Homework 2

Context

This assignment reinforces ideas in Module 2: Optimization. We focus specifically on implementing the Newton's method, EM, and MM algorithms.

Due date and submission

Please submit (via Canvas) a PDF containing a link to the web address of the GitHub repo containing your work for this assignment; git commits after the due date will cause the assignment to be considered late. Due date is Wednesday, 2/19 at 10:00AM.

Points

Problem	Points
Problem 0	15
Problem 1	30
Problem 2	5
Problem 3	30
Problem 4	20

Problem 0

This "problem" focuses on structure of your submission, especially the use git and GitHub for reproducibility, R Projects to organize your work, R Markdown to write reproducible reports, relative paths to load data from local files, and reasonable naming structures for your files.

To that end:

- create a public GitHub repo + local R Project; I suggest naming this repo / directory bios731_hw2_YourLastName (e.g. bios731_hw2_wrobel for Julia)
- Submit your whole project folder to GitHub
- Submit a PDF knitted from Rmd to Canvas. Your solutions to the problems here should be implemented in your .Rmd file, and your git commit history should reflect the process you used to solve these Problems.

Link to repo: https://github.com/dliao1/bios731_hw2_liao

Algorithms for logistic regression

For a given subject in a study, we are interested in modeling $\pi_i = P(Y_i = 1|X_i = x_i)$, where $Y_i \in \{0, 1\}$. The logistic regression model takes the form

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \log\left(\frac{P(Y_i = 1|X_i)}{1 - P(Y_i = 1|X_i)}\right) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \ldots + \beta_p X_{pi}$$

- $Y_1, Y_2, \ldots, Y_n \sim Bernoulli(\pi)$
- PDF is $f(y_i; \pi) = \pi^{y_i}(1-\pi)^{1-y_i}$

Problem 1: Newton's method

- Derive likelihood, gradient, and Hessian for logistic regression for an arbitrary number of predictors p

Let $\pi_i = P(Y_i = 1 \mid X_i = x_i), Y_i \in \{0, 1\}$. The logistic regression model is given by:

$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \log\left(\frac{P(Y_i=1|X_i)}{1-P(Y_i=1|X_i)}\right) = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} = X_i^T\beta$

where β is a $p \times 1$ vector.

The response variables follow a Bernoulli distribution: $Y_1, Y_2, \ldots, Y_n \sim \text{Bern}(\pi)$, with probability mass function:

$f(y_i \mid \pi_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}$

The expectation of Y_i given X_i is $E[Y_i \mid X_i] = \pi_i$. Since $\log\left(\frac{E[Y_i|X_i]}{1-E[Y_i|X_i]}\right) = X_i^T\beta$, we exponentiate both sides:

$e^{X_i^T\beta} = \frac{E[Y_i|X_i]}{1-E[Y_i|X_i]}$

$e^{X_i^T\beta} - E[Y_i \mid X_i]e^{X_i^T\beta} = E[Y_i \mid X_i]$

$e^{X_i^T\beta} = E[Y_i \mid X_i] + E[Y_i \mid X_i]e^{X_i^T\beta}$

$e^{X_i^T\beta} = E[Y_i \mid X_i](1 + e^{X_i^T\beta})$

$\frac{e^{X_i^T\beta}}{1+e^{X_i^T\beta}} = E[Y_i \mid X_i] = \pi_i$

Likelihood The likelihood function is:

$L(\beta) = \prod_{i=1}^n \pi_i^{y_i}(1-\pi_i)^{1-y_i}$

Substituting $\pi_i = \frac{e^{X_i^T\beta}}{1+e^{X_i^T\beta}}$

$$L(\beta) = \prod_{i=1}^{n}\left(\frac{e^{X_i^T\beta}}{1+e^{X_i^T\beta}}\right)^{y_i}\left(\frac{1}{1+e^{X_i^T\beta}}\right)^{1-y_i}$$

The log-likelihood function is given by:

$\ell(\beta) = \log L(\beta) = \log\prod_{i=1}^n\left(\frac{e^{X_i^T\beta}}{1+e^{X_i^T\beta}}\right)^{y_i}\left(1 - \frac{e^{X_i^T\beta}}{1+e^{X_i^T\beta}}\right)^{1-y_i}$

