Fitting logistic regression models

Fisher scoring for logistic regression

$$Y_i \sim \text{Bernoulli}(p_i)$$

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_n x_{in}$$

Given $(x_1, y_1), \ldots, (x_n, y_n)$. want to estimate $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}$

$\Rightarrow$ maximize $L(\beta)$ (or $\ell(\beta)$)

Fisher scoring

1) Initial guess $\beta^{(0)}$

2) update: $\beta^{(r+1)} = \beta^{(r)} + \mathcal{I}^{-1}(\beta^{(r)})\, u(\beta^{(r)})$

$$u(\beta) = \frac{\partial \ell(\beta)}{\partial \beta} \qquad \mathcal{I}(\beta) = \mathbb{E}\left[\frac{-\partial u(\beta)}{\partial \beta}\right]$$

3) Stop when $\beta^{(r+1)} \approx \beta^{(r)}$

$\downarrow$

$\hat{\beta}$

Example: $\ell(\beta) = \sum_{i=1}^{n} \left\{ Y_i(\beta_0 + \beta_1 X_i) - \log\left(1 + e^{\beta_0 + \beta_1 X_i}\right) \right\}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$

$$u(\beta) = \begin{bmatrix} \sum_{i=1}^{n} \left(Y_i - \frac{e^{\beta_0 + \beta_1 X_i}}{1 + e^{\beta_0 + \beta_1 X_i}} \right) \\ \sum_{i=1}^{n} \left(Y_i - \frac{e^{\beta_0 + \beta_1 X_i}}{1 + e^{\beta_0 + \beta_1 X_i}} \right) X_i \end{bmatrix}$$

$$\mathcal{I}(\beta) = \mathbb{E}\left[-\frac{\partial u(\beta)}{\partial \beta} \right] = \begin{bmatrix} \sum_{i=1}^{n} \frac{e^{\beta_0 + \beta_1 X_i}}{(1 + e^{\beta_0 + \beta_1 X_i})^2} & \sum_{i=1}^{n} \frac{e^{\beta_0 + \beta_1 X_i}}{(1 + e^{\beta_0 + \beta_1 X_i})^2} X_i \\ \sum_{i=1}^{n} \frac{e^{\beta_0 + \beta_1 X_i}}{(1 + e^{\beta_0 + \beta_1 X_i})^2} X_i & \sum_{i=1}^{n} \frac{e^{\beta_0 + \beta_1 X_i}}{(1 + e^{\beta_0 + \beta_1 X_i})^2} X_i^2 \end{bmatrix}$$

Practice question: Fisher scoring

solve(...)

$y_0 * y_0$

Suppose that $\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_i$, and we have

$$\beta^{(r)} = \begin{bmatrix} -3.1 \\ 0.9 \end{bmatrix}, \quad U(\beta^{(r)}) = \begin{bmatrix} 9.16 \\ 31.91 \end{bmatrix},$$

$$\mathcal{I}(\beta^{(r)}) = \begin{bmatrix} 17.834 & 53.218 \\ 53.218 & 180.718 \end{bmatrix}$$

> Use the Fisher scoring algorithm to calculate $\beta^{(r+1)}$ (you may use R or a calculator, you do not need to do the matrix arithmetic by hand). Take 2--3 minutes, then we will discuss.

$$\beta^{(r+1)} = \beta^{(r)} + \mathcal{I}^{-1}(\beta^{(r)})\, u(\beta^{(r)})$$

$$\begin{bmatrix} -3.1 \\ 0.9 \end{bmatrix} + \begin{bmatrix} 17.834 & 53.248 \\ 53.248 & 180.718 \end{bmatrix}^{-1} \begin{bmatrix} 9.16 \\ 31.91 \end{bmatrix}$$

$$= \begin{bmatrix} -3.21 \\ 1.11 \end{bmatrix}$$

we get closer!

Actual MLE: $\begin{bmatrix} -3.360 \\ 1.174 \end{bmatrix}$

Alternative to Fisher scoring: gradient ascent

$$Y_i \sim Bernoulli(p_i)$$

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_k X_{i,k}$$

Choose $\beta = (\beta_0, \ldots, \beta_k)^T$ to maximize $L(\beta)$.

