3333 Assignment 2

Ravish Kamath: 213893664

22 February, 2022

```
## Loaded lars 1.2
```

Question 1

In maximum likelihood estimation, the estimator is obtained by maximizing the log likelihood function. However, most of the log likelihood has to be optimized by Newton-Raphson algorithm. In this question, we will learn to program Newton-Raphson algorithm for a univariate function. Consider the function $f(\theta) = \frac{3(\theta)^2-1}{1+(\theta)^2}$, $\theta > 0$. Use the Newton-Raphson method to find the maximizer of the function. Implement the algorithms in R and run your code to obtain the maximizer. (Attach your screenshots of the output in R). You can use the online symbolic differentiation calculator to obtain $f'(\theta)$ and $f''(\theta)$ (https://www.symbolab.com/solver/second-derivative-calculator).

Solution

let θ be represented by x

$$f(x) = \frac{3x^2 - 1}{1 + x^3}$$

Then we have our first derivative to be

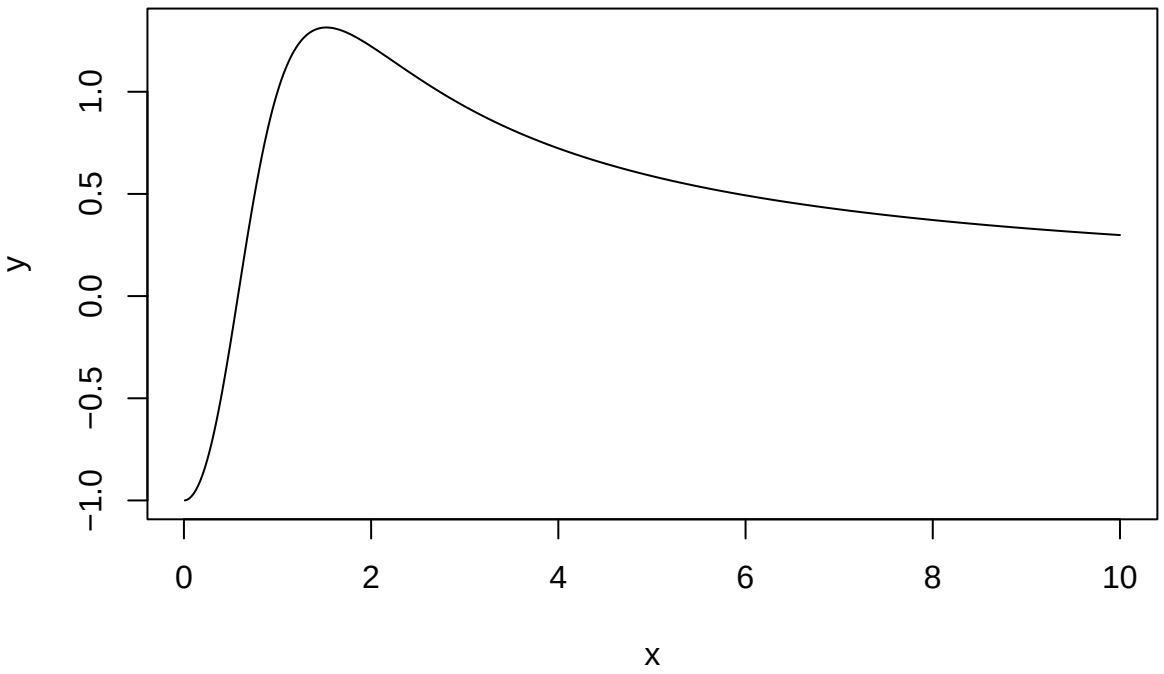
$$\begin{aligned} f'(x) &= \frac{(6x)(1 + x^3) - (3x^2)(3x^2 - 1)}{(1 + x^3)^2} \\ &= \frac{6x + 6x^4 - 9x^4 + 3x^2}{(1 + x^3)^2} \\ &= \frac{-3x^4 + 3x^2 + 6x}{(1 + x^3)^2} \end{aligned}$$

We have our second derivative to be

$$\begin{aligned} f''(x) &= \frac{(-120x^3 + 6x + 6)(1 + x^3)^2 - (6x^2)(1 + x^3)(-3x^4 + 3x^2 + 6x)}{(1 + x^3)^4} \\ &= \frac{(-120x^3 + 6x + 6)(1 + 2x^3 + x^6) - (6x^2)(1 + x^3)(-3x^4 + 3x^2 + 6x)}{(1 + x^3)^4} \\ &= \frac{(x^3 + 1) \times 6(x^6 - 2x^4 - 7x^3 + x + 1)}{(1 + x^3)^4} \\ &= \frac{6(x^6 - 2x^4 - 7x^3 + x + 1)}{(1 + x^3)^3} \end{aligned}$$

