Estimation in Generalized linear models

Thomas Achia

4/7/2022

$$\text{Poisson:}\frac{e^{-\lambda}\lambda^{y}}{y!} = \exp[y\ln\lambda - \lambda - \ln y!] = \exp[\frac{y\underbrace{\ln\lambda}_{\theta} - \underbrace{\lambda}_{b(\theta)}}{1} - \ln y!]$$

$$\boxed{\theta = \ln\lambda \Rightarrow \lambda = e^{\theta} \Rightarrow \frac{\partial\lambda}{\partial\theta} = e^{\theta}}; b(\theta) = \lambda \Rightarrow b'(\theta) = \frac{\partial}{\partial\theta}b(\theta) = \frac{\partial\lambda}{\partial\theta} = e^{\theta} = \lambda = E(Y)$$

$$b''(\theta) = \frac{\partial^2}{\partial\theta^2}b(\theta) = \frac{\partial}{\partial\theta}e^{\theta} = e^{\theta} = \lambda \Rightarrow \underbrace{V(\mu) = \phi b''(\theta) = 1\times\lambda = \lambda}_{Var(Y)}$$

$$f(y;\theta) = \exp\left[\frac{y\theta - b(\theta)}{\phi}\right] + c(y;\phi) \Rightarrow E(Y) = b'(\theta); Var(Y) = \phi b''(\theta)$$

For Generalized Linear Models 1. The pdf of Y needs to belong to the Exponential dispersion family: The stochastic part of the model 2. Given

$$\text{Explanatory variables: } X_1, \ldots, X_k$$

the relationship between predictor and response variables is

$$\eta = g(\mu_i) = \underbrace{\beta_0 + \beta_1 X_{i1} + \ldots + \beta_k X_{ik}}_{\text{The deterministic component}} = \boldsymbol{X}_i'\boldsymbol{\beta};\ \text{where } \mu_i = E(Y_i) \text{ and } g \text{ is called a link function}$$

$$E(Y_i) = \mu_i = g^{-1}(\boldsymbol{X}_i'\boldsymbol{\beta})$$

Estimation

$$\underbrace{Y_1, Y_2, \ldots, Y_n}_{\text{A random sample } f(y;\theta)}$$

$$\text{Likelihood: } L(\theta;\boldsymbol{y}) = \prod_{i=1}^{n} f(y_i;\theta) = \prod_{i=1}^{n}\exp\left[\frac{y_i\theta - b(\theta)}{\phi} + c(y_i;\phi)\right] = \exp\sum_{i=1}^{n}\left[\frac{y_i\theta - b(\theta)}{\phi} + c(y_i;\phi)\right]$$

$$\text{log-Likelihood: } \ell(\theta;\boldsymbol{y}) = \ln L(\theta;\boldsymbol{y}) = \ln\left[\exp\sum_{i=1}^{n}\left[\frac{y_i\theta - b(\theta)}{\phi} + c(y_i;\phi)\right]\right] = \sum_{i=1}^{n}\left[\frac{y_i\theta - b(\theta)}{\phi} + c(y_i;\phi)\right]$$

$$\text{log-Likelihood: } \ell(\theta;\boldsymbol{y}) = \sum_{i=1}^{n}\left[\frac{y_i\theta - b(\theta)}{\phi} + c(y_i;\phi)\right] = \sum_{i=1}^{n}\ell_i$$

$$\ell_i = \left[\frac{y_i\theta - b(\theta)}{\phi} + c(y_i;\phi)\right]$$

We want $\beta_0, \beta_1, \ldots, \beta_k,$ that will maximize $\ell(\theta;\boldsymbol{y})$ and ℓ_i

$$S_{ij}{=}\boxed{\frac{\partial \ell_i}{\partial \beta_j} = \frac{\partial}{\partial \beta_j}\left[\frac{y_i\theta - b(\theta)}{\phi} + c(y_i;\phi)\right]} = 0$$

