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.

```r
library(textir)
library(dplyr)
library(tidyr)
library(knitr)

# 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.

```r
set.seed(1024)

# scale the data
speech_data_scaled = scale(as.matrix(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.

```r
# 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  540024.6
## 10 556111.9
## 15 573661.9
## 20 592780.0
## 25 612500.1
```

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

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

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

# size of each cluster
best_fit$size
```

```
## [1] 125 336  13   1  54
```

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

```
## Carolyn McCarthy
##                4
```

We notice that there are only 3 out of the 5 clusters that are not very small clusters. And notably cluster 2 has by far the largest number of elements with a size of 336.

The one singleton clusters suggests that this may be an outlier, for congressman with unique speech patterns or extreme views that are not typical of others in the dataset. This is “Carolyn McCarthy”.

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)

## 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: 5665.1, 1459.1, 442.5, 97.3, 51.5, 19.3, 7.9, 13.5, 14.4, 4.9, 1.6, 1, 2, 4,
## log BF( 5 ) = 59601.98
## log posterior increase: 4935.3, 278.5, 143, 58.4, 42.6, 55.5, 74.9, 86, 40.4, 7.7, 5.5, 6.4, 18.5, 8
## log BF( 10 ) = 76681.45
## log posterior increase: 3171.8, 220, 149.2, 46.2, 21.7, 12.6, 8.5, 9, 15.7, 17.8, 20.7, 17.1, 6.9, 4
## log BF( 15 ) = 75751.68
## log posterior increase: 2016.2, 186.5, 102.8, 124.5, 47.9, 44.4, 34.2, 45.8, 13, 11.7, 12.9, 7, 6.7,
## log BF( 20 ) = 66374.16
```

So for each K in (5, 10, 15, 20, 25), a topics model model was fitted and K = 10 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] ’national.heritage.corridor’, ’ryan.white.care’, ’violence.sexual.assault’, ’white.care.act’, ’de
## [2] ’southeast.texa’, ’commonly.prescribed.drug’, ’ready.mixed.concrete’, ’million.illegal.alien’, ’a
## [3] ’near.retirement.age’, ’increase.taxe’, ’personal.retirement.account’, ’medic.liability.reform’,
## [4] ’winning.war.iraq’, ’near.earth.object’, ’troop.bring.home’, ’bless.america’, ’nunn.lugar.program
## [5] ’united.airline.employe’, ’record.budget.deficit’, ’student.loan.cut’, ’private.account’, ’privat
## [6] ’republic.cypru’, ’hate.crime.legislation’, ’change.heart.mind’, ’driver.education’, ’va.health.
## [7] ’hearing.scheduled’, ’witness.testify’, ’circuit.judge’, ’business.meeting’, ’judge.alberto.gonza
## [8] ’able.buy.gun’, ’western.energy.crisi’, ’credit.card.industry’, ’caliber.sniper.rifle’, ’wild.bi
## [9] ’pluripotent.stem.cel’, ’low.cost.reliable’, ’national.ad.campaign’, ’cel.stem.cel’, ’regional.t
## [10] ’american.fre.trade’, ’central.american.fre’, ’north.american.fre’, ’financial.accounting.standa
##
## Log Bayes factor and estimated dispersion, by number of topics:
##
##               5        10        15        20
## logBF 59601.98 76681.45 75751.68 66374.16
## Disp      3.71     2.84     2.45     2.20
##
## Selected the K = 10 topic model
```

The summary above shows us the top 5 phrases in each topic that have a high topic-over-null-lift, which indicates how much more likely these phrases are to appear in this topic than the whole dataset.

In addition, for further clarity, we can examine the list of top 10 probabilistic words within each topic to aid our interpretation:

```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] "african.american"  "civil.right"       "domestic.violence"
##  [4] "head.start"        "rosa.park"         "hurricane.katrina"
##  [7] "gulf.coast"        "strong.support"    "violence.women"
## [10] "american.people"
##
## [[2]]
##  [1] "postal.service"      "illegal.alien"       "private.property"
##  [4] "illegal.immigration" "border.security"     "class.action"
##  [7] "driver.license"      "strong.support"      "border.patrol"
## [10] "post.office"
##
## [[3]]
##  [1] "tax.relief"        "american.people"   "death.tax"
##  [4] "economic.growth"   "finance.committe"  "business.owner"
##  [7] "tax.increase"      "prescription.drug" "budget.committe"
## [10] "security.system"
##
## [[4]]
##  [1] "american.people"  "iraqi.people"     "saddam.hussein"   "war.iraq"
##  [5] "national.guard"   "iraq.afghanistan" "war.terror"       "nuclear.weapon"
##  [9] "war.terrorism"    "god.bless"
## 
## [[5]]
##  [1] "american.people"       "private.account"       "hurricane.katrina"
##  [4] "middle.class"          "prescription.drug"     "national.debt"
##  [7] "student.loan"          "security.trust"        "social.security.trust"
## [10] "billion.dollar"
## 
## [[6]]
##  [1] "appropriation.bil"   "low.income"          "veteran.health"
##  [4] "hate.crime"          "veteran.health.care" "president.budget"
##  [7] "prescription.drug"   "food.stamp"          "iraq.afghanistan"
## [10] "look.forward"
## 
## [[7]]
##  [1] "appropriation.bil" "american.people"   "class.action"
##  [4] "judicial.nomine"   "chief.justice"     "circuit.court"
##  [7] "judge.robert"      "court.appeal"      "democratic.leader"
## [10] "look.forward"
## 
## [[8]]
##  [1] "minimum.wage"      "credit.card"       "american.people"
##  [4] "foreign.oil"       "wildlife.refuge"   "civil.right"
##  [7] "nuclear.weapon"    "low.income"        "prescription.drug"
## [10] "hurricane.katrina"
## 
## [[9]]
##  [1] "stem.cel"           "natural.ga"         "cel.research"
##  [4] "embryonic.stem"     "embryonic.stem.cel" "cord.blood"
##  [7] "adult.stem"         "adult.stem.cel"     "cel.line"
## [10] "stem.cel.line"
## 
## [[10]]
##  [1] "trade.agreement"      "fre.trade"            "central.american"
##  [4] "trade.deficit"        "american.fre.trade"   "central.american.fre"
##  [7] "american.worker"      "manufacturing.job"    "trade.policy"
## [10] "american.people"
```

For each topic, we can examine the top phrases as well as the list of most probabilistic words within each topic. After looking at this we can broadly classify each topic that captures discussions regarding certain themes:

- Topic 1: Civil rights, healthcare reforms, and societal issues
- Topic 2: Immigration, and homeland security
- Topic 3: Retirement planning, and fiscal policy
- Topic 4: Foreign policy, and national security
- Topic 5: Social security, and personal finance
- Topic 6: Wealth inequality, and welfare of veterans
- Topic 7: Court proceedings, and hearings

- Topic 8: Socio-economic issues, and environmental topics
- Topic 9: Medical advancements, and scientific research
- Topic 10: Trade policies, and economic models

So the chosen model with 10 topics had a high log Bayes factor of 76681.45 which indicates strong support for this model. 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?

```
# Merge Clustering Results with Party Data
ideology_data$cluster = best_fit$cluster
```

```
# Create a summary table
party_cluster_tabulation = ideology_data |>
  group_by(cluster, party) |>
  summarise(count = n(), .groups = 'drop') |>
  spread(key = party, value = count, fill = 0)