$\ell(\beta) = \sum_{i=1}^n\left[y_i\log\left(\frac{e^{X_i^T\beta}}{1+e^{X_i^T\beta}}\right) + (1-y_i)\log\left(1 - \frac{e^{X_i^T\beta}}{1+e^{X_i^T\beta}}\right)\right]$

$\sum_{i=1}^{n} \left[y_i(\log e^{X_i^T \beta} - \log(1 + e^{X_i^T \beta})) + (1 - y_i) \left(\log \frac{1}{1+e^{X_i^T \beta}} \right) \right]$

$\sum_{i=1}^{n} \left[y_i(X_i^T \beta - \log(1 + e^{X_i^T \beta})) + (1 - y_i)(\log 1 - \log(1 + e^{X_i^T \beta})) \right]$

$\sum_{i=1}^{n} \left[y_i X_i^T \beta - \log(1 + e^{X_i^T \beta}) \right]$

Thus, the final form of the log-likelihood function is:

$\ell(\beta) = \sum_{i=1}^{n} y_i X_i^T \beta - \sum_{i=1}^{n} \log(1 + e^{X_i^T \beta})$

Gradient $\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{n} y_i X_i^T - \sum_{i=1}^{n} \frac{X_i e^{X_i^T \beta}}{1+e^{X_i^T \beta}}$

Factoring common terms out: $\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{n} \left(y_i - \frac{e^{X_i^T \beta}}{1+e^{X_i^T \beta}} \right) X_i$

Rewriting in terms of π_i we get:

$\frac{\partial \ell}{\partial \beta} = \sum_{i=1}^{n} (y_i - \pi_i) X_i$

Hessian $\frac{\partial^2 \ell}{\partial \beta^2} = -\sum_{i=1}^{n} X_i^T X_i \frac{e^{X_i^T \beta}(1+e^{X_i^T \beta}) - e^{2X_i^T \beta}}{(1+e^{X_i^T \beta})^2}$

Simplifying: $\frac{\partial^2 \ell}{\partial \beta^2} = -\sum_{i=1}^{n} \frac{X_i^T X_i e^{X_i^T \beta}[(1+e^{X_i^T \beta}) - e^{X_i^T \beta}]}{(1+e^{X_i^T \beta})^2}$

$\frac{\partial^2 \ell}{\partial \beta^2} = -\sum_{i=1}^{n} X_i^T X_i \frac{e^{X_i^T \beta}}{(1+e^{X_i^T \beta})^2}$

Since:

$\pi_i(1 - \pi_i) = \frac{e^{X_i^T \beta}}{(1+e^{X_i^T \beta})^2}$

$\frac{\partial^2 \ell}{\partial \beta^2} = -\sum_{i=1}^{n} X_i^T (\pi_i(1 - \pi_i)) X_i$

- What is the Newton's method update for β for logistic regression?

$\beta_{t+1} = \beta_t - \left(\frac{\partial^2 \ell}{\partial \beta^2} \right)^{-1} \frac{\partial \ell}{\partial \beta}$

$\beta_{t+1} = \beta_t - \left(\frac{\sum_{i=1}^{n} (y_i - \pi_i) X_i}{-\sum_{i=1}^{n} X_i^T (\pi_i(1-\pi_i)) X_i} \right)$

where $\pi_i = \frac{e^{X_i^T \beta}}{1+e^{X_i^T \beta}}$

- Is logistic regression a convex optimization problem? Why or why not?

Gradient of -log f is the negative Hessian:

$-\frac{\partial^2 \ell}{\partial \beta^2} = +\sum_{i=1}^{n} X_i^T (\pi_i(1 - \pi_i)) X_i \geq 0$

The negative hessian is always positive, thus this is a convex optimization problem.

Problem 2: MM

(A) In constructing a minorizing function, first prove the inequality

$$-\log\{1+\exp x_i^T\theta\} \geq -\log\{1+\exp(X_i^T\theta^{(k)})\} - \frac{\exp(X_i^T\theta) - \exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}$$

with equality when $\theta = \theta^{(k)}$. This eliminates the log terms.

$$\text{Prove} \quad -\log\left(1+e^{x_i^T\theta}\right) \geq -\log\left(1+e^{x_i^T\theta^{(k)}}\right) - \frac{e^{x_i^T\theta^{(k)}} - e^{x_i^T\theta}}{1+e^{x_i^T\theta^{(k)}}}$$