Let us now implement the algorithm in R.

```
x<-(1:1000)/100
y<-numeric(1000)
for (i in 1:1000){
  y[i] = (3*(x[i])^2 - 1) / (1 + (x[i])^3)
}
plot(x,y,type="l")
```




```
fp = function(x){
  p = (-3*(x^4) + 3*(x^2) + 6*(x)) / (1+x^3)^2
  return(p)
}
fpp<-function(x){
  p = 6*(x^6 - 2*(x^4) - 7*(x^3)+ x + 1) / (1 + x^3)^3
  return(p)
}
diff<-4
iter<-0

x = 1
while ((diff>0.001) && (iter<30)) {
  oldx<-x
  x<-x-fp(x)/fpp(x)
  diff<-abs(x-oldx)
  iter<-iter+1
  print(c(iter,diff))
}
```

```
## [1] 1.0000000 0.3333333
## [1] 2.0000000 0.1446166
## [1] 3.00000000 0.04053567
## [1] 4.000000000 0.002880377
## [1] 5.000000e+00 1.372364e-05
```

Question 2

In the following marketing set, we have 9 years with the sales in 10 million euro and the advertising expenditure in million euro.

a) Based on the 9 observation and perform a ridge regression. Program it with R. Output the ridge regression results at a few different values of λ.
b) In your ridge regression, when λ increases, what do you observe from the values of the estimated coefficients. Does any of the estimated coefficients shrink to zero like the L_1 LASSO regression? Describe the difference between the output of a ridge regression and the output of a lasso regression.

Solution

Part A

Let us first recreate our data set

```
year = seq(1,9,1)
sales = c(61, 73, 85, 106, 120, 129, 142, 144, 161)
advertisement = c(19, 26, 30, 34, 43, 48, 52, 57, 68)
df = data.frame(year, sales, advertisement)
df
```

```
##   year sales advertisement
## 1    1    61            19
## 2    2    73            26
## 3    3    85            30
## 4    4   106            34
## 5    5   120            43
## 6    6   129            48
## 7    7   142            52
## 8    8   144            57
## 9    9   161            68
```

```
ones = c(rep(1,9))
X = cbind(ones, advertisement)
XtY = t(X)%*%sales
XtX = t(X)%*%X
```

We will set $\lambda = 0.1$ in our first iteration of ridge regression.

```
lambda = 0.1*diag(2)
XtXinv = solve(XtX + lambda)
ridge_theta = XtXinv%*%XtY
ridge_theta
```

```
##                    [,1]
## ones          21.910647
## advertisement  2.179345
```

As we can see, we get our estimated coefficients to be 21.91 and 2.18.

Let us now try to output our regression results with different values of λ. Here we have $\lambda = 0.01$

```
lambda = 0.01*diag(2)
XtXinv = solve(XtX + lambda)
ridge_theta = XtXinv%*%XtY
ridge_theta
```

```
##                    [,1]
## ones           23.81093
## advertisement   2.13916
```

$\lambda = 10$

```
lambda = 10*diag(2)
XtXinv = solve(XtX + lambda)
ridge_theta = XtXinv%*%XtY
ridge_theta
```

```
##                    [,1]
## ones           2.286053
## advertisement 2.593011
```

$\lambda = 100$

```
lambda = 100*diag(2)
XtXinv = solve(XtX + lambda)
ridge_theta = XtXinv%*%XtY
ridge_theta
```

```
##                     [,1]
## ones           0.2989559
## advertisement 2.6217873
```

$\lambda = 1000$

```
lambda = 1000*diag(2)
XtXinv = solve(XtX + lambda)
ridge_theta = XtXinv%*%XtY
ridge_theta
```

```
##                      [,1]
## ones           0.07747642
## advertisement 2.50086550
```

Part B

As shown above, when we increase our λ, we tend to see the estimated coefficient θ_0 tends to 0 while θ_1 tends to increases, although we saw a drop in the regression coefficient θ_1 when lambda was 1000. Let us try to use the lasso regression for this data set.

```
lasso <-lars(x=as.matrix(df[,3]),y=as.numeric(unlist(df[,2])),trace=TRUE)
```

```
## LASSO sequence
## Computing X'X .....
## LARS Step 1 :     Variable 1     added
## Computing residuals, RSS etc .....
```

```
coef(lasso,s=0.3,mode="fraction")
```

```
## [1] 0.6402779
```

As we can see the ridge regression estimated coefficient does not go to zero unlike the LASSO regression. The difference between the ridge and Lasso method is that, the θ_1 coefficient for ridge becomes bigger while for the lasso, it is 0.