Recall

$$\mu = E(Y) = b'(\theta);\ \boxed{\eta = g(\mu)} = \beta_0 + \beta_1 X_1 + \ldots + \beta_j X_j + \ldots + \beta_k X_k$$

$$\boxed{\frac{\partial \eta}{\partial \beta_j} = X_j};\ \boxed{\frac{\partial \mu}{\partial \theta} = b''(\theta) = \frac{V()}{\phi}};\ \boxed{\frac{\partial \eta}{\partial \mu} = \frac{\partial}{\partial \mu} g(\mu) = g'(\mu)}$$

$$S_{ij} = \frac{\partial \ell_i}{\partial \beta_j} = \frac{\partial \ell_i}{\partial \theta}\frac{\partial \theta}{\partial \beta_j} = \left[\frac{y_i - b'(\theta)}{\phi}\right]\frac{\partial \theta}{\partial \beta_j} = \left[\frac{y_i - b'(\theta)}{\phi}\right]\frac{\partial \eta}{\partial \beta_j}\frac{\partial \theta}{\partial \eta} = \left[\frac{y_i - b'(\theta)}{\phi}\right] X_j \boxed{\frac{\partial \theta}{\partial \eta}}$$

$$S_{ij} = \frac{\partial \ell_i}{\partial \beta_j} == \left[\frac{y_i - b'(\theta)}{\phi}\right] X_j \boxed{\frac{\partial \theta}{\partial \mu}\frac{\partial \mu}{\partial \eta}} = \left[\frac{y_i - b'(\theta)}{\phi}\right]\frac{X_j}{b''(\theta)g'(\mu)} = \left[\frac{y_i - b'(\theta)}{\phi b''(\theta)g'(\mu)}\right] X_j = \underbrace{\boxed{\frac{\partial \ell_i}{\partial \theta}\frac{\partial \theta}{\partial \mu}\frac{\partial \mu}{\partial \eta}\frac{\partial \eta}{\partial \beta_j}}}_{\text{Chain rule}}$$

$$S_{ij} = \frac{\partial \ell_i}{\partial \beta_j} = \left[\frac{y_i - b'(\theta)}{\phi b''(\theta)g'(\mu)}\right] X_j$$

$$\text{log-Likelihood: } \ell(\theta;\boldsymbol{y}) = \sum_{i=1}^{n}\left[\frac{y_i\theta - b(\theta)}{\phi} + c(y_i;\phi)\right] = \sum_{i=1}^{n} \ell_i$$

$$S_j = \frac{\partial}{\partial \beta_j}\ell(\theta;\boldsymbol{y}) = \sum_{i=1}^{n} S_{ij} = \sum_{i=1}^{n}\left[\frac{y_i - b'(\theta)}{V()g'(\mu)}\right] X_j \equiv \text{Estimating equation}, j = 0, 1, \ldots, k$$

$$\boldsymbol{S}(\theta;\boldsymbol{y}) = \begin{pmatrix} S_0 \\ S_1 \\ \vdots \\ S_k \end{pmatrix} = \text{The score vector}$$

To the expected Fisher's Information matrix, we start with

$$S_j = \frac{\partial}{\partial \beta_j}\ell(\theta;\boldsymbol{y}) = \sum_{i=1}^{n} S_{ij} = \sum_{i=1}^{n}\left[\frac{y_i - b'(\theta)}{V()g'(\mu)}\right] X_j$$

and build the expected Fisher's Information matrix using a matrix of the following form

$$\mathcal{I} = ((\mathcal{I}_{ij})), \text{where } \mathcal{I}_{jk} = E(S_j S_k)$$

$$\mathcal{I}_{jk} = E(S_j S_k) = E\left[\sum_{i=1}^{n}\left[\frac{y_i - b'(\theta)}{V()g'(\mu)}\right] X_j \times \sum_{i=1}^{n}\left[\frac{y_i - b'(\theta)}{V()g'(\mu)}\right] X_k\right] = \left[\sum_{i=1}^{n}\left[\frac{E(y_i-\mu)^2}{[V()g'(\mu)]^2}\right] X_j X_k\right] = \sum_{i=1}^{n}\left[\frac{X_j X_k}{V()[g'(\mu)]^2}\right]$$

Note

1. The Score vector

$$\boldsymbol{S}(\boldsymbol{\beta};\boldsymbol{y})$$

   with elements

$$S_j = \sum_{i=1}^{n}\left[\frac{y_i - b'(\theta)}{V()g'(\mu)}\right] X_j,\ \ j = 0, 1, \ldots, k$$

