Logistic and Probit Regression Computational Algorithms

Authored by: Kyle Kolsti, PhD

Kaity Jones

Alex McBride

6 August 2021

HSCoBP@afit.edu
937-255-3636 x 4768

The HS CoBP core mission involves leading a consortium of government, industry, and academic experts to assess future homeland threats to inform strategic plans across nine DHS T&E capability areas.

Table of Contents

Executive Summary

This report provides the computational steps to employ logistic regression and probit regression under the construct of a generalized linear model (GLM). Matrix equations and pseudo-code facilitate the development of scripts to determine the coefficients of maximum likelihood through Newton-Raphson iteration method.

Keywords: logistic regression, maximum likelihood, Newton-Raphson

Introduction

Many systems encountered in the Department of Homeland Security (DHS) are black box systems which provide only binary response data to the test team during Test and Evaluation (T&E). Logistic regression is a widely-used method to analyze binary response data that provides the probability of observing one of the two response values given certain values of a continuous factor (Natoli, Burke, & Oimoen, 2020). A common example is a sensor where the probability of detection depends on the range to the item of interest. Probit regression is a related method applicable to the same scenario that may be encountered is some fields such as medical research. Both logistic and probit regression can be employed under the construct of a generalized linear model (GLM) which is available in most statistical analysis software packages. On occasion, a test team may need to script these methods rather than relying on commercially available software products. The following best practice provides step by step computations for practitioners to improve understanding of these important analytical methods and to implement them in custom code if necessary.

Logistic Regression Development

Data Structure

Data in a system with a binary outcome can be expressed as a vector of n binary outcomes $\boldsymbol{Y}$ ($Y_i = 0$ or 1). For this report we will define $Y_i = 1$ as a successful outcome such as a detection. Each outcome Y_i was observed at a value of the single predictor variable, the continuous factor x. (The matrix methods developed here easily extend to multiple factors if needed.) The $n \times 2$ design matrix $\boldsymbol{X}$ contains 1's in the first column and the values of x used in the trials in the second column. Thus the first trial was performed with x set to x_1 and the observed outcome was Y_1, and so forth.

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}, \quad \boldsymbol{Y} = \begin{Bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{Bmatrix} \tag{1}$$

Model

To model binary response data, we begin by defining $\pi_i = \pi(x_i)$ as the prediction of the probability of observing $Y = 1$ at $x = x_i$. We also define two model parameters β_i such that

$$y_i = y(x_i) = \beta_0 + \beta_1 x_i \tag{2}$$

In matrix form, $\boldsymbol{y} = \boldsymbol{X\beta}$ where the parameter vector $\boldsymbol{\beta} = \{\beta_0, \beta_1\}^T$. The generalized linear model (GLM) relates π to y through a function called the link function. For binary responses, two common link functions are the logit link function, based on the odds, and the probit link function, which is based on the normal distribution (Agresti, 2002). The link function g and inverse link function g^{-1} for logit and probit regression are shown below in Table 1. The symbol Φ is the cumulative distribution function for the standard normal distribution, which has a mean of zero and a standard deviation of one.

Table 1: Link and inverse link functions

Link type	**Link Function** $y = g(\pi)$	**Inverse Link Function** $\pi = g^{-1}(y)$
Logit	$y_i = \log\left(\frac{\pi_i}{1-\pi_i}\right)$	$\pi_i = \frac{1}{1+e^{-y_i}}$
Probit	$y_i = \Phi^{-1}(\pi_i)$	$\pi_i = \Phi(y_i)$

Likelihood

There is no closed-form solution to determine the parameters $\boldsymbol{\beta}$ in logistic or probit regression as there is for standard linear regression, so the model best fit must be determined using a nonlinear solver with the Maximum Likelihood Estimate (MLE) method. The likelihood of a pair of parameter values $\boldsymbol{\beta}$ is the product of the probabilities of obtaining each of the observed outcomes in $\boldsymbol{Y}$. For example, for a single trial at some value of x, if $\pi(\boldsymbol{\beta}) = 0.7$ then the probability of obtaining $Y = 1$ is 0.7 and the probability of obtaining $Y = 0$ is 0.3; thus if actual observation were $Y = 1$, the likelihood of $\boldsymbol{\beta}$ for that trial is 0.7. The formula for the likelihood of a pair of parameter values $\boldsymbol{\beta}$ as a product of the probabilities for each trial is

$$\prod_{i=1}^{n} \pi_i^{Y_i}(1-\pi_i)^{1-Y_i} \tag{3}$$

For better computational behavior we will deal with the natural logarithm or "log likelihood." Recall that since Y_i is either 0 or 1, one of the two terms inside the summation will vanish for each data point.

