Generalized Linear Models (2)

STA 211: The Mathematics of Regression

Yue Jiang

April 11, 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.

Bike crashes

We have data that represented the number of bike crashes per year for each North Carolina county. For instance:

- Alexander: 1
- Alleghany: 1
- Anson: 7
- Ashe: 4
- etc.

Suppose we thought these crashes came from a Poisson distribution with parameter λ: $f_Y(y) = \frac{\lambda^y \exp(-\lambda)}{y!}$.

- How might you estimate the parameter of this Poisson distribution, given our observed data?

Review: Maximum likelihood estimation

We can maximize the likelihood function. Assuming the observations are i.i.d., in general we have:

$$\begin{aligned}\mathcal{L}(\lambda|Y) &= f(y_1, y_2, \cdots, y_n|\lambda) \\ &= f(y_1|\lambda)f(y_2|\lambda)\cdots f(y_n|\lambda) \\ &= \prod_{i=1}^{n} f(y_i|\lambda).\end{aligned}$$

The likelihood function is the probability of "seeing our observed data," *given* a value of λ. Remember, do not get $f(y_i|\lambda)$ confused with $f(\lambda|y_i)$!

- If $Y_1, Y_2, \cdots, Y_n$ are each i.i.d. distributed with $Pois(\lambda)$, then what is the MLE of λ?

Review: Maximum likelihood estimation

$$\begin{aligned}
\mathcal{L}(\lambda|Y) &= \prod_{i=1}^{n} f(y_i|\lambda) \\
&= \prod_{i=1}^{n} \frac{\lambda_i^y e^{-\lambda}}{y_i!} \\
\log \mathcal{L}(\lambda|Y) &= \sum_{i=1}^{n} (y_i \log \lambda - \lambda - \log y_i!) \\
&= \log \lambda \sum_{i=1}^{n} y_i - n\lambda - \sum_{i=1}^{n} \log y_i!
\end{aligned}$$

Review: Maximum likelihood estimation

Setting the **score function** equal to 0:

$$\frac{\partial}{\partial \lambda} \log \mathcal{L}(\lambda | Y) = \frac{1}{\lambda} \sum_{i=1}^{n} y_i - n \overset{set}{=} 0$$

$$\implies \hat{\lambda} = \frac{1}{n} \sum_{i=1}^{n} y_i,$$

as expected. Next, let's verify that $\hat{\lambda}$ is indeed a maximum:

$$\frac{\partial^2}{\partial \lambda^2} \log \mathcal{L}(\lambda | Y) = -\frac{1}{\lambda^2} \sum_{i=1}^{n} y_i - n$$

$$< 0.$$

Can we do better?

We might expect that more populous, more urban counties might have more crashes. There might also be a relationship with traffic volume.

- Can we incorporate this additional information while accounting for potential confounding?

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

- Can we differentiate the (log) likelihood function, set it equal to zero, and solve for the MLEs for $\boldsymbol{\beta} = (\beta_0, \beta_1, \cdots, \beta_p)$ as before?

Poisson regression

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

We would like to solve the equations

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

but there is no closed-form solution, as this is a transcendental equation in the parameters of interest.

- How might we solve these equations numerically?

A one-dimensional problem

Suppose you're trying to find the maximum of the following function:

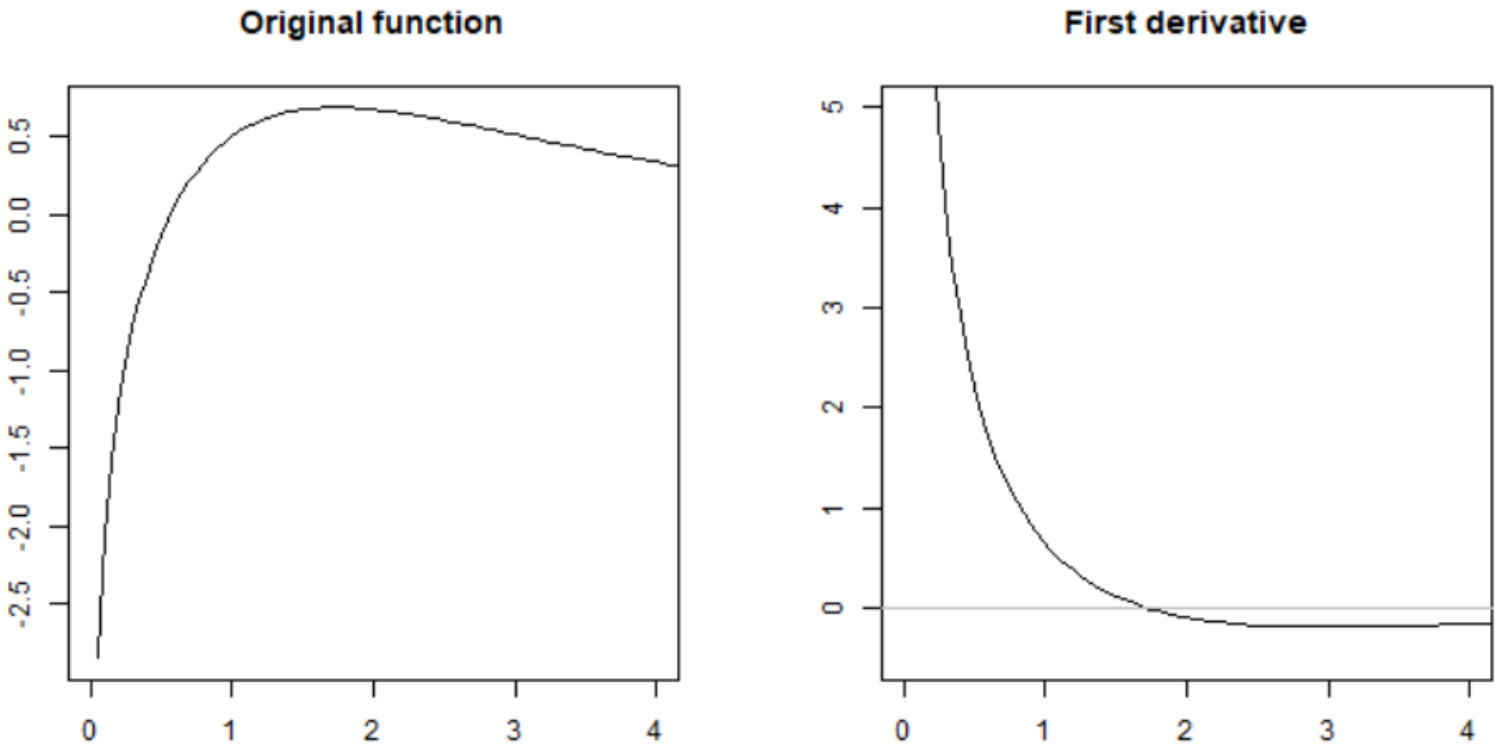
$$f(x) = \frac{x + \log(x)}{2^x}$$

Let's try differentiating, setting equal to 0, and solving:

$$\frac{d}{dx} f(x) = 2^{-x} \left(1 + \frac{1}{x} - \log(2)(x + \log(x))\right).$$

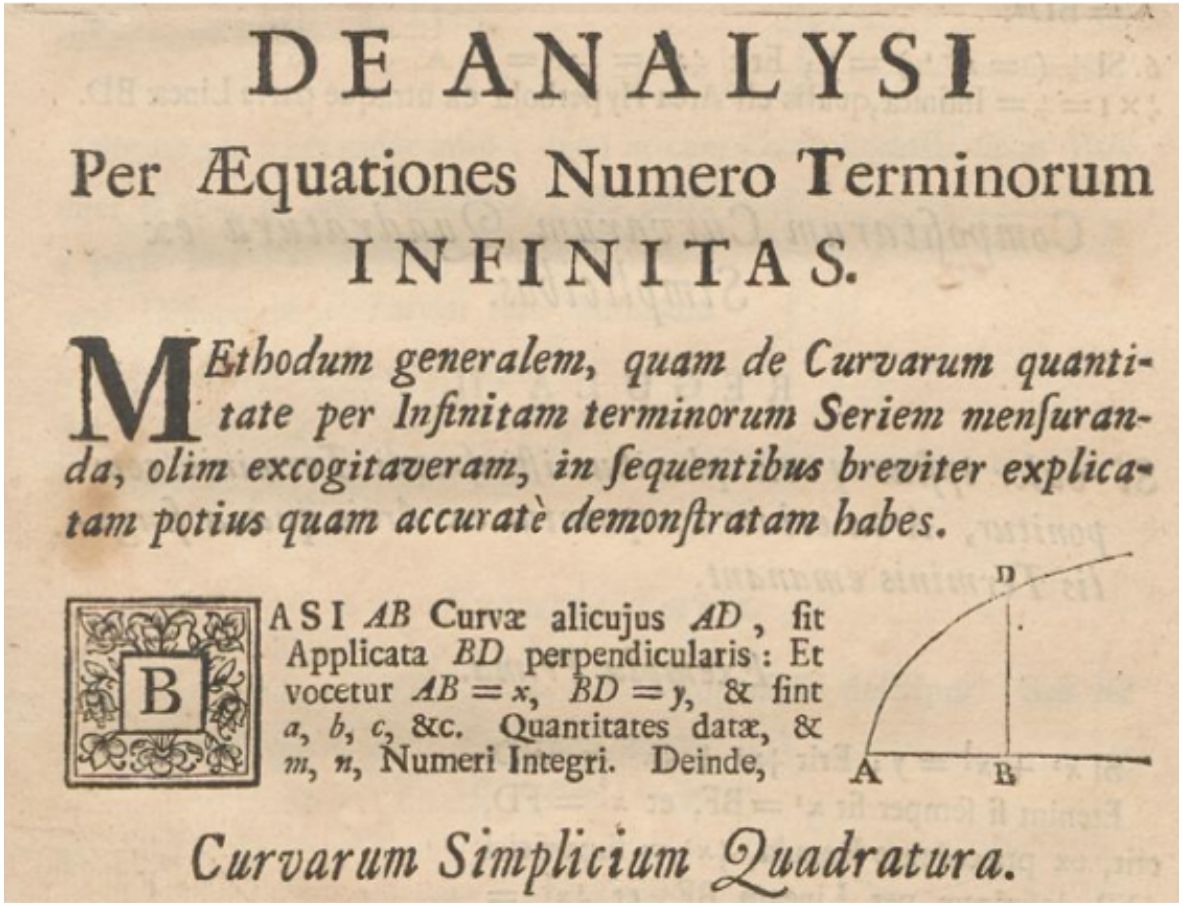
We run into a similar problem: we cannot algebraically solve for the root of this equation.