2. The Information matrix

$$\mathcal{I} = ((\mathcal{I}_{ij})), \text{where } \mathcal{I}_{jk} = E(S_j S_k)$$

$$\mathcal{I}_{jk} = \sum_{i=1}^{n}\left[\frac{X_j X_k}{V()[g'(\mu)]^2}\right]$$

Estimation follows the tailor-series and Newton-Raphson procedure

$$f(x) = f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2!}f''(a) + \ldots \approx f(a) + (x-a)f'(a)$$

$$\boldsymbol{S}(\boldsymbol{\beta};\boldsymbol{y}) = \boldsymbol{S}(\boldsymbol{\beta}^*;\boldsymbol{y}) + \left[\frac{\partial}{\partial\boldsymbol{\beta}}\boldsymbol{S}(\boldsymbol{\beta};\boldsymbol{y})\right]_{\boldsymbol{\beta}=\boldsymbol{\beta}^*} \times (\boldsymbol{\beta}-\boldsymbol{\beta}^*) = \boldsymbol{S}(\boldsymbol{\beta}^*;\boldsymbol{y}) + [\mathcal{H}(\boldsymbol{\beta};\boldsymbol{y})]_{\boldsymbol{\beta}=\boldsymbol{\beta}^*}(\boldsymbol{\beta}-\boldsymbol{\beta}^*) = \boldsymbol{0},$$

where $\mathcal{H}(\boldsymbol{\beta};\boldsymbol{y})$ is the Hessian.

$$\boldsymbol{S}(\boldsymbol{\beta}^*;\boldsymbol{y}) + [\mathcal{H}(\boldsymbol{\beta}^*;\boldsymbol{y})](\boldsymbol{\beta}-\boldsymbol{\beta}^*) = \boldsymbol{0} \Rightarrow \boldsymbol{S}(\boldsymbol{\beta}^*;\boldsymbol{y}) = -[\mathcal{H}(\boldsymbol{\beta}^*;\boldsymbol{y})](\boldsymbol{\beta}-\boldsymbol{\beta}^*)$$

This implies that

$$\mathcal{H}^{-1}(\boldsymbol{\beta}^*;\boldsymbol{y})S(\boldsymbol{\beta}^*;\boldsymbol{y}) = -(\boldsymbol{\beta}-\boldsymbol{\beta}^*) \Rightarrow \boxed{\boldsymbol{\beta} = \boldsymbol{\beta}^* - \mathcal{H}^{-1}(\boldsymbol{\beta}^*;\boldsymbol{y})S(\boldsymbol{\beta}^*;\boldsymbol{y})}$$

$$\boxed{\boldsymbol{\beta}^{(m)} = \boldsymbol{\beta}^{(m-1)} - H^{-1}(\boldsymbol{\beta}^{(m-1)};\boldsymbol{y})S(\boldsymbol{\beta}^{(m-1)};\boldsymbol{y})} \equiv \text{Newton-Raphson method}$$

We replace the Hessian with the expected Fishers information as follows:

$$\boxed{\boldsymbol{\beta}^{(m)} = \boldsymbol{\beta}^{(m-1)} + \mathcal{I}^{-1}(\boldsymbol{\beta}^{(m-1)};\boldsymbol{y})S(\boldsymbol{\beta}^{(m-1)};\boldsymbol{y})} \equiv \text{Fisher's scoring method/Iteratively weighted Least squaress}$$

Special cases

Multiple linear regression analysis

Let us consider the Normal distribution:

$$f(y_i) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{1}{2}\left(\frac{y_i-\mu_i}{\sigma^2}\right)^2$$

Here

$$f(y_i) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{1}{2}\left(\frac{y_i-\mu_i}{\sigma^2}\right)^2 \Rightarrow \mu_i = E(Y_i) = \mu_i = b'(\theta_i);\ \sigma_i^2 = \text{Var}(Y_i) = \sigma^2 = V().$$

The link function and its derivative are:

$$g(\mu_i) = \mu_i \Rightarrow g'(\mu_i) = \frac{\partial}{\partial\mu_i}[\mu_i] = 1.$$

$$S_j = \frac{\partial}{\partial \beta_j}\ell(\theta;\boldsymbol{y}) = \sum_{i=1}^{n}\left[\frac{y_i - b'(\theta)}{V()g'(\mu)}\right]X_j = \sum_{i=1}^{n}\left[\frac{y_i-\mu_i}{\sigma^2\times 1}\right]X_j = \frac{1}{\sigma^2}\sum_{i=1}^{n}\left(y_i-\mu_i\right)X_j, j=0,1,\ldots,k$$

$$S_j = \frac{1}{\sigma^2}\sum_{i=1}^{n}X_j\left(y_i-\mu_i\right), j=0,1,\ldots,k$$

$$S_j = \frac{1}{\sigma^2}\sum_{i=1}^{n}X_j\left[y_i-\mu_i\right] = 0 \Rightarrow \boxed{\sum_{i=1}^{n}X_j y_i = \sum_{i=1}^{n}X_j\mu_i}$$

