, **22**, 499–507.

Kitanidis, Peter K. (1997). *Introduction to Geostatistics: Applications to Hydrogeology.* Cambridge University Press, Cambridge.

Kitanidis, Peter K. and Lane, Robert W. (1985). Maximum likelihood parameter estimation of hydrologic spatial processes by the Gauss–Newton method. *Journal of Hydrology*, **79**, 53–71.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, **2**(12), 1137–1143.

Koopmans, Lambert H. (1974). *The Spectral Analysis of Time Series*. Academic Press, New York.

Kres, Heinz (1983). *Statistical Tables for Multivariate Analysis*. Springer-Verlag, New York.

Khuri, A. I., Mathew, T., and Sinha, B. K. (1998). *Statistical Tests for Mixed Linear Models*. John Wiley and Sons, New York.

Lachenbruch, P.A. (1975). *Discriminate Analysis*. Hafner Press, New York.

Lachenbruch, P.A., Sneeringeer, C., and Revo, L.T. (1973). Robustness of the linear and quadratic discriminant function to certain types of non-normality. *Communications in Statistics*, **1**, 39–57.

Lavine, Michael, Bray, Andrew, and Hodges, Jim (2015). Approximately exact calculations for linear mixed models. *Electronic Journal of Statistics*, **9**, 2293–2323.

Lawley, D.N. and Maxwell, A.E. (1971). *Factor Analysis as a Statistical Methodology*, Second Edition. American Elsevier, New York.

Lenth, Russel V. (2015). The case against normal plots of effects (with discussion). *Journal of Quality Technology*, 47, 91–97.

Levy, Martin S. and Perng, S.K. (1986). An optimal prediction function for the normal linear model. *Journal of the American Statistical Association*, **81**, 196–198.

Lin, Y. and Zhang, H. (2006). Component selection and smoothing in smoothing spline analysis of variance models. *Annals of Statistics*, **34**, 2272–2297.

Loh, Wei-Yin (2011). Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, *1*, 14–23.

Lubischew, Alexander A. (1962). On the use of discriminant functions in taxonomy. *Biometrics*, **18**, 455–477.

Lütkepohl, H. (1985). Comparison of criteria for estimating the order of a vector autoregressive process. *Journal of Time Series Analysis*, **65**, 297–303.

McCulloch, Charles E., Searle, Shayle R., and Neuhaus, John M. (2008). *Generalized, Linear, and Mixed Models*, 2nd Edition. John Wiley and Sones, New York.

McDonald, G. C. and Schwing, R. C. (1973). Instabilities in regression estimates relating air pollution to mortality. *Technometrics*, **15**, 463–481.

McKeon, James J. (1974). F approximations to the distribution of Hotelling's T_0^2. *Biometrika*, **61**, 381–383.

McLeod, A.I. (1977). Improved Box–Jenkins estimators. *Biometrika*, **64**, 531–534.

Mandel, J. (1989a). Some thoughts on variable-selection in multiple regression. *Journal of Quality Technology*, **21**, 2–6.

Mandel, J. (1989b). The nature of collinearity. *Journal of Quality Technology*, **21**, 268–276.

Marden, John I. (2015). *Multivariate Statistics: Old School*. http://statistics.net/Multivariate

Mardia, K.V., Goodall, C. Redfern, E.J., and Alonso, F.J. (1998). The kriged Kalman filter (with discussion). *Test*, **7**, 217–252.

Mardia, K.V., Kent, J.T., and Bibby, J.M. (1979). *Multivariate Analysis*. Academic Press, New York.

Mardia, K.V. and Marshal, R.J. (1984). Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, **71**, 135–146.

Mardia, K.V. and Watkins, A.J. (1989). On multimodality of the likelihood in the spatial linear model. *Biometrika*, **76**, 289–295.

Marquardt, Donald W. (1963). An algorithm for least squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, **2**, 431–441.

Marshall, R.J. and Mardia, K.V. (1985). Minimum norm quadratic estimation of components of spatial covariance. *Journal of the International Association for Mathematical Geology*, **17**, 517–525.

Máté, L. (1990). *Hilbert Space Methods in Science and Engineering*. Taylor & Francis, London.

Matern, Bertil (1986). *Spatial Variation*, Second Edition. Springer-Verlag, New York.

Matheron, G. (1965). Les variable regionalisees et leur estimation: Masson, Paris, xxxp.

Matheron, G. (1969). Le krigeage universal: Fascicule 1, Cahiers du CMMM., 82p.

Matheron, G. (1973). The intrinsic random functions and their applications. *Advances in Applied Probability*, **5**, 439–468.

Meinhold, Richard J. and Singpurwalla, Nozer D. (1983). Understanding the Kalman filter. *The American Statistician*, **37**, 123–127.

