Generalized Linear Models (3)

STA 211: The Mathematics of Regression

Yue Jiang

April 18, 2023

The following material was used by Yue Jiang during a live lecture.

Without the accompanying oral comments, the text is incomplete as a record of the presentation.

Review: GLMs

A **generalized linear model** has three components:

1. An outcome Y that follows a distribution from the exponential family*
2. The linear predictor $\mathbf{X}\beta$
3. A link function g that links the conditional expectation of Y with the linear predictor:

$$E(Y|\mathbf{X}) = g^{-1}(\mathbf{X}\beta)$$

Review: Poisson regression

$$\log(\underbrace{E(Y|\mathbf{X})}_{\lambda}) = \mathbf{X}^T\boldsymbol{\beta}$$

Generalized linear model often used for count (or rate) data, assuming outcome has Poisson distribution and using log link

Poisson regression

$$\begin{aligned}\log \mathcal{L} &= \sum_{i=1}^{n} (y_i \log \lambda - \lambda - \log y_i!) \\ &= \sum_{i=1}^{n} y_i \mathbf{X}_i \boldsymbol{\beta} - e^{\mathbf{X}_i \boldsymbol{\beta}} - \log y_i!\end{aligned}$$

We would like to solve the equations

$$\left(\frac{\partial \log \mathcal{L}}{\partial \beta_j}\right) \overset{set}{=} \mathbf{0},$$

Newton-Raphson in higher dimensions

Score vector and **Hessian** for $\log \mathcal{L}(\boldsymbol{\theta}|\mathbf{X})$ with $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_p)^T$:

$$\nabla \log \mathcal{L} = \begin{pmatrix} \frac{\partial \log \mathcal{L}}{\partial \boldsymbol{\theta}_1} \\ \vdots \\ \frac{\partial \log \mathcal{L}}{\partial \boldsymbol{\theta}_p} \end{pmatrix}$$

$$\nabla^2 \log \mathcal{L} = \begin{pmatrix} \frac{\partial^2 \log \mathcal{L}}{\partial \theta_1^2} & \frac{\partial^2 \log \mathcal{L}}{\partial \theta_1 \theta_2} & \cdots & \frac{\partial^2 \log \mathcal{L}}{\partial \theta_1 \theta_p} \\ \frac{\partial^2 \log \mathcal{L}}{\partial \theta_2 \theta_1} & \frac{\partial^2 \log \mathcal{L}}{\partial \theta_2^2} & \cdots & \frac{\partial^2 \log \mathcal{L}}{\partial \theta_2 \theta_p} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \log \mathcal{L}}{\partial \theta_p \theta_1} & \frac{\partial^2 \log \mathcal{L}}{\partial \theta_p \theta_2} & \cdots & \frac{\partial^2 \log \mathcal{L}}{\partial \theta_p^2} \end{pmatrix}$$

Newton-Raphson in higher dimensions

1. Start with initial guess $\boldsymbol{\theta}^{(0)}$
2. Iterate
   $$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \left(\nabla^2 \log \mathcal{L}(\boldsymbol{\theta}^{(t)}|\mathbf{X})\right)^{-1} \left(\nabla \log \mathcal{L}(\boldsymbol{\theta}^{(t)}|\mathbf{X})\right)$$
3. Stop when convergence criterion is satisfied

Under certain conditions, a global maximum exists; this again is guaranteed for many common applications.

Computing the Hessian can be computationally demanding (and annoying), but there are ways around it in practice.

Poisson regression

$$\log \mathcal{L} = \sum_{i=1}^{n} y_i \mathbf{X}_i \beta - e^{\mathbf{X}_i \boldsymbol{\beta}} - \log y_i!$$

$$\nabla \log \mathcal{L} = \sum_{i=1}^{n} \left(y_i - e^{\mathbf{X}_i \boldsymbol{\beta}} \right) \mathbf{X}_i^T$$

$$\nabla^2 \log \mathcal{L} = -\sum_{i=1}^{n} e^{\mathbf{X}_i \boldsymbol{\beta}} \mathbf{X}_i \mathbf{X}_i^T$$

Newton-Raphson update steps for Poisson regression:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left(-\sum_{i=1}^{n} e^{\mathbf{X}_i \boldsymbol{\beta}^{(t)}} \mathbf{X}_i \mathbf{X}_i^T \right)^{-1} \left(\sum_{i=1}^{n} \left(y_i - e^{\mathbf{X}_i \boldsymbol{\beta}^{(t)}} \right) \mathbf{X}_i^T \right)$$

Back to bike crashes

```
m1 <- glm(crashes ~ traffic_vol + pct_rural,
          data = bike, family = "poisson")
round(summary(m1)$coef, 6)
```

```
##              Estimate Std. Error    z value Pr(>|z|)
## (Intercept)  5.982181   0.053749 111.298625        0
## traffic_vol  0.001541   0.000166   9.262671        0
## pct_rural   -0.044558   0.000875 -50.919036        0
```

