

STAT GU4206/GR5206 Lab 8

Name and Uni

Note this is a practice lab!

Goal

The goal of Lab 8 is to estimate a logistic regression model via maximum likelihood. The main optimization procedure used is Newton's method, which is the most common (and amazing) form of the gradient decent algorithm. We also refine Newton's method and implement Iteratively Reweighted Least Squares, which is the most common method of estimating the logistic regression model. A description of the dataset and logistic statistical model follow below:

Data Description

A university medical center urology group was interested in the association between prostate-specific antigen (PSA) and a number of prognostic clinical measurements in men with advanced prostate cancer. Data were collected on 97 men who were about to undergo radical prostatectomies. The 8 variables are:

| Variable | Variable Name | Description |
|----------|------------------------------|---|
| X_1 | PSA level | Serum prostate-specific antigen level (mg/ml) |
| X_2 | Cancer volume | Estimate of prostate cancer volume (cc) |
| X_3 | Weight | Prostate weight (gm) |
| X_4 | Age | Age of patient (years) |
| X_5 | Benign prostatic hyperplasia | Amount of benign prostatic hyperplasia (cm2) |
| X_6 | Seminal vesicle invasion | Presence or absence of seminal vesicle invasion |
| X_7 | Capsular penetration | Degree of capsular penetration (cm) |
| Y | Gleason score | Pathologically determined grade of disease |

Below we read in the dataset and name it **prostate**.

```
prostate <- read.table("Lab8.txt")
head(prostate)
```

```
##      X1      X2      X3 X4 X5 X6 X7 Y
## 1 0.651 0.5599 15.959 50  0  0  0 6
## 2 0.852 0.3716 27.660 58  0  0  0 7
## 3 0.852 0.6005 14.732 74  0  0  0 7
## 4 0.852 0.3012 26.576 58  0  0  0 6
## 5 1.448 2.1170 30.877 62  0  0  0 6
## 6 2.160 0.3499 25.280 50  0  0  0 6
```

In our setting we create a new binary response variable Y , called high-grade cancer by letting $Y = 1$ if Gleason score equals 8, and $Y = 0$ otherwise (i.e., if Gleason score equals 6 or 7). The goal is to carry out a logistic regression analysis, where the response of interest is high-grade cancer (Y).

```
prostate$Y <- ifelse(prostate$Y==8,1,0)
head(prostate)
```

```
##      X1      X2      X3 X4 X5 X6 X7 Y
## 1 0.651 0.5599 15.959 50  0  0  0 0
## 2 0.852 0.3716 27.660 58  0  0  0 0
## 3 0.852 0.6005 14.732 74  0  0  0 0
## 4 0.852 0.3012 26.576 58  0  0  0 0
## 5 1.448 2.1170 30.877 62  0  0  0 0
## 6 2.160 0.3499 25.280 50  0  0  0 0
```

```
nrow(prostate)
```

```
## [1] 97
```

Logistic Model

Let Y_1, Y_2, \dots, Y_{97} be independent Bernoulli random variables with expected values

$$E[Y_i] = p_i = \frac{\exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \beta_5 X_{5i} + \beta_6 X_{6i} + \beta_7 X_{7i})}{1 + \exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \beta_5 X_{5i} + \beta_6 X_{6i} + \beta_7 X_{7i})}, \quad i = 1, 2, \dots, 97. \quad (1)$$

Part 1: Fit Logistic Model Using glm()

Problem 1)

Use the base **R** function (**glm()**) fit the logistic model. Run the code below:

```
model <- glm(Y~X1+X2+X3+X4+X5+X6+X7,
             data=prostate,
             family=binomial(link = "logit")
            )
model
```

```
##
## Call:  glm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7, family = binomial(link = "logit"),
##       data = prostate)
##
## Coefficients:
## (Intercept)      X1      X2      X3      X4
## -10.276610    0.055228    0.086869    0.001615    0.106633
##      X5      X6      X7
## -0.071827  -1.154137    0.123324
##
## Degrees of Freedom: 96 Total (i.e. Null);  89 Residual
## Null Deviance:      101.4
## Residual Deviance: 61.57    AIC: 77.57
```

Problem 2)

Consider a respondent with the following characteristics:

| Variable | Variable Name | Actual value |
|----------|------------------------------|--------------|
| X_1 | PSA level | 21.3 (mg/ml) |
| X_2 | Cancer volume | 8.4 (cc) |
| X_3 | Weight | 48.4 (gm) |
| X_4 | Age | 68 (years) |
| X_5 | Benign prostatic hyperplasia | 4.7 (cm2) |
| X_6 | Seminal vesicle invasion | 0 |
| X_7 | Capsular penetration | 3.2 (cm) |

Estimate the probability that this respondent is among the high-grade cancer group. Based on this estimated probability, do you believe that a respondent with these characteristics belongs to the high-grade cancer group? The **R** code is explicitly given below.

```
new.case <- data.frame(X1=21.3,X2=8.4,X3=48.4,X4=68,X5=4.7,X6=0,X7=3.2)
linear.prediction <- predict(model,newdata = new.case)
exp(linear.prediction)/(1+exp(linear.prediction))
```

```
##          1
## 0.2720329
```

Part 1: Maximum Likelihood Estimation and Newton's Method

In Model (1), the response values represent high-grade cancer and the features X_1, X_2, \dots, X_7 are outlined in the data description from earlier in this document. To estimate Model (1), we use the method of *Maximum Likelihood*. The objective function of interest (log-likelihood) is

$$\begin{aligned} \ell(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7) = & \sum_{i=1}^n y_i (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \beta_5 X_{5i} + \beta_6 X_{6i} + \beta_7 X_{7i}) \\ & - \sum_{i=1}^n \log(1 + \exp(\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i} + \beta_5 X_{5i} + \beta_6 X_{6i} + \beta_7 X_{7i})) \end{aligned}$$

The above log-likelihood is the same function derived in class except we have several more parameters to estimate, i.e., $\beta_0, \beta_1, \dots, \beta_7$. In class we only considered *simple logistic regression* (one feature).

