Chapter 9
Overview of Maximum Likelihood Estimation

9.1 General Notions—Simple Cases

In ordinary least squares multiple regression, the objective in fitting a model is to find the values of the unknown parameters that minimize the sum of squared errors of prediction. When the response variable is non-normal, polytomous, or not observed completely, one needs a more general objective function to optimize.

Maximum likelihood (ML) estimation is a general technique for estimating parameters and drawing statistical inferences in a variety of situations, especially nonstandard ones. Before laying out the method in general, ML estimation is illustrated with a standard situation, the one-sample binomial problem. Here, independent binary responses are observed and one wishes to draw inferences about an unknown parameter, the probability of an event in a population.

Suppose that in a population of individuals, each individual has the same probability P that an event occurs. We could also say that the event has already been observed, so that P is the prevalence of some condition in the population. For each individual, let $Y = 1$ denote the occurrence of the event and $Y = 0$ denote nonoccurrence. Then $\text{Prob}\{Y = 1\} = P$ for each individual. Suppose that a random sample of size 3 from the population is drawn and that the first individual had $Y = 1$, the second had $Y = 0$, and the third had $Y = 1$. The respective probabilities of these outcomes are P, $1-P$, and P. The joint probability of observing the independent events $Y = 1, 0, 1$ is $P(1-P)P = P^2(1-P)$. Now the value of P is unknown, but we can solve for the value of P that makes the observed data ($Y = 1, 0, 1$) *most likely to have occurred*. In this case, the value of P that maximizes $P^2(1 - P)$ is $P = 2/3$. This value for P is the *maximum likelihood estimate* (*MLE*) of the population probability.

F.E. Harrell, Jr., *Regression Modeling Strategies*, Springer Series in Statistics, DOI 10.1007/978-3-319-19425-7_9

Let us now study the situation of independent binary trials in general. Let the sample size be n and the observed responses be $Y_1, Y_2, \ldots, Y_n$. The joint probability of observing the data is given by

$$L = \prod_{i=1}^{n} P^{Y_i}(1-P)^{1-Y_i}. \tag{9.1}$$

Now let s denote the sum of the Ys or the number of times that the event occurred ($Y_i = 1$), that is the number of "successes." The number of non-occurrences ("failures") is $n - s$. The likelihood of the data can be simplified to

$$L = P^s(1-P)^{n-s}. \tag{9.2}$$

It is easier to work with the *log likelihood function*, which also has desirable statistical properties. For the one-sample binary response problem, the log likelihood is

$$\log L = s\log(P) + (n-s)\log(1-P). \tag{9.3}$$

The MLE of P is that value of P that maximizes L or $\log L$. Since $\log L$ is a smooth function of P, its maximum value can be found by finding the point at which $\log L$ has a slope of 0. The slope or first derivative of $\log L$, with respect to P, is

$$U(P) = \partial \log L/\partial P = s/P - (n-s)/(1-P). \tag{9.4}$$

The first derivative of the log likelihood function with respect to the parameter(s), here $U(P)$, is called the *score function.* Equating this function to zero requires that $s/P = (n-s)/(1-P)$. Multiplying both sides of the equation by $P(1-P)$ yields $s(1-P) = (n-s)P$ or that $s = (n-s)P + sP = nP$. Thus the MLE of P is $p = s/n$.

Another important function is called the *Fisher information* about the unknown parameters. The information function is the expected value of the negative of the curvature in $\log L$, which is the negative of the slope of the slope as a function of the parameter, or the negative of the second derivative of $\log L$. Motivation for consideration of the Fisher information is as follows. If the log likelihood function has a distinct peak, the sample provides information that allows one to readily discriminate between a good parameter estimate (the location of the obvious peak) and a bad one. In such a case the MLE will have good precision or small variance. If on the other hand the likelihood function is relatively flat, almost any estimate will do and the chosen estimate will have poor precision or large variance. The degree of peakedness of a function at a given point is the speed with which the slope is changing at that point, that is, the slope of the slope or second derivative of the function at that point.

Here, the information is

$$\begin{aligned} I(P) &= E\{-\partial^2 \log L/\partial P^2\} \\ &= E\{s/P^2 + (n-s)/(1-P)^2\} \\ &= nP/P^2 + n(1-P)/(1-P)^2 = n/[P(1-P)]. \end{aligned} \tag{9.5}$$

We estimate the information by substituting the MLE of P into $I(P)$, yielding $I(p) = n/[p(1-p)]$.

Figures 9.1, 9.2, and 9.3 depict, respectively, $\log L$, $U(P)$, and $I(P)$, all as a function of P. Three combinations of n and s were used in each graph. These combinations correspond to $p = .5, .6$, and $.6$, respectively.

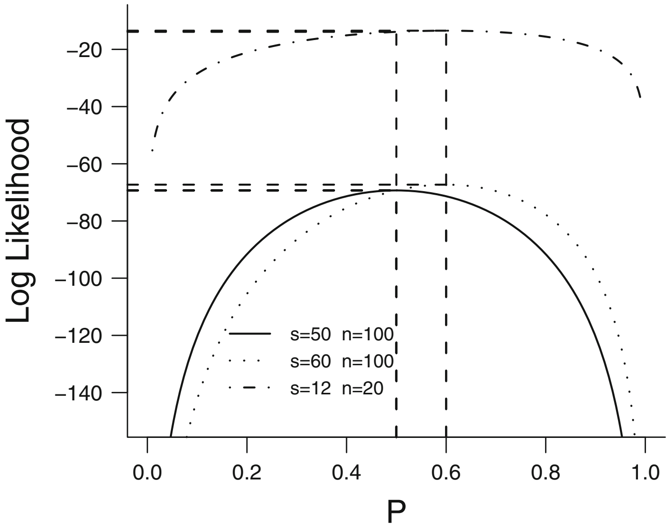

Fig. 9.1 log likelihood functions for three one-sample binomial problems

In each case it can be seen that the value of P that makes the data most likely to have occurred (the value that maximizes L or $\log L$) is p given above. Also, the score function (slope of $\log L$) is zero at $P = p$. Note that the information function $I(P)$ is highest for P approaching 0 or 1 and is lowest for P near .5, where there is maximum uncertainty about P. Note also that while $\log L$ has the same shape for the $s = 60$ and $s = 12$ curves in Figure 9.1, the range of $\log L$ is much greater for the larger sample size. Figures 9.2 and 9.3 show that the larger sample size produces a sharper likelihood. In other words, with larger n, one can zero in on the true value of P with more precision.

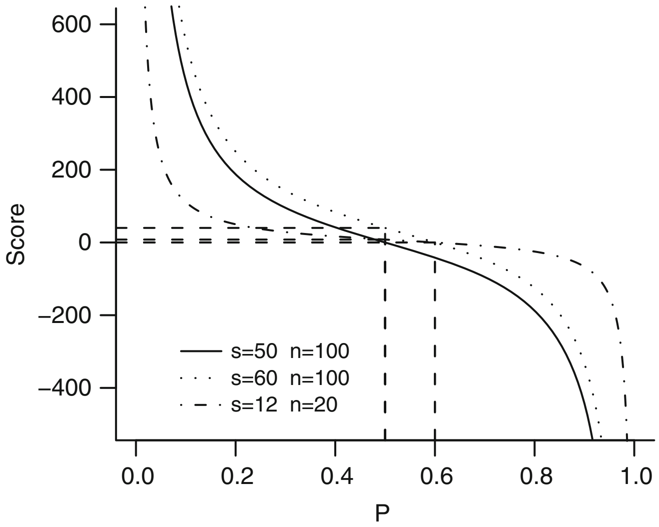

Fig. 9.2 Score functions ($\partial L/\partial P$)

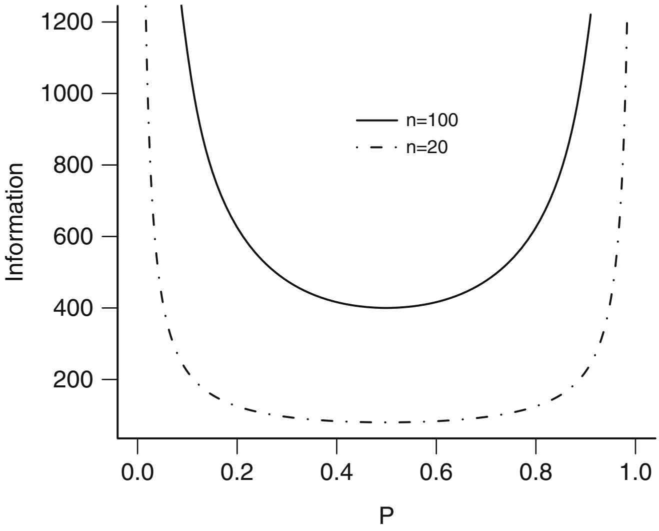

Fig. 9.3 Information functions ($-\partial^2 \log L/\partial P^2$)

In this binary response one-sample example let us now turn to inference about the parameter P. First, we turn to the estimation of the variance of the MLE, p. An estimate of this variance is given by the inverse of the information at $P = p$:

$$Var(p) = I(p)^{-1} = p(1-p)/n. \tag{9.6}$$

Note that the variance is smallest when the information is greatest ($p = 0$ or 1).

The variance estimate forms a basis for confidence limits on the unknown parameter. For large n, the MLE p is approximately normally distributed with expected value (mean) P and variance $P(1-P)/n$. Since $p(1-p)$ is a consistent estimate of $P(1-P)/n$, it follows that $p \pm z[p(1-p)/n]^{1/2}$ is an approximate $1-\alpha$ confidence interval for P if z is the $1-\alpha/2$ critical value of the standard normal distribution.

9.2 Hypothesis Tests

Now let us turn to hypothesis tests about the unknown population parameter P — $H_0 : P = P_0$. There are three kinds of statistical tests that arise from likelihood theory.

9.2.1 Likelihood Ratio Test

This test statistic is the ratio of the likelihood at the hypothesized parameter values to the likelihood of the data at the maximum (i.e., at parameter values = MLEs). It turns out that $-2\times$ the log of this likelihood ratio has desirable statistical properties. The likelihood ratio test statistic is given by

$$\begin{aligned} LR &= -2\log(L \text{ at } H_0/L \text{ at MLEs}) \\ &= -2(\log L \text{ at } H_0) - [-2(\log L \text{ at MLEs})]. \end{aligned} \tag{9.7}$$

The LR statistic, for large enough samples, has approximately a χ^2 distribution with degrees of freedom equal to the number of parameters estimated, if the null hypothesis is "simple," that is, doesn't involve any unknown parameters. Here LR has 1 d.f.