A one-dimensional problem



- It looks like the maximum is a bit shy of 2 (trust me on this one, it's a global maximum). How might we find where it is?

A one-dimensional problem



DE ANALYSI

Per Æquationes Numero Terminorum INFINITAS.

Methodum generalem, quam de Curvarum quantitate per Infinitam terminorum Seriem mensuranda, olim excogitaveram, in sequentibus breviter explicatam potius quam accuratè demonstratam habes.

BASI *AB* Curvæ alicujus *AD*, sit Applicata *BD* perpendicularis: Et vocetur $AB = x$, $BD = y$, & sint a, b, c, &c. Quantitates datæ, & m, n, Numeri Integri. Deinde,

Curvarum Simplicium Quadratura.

- It looks like the maximum is a bit shy of 2 (trust me on this one, it's a global maximum). How might we find where it is?

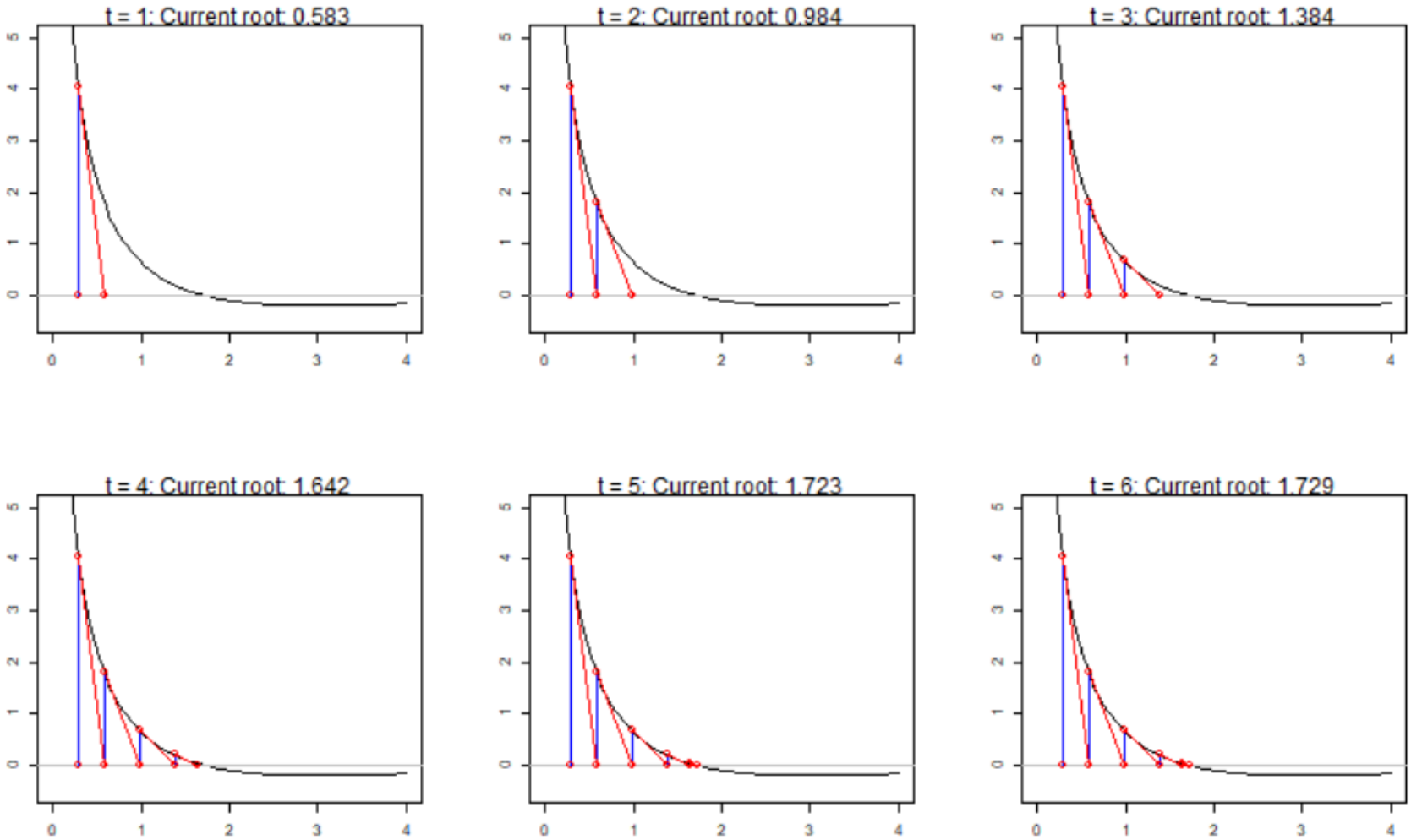
A one-dimensional problem

Newton-Raphson algorithm for root finding is based on second-order Taylor approximation around true root:

1. Start with initial guess $\theta^{(0)}$
2. Iterate $\theta^{(t+1)} = \theta^{(t)} - \frac{f'(\theta^{(t)})}{f''(\theta^{(t)})}$
3. Stop when convergence criterion is satisfied

Although it requires explicit forms of first two derivatives, the convergence speed is quite fast.
There are some necessary conditions for convergence, but this is beyond the scope of STA 211. Many likelihood functions you are likely to encounter (e.g., **GLMs with canonical link**) will in fact converge from any starting value.

A one-dimensional problem



A one-dimensional problem

$$f(x) = \frac{x + \log(x)}{2^x}$$

$$\frac{d}{dx} f(x) = 2^{-x} \left(1 + \frac{1}{x} - \log(2)(x + \log(x)) \right).$$

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

We can modify the Newton-Raphson algorithm for 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

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

- What are the score vector and Hessian corresponding to the Poisson regression log-likelihood? What would the Newton-Raphson update steps be?

Newton-Raphson in higher dimensions

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

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

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

