BUS 41201 Homework 6 Assignment

Group 24: Shihan Ban, Yi Cao, Shri Lekkala, Ningxin Zhang

7 May 2024

Introduction: Congressional Speech

textir contains congress109 data: counts for 1k phrases used by each of 529 members of the 109th US congress.
Load it with data(congress109).
See ?congress109.

The counts are in congress109Counts.

We also have congress109Ideology, a data.frame containing some information about each speaker.

The includes some partisan metrics: - party (Republican, Democrat, or Independent)
- repshare: share of constituents voting for Bush in 2004
- Common Scores [cs1,cs2]: basically, the first two principal components of roll-call votes

No starter script; look at we8there.R and wine.R.

```
library(textir)
```

```
## Loading required package: distrom

## Loading required package: Matrix

## Loading required package: gamlr

## Loading required package: parallel
```

```
# Load the congress109 data
data(congress109)

speech_data = congress109Counts
ideology_data = congress109Ideology
```

Question 1

Fit K-means to speech text for K in 5,10,15,20,25.

```
# scale the data
speech_data_scaled = 100*speech_data/rowSums(speech_data)

# store results in list
k_values = c(5, 10, 15, 20, 25)
clustering_results = list()

for (k in k_values) {
  clustering_results[[as.character(k)]] = kmeans(speech_data_scaled,
                                                  centers = k,
                                                  nstart = 10)
}
```

Use BIC to choose the K and interpret the selected model.

```
# load the kIC function
source("kIC.R")

# store bic_values
bic_values = list()

for (k in k_values) {
  kfit = clustering_results[[as.character(k)]]
  bic_values[[as.character(k)]] = kIC(kfit, rule="B")  # Use BIC calculation
}

data.frame("BIC" = unlist(bic_values))
```

```
##          BIC
## 5  323441.8
## 10 328607.7
## 15 338358.7
## 20 351248.8
## 25 367095.6
```

```
best_k = k_values[which.min(bic_values)]
```

So the best k which minimizes the BIC out of these is k = 5 clusters.

```
best_fit = clustering_results[[as.character(best_k)]]

# size of each cluster
best_fit$size
```

```
## [1]   1 509   1  17   1
```

```
# print clusters of size 1
for (i in which(best_fit$size == 1)){
  print(best_fit$cluster[best_fit$cluster == i])
}
```

```
## Gary Ackerman
##            1
## Ernest Istook
##            3
## Michael Doyle
##            5
```

We notice that there are only 3 out of the 5 clusters that are not singleton sets. And notably cluster 3 has by far the largest number of elements with a size of 442.

The two singleton clusters suggest that these may be outliers, for congressman with unique speech patterns or extreme views that are not typical of others in the dataset. These are "Michael Doyle" and "Gary Ackerman"

The dominant cluster of 442 suggests that there is a large commonality in speech patterns amongst the majority of the congressmen, which might be the "average" behavior.

Question 2

Fit a topic model for the speech counts. Use Bayes factors to choose the number of topics, and interpret your chosen model.

```
library(maptpx)
```

```
## Loading required package: slam
```

```
## Convert speech counts from a Matrix to a `slam' simple_triplet_matrix
x_speech = as.simple_triplet_matrix(speech_data_scaled)

## Supply a vector of topic sizes, and it uses a Bayes factor to choose
## The algorithm stops if BF drops twice in a row

tpcs = topics(x_speech, K = 5*(1:5), verb = 1)
```

```
##
## Estimating on a 529 document collection.
## Fit and Bayes Factor Estimation for K = 5 ... 25
## log posterior increase: 1325.9, 286.1, 166.4, 105.5, 65.8, 94.9, 13.6, 12.5, 20.1, 14, 2.1, 2.8, 0.7
## log BF( 5 ) = 14549.05
## log posterior increase: 2236, 99.2, 47.3, 22.8, 69.8, 50.3, 11.2, 9.3, 5.5, 3, 0.8, 0.4, 0.7, 0.3, 3
## log BF( 10 ) = 10424.16
## log posterior increase: 1505.4, 85.8, 45.2, 15.1, 6.2, 5.4, 14.2, 7.7, 36.8, 18.3, 4.3, 5.7, 2.8, 2.
## log BF( 15 ) = -795.19
```