The value of $\log L$ at H_0 is

$$\log L(H_0) = s\log(P_0) + (n-s)\log(1-P_0). \tag{9.8}$$

The maximum value of $\log L$ (at MLEs) is

$$\log L(P = p) = s\log(p) + (n-s)\log(1-p). \tag{9.9}$$

For the hypothesis $H_0 : P = P_0$, the test statistic is

$$LR = -2\{s \log(P_0/p) + (n-s)\log[(1-P_0)/(1-p)]\}. \tag{9.10}$$

Note that when p happens to equal P_0, $LR = 0$. When p is far from P_0, LR will be large. Suppose that $P_0 = 1/2$, so that H_0 is $P = 1/2$. For $n = 100, s = 50$, $LR = 0$. For $n = 100, s = 60$,

$$LR = -2\{60\log(.5/.6) + 40\log(.5/.4)\} = 4.03. \tag{9.11}$$

For $n = 20, s = 12$,

$$LR = -2\{12\log(.5/.6) + 8\log(.5/.4)\} = .81 = 4.03/5. \tag{9.12}$$

Therefore, even though the best estimate of P is the same for these two cases, the test statistic is more impressive when the sample size is five times larger.

9.2.2 Wald Test

The Wald test statistic is a generalization of a t- or z-statistic. It is a function of the difference in the MLE and its hypothesized value, normalized by an estimate of the standard deviation of the MLE. Here the statistic is

$$W = [p - P_0]^2/[p(1-p)/n]. \tag{9.13}$$

For large enough n, W is distributed as χ^2 with 1 d.f. For $n = 100, s = 50$, $W = 0$. For the other samples, W is, respectively, 4.17 and 0.83 (note 0.83 = 4.17/5).

Many statistical packages treat $\sqrt{W}$ as having a t distribution instead of a normal distribution. As pointed out by Gould,[228] there is no basis for this outside of ordinary linear models[a].

9.2.3 Score Test

If the MLE happens to equal the hypothesized value P_0, P_0 maximizes the likelihood and so $U(P_0) = 0$. Rao's score statistic measures how far from zero the score function is when evaluated at the null hypothesis. The score function

[a] In linear regression, a t distribution is used to penalize for the fact that the variance of $Y|X$ is estimated. In models such as the logistic model, there is no separate variance parameter to estimate. Gould has done simulations that show that the normal distribution provides more accurate P-values than the t for binary logistic regression.

(slope or first derivative of log L) is normalized by the information (curvature or second derivative of $-\log L$). The test statistic for our example is

$$S = U(P_0)^2/I(P_0), \tag{9.14}$$

which formally does not involve the MLE, p. The statistic can be simplified as follows.

$$\begin{aligned} U(P_0) &= s/P_0 - (n-s)/(1-P_0) \\ I(P_0) &= s/P_0^2 + (n-s)/(1-P_0)^2 \\ S &= (s-nP_0)^2/[nP_0(1-P_0)] = n(p-P_0)^2/[P_0(1-P_0)]. \end{aligned} \tag{9.15}$$

Note that the numerator of S involves $s - nP_0$, the difference between the observed number of successes and the number of successes expected under H_0.

As with the other two test statistics, $S = 0$ for the first sample. For the last two samples S is, respectively, 4 and $.8 = 4/5$. 1

9.2.4 Normal Distribution—One Sample

Suppose that a sample of size n is taken from a population for a random variable Y that is known to be normally distributed with unknown mean μ and variance σ^2. Denote the observed values of the random variable by $Y_1, Y_2, \ldots, Y_n$. Now unlike the binary response case ($Y = 0$ or 1), we cannot use the notion of the probability that Y equals an observed value. This is because Y is continuous and the probability that it will take on a given value is zero. We substitute the *density function* for the probability. The density at a point y is the limit as d approaches zero of

$$\text{Prob}\{y < Y \le y + d\}/d = [F(y+d) - F(y)]/d, \tag{9.16}$$

where $F(y)$ is the normal cumulative distribution function (for a mean of μ and variance of σ^2). The limit of the right-hand side of the above equation as d approaches zero is $f(y)$, the density function of a normal distribution with mean μ and variance σ^2. This density function is

$$f(y) = (2\pi\sigma^2)^{-1/2}\exp\{-(y-\mu)^2/2\sigma^2\}. \tag{9.17}$$

The likelihood of observing the observed sample values is the joint density of the Ys. The log likelihood function here is a function of two unknowns, μ and σ^2.

$$\log L = -.5n\log(2\pi\sigma^2) - .5\sum_{i=1}^{n}(Y_i - \mu)^2/\sigma^2. \tag{9.18}$$

It can be shown that the value of μ that maximizes $\log L$ is the value that minimizes the sum of squared deviations about μ, which is the sample mean $\overline{Y}$. The MLE of σ^2 is

$$s^2 = \sum_{i=1}^{n}(Y_i - \overline{Y})^2/n. \tag{9.19}$$

Recall that the sample variance uses $n-1$ instead of n in the denominator. It can be shown that the expected value of the MLE of σ^2, s^2, is $[(n-1)/n]\sigma^2$; in other words, s^2 is too small by a factor of $(n-1)/n$ on the average. The sample variance is unbiased, but being unbiased does not necessarily make it a better estimator. The MLE has greater precision (smaller mean squared error) in many cases.

9.3 General Case

Suppose we need to estimate a vector of unknown parameters $B = \{B_1, B_2, \ldots, B_p\}$ from a sample of size n based on observations $Y_1, \ldots, Y_n$. Denote the probability or density function of the random variable Y for the ith observation by $f_i(y; B)$. The likelihood for the ith observation is $L_i(B) = f_i(Y_i; B)$. In the one-sample binary response case, recall that $L_i(B) = L_i(P) = P^{Y_i}[1-P]^{1-Y_i}$. The likelihood function, or joint likelihood of the sample, is given by

$$L(B) = \prod_{i=1}^{n} f_i(Y_i; B). \tag{9.20}$$

The log likelihood function is

$$\log L(B) = \sum_{i=1}^{n} \log L_i(B). \tag{9.21}$$

The MLE of B is that value of the vector B that maximizes $\log L(B)$ as a function of B. In general, the solution for B requires iterative trial-and-error methods as outlined later. Denote the MLE of B as $b = \{b_1, \ldots, b_p\}$. The *score vector* is the vector of first derivatives of $\log L(B)$ with respect to $B_1, \ldots, B_p$:

$$\begin{aligned} U(B) &= \{\partial/\partial B_1 \log L(B), \ldots, \partial/\partial B_p \log L(B)\} \\ &= (\partial/\partial B) \log L(B). \end{aligned} \tag{9.22}$$

The Fisher *information matrix* is the $p \times p$ matrix whose elements are the negative of the expectation of all second partial derivatives of $\log L(B)$:

$$\begin{aligned} I^*(B) &= -\{E[(\partial^2 \log L(B)/\partial B_j \partial B_k)]\}_{p\times p} \\ &= -E\{(\partial^2/\partial B \partial B') \log L(B)\}. \end{aligned} \tag{9.23}$$

The *observed information matrix* $I(B)$ is $I^*(B)$ without taking the expectation. In other words, observed values remain in the second derivatives:

$$I(B) = -(\partial^2/\partial B \partial B') \log L(B). \tag{9.24}$$

This information matrix is often estimated from the sample using the *estimated observed information* $I(b)$, by inserting b, the MLE of B, into the formula for $I(B)$.

Under suitable conditions, which are satisfied for most situations likely to be encountered, the MLE b for large samples is an optimal estimator (has as great a chance of being close to the true parameter as all other types of estimators) and has an approximate multivariate normal distribution with mean vector B and variance–covariance matrix $I^{*-1}(B)$, where C^{-1} denotes the inverse of the matrix C. (C^{-1} is the matrix such that $C^{-1}C$ is the identity matrix, a matrix with ones on the diagonal and zeros elsewhere. If C is a 1×1 matrix, $C^{-1} = 1/C$.) A consistent estimator of the variance–covariance matrix is given by the matrix V, obtained by inserting b for B in $I(B) : V = I^{-1}(b)$.

9.3.1 Global Test Statistics

Suppose we wish to test the null hypothesis $H_0 : B = B^0$. The likelihood ratio test statistic is

$$\begin{aligned} LR &= -2\log(L \text{ at } H_0/L \text{ at MLEs}) \\ &= -2[\log L(B^0) - \log L(b)]. \end{aligned} \tag{9.25}$$

The corresponding Wald test statistic, using the estimated observed information matrix, is

$$W = (b - B^0)'I(b)(b - B^0) = (b - B^0)'V^{-1}(b - B^0). \tag{9.26}$$

(A quadratic form $a'Va$ is a matrix generalization of a^2V.) Note that if the number of estimated parameters is $p = 1$, W reduces to $(b - B^0)^2/V$, which is the square of a z- or t-type statistic (estimate − hypothesized value divided by estimated standard deviation of estimate).

The score statistic for H_0 is

$$S = U'(B^0)I^{-1}(B^0)U(B^0). \tag{9.27}$$

Note that as before, S does not require solving for the MLE. For large samples, LR, W, and S have a χ^2 distribution with p d.f. under suitable conditions.

9.3.2 Testing a Subset of the Parameters

Let $B = \{B_1, B_2\}$ and suppose that we wish to test $H_0 : B_1 = B_1^0$. We are treating B_2 as a nuisance parameter. For example, we may want to test whether blood pressure and cholesterol are risk factors after adjusting for confounders age and sex. In that case B_1 is the pair of regression coefficients for blood pressure and cholesterol and B_2 is the pair of coefficients for age and sex. B_2 must be estimated to allow adjustment for age and sex, although B_2 is a nuisance parameter and is not of primary interest.

Let the number of parameters of interest be k so that B_1 is a vector of length k. Let the number of "nuisance" or "adjustment" parameters be q, the length of B_2 (note $k + q = p$).

Let b_2^* be the MLE of B_2 under the restriction that $B_1 = B_1^0$. Then the likelihood ratio statistic is

$$LR = -2[\log L \text{ at } H_0 - \log L \text{ at MLE}]. \tag{9.28}$$

Now $\log L$ at H_0 is more complex than before because H_0 involves an unknown nuisance parameter B_2 that must be estimated. $\log L$ at H_0 is the maximum of the likelihood function for any value of B_2 but subject to the condition that $B_1 = B_1^0$. Thus

$$LR = -2[\log L(B_1^0, b_2^*) - \log L(b)], \tag{9.29}$$

where as before b is the overall MLE of B. Note that LR requires maximizing two log likelihood functions. The first component of LR is a restricted maximum likelihood and the second component is the overall or unrestricted maximum.