Question 3

a) In this question, we will investigate the problem of a multiple testing. Consider the hypothesis testing of $H_0 : \mu = 0$, vs. $H_a : \mu \neq 0$. Under the null hypothesis, the Z test statistic is a standard normal random variable. We reject the null hypothesis when $|Z|$ is greater than 1.96 at the significance level of 0.05. Write a R program to simulate 1000 Z test statistic from standard normal $N(0, 1)$. (you can use the sample codes in our lecture notes).

b) If we perform hypothesis testing using the significance level of 0.05. Among the 1000 test statistics you generated, how many of them are rejected?

c) Apply the Bonferroni method to control the overall type I error rate to be 0.05. Based on your program, how many rejections do you obtain from your simulation?

d) As the Zs are generated from the null hypothesis, we consider these rejections are all false positive discoveries. Please use a short paragraph to summarize the problem we are faing when we perform multiple testings.

Solution

Part A

```
discovery<-0
for (i in 1:1000){
  x<-rnorm(30,0,1)
  ztest<-mean(x)/sqrt(var(x)/30)
  discovery<-discovery+(abs(ztest)>=1.96)
}
discovery
```

```
## [1] 71
```

Part B

```
1000*(0.05)
```

```
## [1] 50
```

We will have 50 test statistics to be rejected.

Part C

```
discovery<-0
for (i in 1:1000){
  x<-rnorm(30,0,1)
  ztest<-mean(x)/sqrt(var(x)/30)
  discovery<-discovery+(abs(ztest)>=abs(qnorm(0.05/1000*2)))
}
discovery
```

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

Part D

When we are using the Bonferroni method for multiple testing, when we use a large size test statistic, our rejection cutoff is really small, which means that its too conservative. This can remove the detection of some true positive statistics. Essentially the threshold is too stringent.

Question 4

Consider a data set with the response vector $Y = (y_1, ..., y_n)^T$ and the data matrix

$$\begin{pmatrix} 1 & x_{1,1} & x_{1,2} \\ \vdots & \vdots & \vdots \\ 1 & x_{n,2} & x_{n,2} \end{pmatrix}$$

We model the relationship between X and Y using the linear regression model: $y_i = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \epsilon_i, i = 1, ..., n$, where $\epsilon \sim N(0, \sigma^2)$. Let the parameter vector be denoted as $\theta = (\theta_0, \theta_1, \theta_2)^T$. We wish to minimize the sum of weighted squared residuals: $SSE = \sum_{i=1}^{n} w_i(y_i - (\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2}))^2$. Derive the formula for the solution of θ which minimizies the weighted sum of squared errors.

Solution

$$\begin{aligned}
SSE &= \sum_{i=1}^{n} w_i(y_i - (\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2}))^2 \\
&= \sum_{i=1}^{n} w_i(y_i - \hat{y_i})^2 \\
&= (Y - \hat{Y})^T W (Y - \hat{Y}) \\
&= (Y - X\theta)^T W (Y - X\theta) \\
&= Y^T WY - (X\theta)^T WY - Y^T W(X\theta) + (X\theta)^T W(X\theta) \\
&= Y^T WY - 2Y^T WX\theta + \theta^T X^T WX\theta \\
\frac{\partial SSE}{\partial \theta} &= -2X^T WY - 2X^T WX\theta \\
0 &= -2X^T WY - 2X^T WX\theta \\
X^T WY &= X^T WX\theta \\
\hat{\theta} &= (X^T WX)^{-1} X^T WY
\end{aligned}$$

Question 5

Analyze the German data set from the site: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

a) Perform the logistic regression on the data set. Build a predictive model using some of the predictors in the model. Please use 900 observations as the training set and use your model to predict the default status of the remaining 100 loans. Choose one of the regression coefficients and interpret the regression coefficient. What is the cutoff value of the probability do you use for your analysis? How many default ones are predicted to be non-default ones (number of false negative)? How many non-default ones are predicted to be default ones (number of false positive). Then you need to improve your model by adding more predictors or adding some higher order terms or interaction terms. Please demonstrate that your new model has lesser errors than your first model in the 100 testing cases.

b) Please investigate how the sensitivity and specificity change with respect to the different cutoff value of probability.