Supporting hyperplane property:

$$f(y) \geq f(x) + \nabla f(x)^T(y-x), \quad \forall x, y \in \mathbb{R}^p$$

$$-\log(y) \geq -\log(x) - \frac{1}{x}(y-x)$$

We can substitute our expressions for x and y:

$$y = 1 + e^{x_i^T\theta}$$

$$x = 1 + e^{x_i^T\theta^{(k)}}$$

$$-\log\left(1+e^{x_i^T\theta}\right) \geq -\log\left(1+e^{x_i^T\theta^{(k)}}\right) - \frac{(1+e^{x_i^T\theta^{(k)}}) - (1+e^{x_i^T\theta})}{1+e^{x_i^T\theta^{(k)}}}$$

$$-\log\left(1+e^{x_i^T\theta}\right) \geq -\log\left(1+e^{x_i^T\theta^{(k)}}\right) - \frac{e^{x_i^T\theta^{(k)}} - e^{x_i^T\theta}}{1+e^{x_i^T\theta^{(k)}}}$$

(B) Now apply the arithmetic-geometric mean inequality to the exponential function $\exp(X_i^T\theta)$ to separate the parameters. Assuming that θ has p components and that there are n observations, show that these maneuvers lead to a minorizing function

$$g(\theta|\theta^{(k)}) = -\frac{1}{p}\sum_{i=1}^{n}\frac{\exp(X_i^T\theta^{(k)})}{1+\exp(X_i^T\theta^{(k)})}\sum_{j=1}^{p}\exp\{pX_{ij}(\theta_j - \theta_j^{(k)})\} + \sum_{i=1}^{n}Y_iX_i^T\theta = 0$$

up to a constant that does not depend on θ.

(C) Finally, prove that maximizing $g(\theta|\theta^{(k)})$ consists of solving the equation

$$-\sum_{i=1}^{n} \frac{\exp(X_i^T\theta^{(k)})X_{ij}\exp(-pX_{ij}\theta_j^{(k)})}{1+\exp(X_i^T\theta^{(k)})}\exp(pX_{ij}\theta_j)+\sum_{i=1}^{n} Y_iX_{ij}=0$$

To maximize, take derivative w.r.t. θ_j and set to 0:

$$\frac{d}{d\theta_j}g=-\frac{1}{p}\sum_{i=1}^{n}\frac{e^{x_i^T\theta^k}}{1+e^{x_i^T\theta^k}}px_{ij}e^{px_{ij}(\theta_j-\theta_j^k)}+\sum_{i=1}^{n}y_ix_{ij}=0$$

Rewriting:

$$-\sum_{i=1}^{n}\frac{e^{x_i^T\theta^k}}{1+e^{x_i^T\theta^k}}x_{ij}e^{px_{ij}(\theta_j-\theta_j^k)}+\sum_{i=1}^{n}y_ix_{ij}=0$$

Simplifying further:

$$-\sum_{i=1}^{n}\frac{x_{ij}e^{x_i^T\theta^k}}{1+e^{x_i^T\theta^k}}e^{px_{ij}\theta_j}e^{-px_{ij}\theta_j^k}+\sum_{i=1}^{n}y_ix_{ij}=0$$

Problem 3: simulation

Next we will implement logistic regression in R four different ways and compare the results using a short simulation study.

- implement using Newton's method from 1.1 in R
- implement using MM from 1.2 in R
- implement using `glm()` in R
- implement using `optim()` in R:
  - Use the option `method = "BFGS"`, which implements a Quasi-Newton approach

Simulation study specification:

- simulate from the model $logit(P(Y_i=1|X_i))=\beta_0+\beta_1X_i$
  - $\beta_0=1$
  - $\beta_1=0.3$
  - $X_i\sim N(0,1)$
  - $n=200$
  - $nsim=1$
- For your implementation of MM and Newton's method, select your own starting value and stopping criterion, but make sure they are the same for the two algorithms