LR is often computed by examining successively more complex models in a stepwise fashion and calculating the increment in likelihood ratio χ^2 in the overall model. The LR χ^2 for testing $H_0 : B_2 = 0$ when B_1 is not in the model is

$$LR(H_0 : B_2 = 0|B_1 = 0) = -2[\log L(0, 0) - \log L(0, b_2^*)]. \tag{9.30}$$

Here we are specifying that B_1 is not in the model by setting $B_1 = B_1^0 = 0$, and we are testing $H_0 : B_2 = 0$. (We are also ignoring nuisance parameters such as an intercept term in the test for $B_2 = 0$.)

The LR χ^2 for testing $H_0 : B_1 = B_2 = 0$ is given by

$$LR(H_0 : B_1 = B_2 = 0) = -2[\log L(0, 0) - \log L(b)]. \tag{9.31}$$

Subtracting LR χ^2 for the smaller model from that of the larger model yields

$$\begin{aligned} & -2[\log L(0, 0) - \log L(b)] - -2[\log L(0, 0) - \log L(0, b_{2*})] \\ = & \quad -2[\log L(0, b_2^*) - \log L(b)], \end{aligned} \tag{9.32}$$

which is the same as above (letting $B_1^0 = 0$).

Table 9.1 Example tests

Variables (Parameters) in Model	$LR\ \chi^2$	Number of Parameters
Intercept, age	1000	2
Intercept, age, age^2	1010	3
Intercept, age, age^2, sex	1013	4

For example, suppose successively larger models yield the $LR\ \chi^2$s in Table 9.1. The $LR\ \chi^2$ for testing for linearity in age (not adjusting for sex) against quadratic alternatives is $1010 - 1000 = 10$ with 1 d.f. The $LR\ \chi^2$ for testing the added information provided by sex, adjusting for a quadratic effect of age, is $1013 - 1010 = 3$ with 1 d.f. The $LR\ \chi^2$ for testing the joint importance of sex and the nonlinear (quadratic) effect of age is $1013 - 1000 = 13$ with 2 d.f.

To derive the Wald statistic for testing $H_0 : B_1 = B_1^0$ with B_2 being a nuisance parameter, let the MLE b be partitioned into $b = \{b_1, b_2\}$. We can likewise partition the estimated variance–covariance matrix V into

$$V = \begin{bmatrix} V_{11} & V_{12} \\ V_{12}' & V_{22} \end{bmatrix}. \tag{9.33}$$

The Wald statistic is

$$W = (b_1 - B_1^0)' V_{11}^{-1} (b_1 - B_1^0), \tag{9.34}$$

which when $k = 1$ reduces to (estimate $-$ hypothesized value)^2/ estimated variance, with the estimates adjusted for the parameters in B_2.

The score statistic for testing $H_0 : B_1 = B_1^0$ does not require solving for the full set of unknown parameters. Only the MLEs of B_2 must be computed, under the restriction that $B_1 = B_1^0$. This restricted MLE is b_2^* from above. Let $U(B_1^0, b_2^*)$ denote the vector of first derivatives of $\log L$ with respect to all parameters in B, evaluated at the hypothesized parameter values B_1^0 for the first k parameters and at the restricted MLE b_2^* for the last q parameters. (Since the last q estimates are MLEs, the last q elements of U are zero, so the formulas that follow simplify.) Let $I(B_1^0, b_2^*)$ be the observed information matrix evaluated at the same values of B as is U. The score statistic for testing $H_0 : B_1 = B_1^0$ is

$$S = U'(B_1^0, b_2^*) I^{-1}(B_1^0, b_2^*) U(B_1^0, b_2^*). \tag{9.35}$$

Under suitable conditions, the distribution of LR, W, and S can be adequately approximated by a χ^2 distribution with k d.f. 2

9.3.3 Tests Based on Contrasts

Wald tests are also done by setting up a general linear contrast. $H_0 : CB = 0$ is tested by a Wald statistic of the form

$$W = (Cb)'(CVC')^{-1}(Cb), \tag{9.36}$$

where C is a contrast matrix that "picks off" the proper elements of B. The contrasts can be much more general by allowing elements of C to be other than zero and one. For the normal linear model, W is converted to an F-statistic by dividing by the rank r of C (normally the number of rows in C), yielding a statistic with an F-distribution with r numerator degrees of freedom.

Many interesting contrasts are tested by forming differences in predicted values. By forming more contrasts than are really needed, one can develop a surprisingly flexible approach to hypothesis testing using predicted values. This has the major advantage of not requiring the analyst to account for how the predictors are coded. Suppose that one wanted to assess the difference in two vectors of predicted values, $X_1 b - X_2 b = (X_1 - X_2)b = \Delta b$ to test $H_0 : \Delta B = 0$, where $\Delta = X_1 - X_2$. The covariance matrix for Δb is given by

$$\mathrm{var}(\Delta b) = \Delta V \Delta'. \tag{9.37}$$

Let r be the rank of $\mathrm{var}(\Delta b)$, i.e., the number of non-linearly-dependent (non-redundant) differences of predicted values of Δ. The value of r and the rows of Δ that are not redundant may easily be determined using the QR decomposition as done by the R function `qr`[b]. The χ^2 statistic with r degrees of freedom (or F-statistic upon dividing the statistic by r) may be obtained by computing $\Delta^* V^* \Delta^{*\prime}$ where Δ^* is the subset of elements of Δ corresponding to non-redundant contrasts and V^* is the corresponding sub-matrix of V.

The "difference in predictions" approach can be used to compare means in a 30 year old male with a 40 year old female[c]. But the true utility of the approach is most obvious when the contrast involves multiple nonlinear terms for a single predictor, e.g., a spline function. To test for a difference in two curves, one can compare predictions at one predictor value against predictions at a series of values with at least one value that pertains to each basis function. Points can be placed between every pair of knots and beyond the outer knots, or just obtain predictions at 100 equally spaced X-values.

[b] For example, in a 3-treatment comparison one could examine contrasts between treatments A and B, A and C, and B and C by obtaining predicted values for those treatments, even though only two differences are required.

[c] The `rms` command could be `contrast(fit, list(sex='male',age=30), list(sex='female',age=40))` where all other predictors are set to medians or modes.

Suppose that there are three treatment groups (A, B, C) interacting with a cubic spline function of X. If one wants to test the multiple degree of freedom hypothesis that the profile for X is the same for treatment A and B vs. the alternative hypothesis that there is a difference between A and B for at least one value of X, one can compare predicted values at treatment A and a vector of X values against predicted values at treatment B and the same vector of X values. If the X relationship is linear, any two X values will suffice, and if X is quadratic, any three points will suffice. It would be difficult to test complex hypotheses involving only 2 of 3 treatments using other methods.

The `contrast` function in `rms` can estimate a wide variety of contrasts and make joint tests involving them, automatically computing the number of non-linearly-dependent contrasts as the test's degrees of freedom. See its help file for several examples.

9.3.4 Which Test Statistics to Use When

At this point, one may ask why three types of test statistics are needed. The answer lies in the statistical properties of the three tests as well as in computational expense in different situations. From the standpoint of statistical properties, LR is the best statistic, followed by S and W. The major statistical problem with W is that it is sensitive to problems in the estimated variance–covariance matrix in the full model. For some models, most notably the logistic regression model,[278] the variance–covariance estimates can be too large as the effects in the model become very strong, resulting in values of W that are too small (or significance levels that are too large). W is also sensitive to the way the parameter appears in the model. For example, a test of H_0 : log odds ratio $= 0$ will yield a different value of W than will H_0 : odds ratio $= 1$.

Relative computational efficiency of the three types of tests is also an issue. Computation of LR and W requires estimating all p unknown parameters, and in addition LR requires re-estimating the last q parameters under that restriction that the first k parameters $= B_1^0$. Therefore, when one is contemplating whether a set of parameters should be added to a model, the score test is the easiest test to carry out. For example, if one were interested in testing all two-way interactions among 4 predictors, the score test statistic for H_0 : "no interactions present" could be computed without estimating the $4 \times 3/2 = 6$ interaction effects. S would also be appealing for testing linearity of effects in a model—the nonlinear spline terms could be tested for significance after adjusting for the linear effects (with estimation of only the linear effects). Only parameters for linear effects must be estimated to compute S, resulting in fewer numerical problems such as lack of convergence of the Newton–Raphson algorithm.

Table 9.2 Choice of test statistics

Type of Test	Recommended Test Statistic
Global association	LR (S for large no. parameters)
Partial association	W (LR or S if problem with W)
Lack of fit, 1 d.f.	W or S
Lack of fit, >1 d.f.	S
Inclusion of additional predictors	S

The Wald tests are very easy to make after all the parameters in a model have been estimated. Wald tests are thus appealing in a multiple regression setup when one wants to test whether a given predictor or set of predictors is "significant." A score test would require re-estimating the regression coefficients under the restriction that the parameters of interest equal zero.

Likelihood ratio tests are used often for testing the global hypothesis that no effects are significant, as the log likelihood evaluated at the MLEs is already available from fitting the model and the log likelihood evaluated at a "null model" (e.g., a model containing only an intercept) is often easy to compute. Likelihood ratio tests should also be used when the validity of a Wald test is in question as in the example cited above.

Table 9.2 summarizes recommendations for choice of test statistics for various situations.

9.3.5 Example: Binomial—Comparing Two Proportions

Suppose that a binary random variable Y_1 represents responses for population 1 and Y_2 represents responses for population 2. Let $P_i = \text{Prob}\{Y_i = 1\}$ and assume that a random sample has been drawn from each population with respective sample sizes n_1 and n_2. The sample values are denoted by $Y_{i1}, \ldots, Y_{in_i}, i = 1$ or 2. Let

$$s_1 = \sum_{j=1}^{n_1} Y_{1j} \qquad s_2 = \sum_{j=1}^{n_2} Y_{2j}, \tag{9.38}$$

the respective observed number of "successes" in the two samples. Let us test the null hypothesis $H_0 : P_1 = P_2$ based on the two samples.

The likelihood function is

$$L = \prod_{i=1}^{2} \prod_{j=1}^{n_i} P_i^{Y_{ij}} (1 - P_i)^{1-Y_{ij}}$$

$$= \prod_{i=1}^{2} P_i^{s_i}(1-P_i)^{n_i-s_i} \tag{9.39}$$

$$\log L = \sum_{i=1}^{2}\{s_i \log(P_i) + (n_i - s_i)\log(1-P_i)\}. \tag{9.40}$$

Under $H_0, P_1 = P_2 = P$, so

$$\log L(H_0) = s\log(P) + (n-s)\log(1-P), \tag{9.41}$$

where $s = s_1 + s_2, n = n_1 + n_2$. The (restricted) MLE of this common P is $p = s/n$ and $\log L$ at this value is $s\log(p) + (n-s)\log(1-p)$.