Newton-Raphson update steps for Poisson regression:

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

Newton-Raphson in higher dimensions

Newton-Raphson update steps for Poisson regression:

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

Some R code

```
d1func <- function(beta, X, y){
  d1 <- rep(0, length(beta))
  for(i in 1:length(y)){
    d1 <- d1 + (y[i] - exp(X[i,] %*% beta)) %*% X[i,]
  }
  return(colSums(d1))
}
```

Some R code

```
d2func <- function(beta, X, y){
  d2 <- matrix(0, nrow = length(beta), ncol = length(beta))
  for(i in 1:length(y)){
    d2 <- d2 - t((exp(X[i,] %*% beta)) %*% X[i,]) %*% (X[i,
  }
  return(d2)
}
```

Some R code

```
beta <- c(mean(log(y)), 0, 0)
X <- cbind(1, x_1, x_2)
y <- y
iter <- 1
delta <- 1

temp <- matrix(0, nrow = 500, ncol = 3)
```

Some R code

```
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
}
```

Homework (Due April 18) - part 1 of 2

1. Derive the score and Hessian functions of the log-likelihood for a logistic regression model (i.e., binary regression under canonical link).
2. The file `bikecrash_agg.csv` on Sakai/Resources contains yearly fatal bike crash data for each of North Carolina's 100 counties in 2017. Implement a **Poisson** regression model "by hand" (i.e., without using `glm()`) that predicts the number of fatal bike crashes based on `pop` (population) and `traffic_vol` (traffic volume), and also be sure to include an intercept term by adapting the code from lecture (you might have to copy/paste to get to the stuff that's off-screen). Verify that your estimates match what you get from using `glm()` (hint: it should).

Homework (Due April 18) - part 1 of 2

3. Implement a **logistic** regression model "by hand" (i.e., without using `glm()`) that predicts whether there are more than 50 fatal bike crashes per 100,000 residents based on `pop` and `traffic_vol`, and also be sure to include an intercept term. Note that 45 counties should satisfy this criterion. Verify that your estimates match what you get from using `glm()` (hint: it should).

Important note: Ex. 1 and Ex. 2 are worth 13 of 15 total points on this assignment; Ex. 3 is worth only 2 points.