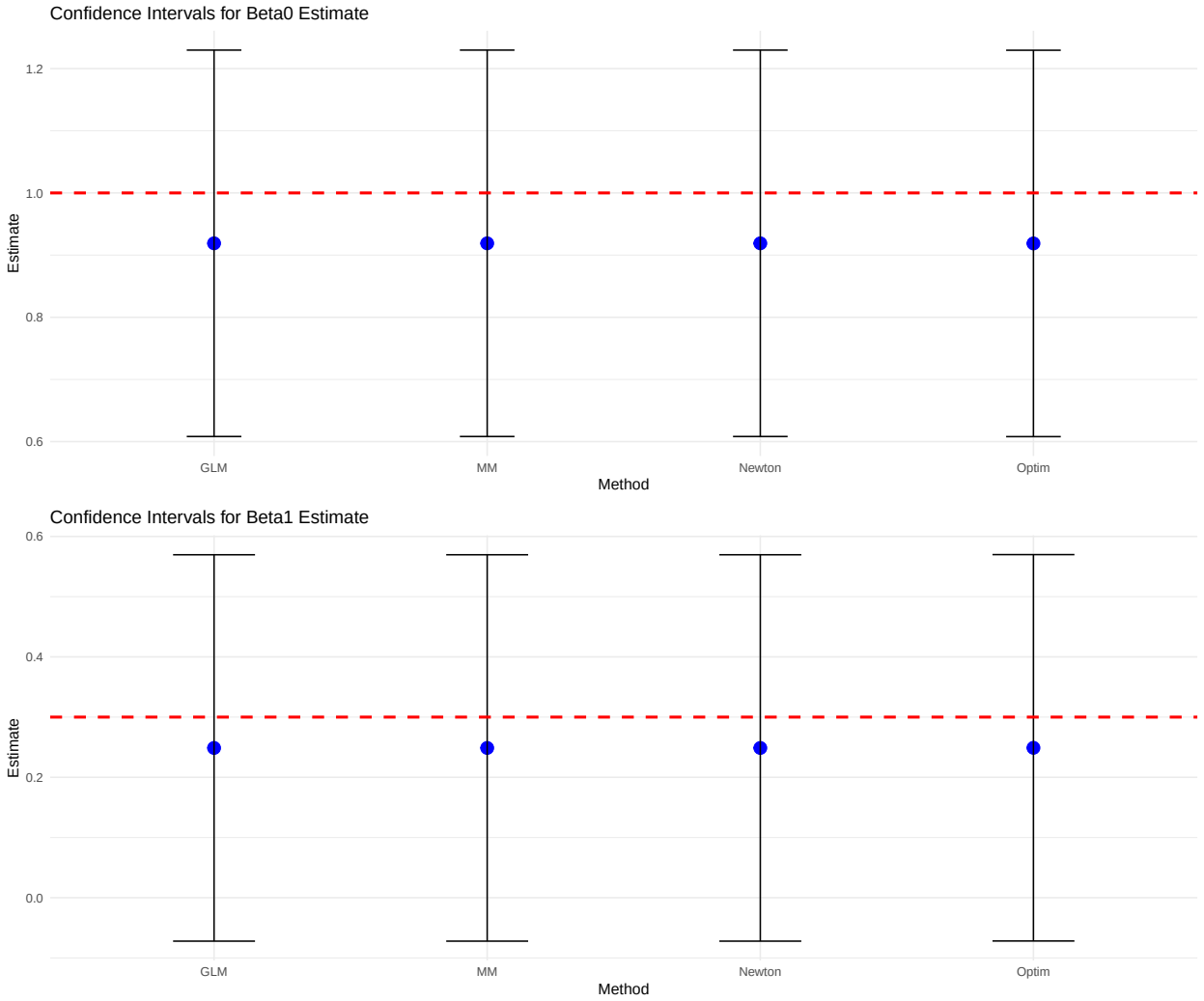
You only need to run the simulation using **one simulated dataset**. For each of the four methods, report:

- $\hat{\beta}_0$, $\hat{\beta}_1$
- 95% confidence intervals for $\hat{\beta}_0$, $\hat{\beta}_1$
- computation time
- number of iterations to convergence

Method	Beta0	Beta1	CI_Beta0_Lower	CI_Beta0_Upper	CI_Beta1_Lower	CI_Beta1_Upper	Time	Converged
Newton	0.9190	0.2487	0.6082	1.2297	-0.0721	0.5696	0.00	7
Optim	0.9188	0.2490	0.6080	1.2295	-0.0719	0.5698	0.01	7
GLM	0.9190	0.2487	0.6082	1.2297	-0.0721	0.5696	0.00	4
MM	0.9190	0.2487	0.6082	1.2297	-0.0721	0.5696	0.14	105