Since the original unrestricted log likelihood function contains two terms with separate parameters, the two parts may be maximized separately giving MLEs

$$p_1 = s_1/n_1 \qquad \text{and} \qquad p_2 = s_2/n_2. \tag{9.42}$$

$\log L$ evaluated at these (unrestricted) MLEs is

$$\begin{aligned}\log L = {} & s_1\log(p_1) + (n_1 - s_1)\log(1-p_1)\\ & + s_2\log(p_2) + (n_2 - s_2)\log(1-p_2).\end{aligned} \tag{9.43}$$

The likelihood ratio statistic for testing $H_0 : P_1 = P_2$ is then

$$\begin{aligned}LR = {} & -2\{s\log(p) + (n-s)\log(1-p)\\ & - [s_1\log(p_1) + (n_1 - s_1)\log(1-p_1)\\ & + s_2\log(p_2) + (n_2 - s_2)\log(1-p_2)]\}.\end{aligned} \tag{9.44}$$

This statistic for large enough n_1 and n_2 has a χ^2 distribution with 1 d.f. since the null hypothesis involves the estimation of one fewer parameter than does the unrestricted case. This LR statistic is the likelihood ratio χ^2 statistic for a 2×2 contingency table. It can be shown that the corresponding score statistic is equivalent to the Pearson χ^2 statistic. The better LR statistic can be used routinely over the Pearson χ^2 for testing hypotheses in contingency tables.

9.4 Iterative ML Estimation

In most cases, one cannot explicitly solve for MLEs but must use trial-and-error numerical methods to solve for parameter values B that maximize $\log L(B)$ or yield a score vector $U(B) = 0$. One of the fastest and most applicable methods for maximizing a function is the Newton–Raphson method, which is based on approximating $U(B)$ by a linear function of B in a small region. A starting estimate b^0 of the MLE b is made. The linear approximation (a first-order Taylor series approximation)

$$U(b) = U(b^0) - I(b^0)(b - b^0) \tag{9.45}$$

is equated to 0 and solved by b yielding

$$b = b^0 + I^{-1}(b^0)U(b^0). \tag{9.46}$$

The process is continued in like fashion. At the ith step the next estimate is obtained from the previous estimate using the formula

$$b^{i+1} = b^i + I^{-1}(b^i)U(b^i). \tag{9.47}$$

If the log likelihood actually worsened at b^{i+1}, “step halving” is used; b^{i+1} is replaced with $(b^i + b^{i+1})/2$. Further step halving is done if the log likelihood still is worse than the log likelihood at b^i, after which the original iterative strategy is resumed. The Newton–Raphson iterations continue until the $-2 \log$ likelihood changes by only a small amount over the previous iteration (say .025). The reasoning behind this stopping rule is that estimates of B that change the $-2 \log$ likelihood by less than this amount do not affect statistical inference since $-2 \log$ likelihood is on the χ^2 scale. 3

9.5 Robust Estimation of the Covariance Matrix

The estimator for the covariance matrix of b found in Section 9.3 assumes that the model is correctly specified in terms of distribution, regression assumptions, and independence assumptions. The model may be incorrect in a variety of ways such as non-independence (e.g., repeated measurements within subjects), lack of fit (e.g., omitted covariable, incorrect covariable transformation, omitted interaction), and distributional (e.g., Y has a Γ distribution instead of a normal distribution). Variances and covariances, and hence confidence intervals and Wald tests, will be incorrect when these assumptions are violated.

For the case in which the observations are independent and identically distributed but other assumptions are possibly violated, Huber[312] provided a covariance matrix estimator that is consistent. His “sandwich” estimator is given by

$$H = I^{-1}(b)[\sum_{i=1}^{n} U_i U_i'] I^{-1}(b), \tag{9.48}$$

where $I(b)$ is the observed information matrix (Equation 9.24) and U_i is the vector of derivatives, with respect to all parameters, of the log likelihood component for the ith observation (assuming the log likelihood can be partitioned into per-observation contributions). For the normal multiple linear regression case, H was derived by White:[659]

$$(X'X)^{-1}[\sum_{i=1}^{n}(Y_i - X_i b)^2 X_i X_i'](X'X)^{-1}, \tag{9.49}$$

where X is the design matrix (including an intercept if appropriate) and X_i is the vector of predictors (including an intercept) for the ith observation. This covariance estimator allows for any pattern of variances of $Y|X$ across observations. Note that even though H improves the bias of the covariance 4
matrix of b, it may actually have larger mean squared error than the ordinary estimate in some cases due to increased variance.[164, 529]

When observations are dependent within clusters, and the number of observations within clusters is very small in comparison to the total sample size, a simple adjustment to Equation 9.48 can be used to derive appropriate covariance matrix estimates (see Lin [407, p. 2237], Rogers,[529] and Lee et al. [393, Eq. 5.1, p. 246]). One merely accumulates sums of elements of U within clusters before computing cross-product terms:

$$H_c = I^{-1}(b)[\sum_{i=1}^{c}\{(\sum_{j=1}^{n_i} U_{ij})(\sum_{j=1}^{n_i} U_{ij})'\}]I^{-1}(b), \tag{9.50}$$

where c is the number of clusters, n_i is the number of observations in the ith cluster, U_{ij} is the contribution of the jth observation within the ith cluster to the score vector, and $I(b)$ is computed as before ignoring clusters. For a model such as the Cox model which has no per-observation score contributions, special score residuals[393, 407, 410, 605] are used for U.

Bootstrapping can also be used to derive robust covariance matrix estimates[177, 178] in many cases, especially if covariances of b that are not conditional on X are appropriate. One merely generates approximately 200 samples with replacement from the original dataset, computes 200 sets of parameter estimates, and computes the sample covariance matrix of these parameter estimates. Sampling with replacement from entire clusters can be used to derive variance estimates in the presence of intracluster correlation.[188] Bootstrap 5
estimates of the conditional variance–covariance matrix given X are harder to obtain and depend on the model assumptions being satisfied. The simpler unconditional estimates may be more appropriate for many non-experimental studies where one may desire to "penalize" for the X being random variables. It is interesting that these unconditional estimates may be very difficult to obtain parametrically, since a multivariate distribution may need to be assumed for X.

The previous discussion addresses the use of a "working independence model" with clustered data. Here one estimates regression coefficients assuming independence of all records (observations). Then a sandwich or bootstrap method is used to increase standard errors to reflect some redundancy in the correlated observations. The parameter estimates will often be consistent estimates of the true parameter values, but they may be inefficient for certain cluster or correlation structures. 6

The `rms` package's `robcov` function computes the Huber robust covariance matrix estimator, and the `bootcov` function computes the bootstrap covariance estimator. Both of these functions allow for clustering.

9.6 Wald, Score, and Likelihood-Based Confidence Intervals

A $1 - \alpha$ confidence interval for a parameter β_i is the set of all values β_i^0 that if hypothesized would be accepted in a test of $H_0 : \beta_i = \beta_i^0$ at the α level. What test should form the basis for the confidence interval? The Wald test is most frequently used because of its simplicity. A two-sided $1-\alpha$ confidence interval is $b_i \pm z_{1-\alpha/2}s$, where z is the critical value from the normal distribution and s is the estimated standard error of the parameter estimate b_i.[d] The problem with s discussed in Section 9.3.4 points out that Wald statistics may not always be a good basis. Wald-based confidence intervals are also symmetric even though the coverage probability may not be.[160] Score- and LR-based confidence limits have definite advantages. When Wald-type confidence intervals are appropriate, the analyst may consider insertion of robust covariance estimates (Section 9.5) into the confidence interval formulas (note that adjustments for heterogeneity and correlated observations are not available for score and LR statistics). 7

Wald– (asymptotic normality) based statistics are convenient for deriving confidence intervals for linear or more complex combinations of the model's parameters. As in Equation 9.36, the variance–covariance matrix of Cb, where C is an appropriate matrix and b is the vector of parameter estimates, is CVC', where V is the variance matrix of b. In regression models we commonly substitute a vector of predictors (and optional intercept) for C to obtain the variance of the linear predictor Xb as

$$\mathrm{var}(Xb) = XVX'. \tag{9.51}$$

See Section 9.3.3 for related information.

[d] This is the basis for confidence limits computed by the `R` `rms` package's `Predict`, `summary`, and `contrast` functions. When the `robcov` function has been used to replace the information-matrix-based covariance matrix with a Huber robust covariance estimate with an optional cluster sampling correction, the functions are using a "robust" Wald statistic basis. When the `bootcov` function has been used to replace the model fit's covariance matrix with a bootstrap unconditional covariance matrix estimate, the two functions are computing confidence limits based on a normal distribution but using more nonparametric covariance estimates.

9.6.1 Simultaneous Wald Confidence Regions

The confidence intervals just discussed are *pointwise* confidence intervals. For OLS regression there are methods for computing confidence intervals with exact *simultaneous* confidence coverage for multiple estimates[374]. There are approximate methods for simultaneous confidence limits for all models for which the vector of estimates b is approximately multivariately normally distributed. The method of Hothorn et al.[307] is quite general; in their R package `multcomp`'s `glht` function, the user can specify any contrast matrix over which the individual confidence limits will be simultaneous. A special case of a contrast matrix is the design matrix X itself, resulting in simultaneous confidence bands for any number of predicted values. An example is shown in Figure 9.5. See Section 9.3.3 for a good use for simultaneous contrasts.

