

APPENDIX B

LOGISTIC REGRESSION MODEL

Maximum Likelihood Estimation

To estimate the logistic regression parameters $\beta_0, \beta_1, \beta_2, \dots, \beta_n$, the method that was used in this study was the maximum likelihood estimation. Likelihood means the probability has been evaluated as a function of the parameters with the data fixed (Dowdy et al., 2004). Likelihood allows the estimation of *unknown parameters* based on *known outcomes*. In a sense, likelihood can be thought a reversed version of conditional probability. For example, given parameter β , then the conditional probability of condition x is $P(x|\beta)$. It is also called the posterior probability because it is derived from or depends on the specified value of β (Myung, 2002). Reversing this reasoning, the likelihood function can be constructed: given outcome x , use the likelihood function $L(\beta|x)$ to reason parameter β . This is formalized in Bayes' theorem which states:

$$P(\beta|x) = \frac{P(x|\beta)P(\beta)}{P(x)} \quad (\text{B1})$$

The calculation of the likelihood estimators is simplified by two shortcuts:

- The joint probability of all the observations is the product of the probability function for each observation.
- Maximizing the log likelihood produces the same result as maximizing the likelihood. The log likelihood is the sum of the logarithms of the probabilities. Finding the maximum-likelihood estimators is the same as minimizing the negative sum of logs of the probabilities attributed to the response levels that actually occurred for each observation.

The estimates a and b in the case of simple linear regression are, in fact, maximum-likelihood estimators of α and β because minimizing the negative sum the logs of the probabilities produces the same function as the least-squares method.

Log-likelihood equations

If each response is coded as $Y_i = 0$ or 1 , and let the x_i represent the variable, the contribution to an observation to the likelihood is

$$\pi(x_i)^{y_i}[1 - \pi(x_i)]^{1-y_i} \quad (\text{B2})$$

Since the observations are independent, the likelihood of all the observations is the product of each contribution

$$l(\alpha, \beta) = \prod_{i=1}^n \pi(x_i)^{y_i}[1 - \pi(x_i)]^{1-y_i} \quad (\text{B3})$$

and the log likelihood is

$$L(\alpha, \beta) = \sum_i^n \{y_i \log_e[\pi(x_i)] + (1 - y_i) \log_e[1 - \pi(x_i)]\} \quad (\text{B4})$$

To find the values of α and β that maximize $L(\alpha, \beta)$, we differentiate $L(\alpha, \beta)$ with respect to α and β and set the resulting equations to zero. These likelihood equations are

$$\sum [y_i - \pi(x_i)] = 0 \quad (\text{B5})$$

and

$$\sum x_i [y_i - \pi(x_i)] = 0 \quad (\text{B6})$$

Unlike linear regression where $L(\alpha, \beta)$ is linear, for logistic regression, $L(\alpha, \beta)$, is nonlinear in α and β and the solutions of the likelihood equations need special methods. One such method is the iterative Newton-Raphson procedure.

Newton-Raphson Iterative method

Newton-Raphson is an iterative method used to obtain parameter estimation for maximum likelihood. This method requires the second derivatives of the log likelihood equations with respect to α and β . The second derivative with respect to α is

$$\frac{\delta^2 L(\alpha, \beta)}{\delta \alpha^2} = -\sum \pi(x_i)[1 - \pi(x_i)] \quad (\text{B7})$$

The derivative with respect to α and β is

$$\frac{\delta^2 L(\alpha, \beta)}{\delta \alpha \delta \beta} = -\sum x_i \pi(x_i)[1 - \pi(x_i)] \quad (\text{B8})$$

The second derivative with respect to β is

$$\frac{\delta^2 L(\alpha, \beta)}{\delta \beta^2} = -\sum x_i^2 \pi(x_i)[1 - \pi(x_i)] \quad (\text{B9})$$

Theoretically, this procedure chooses initial estimates of the α and β , such as $\alpha = 0$ and $\beta = 0$. It calculates the log likelihood and evaluates the likelihood equations and the second derivatives. It uses the results of the product of the inverse of the second derivative matrix and the likelihood functions to calculate adjustments for α and β .

Using matrix notation, the adjustment is

$$\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix}^{new} = \begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix}^{old} + \begin{bmatrix} -\frac{\delta^2 L(\alpha, \beta)}{\delta \alpha^2} & -\frac{\delta^2 L(\alpha, \beta)}{\delta \alpha \delta \beta} \\ -\frac{\delta^2 L(\alpha, \beta)}{\delta \alpha \delta \beta} & -\frac{\delta^2 L(\alpha, \beta)}{\delta \beta^2} \end{bmatrix}^{-1} \begin{bmatrix} \sum [y_i - \pi(x_i)] = 0 \\ \sum x_i [y_i - \pi(x_i)] = 0 \end{bmatrix} \quad (\text{B10})$$

where at each iteration t , it will update the coefficient.