Solution

Part A

Let us first factor our non-numeric variables into dummy variables

```
index = c(1,2,4,5,7,8,10,11,13,15,16,18,20,21)
for( i in index){
  credit[,i] = factor(credit[,i])
}
```

Now we will perform logistic regression on the full model.

```
logist_fit = glm(Default~., data=credit,family=binomial(link=logit))
summary(logist_fit)
```

```
##
## Call:
## glm(formula = Default ~ ., family = binomial(link = logit), data = credit)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3410  -0.6994  -0.3752   0.7095   2.6116
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          4.005e-01  1.084e+00   0.369 0.711869
## checkingstatus1A12 -3.749e-01  2.179e-01  -1.720 0.085400 .
## checkingstatus1A13 -9.657e-01  3.692e-01  -2.616 0.008905 **
## checkingstatus1A14 -1.712e+00  2.322e-01  -7.373 1.66e-13 ***
## duration            2.786e-02  9.296e-03   2.997 0.002724 **
## historyA31          1.434e-01  5.489e-01   0.261 0.793921
## historyA32         -5.861e-01  4.305e-01  -1.362 0.173348
## historyA33         -8.532e-01  4.717e-01  -1.809 0.070470 .
## historyA34         -1.436e+00  4.399e-01  -3.264 0.001099 **
## purposeA41         -1.666e+00  3.743e-01  -4.452 8.51e-06 ***
## purposeA410        -1.489e+00  7.764e-01  -1.918 0.055163 .
## purposeA42         -7.916e-01  2.610e-01  -3.033 0.002421 **
## purposeA43         -8.916e-01  2.471e-01  -3.609 0.000308 ***
## purposeA44         -5.228e-01  7.623e-01  -0.686 0.492831
```

```
## purposeA45         -2.164e-01  5.500e-01  -0.393 0.694000
## purposeA46          3.628e-02  3.965e-01   0.092 0.927082
## purposeA48         -2.059e+00  1.212e+00  -1.699 0.089297 .
## purposeA49         -7.401e-01  3.339e-01  -2.216 0.026668 *
## amount              1.283e-04  4.444e-05   2.887 0.003894 **
## savingsA62         -3.577e-01  2.861e-01  -1.250 0.211130
## savingsA63         -3.761e-01  4.011e-01  -0.938 0.348476
## savingsA64         -1.339e+00  5.249e-01  -2.551 0.010729 *
## savingsA65         -9.467e-01  2.625e-01  -3.607 0.000310 ***
## employA72          -6.691e-02  4.270e-01  -0.157 0.875475
## employA73          -1.828e-01  4.105e-01  -0.445 0.656049
## employA74          -8.310e-01  4.455e-01  -1.866 0.062110 .
## employA75          -2.766e-01  4.134e-01  -0.669 0.503410
## installment         3.301e-01  8.828e-02   3.739 0.000185 ***
## statusA92          -2.755e-01  3.865e-01  -0.713 0.476040
## statusA93          -8.161e-01  3.799e-01  -2.148 0.031718 *
## statusA94          -3.671e-01  4.537e-01  -0.809 0.418448
## othersA102          4.360e-01  4.101e-01   1.063 0.287700
## othersA103         -9.786e-01  4.243e-01  -2.307 0.021072 *
## residence           4.776e-03  8.641e-02   0.055 0.955920
## propertyA122        2.814e-01  2.534e-01   1.111 0.266630
## propertyA123        1.945e-01  2.360e-01   0.824 0.409743
## propertyA124        7.304e-01  4.245e-01   1.721 0.085308 .
## age                -1.454e-02  9.222e-03  -1.576 0.114982
## otherplansA142     -1.232e-01  4.119e-01  -0.299 0.764878
## otherplansA143     -6.463e-01  2.391e-01  -2.703 0.006871 **
## housingA152        -4.436e-01  2.347e-01  -1.890 0.058715 .
## housingA153        -6.839e-01  4.770e-01  -1.434 0.151657
## cards               2.721e-01  1.895e-01   1.436 0.151109
## jobA172             5.361e-01  6.796e-01   0.789 0.430160
## jobA173             5.547e-01  6.549e-01   0.847 0.397015
## jobA174             4.795e-01  6.623e-01   0.724 0.469086
## liable              2.647e-01  2.492e-01   1.062 0.288249
## teleA192           -3.000e-01  2.013e-01  -1.491 0.136060
## foreignA202        -1.392e+00  6.258e-01  -2.225 0.026095 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.73  on 999  degrees of freedom
## Residual deviance:  895.82  on 951  degrees of freedom
## AIC: 993.82
##
## Number of Fisher Scoring iterations: 5
```