Make 2-3 plots or tables comparing your results, and summarize these findings in one paragraph.

```
##   Method     Beta0     Beta1 CI_Beta0_Lower CI_Beta0_Upper CI_Beta1_Lower
## 1 Newton 0.9189634 0.2487389      0.6082311       1.229696    -0.07212386
## 2  Optim 0.9187629 0.2489828      0.6080382       1.229488    -0.07187947
## 3    GLM 0.9189634 0.2487389      0.6082312       1.229696    -0.07212369
## 4     MM 0.9189674 0.2487415      0.6082347       1.229700    -0.07212170
##   CI_Beta1_Upper Time Converged
## 1      0.5696017 0.00         7
## 2      0.5698450 0.01         7
## 3      0.5696015 0.00         4
## 4      0.5696047 0.14       105
```



To summarize, all of the 4 methods (newton, mm, glm, and optim) did reasonably well and performed similarly at converging to the true beta0 of 1 and true beta1 of 0.3. All converged to around 0.9 for beta

0 and 0.24 for beta 1. However, it looks like the newton and mm methods implemented on average took a longer number of iterations to converge and the computation time was longer as well.

Problem 4: EM algorithm for censored exponential data

This will be a continuation of the lab problem on EM for censored exponential data. Suppose we have survival times $t_1, \ldots, t_n \sim Exponential(\lambda)$.

- Do not observe all survival times because some are censored at times $c_1, \ldots, c_n$.
- Actually observe $y_1, \ldots, y_n$, where $y_i = min(y_0, c_i)$
  - Also have an indicator δ_i where $\delta_i = 1$ is $y_i \leq c_i$
    - i.e. $\delta_i = 1$ if not censored and $\delta_i = 0$ if censored

Do the following:

- Derive an EM algorithm to estimate the parameter λ. Show your derivation here and report updates for the **E-step** and **M-Step**.

Derivation of EM algorithm

- Survival times $t_1, \ldots, t_n \sim \text{Exp}(\lambda)$.
- Times censored at $c_1, \ldots, c_n$.
- We observe y_i, where $y_i = \min(t_i, c_i)$.
- Indicator $\delta_i = 1$ if $t_i \leq c_i$ (not censored), $\delta_i = 0$ if $t_i > c_i$ (censored).

We need $p(y \mid z, \theta)$ the complete data density.

z represents the survival times of those who were censored (the missing data in this case)

The exponential density function is:

$$p(t \mid \lambda) = \frac{1}{\lambda} e^{-t/\lambda}$$

where $E(\lambda) = \lambda$.

Given that $t \sim \text{Exp}(\lambda)$:

$t_i = \delta_i y_i + (1 - \delta_i) z_i$

(We need the expected value of this for the E step)

and:

$$p(y, t \mid \lambda) = \frac{1}{\lambda} e^{-t/\lambda}$$

Thus, the complete data density is:

$$p(y, t \mid \theta) = \frac{1}{\lambda} e^{-(\delta_i y_i + (1-\delta_i) z_i)/\lambda}$$

Log-Likelihood Function

$$\log p(y, t \mid \theta) = \sum_{i=1}^{n} \log \lambda - \frac{1}{\lambda} \sum_{i=1}^{n} (\delta_i y_i + (1 - \delta_i) z_i)$$

$$-n \log \lambda - \frac{1}{\lambda} \sum_{i=1}^{n} (\delta_i y_i + (1 - \delta_i) z_i)$$

Q function:

$$Q(\lambda \mid \lambda_0) = E_z \left[-n \log \lambda - \frac{1}{\lambda} \sum_{i=1}^{n} (\delta_i y_i + (1 - \delta_i) z_i) \right]$$

Since y_i and δ_i are already observed:

$$-n \log \lambda - \frac{1}{\lambda} \sum_{i=1}^{n} \left(E_z[\delta_i y_i] + E_z[(1 - \delta_i) z_i] \right)$$

For censored observations ($\delta_i = 0$), we use the memorylessness property of exponential distribution to get:

$E[z_i \mid z_i > y_i, \lambda] = y_i + \frac{1}{\lambda}$

Thus:

$Q(\lambda \mid \lambda_0) = -n \log \lambda - \frac{1}{\lambda} \sum_{i=1}^{n} \left(\delta_i y_i + (1 - \delta_i)(y_i + \lambda) \right])$

Expanding:

$Q(\lambda \mid \lambda_0) = -n \log \lambda - \frac{1}{\lambda} \sum_{i=1}^{n} \left(\delta_i y_i + y_i + \lambda - \delta_i y_i - \delta_i \lambda) \right])$

