3

Linear Methods for Regression

3.1 Introduction

A linear regression model assumes that the regression function $\mathrm{E}(Y|X)$ is linear in the inputs $X_1, \ldots, X_p$. Linear models were largely developed in the precomputer age of statistics, but even in today's computer era there are still good reasons to study and use them. They are simple and often provide an adequate and interpretable description of how the inputs affect the output. For prediction purposes they can sometimes outperform fancier nonlinear models, especially in situations with small numbers of training cases, low signal-to-noise ratio or sparse data. Finally, linear methods can be applied to transformations of the inputs and this considerably expands their scope. These generalizations are sometimes called basis-function methods, and are discussed in Chapter 5.

In this chapter we describe linear methods for regression, while in the next chapter we discuss linear methods for classification. On some topics we go into considerable detail, as it is our firm belief that an understanding of linear methods is essential for understanding nonlinear ones. In fact, many nonlinear techniques are direct generalizations of the linear methods discussed here.

3.2 Linear Regression Models and Least Squares

As introduced in Chapter 2, we have an input vector $X^T = (X_1, X_2, \ldots, X_p)$, and want to predict a real-valued output Y. The linear regression model has the form

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j. \tag{3.1}$$

The linear model either assumes that the regression function $\mathrm{E}(Y|X)$ is linear, or that the linear model is a reasonable approximation. Here the β_j's are unknown parameters or coefficients, and the variables X_j can come from different sources:

- quantitative inputs;
- transformations of quantitative inputs, such as log, square-root or square;
- basis expansions, such as $X_2 = X_1^2$, $X_3 = X_1^3$, leading to a polynomial representation;
- numeric or "dummy" coding of the levels of qualitative inputs. For example, if G is a five-level factor input, we might create X_j, $j = 1, \ldots, 5$, such that $X_j = I(G = j)$. Together this group of X_j represents the effect of G by a set of level-dependent constants, since in $\sum_{j=1}^{5} X_j \beta_j$, one of the X_js is one, and the others are zero.
- interactions between variables, for example, $X_3 = X_1 \cdot X_2$.

No matter the source of the X_j, the model is linear in the parameters.

Typically we have a set of training data $(x_1, y_1) \ldots (x_N, y_N)$ from which to estimate the parameters β. Each $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T$ is a vector of feature measurements for the ith case. The most popular estimation method is *least squares*, in which we pick the coefficients $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$ to minimize the residual sum of squares

$$\begin{aligned} \mathrm{RSS}(\beta) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \Big)^2. \end{aligned} \tag{3.2}$$

From a statistical point of view, this criterion is reasonable if the training observations (x_i, y_i) represent independent random draws from their population. Even if the x_i's were not drawn randomly, the criterion is still valid if the y_i's are conditionally independent given the inputs x_i. Figure 3.1 illustrates the geometry of least-squares fitting in the $\mathbb{R}^{p+1}$-dimensional

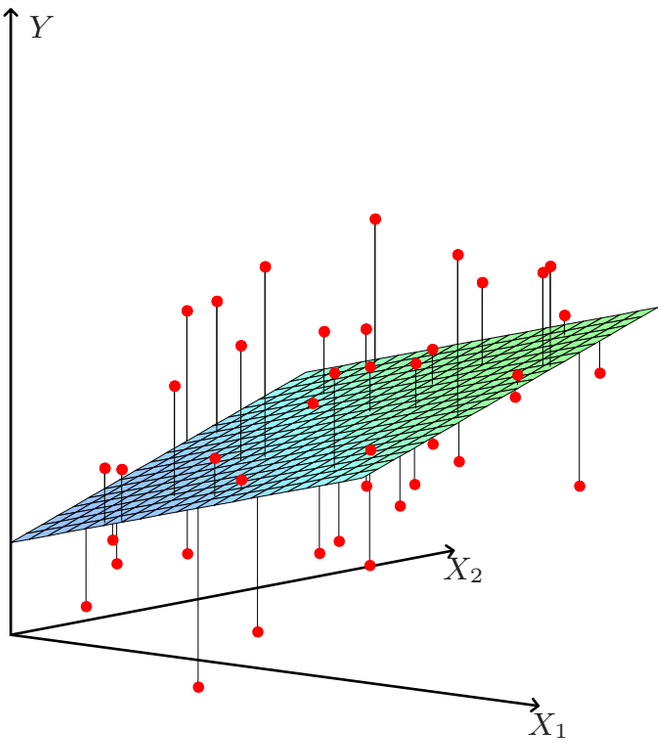

FIGURE 3.1. *Linear least squares fitting with* $X \in \mathbb{R}^2$. *We seek the linear function of* X *that minimizes the sum of squared residuals from* Y.

space occupied by the pairs (X, Y). Note that (3.2) makes no assumptions about the validity of model (3.1); it simply finds the best linear fit to the data. Least squares fitting is intuitively satisfying no matter how the data arise; the criterion measures the average lack of fit.

