Lecture 3.2: Logistic Regression MLE and Log-Likelihood

BIOS 6612

Julia Wrobel

February 8, 2021

Overview

Today, we cover:

- Logistic regression likelihood function
- MLE for coefficients in logistic regression

Readings:

- Agresti, 5 up to 5.23

Likelihood for logistic regression

$\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} \ldots \beta_p X_{pi}$

Start with $Y_1, Y_2, \ldots, Y_n \sim Bernoulli(\pi)$

- PDF is $f(y_i; \pi) = \pi^{y_i}(1-\pi)^{1-y_i}$
- Model for $\pi_i = P(Y_i = 1|X_i)$ is $\text{logit}(\pi_i) = \mathbf{x}_i\boldsymbol{\beta}$
- For simplicity, let $\mathbf{x}_i\boldsymbol{\beta} = \beta_0 + \beta_1 x_{1i}$

$\pi_i = e^{X_i \beta}$

$$\prod_{i=1}^{n} f(y_i; \pi_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i} \leftarrow \pi_i$$

$$= \prod_{i=1}^{n} \left[e^{\beta_0 + \beta_1 X_{1i}}\right]^{y_i} \left[1 - e^{\beta_0 + \beta_1 X_{1i}}\right]^{1-y_i} = L(\beta_0, \beta_1; y_i, X_i)$$

coloradoSPH

Likelihood for logistic regression

End up with

β_1, β_0

$$L(\boldsymbol{\beta}; Y_i, X_i) = \prod_{i=1}^{n} \left(\frac{e^{\beta_0+\beta_1 x_{1i}}}{1 + e^{\beta_0+\beta_1 x_{1i}}} \right)^{Y_i} \left(1 - \frac{e^{\beta_0+\beta_1 x_{1i}}}{1 + e^{\beta_0+\beta_1 x_{1i}}} \right)^{1-y_i}$$

$$l(\boldsymbol{\beta}; Y_i, X_i) = \sum_{i=1}^{n} \left[Y_i \log \left(\frac{e^{\beta_0+\beta_1 x_{1i}}}{1 + e^{\beta_0+\beta_1 x_{1i}}} \right) + (1 - Y_i) \log \left(1 - \frac{e^{\beta_0+\beta_1 x_{1i}}}{1 + e^{\beta_0+\beta_1 x_{1i}}} \right) \right]$$

Likelihood and log likelihood are given above. In principle, we can solve these for the MLE of β_0 and β_1 the same way we have done before to obtain estimates for the regression coefficients.

$$U(\beta_0) = \frac{\partial l(\beta_0)}{\partial \beta_0} \overset{set}{=} 0$$

coloradoSPH

Score functions for coefficients in logistic regression

$p = 2$

β_0 $X_1, X_2 \ldots X_p$ $\beta_0 + \beta_1 X_1 + \beta_2 X_2$

Number of score functions will be equal to $p + 1$ where p is number of covariates * If $p = 2$ need to estimate $\beta_0, \beta_1, \beta_2$

$$U(\beta_0) = \frac{\partial l(\beta_0, \beta_1; Y_i, X_i)}{\partial \beta_0}$$

$$U(\beta_1) = \frac{\partial l(\beta_0, \beta_1; Y_i, X_i)}{\partial \beta_1}$$

- Solution at $U(\beta) = 0$ cannot be obtained analytically (by hand)

Information for coefficients in logistic regression

$p=1$ β_0, β_1

- $I(\boldsymbol{\beta})$ will be a $(p+1)\times(p+1)$ matrix of the partial second derivatives of l with respect to the parameters β

$$\begin{pmatrix} [I(\boldsymbol{\beta})]_{1,1} & [I(\boldsymbol{\beta})]_{1,2} \\ [I(\boldsymbol{\beta})]_{2,1} & [I(\boldsymbol{\beta})]_{2,2} \end{pmatrix}$$

2×2

$$[I(\boldsymbol{\beta})]_{1,1} = \frac{\partial^2 l(\beta_0, \beta_1; Y_i, X_i)}{\partial \beta_0^2}$$

$$[I(\boldsymbol{\beta})]_{1,2} = \frac{\partial}{\partial \beta_1}\left(\frac{\partial l(\beta)}{\partial \beta_0}\right)$$