Simplifying:

$Q(\lambda \mid \lambda_0) = -n \log \lambda - \frac{1}{\lambda} \sum_{i=1}^{n} \left(\delta_i y_i + y_i + \lambda - \delta_i y_i - \delta_i \lambda) \right])$

Factoring:

$$Q(\lambda \mid \lambda_0) = -n \log \lambda - \frac{1}{\lambda} \sum_{i=1}^{n} (y_i + (1 - \delta_i) \lambda)$$

$$-n \log \lambda - \frac{1}{\lambda} \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} (1 - \delta_i)$$

Taking the derivative of Q w.r.t λ and setting it to 0:

$$\frac{\partial \ell}{\partial \lambda} = \frac{-n}{\lambda} + \frac{\sum_{i=1}^{n} y_i}{\lambda^2}$$

$$\frac{\sum_{i=1}^{n} y_i}{\lambda^2} = \frac{n}{\lambda}$$

$$\sum_{i=1}^{n} y_i = n\lambda$$

M-step: $\frac{\sum_{i=1}^{n} y_i}{n} = \lambda$

- Implement your EM in R and fit it to the `veteran` dataset from the `survival` package.
  - Report your fitted λ value. How did you monitor convergence?
  - Report a 95% confidence interval for λ, and explain how it was obtained.
  - Compare 95% confidence interval and results from those obtained by fitting an accelerated failure time model (AFT) in R with exponential errors. You can fit an AFT model using the `survreg()` function from the `survival` package. If you choose `dist = "weibull` and `shape = 1` as parameter arguments, this will provide exponential errors.

Implementation

```
## Estimated lambda from EM: 130.1797

## 95% CI for lambda (EM): [ 103.4073 , 162.3346 ]

## Estimated lambda from phreg: 130.1797

## 95% CI for lambda (phreg): [ 109.4726 , 154.8035 ]
```

The fitted λ obtained using EM algorithm was 130.1797, with a CI of [103.4073 , 162.3346]. I monitored convergence by comparing the difference between the observed likelihood at the current iteration and the observed likelihood at the previous iteration - if the difference was $<$ 1e-12, the algorithm was determined to have "converged". The variance for the confidence interval was computed using bootstrap sampling.

Using the phreg command, the estimated lambda was 130.1797 with a CI of [109.4726 , 154.8035]. Using our EM algorithm, we converged to the same estimated lambda/mean survival time, but it looks like our EM confidence interval that we obtained via bootstrap percentiles was wider.

Extra credit (up to 10 points)! Expected vs. observed information

Part A: Show that the expected and observed information are equivalent for logistic regression Fisher/expected information:

$$I(\beta) = -E[H(\beta)]$$

Expanding,

$$I(\beta) = -E\left[\left(X_i^T(\pi_i(1-\pi_i))X_i\right)\right]$$

Since expectation does not depend on the observed data:

$$I(\beta) = X_i^T(\pi_i(1-\pi_i))X_i$$

Observed information:

$$I_n(\beta) = -H(\beta) = X_i^T X_i(\pi_i(1-\pi_i))$$

Thus, observed and expected information are the same for logistic regression.

Part B: Let's say you are instead performing probit regression, which is similar to logistic regression but with a different link function. Specifically, probit regression uses a probit link:

$$\Phi^{-1}(Pr[Y_i = 1|X_i]) = X_i^T\beta,$$

where Φ^{-1} is inverse of the CDF for the standard normal distribution. **Are the expected and observed information equivalent for probit regression?** Justify why or why not.

Probit regression

$$\Phi^{-1}\left(P(Y_i = 1 \mid X_i)\right) = X_i^T\beta$$
$$Pr[Y_i \mid X_i] - \Phi(X_i^T\beta)$$
$$Y_i \sim Bern(\pi_i)$$

$$Y_i \sim Bern(\pi_i)$$

$$f(y_i \mid \pi_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

CDF for Φ :

$$\Phi(X_i^T\beta) = \int_{-\infty}^{X_i^T\beta} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}\, dz$$

Log-Likelihood Function

The likelihood function is:

$$L(\beta) = \prod_{i=1}^{n} \pi_i^{y_i}(1-\pi_i)^{1-y_i}$$

Substituting $\pi_i = \Phi(X_i^T\beta)$:

$$L(\beta) = \prod_{i=1}^{n} \left(\Phi(X_i^T\beta)\right)^{y_i} \left(1 - \Phi(X_i^T\beta)\right)^{1-y_i}$$