$$g(\mu_i) = \mu_i = \boldsymbol{X}_i'\boldsymbol{\beta}$$

$$\boxed{\sum_{i=1}^{n}X_j y_i = \sum_{i=1}^{n}X_j\mu_i} \Rightarrow \boxed{\sum_{i=1}^{n}X_j y_i = \sum_{i=1}^{n}X_j\boldsymbol{X}_i'\boldsymbol{\beta}}, j=0,1,\ldots,k$$

$$\boldsymbol{X}'\boldsymbol{y} = \boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} \Rightarrow \boxed{\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}} \equiv \text{Analytical solution}$$

The Information matrix

$$\mathcal{I}_{jk} = \sum_{i=1}^{n}\left[\frac{X_jX_k}{V()[g'(\mu)]^2}\right] = \sum_{i=1}^{n}\left[\frac{X_jX_k}{\sigma^2\times 1}\right] = \frac{1}{\sigma^2}\sum_{i=1}^{n}X_jX_k$$

$$\mathcal{I} = \frac{1}{\sigma^2}\boldsymbol{X}'\boldsymbol{X} = \boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}$$

$$\boldsymbol{W} = \begin{pmatrix} \frac{1}{\sigma^2} & 0 & \ldots & 0 \\ 0 & \frac{1}{\sigma^2} & \ldots & 0 \\ \vdots & & & \\ 0 & 0 & \ldots & \frac{1}{\sigma^2} \end{pmatrix}$$

Logistic regression analysis (Binomial/Bernoulli data)

$$f(y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i} = \exp\left[\frac{y_i\boxed{\ln\left(\frac{\pi_i}{1-\pi_i}\right)} - \ln(1-\pi_i)}{1}\right]$$

this implies that

$$\mu_i = E(Y_i) = \pi_i = b'(\theta_i);\ \sigma_i^2 = \text{Var}(Y_i) = \pi_i(1-\pi_i) = V().$$

The link functin and it's derivative are:

$$\boxed{g(\mu_i) = \text{logit}(\pi_i) = \ln\left(\frac{\pi_i}{1-\pi_i}\right)} \Rightarrow g'(\mu_i) = \frac{\partial}{\partial\mu_i}\left[g(\mu_i)\right] = \frac{\partial}{\partial\mu_i}\left[\ln\left(\frac{\pi_i}{1-\pi_i}\right)\right]$$

And so

$$g'(\mu_i) = \frac{\partial}{\partial\pi_i}\left[\ln\left(\frac{\pi_i}{1-\pi_i}\right)\right]\frac{\partial\pi_i}{\partial\mu_i} = \left(\frac{\pi_i}{1-\pi_i}\right)^{-1}\times\left[\frac{[(1-\pi_i)\times 1]-[\pi_i(-1)]}{(1-\pi_i)^2}\right]\times 1$$

Thus

$$g'(\mu_i) = \left[\frac{1-\pi_i}{\pi_i}\right]\times\frac{1}{(1-\pi_i)^2} = \frac{1}{\pi_i(1-\pi_i)} = \frac{1}{V()}$$

$$S_j = \frac{\partial}{\partial \beta_j}\ell(\theta;\boldsymbol{y}) = \sum_{i=1}^{n}\left[\frac{y_i - b'(\theta)}{V()g'(\mu)}\right]X_j = \sum_{i=1}^{n}\left[\frac{y_i-\mu_i}{V()\frac{1}{V()}}\right]X_j = \sum_{i=1}^{n}\left(y_i-\mu_i\right)X_j, j=0,1,\ldots,k$$

$$S_j = \sum_{i=1}^{n} X_j\left(y_i-\mu_i\right), j = 0,1,\ldots,k$$

$$g(\mu_i) = \text{logit}(\pi_i) = \ln\left(\frac{\pi_i}{1-\pi_i}\right) = \boldsymbol{X}'_i\boldsymbol{\beta} \Rightarrow \pi_i = \frac{e^{\boldsymbol{X}'_i\boldsymbol{\beta}}}{1+e^{\boldsymbol{X}'_i\boldsymbol{\beta}}} = \mu_i$$