9.7 Bootstrap Confidence Regions

A more nonparametric method for computing confidence intervals for functions of the vector of parameters B can be based on bootstrap percentile confidence limits. For each sample with replacement from the original dataset, one computes the MLE of B, b, and then the quantity of interest $g(b)$. Then the gs are sorted and the desired quantiles are computed. At least 1000 bootstrap samples will be needed for accurate assessment of outer confidence limits. This method is suitable for obtaining pointwise confidence bands for 8 a nonlinear regression function, say, the relationship between age and the log odds of disease. At each of 100 age values the predicted logits are computed for each bootstrap sample. Then separately for each age point the 0.025 and 0.975 quantiles of 1000 estimates of the logit are computed to derive a 0.95 confidence band. Other more complex bootstrap schemes will achieve somewhat greater accuracy of confidence interval coverage,[178] and as described in Section 9.5 one can use variations on the basic bootstrap in which the predictors are considered fixed and/or cluster sampling is taken into account. The R function `bootcov` in the `rms` package bootstraps model fits to obtain unconditional (with respect to predictors) bootstrap distributions with or without cluster sampling. `bootcov` stores the matrix of bootstrap regression coefficients so that the bootstrapped quantities of interest can be computed in one sweep of the coefficient matrix once bootstrapping is completed. 9

For many regression models. the `rms` package's `Predict`, `summary`, and `contrast` functions make it easy to compute pointwise bootstrap confidence intervals in a variety of contexts. As an example, consider 200 simulated x values from a log-normal distribution and simulate binary y from a true population binary logistic model given by

$$\text{Prob}(Y = 1|X = x) = \frac{1}{1 + \exp[-(1 + x/2)]}. \tag{9.52}$$

Not knowing the true model, a quadratic logistic model is fitted. The R code needed to generate the data and fit the model is given below.

```
require(rms)
```

```
n ← 200
set.seed(15)
x1 ← rnorm(n)
logit ← x1/2
y ← ifelse(runif(n) ≤ plogis(logit), 1, 0)
dd ← datadist(x1);  options(datadist='dd')
f ← lrm(y ∼ pol(x1,2), x=TRUE, y=TRUE)
print(f, latex=TRUE)
```

Logistic Regression Model

```
lrm(formula = y ~ pol(x1, 2), x = TRUE, y = TRUE)
```

		Model Likelihood Ratio Test		Discrimination Indexes		Rank Discrim. Indexes	
Obs	200	LR χ^2	16.37	R^2	0.105	C	0.642
0	97	d.f.	2	g	0.680	D_{xy}	0.285
1	103	$\Pr(>\chi^2)$	0.0003	g_r	1.973	γ	0.286
max $\lvert\frac{\partial \log L}{\partial \beta}\rvert$	3×10^{-9}			g_p	0.156	τ_a	0.143
				Brier	0.231		

	Coef	S.E.	Wald Z	$\Pr(>\lvert Z\rvert)$
Intercept	-0.0842	0.1823	-0.46	0.6441
x1	0.5902	0.1580	3.74	0.0002
x1^2	0.1557	0.1136	1.37	0.1708

```
latex(anova(f), file='', table.env=FALSE)
```

	χ^2	d.f.	P
x1	13.99	2	0.0009
Nonlinear	1.88	1	0.1708
TOTAL	13.99	2	0.0009

The `bootcov` function is used to draw 1000 resamples to obtain bootstrap estimates of the covariance matrix of the regression coefficients as well as to save the 1000 × 3 matrix of regression coefficients. Then, because individual regression coefficients for x do not tell us much, we summarize the

x-effect by computing the effect (on the logit scale) of increasing x from 1 to 5. We first compute bootstrap nonparametric percentile confidence intervals the long way. The 1000 bootstrap estimates of the log odds ratio are computed easily using a single matrix multiplication with the difference in predictions approach, multiplying the difference in two design matrices, and we obtain the bootstrap estimate of the standard error of the log odds ratio by computing the sample standard deviation of the 1000 values[e]. Bootstrap percentile confidence limits are just sample quantiles from the bootstrapped log odds ratios.

```
# Get 2-row design matrix for obtaining predicted values
# for x = 1 and 5
X ← cbind(Intercept=1,
          predict(f, data.frame(x1=c(1,5)), type='x'))
Xdif ← X[2,,drop=FALSE] - X[1,,drop=FALSE]
Xdif
```

```
  Intercept pol(x1, 2)x1 pol(x1, 2)x1^2
2         0            4             24
```

```
b ← bootcov(f, B=1000)
boot.log.odds.ratio ← b$boot.Coef %*% t(Xdif)
sd(boot.log.odds.ratio)
```

```
[1] 2.752103
```

```
# This is the same as from summary(b, x=c(1,5)) as summary
# uses the bootstrap covariance matrix
summary(b, x1=c(1,5))[1, 'S.E.']
```

```
[1] 2.752103
```

```
# Compare this s.d. with one from information matrix
summary(f, x1=c(1,5))[1,'S.E.']
```

```
[1] 2.988373
```

```
# Compute percentiles of bootstrap odds ratio
exp(quantile(boot.log.odds.ratio, c(.025, .975)))
```

```
        2.5%        97.5%
2.795032e+00 2.067146e+05
```

```
# Automatic:
summary(b, x1=c(1,5))[' Odds Ratio',]
```

[e] As indicated below, this standard deviation can also be obtained by using the `summary` function on the object returned by `bootcov`, as `bootcov` returns a fit object like one from `lrm` except with the bootstrap covariance matrix substituted for the information-based one.

```
         Low         High        Diff.       Effect          S.E.
1.000000e+00 5.000000e+00 4.000000e+00 4.443932e+02            NA
  Lower 0.95   Upper 0.95         Type
2.795032e+00 2.067146e+05 2.000000e+00
```

```
print(contrast(b, list(x1=5), list(x1=1), fun=exp))
```

```
   Contrast     S.E.    Lower    Upper    Z Pr(>|z|)
11  6.09671 2.752103 1.027843 12.23909 2.22   0.0267

Confidence intervals are 0.95 bootstrap nonparametric percentile intervals
```

```
# Figure 9.4
hist(boot.log.odds.ratio, nclass=100, xlab='log(OR)',
  main='')
```

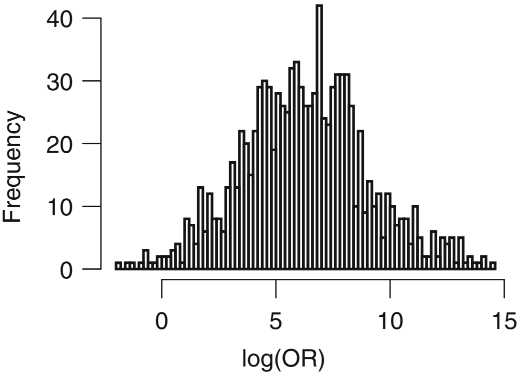

Fig. 9.4 Distribution of 1000 bootstrap x=1:5 log odds ratios

Figure 9.4 shows the distribution of log odds ratios.

Now consider confidence bands for the true log odds that $y = 1$, across a sequence of x values. The `Predict` function automatically calculates point-by-point bootstrap percentiles, basic bootstrap, or BCa[203] confidence limits when the fit has passed through `bootcov`. Simultaneous Wald-based confidence intervals[307] and Wald intervals substituting the bootstrap covariance matrix estimator are added to the plot when `Predict` calls the `multcomp` package (Figure 9.5).

```
x1s ← seq(0, 5, length=100)
pwald     ← Predict(f, x1=x1s)
psand     ← Predict(robcov(f), x1=x1s)
pbootcov  ← Predict(b, x1=x1s, usebootcoef=FALSE)
pbootnp   ← Predict(b, x1=x1s)
pbootbca  ← Predict(b, x1=x1s, boot.type='bca')
pbootbas  ← Predict(b, x1=x1s, boot.type='basic')
psimult   ← Predict(b, x1=x1s, conf.type='simultaneous')
```

```
z ← rbind('Boot percentile'      = pbootnp,
          'Robust sandwich'      = psand,
          'Boot BCa'             = pbootbca,
          'Boot covariance+Wald'= pbootcov,
          Wald                   = pwald,
          'Boot basic'           = pbootbas,
          Simultaneous           = psimult)

z$class ← ifelse(z$.set. %in% c('Boot percentile','Boot bca',
                  'Boot basic'), 'Other', 'Wald')
ggplot(z, groups=c('.set.', 'class'),
       conf='line', ylim=c(-1, 9), legend.label=FALSE)
```

See Problems at chapter's end for a worrisome investigation of bootstrap confidence interval coverage using simulation. It appears that when the model's log odds distribution is not symmetric and includes very high or very low probabilities, neither the bootstrap percentile nor the bootstrap BCa intervals have good coverage, while the basic bootstrap and ordinary Wald intervals are fairly accurate[f]. It is difficult in general to know when to trust the bootstrap for logistic and perhaps other models when computing confidence intervals, and the simulation problem suggests that the basic bootstrap should be used more frequently. Similarly, the distribution of bootstrap effect estimates can be suspect. Asymmetry in this distribution does not imply that the true sampling distribution is asymmetric or that the percentile intervals are preferred.

9.8 Further Use of the Log Likelihood

9.8.1 Rating Two Models, Penalizing for Complexity

Suppose that from a single sample two competing models were developed. Let the respective $-2\log$ likelihoods for these models be denoted by L_1 and L_2, and let p_1 and p_2 denote the number of parameters estimated in each model. Suppose that $L_1 < L_2$. It may be tempting to rate model one as the "best" fitting or "best" predicting model. That model may provide a better fit for the data at hand, but if it required many more parameters to be estimated, it may not be better "for the money." If both models were applied to a new sample, model one's overfitting of the original dataset may actually result in a worse fit on the new dataset.

[f] Limited simulations using the conditional bootstrap and Firth's penalized likelihood[281] did not show significant improvement in confidence interval coverage.

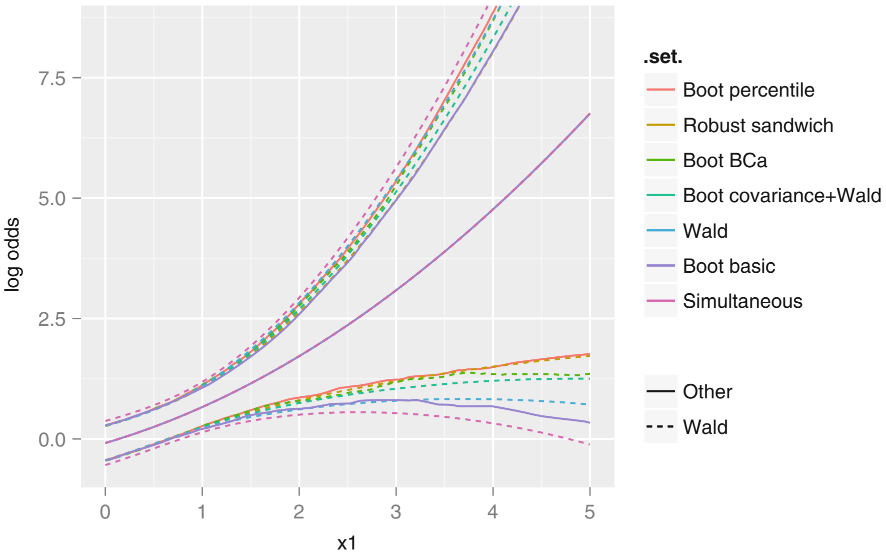

Fig. 9.5 Predicted log odds and confidence bands for seven types of confidence intervals. Seven categories are ordered top to bottom corresponding to order of lower confidence bands at `x1=5`. Dotted lines are for Wald–type methods that yield symmetric confidence intervals and assume normality of point estimators.

