Hackathon on US elections 2020?

Agnes Gaspart, Romà Masana

February 2020

Contents

1 Introduction

We are planning to hold a hackathon of prediction contest at around 4th of April (a Saturday) during the whole day at PS111. I thought the topic of trying to predict the **US presidential election for 2020**, but Agnes and Prof. Kang say this is going to be complicated, if not impossible, and that another topic should be chosen. Agnes advocates for predicting and growing our models on **Emotion prediction** which I actually think is quite interesting. Agnes claims she already has the dataset and that all we have to do is run it.

However, when we asked a passer-by (by the name of *Hunter the sharp*) he said that the topic of the US election would be *spicy*, and that could be an attractive in order to draw the attention of the mainstream IIT studentdom.

For me we can do it about any other topic. But my passion lies on the US elections of 2020. We asked Prof. Kang and she was amused by the fact that we were going to attempt that. She said we won't obtain any rigorous results. Maybe she is right. Or maybe she is not. ;)

2 Databases

Prof. Kang suggested the following places were to mine for relevant data:

1. 538 political website https://fivethirtyeight.com/
2. Form the US Census bureau, I downloaded here https://www.census.gov/data/tables/2019/demo/income-poverty/p60-266.html the **table A-2**, containing the **mean, median and other relevant percentiles** of the **income per household** from **1967 until 2018**.
3. from US Census bureau: **Historical poverty** tables: people and families : https://www.census.gov/data/tables/time-series/demo/income-poverty/historical-poverty-people.html 1959 until 2018.
4. from US Census bureau: **Historical School enrollment** tables: https://www.census.gov/data/tables/time-series/demo/school-enrollment/cps-historical-time-series.html

5. Polls from primary elections since 1970. https://fivethirtyeight.com/features/what-more-than-40-years-of-early-primary-polls-tell-us-about-2020-part-1/

I downloaded all these databases:

1. TableV9 Poverty by region from 1959 to 2018
2. Table v13 Families below poverty from 1959 to 2018
3. Table v15 Age distribution of the poor from 1959 to 2018
4. Table v23 Poverty by nativity from 1993 to 2018
5. Table A2 total money income by race from 1967 to 2018.
6. Table A4 Household income dispersion measures 1967 to 2018.
7. Table A7 Real median and mean earnings of total workers and full time workers, female-male from 1960 to 2018.
8. Table A2 Population 3 years and older enrolled in some sort of school from 1947 to 2018.
9. Table 5B Population 18 and 19 yrs old in high school, that dropped out, or that are in college, or none of the previous but that graduated from high school. From 1967 to 2018.

2.1 Interesting database

To be found at
https://archive.ics.uci.edu/ml/datasets/student+performance

3 Add emotions of people

4 R coding

My initial idea was to make students grow their own machine learning models using the R software. In other words, I already did this in R and I know how to do it in this software. Probably this could be done in any other language such as Python, but I don't exactly know how it works there and we would have to spend a lot of time in order to do it there. At least I would have to. So for me is more efficient to explain things using R.

4.1 Introduction to R

1. First download and install the R software for free at https://www.r-project.org/
   Then download the R studio desktop for free at https://rstudio.com/products/rstudio/download/
2. Basic Introduction to R (up to dataframes and matrices)
   http://www.r-tutor.com/r-introduction/matrix
3. Explanation with examples of how to use the "apply, lapply, vapply" functions on R:
   https://ademos.people.uic.edu/Chapter4.html

4.2 Preliminar R code for growing the random trees

Feel free to copy paste this code into your R studio. For instance into an R script. For each **library** you will have to install that package first using the command

```
install.packages('pracma')
```

for instance. Do this for all libraries. Then study what's written in here. The code should not work, but it shows the function we will be using for growing a random forsest, and some of the parameters that I consider relevant. there are many more. To fully know which parameters are there in the function

```
quantregForest()
```

Just run

```
help(``quantregForest'')
```

Or read the full description of the "quantregForest" package here
https://cran.r-project.org/web/packages/quantregForest/quantregForest.pdf .

```
library('pracma')
library(e1071)
library(quantregForest)
library(rpart)
library(gamlss)
library(dplyr)
library(ggplot2)
library(tidyr)
library(Rcpp)
library(SpecsVerification)
library(logspline)
library(evmix)

#qrforest function grows the tree
#for instance, using qrforest(pset,400,20,wdata,ppqu)

# Explanation of the parameters of the quantregForest: ################
#
#   x is the dataframe containing the predictor variables
#   y is the response variable, the observations?
#   nthreads = number of threads or computer cores to use
    nc = detectCores() #detects the number of cores of your computer
#   ntree = number of trees to use. Predictions will be the average tree prediction of the wood
#   nodesize : is the minimal amount of datapoints that a node should have in order to further be split.
          #it is a stopping criteria


print(system.time( qtree <- quantregForest(x=df2,y=df$train$obs,nthread = 3, ntree=ntree, nodesize=nsize) )

predictors = c('elev','lon','MEANavg','MEAN','lat')

nthreads=3
ntree=1
nodesize=20
quantiles=t(t( linspace(0.01,0.99,99) ))#I don't know if we use the percentiles in this problem

qrforest(ntree=1,nsize=nodesize, ptors=predictors, ppqu=quantiles )

qrforest <- function(ntree, nsize, ptors, ppqu, note="") {
  print(paste('forest',ntree,nsize))
  print(paste("note: ",note))
  print(paste('predictors:',paste(ptors,collapse=', ')))
  print('computing forest...')
  df2 <- data.frame(df[,ptors ])
  names(df2) <- ptors

  print(system.time( qtree <- quantregForest(df2,df$train$obs,nthread = 3, ntree=ntree, nodesize=nsize) ))

  print('compu time of predicting the response variable in an independent dataset:');
```

```
  print(system.time(verifquant <-  predict(qtree, newdata = df$verif, what = ppqu)))
  return(list(forest=qtree,fquants=verifquant))
}
#clean trees are growing Roma Domenech Masana, Chicago Feb 2020
```

5 Literature

The Classification and Regression Trees (CART) were created (or at least very well explained) by Leo Breiman, Friedman, Olshen and Stone. You can find the book CART here
https://www.amazon.com/Classification-Regression-Trees-Leo-Breiman/dp/
1138469521 This is the book I read. I think they manage to explain with extreme detail how a tree works and how it is constructed. In my opinion all the other books build on this very basic but important algorithm.