$$S_j = \sum_{i=1}^{n} X_j\left[y_i - \left(\frac{e^{\boldsymbol{X}'_i\boldsymbol{\beta}}}{1+e^{\boldsymbol{X}'_i\boldsymbol{\beta}}}\right)\right] = 0 \Rightarrow \boxed{\sum_{i=1}^{n} X_j y_i = \sum_{i=1}^{n} X_j\left(\frac{e^{\boldsymbol{X}'_i\boldsymbol{\beta}}}{1+e^{\boldsymbol{X}'_i\boldsymbol{\beta}}}\right)} \equiv \text{No analytical solution}$$

$$S_j = \sum_{i=1}^{n} X_j\left[y_i-\mu_i\right] = \sum_{i=1}^{n} X_j\left[y_i - \left(\frac{e^{\boldsymbol{X}'_i\boldsymbol{\beta}}}{1+e^{\boldsymbol{X}'_i\boldsymbol{\beta}}}\right)\right], j=0,1,\ldots,k \Rightarrow \boxed{\boldsymbol{S}(\boldsymbol{\beta};\boldsymbol{y}) = \frac{\partial}{\partial\boldsymbol{\beta}}\ell(\boldsymbol{\beta};\boldsymbol{y}) = \boldsymbol{X}'(\boldsymbol{y}-\boldsymbol{\mu})} = \text{Score vector}$$

$$\boldsymbol{\mu} = \begin{pmatrix}\mu_1\\ \mu_2\\ \vdots\\ \mu_p\end{pmatrix} = \begin{pmatrix}\pi_1\\ \pi_2\\ \vdots\\ \pi_p\end{pmatrix} = \begin{pmatrix}\frac{e^{\boldsymbol{X}'_1\boldsymbol{\beta}}}{1+e^{\boldsymbol{X}'_1\boldsymbol{\beta}}}\\ \frac{e^{\boldsymbol{X}'_2\boldsymbol{\beta}}}{1+e^{\boldsymbol{X}'_2\boldsymbol{\beta}}}\\ \vdots\\ \frac{e^{\boldsymbol{X}'_p\boldsymbol{\beta}}}{1+e^{\boldsymbol{X}'_p\boldsymbol{\beta}}}\end{pmatrix}$$

The Information matrix

$$\mathcal{I}_{jk} = \sum_{i=1}^{n}\left[\frac{X_jX_k}{V()[g'(\mu)]^2}\right] = \sum_{i=1}^{n}\left[\frac{X_jX_k}{V()\left[\frac{1}{V()}\right]^2}\right] = \sum_{i=1}^{n}\left[\frac{X_jX_k}{\left[\frac{1}{V()}\right]}\right]$$

Then

$$\mathcal{I}_{jk} = \sum_{i=1}^{n} V()X_jX_k = \sum_{i=1}^{n}\pi_i(1-\pi_i)X_jX_k = \sum_{i=1}^{n} X_j\left[\pi_i(1-\pi_i)\right]X_k$$

Then the Information matrix:

$$\mathcal{I} = \boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}$$

$$\boldsymbol{W} = \begin{pmatrix}\pi_1(1-\pi_1) & 0 & \ldots & 0\\ 0 & \pi_2(1-\pi_2) & \ldots & 0\\ \vdots & & & \\ 0 & 0 & \ldots & \pi_p(1-\pi_p)\end{pmatrix}$$

Also the Hessian:

$$H = \frac{\partial^2}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}'}\ell(\boldsymbol{\beta};\boldsymbol{y}) = -\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}$$

Fisher's scoring

$$\boxed{\boldsymbol{\beta}^{(m)} = \boldsymbol{\beta}^{(m-1)} + \mathcal{I}^{-1}(\boldsymbol{\beta}^{(m-1)};\boldsymbol{y})S(\boldsymbol{\beta}^{(m-1)};\boldsymbol{y})} = \boldsymbol{\beta}^{(m-1)} + [\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}]^{-1}\boldsymbol{X}'(\boldsymbol{y}-\boldsymbol{\mu})$$

Example

This was obtained from http://www.jtrive.com/estimating-logistic-regression-coefficents-from-scratch-r-version.html

This data represents O-Ring failures in the 23 pre-Challenger space shuttle missions. In this dataset, "TEMPERATURE" will serve as the single explnatory variable which will be used to predict "O_RING_FAILURE," which is "1" if a failure occurred, "0" otherwise.