$||$

$$[I(\boldsymbol{\beta})]_{2,1} = \frac{\partial}{\partial \beta_0}\left(\frac{\partial l(\beta)}{\partial \beta_1}\right)$$

$$[I(\boldsymbol{\beta})]_{2,2} = \frac{\partial^2 l(\beta_0, \beta_1; Y_i, X_i)}{\partial \beta_1^2}$$

MLE for coefficients in logistic regression

Cannot be solved analytically (by hand)

- Need to use numerical algorithm to obtain MLE

Numerical methods search through different possible values of β_0 and β_1 to obtain values that give the highest possible value of $l(\theta)$

- To do inference, you need a numerical estimate of the information matrix too
  - $I(\theta)$ is often called the *gradient* in numerical optimization settings

Alternative Fitting Methods

What do we do when MLE of β can't be solved for analytically?

- Use a numerical algorithm

Example: Lung cancer and exposure to passive smoking (second hand smoke)

- `passive`: 0 or 1, indicates exposure to passive smoking
- `y`: cancer cases
- `n`: total observations

```
##     y   n passive
## 1 281 491       1
## 2 228 507       0
```

Fisher Scoring

How both SAS and R fit logistic regression models.

- Numerical optimization for MLEs specific to GLMs
- Uses Score function and Expected Fisher information

Fisher Scoring

We fit logistic regression (and other GLMs) in `R` using the `glm()` function

```
logistic_regression = glm(cbind(y, n-y) ~ passive,
                          data = passive_smoking,
                          family = binomial)

logLik(logistic_regression)

## 'log Lik.' -6.649433 (df=2)
```

$\hat{\beta}_0$ $\hat{\beta}_1$ -6.649

Numerical Optimization

Many more general numerical optimization method exist

- Goal is to optimize (minimize or maximize) some function that can't be solved by hand
- Applicable to much more general settings.
- Need to define a log likelihood function.

```
logit_LL = function(beta, y, n, xmat){
  eta = cbind(1, xmat) %*% beta # linear predictor
  pi = plogis(eta) # the estimated probability
  sum(dbinom(x = y, size = n, prob = pi, log = TRUE)) # sum up the
individual log probability mass functions
}
```

```
logit_LL(beta = coef(logistic_regression),
         y = passive_smoking$y,
         n = passive_smoking$n,
         xmat = as.matrix(passive_smoking$passive))
```

```
## [1] -6.649433
```

glm()

Numerical Optimization

$\beta_0, \beta_1, \beta_2$

$\beta = 0$ β_0 β_0

Also need to specify starting values for β

```
start_beta = c(0,0)
start_beta2 = c(0,0.5)

logit_LL(beta = start_beta2,
         y = passive_smoking$y,
         n = passive_smoking$n,
         xmat = as.matrix(passive_smoking$passive))

## [1] -11.80684
```

< -6.079

```
numerical_opt = optim(start_beta, # starting values for coefficients
                      logit_LL, # function to optimize
                      hessian = TRUE, # calculates numerical gradient
(information) so we can get standard errors
                      # arguments to log likelihood function
                      y = passive_smoking$y,
                      n = passive_smoking$n,
                      xmat = as.matrix(passive_smoking$passive),
                      method = "BFGS", # optimization method (quasi-Newton)
                      control=list(fnscale = -1) # tells optim() to maximize
instead of minimize
                      )
```

$\rightarrow I(\theta)$

Comparison of MLE estimates

Compare estimates from `glm()` and `optim()`

- β values

```
## (Intercept)     passive
##  -0.2018662   0.4931133
## [1] -0.2018661  0.4931132
```

glm( )

optim( )

$I(\beta)_{2\times 2}$

- Covariance matrix

```
##                    (Intercept)      passive
## (Intercept)   0.007970194 -0.007970194
## passive      -0.007970194  0.016290818
##                    [,1]         [,2]
## [1,]  0.007970196 -0.007970196
## [2,] -0.007970196  0.016290820
```

glm

optim

→ Hessian

$cov(\beta_0, \beta_1)$

$= [I(\beta)]^{-1}$

colorado**SPH**