The goal is to understand in more detail the role of prediction error in model selection and assessment, and to study the notion of bias and variance trade-off. We assume that we have a set of training data \( T \), typically \( N \) instances \((x_1, y_1), \ldots, (x_N, y_N)\), where \( x \) is usually a vector of features and \( y \) a response, either continuous or discrete. We further assume that we have a fitted function \( \hat{f}(\cdot) \), which is a rule that has been constructed from \( T \), that maps a vector \( x \) to a value \( y \). In this notation the dependence of \( \hat{f}(\cdot) \) on \( T \) is suppressed. In most applications, \( \hat{f}(\cdot) \) is obtained by minimizing

\[
\sum_{i=1}^{N} L(y_i, f(x_i))
\]

where \( L(Y, f(X)) \) is a loss function, and the minimum is calculated over some class of functions \( \{f\} \) to be specified.

Common choices of loss functions are squared error loss, absolute error, and more rarely losses based on the \( p \)th norm, for continuous responses \( y \) and especially for models where \( f(x) = E(Y \mid x) \). For discrete responses a loss function that counts misclassifications is quite common. The loss function that leads to maximum likelihood estimation is \(-2 \log \Pr(Y; \theta)\) or more generally \(-2 \log \Pr(Y; f(\cdot))\).

In considering errors made in using \( \hat{f}(\cdot) \) for prediction, or more specifically on test data, two quantities of interest are

\[
\text{Err}_T = E\{L(Y, \hat{f}(X)) \mid T\},
\]

\[
\text{Err} = E\{\text{Err}_T\}
\]

where the expectation in (??) is over the joint distribution of \( (Y, X) \), conditional on the training data \( T \), and the expectation in (??) is over the training data. \( \text{Err}_T \) is called test error, prediction error or generalization error, and \( \text{Err} \) is called expected prediction/test error. Loosely speaking, \( \text{Err}_T \) is relevant to model selection: how well will a range of models to the training data serve to predict new data?, whereas \( \text{Err} \) is relevant to model assessment: how well will a range of possible models work for a range of applications? However, \( \text{Err}_T \) is more difficult to analyse theoretically than \( \text{Err} \). Of course neither of these measures can address errors in prediction due to changes in the underlying structure of the data or the errors; the test data must have something in common with the training data in order that the training data be useful for prediction.

**Example: additive errors.** Suppose the model is \( Y = f(X) + \epsilon \), where \( E(\epsilon) = 0 \), \( \text{var}(\epsilon) = \sigma_\epsilon^2 \). It is natural in this setting to use squared error loss, and to condition on the feature variables. Thus we compute, as in §7.3,

\[
\text{Err}(x_0) = E\{(Y - \hat{f}(x_0))^2 \mid X = x_0\} = \sigma_\epsilon^2 + \{E(\hat{f}(x_0)) - f(x_0)\}^2 + \text{var}\{f(x_0)\},
\]
where now the expectation is over $Y$ and $y_1, \ldots, y_N$, with $X = x_0$ and $x_1, \ldots, x_N$ fixed. To verify this the RHS of the first line is expanded, after adding and subtracting $E(Y \mid x_0) = f(x_0)$. The first term is “irreducible error”, reflecting the fact that even if we knew $f(x_0)$, we would not know $Y$. To get further with this formula we need to say more about $\hat{f}(\cdot)$. Suppose, for example, that $\hat{f}(x_0)$ is estimated by the average of the $y$-values for $k$ nearest neighbours of $x_0$ in feature space, i.e.

$$\hat{f}(x_0) = \frac{1}{k} \sum_{\ell=1}^{k} y(\ell),$$

where $|x(1) - x_0| < |x(2) - x_0| < \ldots < |x(N) - x_0|$, using some distance measure in feature space, and $y(\ell)$ is the response for input $x(\ell)$. In this case we have

$$\text{Err}(x_0) = \{f(x_0) - \frac{1}{k} \sum_{\ell=1}^{k} f(x(\ell))\}^2 + \frac{\sigma^2}{k}.$$

We can see that the third term will decrease as $k$ increases, and it seems plausible that the second term will increase with $k$, although this will depend on how quickly $f(\cdot)$ is changing near $x_0$, and how far away the $x(\ell)$ points are relative to the change in $f(\cdot)$.