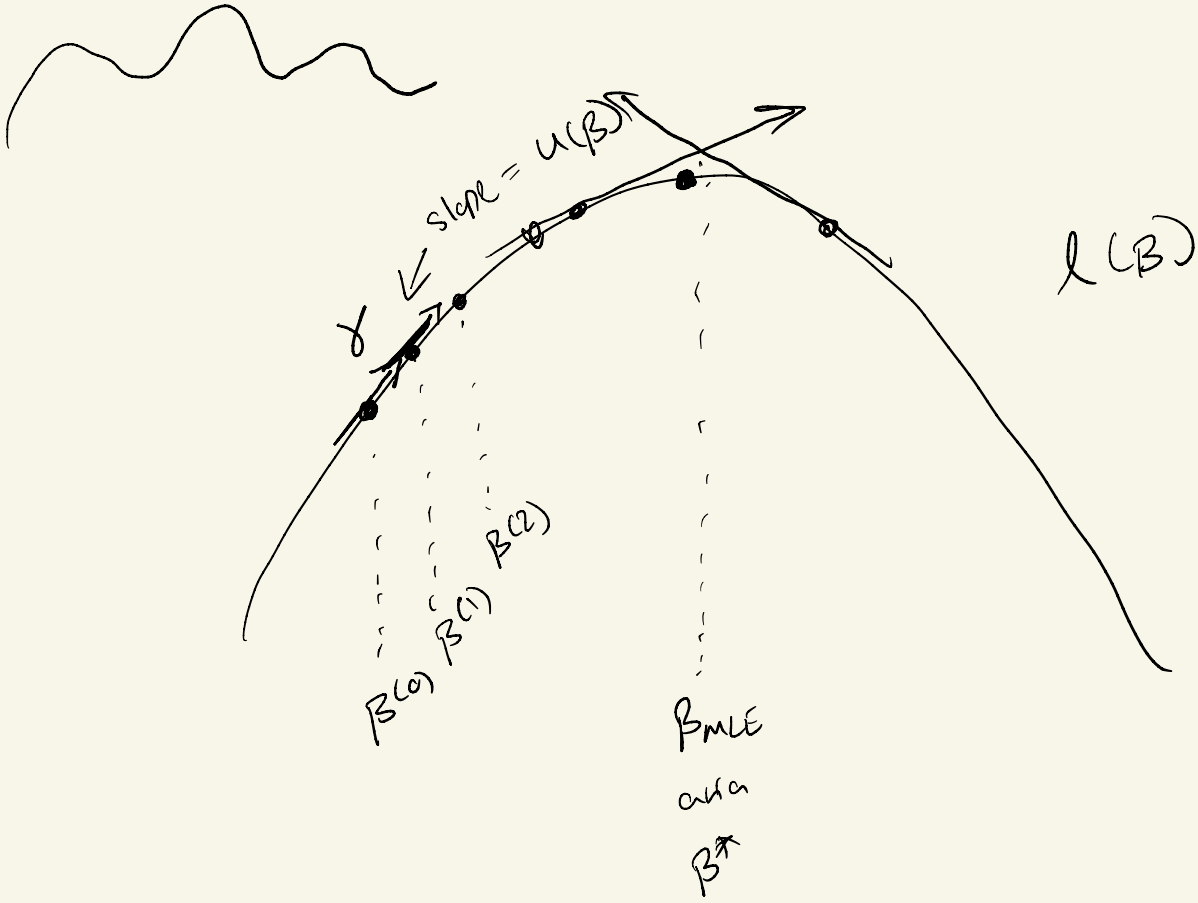
Gradient ascent:

1) Initial guess $\beta^{(0)}$

2) update: $\beta^{(r+1)} = \beta^{(r)} + \gamma\, u(\beta^{(r)})$ ← $\gamma > 0$ (usually close to 0) learning rate or stepsize

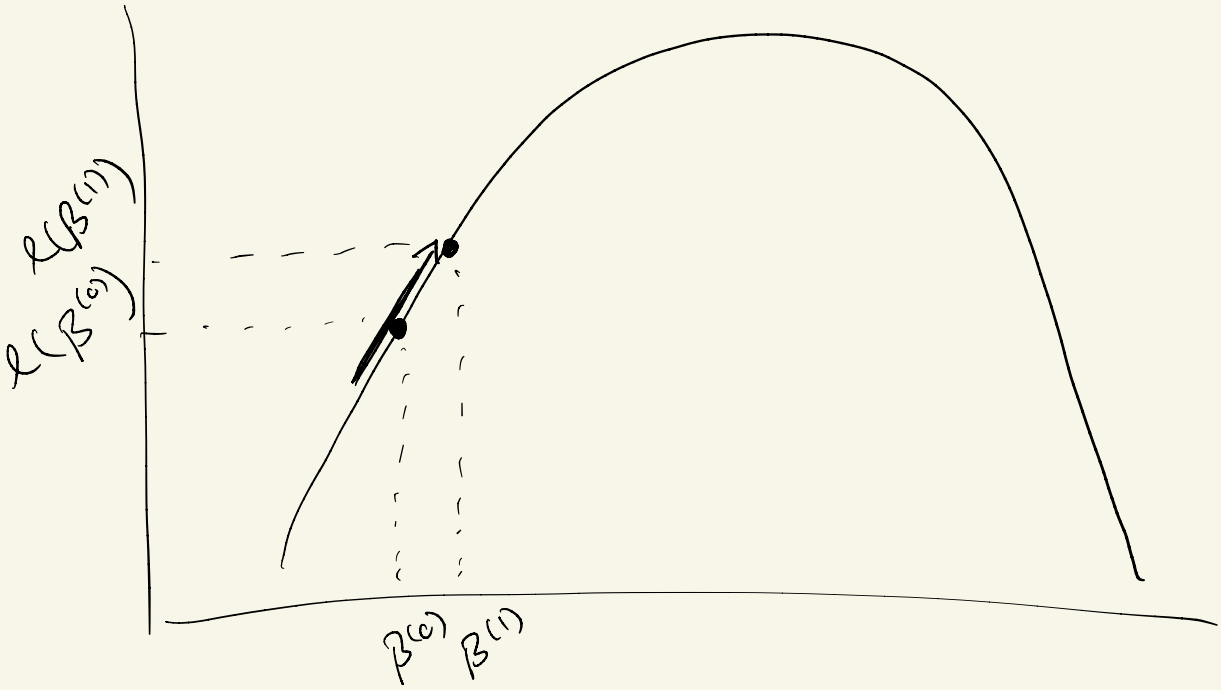
3) Stop when $\beta^{(r+1)} \approx \beta^{(r)}$