How do we minimize (3.2)? Denote by $\mathbf{X}$ the $N \times (p+1)$ matrix with each row an input vector (with a 1 in the first position), and similarly let $\mathbf{y}$ be the N-vector of outputs in the training set. Then we can write the residual sum-of-squares as

$$\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta). \tag{3.3}$$

This is a quadratic function in the $p+1$ parameters. Differentiating with respect to β we obtain

$$\begin{aligned} \frac{\partial \text{RSS}}{\partial \beta} &= -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) \\ \frac{\partial^2 \text{RSS}}{\partial \beta \partial \beta^T} &= 2\mathbf{X}^T\mathbf{X}. \end{aligned} \tag{3.4}$$

Assuming (for the moment) that $\mathbf{X}$ has full column rank, and hence $\mathbf{X}^T\mathbf{X}$ is positive definite, we set the first derivative to zero

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0 \tag{3.5}$$

to obtain the unique solution

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.6}$$

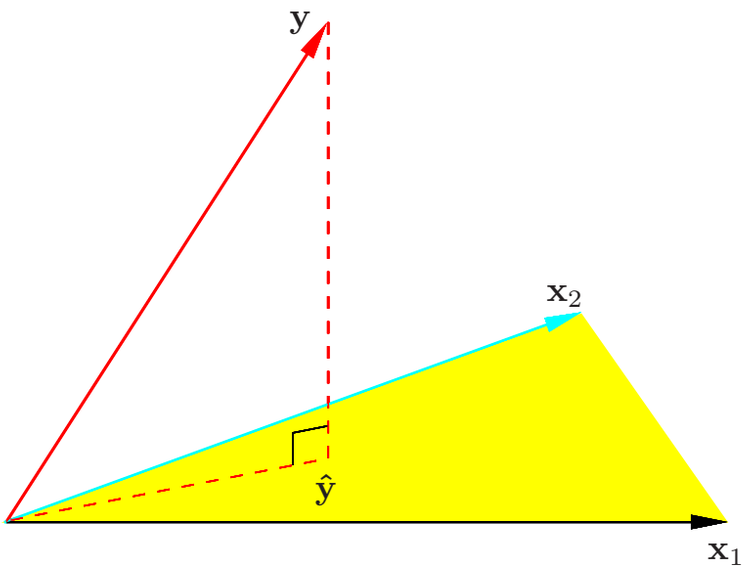

FIGURE 3.2. *The N-dimensional geometry of least squares regression with two predictors. The outcome vector* $\mathbf{y}$ *is orthogonally projected onto the hyperplane spanned by the input vectors* $\mathbf{x}_1$ *and* $\mathbf{x}_2$*. The projection* $\hat{\mathbf{y}}$ *represents the vector of the least squares predictions*

The predicted values at an input vector x_0 are given by $\hat{f}(x_0) = (1 : x_0)^T\hat{\beta}$; the fitted values at the training inputs are

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}, \tag{3.7}$$

where $\hat{y}_i = \hat{f}(x_i)$. The matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ appearing in equation (3.7) is sometimes called the "hat" matrix because it puts the hat on $\mathbf{y}$.

Figure 3.2 shows a different geometrical representation of the least squares estimate, this time in $\mathbb{R}^N$. We denote the column vectors of $\mathbf{X}$ by $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_p$, with $\mathbf{x}_0 \equiv 1$. For much of what follows, this first column is treated like any other. These vectors span a subspace of $\mathbb{R}^N$, also referred to as the column space of $\mathbf{X}$. We minimize $\mathrm{RSS}(\beta) = \|\mathbf{y} - \mathbf{X}\beta\|^2$ by choosing $\hat{\beta}$ so that the residual vector $\mathbf{y} - \hat{\mathbf{y}}$ is orthogonal to this subspace. This orthogonality is expressed in (3.5), and the resulting estimate $\hat{\mathbf{y}}$ is hence the *orthogonal projection* of $\mathbf{y}$ onto this subspace. The hat matrix $\mathbf{H}$ computes the orthogonal projection, and hence it is also known as a projection matrix.

It might happen that the columns of $\mathbf{X}$ are not linearly independent, so that $\mathbf{X}$ is not of full rank. This would occur, for example, if two of the inputs were perfectly correlated, (e.g., $\mathbf{x}_2 = 3\mathbf{x}_1$). Then $\mathbf{X}^T\mathbf{X}$ is singular and the least squares coefficients $\hat{\beta}$ are not uniquely defined. However, the fitted values $\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}$ are still the projection of $\mathbf{y}$ onto the column space of $\mathbf{X}$; there is just more than one way to express that projection in terms of the column vectors of $\mathbf{X}$. The non-full-rank case occurs most often when one or more qualitative inputs are coded in a redundant fashion. There is usually a natural way to resolve the non-unique representation, by recoding and/or dropping redundant columns in $\mathbf{X}$. Most regression software packages detect these redundancies and automatically implement

some strategy for removing them. Rank deficiencies can also occur in signal and image analysis, where the number of inputs p can exceed the number of training cases N. In this case, the features are typically reduced by filtering or else the fitting is controlled by regularization (Section 5.2.3 and Chapter 18).

