Capstone Report

Fábio Franco Costa

November 4, 2015

Title

Do similar users rates places equally?

Introduction

When you are using apps like YELP, you get ratings based on the average reviews from all users. In that situation I may get good ratings for a place that I would not really appreciate, because not every user are similar to others. My problem is testing if clusters of similar users rate places equally, so that I can use the average rate of this clusters to predict how I would rate some new place.

Methods and Data

For our analysis, the first step was to decide a clustering strategy. We decided to use the features in the user database as our characteristics. We did some transformations and end up with the following structure:

```
## 'data.frame':    50000 obs. of  25 variables:
##  $ user_id           : chr  "Kb2FOnGteVLhN0ZhsmgqNw" "YlhgtRS9yArNolg8j_tpgw" "UBPKjtrReZ01gOD3953T
##  $ fans              : int  0 5 0 0 0 0 1 2 1 0 ...
##  $ average_stars     : num  3.67 3.38 4.5 3 4.33 3 5 4.26 3.59 3.67 ...
##  $ votes.funny       : int  1 269 0 4 0 9 0 3 24 1 ...
##  $ votes.useful      : int  2 293 1 5 6 15 1 17 125 2 ...
##  $ votes.cool        : int  2 206 0 2 2 1 0 5 25 1 ...
##  $ compliments.profile: num  0 4 0 0 0 0 0 0 0 0 ...
##  $ compliments.cute   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ compliments.funny  : num  0 32 0 0 0 0 0 0 1 0 ...
##  $ compliments.plain  : num  0 29 0 0 0 0 0 0 3 0 ...
##  $ compliments.writer : num  1 11 0 0 0 0 0 0 1 0 ...
##  $ compliments.note   : num  0 22 0 0 2 1 0 0 4 0 ...
##  $ compliments.photos : num  0 11 0 0 0 0 0 1 0 0 ...
##  $ compliments.hot    : num  0 65 0 0 0 0 0 2 0 0 ...
##  $ compliments.cool   : num  0 101 0 0 0 0 0 0 4 0 ...
##  $ compliments.more   : num  0 3 0 0 0 0 0 0 0 0 ...
##  $ compliments.list   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ reviews           : int  3 233 4 2 10 7 1 36 97 3 ...
##  $ friends_no        : int  0 120 0 1 0 0 9 21 22 0 ...
##  $ yelping_months    : num  39 103 28 17 27 65 37 27 84 46 ...
##  $ C1                : int  13 11 13 13 13 7 13 13 10 13 ...
##  $ C2                : int  7 5 7 7 7 13 7 7 3 7 ...
##  $ C3                : int  11 9 11 11 11 15 11 11 7 11 ...
##  $ C4                : int  14 5 14 14 14 15 14 14 7 14 ...
##  $ C5                : int  10 1 10 10 10 4 10 10 12 10 ...
```

We clustered the data using K means, for 15 clusters using the following command:

```
cl_users <- kmeans(mtx_user, clusters_no,
                   iter.max = 1000, nstart = 5)
```

After that, we used Hypothesis Test to check if different groups have different average rates for the same business. We used a 90% confidence interval to our analysis because we didn't want to impose a very strict test.

Results

We found that the groups are not very well distributed if you do not scale the variables previously the clusterization. We also find out that the clusters are rather stable. You can see the size of the clusters for a sample with 50K users for 5 different trails of *kmeans()*

```
size
```

```
##         C1    C2    C3    C4    C5
## 1       28     7    25    72   965
## 2       18   529   407    23    72
## 3       23  4517     5   532    28
## 4      276     5   208  1795 17587
## 5     1801   965    18   972     5
## 6      152    23  1677    18   276
## 7    17587 23917  4086  4521   529
## 8        7    72    80     5    23
## 9      529    28   835   277  1801
## 10    4517    18    37     7 23917
## 11     965   103 25309   142   152
## 12      72  1801    76    99  4517
## 13   23917 17587   101    28     7
## 14     103   276     7 23920    18
## 15       5   152 17129 17589   103
```

We decided to go without scaling the variables because it may capture better the way this particular society groups.

After clustering, we calculated the *mean*, *standard deviation* and *count* of reviews for each business. We also kept only business with 20 or more reviews for analysis and with at least rates for 2 clusters.

```
# Calculate average, standard deviation and count
resumo <- with(mtx_business,
               aggregate(stars ~ business_id + group,
                         FUN = function(x) c(MN = mean(x), SD = sd(x),
                                             COUNT = length(x))))

# Select only the business with 20 or more review for each group
resumo <- resumo[resumo$stars[,3] >= 20, ]

# Find business that have reviews for at least 2 groups
resumo <- resumo[duplicated(resumo$business_id), ]
```

```r
# Calculate standard error
resumo$SR <- resumo$stars[,2] / resumo$stars[,3]

# Calculate T score for two sided test
system.time({
    resumo$TS <- sapply(resumo$stars[,3],
                                 FUN = function(x) qt(1-alpha/2,df = x))
})
```

Discussion

We didn't hand analyse the users assigned to each group. This could possibly shows if the groups are meaningful in other ways to the YELP business.

We will implement a code to automatically test if the rates for a business that have reviews in more than one group are statistically different.