Taking the log:

$$\ell(\beta) = \sum_{i=1}^{n} \left[y_i \log(\Phi(X_i^T\beta)) + (1-y_i)\log(1-\Phi(X_i^T\beta))\right]$$

Gradient Calculation

We start with the left half first:

$$\frac{\partial}{\partial\beta}\left[y_i \log(\Phi(X_i^T\beta))\right]$$

Chain rule:

$$\frac{\partial}{\partial\beta}[y_i \log(\Phi(X_i^T\beta))] = \left(\frac{y_i}{\Phi(X_i^T\beta)}\right) \cdot \frac{\partial}{\partial\beta}\left(\int_{-\infty}^{X_i^T\beta} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx\right)$$

Since the derivative of the a CDF is the PDF:

$$\frac{\partial}{\partial\beta}\Phi(X_i^T\beta) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}} X_i$$

$$\frac{y_i}{\Phi(X_i^T\beta)} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}} X_i$$

Now for the right term:

$$\frac{\partial}{\partial\beta}[(1-y_i)\log(1-\Phi(X_i^T\beta))]$$

Applying the chain rule:

$$(1-y_i) \cdot \left(-\frac{1}{1-\Phi(X_i^T\beta)}\right) \cdot \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}} X_i\right)$$

Full Gradient

$$\sum_{i=1}^{n} \left[\frac{y_i}{\Phi(X_i^T\beta)} \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}} X_i - \frac{1-y_i}{1-\Phi(X_i^T\beta)} \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}} X_i\right]$$

Factoring:

$$\sum_{i=1}^{n} X_i^T \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}} \left[\frac{y_i}{\Phi(X_i^T\beta)} - \frac{1-y_i}{1-\Phi(X_i^T\beta)}\right]$$

Hessian Calculation

Taking the second derivative using the product rule:

$$\frac{\partial^2\ell}{\partial\beta^2} = \sum_{i=1}^{n} \left(X_i^T \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}}\right) \left[\frac{\partial}{\partial\beta}\left(\frac{y_i}{\Phi(X_i^T\beta)} - \frac{1-y_i}{1-\Phi(X_i^T\beta)}\right)\right] + \left(\frac{y_i}{\Phi(X_i^T\beta)} - \frac{1-y_i}{1-\Phi(X_i^T\beta)}\right) \left[X_i^T \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}} X_i^T\beta X_i\right]$$

The first term in the Hessian calculation:

$$\left(X_i^T \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}}\right) \left[\frac{\partial}{\partial\beta}\left(\frac{y_i}{\Phi(X_i^T\beta)} - \frac{1-y_i}{1-\Phi(X_i^T\beta)}\right)\right]$$

Using the quotient rule:

$$\frac{\partial}{\partial\beta}\left(\frac{y_i}{\Phi(X_i^T\beta)}\right) = \frac{\Phi(X_i^T\beta)(0) - y_i\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}}\right)}{\left[\Phi(X_i^T\beta)\right]^2}$$

$$= -\frac{y_i}{\left[\Phi(X_i^T\beta)\right]^2} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}}$$

Similarly, for the second term:

$$\frac{\partial}{\partial\beta}\left(\frac{1-y_i}{1-\Phi(X_i^T\beta)}\right) = \frac{(1-\Phi(X_i^T\beta))(0) - (1-y_i)\left(-\frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}}\right)}{\left(1-\Phi(X_i^T\beta)\right)^2}$$

$$= \frac{(1-y_i)}{\left(1-\Phi(X_i^T\beta)\right)^2} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}}$$

Full Hessian

$$\sum_{i=1}^{n}\left[-y_i\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}}\right)\frac{1}{\left[\Phi(X_i^T\beta)\right]^2} + (1-y_i)\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}}\right)\frac{1}{(1-\Phi(X_i^T\beta))^2}\right]$$

$$+\left[\frac{y_i}{\Phi(X_i^T\beta)} - \frac{1-y_i}{1-\Phi(X_i^T\beta)}\right]\left(X_i^T\frac{1}{\sqrt{2\pi}} e^{-\frac{(X_i^T\beta)^2}{2}} - X_i^T\beta X_i\right)$$

We can see that the expected and observed information will be different because y_i is present in the Hessian, so it's dependent on our data.