**Example: Least Squares.** Suppose now that $f(x_0) = x_0^T \beta$, where $\beta$ is $p \times 1$, and we estimate $\beta$ by least squares. Then $\hat{f}(x_0) = x_0^T \beta = f(x_0)$, and

$$\text{var}(\hat{f}(x_0)) = x_0^T (X^T X)^{-1} x_0 \sigma^2,$$

where $X$ is the $N \times p$ feature matrix in the training data. Then

$$\text{Err}(x_0) = \sigma^2 \{1 + x_0 (X^T X)^{-1} x_0\}, \quad (3)$$

the prediction error for a new $Y$ at $x_0$. The bias term is zero. If we further simplify the model by assuming that there is only a single feature, plus a constant term, then

$$(X^T X)^{-1} = \left(\begin{array}{cc} \frac{1}{n} \sum x_i^2 & -\bar{x} \\ -\bar{x} & 1 \end{array}\right) \frac{1}{\sum (x_i - \bar{x})^2}$$

and

$$(1 \quad x_0)(X^T X)^{-1} \left(\begin{array}{c} 1 \\ x_0 \end{array}\right) = \frac{1}{n} + \frac{(\bar{x} - x_0)^2}{\sum (x_i - \bar{x})^2},$$

so

$$\text{Err}(x_0) = \sigma^2 \{1 + \frac{1}{n} + \frac{(\bar{x} - x_0)^2}{\sum (x_i - \bar{x})^2}\}.$$

To see how the expression (33) depends on $p$, we follow (7.12) and consider the average of $\text{Err}(x_0)$ where $x_0$ takes on the values in the training data with equal probability. This gives

$$\frac{1}{N} \sum_{i=1}^{N} \text{Err}(x_i) = \sigma^2 \left(1 + \frac{p}{N}\right).$$
In the text in (7.12) the squared bias term is included, but for this particular example it is zero.

**Example: ridge regression.** In the same linear model, if we have \( \hat{\beta} = (X^T X + \lambda I)^{-1} X^T y \), then \( E \hat{f}(x_0) = x_0^T (X^T X + \lambda I)^{-1} X^T X \beta \) and \( \text{var}(\hat{f}(x_0)) = x_0^T (X^T X + \lambda I)^{-1} x_0 \sigma^2 \), and it is at least plausible that there is some value of \( \lambda \) for which \( \text{Err}(x_0) \) is smaller than its value when \( \lambda = 0 \).

We now consider the estimation of \( \text{Err} \) in more general settings. The *training error* is defined as

\[
\text{err} = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}(x_i)),
\]

and it is clearly an underestimate of \( \text{Err} \), because \( \hat{f}(\cdot) \) is usually determined to minimize \( \text{err} \). In §7.4 the “in-sample generalization error”, or “in-sample test error” is defined as

\[
\text{Err}_{\text{in}} = \frac{1}{N} \sum_{i=1}^{N} E \{ L(Y_i^0, \hat{f}(x_i)) \mid T \},
\]

where the expectation is over a new sample of \( Y \)'s of size \( N \); one \( Y \) for each \( x_i \). This is still conditional on \( x_1, \ldots, x_N \). It can be shown that

\[
E_y(\text{Err}_{\text{in}} - \text{err}) = \frac{2}{N} \sum_{i=1}^{N} \text{cov}(y_i, \hat{f}(x_i))
\]

for both squared error loss and 0-1 loss, and that this holds approximately for log-likelihood loss. \(^1\) Further, for \( \hat{f}(x_i) \) determined by linear regression with \( d \) basis functions, the second term is \( (2/N)d\sigma^2 \). This could be simple least squares regression as above, with \( d = p \), or it could be any type of regression spline, or regression on wavelet basis. More generally, as stated in §7.6, \( \sum_{i=1}^{N} \text{cov}(y_i, \hat{f}(x_i)) = \text{trace}(S)\sigma^2 \), for a linear model and any linear fitting method \( \hat{f} = Sy \), which includes regularization methods such as spline smoothing and ridge regression.