Up to now we have made minimal assumptions about the true distribution of the data. In order to pin down the sampling properties of $\hat{\beta}$, we now assume that the observations y_i are uncorrelated and have constant variance σ^2, and that the x_i are fixed (non random). The variance–covariance matrix of the least squares parameter estimates is easily derived from (3.6) and is given by

$$\text{Var}(\hat{\beta}) = (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2. \tag{3.8}$$

Typically one estimates the variance σ^2 by

$$\hat{\sigma}^2 = \frac{1}{N-p-1}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2.$$

The $N-p-1$ rather than N in the denominator makes $\hat{\sigma}^2$ an unbiased estimate of σ^2: $\text{E}(\hat{\sigma}^2) = \sigma^2$.

To draw inferences about the parameters and the model, additional assumptions are needed. We now assume that (3.1) is the correct model for the mean; that is, the conditional expectation of Y is linear in $X_1, \ldots, X_p$. We also assume that the deviations of Y around its expectation are additive and Gaussian. Hence

$$\begin{aligned} Y &= \text{E}(Y|X_1, \ldots, X_p) + \varepsilon \\ &= \beta_0 + \sum_{j=1}^{p} X_j\beta_j + \varepsilon, \end{aligned} \tag{3.9}$$

where the error ε is a Gaussian random variable with expectation zero and variance σ^2, written $\varepsilon \sim N(0, \sigma^2)$.

Under (3.9), it is easy to show that

$$\hat{\beta} \sim N(\beta, (\mathbf{X}^T\mathbf{X})^{-1}\sigma^2). \tag{3.10}$$

This is a multivariate normal distribution with mean vector and variance–covariance matrix as shown. Also

$$(N-p-1)\hat{\sigma}^2 \sim \sigma^2\chi^2_{N-p-1}, \tag{3.11}$$

a chi-squared distribution with $N-p-1$ degrees of freedom. In addition $\hat{\beta}$ and $\hat{\sigma}^2$ are statistically independent. We use these distributional properties to form tests of hypothesis and confidence intervals for the parameters β_j.

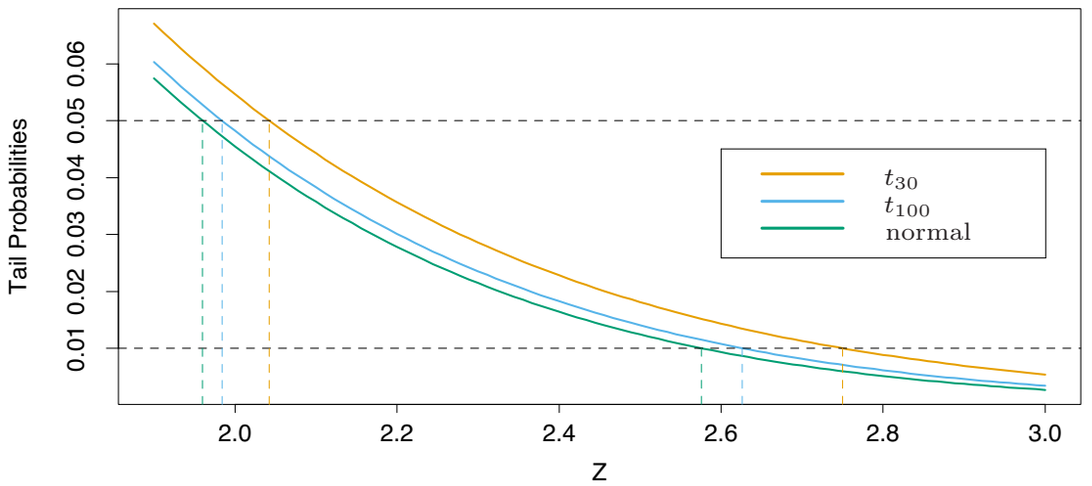

FIGURE 3.3. *The tail probabilities* $\Pr(|Z| > z)$ *for three distributions,* t_{30}*,* t_{100} *and standard normal. Shown are the appropriate quantiles for testing significance at the* $p = 0.05$ *and* 0.01 *levels. The difference between* t *and the standard normal becomes negligible for* N *bigger than about* 100*.*

To test the hypothesis that a particular coefficient $\beta_j = 0$, we form the standardized coefficient or *Z-score*

$$z_j = \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{v_j}}, \tag{3.12}$$

where v_j is the jth diagonal element of $(\mathbf{X}^T\mathbf{X})^{-1}$. Under the null hypothesis that $\beta_j = 0$, z_j is distributed as t_{N-p-1} (a t distribution with $N - p - 1$ degrees of freedom), and hence a large (absolute) value of z_j will lead to rejection of this null hypothesis. If $\hat{\sigma}$ is replaced by a known value σ, then z_j would have a standard normal distribution. The difference between the tail quantiles of a t-distribution and a standard normal become negligible as the sample size increases, and so we typically use the normal quantiles (see Figure 3.3).

