**Section 2.2.1 – Parameter estimation (continued)**

Variance-Covariance matrix

The estimated variance-covariance matrix for  …,  has the usual form:



Thus, the (1,1) element is the estimated variance of , the (1,2) element is the estimate covariance of  and  … . I will more simply call this matrix a “covariance matrix” for the remainder of the notes.

What is a matrix?

A matrix consists of elements assembled into rows and columns. For example,



is a 2×3 matrix. Arithmetic and algebra can be performed with matrices, just like with single numerical values. Please see my multivariate course notes for an introduction to matrix algebra if desired.

The use of matrix algebra for our course will be limited as much as possible. You are responsible only for understanding the structure of a matrix, rather than to be able to perform matrix calculations yourself. Functions in R will automatically perform these calculations for us.

This matrix is found through using the same likelihood based methods discussed in Chapter 1.

If  is the maximum likelihood estimator for θ, we can say that it has an approximate normal distribution with mean θ and variance

 

for a large sample.

For the logistic regression model, we need to account for not only 1 MLE, but for p + 1 MLEs. This leads to a matrix of second partial derivatives (a Hessian matrix) of the log-likelihood function evaluated at the parameter estimates. For example, if there are only parameters β0 and β1, the matrix is



In the end, the matrix in general ends up having the nice form of  where



and  is a n×n matrix with  on the diagonal and 0’s elsewhere.

Example: Placekicking (Placekick.R, Placekick.csv)

This example focused on the model with distance only.

The vcov() function produces the estimated covariance matrix:

> vcov(mod.fit)

 (Intercept) distance

(Intercept) 0.10645675 -2.606250e-03

distance -0.00260625 6.953996e-05

> vcov(mod.fit)[2,2] # Var-hat(beta-hat\_1)

[1] 6.953996e-05

From the summary(mod.fit) output earlier, we have

> summary(object = mod.fit)

Call: glm(formula = good ~ distance, family = binomial(link = logit), data = placekick)

Deviance Residuals:

 Min 1Q Median 3Q Max

 -2.7441 0.2425 0.2425 0.3801 1.6092

Coefficients:

 Estimate Std. Error z value Pr(>|z|)

(Intercept) 5.812080 0.326277 17.81 <2e-16 \*\*\*

distance -0.115027 0.008339 -13.79 <2e-16 \*\*\*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

 Null deviance: 1013.43 on 1424 degrees of freedom

Residual deviance: 775.75 on 1423 degrees of freedom

AIC: 779.75

Number of Fisher Scoring iterations: 6

If we square the items in the “Std. Error” column of the output, we obtain the corresponding variances:

> 0.326277^2

[1] 0.1064567

> 0.008339^2

[1] 6.953892e-05

If we wanted to go through the actual matrix calculations ourselves (not usually of interest!), we obtain the same matrix:

> pi.hat <- mod.fit$fitted.values

> V <- diag(pi.hat\*(1-pi.hat))

> X <- cbind(1, placekick$distance)

> solve(t(X)%\*%V%\*%X)

 [,1] [,2]

[1,] 0.10645678 -2.606250e-03

[2,] -0.00260625 6.953997e-05

Binomial data format

So far, our data has consisted of Y1, …, Yn responses from Bernoulli distributions with parameters πi for i = 1, …, n. Alternatively, we may actually have W1, …, WJ responses each from binomial distributions with nj trials and probability of success πj where j = 1, …, J. Note that I am using j’s here to help differentiate this notation from the Bernoulli observation notation.

The likelihood function is



The log likelihood function is



This is the same likelihood function as if we used a Bernoulli form of the data (think of each wj as being the sum of particular Yi’s which have the exact same explanatory variable values) EXCEPT for . Because this additional term is constant for all possible values of our parameters (there is no πj) in it, we will still obtain the same parameter estimates as if we worked with a Bernoulli form of the data.

Note that the maximized value of the log likelihood function will be different due to . Also, we will see in Chapter 5 that there will be differences when evaluating how well the model fits the data.

Example: Placekicking (placekick.R, placekick.csv)

To illustrate that the Bernoulli and binomial forms of a data set result in the same parameter estimates, I am going to first transform the placekicking data set to a binomial form. For this example, I will only use distance as the explanatory variable.

The aggregate() function is used to find the number of successes (wj) and number of observations for each distance (nj):

> w <- aggregate(x = good ~ distance, data = placekick, FUN = sum)

> n <- aggregate(x = good ~ distance, data = placekick, FUN = length)

> w.n <- data.frame(distance = w$distance, success = w$good, trials = n$good, proportion = round(w$good/n$good, 4))

> head(w.n)

 distance success trials proportion

1 18 2 3 0.6667

2 19 7 7 1.0000

3 20 776 789 0.9835

4 21 19 20 0.9500

5 22 12 14 0.8571

6 23 26 27 0.9630

For example, there are 2 successes out of 3 trials at a distance of 18 yards, which results in an observed proportion of successes of 2/3 ≈ 0.6667. Note that the reason for the large number of observations at 20 yards is because most PATs were attempted from this distance due to the NFL rules at that time.

Below is the code used to estimate the model:

> mod.fit.bin <- glm(formula = success/trials ~ distance,

 weight = trials, family = binomial(link = logit), data

 = w.n)

> summary(mod.fit.bin)

Call: glm(formula = success/trials ~ distance, family = binomial(link = logit), data = w.n, weights = trials)

Deviance Residuals:

 Min 1Q Median 3Q Max

-2.0373 -0.6449 -0.1424 0.5004 2.2758

Coefficients:

 Estimate Std. Error z value Pr(>|z|)

(Intercept) 5.812080 0.326277 17.81 <2e-16 \*\*\*

distance -0.115027 0.008339 -13.79 <2e-16 \*\*\*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

 Null deviance: 282.181 on 42 degrees of freedom

Residual deviance: 44.499 on 41 degrees of freedom

AIC: 148.46

Number of Fisher Scoring iterations: 5

The estimated model is the same as before:



All 2×2 contingency tables given earlier can be rewritten in a data frame format so that a logistic regression model can be fit to them!