$$L(\beta_0, \beta_1) = \sum_{i=1}^{n} \{\, Y_i \log(\pi_i) + (1 - Y_i) \log(1 - \pi_i) \,\} \tag{4}$$

Now we can fit the statistical model to the data using MLE; i.e., find β_0 and β_1 such that the log likelihood L is at its maximum value. There is a wealth of numerical methods that can be used to minimize $-L$, which provides the same solution as maximizing L. These methods, which include the conjugate gradient method, steepest descent method, BFGS (Broyden–Fletcher–Goldfarb–Shanno algorithm, and differential evolution, are available for most scripting languages. As an alternative, the following section will provide a root-finding method which may be computationally faster than these minimization procedures.

Firth Bias Correction

If there is any value of x below which all the Y values are 0 and above which all the Y values are 1 (or vice versa), the likelihood function does not have a maximum value and the iterative procedure therefore fails. This behavior is what is known as complete separation (Zorn, 2005). An example of a data set which has complete separation is shown in Figure 1.

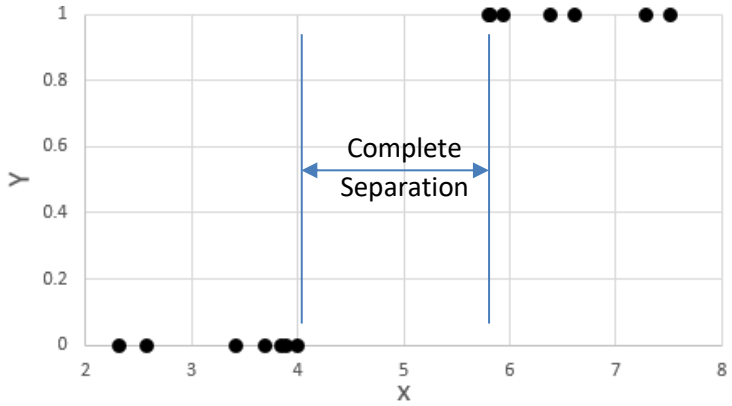

Figure 1: Example data set exhibiting complete separation.

The Firth correction is a commonly used and effective way to address this situation. It is performed by adding an additional term to the log likelihood equation. The term equals one half of the natural logarithm of the determinant of the expected Fisher Information Matrix, $\boldsymbol{I}$ (Zorn, 2005), which will be defined in subsequent sections of this paper.

$$L(\boldsymbol{\beta}) = \sum_{i=1}^{n} \{ Y_i \log(\pi_i) + (1 - Y_i) \log(1 - \pi_i) \} + \frac{1}{2} \log|\boldsymbol{I}(\boldsymbol{\beta})| \tag{5}$$

Determining Parameters

Newton Iterations

This section will provide the step by step instructions for fitting the GLM using the Newton-Raphson method (Agresti, 2002). Like standing at the top of a hill, the maximum likelihood occurs where the slope in every direction is zero. Mathematically, the solution for $\boldsymbol{\beta}$ is where the partial derivatives of the

log likelihood function with respect to the parameters β_0 and β_1 are all equal to zero. The vector of the partial derivatives is called the gradient vector. Since there are two parameters in $\boldsymbol{\beta}$, the gradient vector is of shape 2×1.

$$\boldsymbol{\nabla L}(\beta_0, \beta_1) = \begin{Bmatrix} \dfrac{\partial L}{\partial \beta_0} \\ \dfrac{\partial L}{\partial \beta_1} \end{Bmatrix} \tag{6}$$