Often we need to test for the significance of groups of coefficients simultaneously. For example, to test if a categorical variable with k levels can be excluded from a model, we need to test whether the coefficients of the dummy variables used to represent the levels can all be set to zero. Here we use the F statistic,

$$F = \frac{(\mathrm{RSS}_0 - \mathrm{RSS}_1)/(p_1 - p_0)}{\mathrm{RSS}_1/(N - p_1 - 1)}, \tag{3.13}$$

where RSS_1 is the residual sum-of-squares for the least squares fit of the bigger model with $p_1 + 1$ parameters, and RSS_0 the same for the nested smaller model with $p_0 + 1$ parameters, having $p_1 - p_0$ parameters constrained to be

zero. The F statistic measures the change in residual sum-of-squares per additional parameter in the bigger model, and it is normalized by an estimate of σ^2. Under the Gaussian assumptions, and the null hypothesis that the smaller model is correct, the F statistic will have a $F_{p_1-p_0,N-p_1-1}$ distribution. It can be shown (Exercise 3.1) that the z_j in (3.12) are equivalent to the F statistic for dropping the single coefficient β_j from the model. For large N, the quantiles of $F_{p_1-p_0,N-p_1-1}$ approach those of $\chi^2_{p_1-p_0}/(p_1-p_0)$.

Similarly, we can isolate β_j in (3.10) to obtain a $1-2\alpha$ confidence interval for β_j:

$$(\hat{\beta}_j - z^{(1-\alpha)} v_j^{\frac{1}{2}} \hat{\sigma}, \quad \hat{\beta}_j + z^{(1-\alpha)} v_j^{\frac{1}{2}} \hat{\sigma}). \tag{3.14}$$

Here $z^{(1-\alpha)}$ is the $1-\alpha$ percentile of the normal distribution:

$$\begin{aligned} z^{(1-0.025)} &= 1.96, \\ z^{(1-.05)} &= 1.645, \quad \text{etc.} \end{aligned}$$

Hence the standard practice of reporting $\hat{\beta} \pm 2 \cdot \text{se}(\hat{\beta})$ amounts to an approximate 95% confidence interval. Even if the Gaussian error assumption does not hold, this interval will be approximately correct, with its coverage approaching $1-2\alpha$ as the sample size $N \to \infty$.

In a similar fashion we can obtain an approximate confidence set for the entire parameter vector β, namely

$$C_\beta = \{\beta | (\hat{\beta} - \beta)^T \mathbf{X}^T \mathbf{X} (\hat{\beta} - \beta) \leq \hat{\sigma}^2 {\chi^2_{p+1}}^{(1-\alpha)}\}, \tag{3.15}$$

where $\chi^{2(1-\alpha)}_\ell$ is the $1-\alpha$ percentile of the chi-squared distribution on ℓ degrees of freedom: for example, $\chi^{2(1-0.05)}_5 = 11.1$, $\chi^{2(1-0.1)}_5 = 9.2$. This confidence set for β generates a corresponding confidence set for the true function $f(x) = x^T\beta$, namely $\{x^T\beta | \beta \in C_\beta\}$ (Exercise 3.2; see also Figure 5.4 in Section 5.2.2 for examples of confidence bands for functions).

3.2.1 Example: Prostate Cancer

The data for this example come from a study by Stamey et al. (1989). They examined the correlation between the level of prostate-specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy. The variables are log cancer volume (`lcavol`), log prostate weight (`lweight`), `age`, log of the amount of benign prostatic hyperplasia (`lbph`), seminal vesicle invasion (`svi`), log of capsular penetration (`lcp`), Gleason score (`gleason`), and percent of Gleason scores 4 or 5 (`pgg45`). The correlation matrix of the predictors given in Table 3.1 shows many strong correlations. Figure 1.1 (page 3) of Chapter 1 is a scatterplot matrix showing every pairwise plot between the variables. We see that `svi` is a binary variable, and `gleason` is an ordered categorical variable. We see, for

TABLE 3.1. *Correlations of predictors in the prostate cancer data.*

	lcavol	lweight	age	lbph	svi	lcp	gleason
lweight	0.300						
age	0.286	0.317					
lbph	0.063	0.437	0.287				
svi	0.593	0.181	0.129	−0.139			
lcp	0.692	0.157	0.173	−0.089	0.671		
gleason	0.426	0.024	0.366	0.033	0.307	0.476	
pgg45	0.483	0.074	0.276	−0.030	0.481	0.663	0.757

TABLE 3.2. *Linear model fit to the prostate cancer data. The Z score is the coefficient divided by its standard error (3.12). Roughly a Z score larger than two in absolute value is significantly nonzero at the* $p = 0.05$ *level.*

Term	Coefficient	Std. Error	Z Score
Intercept	2.46	0.09	27.60
lcavol	0.68	0.13	5.37
lweight	0.26	0.10	2.75
age	−0.14	0.10	−1.40
lbph	0.21	0.10	2.06
svi	0.31	0.12	2.47
lcp	−0.29	0.15	−1.87
gleason	−0.02	0.15	−0.15
pgg45	0.27	0.15	1.74

example, that both lcavol and lcp show a strong relationship with the response lpsa, and with each other. We need to fit the effects jointly to untangle the relationships between the predictors and the response.