The data and some preliminary analysis follow:

```
df <- read.table(
  file="C:/Users/yrb2/OneDrive - CDC/+My_Documents/Strathmore MSC/DSA 8302 Computational techniques in I
  header=TRUE,
  sep=",",
  stringsAsFactors=FALSE
)

fit<-glm(O_RING_FAILURE~TEMPERATURE,data = df,family = binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = O_RING_FAILURE ~ TEMPERATURE, family = binomial,
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0611  -0.7613  -0.3783   0.4524   2.2175
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  15.0429     7.3786   2.039   0.0415 *
## TEMPERATURE  -0.2322     0.1082  -2.145   0.0320 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 28.267  on 22  degrees of freedom
## Residual deviance: 20.315  on 21  degrees of freedom
## AIC: 24.315
##
## Number of Fisher Scoring iterations: 5
```

Once the parameters have been determined, the model estimate of the probability of success for a given observation can be calculated with:

$$\hat{\pi}_i = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x_{i1}+\cdots+\hat{\beta}_k x_{ik}}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x_{i1}+\cdots+\hat{\beta}_k x_{ik}}}$$

In the following code segment, we define a single function, getCoefficients, which returns the estimated model coefficents as a $(k+1)\times 1$ matrix. In addition, the function returns the number of scoring iterations, fitted values and the variance-covariance matrix for the estimated coefficients:

```
getCoefficients <- function(design_matrix, response_vector, epsilon=.0001) {
    # =======================================================================
    # design_matrix      `X`      => n-by-(p+1)                              |
```

```r
    # response_vector    `y`     => n-by-1                                 |
    # probability_vector `p`     => n-by-1                                 |
    # weights_matrix     `W`     => n-by-n                                 |
    # epsilon                    => threshold above which iteration continues |
    # =====================================================================
    # n                          => # of observations                      |
    # (p + 1)                    => # of parameterss, +1 for intercept term   |
    # =====================================================================
    # U => First derivative of Log-Likelihood with respect to              |
    #      each beta_i, i.e. `Score Function`: X_transpose * (y - p)       |
    #                                                                      |
    # I => Second derivative of Log-Likelihood with respect to             |
    #      each beta_i. The `Information Matrix`: (X_transpose * W * X)    |
    #                                                                      |
    # X^T*W*X results in a (p+1)-by-(p+1) matrix                           |
    # X^T(y - p) results in a (p+1)-by-1 matrix                            |
    # (X^T*W*X)^-1 * X^T(y - p) results in a (p+1)-by-1 matrix             |
    # ====================================================================|
    X <- as.matrix(design_matrix)
    y <- as.matrix(response_vector)

    # initialize logistic function used for Scoring calculations =>
    pi_i <- function(v) return(exp(v)/(1 + exp(v)))

    # initialize beta_0, p_0, W_0, I_0 & U_0 =>
    beta_0 <- matrix(rep(0, ncol(X)), nrow=ncol(X), ncol=1, byrow=FALSE, dimnames=NULL)
    p_0    <- pi_i(X %*% beta_0)
    W_0    <- diag(as.vector(p_0*(1-p_0)))
    I_0    <- t(X) %*% W_0 %*% X
    U_0    <- t(X) %*% (y - p_0)

    # initialize variables for iteration =>
    beta_old                   <- beta_0
    iter_I                     <- I_0
    iter_U                     <- U_0
    iter_p                     <- p_0
    iter_W                     <- W_0
    fisher_scoring_iterations  <- 0

    # iterate until difference between abs(beta_new - beta_old) < epsilon =>
    while(TRUE) {

        # Fisher Scoring Update Step =>
        fisher_scoring_iterations <- fisher_scoring_iterations + 1
        beta_new <- beta_old + solve(iter_I) %*% iter_U

        if (all(abs(beta_new - beta_old) < epsilon)) {
            model_parameters  <- beta_new
            fitted_values     <- pi_i(X %*% model_parameters)
            covariance_matrix <- solve(iter_I)
            break

        } else {
```

```r
            iter_p   <- pi_i(X %*% beta_new)
            iter_W   <- diag(as.vector(iter_p*(1-iter_p)))
            iter_I   <- t(X) %*% iter_W %*% X
            iter_U   <- t(X) %*% (y - iter_p)
            beta_old <- beta_new
        }
    }

    summaryList <- list(
        'model_parameters'=model_parameters,
        'covariance_matrix'=covariance_matrix,
        'fitted_values'=fitted_values,
        'number_iterations'=fisher_scoring_iterations
    )
    return(summaryList)
}
```

A quick summary of R's matrix symbols and operators:

- %*% is a stand-in for matrix multiplication
- diag returns a matrix with the provided vector as the diagonal and zero off-diagonal entries
- t returns the transpose of the provided matrix
- solve returns the inverse of the provided matrix, if applicable