Akaike's information criterion (AIC[33, 359, 633]) provides a method for penalizing the log likelihood achieved by a given model for its complexity to obtain a more unbiased assessment of the model's worth. The penalty is to subtract the number of parameters estimated from the log likelihood, or equivalently to add twice the number of parameters to the $-2\log$ likelihood. The penalized log likelihood is analogous to Mallows' C_p in ordinary multiple regression. AIC would choose the model by comparing $L_1 + 2p_1$ to $L_2 + 2p_2$ and picking the model with the lower value. We often use AIC in "adjusted χ^2" form: 10

$$\mathrm{AIC} = \mathrm{LR}\ \chi^2 - 2p. \tag{9.53}$$

Breiman [66, Section 1.3] and Chatfield [100, Section 4] discuss the fallacy of AIC and C_p for selecting from a series of non-prespecified models. 11

9.8.2 Testing Whether One Model Is Better than Another

One way to test whether one model (A) is better than another (B) is to embed both models in a more general model ($A + B$). Then a $LR\ \chi^2$ test can be done to test whether A is better than B by changing the hypothesis to test whether A adds predictive information to B ($H_0 : A + B > B$) and whether B adds information to A ($H_0 : A+B > A$). The approach of testing $A > B$ via testing $A+B > B$ and $A+B > A$ is especially useful for selecting from competing predictors such as a multivariable model and a subjective assessor.[131, 264, 395, 669]

Note that $LR\ \chi^2$ for $H_0 : A + B > B$ minus $LR\ \chi^2$ for $H_0 : A + B > A$ equals $LR\ \chi^2$ for $H_0 : A$ has no predictive information minus $LR\ \chi^2$ for $H_0 : B$ has no predictive information,[665] the difference in $LR\ \chi^2$ for testing each model (set of variables) separately. This gives further support to the use of two separately computed Akaike's information criteria for rating the two sets of variables. 12

See Section 9.8.4 for an example.

9.8.3 Unitless Index of Predictive Ability

The global likelihood ratio test for regression is useful for determining whether any predictor is associated with the response. If the sample is large enough, even weak associations can be "statistically significant." Even though a likelihood ratio test does not shed light on a model's predictive strength, the log likelihood (L.L.) can still be useful here. Consider the following L.L.s:

Best (lowest) possible -2 L.L.:
$L^* = -2$ L.L. for a hypothetical model that perfectly predicts the outcome.

-2 L.L. achieved:
$L = -2$ L.L. for the fitted model.

Worst -2 L.L.:
$L^0 = -2$ L.L. for a model that has no predictive information.

The last -2 L.L., for a "no information" model, is the -2 L.L. under the null hypothesis that all regression coefficients except for intercepts are zero. A "no information" model often contains only an intercept and some distributional parameters (a variance, for example). 13

The quantity $L^0 - L$ is LR, the log likelihood ratio statistic for testing the global null hypothesis that no predictors are related to the response. It is also the -2 log likelihood "explained" by the model. The best (lowest) -2 L.L. is L^*, so the amount of L.L. that is capable of being explained by the model is $L^0 - L^*$. The fraction of -2 L.L. explained that was capable of being explained is

$$(L^0 - L)/(L^0 - L^*) \quad = \quad LR/(L^0 - L^*). \tag{9.54}$$

The fraction of log likelihood explained is analogous to R^2 in an ordinary linear model, although Korn and Simon[365, 366] provide a much more precise notion.

Akaike's information criterion can be used to penalize this measure of association for the number of parameters estimated (p, say) to transform this unitless measure of association into a quantity that is analogous to the adjusted R^2 or Mallows' C_p in ordinary linear regression. We let R denote the square root of such a penalized fraction of log likelihood explained. R is defined by

$$R^2 = (LR - 2p)/(L_0 - L^*). \tag{9.55}$$

The R index can be used to assess how well the model compares with a "perfect" model, as well as to judge whether a more complex model has predictive strength that justifies its additional parameters. Had p been used in Equation 9.55 rather than $2p$, R^2 is negative if the log likelihood explained is less than what one would expect by chance. R will be the square root of $1 - 2p/(L_0 - L^*)$ if the model perfectly predicts the response. This upper limit will be near one if the sample size is large.

Partial R indexes can also be defined by substituting the -2 L.L. explained for a given factor in place of that for the entire model, LR. The "penalty factor" p becomes one. This index R_{partial} is defined by

$$R^2_{\text{partial}} = (LR_{\text{partial}} - 2)/(L_0 - L^*), \tag{9.56}$$

which is the (penalized) fraction of -2 log likelihood explained by the predictor. Here LR_{partial} is the log likelihood ratio statistic for testing whether the predictor is associated with the response, after adjustment for the other predictors. Since such likelihood ratio statistics are tedious to compute, the 1 d.f. Wald χ^2 can be substituted for the LR statistic (keeping in mind that difficulties with the Wald statistic can arise).

Liu and Dyer[424] and Cox and Wermuth[136] point out difficulties with the R^2 measure for binary logistic models. Cox and Snell[135] and Magee[432] used other analogies to derive other R^2 measures that may have better properties. For a sample of size n and a Wald statistic for testing overall association, they defined

$$\begin{aligned} R^2_{\text{W}} &= \frac{W}{n+W} \\ R^2_{\text{LR}} &= 1 - \exp(-\text{LR}/n) \\ &= 1 - \lambda^{2/n}, \end{aligned} \tag{9.57}$$

where λ is the null model likelihood divided by the fitted model likelihood. In the case of ordinary least squares with normality both of the above indexes are equal to the traditional R^2. R^2_{LR} is equivalent to Maddala's index [431, Eq. 2.44]. Cragg and Uhler[137] and Nagelkerke[471] suggested dividing R^2_{LR} by its maximum attainable value

$$R^2_{\text{max}} = 1 - \exp(-L^0/n) \tag{9.58}$$

to derive R^2_{N} which ranges from 0 to 1. This is the form of the R^2 index we use throughout.

For penalizing for overfitting, see Verweij and van Houwelingen[640] for an overfitting-corrected R^2 that uses a cross-validated likelihood. 14

9.8.4 Unitless Index of Adequacy of a Subset of Predictors

Log likelihoods are also useful for quantifying the predictive information contained in a subset of the predictors compared with the information contained in the entire set of predictors.[264] Let LR again denote the -2 log likelihood ratio statistic for testing the joint significance of the full set of predictors. Let LR^s denote the -2 log likelihood ratio statistic for testing the importance of the subset of predictors of interest, excluding the other predictors from the model. A measure of adequacy of the subset for predicting the response is given by

$$A = LR^s/LR. \tag{9.59}$$

A is then the proportion of log likelihood explained by the subset with reference to the log likelihood explained by the entire set. When $A = 1$, the subset contains all the predictive information found in the whole set of predictors; that is, the subset is adequate by itself and the additional predictors contain no independent information. When $A = 0$, the subset contains no predictive information by itself.

Califf et al.[89] used the A index to quantify the adequacy (with respect to prognosis) of two competing sets of predictors that each describe the extent of coronary artery disease. The response variable was time until cardiovascular death and the statistical model used was the Cox[132] proportional hazards model. Some of their results are reproduced in Table 9.3. 15 A chance-corrected adequacy measure could be derived by squaring the ratio of the R-index for the subset to the R-index for the whole set. A formal test of superiority of X_1 = maximum % stenosis over X_2 = jeopardy score can be obtained by testing whether X_1 adds to X_2 ($LR\ \chi^2 = 57.5 - 42.6 = 14.9$) and whether X_2 adds to X_1 ($LR\ \chi^2 = 57.5 - 51.8 = 5.7$). X_1 adds more to X_2 (14.9) than X_2 adds to X_1 (5.7). The difference $14.9 - 5.7 = 9.2$ equals the difference in single factor χ^2 ($51.8 - 42.6$)[665].

Table 9.3 Completing prognostic markers

Predictors Used	$LR\ \chi^2$	Adequacy
Coronary jeopardy score	42.6	0.74
Maximum % stenosis in each artery	51.8	0.90
Combined	57.5	1.00

9.9 Weighted Maximum Likelihood Estimation

It is commonly the case that data elements represent combinations of values that pertain to a set of individuals. This occurs, for example, when unique combinations of X and Y are determined from a massive dataset, along with the frequency of occurrence of each combination, for the purpose of reducing the size of the dataset to analyze. For the ith combination we have a *case weight* w_i that is a positive integer representing a frequency. Assuming that observations represented by combination i are independent, the likelihood needed to represent all w_i observations is computed simply by multiplying all of the likelihood elements (each having value L_i), yielding a total likelihood contribution for combination i of $L_i^{w_i}$ or a log likelihood contribution of $w_i \log L_i$. To obtain a likelihood for the entire dataset one computes the product over all combinations. The total log likelihood is $\sum w_i \log L_i$. As an example, the weighted likelihood that would be used to fit a weighted logistic regression model is given by

$$L = \prod_{i=1}^{n} P_i^{w_i Y_i} (1 - P_i)^{w_i(1-Y_i)}, \tag{9.60}$$

where there are n combinations, $\sum_{i=1}^{n} w_i > n$, and P_i is $\text{Prob}[Y_i = 1|X_i]$ as dictated by the model. Note that in general the correct likelihood function cannot be obtained by weighting the data and using an unweighted likelihood.

By a small leap one can obtain weighted maximum likelihood estimates from the above method even if the weights do not represent frequencies or even integers, as long as the weights are non-negative. Non-frequency weights are commonly used in sample surveys to adjust estimates back to better represent a target population when some types of subjects have been oversampled from that population. Analysts should beware of possible losses in efficiency when obtaining weighted estimates in sample surveys.[363, 364] Making the regression estimates conditional on sampling strata by including strata as covariables may be preferable to re-weighting the strata. If weighted estimates must be obtained, the weighted likelihood function is generally valid for obtaining properly weighted parameter estimates. However, the variance–covariance matrix obtained by inverting the information matrix from the weighted likelihood will not be correct in general. For one thing, the sum of the weights may be far from the number of subjects in the sample. A rough approximation to the variance–covariance matrix may be obtained by first multiplying each weight by $n/\sum w_i$ and then computing the weighted information matrix, where n is the number of actual subjects in the sample. 16

9.10 Penalized Maximum Likelihood Estimation

Maximizing the log likelihood provides the best fit to the dataset at hand, but this can also result in fitting noise in the data. For example, a categorical predictor with 20 levels can produce extreme estimates for some of the 19 regression parameters, especially for the small cells (see Section 4.5). A shrinkage approach will often result in regression coefficient estimates that while biased are lower in mean squared error and hence are more likely to be close to the true unknown parameter values. Ridge regression is one approach to shrinkage, but a more general and better developed approach is penalized maximum likelihood estimation,[237, 388, 639, 641] which is really a special case 17 of Bayesian modeling with a Gaussian prior. Letting L denote the usual likelihood function and λ be a penalty factor, we maximize the penalized log likelihood given by

$$\log L - \frac{1}{2}\lambda \sum_{i=1}^{p} (s_i\beta_i)^2, \tag{9.61}$$

where $s_1, s_2, \ldots, s_p$ are scale factors chosen to make $s_i\beta_i$ unitless. Most authors standardize the data first and do not have scale factors in the equation, but Equation 9.61 has the advantage of allowing estimation of β on the original scale of the data. The usual methods (e.g., Newton–Raphson) are used to maximize 9.61.