As we can see, we have quite a few predictors in our model.

Now we will try to compute a predictive model using the variables: duration, amount, installment, age, history, purpose, foreign and housing.

```
for (j in 1:1){
  training = sample(1:1000, 900)
  trainingset = credit[training,]
  testingset = credit[-training,]
  model = glm(Default ~ duration + amount + installment + age
              + history + purpose + foreign + housing
              , data = trainingset, family = binomial(link = logit))
  pred = predict.glm(model, new = testingset)
  predprob = exp(pred)/(1 + exp(pred))
}
summary(model)
```

```
##
## Call:
## glm(formula = Default ~ duration + amount + installment + age +
##     history + purpose + foreign + housing, family = binomial(link = logit),
##     data = trainingset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3849  -0.7912  -0.5253   0.9077   2.4753
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.272e-01  5.735e-01   0.919 0.357935
## duration     2.568e-02  8.887e-03   2.890 0.003850 **
## amount       9.761e-05  4.020e-05   2.428 0.015182 *
## installment  2.565e-01  8.185e-02   3.133 0.001729 **
## age         -1.978e-02  8.215e-03  -2.408 0.016057 *
## historyA31  -1.585e-01  5.037e-01  -0.315 0.752981
## historyA32  -1.176e+00  3.950e-01  -2.979 0.002894 **
## historyA33  -1.506e+00  4.545e-01  -3.313 0.000922 ***
## historyA34  -2.106e+00  4.211e-01  -5.001 5.69e-07 ***
## purposeA41  -1.687e+00  3.446e-01  -4.896 9.79e-07 ***
## purposeA410 -6.089e-01  7.224e-01  -0.843 0.399253
## purposeA42  -5.596e-01  2.450e-01  -2.284 0.022344 *
## purposeA43  -1.029e+00  2.342e-01  -4.392 1.12e-05 ***
## purposeA44  -4.138e-01  6.734e-01  -0.614 0.538887
## purposeA45  -5.289e-01  5.633e-01  -0.939 0.347769
## purposeA46   8.952e-03  3.738e-01   0.024 0.980896
## purposeA48  -1.539e+01  4.719e+02  -0.033 0.973986
## purposeA49  -8.450e-01  3.218e-01  -2.626 0.008641 **
## foreignA202 -1.299e+00  5.996e-01  -2.166 0.030291 *
## housingA152 -5.155e-01  2.110e-01  -2.443 0.014561 *
## housingA153 -3.192e-01  3.323e-01  -0.961 0.336675
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1083.87  on 899  degrees of freedom
## Residual deviance:  921.73  on 879  degrees of freedom
```

```
## AIC: 963.73
##
## Number of Fisher Scoring iterations: 14
```

Regression Coefficient Interpretation: When taking a look at the **installment** regression coefficient for our predicitve logistic model,we can see that our estimated coefficient is 0.2. This implies that a change from x_1 to $x_1 + 1$ will change the odds of occurrence by a factor of $e^{0.2} = 1.2214$. It increases the odds by $100(1.28904 - 1) = 22.1403\%$.

```
predstatus = predprob >= 0.1666666666666
```

Here we choose our cut off rate to be 1/6 to be conservative with our testing.

```
tp = sum((predstatus == 1) * (testingset$Default == 1))
tn = sum((predstatus == 0) * (testingset$Default == 0))
fp = sum((predstatus == 1) * (testingset$Default == 0))
fn = sum((predstatus == 0) * (testingset$Default == 1))
misrate = (fp + fn)/100
cbind(tp,tn,fp,fn,misrate)
```

```
##      tp tn fp fn misrate
## [1,] 32 23 38  7    0.45
```

Here we get our false negative to be in the single digits, and our false positive to be around the 40s.