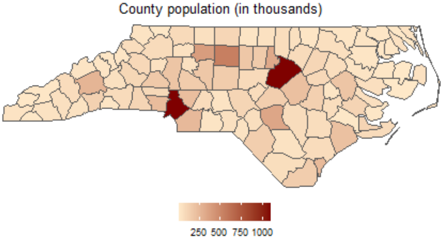
Back to bike crashes

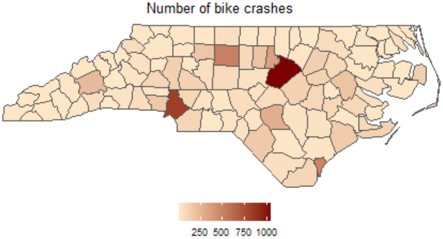
```r
while(delta > 0.000001 & iter < 500){
  old <- beta
  beta <- old - solve(d2func(beta = beta, X = X, y = y)) %
    d1func(beta = beta, X = X, y = y)
  temp[iter,] <- beta
  delta <- sqrt(sum((beta - old)^2))
  iter <- iter + 1
}
```

Back to bike crashes

```
iter
## [1] 22

delta
## [1] 3.911961e-07

beta
##             [,1]
## [1,]  5.98218054
## [2,]  0.00154064
## [3,] -0.04455809

m1$coefficients
## (Intercept) traffic_vol   pct_rural
##  5.98218054  0.00154064 -0.04455809
```

Back to bike crashes





Two potential models

$$\log\left(E(Y|\mathbf{X})\right) = \beta_0 + \beta_1(pop) + \beta_2(traffic) + \beta_3(rural)$$

$$\log\left(\frac{E(Y|\mathbf{X})}{pop}\right) = \beta_0 + \beta_1(traffic) + \beta_2(rural)$$

- What are the differences in the two models above?
- How is population being used, and how might it be interpreted?

Two potential models

$$\log\left(E(Y|\mathbf{X})\right) = \beta_0 + \beta_1(pop) + \beta_2(traffic) + \beta_3(rural)$$

```
m2 <- glm(crashes ~ traffic_vol + pct_rural + pop,
data = bike, family = "poisson")
```

$$\log\left(\frac{E(Y|\mathbf{X})}{pop}\right) = \beta_0 + \beta_1(traffic) + \beta_2(rural)$$

```
m3 <- glm(crashes ~ traffic_vol + pct_rural,
offset = log(pop),
data = bike, family = "poisson")
```

Two potential models

```
##              Estimate Std. Error    z value Pr(>|z|)
## (Intercept)  5.655725   0.054837 103.136325 0.000000
## traffic_vol -0.000093   0.000179  -0.518756 0.603931
## pct_rural   -0.037761   0.000878 -43.015409 0.000000
## pop          0.000001   0.000000  30.215337 0.000000
```

Two potential models

```
round(summary(m3)$coef, 6)
##              Estimate Std. Error     z value Pr(>|z|)
## (Intercept) -6.916803   0.054480 -126.961100 0.000000
## traffic_vol -0.000047   0.000171   -0.272118 0.785531
## pct_rural   -0.010936   0.000857  -12.766690 0.000000
```

- Can we simply use `bike$crashes/pop` as our outcome variable in the code we've already written?

Comparing results

```
round(beta, 6)
##           [,1]
## [1,] -6.810266
## [2,]  0.000314
## [3,] -0.011783

round(m3$coefficients, 6)
## (Intercept) traffic_vol   pct_rural
##   -6.916803   -0.000047   -0.010936
```

Comparing results

```
m3_wrong <- glm(crashes/pop ~ traffic_vol + pct_rural,
data = bike, family = "poisson")

round(m3_wrong$coefficients, 6)
## (Intercept) traffic_vol   pct_rural
##   -6.810266    0.000314   -0.011783
```

The offset model

Denote the offset by $\boldsymbol{\omega}$. If we directly use `crashes/pop` in the Poisson regression likelihood, we would have a log-likelihood along the lines of

$$\log \mathcal{L} \propto \sum_{i=1}^{n} \frac{y_i}{\omega_i} \mathbf{X}_i \boldsymbol{\beta} - e^{\mathbf{X}_i \boldsymbol{\beta}}$$

- What do you think of this approach?

The actual log-likelihood

The model with offset is given by:

$$\log\left(E(Y|\mathbf{X})\right) = \mathbf{X}^T\beta - \log\boldsymbol{\omega}$$

- What is the actual log-likelihood function to maximize here (note, $\boldsymbol{\omega}$ is also observed data)?
- What are the actual Newton-Raphson steps here?
- Can you write code that numerically implements this model?

Last homework!

1. The last two questions on the previous slide. For the last question (code), run your code on the dataset from last week (this question will only be worth 2 points out of 10).

(I figured people might want more practice with this; originally this lecture/homework was going to be about Fisher scoring and iteratively-reweighted least squares, but we'll cover those in our last lecture next week instead).