BACS HW (Week10)

106070038

2021-05-01

Question 1 Let's make an automated recommendation system for the PicCollage mobile app.

a. Let's explore to see if any sticker bundles seem intuitively similar

i. How many recommendations does each bundle have?

- Answer: **six**(iOS version)

ii. Use your intuition to recommend five other bundles in our dataset that might have similar usage patterns as this bundle.

- Answer: **HeartStickerPack** -> Similar usage patterns: supersweet, fallinlovewiththefall, hellobaby, valentineStickers, warmncozy(by intuition) Because all of the stickers above are related to "love"

```
# install.packages("data.table")
library(data.table)
# setwd("/Users/weiwei/Desktop/2021Spring_Courses/BACS/HW8")
ac_bundles_dt <- fread("piccollage_accounts_bundles.csv")
ac_bundles_matrix <- as.matrix(ac_bundles_dt[, -1, with=FALSE])
```

b. Let's find similar bundles using geometric models of similarity

i. Let's create cosine similarity based recommendations for all bundles:

1. Create a matrix or data.frame of the **top 5 recommendations** for all bundles

```
# install.packages("lsa")
# install.packages("SnowballC")
library(SnowballC)
library(lsa)
cos_sim <- cosine(ac_bundles_matrix)
## apply(,1,): by row
cos_sim_add <- apply(cos_sim, 1, mean)
cos_sim_add_rank <- cos_sim_add[order(cos_sim_add, decreasing = TRUE)]
cos_sim_add_rank[1:5]
```

```
##     springrose eastersurprise          bemine     watercolor hipsterholiday
##      0.1578966      0.1459645      0.1383451      0.1375165      0.1368757
```

- Answer: springrose, eastersurprise, bemine, watercolor, hipsterholiday

2. Create a new function that automates the above functionality: it should take an accounts-bundles matrix as a parameter, and return a data object with the top 5 recommendations for each bundle in our data set, using cosine similarity.

```
get_top5 <- function (bundle_name,data) {
  reg1 <- data[bundle_name,]
  reg2 <- reg1[order(reg1, decreasing = TRUE)]
  return (reg2[2:6]) ## top1-5, exclude itself(cos_sim==1)
}
```

3. What are the top 5 recommendations for the bundle you chose to explore earlier?

```
get_top5("HeartStickerPack",cos_sim)
```

```
##      StickerLite            Emome WordsStickerPack  HipsterChicSara
##        0.4256352        0.3870007        0.3834636        0.3292921
## BlingStickerPack
##        0.3181781
```

- Answer: **HeartStickerPack** -> Similar usage patterns: StickerLite, Emome, WordsStickerPack, HipsterChicSara, BlingStickerPack(by caculation) Totally not same as what I guess in a-ii.

ii. Let's create **correlation** based recommendations.

1. Reuse the function you created above (don't change it; don't use the cor() function)
2. But this time give the function an accounts-bundles matrix where each bundle (column) has already been **mean-centered** in advance.
3. Now what are the top 5 recommendations for the bundle you chose to explore earlier?

```
bundle_means <- apply(ac_bundles_matrix, 2, mean)
bundle_means_matrix <- t(replicate(nrow(ac_bundles_matrix), bundle_means))
ac_bundles_mc_b <- ac_bundles_matrix - bundle_means_matrix
row.names(ac_bundles_mc_b) <- row.names(ac_bundles_dt)
cor_sim_2 <- cosine(ac_bundles_mc_b)
get_top5("HeartStickerPack", cor_sim_2)
```

```
##      StickerLite WordsStickerPack            Emome BlingStickerPack
##        0.3870573        0.3832913        0.3714926        0.3203997
##  HipsterChicSara
##        0.3071336
```

```
class(ac_bundles_mc_b)
```

```
## [1] "matrix" "array"
```

- Answer: **HeartStickerPack** -> Similar usage patterns: StickerLite, WordsStickerPack, Emome, BlingStickerPack, HipsterChicSara The results are same as (b), however, the order is not same.

iii. Let's create **adjusted-cosine** based recommendations.

1. Reuse the function you created above (you should not have to change it)
2. But this time give the function an accounts-bundles matrix where each account (row) has already been mean-centered in advance.
3. What are the top 5 recommendations for the bundle you chose to explore earlier?

```
#install.packages("data.table")
library(data.table)
library(lsa)
bundle_means <- apply(ac_bundles_matrix, 1, mean)
bundle_means_matrix <- replicate(ncol(ac_bundles_matrix), bundle_means)
ac_bundles_mc_b <- ac_bundles_matrix - bundle_means_matrix
cor_sim_3 <- cosine(ac_bundles_mc_b)
get_top5("HeartStickerPack", cor_sim_3)
```

```
##      StickerLite            Emome BlingStickerPack  HipsterChicSara
##        0.3974934        0.3311735        0.2865498        0.2726981
## WordsStickerPack
##        0.2594191
```

- Answer: **HeartStickerPack** -> Similar usage patterns: StickerLite, Emome, BlingStickerPack, HipsterChicSara, WordsStickerPack The results are same as (b), however, the order is not same.

c. (not graded) Are the three sets of geometric recommendations similar in nature (theme/keywords) to the recommendations you picked earlier using your intuition alone? What reasons might explain why your computational geometric recommendation models produce different results from your intuition?

- Answer: No, because I just guess it by literal meaning, but we should calculate the similarity. ## d. (not graded) What do you

think is the conceptual difference in cosine similarity, correlation, and adjusted-cosine?

- Answer:
  - **Cosine similarity** defined as the angular similarity between two vectors. angle 0 defines match and 90 otherwise.
  - **correlation coefficient** defined as the covariance between two vectors divided by their standard deviations.
  - **adjusted-cosine** is very much similar to pearson similarity except if two are calculated over different set of rated vectors.

Question 2 Correlation is at the heart of many data analytic methods so let's explore it further.

a. Create a horizontal set of random points, with a relatively narrow but flat distribution.

i. What raw slope of x and y would you generally expect? ii.What is the correlation of x and y that you would generally expect? ## b. Create a completely random set of points to fill the entire plotting area, along both x-axis and y-axis i.What raw slope of the x and y would you generally expect?

ii. What is the correlation of x and y that you would generally expect? ## c. Create a diagonal set of random points trending upwards at 45 degrees

iii. What raw slope of the x and y would you generally expect? (note that x, y have the same scale)

iv. What is the correlation of x and y that you would generally expect? ## d. Create a diagonal set of random trending downwards at 45 degrees

v. What raw slope of the x and y would you generally expect? (note that x, y have the same scale)

vi. What is the correlation of x and y that you would generally expect? ## e. Apart from any of the above scenarios, find another pattern of data points with no correlation (r 0). (optionally: can create a pattern that visually suggests a strong relationship but produces r 0?) ## f. Apart from any of the above scenarios, find another pattern of data points with perfect correlation (r 1). (optionally: can you find a scenario where the pattern visually suggests a different relationship?) ## g. Let's see how correlation relates to simple regression, by simulating any linear relationship you wish:

vii. Run the simulation and record the points you create: pts <- interactive_regression()

viii. Use the lm() function to estimate the regression intercept and

slope of pts to ensure they are the same as the values reported in the simulation plot: summary(lm(pts$y$ ptsx))

ix. Estimate the correlation of x and y to see it is the same as reported in the plot: cor(pts)

x. Now, re-estimate the regression using standardized values of both x and y from pts

xi. What is the relationship between correlation and the standardized simple-regression estimates?