Let us now try adding more predictors to our predictive model to better reduce the errors. We will now be adding residence, checkingstatus1 and job to our predictors.

```
for (j in 1:1){
  training = sample(1:1000, 900)
  trainingset = credit[training,]
  testingset = credit[-training,]
  model = glm(Default ~ duration + amount + installment + age
              + history + purpose + foreign + housing + residence + job + checkingstatus1
              , data = trainingset, family = binomial(link = logit))
  pred = predict.glm(model, new = testingset)
  predprob = exp(pred)/(1 + exp(pred))
}
predstatus = predprob >= 0.1666666666666
tp = sum((predstatus == 1) * (testingset$Default == 1))
tn = sum((predstatus == 0) * (testingset$Default == 0))
fp = sum((predstatus == 1) * (testingset$Default == 0))
fn = sum((predstatus == 0) * (testingset$Default == 1))
misrate = (fp + fn)/100
cbind(tp,tn,fp,fn,misrate)
```

```
##      tp tn fp fn misrate
## [1,] 29 37 31  3    0.34
```

As we can see from the false negatives and positives, they have reduced from our previous predictive model and we can see a much better improvement with out misrate.

Part B

Let us try a less conservative cutoff, maybe $\frac{7}{10}$.

```
predstatus = predprob >= 0.7
tp = sum((predstatus == 1) * (testingset$Default == 1))
tn = sum((predstatus == 0) * (testingset$Default == 0))
fp = sum((predstatus == 1) * (testingset$Default == 0))
fn = sum((predstatus == 0) * (testingset$Default == 1))
sensitivity<-tp/(tp+fn)
specificity<-tn/(tn+fp)
cbind(sensitivity,specificity)
```

```
##      sensitivity specificity
## [1,]        0.25   0.9705882
```

Now let us try a more middle cutoff, let the cutoff be $\frac{1}{2}$

```
predstatus = predprob >= 0.5
tp = sum((predstatus == 1) * (testingset$Default == 1))
tn = sum((predstatus == 0) * (testingset$Default == 0))
fp = sum((predstatus == 1) * (testingset$Default == 0))
fn = sum((predstatus == 0) * (testingset$Default == 1))
sensitivity<-tp/(tp+fn)
specificity<-tn/(tn+fp)
cbind(sensitivity,specificity)
```

```
##      sensitivity specificity
## [1,]         0.5   0.8970588
```

Finally let us try a conservative cutoff, let it be what we chose early on, which was $\frac{1}{6}$.

```
predstatus = predprob >= 0.1666666666666
tp = sum((predstatus == 1) * (testingset$Default == 1))
tn = sum((predstatus == 0) * (testingset$Default == 0))
fp = sum((predstatus == 1) * (testingset$Default == 0))
fn = sum((predstatus == 0) * (testingset$Default == 1))
sensitivity<-tp/(tp+fn)
specificity<-tn/(tn+fp)
cbind(sensitivity,specificity)
```

```
##      sensitivity specificity
## [1,]     0.90625   0.5441176
```

In this case the sensitivity increases as we become more conservative with our cutoff, which means as the cutoff value becomes smaller, the sensitivity increases. For this application, we the rate of risky applicants being rejected. When it comes to specificity, we see an opposite direction when the cutoff value becomes smaller. This means as the cutoff value becomes smaller, so does the specificity rate. For our application the specificity refers to the number of applicants approved.

Question 6

In logistic regression, we assume $Y = (Y_1, ..., Y_n)^T$ are a collection of n binary observations. For each Y_i, we observe $X_i = (X_{i1}, X_{i2}, X_{i3})^T$ predictors. We assume

$$log(\frac{p_i}{1-p_i}) = X_i^T\theta$$

where $\theta = (\theta_1, ...\theta_3)^T$ is the vector of regression coefficients.

a) Formulate the overall log likelihood of the data set $l(Y)$

b) Derive the first derivative $\frac{\partial l(Y)}{\partial \theta_2}$.

c) Derive the second derivative $\frac{\partial^2 l(Y)}{\partial \theta_2 \partial \theta_3}$.

d) Suppose $\hat{\theta} = (0.1, 0.2, 0.3)^T$, and we have a new observation $X_{n+1} = (3, 2, 4)^T$. Predict the probability p of success for this new observation.