kable(party_cluster_tabulation)
```

cluster	D	I	R
1	124	1	0
2	112	1	223
3	1	0	12
4	1	0	0
5	4	0	50

Thus it appears that cluster 2 appears to be a non partisan topic as it has large members in both D and R. It is difficult to say whether clusters 3 and 4 are non partisan topics as there is insufficient data about party memberships for these topics to form a definitive opinion.
On the otherhand clusters 1 and 5 appear to be clearly partisan topics with the former being dominated by party D and the latter by party R.

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

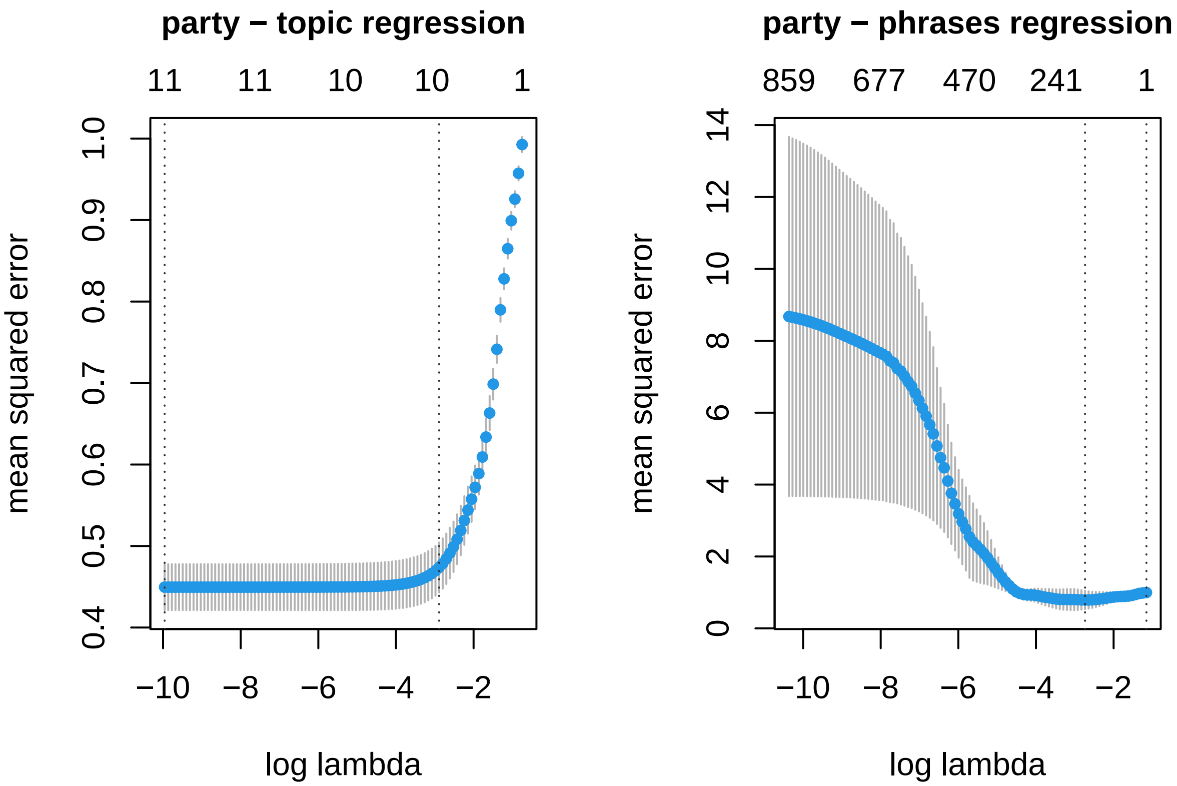
```
x = 100*congress109Counts/rowSums(congress109Counts)

# Fit topic regressions onto party

regtopics_party = cv.gamlr(tpcs$omega, ideology_data$party, lambda.min.ratio=10^{-4})
regphrases_party =cv.gamlr(x, ideology_data$party, lambda.min.ratio=10^{-4})

par(mfrow=c(1,2))
plot(regtopics_party)
mtext("party - topic regression", font=2, line=2)