We read the Challenger dataset into R and partition it into the design matrix and the response, which will then be passed to getCoefficients:

```r
df <- read.table(
            file="Challenger.csv",
            header=TRUE,
            sep=",",
            stringsAsFactors=FALSE
            )

X  <- as.matrix(cbind(1, df['TEMPERATURE']))     # design matrix
y  <- as.matrix(df['O_RING_FAILURE'])            # response vector

colnames(X) <- NULL
colnames(y) <- NULL

# call `getCoefficients`, keeping epsilon at .0001 =>
mySummary <- getCoefficients(design_matrix=X, response_vector=y, epsilon=.0001)
```

Printing mySummary displays the model's estimated coefficents (model_parameters), the variance-covariance matrix of the coefficent estimates (covariance_matrix), the fitted values (fitted_values) and the number of Fisher Scoring iterations (number_iterations):

```r
print(mySummary)
```

```
## $model_parameters
##             [,1]
## [1,] 15.0429016
## [2,] -0.2321627
##
## $covariance_matrix
```

```
##              [,1]        [,2]
## [1,] 54.4442748 -0.79638682
## [2,] -0.7963868  0.01171514
##
## $fitted_values
##               [,1]
##  [1,] 0.43049313
##  [2,] 0.22996826
##  [3,] 0.27362105
##  [4,] 0.32209405
##  [5,] 0.37472428
##  [6,] 0.15804910
##  [7,] 0.12954602
##  [8,] 0.22996826
##  [9,] 0.85931657
## [10,] 0.60268105
## [11,] 0.22996826
## [12,] 0.04454055
## [13,] 0.37472428
## [14,] 0.93924781
## [15,] 0.37472428
## [16,] 0.08554356
## [17,] 0.22996826
## [18,] 0.02270329
## [19,] 0.06904407
## [20,] 0.03564141
## [21,] 0.08554356
## [22,] 0.06904407
## [23,] 0.82884484
##
## $number_iterations
## [1] 6
```

Poisson regression analysis (Count data)

$$f(y_i) = \frac{e^{\lambda_i}\lambda_i^{y_i}}{y_i!} = \exp\left[\frac{y\boxed{\ln\lambda_i} - \lambda_i}{1} - \ln y_i!\right] \Rightarrow \boxed{\mu_i = E(Y_i) = \lambda_i = b'(\theta_i)};\ \boxed{\sigma_i^2 = \mathrm{Var}(Y_i) = \lambda_i = V()};$$

$$\boxed{g(\mu_i) = \ln(\lambda_i)} \Rightarrow g'(\mu_i) = \frac{\partial}{\partial\mu_i}\left[g(\mu_i)\right] = \frac{\partial}{\partial\mu_i}\left[\ln(\lambda_i)\right] = \frac{1}{\lambda_i} = \frac{1}{V()}.$$

$$S_j = \frac{\partial}{\partial\beta_j}\ell(\theta;\boldsymbol{y}) = \sum_{i=1}^{n}\left[\frac{y_i - b'(\theta)}{V()g'(\mu)}\right]X_j = \sum_{i=1}^{n}\left[\frac{y_i - \mu_i}{V()\frac{1}{V()}}\right]X_j = \sum_{i=1}^{n}\left(y_i - \mu_i\right)X_j, j = 0, 1, \ldots, k$$

$$S_j = \sum_{i=1}^{n} X_j\left(y_i - \mu_i\right), j = 0, 1, \ldots, k$$

$$S_j = \sum_{i=1}^{n} X_j\left[y_i - \mu_i\right] = \sum_{i=1}^{n} X_j\left[y_i - e^{\boldsymbol{X}'_i\boldsymbol{\beta}}\right], j = 0, 1, \ldots, k \Rightarrow \boxed{\boldsymbol{S}(\boldsymbol{\beta};\boldsymbol{y}) = \frac{\partial}{\partial\boldsymbol{\beta}}\ell(\boldsymbol{\beta};\boldsymbol{y}) = \boldsymbol{X}'(\boldsymbol{y} - \boldsymbol{\mu})} = \text{Score vector}$$

