Tennis Group Project

Samuel Stockman, Angie Gray, Jack Simons

10/12/2020

Let's first load in the relevant data (see `tennis.R`)

```
file_list = list.files()
```

Now we want to compile the data into a better format. We need the input `variables` which should be equal to something of the form: `variables = c('surface', 'winner_id', 'loser_id', 'loser_age', 'winner_age')` - alternatively if it left blank then we will load all the variables

```
load_data <- function(file_list, variables=FALSE) {
  no_files <- length(file_list)
  for (file_index in 1:no_files) {
    temp_file <- read.csv(file_list[[file_index]])
    if (class(variables)=="character") {
      temp_file <- subset(temp_file, select = variables)
    }
    if (file_index==1) {
      melted_files <- temp_file
    } else {
      melted_files <- rbind(melted_files, temp_file)
    }
  }
  return(melted_files)
}
```

Let's now use this function

```
# variables <- c('surface', 'winner_id', 'loser_id', 'winner_age', 'loser_age')
my_data <- load_data(file_list)
```

Let's see what the data looks like

```
head(my_data, 5)
```

```
##   tourney_id              tourney_name surface draw_size tourney_level
## 1  2015-D001  Fed Cup WG F: CZE vs RUS    Hard         4             D
## 2  2015-D001  Fed Cup WG F: CZE vs RUS    Hard         4             D
## 3  2015-D001  Fed Cup WG F: CZE vs RUS    Hard         4             D
## 4  2015-D001  Fed Cup WG F: CZE vs RUS    Hard         4             D
## 5  2015-D002 Fed Cup WG R1: CAN vs CZE    Hard         4             D
##   tourney_date match_num winner_id winner_seed winner_entry       winner_name
## 1     20151114         1    201520        <NA>                  Petra Kvitova
## 2     20151114         2    201345        <NA>                Maria Sharapova
## 3     20151114         3    201345        <NA>                Maria Sharapova
## 4     20151114         4    201662        <NA>              Karolina Pliskova
## 5     20150207         1    201662        <NA>              Karolina Pliskova
```

```
##   winner_hand winner_ht winner_ioc winner_age loser_id loser_seed loser_entry
## 1           L       183        CZE   25.68652   201499       <NA>
## 2           R        NA        RUS   28.57221   201662       <NA>
## 3           R        NA        RUS   28.57221   201520       <NA>
## 4           R       184        CZE   23.64956   201499       <NA>
## 5           R       184        CZE   22.88296   211796       <NA>
##                 loser_name loser_hand loser_ht loser_ioc loser_age       score
## 1 Anastasia Pavlyuchenkova          R      177       RUS  24.36687 2-6 6-1 6-1
## 2        Karolina Pliskova          R      184       CZE  23.64956     6-3 6-4
## 3            Petra Kvitova          L      183       CZE  25.68652 3-6 6-4 6-2
## 4 Anastasia Pavlyuchenkova          R      177       RUS  24.36687     6-3 6-4
## 5         Francoise Abanda          R       NA       CAN  18.00411     6-2 6-4
##   best_of round minutes w_ace w_df w_svpt w_1stIn w_1stWon w_2ndWon w_SvGms
## 1       3    RR      NA    NA   NA     NA      NA       NA       NA      NA
## 2       3    RR      NA    NA   NA     NA      NA       NA       NA      NA
## 3       3    RR      NA    NA   NA     NA      NA       NA       NA      NA
## 4       3    RR      NA    NA   NA     NA      NA       NA       NA      NA
## 5       3    RR      NA    NA   NA     NA      NA       NA       NA      NA
##   w_bpSaved w_bpFaced l_ace l_df l_svpt l_1stIn l_1stWon l_2ndWon l_SvGms
## 1        NA        NA    NA   NA     NA      NA       NA       NA      NA
## 2        NA        NA    NA   NA     NA      NA       NA       NA      NA
## 3        NA        NA    NA   NA     NA      NA       NA       NA      NA
## 4        NA        NA    NA   NA     NA      NA       NA       NA      NA
## 5        NA        NA    NA   NA     NA      NA       NA       NA      NA
##   l_bpSaved l_bpFaced winner_rank winner_rank_points loser_rank
## 1        NA        NA           6               4220         28
## 2        NA        NA           4               5011         11
## 3        NA        NA           4               5011          6
## 4        NA        NA          11               3285         28
## 5        NA        NA          22               2015        230
##   loser_rank_points
## 1              1840
## 2              3285
## 3              4220
## 4              1840
## 5               195
```