We fit a linear model to the log of prostate-specific antigen, lpsa, after first standardizing the predictors to have unit variance. We randomly split the dataset into a training set of size 67 and a test set of size 30. We applied least squares estimation to the training set, producing the estimates, standard errors and Z-scores shown in Table 3.2. The Z-scores are defined in (3.12), and measure the effect of dropping that variable from the model. A Z-score greater than 2 in absolute value is approximately significant at the 5% level. (For our example, we have nine parameters, and the 0.025 tail quantiles of the t_{67-9} distribution are ± 2.002!) The predictor lcavol shows the strongest effect, with lweight and svi also strong. Notice that lcp is not significant, once lcavol is in the model (when used in a model without lcavol, lcp is strongly significant). We can also test for the exclusion of a number of terms at once, using the F-statistic (3.13). For example, we consider dropping all the non-significant terms in Table 3.2, namely age,

`lcp`, `gleason`, and `pgg45`. We get

$$F = \frac{(32.81 - 29.43)/(9-5)}{29.43/(67-9)} = 1.67, \tag{3.16}$$

which has a p-value of 0.17 ($\Pr(F_{4,58} > 1.67) = 0.17$), and hence is not significant.

The mean prediction error on the test data is 0.521. In contrast, prediction using the mean training value of `lpsa` has a test error of 1.057, which is called the "base error rate." Hence the linear model reduces the base error rate by about 50%. We will return to this example later to compare various selection and shrinkage methods.

3.2.2 The Gauss–Markov Theorem

One of the most famous results in statistics asserts that the least squares estimates of the parameters β have the smallest variance among all linear unbiased estimates. We will make this precise here, and also make clear that the restriction to unbiased estimates is not necessarily a wise one. This observation will lead us to consider biased estimates such as ridge regression later in the chapter. We focus on estimation of any linear combination of the parameters $\theta = a^T\beta$; for example, predictions $f(x_0) = x_0^T\beta$ are of this form. The least squares estimate of $a^T\beta$ is

$$\hat{\theta} = a^T\hat{\beta} = a^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.17}$$

Considering $\mathbf{X}$ to be fixed, this is a linear function $\mathbf{c}_0^T\mathbf{y}$ of the response vector $\mathbf{y}$. If we assume that the linear model is correct, $a^T\hat{\beta}$ is unbiased since

$$\begin{aligned} \mathrm{E}(a^T\hat{\beta}) &= \mathrm{E}(a^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}) \\ &= a^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\beta \\ &= a^T\beta. \end{aligned} \tag{3.18}$$

The Gauss–Markov theorem states that if we have any other linear estimator $\tilde{\theta} = \mathbf{c}^T\mathbf{y}$ that is unbiased for $a^T\beta$, that is, $\mathrm{E}(\mathbf{c}^T\mathbf{y}) = a^T\beta$, then

$$\mathrm{Var}(a^T\hat{\beta}) \le \mathrm{Var}(\mathbf{c}^T\mathbf{y}). \tag{3.19}$$

The proof (Exercise 3.3) uses the triangle inequality. For simplicity we have stated the result in terms of estimation of a single parameter $a^T\beta$, but with a few more definitions one can state it in terms of the entire parameter vector β (Exercise 3.3).

Consider the mean squared error of an estimator $\tilde{\theta}$ in estimating θ:

$$\begin{aligned} \mathrm{MSE}(\tilde{\theta}) &= \mathrm{E}(\tilde{\theta} - \theta)^2 \\ &= \mathrm{Var}(\tilde{\theta}) + [\mathrm{E}(\tilde{\theta}) - \theta]^2. \end{aligned} \tag{3.20}$$

The first term is the variance, while the second term is the squared bias. The Gauss-Markov theorem implies that the least squares estimator has the smallest mean squared error of all linear estimators with no bias. However, there may well exist a biased estimator with smaller mean squared error. Such an estimator would trade a little bias for a larger reduction in variance. Biased estimates are commonly used. Any method that shrinks or sets to zero some of the least squares coefficients may result in a biased estimate. We discuss many examples, including variable subset selection and ridge regression, later in this chapter. From a more pragmatic point of view, most models are distortions of the truth, and hence are biased; picking the right model amounts to creating the right balance between bias and variance. We go into these issues in more detail in Chapter 7.

Mean squared error is intimately related to prediction accuracy, as discussed in Chapter 2. Consider the prediction of the new response at input x_0,

$$Y_0 = f(x_0) + \varepsilon_0. \tag{3.21}$$

Then the expected prediction error of an estimate $\tilde{f}(x_0) = x_0^T\tilde{\beta}$ is

$$\begin{aligned} \mathrm{E}(Y_0 - \tilde{f}(x_0))^2 &= \sigma^2 + \mathrm{E}(x_0^T\tilde{\beta} - f(x_0))^2 \\ &= \sigma^2 + \mathrm{MSE}(\tilde{f}(x_0)). \end{aligned} \tag{3.22}$$

Therefore, expected prediction error and mean squared error differ only by the constant σ^2, representing the variance of the new observation y_0.