If a guess for $\boldsymbol{\beta}$ does not satisfy $\boldsymbol{\nabla L}(\boldsymbol{\beta}) = \mathbf{0}$, an update has to be made to $\boldsymbol{\beta}$ to nudge it closer to the solution. This update requires computation of the symmetric 2×2 Hessian matrix, $\boldsymbol{H}$. The Hessian describes the local curvature of the surface, as indicated by the second partial derivatives that comprise it.

$$\boldsymbol{H}(\beta_0, \beta_1) = \begin{bmatrix} \dfrac{\partial^2 L}{\partial \beta_0^2} & \dfrac{\partial^2 L}{\partial \beta_0 \partial \beta_1} \\ \dfrac{\partial^2 L}{\partial \beta_0 \partial \beta_1} & \dfrac{\partial^2 L}{\partial \beta_1^2} \end{bmatrix} \tag{7}$$

Table 2 contains the steps for the Newton-Raphson iterations. The gradient vector and Hessian matrix are fully defined in the next section. After obtaining the solution, the model fit may be assessed using the methods detailed in (Allison, 2014).

Table 2: Pseudo-code for Newton-Raphson iterations

1. Set $k = 0$.
2. Start with initial guess for the vector of parameters, $\boldsymbol{\beta}_k$. It is usually sufficient to make an initial guess of $\boldsymbol{\beta}_0 = \{0,0\}^T$.
3. Calculate the gradient vector, $\boldsymbol{\nabla L}(\boldsymbol{\beta}_k)$
4. Calculate the Hessian, $\boldsymbol{H}(\boldsymbol{\beta}_k)$.
5. Calculate the update vector for the parameters: $\Delta\boldsymbol{\beta} = -\boldsymbol{H}^{-1}\boldsymbol{\nabla L}$. It is better computationally to avoid inverting the Hessian matrix and instead solve the linear system:
$$\boldsymbol{H}\Delta\boldsymbol{\beta} = -\boldsymbol{\nabla L}$$
6. Update the parameter vector to the latest guess. $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \Delta\boldsymbol{\beta}$
7. Check convergence. One way to declare convergence is when the largest change magnitude in β (hence biggest absolute value in $\Delta\boldsymbol{\beta}$) was smaller than a tolerance. Another way is to check that the L_2 norm of the residual vector $\boldsymbol{\nabla L}$ is below a preset tolerance; i.e., both values of the residual vector are nearly zero.
    a. If converged: STOP.
    b. If not converged: $k = k + 1$; Go to Step 3 and perform another Newton iteration.

Matrix Formulas

This section provides the recipes for the matrices used in the Newton iteration algorithm. For reference the matrix sizes are $\boldsymbol{X}_{n\times 2}$, $\boldsymbol{Y}_{n\times 1}$, $\boldsymbol{\pi}_{n\times 1}$, $\boldsymbol{Y}_{n\times 1}$, $\boldsymbol{H}_{2\times 2}$, $\boldsymbol{\nabla L}_{2\times 1}$, $\boldsymbol{W}_{n\times n}$, and $\boldsymbol{Z}_{n\times n}$ where n is the number of trials that were conducted. The constituent matrices are calculated using differently for logistic and probit regression, as shown in Table 4. Note that the function f is the standard normal probability distribution function ($\mu = 0, \sigma = 1$). The formula for $\boldsymbol{Z}$ may be found in (Agresti, 2002), Problem 6.32.

Table 3: Definitions of the matrices $\boldsymbol{W}$ and $\boldsymbol{Z}$.

Link Function	Matrix Definitions
Logistic	$\boldsymbol{W} = \text{diag}\{\pi_i(1-\pi_i)\}$ $\boldsymbol{Z} = \text{identity}(n \times n)$
Probit	$\boldsymbol{W} = \text{diag}\left\{\frac{[f(y_i)]^2}{\pi_i(1-\pi_i)}\right\}$ $\boldsymbol{Z} = \text{diag}\left\{\frac{f(y_i)}{\pi_i(1-\pi_i)}\right\}$

The Hessian matrix is calculated using Equation 8. In truth the matrix is different depending upon whether the Firth correction is used; however, convergence to the solution is not significantly affected either way, so for simplicity Equation 8 will be used in either case.

$$\boldsymbol{H} = -\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} \tag{8}$$

