MA50260 Statistical Modelling

Lecture 10: Estimation of GLMs

Ilaria Bussoli

March 8, 2024

Motivation – the **glm** function

To fit a GLM in R, you use

```
glm( y ~ X, family = ..., data = ... )
```

which requires specification of

- The **linear predictor** $\eta_i = \beta_1 x_{i,1} + \ldots + \beta_p x_{i,p} = \mathbf{x}_i^T \underline{\beta}$.
- The **link function** $g(\mu_i) = \eta_i$
- **Probability distribution** $Y_i \sim F(\mu_i)$ from **the exponential family**.

Motivation – the **glm** function

To fit a GLM in R, you use

```
glm( y ~ X, family = ..., data = ... )
```

which requires specification of

- The **linear predictor** $\eta_i = \beta_1 x_{i,1} + \ldots + \beta_p x_{i,p} = \mathbf{x}_i^T \underline{\beta}$.
- The **link function** $g(\mu_i) = \eta_i$
- **Probability distribution** $Y_i \sim F(\mu_i)$ from **the exponential family**.

How does the glm function estimate the regression coefficients?

The log-likelihood function

We have derived the log-likelihood as

$$\ell(\underline{\theta}, \phi \mid y_1, \ldots, y_n) = \frac{1}{\phi} \left\{ \sum_{i=1}^{n} w_i [y_i \theta_i - b(\theta_i)] \right\} + \sum_{i=1}^{n} c(y_i, \phi).$$

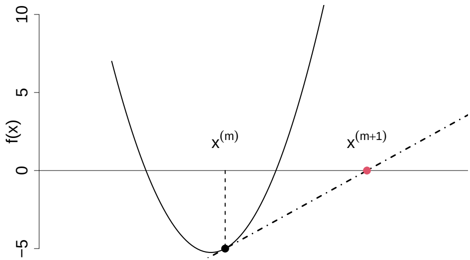
How do we find the maximum-likelihood estimate?

Newton-Raphson Method

- We want to solve $f(x) = 0$ numerically.
- Set initial guess $x^{(0)}$ and update the estimate iteratively as

$$x^{(m+1)} = x^{(m)} - \frac{f(x^{(m)})}{f'(x^{(m)})}$$

until convergence.



Fisher scoring

Consider the likelihood function $\ell(\theta)$.

To find the maximum-likelihood estimate, we need to solve

$$U(\theta) = \frac{\partial \ell(\theta \mid \mathbf{y})}{\partial \theta} = 0.$$

If we use the Newton-Raphson algorithm to solve this equation, the update is given by

$$\theta^{(m+1)} = \theta^{(m)} - \frac{U(\theta)}{U'(\theta)}.$$

Fisher scoring

Consider the likelihood function $\ell(\theta)$.

To find the maximum-likelihood estimate, we need to solve

$$U(\theta) = \frac{\partial \ell(\theta \mid \mathbf{y})}{\partial \theta} = 0.$$

If we use the Newton-Raphson algorithm to solve this equation, the update is given by

$$\theta^{(m+1)} = \theta^{(m)} - \frac{U(\theta)}{U'(\theta)}.$$

The Fisher scoring algorithm replaces $U'(\theta)$ by its expected value, the Fisher information,

$$\mathbb{E}\left[U'(\theta)\right] = -\mathcal{I}.$$

Therefore, the update step is

$$\theta^{(m+1)} = \theta^{(m)} + \frac{U(\theta)}{\mathcal{I}}.$$

Estimation of Regression Coefficients (I)

We calculate

$$\mathbf{U}\left(\underline{\beta}\right) = \frac{1}{\phi}\sum_{i=1}^{n}(y_i - \mu_i)W_{ii}\frac{\partial \eta_i}{\partial \mu_i}\mathbf{x}_i = 0,$$

with

$$W_{ii} = \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2 / V(\mu_i) = \frac{1}{V(\mu_i)g'(\mu_i)^2}.$$

The Fisher information is

$$\mathcal{I} = \left[\frac{1}{\phi}\sum_{i=1}^{n}\frac{x_{ij}x_{ik}}{V(\mu_i)}\left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2\right] = \frac{1}{\phi}\mathbf{X}^T\mathbf{W}\mathbf{X}.$$

Estimation of Regression Coefficients (II)

The update step in the Fisher scoring algorithm is then

$$\underline{\beta}^{(m+1)} = \underline{\beta}^{(m)} + \left[\mathcal{I}^{(m)}\right]^{-1} \mathbf{U}\left(\underline{\beta}^{(m)}\right).$$

We can write this as

$$\underline{\beta}^{(m+1)} = \underline{\beta}^{(m)} + (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z}^*,$$

with

$$z_i^* = (y_i - \mu_i)\left(\frac{\partial \eta_i}{\partial \mu_i}\right) = (y_i - \mu_i)g'(\mu_i).$$

Alternatively, we can write

$$\underline{\beta}^{(m+1)} = \left(\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{W}\mathbf{z},$$

$$z_i = (y_i - \mu_i)g'(\mu_i) + \mathbf{x}_i^{\mathrm{T}}\underline{\beta}^{(m)}.$$

Estimation of Regression Coefficients (III)