3.2.3 Multiple Regression from Simple Univariate Regression

The linear model (3.1) with $p > 1$ inputs is called the *multiple linear regression model*. The least squares estimates (3.6) for this model are best understood in terms of the estimates for the *univariate* ($p = 1$) linear model, as we indicate in this section.

Suppose first that we have a univariate model with no intercept, that is,

$$Y = X\beta + \varepsilon. \tag{3.23}$$

The least squares estimate and residuals are

$$\begin{aligned} \hat{\beta} &= \frac{\sum_1^N x_i y_i}{\sum_1^N x_i^2}, \\ r_i &= y_i - x_i\hat{\beta}. \end{aligned} \tag{3.24}$$

In convenient vector notation, we let $\mathbf{y} = (y_1, \ldots, y_N)^T$, $\mathbf{x} = (x_1, \ldots, x_N)^T$ and define

$$\begin{aligned} \langle \mathbf{x}, \mathbf{y} \rangle &= \sum_{i=1}^N x_i y_i, \\ &= \mathbf{x}^T\mathbf{y}, \end{aligned} \tag{3.25}$$

the *inner product* between $\mathbf{x}$ and $\mathbf{y}$[1]. Then we can write

$$\begin{aligned}\hat{\beta} &= \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle}, \\ \mathbf{r} &= \mathbf{y} - \mathbf{x}\hat{\beta}.\end{aligned} \tag{3.26}$$

As we will see, this simple univariate regression provides the building block for multiple linear regression. Suppose next that the inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ (the columns of the data matrix $\mathbf{X}$) are orthogonal; that is $\langle \mathbf{x}_j, \mathbf{x}_k \rangle = 0$ for all $j \neq k$. Then it is easy to check that the multiple least squares estimates $\hat{\beta}_j$ are equal to $\langle \mathbf{x}_j, \mathbf{y} \rangle / \langle \mathbf{x}_j, \mathbf{x}_j \rangle$—the univariate estimates. In other words, when the inputs are orthogonal, they have no effect on each other's parameter estimates in the model.

Orthogonal inputs occur most often with balanced, designed experiments (where orthogonality is enforced), but almost never with observational data. Hence we will have to orthogonalize them in order to carry this idea further. Suppose next that we have an intercept and a single input $\mathbf{x}$. Then the least squares coefficient of $\mathbf{x}$ has the form

$$\hat{\beta}_1 = \frac{\langle \mathbf{x} - \bar{x}\mathbf{1}, \mathbf{y} \rangle}{\langle \mathbf{x} - \bar{x}\mathbf{1}, \mathbf{x} - \bar{x}\mathbf{1} \rangle}, \tag{3.27}$$

where $\bar{x} = \sum_i x_i / N$, and $\mathbf{1} = \mathbf{x}_0$, the vector of N ones. We can view the estimate (3.27) as the result of two applications of the simple regression (3.26). The steps are:

1. regress $\mathbf{x}$ on $\mathbf{1}$ to produce the residual $\mathbf{z} = \mathbf{x} - \bar{x}\mathbf{1}$;

2. regress $\mathbf{y}$ on the residual $\mathbf{z}$ to give the coefficient $\hat{\beta}_1$.

In this procedure, "regress $\mathbf{b}$ on $\mathbf{a}$" means a simple univariate regression of $\mathbf{b}$ on $\mathbf{a}$ with no intercept, producing coefficient $\hat{\gamma} = \langle \mathbf{a}, \mathbf{b} \rangle / \langle \mathbf{a}, \mathbf{a} \rangle$ and residual vector $\mathbf{b} - \hat{\gamma}\mathbf{a}$. We say that $\mathbf{b}$ is adjusted for $\mathbf{a}$, or is "orthogonalized" with respect to $\mathbf{a}$.

Step 1 orthogonalizes $\mathbf{x}$ with respect to $\mathbf{x}_0 = \mathbf{1}$. Step 2 is just a simple univariate regression, using the orthogonal predictors $\mathbf{1}$ and $\mathbf{z}$. Figure 3.4 shows this process for two general inputs $\mathbf{x}_1$ and $\mathbf{x}_2$. The orthogonalization does not change the subspace spanned by $\mathbf{x}_1$ and $\mathbf{x}_2$, it simply produces an orthogonal basis for representing it.

This recipe generalizes to the case of p inputs, as shown in Algorithm 3.1. Note that the inputs $\mathbf{z}_0, \ldots, \mathbf{z}_{j-1}$ in step 2 are orthogonal, hence the simple regression coefficients computed there are in fact also the multiple regression coefficients.

---

[1]The inner-product notation is suggestive of generalizations of linear regression to different metric spaces, as well as to probability spaces.