The choice of the scaling constants has received far too little attention in the ridge regression and penalized MLE literature. It is common to use the 18 standard deviation of each column of the design matrix to scale the corresponding parameter. For models containing nothing but continuous variables that enter the regression linearly, this is usually a reasonable approach. For continuous variables represented with multiple terms (one of which is linear), it is not always reasonable to scale each nonlinear term with its own standard deviation. For dummy variables, scaling using the standard deviation ($\sqrt{d(1-d)}$, where d is the mean of the dummy variable, i.e., the fraction of observations in that cell) is problematic since this will result in high prevalance cells getting more shrinkage than low prevalence ones because the high prevalence cells will dominate the penalty function.

An advantage of the formulation in Equation 9.61 is that one can assign scale constants of zero for parameters for which no shrinkage is desired.[237, 639] For example, one may have prior beliefs that a linear additive model will fit the data. In that case, nonlinear and non-additive terms may be penalized.

For a categorical predictor having c levels, users of ridge regression often do not recognize that the amount of shrinkage and the predicted values from the fitted model depend on how the design matrix is coded. For example, one will get different predictions depending on which cell is chosen as the reference cell when constructing dummy variables. The setup in Equation 9.61 has the same problem. For example, if for a three-category factor we use category 1 as the reference cell and have parameters β_2 and β_3, the unscaled penalty function is $\beta_2^2 + \beta_3^2$. If category 3 were used as the reference cell instead, the penalty would be $\beta_3^2 + (\beta_2 - \beta_3)^2$. To get around this problem, Verweij and van Houwelingen[639] proposed using the penalty function $\sum_i^c (\beta_i - \overline{\beta})^2$, where $\overline{\beta}$ is the mean of all c βs. This causes shrinkage of all parameters toward the mean parameter value. Letting the first category be the reference cell, we use $c - 1$ dummy variables and define $\beta_1 \equiv 0$. For the case $c = 3$ the sum of squares is $2[\beta_2^2 + \beta_3^2 - \beta_2\beta_3]/3$. For $c = 2$ the penalty is $\beta_2^2/2$. If no scale constant is used, this is the same as scaling β_2 with $\sqrt{2}$ × the standard deviation of a binary dummy variable with prevalance of 0.5.

The sum of squares can be written in matrix form as $[\beta_2, \ldots, \beta_c]'$ $(A - B)[\beta_2, \ldots, \beta_c]$, where A is a $c - 1 \times c - 1$ identity matrix and B is a $c - 1 \times c - 1$ matrix all of whose elements are $\frac{1}{c}$. 19

For general penalty functions such as that just described, the penalized log likelihood can be generalized to

$$\log L - \frac{1}{2}\lambda\beta' P\beta. \qquad (9.62)$$

For purposes of using the Newton–Raphson procedure, the first derivative of the penalty function with respect to β is $-\lambda P\beta$, and the negative of the second derivative is λP.

20 Another problem in penalized estimation is how the choice of λ is made. Many authors use cross-validation. A limited number of simulation studies in binary logistic regression modeling has shown that for each λ being considered, at least 10-fold cross-validation must be done so as to obtain a reasonable estimate of predictive accuracy. Even then, a smoother[207] ("super smoother") must be used on the (λ, accuracy) pairs to allow location of the optimum value unless one is careful in choosing the initial sub-samples and uses these same splits throughout. Simulation studies have shown that a modified AIC is not only much quicker to compute (since it requires no cross-validation) but performs better at finding a good value of λ (see below). 21

For a given λ, the effective number of parameters being estimated is reduced because of shrinkage. Gray [237, Eq. 2.9] and others estimate the effective degrees of freedom by computing the expected value of a global Wald statistic for testing association, when the null hypothesis of no association is true. The d.f. is equal to

$$\text{trace}[I(\hat{\beta}^P)V(\hat{\beta}^P)], \qquad (9.63)$$

where $\hat{\beta}^P$ is the penalized MLE (the parameters that maximize Equation 9.61), I is the information matrix computed from ignoring the penalty function, and V is the covariance matrix computed by inverting the information matrix that included the second derivatives with respect to β in the penalty function. 22

Gray [237, Eq. 2.6] states that a better estimate of the variance–covariance matrix for $\hat{\beta}^P$ than $V(\hat{\beta}^P)$ is

$$V^* = V(\hat{\beta}^P)I(\hat{\beta}^P)V(\hat{\beta}^P). \tag{9.64}$$

Therneau (personal communication, 2000) has found in a limited number of simulation studies that V^* underestimates the true variances, and that a better estimate of the variance–covariance matrix is simply $V(\hat{\beta}^P)$, assuming that the model is correctly specified. This is the covariance matrix used by default in the `rms` package (the user can request that the sandwich estimator be used instead) and is in fact the one Gray used for Wald tests.

Penalization will bias estimates of β, so hypothesis tests and confidence intervals using $\hat{\beta}^P$ may not have a simple interpretation. The same problem arises in score and likelihood ratio tests. So far, penalization is better understood in pure prediction mode unless Bayesian methods are used.

Equation 9.63 can be used to derive a modified AIC (see [639, Eq. 6] and [641, Eq. 7]) on the model χ^2 scale:

$$\text{LR } \chi^2 - 2 \times \text{ effective d.f.}, \tag{9.65}$$

where LR χ^2 is the likelihood ratio χ^2 for the penalized model, but ignoring the penalty function. If a variety of λ are tried and one plots the (λ, AIC) pairs, the λ that maximizes AIC will often be a good choice, that is, it is likely to be near the value of λ that maximizes predictive accuracy on a future dataset[g].

Note that if one does penalized maximum likelihood estimation where a set of variables being penalized has a negative value for the unpenalized $\chi^2 - 2 \times$ d.f., the value of λ that will optimize the overall model AIC will be ∞.

As an example, consider some simulated data ($n = 100$) with one predictor in which the true model is $Y = X_1 + \epsilon$, where ϵ has a standard normal distribution and so does X_1. We use a series of penalties (found by trial and error) that give rise to sensible effective d.f., and fit penalized restricted cubic spline functions with five knots. We penalize two ways: all terms in the model including the coefficient of X_1, which in reality needs no penalty; and only the nonlinear terms. The following `R` program, in conjunction with the `rms` package, does the job.

[g] Several examples from simulated datasets have shown that using BIC to choose a penalty results in far too much shrinkage.

```
set.seed(191)
x1 ← rnorm(100)
y  ← x1 + rnorm(100)
pens ← df ← aic ← c(0,.07,.5,2,6,15,60)
all ← nl ← list()

for(penalize in 1:2) {
  for(i in 1:length(pens)) {
    f ← ols(y ~ rcs(x1,5), penalty=
              list(simple=if(penalize==1)pens[i] else 0,
                   nonlinear=pens[i]))
    df[i] ←  f$stats['d.f.']
    aic[i] ← AIC(f)
    nam ← paste(if(penalize == 1) 'all' else 'nl',
                 ' penalty:', pens[i], sep='')
    nam ← as.character(pens[i])
    p ← Predict(f, x1=seq(-2.5, 2.5, length=100),
                 conf.int=FALSE)
    if(penalize == 1) all[[nam]] ← p else nl[[nam]] ← p
  }
  print(rbind(df=df, aic=aic))
}
```

```
        [,1]       [,2]       [,3]      [,4]       [,5]       [,6]
df    4.0000   3.213591   2.706069   2.30273   2.029282   1.822758
aic 270.6653 269.154045 268.222855 267.56594 267.288988 267.552915
          [,7]
df    1.513609
aic 270.805033
        [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
df    4.0000   3.219149   2.728126   2.344807   2.109741   1.960863
aic 270.6653 269.167108 268.287933 267.718681 267.441197 267.347475
          [,7]
df    1.684421
aic 267.892073
```

```
all ← do.call('rbind', all); all$type ← 'Penalize All'
nl  ← do.call('rbind', nl) ; nl$type  ← 'Penalize Nonlinear'
both ← as.data.frame(rbind.data.frame(all, nl))
both$Penalty ← both$.set.
ggplot(both, aes(x=x1, y=yhat, color=Penalty)) + geom_line() +
  geom_abline(col=gray(.7)) + facet_grid(~ type)
# Figure 9.6
```

The left panel in Figure 9.6 corresponds to `penalty = list(simple=a, nonlinear=a)` in the R program, meaning that all parameters except the intercept are shrunk by the same amount `a` (this would be more appropriate had there been multiple predictors). As effective d.f. get smaller (penalty factor gets larger), the regression fits get flatter (too flat for the largest penalties) and confidence bands get narrower. The right graph corresponds to `penalty=list(simple=0, nonlinear=a)`, causing only the cubic spline terms that are nonlinear in X_1 to be shrunk. As the amount of shrinkage increases (d.f. lowered), the fits become more linear and closer to the true regression line (longer dotted line).
23 Again, confidence intervals become smaller.

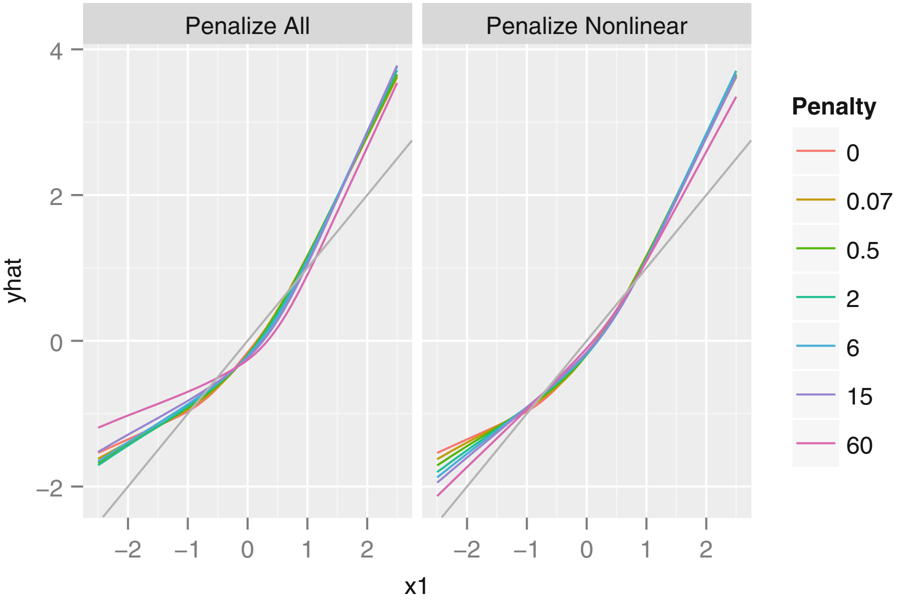

Fig. 9.6 Penalized least squares estimates for an unnecessary five-knot restricted cubic spline function. In the left graph all parameters (except the intercept) are penalized. The effective d.f. are $4, 3.21, 2.71, 2.30, 2.03, 1.82$, and 1.51. In the right graph, only parameters associated with nonlinear functions of X_1 are penalized. The effective d.f. are $4, 3.22, 2.73, 2.34, 2.11, 1.96$, and 1.68.