Solution

Part A

For the i^{th} observation:

$$\begin{aligned} P(Y_i) &= \binom{1}{y_i} p_i^{y_i}(1-p_i)^{1-y_i} \\ P(Y_i) &= p_i^{y_i}(1-p_i)^{1-y_i} \\ log(P(Y_i)) &= log(p_i^{y_i}(1-p_i)^{1-y_i}) \\ l(P(y_i)) &= y_i log(p_i) + (1-y_i)log(1-p_i) \end{aligned}$$

To get the whole data set, we will sum from 1 to n.

$$\begin{aligned} &= \sum_{i=1}^{n} l(Y_i) \\ &= \sum_{i=1}^{n} y_i log(p_i) + (1-y_i)log(1-p_i) \\ &= \sum_{i=1}^{n} y_i log\Big(\frac{e^{x_i^T\theta}}{1+e^{x_i^T\theta}}\Big) + (1-y_i)log\Big(1 - \frac{e^{x_i^T\theta}}{1+e^{x_i^T\theta}}\Big) \\ &= \sum_{i=1}^{n} y_i log\Big(\frac{e^{x_i^T\theta}}{1+e^{x_i^T\theta}}\Big) + (1-y_i)log\Big(\frac{1}{1+e^{x_i^T\theta}}\Big) \\ &= \sum_{i=1}^{n} y_i log\Big(\frac{e^{a}}{1+e^{a}}\Big) + (1-y_i)log\Big(1 - \frac{e^{a}}{1+e^{a}}\Big); a = x_i^T\theta = \sum_{j=1}^{3} x_{ij}\theta_j \end{aligned}$$

Part B

Note that chain rule will be applied to solve the first derivative.

$$\frac{\partial l(Y_i)}{\partial \theta_2} = \sum_{i=1}^{n} \frac{\partial l(Y_i)}{p_i} \frac{\partial p_i}{\partial a} \frac{\partial a}{\partial \theta_2}$$

let us look at the $\frac{\partial p_i}{\partial a}$ and $\frac{\partial a}{\partial \theta_2}$ separately first.

$$\begin{aligned} \frac{\partial p_i}{\partial a} &= \frac{e^a(1+e^a) - (e^a)(e^a)}{(1+e^a)^2} \\ &= \frac{e^a}{(1+e^a)^2} \\ &= p_i(1-p_i) \end{aligned}$$

$$\frac{\partial a}{\partial \theta_2} = x_{i2}$$

Now let us solve the first derivative by using the previous derivatives we solved.

$$\begin{aligned} \frac{\partial l(Y_i)}{\partial \theta_2} &= \sum_{i=1}^{n} \frac{\partial l(Y_i)}{p_i} \frac{\partial p_i}{\partial a} \frac{\partial a}{\partial \theta_2} \\ &= \sum_{i=1}^{n} \Big(\frac{y_i}{p_i} - \frac{1-y_i}{1-p_i} \Big) (p_i(1-p_i)) x_{i2} \\ &= \sum_{i=1}^{n} \Big(\frac{(y_i(1-p_i)) - (1+y_i)p_i}{p_i(1-p_i)} \Big) (p_i(1-p_i)) x_{i2} \\ &= \sum_{i=1}^{n} (y_i - p_i) x_{i2} \end{aligned}$$

Part C

Let us now calculate the 2nd derivative.

$$\begin{aligned} \frac{\partial^2 l(Y_i)}{\partial \theta_2 \theta_3} &= \frac{\partial}{\partial \theta_3} \Big(\sum_{i=1}^{n} (y_i - p_i) x_{i2} \Big) \\ &= \frac{\partial}{\partial \theta_3} \Big(\sum_{i=1}^{n} \Big(y_i - \frac{e^a}{1+e^a} \Big) x_{i2} \Big) \\ &= \sum_{i=1}^{n} \frac{\partial \Big(yi - \frac{e^a}{1+e^a} \Big)}{\partial a} \frac{\partial a}{\partial \theta_3} x_{i2} \\ &= \sum_{i=1}^{n} -p_i(1-p_i) x_{i2} x_{i3} \end{aligned}$$

Part D

```
theta = t(c(0.1,0.2,0.3))
x_new = c(3,2,4)
p_i = theta%*%x_new
prob = round(exp(p_i)/ (1 + exp(p_i)),4)
table = cbind(p_i,prob)
colnames(table) = c('p_i','probability')
table
```

```
##      p_i probability
## [1,] 1.9      0.8699
```

As we can see the probability p of success for the new observation will be 0.8699.