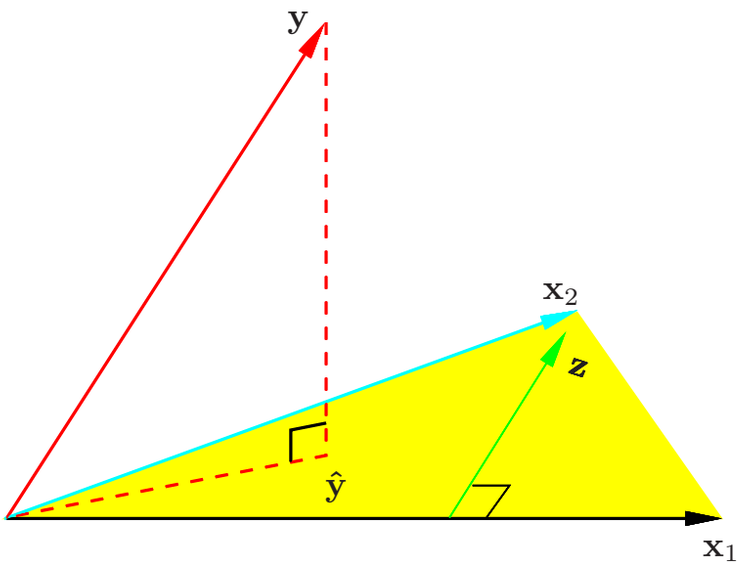

FIGURE 3.4. *Least squares regression by orthogonalization of the inputs. The vector* $\mathbf{x}_2$ *is regressed on the vector* $\mathbf{x}_1$*, leaving the residual vector* $\mathbf{z}$*. The regression of* $\mathbf{y}$ *on* $\mathbf{z}$ *gives the multiple regression coefficient of* $\mathbf{x}_2$*. Adding together the projections of* $\mathbf{y}$ *on each of* $\mathbf{x}_1$ *and* $\mathbf{z}$ *gives the least squares fit* $\hat{\mathbf{y}}$*.*

Algorithm 3.1 *Regression by Successive Orthogonalization.*

1. Initialize $\mathbf{z}_0 = \mathbf{x}_0 = \mathbf{1}$.
2. For $j = 1, 2, \ldots, p$

   Regress $\mathbf{x}_j$ on $\mathbf{z}_0, \mathbf{z}_1, \ldots, , \mathbf{z}_{j-1}$ to produce coefficients $\hat{\gamma}_{\ell j} = \langle \mathbf{z}_\ell, \mathbf{x}_j \rangle / \langle \mathbf{z}_\ell, \mathbf{z}_\ell \rangle$, $\ell = 0, \ldots, j-1$ and residual vector $\mathbf{z}_j = \mathbf{x}_j - \sum_{k=0}^{j-1} \hat{\gamma}_{kj} \mathbf{z}_k$.
3. Regress $\mathbf{y}$ on the residual $\mathbf{z}_p$ to give the estimate $\hat{\beta}_p$.

The result of this algorithm is

$$\hat{\beta}_p = \frac{\langle \mathbf{z}_p, \mathbf{y} \rangle}{\langle \mathbf{z}_p, \mathbf{z}_p \rangle}. \tag{3.28}$$

Re-arranging the residual in step 2, we can see that each of the $\mathbf{x}_j$ is a linear combination of the $\mathbf{z}_k$, $k \leq j$. Since the $\mathbf{z}_j$ are all orthogonal, they form a basis for the column space of $\mathbf{X}$, and hence the least squares projection onto this subspace is $\hat{\mathbf{y}}$. Since $\mathbf{z}_p$ alone involves $\mathbf{x}_p$ (with coefficient 1), we see that the coefficient (3.28) is indeed the multiple regression coefficient of $\mathbf{y}$ on $\mathbf{x}_p$. This key result exposes the effect of correlated inputs in multiple regression. Note also that by rearranging the $\mathbf{x}_j$, any one of them could be in the last position, and a similar results holds. Hence stated more generally, we have shown that the jth multiple regression coefficient is the univariate regression coefficient of $\mathbf{y}$ on $\mathbf{x}_{j \cdot 012\ldots(j-1)(j+1)\ldots,p}$, the residual after regressing $\mathbf{x}_j$ on $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{j-1}, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_p$:

The multiple regression coefficient $\hat{\beta}_j$ represents the additional contribution of $\mathbf{x}_j$ on $\mathbf{y}$, after $\mathbf{x}_j$ has been adjusted for $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{j-1}, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_p$.