Simpler than Fisher scoring



If γ is small enough, then eventually $\beta^{(t+1)} \approx \beta_{MLE}$

$$\text{slope} = u(\beta) \approx \frac{\ell(\beta^{(1)}) - \ell(\beta^{(0)})}{\beta^{(1)} - \beta^{(0)}}$$



Motivation for gradient ascent: walking uphill

Practice question: gradient ascent

Suppose that $\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_i$, and we have

$$\beta^{(r)} = \begin{bmatrix} -3.1 \\ 0.9 \end{bmatrix}, \quad U(\beta^{(r)}) = \begin{bmatrix} 9.16 \\ 31.91 \end{bmatrix}$$

- Use gradient ascent with a learning rate (aka step size) of $\gamma = 0.01$ to calculate $\beta^{(r+1)}$.
- The actual maximum likelihood estimate is $\widehat{\beta} = (-3.360, 1.174)$. Does one iteration of gradient ascent or Fisher scoring get us closer to the optimal $\widehat{\beta}$?
- Discuss in pairs for 2--3 minutes.

$$\beta^{(r+1)} = \beta^{(r)} + \gamma\, u(\beta^{(r)})$$

$$\begin{bmatrix} -3.1 \\ 0.9 \end{bmatrix} + 0.01 \begin{bmatrix} 9.16 \\ 31.91 \end{bmatrix}$$

$$= \begin{bmatrix} -3.008 \\ 1.219 \end{bmatrix}$$

probably too big

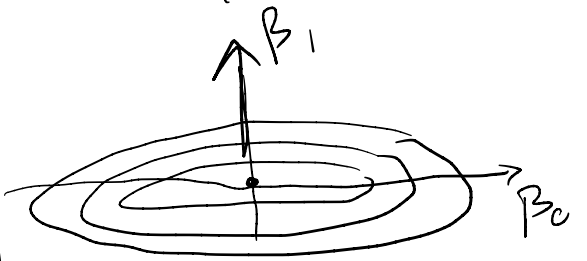
$$\begin{bmatrix} -3.36 \\ 1.174 \end{bmatrix}$$

Gradient ascent vs. Fisher scoring

step size γ ↓ uses $\mathcal{I}(\beta)$ ↓

Suppose $\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \in \mathbb{R}^2$

$\ell(\beta) = -\beta_0^2 - 100\beta_1^2$



$\ell(\beta)$ contour plot

maximized at $\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$u(\beta) = \begin{bmatrix} -2\beta_0 \\ -200\beta_1 \end{bmatrix}$$

$$\frac{-\partial u(\beta)}{\partial \beta} = \begin{bmatrix} \frac{\partial^2 \ell}{\partial \beta_0^2} & \frac{\partial^2 \ell}{\partial \beta_0 \partial \beta_1} \\ \frac{\partial^2 \ell}{\partial \beta_1 \partial \beta_0} & \frac{\partial^2 \ell}{\partial \beta_1^2} \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 \\ 0 & 200 \end{bmatrix} = \mathcal{I}(\beta)$$

$$\hat{I}(\beta) = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{200} \end{bmatrix}$$

$$\beta^{(r)} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}, \quad \gamma = 0.01$$

Gradient ascent:

$$\begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} + 0.01 \begin{bmatrix} -2(0.1) \\ -200(0.1) \end{bmatrix} = \begin{bmatrix} 0.098 \\ -0.1 \end{bmatrix}$$

Fisher scoring:

$$\begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} + \underbrace{\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{200} \end{bmatrix}}_{\text{adapte to shape of } \ell(\beta)} \begin{bmatrix} -2(0.1) \\ -200(0.1) \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Special topic: Feedforward neural networks

Fitting neural networks: stochastic gradient descent