This iteration will stop when the percentages of error is decrease to the smallest value which approximately becomes zero.

To simply Eq. B10, it can be written as follows:

$$b_t = b_{t-1} + (X'V_{t-1}X)^{-1}X'(y - p_{t-1}) \quad (\text{B11})$$

where X is the model matrix, with $x'i$ as its i th row;

y is the response vector (containing 0's and 1's);

p_{t-1} is the vector of fitted response variable probabilities from the previous iteration, the i th entry of which is

$$p_{t-1} = \frac{1}{1 + \exp(-x'i b_{t-1})} \quad (\text{B12})$$

V_{t-1} is a diagonal matrix with diagonal entries $p_{i,t-1}(1 - p_{i,t-1})$

MATLAB Functions

The logistic model is developed in MATLAB R2009a using two main functions; *glmfit* and *glmval* functions. In MATLAB, these functions are used in such a way:

$$b = \text{glmfit}(x, y, \text{distr}) \quad (\text{B13})$$

The coding in Eq. B13 returns as output of a p -by-1 vector β of coefficient estimates for a generalized linear regression of the responses in y on the predictors in \mathbf{x} , using the distribution *distr*. The *distr* can be any of the following strings, ‘binomial’, ‘inverse Gaussian’, ‘normal’ and ‘Poisson’. For CUI case, *distr* refer to binomial distribution.

In most cases, y is an n -by-1 vector of observed responses whereby for this case, the response is either CUI is observed (1) or CUI is not observed (0). For the binomial distribution, y can be a binary vector (0 or 1) indicating success or failure at each observation, or a two column matrix with the first column indicating the number of successes for each observation and the second column indicating the number of trials for each observation.

In this project, the code in Eq. B13 is used as:

$$b = \text{glmfit}(x, [y \text{ ones}(339,1)], 'binomial', 'link', 'logit')$$

where

‘ones’ creates an array of all ones

‘logit’, a default for the distribution ‘binomial’ and represents $\log(\mu/(1-\mu)) = \beta x$

‘link’ is to link the anchor or pointer to the report

Next function used is ‘glmval’

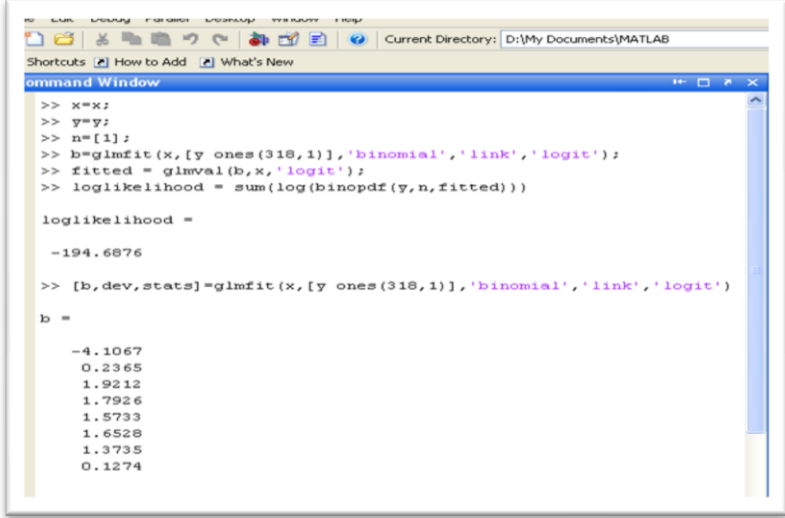
$$\text{fitted} = \text{glmval}(b, x, 'logit')$$

where computes the predicted distribution parameters for observations with predictor values \mathbf{x} using the coefficient vector β and link function ‘link’. Typically, β is a vector of coefficient estimates computed by the *glmfit* function. Here the code ‘link’ represents the logit function used in previous *glmfit* function whereby the value must be the same. Instead of producing coefficient estimates for the beta, the powerful software also provide several output using coding below:

`[b,dev,stats] = glmfit(...)`

It returns deviance (*dev*) which is also known as the generalization of the residual sum of squares. Apart from that, stats returns those several output which are listed below:

- Beta: Coefficient estimates *b*
- dfe: Degrees of freedom for error
- se: Vector of standard errors of the coefficient estimates *b*
- *t*: *t* statistics for *b*
- *p*: *p*-values for *b*



```
>> x=x;
>> y=y;
>> n=[1];
>> b=glmfit(x,[y ones(318,1)],'binomial','link','logit');
>> fitted = glmval(b,x,'logit');
>> loglikelihood = sum(log(binopdf(y,n,fitted)))

loglikelihood =

-194.6876

>> [b,dev,stats]=glmfit(x,[y ones(318,1)],'binomial','link','logit')

b =

-4.1067
 0.2365
 1.9212
 1.7926
 1.5733
 1.6528
 1.3735
 0.1274
```

Figure B1: Example of command window for MATLAB coding