MATH 208 Assignment 3

The assignment contains one question **with 5 parts (a)-(e)**, each worth 10 points, for a total of 50 points. Your answers must be submitted in the form of a PDF and include both the answers to the question, along with your R code and output used to generate your answers.

Question 1 (50 points)

The basics

Logistic regression is a fundamental prediction model in statistics and modern data science. Assume that we have observed two predictors, X_{i1} and X_{i2} and want to predict a **binary** outcome Y_i (i.e. $Y_i = 0$ or $Y_i = 1$). A logistic regression model assumes that the probability that $Y_i = 1$ can be modelled using the following function of $X_{i1} = x_{i1}$ and $X_{i2} = x_{i2}$.

$$Pr(Y_i = 1 | X_{i1} = x_{i1}, X_{i2} = x_{i2}, \theta_1, \theta_2, \theta_3) = p(x_{i1}, x_{i2}) = \frac{1}{1 + \exp(-x_{i1}\theta_1 - x_{i2}\theta_2 - \theta_3))}.$$

(a) Write a function to compute $p(x_1, x_2)$ for n observations which takes as arguments:

i) A vector of three parameters $\theta = (\theta_1, \theta_2, \theta_3)$.
ii) Two predictor vectors, $x_1 = (x_{1,1}, ..., x_{n,1})$ and $x_2 = (x_{1,2}, ...x_{n,2})$

and returns a length n vector corresponding to $p(x_{11}, p_{12}), ...p(x_{n1}, x_{n2})$ for the corresponding θ values. **Hint:** You can do this without loops by subscripting for θ and using vectorized calculations for x_1 and x_2.

Given a dataset of n observations where we observe $(Y, X_1, X_2) = (y_i, x_{i1}, x_{i2})$ for each observation i, one way to estimate values for θ_1, θ_2 and θ_3 is to minimize the *cross-entropy loss*:

$$L(\theta_1, \theta_2, \theta_3) = -\sum_{i=1}^{n} [y_i \times \log(p(x_{i1}, x_{i2})) + (1 - y_i) \times \log(1 - p(x_{i1}, x_{i2}))]$$

Note that because $0 \leq p(x_1, x_2) \leq 1$, $L(\theta_1, \theta_2, \theta_3)$ will be smaller when $p(x_{i1}, x_{i2})$ is close to 1 for $y_i = 1$ and $p(x_{i1}, x_{i2})$ is close to 0 for $y_i = 0$.

(b) Write a function to compute $L(\theta_1, \theta_2, \theta_3)$ for n observations which takes as arguments:

i) A vector of three parameters $\theta = (\theta_1, \theta_2, \theta_3)$.
ii) Two predictor vectors, $x_1 = (x_{1,1}, ..., x_{n,1})$ and $x_2 = (x_{1,2}, ...x_{n,2})$
iii) An outcome vector, $y = (y_1, ..., y_n)$

Hint: Use your function $p(x_1, x_2)$ from part (a).

CONTINUED ON NEXT PAGE

Writing a function to use with optim

`optim` is an opitmizer function that, by default, minimizes an argument function `fn` as a function of a vector first argument of `fn`, starting from initial values `par`. Other arguments for `fn` can be passed in `...`. An example function of using `optim` would be:

```
## The loss function is (x_1-a)^4 + (x_2 - b)^4, which is minimized at
## x_1 = a/2 and x_2 = b/2.

f_x <- function(x,a,b){
      (x[1]-a/2)^4 + (x[2]-b/2)^4 + 8
}

### optim can approximately minimize this function
### using its default optimization algorithm
result<-optim(par=c(10,15), fn=f_x, a=3, b=2)

## result$par:  The values that minimize (x[1], x[2])
## result$value: the minimum value of f acheived at result$estimate
## result$counts: The number of iterations the algorithm took
##                 to converge (ignore gradient for now)
## result$code:  0 indicates a reliable convergence result, anything else
##              is a problem
## result$message: A written description of any issues in converging

result
```

```
$par
[1] 1.4257683 0.9559005

$value
[1] 8.000034

$counts
function gradient
      39       NA

$convergence
[1] 0

$message
NULL
```

(c) Fit a logistic regression classifier to the HTRU2 data, choosing Y to be the Class values (coded as 0 and 1), X_1 to be the Mean IP values and X_2 to be the Mean DMSNR values using the `optim()` function in R. Using `optim` and your loss function from part (b), find the values of `theta[1], theta[2], theta[3]` that minimize the cross-entropy loss. Report your estimates for $(\theta_1.\theta_2,\theta_3)$ and the estimated loss (and be sure to include the code that allowed you to achieve it). Note, you do not need to write a new function to do this with associated arguments, you simply can write a block of R code accomplishes the task. Starting `optim` at `par=c(0,0,0)` works well this model.

CONTINUED ON NEXT PAGE

Applying your code

(d) For this part, you should write code using a `for` loop (or loops) to compute the minimized cross-entropy loss for each possible pair of predictors for the HTRU2 data (note there are $\binom{8}{2} = 28$ possible models) and then store the results in a tibble with each row containing the names of the two variables used in the modelling and their cross-entropy loss). You can then arrange the rows by the value of the loss to find create a table ordered from best pairs of predictors to worst pairs according to estimated loss. Display your ordered table using the `kable(.)` function. Include all the code used to generate your results.
Note: starting `optim` at `par=c(0,0,0)` actually works well in all 28 models (this will not always be the case!).

Hint: I found it easiest to first use the `combn()` function to generate a 2×28 matrix where the columns contain all possible pairs pairs of names.

```
var_combs<-combn(names(HTRU2[,-9]),2) ## -9 excludes the 9th column, the Class variable
dim(var_combs)
```

```
[1]  2 28
```

```
var_combs[,1:4]
```

```
     [,1]      [,2]      [,3]      [,4]
[1,] "Mean_IP" "Mean_IP" "Mean_IP" "Mean_IP"
[2,] "SD_IP"   "EK_IP"   "SKW_IP"  "Mean_DMSNR"
```

By using this matrix, you need to only use a single `for` loop over the 28 columns, extract the correct two predictor columns from HTRU2, run the code from part (c) and collect the results in a tibble. You may also use two **nested** `for` loops and the vector of column names, but it is a bit trickier to do so (as well as store the results).

(e) Finally, produce the same tibble as in part (d), only using the `var_combs` matrix above and `map_dfr(.)`.
Hint: You may find it useful to convert `var_combs` to a `data.frame` or `tibble` first.