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{pmatrix} = \begin{pmatrix} e^{\boldsymbol{X}'_1\boldsymbol{\beta}} \\ e^{\boldsymbol{X}'_2\boldsymbol{\beta}} \\ \vdots \\ e^{\boldsymbol{X}'_p\boldsymbol{\beta}} \end{pmatrix}$$

$$g(\mu_i) = \ln(\mu_i) = \boldsymbol{X}'_i\boldsymbol{\beta} \Rightarrow \boxed{\lambda_i = e^{\boldsymbol{X}'_i\boldsymbol{\beta}} = \mu_i}$$

$$S_j = \sum_{i=1}^{n} X_j \left[y_i - e^{\boldsymbol{X}'_i\boldsymbol{\beta}}\right] = 0 \Rightarrow \boxed{\sum_{i=1}^{n} X_j y_i = \sum_{i=1}^{n} X_j e^{\boldsymbol{X}'_i\boldsymbol{\beta}}} \equiv \text{Estimating with No analytical solution}$$

The Information matrix

$$\mathcal{I}_{jk} = \sum_{i=1}^{n} \left[\frac{X_j X_k}{V()[g'(\mu)]^2}\right] = \sum_{i=1}^{n} \left[\frac{X_j X_k}{V()\left[\frac{1}{V()}\right]^2}\right] = \sum_{i=1}^{n} \left[\frac{X_j X_k}{\left[\frac{1}{V()}\right]}\right] = \sum_{i=1}^{n} V() X_j X_k = \sum_{i=1}^{n} \lambda_i X_j X_k = \sum_{i=1}^{n} X_j \lambda_i X_k$$

$$\mathcal{I} = \boldsymbol{X}'\boldsymbol{W}\boldsymbol{X}$$

$$\boldsymbol{W} = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & \lambda_p \end{pmatrix}$$

Task

1. Given the exponential distribution

$$f(y) = \lambda e^{-\lambda y}$$

   Find the Score function, the estimating equation and the information matrix.

2. Write an R-code that will carryout a Poisson regression analysis and use the data in Table 9.1 of (Dobson and Barnett 2018).

3. The data in Table 4.5 of (Dobson and Barnett 2018) show the numbers of cases of AIDS in Australia by date of diagnosis for successive 3-month periods from 1984 to 1988. (Data from National Centre for HIV Epidemiology and Clinical Research, 1994.) In this early phase of the epidemic, the numbers of cases seemed to be increasing exponentially.

a. Plot the number of cases y_i against time period $i,\ i = 1, \dots, 20$).
b. A possible model is the Poisson distribution with parameter $\lambda_i = i^{\theta}$, or equivalently

$$\ln \lambda_i = \theta_i \ln i.$$

   Plot $\ln y_i$ against $\ln i$ to examine this model.
c. Fit a generalized linear model to these data using the Poisson distribution, the log-link function and the equation

$$g(\lambda_i) = \ln \lambda_i = \beta_1 + \beta_2 x_i,$$

   where $x_i = \ln i$.

Firstly, do this from first principles, using **Fisher's scoring** and using software which can perform matrix operations to carry out the calculations.

Find the Score function, the estimating equation and the information matrix.

d. Fit the model described in (c) using statistical software which can perform Poisson regression. Compare the results with those obtained in (c).

4. The data in Table 4.6 of (Dobson and Barnett 2018) are times to death, y_i, in weeks from diagnosis and $\log_{10}$ (initial white blood cell count), x_i, for seventeen patients suffering from leukemia. (This is Example U from [@{cox2018applied])

a. Plot y_i against x_i. Do the data show any trend?
b. A possible specification for $E(Y)$ is

$$\mu_i = E(Y_i) = \exp(\beta_0 + \beta_1 x_i),$$

which will ensure that $E(Y)$ is non-negative for all values of the parameters and all values of x. Which link function is appropriate in this case?
c. The Exponential distribution is often used to describe survival times.

The probability distribution is $f(y; \theta) = \theta e^{-\theta y}$. This is a special case of the Gamma distribution with shape parameter $\alpha = 1$. Show that $E(Y) = \frac{1}{\theta}$ and $\text{var}(Y) = \frac{1}{\theta^2}$. d. Fit a model with the equation for $\mu_i = E(Y_i)$ given in (b) and the Exponential distribution using appropriate statistical software.

Find the Score function, the estimating equation and the information matrix.

5. Let $Y_1, \ldots, Y_N$ be a random sample from the Normal distribution $Y_i \sim N(\ln \beta, s^2)$ where s^2 is known. Find the maximum likelihood estimator of b from first principles.

Find the Score function, the estimating equation and the information matrix.

References

Dobson, Annette J, and Adrian G Barnett. 2018. *An Introduction to Generalized Linear Models*. Chapman; Hall/CRC.