If $\mathbf{x}_p$ is highly correlated with some of the other $\mathbf{x}_k$'s, the residual vector $\mathbf{z}_p$ will be close to zero, and from (3.28) the coefficient $\hat{\beta}_p$ will be very unstable. This will be true for all the variables in the correlated set. In such situations, we might have all the Z-scores (as in Table 3.2) be small—any one of the set can be deleted—yet we cannot delete them all. From (3.28) we also obtain an alternate formula for the variance estimates (3.8),

$$\mathrm{Var}(\hat{\beta}_p) = \frac{\sigma^2}{\langle \mathbf{z}_p, \mathbf{z}_p \rangle} = \frac{\sigma^2}{\|\mathbf{z}_p\|^2}. \tag{3.29}$$

In other words, the precision with which we can estimate $\hat{\beta}_p$ depends on the length of the residual vector $\mathbf{z}_p$; this represents how much of $\mathbf{x}_p$ is unexplained by the other $\mathbf{x}_k$'s.

Algorithm 3.1 is known as the *Gram–Schmidt* procedure for multiple regression, and is also a useful numerical strategy for computing the estimates. We can obtain from it not just $\hat{\beta}_p$, but also the entire multiple least squares fit, as shown in Exercise 3.4.

We can represent step 2 of Algorithm 3.1 in matrix form:

$$\mathbf{X} = \mathbf{Z}\mathbf{\Gamma}, \tag{3.30}$$

where $\mathbf{Z}$ has as columns the $\mathbf{z}_j$ (in order), and $\mathbf{\Gamma}$ is the upper triangular matrix with entries $\hat{\gamma}_{kj}$. Introducing the diagonal matrix $\mathbf{D}$ with jth diagonal entry $D_{jj} = \|\mathbf{z}_j\|$, we get

$$\begin{aligned} \mathbf{X} &= \mathbf{Z}\mathbf{D}^{-1}\mathbf{D}\mathbf{\Gamma} \\ &= \mathbf{Q}\mathbf{R}, \end{aligned} \tag{3.31}$$

the so-called QR decomposition of $\mathbf{X}$. Here $\mathbf{Q}$ is an $N \times (p+1)$ orthogonal matrix, $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$, and $\mathbf{R}$ is a $(p+1) \times (p+1)$ upper triangular matrix.

The $\mathbf{QR}$ decomposition represents a convenient orthogonal basis for the column space of $\mathbf{X}$. It is easy to see, for example, that the least squares solution is given by

$$\hat{\beta} = \mathbf{R}^{-1}\mathbf{Q}^T\mathbf{y}, \tag{3.32}$$

$$\hat{\mathbf{y}} = \mathbf{Q}\mathbf{Q}^T\mathbf{y}. \tag{3.33}$$

Equation (3.32) is easy to solve because $\mathbf{R}$ is upper triangular (Exercise 3.4).

3.2.4 Multiple Outputs

Suppose we have multiple outputs $Y_1, Y_2, \ldots, Y_K$ that we wish to predict from our inputs $X_0, X_1, X_2, \ldots, X_p$. We assume a linear model for each output

$$\begin{aligned} Y_k &= \beta_{0k} + \sum_{j=1}^{p} X_j \beta_{jk} + \varepsilon_k \qquad (3.34) \\ &= f_k(X) + \varepsilon_k. \qquad (3.35) \end{aligned}$$

With N training cases we can write the model in matrix notation

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}. \qquad (3.36)$$

Here $\mathbf{Y}$ is the $N \times K$ response matrix, with ik entry y_{ik}, $\mathbf{X}$ is the $N \times (p+1)$ input matrix, $\mathbf{B}$ is the $(p+1) \times K$ matrix of parameters and $\mathbf{E}$ is the $N \times K$ matrix of errors. A straightforward generalization of the univariate loss function (3.2) is

$$\begin{aligned} \mathrm{RSS}(\mathbf{B}) &= \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f_k(x_i))^2 \qquad (3.37) \\ &= \mathrm{tr}[(\mathbf{Y} - \mathbf{XB})^T (\mathbf{Y} - \mathbf{XB})]. \qquad (3.38) \end{aligned}$$

The least squares estimates have exactly the same form as before

$$\hat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \qquad (3.39)$$

Hence the coefficients for the kth outcome are just the least squares estimates in the regression of $\mathbf{y}_k$ on $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_p$. Multiple outputs do not affect one another's least squares estimates.

If the errors $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_K)$ in (3.34) are correlated, then it might seem appropriate to modify (3.37) in favor of a multivariate version. Specifically, suppose $\mathrm{Cov}(\varepsilon) = \mathbf{\Sigma}$, then the multivariate weighted criterion

$$\mathrm{RSS}(\mathbf{B}; \mathbf{\Sigma}) = \sum_{i=1}^{N} (y_i - f(x_i))^T \mathbf{\Sigma}^{-1} (y_i - f(x_i)) \qquad (3.40)$$

arises naturally from multivariate Gaussian theory. Here $f(x)$ is the vector function $(f_1(x), \ldots, f_K(x))^T$, and y_i the vector of K responses for observation i. However, it can be shown that again the solution is given by (3.39); K separate regressions that ignore the correlations (Exercise 3.11). If the $\mathbf{\Sigma}_i$ vary among observations, then this is no longer the case, and the solution for $\mathbf{B}$ no longer decouples.

In Section 3.7 we pursue the multiple outcome problem, and consider situations where it does pay to combine the regressions.