9.11 Further Reading

$\boxed{1}$ Boos[60] has some nice generalizations of the score test. Morgan et al.[464] show how score test χ^2 statistics may negative unless the *expected* information matrix is used.

$\boxed{2}$ See Marubini and Valsecchi [444, pp. 164–169] for an excellent description of the relationship between the three types of test statistics.

$\boxed{3}$ References [115,507] have good descriptions of methods used to maximize $\log L$.

$\boxed{4}$ As Long and Ervin[426] argue, for small sample sizes, the usual Huber–White covariance estimator should not be used because there the residuals do not have constant variance even under homoscedasticity. They showed that a simple correction due to Efron and others can result in substantially better estimates. Lin and Wei,[410] Binder,[55] and Lin[407] have applied the Huber estimator to the Cox[132] survival model. Freedman[206] questioned the use of sandwich estimators because they are often used to obtain the right variances on the wrong parameters when the model doesn't fit. He also has some excellent background information.

$\boxed{5}$ Feng et al.[188] showed that in the case of cluster correlations arising from repeated measurement data with Gaussian errors, the cluster bootstrap performs excellently even when the number of observations per cluster is large and the number of subjects is small. Xiao and Abrahamowicz[676] compared the cluster bootstrap with a two-stage cluster bootstrap in the context of the Cox model.

6 Graubard and Korn[235] and Fitzmaurice[195] describe the kinds of situations in which the working independence model can be trusted.

7 Minkin,[460] Alho,[11] Doganaksoy and Schmee,[160] and Meeker and Escobar[452] discuss the need for LR and score-based confidence intervals. Alho found that score-based intervals are usually more tedious to compute, and provided useful algorithms for the computation of either type of interval (see also [452] and [444, p. 167]). Score and LR intervals require iterative computations and have to deal with the fact that when one parameter is changed (e.g., b_i is restricted to be zero), all other parameter estimates change. DiCiccio and Efron[157] provide a method for very accurate confidence intervals for exponential families that requires a modest amount of additional computation. Venzon and Moolgavkar provide an efficient general method for computing LR-based intervals.[636] Brazzale and Davison[65] developed some promising and feasible ways to make unconditional likelihood-based inferences more accurate in small samples.

8 Carpenter and Bithell[92] have an excellent overview of several variations on the bootstrap for obtaining confidence limits.

9 Tibshirani and Knight[610] developed an easy to program approach for deriving simultaneous confidence sets that is likely to be useful for getting simultaneous confidence regions for the entire vector of model parameters, for population values for an entire sequence of predictor values, and for a set of regression effects (e.g., interquartile-range odds ratios for age for both sexes). The basic idea is that during the, say, 1000 bootstrap repetitions one stores the -2 log likelihood for each model fit, being careful to compute the likelihood at the current bootstrap parameter estimates but with respect to the *original* data matrix, not the bootstrap sample of the data matrix. To obtain an approximate simultaneous 0.95 confidence set one computes the 0.95 quantile of the -2 log likelihood values and determines which vectors of parameter estimates correspond to -2 log likelihoods that are at least as small as the 0.95 quantile of all -2 log likelihoods. Once the qualifying parameter estimates are found, the quantities of interest are computed from those parameter estimates and an outer envelope of those quantities is found. Computations are facilitated with the `rms` package `confplot` function.

10 van Houwelingen and le Cessie [633, Eq. 52] showed, consistent with AIC, that the average optimism in a mean logarithmic (minus log likelihood) quality score for logistic models is p/n.

11 Schwarz[560] derived a different penalty using large-sample Bayesian properties of competing models. His Bayesian Information Criterion (BIC) chooses the model having the lowest value of $L + 1/2p\log n$ or the highest value of LR $\chi^2 - p\log n$. Kass and Raftery have done several studies of BIC.[337] Smith and Spiegelhalter[576] and Laud and Ibrahim[377] discussed other useful generalizations of likelihood penalties. Zheng and Loh[685] studied several penalty measures, and found that AIC does not penalize enough for overfitting in the ordinary regression case. Kass and Raftery [337, p. 790] provide a nice review of this topic, stating that “AIC picks the correct model asymptotically if the complexity of the true model grows with sample size” and that “AIC selects models that are too big even when the sample size is large.” But they also cite other papers that show the existence of cases where AIC can work better than BIC. According to Buckland et al.,[80] BIC “assumes that a true model exists and is low-dimensional.”

Hurvich and Tsai[314, 315] made an improvement in AIC that resulted in much better model selection for small n. They defined the corrected AIC as

$$\mathrm{AIC}_C = \mathrm{LR}\ \chi^2 - 2p[1 + \frac{p+1}{n-p-1}]. \tag{9.66}$$

In [314] they contrast asymptotically efficient model selection with AIC when the true model has infinitely many parameters with improvements using other indexes such as AIC$_C$ when the model is finite.

One difficulty in applying the Schwarz, AIC$_C$, and related criteria is that with censored or binary responses it is not clear that the actual sample size n should be used in the formula.

12 Goldstein,[222] Willan et al.,[669] and Royston and Thompson[534] have nice discussions on comparing non-nested regression models. Schemper's method[549] is useful for testing whether a set of variables provides significantly greater information (using an R^2 measure) than another set of variables.

13 van Houwelingen and le Cessie [633, Eq. 22] recommended using $L/2$ (also called the Kullback–Leibler error rate) as a quality index.

14 Schemper[549] provides a bootstrap technique for testing for significant differences between correlated R^2 measures. Mittlböck and Schemper,[461] Schemper and Stare,[554] Korn and Simon,[365, 366] Menard,[454] and Zheng and Agresti[684] have excellent discussions about the pros and cons of various indexes of the predictive value of a model.

15 Al-Radi et al.[10] presented another analysis comparing competing predictors using the adequacy index and a receiver operating characteristic curve area approach based on a test for whether one predictor has a higher probability of being "more concordant" than another.

16 [55, 97, 409] provide good variance–covariance estimators from a weighted maximum likelihood analysis.

17 Huang and Harrington[310] developed penalized partial likelihood estimates for Cox models and provided useful background information and theoretical results about improvements in mean squared errors of regression estimates. They used a bootstrap error estimate for selection of the penalty parameter.

18 Sardy[538] proposes that the square roots of the diagonals of the inverse of the covariance matrix for the predictors be used for scaling rather than the standard deviations.

19 Park and Hastie[483] and articles referenced therein describe how quadratic penalized logistic regression automatically sets coefficient estimates for empty cells to zero and forces the sum of k coefficients for a k-level categorical predictor to equal zero.

20 Greenland[241] has a nice discussion of the relationship between penalized maximum likelihood estimation and mixed effects models. He cautions against estimating the shrinkage parameter.

21 See[310] for a bootstrap approach to selection of λ.

22 Verweij and van Houwelingen [639, Eq. 4] derived another expression for d.f., but it requires more computation and did not perform any better than Equation 9.63 in choosing λ in several examples tested.

23 See van Houwelingen and Thorogood[631] for an approximate empirical Bayes approach to shrinkage. See Tibshirani[608] for the use of a non-smooth penalty function that results in variable selection as well as shrinkage (see Section 4.3). Verweij and van Houwelingen[640] used a "cross-validated likelihood" based on leave-out-one estimates to penalize for overfitting. Wang and Taylor[652] presented some methods for carrying out hypothesis tests and computing confidence limits under penalization. Moons et al.[462] presented a case study of penalized estimation and discussed the advantages of penalization.

Table 9.4 Likelihood ratio global test statistics

Variables in Model	$LR\ \chi^2$
age	100
sex	108
age, sex	111
age^2	60
age, age^2	102
age, age^2, sex	115

9.12 Problems

1. A sample of size 100 from a normal distribution with unknown mean and standard deviation (μ and σ) yielded the following log likelihood values when computed at two values of μ.

$$\log L(\mu = 10, \sigma = 5) = -800$$
$$\log L(\mu = 20, \sigma = 5) = -820.$$

What do you know about μ? What do you know about $\overline{Y}$?

2. Several regression models were considered for predicting a response. $LR\ \chi^2$ (corrected for the intercept) for models containing various combinations of variables are found in Table 9.4. Compute all possible meaningful $LR\ \chi^2$. For each, state the d.f. and an approximate P-value. State which $LR\ \chi^2$ involving only one variable is not very meaningful.
3. For each problem below, rank Wald, score, and LR statistics by overall statistical properties and then by computational convenience.
   a. A forward stepwise variable selection (to be later accounted for with the bootstrap) is desired to determine a concise model that contains most of the independent information in all potential predictors.
   b. A test of independent association of each variable in a given model (each variable adjusted for the effects of all other variables in the given model) is to be obtained.
   c. A model that contains only additive effects is fitted. A large number of potential interaction terms are to be tested using a global (multiple d.f.) test.
4. Consider a univariate saturated model in 3 treatments (A, B, C) that is quadratic in age. Write out the model with all the βs, and write in detail the contrast for comparing treatment B with treatment C for 30 year olds. Sketch out the same contrast using the “difference in predictions” approach without simplification.

5. Simulate a binary logistic model for $n = 300$ with an average fraction of events somewhere between 0.15 and 0.3. Use 5 continuous covariates and assume the model is everywhere linear. Fit an unpenalized model, then solve for the optimum quadratic penalty λ. Relate the resulting effective d.f. to the 15:1 rule of thumb, and compute the heuristic shrinkage coefficient $\hat{\gamma}$ for the unpenalized model and for the optimally penalized model, inserting the effective d.f. for the number of non-intercept parameters in the model.
6. For a similar setup as the binary logistic model simulation in Section 9.7, do a Monte Carlo simulation to determine the coverage probabilities for ordinary Wald and for three types of bootstrap confidence intervals for the true x=5 to x=1 log odds ratio. In addition, consider the Wald-type confidence interval arising from the sandwich covariance estimator. Estimate the non-coverage probabilities in both tails. Use a sample size $n = 200$ with the single predictor x_1 having a standard log-normal distribution, and the true model being $\text{logit}(Y = 1) = 1 + x_1/2$. Determine whether increasing the sample size relieves any problem you observed. Some R code for this simulation is on the web site.