So for each K in (5, 10, 15, 20, 25), a topics model model was fitted and K = 5 is chosen as it has the highest Bayes Factor (analogous to lowest BIC).

```
summary(tpcs)
```

```
##
## Top 5 phrases by topic-over-null term lift (and usage %):
##
## [1] 'violent.sexual.predator', 'world.poorest.people', 'united.airline.employe', 'tax.break.wealthy'
## [2] 'near.retirement.age', 'nunn.lugar.program', 'ayman.al.zawahiri', 'nation.oil.food', 'driver.edu
## [3] 'little.rock.nine', 'minority.women.owned', 'tuskege.airmen', 'illegal.alien', 'little.rock' (17
## [4] 'support.boy.scout', 'national.flood.insurance', 'flood.insurance.program', 'domestic.violence',
## [5] 'pluripotent.stem.cel', 'produce.stem.cel', 'republic.cypru', 'commonly.prescribed.drug', 'natio
##
## Log Bayes factor and estimated dispersion, by number of topics:
##
##              5       10      15
## logBF 14549.05 10424.16 -795.19
## Disp      2.32     1.80    1.60
##
## Selected the K = 5 topic model
```

```r
# Also look at words ordered by simple in-topic prob
# topic-term probability matrix is called 'theta'

# Rank terms by probability within topics

# Number of topics in the model
num_topics = dim(tpcs$theta)[2]
top_words_by_topic = list()

# Loop through each topic to get the top 10 words
for (i in 1:num_topics) {
    top_words = rownames(tpcs$theta)[order(tpcs$theta[,i], decreasing = TRUE)[1:10]]
    top_words_by_topic[[i]] = top_words
}

# Convert the list to a dataframe
topics_dataframe = data.frame(
    Topic = 1:num_topics,
    Words = I(top_words_by_topic)
)

print(topics_dataframe$Words)
```

```
## [[1]]
##  [1] "american.people"     "trade.agreement"     "low.income"
##  [4] "middle.class"        "fre.trade"           "public.broadcasting"
##  [7] "war.iraq"            "central.american"    "million.american"
## [10] "hurricane.katrina"
##
## [[2]]
##  [1] "american.people" "war.terror"      "class.action"    "saddam.hussein"
##  [5] "nuclear.weapon"  "death.tax"       "iraqi.people"    "chief.justice"
##  [9] "war.terrorism"   "billion.dollar"
##
## [[3]]
##  [1] "african.american"    "civil.right"         "head.start"
##  [4] "rosa.park"           "postal.service"      "illegal.alien"
##  [7] "illegal.immigration" "border.security"     "post.office"
## [10] "strong.support"
##
## [[4]]
##  [1] "strong.support"    "appropriation.bil" "hurricane.katrina"
##  [4] "domestic.violence" "pass.bil"          "gulf.coast"
##  [7] "look.forward"      "terri.schiavo"     "urge.support"
## [10] "natural.disaster"
##
## [[5]]
##  [1] "stem.cel"           "natural.ga"         "prescription.drug"
##  [4] "cel.research"       "look.forward"       "embryonic.stem"
##  [7] "embryonic.stem.cel" "american.people"    "million.american"
## [10] "foreign.oil"
```

For each topic, we can examine the top phrases from the summary above:

- Topic 1: appears to focus on issues relating to crime and wealth inequality as suggested by the phrases “violent.sexual.predator”, “world.poorest.people”, and “tax.break.wealthy”. It is notable that top 5 phrases for this has a high topic-over-null lift of 30.2%, which indicates how much more likely these phrases are likely to appear in this topic compared to the whole dataset.

- Topic 2: seems to capturing political discussions regarding terrorism and national security. Whilst this isn’t immediately apparent from the top 5 phrases list, it becomes more apparent when looking at the list of most probabilistic words within each topic, with words such as “war.terror”, “saddam.hussein”, and “nuclear.weapon”.

- Topic 3: captures issues and discussions relating to civil rights, racism, and minorities as suggested by the phrases ’little.rock.nine’, and ’minority.women.owned’. This is also apparent when examining the top 10 probabilistic words for this topic.

- Topic 4: includes phrases related to natural disasters and national issues as suggested by ’flood.insurance.program’, and ’domestic.violence’.

- Topic 5: seems to capture discussions regarding scientific research and medical topics as evidenced by the repeated phrase “stem.cel”, as well as “commonly.prescribed.drug”.

So the chosen model with 5 topics had a high log Bayes factor of 14549.05 which indicates strong support for this mode. Further, each topic seems to capture a different set of themes with a clear focus on issues such as public policy, science, national security, and social issues.

Question 3

Connect the unsupervised clusters to partisanship. Tabulate party membership by K-means cluster. Are there any non-partisan topics?

Fit topic regressions for each of party and repshare. Compare to regression onto phrase percentages:

```
# x<-100*congress109Counts/rowSums(congress109Counts)
```