Mercer, J. (1909) Functions of positive and negative type and their connection with the theory of integral equations. *Philophical Transactions of the Royal Society of London, A*, 209, 415–446.

Moguerza, Javier M. and Muñoz, Alberto (2006). Support vector machines with applications. *Statistical Science*, **21**, 322–336.

Monahan, John F. (2008). *A Primer on Linear Models.* Chapman & Hall/CRC Press, Boca Raton, FL.

Montgomery, D.C. and Peck, E.A. (1982). *Introduction to Linear Regression Analysis*. John Wiley, New York.

Morrison, Donald F. (2004). *Multivariate Statistical Methods*, Fourth Edition. Duxbury Press, Pacific Grove CA.

Mosteller, Frederick and Tukey, John W. (1977). *Data Analysis and Regression*. Addison–Wesley, Reading, MA.

Muirhead, Robb J. (1982). *Aspects of Multivariate Statistical Theory*. John Wiley and Sons, New York.

Murray, G.D. (1977). A note on the estimation of probability density functions. *Biometrika*, **64**, 150–152.

Murtagh, Fionn and Legendre, Pierre (2014). Ward's hierarchical agglomerative clustering method: Which algorithms implement Ward's criterion? *Journal of Classification*, **31**, 274-295.

Naylor, A. and Sell, G. (1982). *Linear Operator Theory in Engineering and Science*. Springer, New York.

Ng, Andrew (2018). *Machine Learning Yearning* (Draft). deeplearning.ai. www.dbooks.org/machine-learning-yearning-1501/

Nievergelt, Yves (2000). A tutorial history of least squares with applications to astronomy and geodesy. *Journal of Computational and Applied Mathematics*, **121**, 37-72.

Nosedal-Sanchez, A., Storlie, C.B., Lee, T.C.M., Christensen, R. (2012). Reproducing kernel Hilbert spaces for penalized regression: A tutorial. *The American Statistician*, **66**, 50–60

Ogden, R. Todd (1997). *Essential Wavelets for Statistical Applications and Data Analysis*. Birkhäuser, Boston.

generalized linear models. *Journal of the American Statistical Association*, **81**, 96–103.

Panel on Discriminant Analysis, Classification, and Clustering (1989). Discriminant analysis and clustering. *Statistical Science*, **4**, 34–69.

Pearson, Karl (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, **6(2)**, 559–572.

Press, S. James (1982). *Applied Multivariate Analysis: Using Bayesian and Frequentist Methods of Inference*, Second Edition. R.E. Krieger, Malabar, FL. (Latest reprinting, Dover Press, 2005).

Press, S. James and Wilson, S. (1978). Choosing between logistic regression and discriminant analysis. *Journal of the American Statistical Association*, **73**, 699–705.

Ramsay, J. and Silverman, B. (2005). *Functional Data Analysis*. Springer, New York.

Rao, C.R. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society, Series B*, **10**, 159–203.

Rao, C.R. (1952). *Advanced Statistical Methods in Biometric Research*. John Wiley and Sons, New York.

Rao, C.R. (1965). The theory of least squares when the parameters are stochastic and its application to the analysis of growth curves. *Biometrika*, **52**, 447–458.

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

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge.

Reiss, Philip T., Goldsmith, Jeff, Shang, Han Lin and Ogden, R. Todd (2017). Methods for scalar-on-function regression. *International Statistical Review*, 85, 228–249.

Ripley, Brian D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge.

Roy, S.N. (1953). On a heuristic method of test construction and its use in multivariate analysis. *Annals of Mathematical Statistics*, **24**, 220–238.

Roy, S.N. and Bose, R.C. (1953). Simultaneous confidence interval estimation. *Annals of Mathematical Statistics*, **24**, 513–536.

Rustagi, J. (1994). *Optimization Techniques in Statistics*. Academic Press, London.

Satterthwaite, F. E. (1946). An approximate distribution of estimates of variance components. *Biometrics*, **2**, 110–114.

Schabenberger, Oliver and Gotway, Carol A. (2005). *Statistical Methods for Spatial Data Analysis* Chapman & Hall/CRC, Boca Raton, FL.

Schatzoff, M., Tsao, R., and Fienberg, S. (1968). Efficient calculations of all possible regressions. *Technometrics*, **10**, 768-779.

Scheffé, Henry (1959). *The Analysis of Variance*. John Wiley and Sons, New York.

Schervish, Mark J. (1986). A predictive derivation of principal components. Technical Report 378, Department of Statistics, Carnegie-Mellon University, Pittsburgh, PA.

Schwarz, Gideon (1978). Estimating the dimension of a model. *Annals of Statistics*, **16**, 461–464.