The gradient vector is calculated using Equation 9,

$$\text{grad} = \boldsymbol{\nabla L} = \boldsymbol{X}^T\boldsymbol{Z}(\boldsymbol{Y} - \boldsymbol{\pi}) + \boldsymbol{\nabla L}_{Firth} \tag{9}$$

If not using Firth bias correction, the Firth gradient correction vector $\boldsymbol{\nabla L}_{Firth}$ vanishes. If using the Firth bias correction, $\boldsymbol{\nabla L}_{Firth}$ is calculated using Equations 10 through 12. The Fisher Information Matrix $\boldsymbol{I}$ is defined in Equation 10. To take the square root of the diagonal matrix $\boldsymbol{W}$, simply take the square root of each element along the diagonal. The bolded **0.5** indicates a column vector of size $n \times 1$ with all elements equal to 0.5.

$$\boldsymbol{I} = -\boldsymbol{H} = \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X} \tag{10}$$

$$\widehat{\boldsymbol{H}} = \boldsymbol{W}^{1/2}\boldsymbol{X}\boldsymbol{I}^{-1}\boldsymbol{X}^T\boldsymbol{W}^{1/2} \tag{11}$$

$$\nabla \boldsymbol{L}_{Firth} = \boldsymbol{X}^T \widehat{\boldsymbol{H}}(\boldsymbol{0.5} - \boldsymbol{\pi}) \tag{12}$$

Uncertainty

Uncertainty is captured by the 2×2 covariance matrix $\boldsymbol{\Sigma}$, which is the inverse of the observed Fisher Information Matrix, $\boldsymbol{I}$.

$$\boldsymbol{\Sigma} = \boldsymbol{I}^{-1} = (-\boldsymbol{H})^{-1} \tag{13}$$

Because of a special property of the logit link function (it is called "canonical"), the observed Hessian is the same as the expected Hessian we have used up to this point (Cao, 2013). Therefore, the covariance matrix is easy to compute using Equation 10. Unfortunately, the probit link is not canonical and for the purposes of the covariance matrix, Equation 10 cannot be used. For more a more accurate probit covariance matrix, one technique is to employ numerical approximations of the second partial derivatives of the log likelihood function to approximate the Hessian. This procedure is computationally expensive; fortunately, it only must be done once and the matrix is only 2×2 so the expense is reasonable in this case.

The standard deviations of the MLE estimates of β_0 and β_1 are the square roots of the diagonal of the covariance matrix.

The confidence intervals on the predicted probability of success π at a set of desired prediction points $\boldsymbol{x}_p$ can be calculated using the covariance matrix. Assume an $m \times 2$ prediction design matrix $\boldsymbol{X}_p$ where m is the number of points at which to calculate the confidence limits (it may be a dense grid in x for plotting purposes). As before, the first column of $\boldsymbol{X}_p$ is all ones; the second column is the x coordinates of interest, $\boldsymbol{x}_p$. The $m \times 1$ vector of standard error, $\boldsymbol{SE}$, is

$$\boldsymbol{SE} = \sqrt{\text{diag}\left(\boldsymbol{X}_p \boldsymbol{\Sigma} \boldsymbol{X}_p^T\right)} \tag{14}$$

Calculate $\hat{y}_j = \hat{\beta}_0 + \hat{\beta}_1 x_{pj}$ at the m prediction points with the best fit parameters from the Newton-Raphson MLE solution $\widehat{\boldsymbol{\beta}}$. (In matrix form the equation is $\widehat{\boldsymbol{y}} = \boldsymbol{X}_p \widehat{\boldsymbol{\beta}}$). Calculate the z value based on the desired confidence using the standard normal distribution – for example, z = 1.28 for 95% confidence and z = 1.96 for 95% confidence (2-sided interval). The lower and upper confidence intervals are computed using the link function defined in Table 1.

$$\begin{aligned} \text{LCL}_j &= g^{-1}\left(\hat{y}_j - zSE_j\right) \\ \text{UCL}_j &= g^{-1}\left(\hat{y}_j + zSE_j\right) \end{aligned} \tag{15}$$

The resulting confidence limits should be in the range 0 to 1 except for round-off errors which should be controlled using min/max statements.