plot(regphrases_party)
mtext("party - phrases regression", font=2, line=2)
```



```
min(regtopics_party$cvm)
```

```
## [1] 0.4495572
```

```
min(regphrases_party$cvm)
```

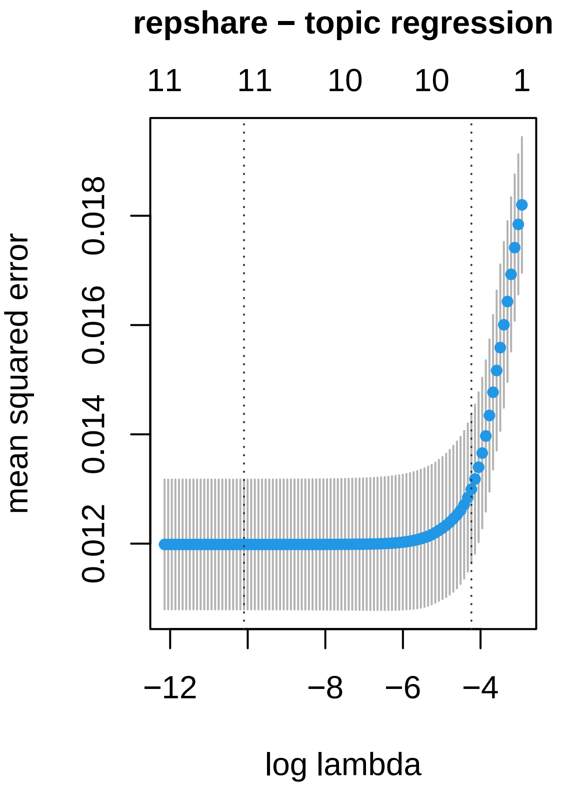
```
## [1] 0.7897052
```

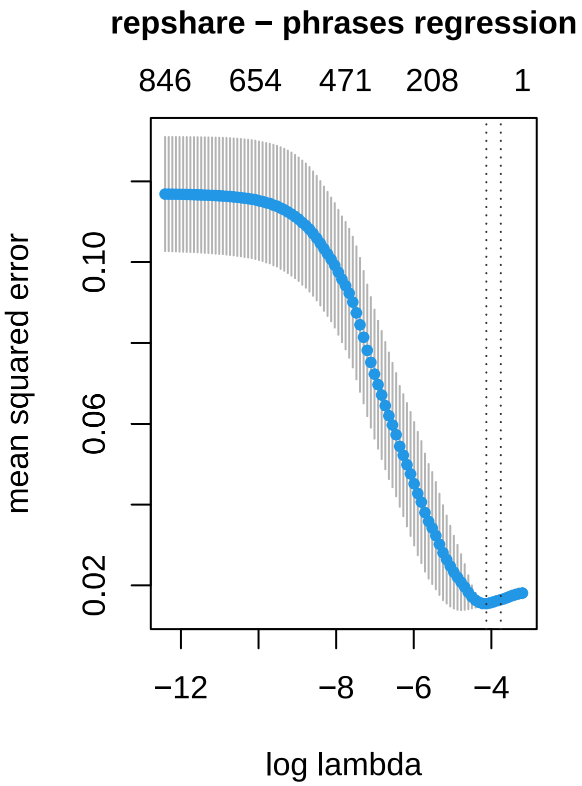
```
# Fit topic regressions onto repshare

regtopics_repshare = cv.gamlr(tpcs$omega, ideology_data$repshare, lambda.min.ratio=10^{-4})
regphrases_repshare = cv.gamlr(x, ideology_data$repshare, lambda.min.ratio=10^{-4})

par(mfrow=c(1,2))
plot(regtopics_repshare)
mtext("repshare - topic regression", font=2, line=2)

plot(regphrases_repshare)
mtext("repshare - phrases regression", font=2, line=2)
```





```r
min(regtopics_repshare$cvm)
```

```
## [1] 0.01198514
```

```r
min(regphrases_repshare$cvm)
```

```
## [1] 0.01548518
```

In both cases we observe that the topic model performs better than regression on to phrase percentages.

For topic regression onto party, the minimum out of sample MSE was ≈ 0.45 vs ≈ 0.82 for standard regression.

For regressions onto repshare, the minimum out of sample MSE for topics was still lower (≈ 0.012 vs ≈ 0.015), but the margins were closer.