Searle, S. R. (1971). *Linear Models*. John Wiley and Sons, New York.

Searle, S. R., Casella, G., and McCulloch, C. (1992). *Variance Components*. John Wiley and Sons, New York.

Seber, G.A.F. (1984). *Multivariate Observations*. John Wiley and Sons, New York.

Seber, G.A.F. and Wild, C.J. (1989, 2003). *Nonlinear Regression*. John Wiley and Sons, New York. (The 2003 version seems to be a reprint of the 1989 version.)

Seely, J. F. and El-Bassiouni, Y. (1983). Applying Wald's variance component test. *The Annals of Statistics*, **11**, 197–201.

Sherman, Michael (2011). *Spatial Statistics and Spatio-Temporal Data: Covariance Functions and Directional Properties*. John Wiley and Sons, New York.

Shumway, Robert H. (1988). *Applied Statistical Time Series Analysis*. Prentice–Hall, Englewood Cliffs, NJ.

Shumway, R.H. and Stoffer, D.S. (1982). An approach to time-series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, **3**, 253–264.

Shumway and Stoffer (2011). *Time Series Analysis and Its Applications: With R Examples,* Third Edition. Springer, New York.

Smith, H., Gnanadesikan, R. and Hughes, J.B. (1962). Multivariate analysis of variance (MANOVA). *Biometrics*, **18**, 22–41.

Stein, Michael L. (1987). Minimum norm quadratic estimation of spatial variograms. *Journal of the American Statistical Association*, **82**, 765–772.

Stein, Michael L. (1988). Asymptotically efficient prediction of a random field with misspecified covariance function. The *Annals of Statistics*, **16**, 55–64.

Stein, Michael L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, New York.

Sugiura, N. (1978). Further analysis of the data by Akaike's information criterion and the finite corrections. *Communications in Statistics, Part A, Theory and Methods*, **7**, 13-26.

Tarpey, Thaddeus, Ogden, R. Todd, Petkova, Eva, and Christensen, Ronald (2014). A paradoxical result in estimating regression coefficients. *The American Statistician*, **68**, 271–276.

Thompson, G.H. (1934). Hotelling's method modified to give Spearman's *g*. *Journal of Educational Psychology*. **25**, 366–374.

Thurstone, L.L. (1931). Multiple factor analysis. *Psychological Review*, **38**, 406–427.

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

Tukey, John W. (1977). *Exploratory Data Analysis*. Addison–Wesley, Reading, MA.

Utts, J. (1982). The rainbow test for lack of fit in regression. *Communications in Statistics—Theory and Methods*, **11**, 2801–2815.

van Houwelingen, J.C. (1988). Use and abuse of variance models in regression. *Biometrics*, **44**, 1073–1081.

Wahba, G. (1990). *Spline Models for Observational Data*. (Vol. 59, CBMS-NSF Regional Conference Series in Applied Mathematics.) SIAM, Philadelphia.

Wainwright, Martin J. (2014). Structured regularizers for high-dimensional problems: Statistical and computational issues. *Annual Review of Statistics and Its Applications*, **1**, 233–53.

Wecker, W. and Ansley, C. (1983). The signal extraction approach to nonlinear regression and spline smoothing. *Journal of the American Statistical Association*, **78**, 81–89.

Weisberg, S. (1985). *Applied Linear Regression*. Second Edition. John Wiley and Sons, New York.

Whittaker, Joe (1990). *Graphical Models in Applied Multivariate Statistics*. John Wiley and Sons, New York.

Whittle, P. (1954). On stationary processes in the plane. *Biometrika*, **41**, 434–449.

Whittle, P. (1963) Stochastic processes in several dimensions. *Bulletin of the International Statistical Institute*, **40(1)**, 974–994.

Williams, E. J. (1959). *Regression Analysis*. John Wiley and Sons, New York.

Williams, J.S. (1979). A synthetic basis for comprehensive factor-analysis theory. *Biometrics*, **35**, 719–733.

Wolfinger, Russell D. (1996). Heterogeneous variance-covariance structures for repeated measures. *Journal of Agricultural, Biological, and Environmental Statitsics*, 1, 205–230.

Wood, S. (2006). *Generalized Additive Models: An Introduction with R*. CRC Press, Boca Raton.

Younger, M. S. (1979). *A Handbook for Linear Regression*, Duxbury Press, Belmont, CA.

Zhang, Chong; Liu, Yufeng; and Wu, Zhengxiao (2013). On the effect and remedies of shrinkage on classification probability estimation, *The American Statistician*, **67**, 134–142.

Zhu, Mu (2008). Kernels and ensembles. *The American Statistician*, **62**, 97–109.

Index