Iteratively Re-Weighted Least Squares (IRWLS)

1. Set $\underline{\hat{\eta}}^{(0)}$ and $\underline{\hat{\mu}}^{(0)}$.

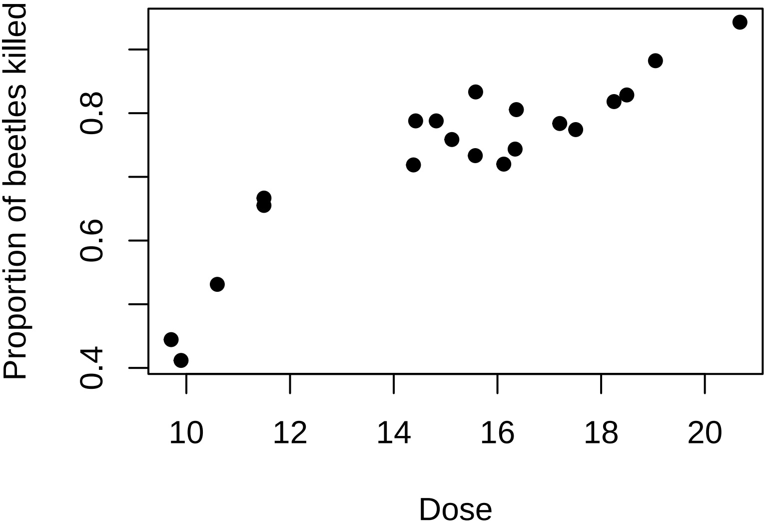
2. Compute the adjusted variable $\mathbf{z}^{(0)} = \underline{\hat{\eta}}^{(0)} + \left(\hat{\mathbf{y}} - \underline{\hat{\mu}}^{(0)}\right) \left.\frac{d\underline{\hat{\eta}}}{d\underline{\hat{\mu}}}\right|_{\underline{\hat{\eta}}^{(0)}}$.

3. Compute the weights $w_0^{-1} = \left.\left(\frac{d\underline{\hat{\eta}}}{d\underline{\hat{\mu}}}\right)^2\right|_{\underline{\hat{\eta}}^{(0)}} V\left(\underline{\hat{\mu}}^{(0)}\right)$.

4. Estimate $\underline{\hat{\beta}}^{(1)}$ using the weights to get $\underline{\hat{\eta}}^{(1)}$.

5. Iterate steps 2-4 until convergence (subject to some tolerance).

Example - Beetle Mortality (I)



Example - Beetle Mortality (II)

```
##   dead alive  dose
## 1   33     2 20.68
## 2   24     7 17.51
## 3   30     4 19.05
## 4   26     7 14.42
## 5   25     5 15.58
## 6   27     6 18.25
```

The **glm** function in **R** gives

```r
coef( glm( cbind(dead,alive) ~ dose,
           family=binomial, data=beetles ) )
```

```
## (Intercept)        dose
##   -2.058899    0.207153
```

Example - Beetle Mortality (III)

We have a binomial regression model with

$$\eta = \log\left(\frac{\mu}{1-\mu}\right)$$

$$\frac{\partial \eta}{\partial \mu} = \frac{1}{\mu(1-\mu)}$$

$$V(\mu) = \mu(1-\mu)$$

$$w = m\mu(1-\mu).$$

Example - Beetle Mortality (IV)

```r
m  <- beetles$dead + beetles$alive
y  <- beetles$dead / m
mu <- y
eta <- log( mu / (1-mu) )
z <- eta + (y-mu) / ( mu * (1-mu) )
w <- m * mu * ( 1 - mu )
lmod <- lm( z ~ dose, weights=w, data=beetles )
coef( lmod )
```

```
## (Intercept)        dose
##  -2.0215414   0.2034804
```

Example - Beetle Mortality (V)

```r
for( i in 1:3 ){
  eta <- lmod$fit
  mu <- exp(eta) / ( 1 + exp(eta) )
  z <- eta + (y-mu) / ( mu * (1-mu) )
  w <- m * mu * ( 1 - mu )
  lmod <- lm( z ~ dose, weights=w, data=beetles )
  print( coef( lmod ) )
}
```

```
## (Intercept)        dose
##  -2.0584665   0.2071151
## (Intercept)        dose
##   -2.058899    0.207153
## (Intercept)        dose
##   -2.058899    0.207153
```

Estimation of ϕ

Since $\mathrm{Var}(Y_i) = \phi V(\mu_i)$,

$$\frac{1}{\phi}\sum_{i=1}^{n}\frac{(y_i-\hat{\mu}_i)^2}{V(\hat{\mu}_i)} \sim \chi^2_{n-p} \text{ approximately.}$$

Due to the expectation of a χ^2_{n-p} random variable being $(n-p)$, this suggests an estimator for ϕ as

$$\hat{\phi}_P = \frac{1}{n-p}\sum_{i=1}^{n}\frac{(y_i-\hat{\mu}_i)^2}{V(\hat{\mu}_i)},$$

the Pearson's chi-square statistic $\mathcal{X}^2$, scaled by the degrees of freedom.

For the (normal regression) model, $\hat{\phi}_P$ is identical to $\hat{\sigma}^2$.