**Section 2.2.7 – Convergence of parameter estimates**

The glm() function uses IRLS until convergence is obtained or until the maximum number of iterations are reached. To determine convergence, glm() does not look at the successive estimates of the regression parameters directly, rather it examines the residual deviance. If we let G(k) denote the residual deviance at iteration k, then convergence occurs when



where ε is some specified small number greater than 0. The numerator provides a measure to determine if the  for i = 1, …, n are changing much from one iteration to the next. The denominator helps to take into account the relative size of the residual deviance.

The glm() function provides a few ways to control how convergence decided:

* The epsilon argument sets ε above. The default is epsilon = 10^(-8).
* The maxit argument states the maximum number of iterations allowed for the numerical procedure. The default is maxit = 25.
* The trace argument value can be used to see the actual G(k) values for each iteration. The default is trace = FALSE (do not show these values).

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

Consider the model with only distance as the explanatory variable:



Below are the results from using the glm() to estimate the model and from including the three arguments controlling convergence. Note that these three argument values were chosen for illustrative purposes only.

mod.fit <- glm(formula = good ~ distance, family =

binomial(link = logit), data = placekick, trace = TRUE,

epsilon = 0.0001, maxit = 50)

Deviance = 836.7715 Iterations - 1

Deviance = 781.1072 Iterations - 2

Deviance = 775.8357 Iterations - 3

Deviance = 775.7451 Iterations - 4

Deviance = 775.745 Iterations - 5

> mod.fit$control

$epsilon

[1] 1e-04

$maxit

[1] 50

$trace

[1] TRUE

> mod.fit$converged

[1] TRUE

The convergence criteria value for iteration k = 5 is



which is less than the stated ε = 0.0001, so the iterative numerical procedure stopped. For iteration k = 4, the convergence criteria value is 0.00012, which is greater than 0.0001, so this is why the procedure continued.

If the value for maxit was changed to 3, the message

Warning message:

glm.fit: algorithm did not converge

would be printed to warn you that convergence was not obtained. Of course, you would NOT use the parameter estimates in this situation!

What if convergence is not obtained?

Try a larger number of iterations!

If this does not work, there may be some fundamental problem with the data making the iterative numerical procedures not suitable. The most common problem occurs when an explanatory variable(s) perfectly separates the data between y = 0 and 1 values; this is often referred to as complete separation. In addition to a convergence warning message, another warning message that glm() will likely give for this situation is

glm.fit: fitted probabilities numerically 0 or 1 occurred

This message corresponds to  = 0 or 1 for some observations. Please see the plot in the next example if it is not clear why this happens.

It is important to note that the previous message of

glm.fit: fitted probabilities numerically 0 or 1 occurred

can be given in situations when there is not a problem. It could be that some estimated probabilities are just very close to 0 or 1. If you are unsure, check the frequency of these estimated probability values. Also, decrease ε and check if the parameter estimates change for a different number of iterations.

Example: Complete separation (Non-convergence.R)

Consider a simple data set with one explanatory variable x1 that is less than 6 when y = 0 and greater than or equal to 6 when y = 1. Because x1 perfectly separates out the two possible values of y, complete separation occurs. Below is the corresponding R code and output:

> set1 <- data.frame(x1 = c(1,2,3,4,5,6,7,8,9,10), y =

c(0,0,0,0,0,1,1,1,1,1))

> set1

x1 y

1 1 0

2 2 0

3 3 0

4 4 0

5 5 0

6 6 1

7 7 1

8 8 1

9 9 1

10 10 1

> mod.fit1 <- glm(formula = y ~ x1, data = set1, family = binomial(link = logit), trace = TRUE)

Deviance = 4.270292 Iterations - 1

Deviance = 2.574098 Iterations - 2

<OUTPUT IS EDITED>

Deviance = 7.864767e-10 Iterations - 25

Warning messages:

1: glm.fit: algorithm did not converge

2: glm.fit: fitted probabilities numerically 0 or 1 occurred

> mod.fit1$coefficients

(Intercept) x1

-245.84732 44.69951

R indicates that both convergence did not occur and at least some estimates of π are 0 or 1. Below is a plot (left side) of the data and the model at iteration #25:



Because there is a separation between the y = 0 and 1 values, the slope of the line between x = 5 and 6 will continue to get larger as the iterations continue. Essentially, the β1 estimate is going to infinity with continued iterations. Notice this means the estimate is VERY biased.

Interestingly, R indicates “convergence” after 26 iterations if you increase maxit! However, the same

glm.fit: fitted probabilities numerically 0 or 1 occurred

message will occur. You should not use logistic regression here because the parameter estimates will continue to change for a larger number of iterations. Try this yourself with a larger maxit and smaller epsilon.

By reversing the y values at x1 = 5 and 6, we obtain model convergence in 6 iterations (not shown here). The right (set2) plot shows the data and the final model. The slope of the model is now not as great as was before.

What can you do if complete separation occurs?

Heinze and Schemper (2002) outline a number of possible options. The most desirable options are

* Exact logistic regression – The exact distribution of the estimators of the regression parameters is used through the use of computational intensive algorithms. A section in a later chapter of my book discusses exact inference methods.
* Modify the likelihood function – Because the likelihood function increases without bound during the iterative numerical procedure, this function can be modified to potentially prevent the problems from happening. One approach is to include a “penalty” in the likelihood function as proposed by Firth (1993). Details are given in the corresponding section of my book.

Final comments:

* Complete separation is not necessarily bad if you want to distinguish between the response levels of y. The problem is that the model estimated by maximum likelihood does not provide a good way to interpret the relationship between y and the explanatory variables.
* It can be difficult to see complete separation graphically if there is more than one explanatory variable.
* There may be times when the glm() function does not provide a warning. When parameter estimates are very large or very small with large estimated standard deviations, this is sign that complete separation may exist. These types of parameter estimates can then lead to many observations with estimated probabilities of success close to 0 or 1.
* An often used solution to complete separation is to add a pseudo observation to the data in an attempt to obtain convergence of the parameter estimates. This is somewhat similar to what was done for odds ratios with a 0 cell count present within a contingency table. We recommend using the exact logistic regression or penalized likelihood approaches instead.