

MATH 208 Assignment3

Yan Miao

2019-11-04

```
library(tidyverse)
library(kableExtra)
```

Question1

```
HTRU2<-read_csv("HTRU_2.csv",col_names = FALSE)
names(HTRU2) = c("Mean_IP", "SD_IP", "EK_IP", "SKW_IP", "Mean_DMSNR",
                 "SD_DMSNR", "EK_DMSNR", "SKW_DMSNR","Class")
head(HTRU2)
```

```
## # A tibble: 6 x 9
##   Mean_IP SD_IP   EK_IP SKW_IP Mean_DMSNR SD_DMSNR EK_DMSNR SKW_DMSNR Class
##   <dbl> <dbl>   <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>
## 1  141.   55.7 -0.235 -0.700     3.20    19.1     7.98    74.2     0
## 2  103.   58.9  0.465 -0.515     1.68    14.9    10.6    127.     0
## 3  103.   39.3  0.323  1.05     3.12    21.7     7.74    63.2     0
## 4  137.   57.2 -0.0684 -0.636     3.64    21.0     6.90    53.6     0
## 5   88.7  40.7  0.601  1.12     1.18    11.5    14.3    253.     0
## 6   93.6  46.7  0.532  0.417     1.64    14.5    10.6    131.     0
```

```
X1<-HTRU2$Mean_IP
X2<-HTRU2$Mean_DMSNR
Y<-HTRU2$Class
```

(a)

```
sigmoid <- function(theta,x1,x2){
  z<-theta[1]*x1+theta[2]*x2+theta[3]
  return(1/(1+exp(-z)))
}
```

(b)

```
loss <- function(theta,x1,x2,y){
  p<-sigmoid(theta,x1,x2)
  L<-sum(y * log(p) + (1 - y) * log(1 - p))
  return(L)
}
```

(c)

```
result<-optim(par=c(0,0,0), fn=loss,x1=X1,x2=X2,y=Y)
result
```

```
## $par
## [1] -0.10569326  0.01629013  7.28979911
##
## $value
## [1] 1991.015
```

```
##
## $counts
## function gradient
##      218      NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

So the estimates for $(\theta_1, \theta_2, \theta_3)$ is $(-0.10569326, 0.01629013, 7.28979911)$ and the estimated loss is 1991.015.

(d)

```
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"
```

```
result_table<-tibble()
for (idx in 1:28){
  pair<-var_combs[,idx]

  col1<-pair[1]
  col2<-pair[2]

  col_names<-paste(col1, col2, sep=" & ") # String concatenation

  X1<-HTRU2[[col1]]
  X2<-HTRU2[[col2]]

  result<-optim(par=c(0,0,0), fn=loss,x1=X1,x2=X2,y=Y)
  new_row<-tibble(value=result$value,names=col_names)

  result_table<-result_table%>% bind_rows(new_row) #append new rows to tibble
}
```

```
result_table %>% arrange(value) %>% kable(.)
```

value	names
1427.745	EK_IP & SD_DMSNR
1429.591	EK_IP & EK_DMSNR
1434.257	EK_IP & SKW_DMSNR
1450.829	EK_IP & SKW_IP
1483.505	Mean_IP & EK_IP
1490.764	SD_IP & EK_IP
1502.008	EK_IP & Mean_DMSNR
1759.214	Mean_IP & SD_DMSNR
1763.425	Mean_IP & EK_DMSNR
1790.573	Mean_IP & SKW_DMSNR
1834.243	SKW_IP & SD_DMSNR
1839.221	SKW_IP & EK_DMSNR
1875.364	SKW_IP & SKW_DMSNR
1918.023	Mean_IP & SKW_IP
1991.015	Mean_IP & Mean_DMSNR
2021.685	SKW_IP & Mean_DMSNR
2052.101	Mean_IP & SD_IP
2305.642	SD_IP & SKW_IP
2777.460	SD_IP & EK_DMSNR
2877.531	SD_IP & SD_DMSNR
2953.056	SD_IP & SKW_DMSNR
3365.135	SD_IP & Mean_DMSNR
3772.916	Mean_DMSNR & EK_DMSNR
3800.222	SD_DMSNR & SKW_DMSNR
3808.527	EK_DMSNR & SKW_DMSNR
3809.508	SD_DMSNR & EK_DMSNR
3869.097	Mean_DMSNR & SKW_DMSNR
3971.733	Mean_DMSNR & SD_DMSNR

The best pair with the minimum loss 1427.745 is “EK_IP” and “SD_DMSNR”. And the worst pair with the maximum loss 3971.733 is “Mean_DMSNR” & “SD_DMSNR”.

(e)