This result gives a way to correct \( \text{err} \) to give an estimate of \( \text{Err}_{\text{in}} \): since

\[
E_y(\text{Err}_{\text{in}}) = E_y(\text{err}) + \frac{2}{N} d\sigma^2,
\]

then an estimate of the LHS is given by

\[
\text{err} + \frac{2}{N} d\sigma^2.
\]

It is further claimed that for log-likelihood loss, it can be shown that as \( N \to \infty \),

\[
-2E\{\log \Pr(y; \hat{\theta})\} \simeq -\frac{2}{N} \sum_{i=1}^{N} \log \Pr(y_i; \hat{\theta}) + \frac{2d}{N}
\]

\(^1\)Here \( E_y \) means expectation over the training \( y_1, \ldots, y_N \). This is similar to the calculation that takes \( \text{Err}_T \) to \( \text{Err} \), but emphasizes via the notation that the training \( x \)'s are fixed.
where \(d\) is the number of parameters in \(\hat{\theta}\), thus motivating the estimating the expected loss by the RHS, which is Akaike’s information criterion AIC: see, e.g. (7.30) (where however there is a typo: \(\sigma^2\) should not be in the equation).

A different derivation of AIC is given in Davison (2003, §4.7), tied more closely to maximum likelihood fitting. Suppose we have a random sample \(Y_1, \ldots, Y_N\) from an unknown true density \(g(y)\), but that we fit the family of models \(\{f(y; \theta); \theta \in \Theta\}\) by maximizing the log-likelihood function \(\ell(\theta) = \sum \log f(y; \theta)\). Define the Kullback-Liebler discrepancy

\[
D(f_\theta, g) = \int \log \left\{ \frac{g(y)}{f(y; \theta)} \right\} g(y) dy;
\]

this is 0 if \(f(y; \theta) = g(y)\) and otherwise is positive. Denote by \(\theta_g\) the value of \(\theta\) that minimizes \(D(f_\theta, g)\). The expected likelihood ratio statistic for comparing \(g\) with \(f_\theta\) at \(\theta = \hat{\theta}\) for a new random sample \(Y_1^+, \ldots, Y_N^+\) from \(g\), independent of \(Y_1, \ldots, Y_N\) is

\[
E_g^+ \left[ \sum_{i=1}^N \log \left\{ \frac{g(Y_i^+)}{f(Y_i^+; \hat{\theta})} \right\} \right] = nD(f_{\hat{\theta}}, g) \geq nD(f_{\theta_g}, g).
\]

Davison shows that

\[
nD(f_{\hat{\theta}}, g) = nD(f_{\theta_g}, g) + (1/2)\text{tr}\left\{ (\hat{\theta} - \theta_g)(\hat{\theta} - \theta_g)^T I_g(\theta_g) \right\},
\]

where \(I_g = -n \int \{ \partial^2 \log f(y; \theta) / \partial \theta \partial \theta^T \} g(y) dy\) and \(E_g\) is over the distribution of \(\hat{\theta}\). He then shows that this can be estimated by

\[
-\ell(\hat{\theta}) + c,
\]

where \(c\) estimates \((1/2)\text{tr}\{ (\hat{\theta} - \theta_g)(\hat{\theta} - \theta_g)^T I_g(\theta_g) \}\), and finally shows that \(c = p = \text{dim}(\theta)\) is the the expected value of this quantity, so serves as a reasonable estimator. This gives the estimator

\[
-\ell(\hat{\theta}) + p,
\]

the AIC is typically a scalar multiple of this. Our book uses the multiple \((2/N)\); other books use 2.

Finally, a seemingly more direct estimate of ERR\(_T\) is the cross-validation estimate

\[
CV(\hat{f}) = \frac{1}{N} \sum_{i=1}^N L\{y_i, \hat{f}^{-\kappa(i)}(x_i)\},
\]

where \(\hat{f}^{-\kappa(i)}(x_i)\) is the prediction of \(y_i\) based on a fitted model that omits either the partition \(\kappa(i)\) that \(y_i\) falls in (\(K\)-fold cross-validation), or simply the \(i\)th observation (leave-one-out cross-validation). This would seem to give an internal estimate of Err\(_T\), but the book argues that in fact it seems to estimate Err instead. For linear smoothers and squared-error loss it can be shown that

\[
CV = \frac{1}{N} \sum_{i=1}^N \frac{(y_i - \hat{f}(x_i))^2}{(1 - S_{ii})^2};
\]

replacing \((1 - S_{ii})\) by \(1 - \text{tr}(S)/N\) makes computations even simpler. This latter expression is known as the GCV criterion.