Now, for the logistic regression. We need to create our $D = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ - we will start off very naively

```
trans_data_naive <- function(data_mat, features) {

  no_feats <- length(features)
  new_dat <- subset(data_mat, select = features) #select the relewvant features
  orig_rows <- nrow(new_dat) #number of orginal games (which we double)

  #flip the dataset
  for (i in 1:(no_feats/2)) {
    temp_df <- data.frame(new_dat[,(2*i)], new_dat[,(2*i-1)])
  }
  colnames(temp_df)=features
  new_dat <- rbind(new_dat, temp_df)

  #add the results
  new_dat$result <- c(rep(1,orig_rows), rep(-1, orig_rows))
  new_dat$result <- as.numeric(new_dat$result)
```

```
  #change the column names to reflect that we have added result column
  colnames(new_dat)[1:2] <- c("p1_hand", "p2_hand")

  #add the bias column
  new_dat$bias <- 1

  return(new_dat)
}
```

Now let's look at this data

```
features <- c("winner_hand", "loser_hand")
new_dat <- trans_data_naive(my_data, features)
head(new_dat)
```

```
##   p1_hand p2_hand result bias
## 1       L       R      1    1
## 2       R       R      1    1
## 3       R       L      1    1
## 4       R       R      1    1
## 5       R       R      1    1
## 6       R       R      1    1
```

We notice that some people do not have a hand are are listed as "" or "U". **Let's choose to remove these from the data set.**

```
new_dat <- new_dat[-which(new_dat$p1_hand=="U"),]
new_dat <- new_dat[-which(new_dat$p1_hand==""),]
new_dat <- new_dat[-which(new_dat$p2_hand=="U"),]
new_dat <- new_dat[-which(new_dat$p2_hand==""),]
unique(new_dat$p2_hand)
```

```
## [1] "R" "L"
```

```
unique(new_dat$p1_hand)
```

```
## [1] "L" "R"
```

Let's one hot encode this variable

```
new_dat$p1_hand[which(new_dat$p1_hand=="L")] <- 0
new_dat$p1_hand[which(new_dat$p1_hand=="R")] <- 1
new_dat$p2_hand[which(new_dat$p2_hand=="L")] <- 0
new_dat$p2_hand[which(new_dat$p2_hand=="R")] <- 1
new_dat$p1_hand <- as.numeric(new_dat$p1_hand)
new_dat$p2_hand <- as.numeric(new_dat$p2_hand)
```

We need to separate y and $\boldsymbol{x}$.

```
y_dat <- subset(new_dat, select=result)
x_dat <- subset(new_dat, select=-result)
```

Let's make the logistic regression

```
sigma_f <- function(t) {
  return(1/(1+exp(t)))
}
```

```r
log_reg_naive <- function(w, x, y) {
  f <- as.matrix(x)%*%w
  return(-sum(log(sigma_f(f*y))))
}

log_reg_naive_grad <- function(w, x, y) {
  f <- as.matrix(x)%*%w
  temp <- rep(0,dim(x)[2])
  for (i in 1:dim(x)[2]) {
    temp[i] <- sum(sigma_f(f*y)*exp(f*y)*y*x[,i])
  }
  return(temp)
}

set.seed(2020) #random seed for reproducibility
w <- rnorm(3) #random initialization
log_reg_naive(w, x_dat, y_dat)
```

```
## [1] 22208.67
```

```r
optim(rnorm(3), log_reg_naive, x=x_dat, y=y_dat, control=list(maxit=2000), gr=log_reg_naive_grad)
```

```
## $par
## [1] -1.295794e-02  1.286150e-02  7.472947e-05
##
## $value
## [1] 21280.91
##
## $counts
## function gradient
##      124       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```