```
var_combs_df<-as.data.frame(var_combs)
var_combs_df
```

```
##          V1          V2          V3          V4          V5          V6          V7          V8
## 1 Mean_IP Mean_IP Mean_IP Mean_IP Mean_IP Mean_IP Mean_IP Mean_IP SD_IP
## 2 SD_IP EK_IP SKW_IP Mean_DMSNR SD_DMSNR EK_DMSNR SKW_DMSNR EK_IP
##          V9          V10          V11          V12          V13          V14          V15          V16
## 1 SD_IP SD_IP SD_IP SD_IP SD_IP EK_IP EK_IP EK_IP
## 2 SKW_IP Mean_DMSNR SD_DMSNR EK_DMSNR SKW_DMSNR SKW_IP Mean_DMSNR SD_DMSNR
##          V17          V18          V19          V20          V21          V22          V23
## 1 EK_IP EK_IP SKW_IP SKW_IP SKW_IP SKW_IP Mean_DMSNR
## 2 EK_DMSNR SKW_DMSNR Mean_DMSNR SD_DMSNR EK_DMSNR SKW_DMSNR SD_DMSNR
##          V24          V25          V26          V27          V28
## 1 Mean_DMSNR Mean_DMSNR SD_DMSNR SD_DMSNR EK_DMSNR
## 2 EK_DMSNR SKW_DMSNR EK_DMSNR SKW_DMSNR SKW_DMSNR
```

```
# Work with the names
a<-var_combs %>% t(.)
b<-paste(a[,1],a[,2], sep=" & ")
b
```

```
## [1] "Mean_IP & SD_IP"      "Mean_IP & EK_IP"
## [3] "Mean_IP & SKW_IP"     "Mean_IP & Mean_DMSNR"
## [5] "Mean_IP & SD_DMSNR"   "Mean_IP & EK_DMSNR"
## [7] "Mean_IP & SKW_DMSNR"  "SD_IP & EK_IP"
## [9] "SD_IP & SKW_IP"       "SD_IP & Mean_DMSNR"
## [11] "SD_IP & SD_DMSNR"     "SD_IP & EK_DMSNR"
## [13] "SD_IP & SKW_DMSNR"    "EK_IP & SKW_IP"
## [15] "EK_IP & Mean_DMSNR"    "EK_IP & SD_DMSNR"
## [17] "EK_IP & EK_DMSNR"     "EK_IP & SKW_DMSNR"
## [19] "SKW_IP & Mean_DMSNR"   "SKW_IP & SD_DMSNR"
## [21] "SKW_IP & EK_DMSNR"    "SKW_IP & SKW_DMSNR"
## [23] "Mean_DMSNR & SD_DMSNR" "Mean_DMSNR & EK_DMSNR"
## [25] "Mean_DMSNR & SKW_DMSNR" "SD_DMSNR & EK_DMSNR"
## [27] "SD_DMSNR & SKW_DMSNR" "EK_DMSNR & SKW_DMSNR"
```

```
var_combs_df %>% map_dfr(~HTRU2 %>% select(as.character(.x)) %>%
  optim(par=c(0,0,0), fn=loss, x1=. [[1]],x2=. [[2]],y=Y) %>%
  .$value %>% as_tibble()) %>%
  add_column(.,names=b)%>% arrange(value) %>% kable(.)
```

value	names
1427.745	EK_IP & SD_DMSNR
1429.591	EK_IP & EK_DMSNR
1434.257	EK_IP & SKW_DMSNR
1450.829	EK_IP & SKW_IP
1483.505	Mean_IP & EK_IP
1490.764	SD_IP & EK_IP
1502.008	EK_IP & Mean_DMSNR
1759.214	Mean_IP & SD_DMSNR
1763.425	Mean_IP & EK_DMSNR
1790.573	Mean_IP & SKW_DMSNR
1834.243	SKW_IP & SD_DMSNR
1839.221	SKW_IP & EK_DMSNR
1875.364	SKW_IP & SKW_DMSNR
1918.023	Mean_IP & SKW_IP
1991.015	Mean_IP & Mean_DMSNR
2021.685	SKW_IP & Mean_DMSNR
2052.101	Mean_IP & SD_IP
2305.642	SD_IP & SKW_IP
2777.460	SD_IP & EK_DMSNR
2877.531	SD_IP & SD_DMSNR
2953.056	SD_IP & SKW_DMSNR
3365.135	SD_IP & Mean_DMSNR
3772.916	Mean_DMSNR & EK_DMSNR
3800.222	SD_DMSNR & SKW_DMSNR
3808.527	EK_DMSNR & SKW_DMSNR
3809.508	SD_DMSNR & EK_DMSNR
3869.097	Mean_DMSNR & SKW_DMSNR
3971.733	Mean_DMSNR & SD_DMSNR

We can see that this is exactly the same as the table in part (d), which is created using for loop.