Example

This example is provided for code verification. The fabricated data consists of 12 observations ($n = 12$). Figure 2 shows a plot of the predicted probability $\pi(x)$ and its 80% confidence intervals for logistic regression, logistic regression with the Firth bias correction, and probit regression.

X values: 0.8, 0.9, 1.2, 1.7, 1.8, 1.9, 2.0, 2.1, 2.7, 2.9, 3.3, 3.3

Y values: 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1

```
link   Firth?  -loglike    params and SE            80% CI at x = 2.0
-----  -----   ---------  ------------------------  -----------------
logit  NO      5.9328627  -3.3824460 +/- 2.2652052  [0.37812,0.79843]
                           1.9109745 +/- 1.1540737
logit  YES     6.05890473 -2.433902  +/- 1.9137435  [0.37063,0.76422]
                           1.3785659 +/- 0.9381849
probit NO      5.85762923 -2.099632  +/- 1.3043286  [0.3919, 0.79961]
                           1.1912808 +/- 0.66827
```

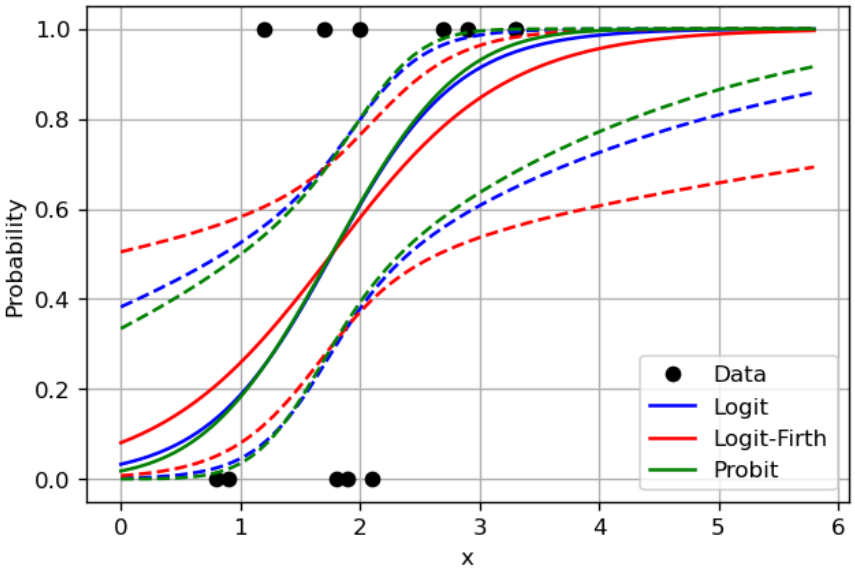

Figure 1: Graph of the MLE solutions (solid lines) and corresponding confidence intervals (dashed lines) for logistic regression with and without Firth correction and probit regression.

Conclusion

Logistic and probit regression are effective tools for predicting the probability of success for a system when the probability changes with a continuous variable, x; for example, a probability of detection that depends on the range to the item of interest. This paper provided the steps for constructing the matrices and implementing Newton-Raphson iterations to obtain a solution based on the test data. Formulas were also provided for estimating the uncertainty in the solution's parameters and in the probability of success at any value of x. These recipes can be applied in scripts to automate analysis of a large number of data sets.

Works Cited

Agresti, A. (2002). *Categorical Data Analysis.* John Wiley & Sons, Inc.

Allison, P. (2014). Measures of Fit for Logistic Regression. *Proceedings of the SAS® Global Forum 2014 Conference. Paper 1485-2014.* Cary, NC: SAS Institute Inc.

Cao, X. (2013). *Relative Performance of Expected and Observed Fisher Information in Covariance Estimation for Maximum Likelihood Estimates.* Dissertation, Johns Hopkins University.

Natoli, C., Burke, S., & Oimoen, S. (2020). *Categorical Data in a Designed Experiment Part 3: Logistic Regression.* Scientific Test and Analysis Techniques (STAT COE) Report 10-2020.

Zorn, C. (2005). A Solution to Separation in Binary Response Models. *Political Analysis, 13*(2), 157-170. doi:10.1093/pan/mpi009