Problem 3)

Create a function in **R** called **logistic.NLL** with inputs **beta** and **data**, where **beta** is a vector of β coefficients and **data** is a dataframe defaulted by **data=prostate**. The function **logistic.NLL** should output the negative of the log-likelihood, i.e.,

$$-\ell(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7).$$

Recall that maximizing the log-likelihood is equivalent to minimizing the negative log-likelihood. Also evaluate the function using the vector **beta=rep(0,8)**, i.e., run the code **logistic.NLL(beta=rep(0,8))**.

```
## Solution goes here
logistic.NLL <- function(beta,data=prostate) {

  beta_0 <- beta[1]
  beta_1_to_7 <- beta[2:8]
  Y <- data$Y
  X.mat <- as.matrix(data[,1:7])
  linear.component <- beta_0 + X.mat%%beta_1_to_7
  p.i <- exp(linear.component)/(1+exp(linear.component))
  return(-sum(dbinom(Y,size=1,prob=p.i,log=TRUE)))
}
logistic.NLL(beta=rep(0,8),data=prostate)
```

```
## [1] 67.23528
```

Problem 4)

Write a **R** function called **Newtons.Method** that performs *Newton's Optimization Method* on a **generic function f**. The function should have inputs **f**, **x0**, **max.iter**, **stopping.deriv** and **...**. The input **f** is the generic function we wish to **minimize**, **x0** is the starting point in the Newton's Method algorithm, **max.iter** is the maximum number of iterations (defaulted at 200), **stopping.deriv** is the gradient's threshold at which the algorithm terminates (defaulted at 0.001) and **...** allows you to pass additional arguments, based on **f**, into the **Newtons.Method** function. The output of **Newtons.Method** should be a list giving all updates of our minimizer and the number of iterations required to perform the procedure. You are welcome to add additional outputs if you would like. Hint: just copy and paste the **grad.descent** function from class and change the update step.

```
## Solution goes here
library(numDeriv)

#Newton.method <- function(f, x0, max.iter = 200, stopping.deriv = 0.001, ...) {

  ## Body of loop
#}
```

Problem 5)

Run the function **Newtons.Method** to minimize the function **logistic.NLL** using initial value **x0=rep(0,8)**, maximum iterations **max.iter=200** and **stopping.deriv=.001**. Display the estimated parameters for $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7$. How many iterations did the algorithm take to converge?

```
## Solution goes here
#logistic.est <- Newton.method(logistic.NLL,
#                               x0=rep(0,8),
#                               max.iter = 200,
#                               stopping.deriv = 0.001,
#                               data=prostate)

#logistic.est$x
#logistic.est$k
```

Problem 6)

Use the base **R** function `nlm()` to minimize the negative log-likelihood using initial value `x0=rep(0,8)`. Display the estimated parameters for $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7$. How many iterations did the algorithm take to converge?

```
## Solution goes here
```

Part 2: Iteratively Reweighted Least Squares

Problem 7)

Write a **R** function called `IRLS.logistic` that performs the *Iteratively Reweighted Least Squares* algorithm applied to a generic logistic regression model. The function should have inputs **Y**, **X**, **max.iter** and **threshold**. The input **Y** is the response vector of Bernoulli values, **X** is the data matrix of features, **max.iter** is the maximum number of iterations (defaulted at 20), **threshold** is the threshold at which the algorithm terminates (defaulted at 0.001). Define the threshold $\delta > 0$ in terms of how close β_t is to β_{t+1} , i.e.,

$$\|\beta_{t+1} - \beta_t\| < \delta.$$

There are several choices for the above norm. I recommend breaking the loop if all estimated coefficients are less than δ , i.e., $|\beta_{i(t+1)} - \beta_{i(t)}| < \delta$ for $i = 0, 1, \dots, p$. Let the starting value be defined inside the function by $\beta_{(0)} = [\beta_0^*, 0, 0, \dots, 0]^T$, where

$$\beta_0^* = \log(\bar{Y}/(1 - \bar{Y})).$$

```
## Solution goes here
#IRLS.logistic <- function(Y,X,max.iter=20,threshold=.001) {

  ## Body of loop
#}
```

Problem 8)

Run the function `IRLS.logistic` to estimate our logistic model using **Y=response**, **X=design.matrix**, **max.iter=20**, and **threshold=.00001**. Display the estimated parameters for $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7$. How many iterations did the algorithm take to converge?

```
# Define data matrix X and response vector Y
response <- prostate$Y
design.matrix <- prostate[,1:7]

## Solution goes here
#logistic.IRLS <- IRLS.logistic(Y=response,X=design.matrix,
#                               max.iter=20,threshold=.00001)

#logistic.IRLS$betas
#logistic.IRLS$k
```

Problem 9)

Check that your parameter estimates from all methods are reasonably close to the estimates coming from the `glm()` function.

```
## Solution goes here
```