GOV 51 Section

Week 9: Unsupervised Text Analysis

Pranav Moudgalya

Harvard College

Agenda

- Housekeeping
- Unsupervised Learning Overview
- Implementation

Housekeeping

- Reminder of deadlines
- April 11th $\rightarrow$ Initial result due (one page write up). (10%)
- April 18th $\rightarrow$ First draft of poster due.
- April 24th $\rightarrow$ Final draft of poster due.
- April 29th $\rightarrow$ Poster session (usual lecture time) (**no extensions!**)

Terms and Things

- Corpus - a collection of texts that we want to analyze
  - Complete works of Charlotte Bronte, newspaper articles from the AP
- Document - a unit within the corpus
  - Jane Eyre; an article from the AP

Unsupervised Learning Example

- Lot's of criticism of the peer review model - send your article in to two double blind reviewers
  - Despite double blind, questions about anonymity - people post their papers online now
  - Reviewers can be quite biased as well!
- Two questions arise

1. Who is publishing in top political science journals? Are their ascriptive disparities?
2. What is being published? Are some topics avoided?

Unsupervised Learning Example

- Lot's of criticism of the peer review model - send your article in to two double blind reviewers
  - Despite double blind, questions about anonymity - people post their papers online now
  - Reviewers can be quite biased as well!
- Two questions arise

1. Who is publishing in top political science journals? Are their ascriptive disparities?
2. **What is being published? Are some topics avoided?**

Topic Modeling

- One advantage of unsupervised learning is that pattern finding
  - Data could be high-dimensional, messy, huge observations (e.g. Jane Eyre) ...
- Patterns within the data are incredibly useful!
  - For example, the topics found in a flagship political science journal
- One application is **topic modeling**

Latent Dirichlet Allocation (LDA)

- When we topic model, we need to make some assumptions about how text is generated
  - Recall basic assumptions about topics $\rightarrow$ probability distribution of words
- Latent Dirichlet Allocation is just one model that we can use to topic model
  - LDA relies on a similar assumption of the data generating process of text

1. For each topic, draw a topic-word distribution

Latent Dirichlet Allocation (LDA)

- When we topic model, we need to make some assumptions about how text is generated
  - Recall basic assumptions about topics $\rightarrow$ probability distribution of words
- Latent Dirichlet Allocation is just one model that we can use to topic model
  - LDA relies on a similar assumption of the data generating process of text

1. For each topic, draw a topic-word distribution $\rightarrow$ how likely is a topic given a word?

Latent Dirichlet Allocation (LDA)

- When we topic model, we need to make some assumptions about how text is generated
  - Recall basic assumptions about topics $\rightarrow$ probability distribution of words
- Latent Dirichlet Allocation is just one model that we can use to topic model
  - LDA relies on a similar assumption of the data generating process of text

1. For each topic, draw a topic-word distribution $\rightarrow$ how likely is a topic given a word?
2. For each document, draw a document-topic distribution

Latent Dirichlet Allocation (LDA)

- When we topic model, we need to make some assumptions about how text is generated
  - Recall basic assumptions about topics $\rightarrow$ probability distribution of words
- Latent Dirichlet Allocation is just one model that we can use to topic model
  - LDA relies on a similar assumption of the data generating process of text

1. For each topic, draw a topic-word distribution $\rightarrow$ how likely is a topic given a word?
2. For each document, draw a document-topic distribution $\rightarrow$ how likely is a document about certain topics?

Lecture Example

I had donuts this morning. I don't have diabetes yet.

- LDA is based on the following generative process:
    1. For each topic, draw a topic-word distribution.

       food: donuts 0.7, have 0.3, diabetes 0

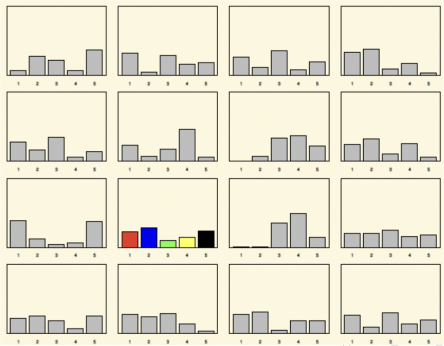
       Health: Diabetes 0.7, have 0.3, donuts 0
    2. For each document , draw a document-topic distribution.

       Sentence 1: food 0.999, health 0.001

       Sentence 2: food 0.001, health 0.999

LDA Visualization

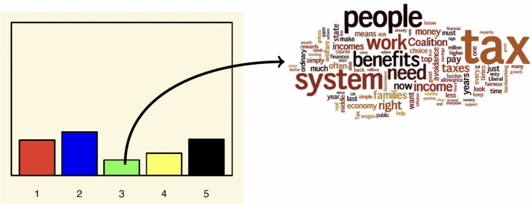
First, for each **document**, randomly choose a distribution over topics. Each of the distributions mixes the topics in different proportions:



LDA Visualization

Next, for **every word in that document**...

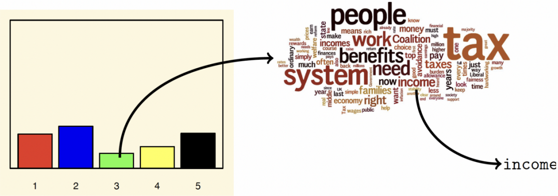
- Randomly choose a topic from the distribution over topics from the previous step.



LDA Visualization

Next, for **every word in that document**...

- Randomly choose a topic from the distribution over topics from the previous step.
- Randomly choose a word from the distribution over the vocabulary that the chosen topic implies



Installation

- `topicmodels` package is useful for implementing a LDA model
- In our example here, we will be taking a look at a sample of articles in the AP in 1992

```r
library(topicmodels)
data("AssociatedPress")
```

Preview

- Data is at the document-word level - for each document, each unique word is counted

```r
head(tidy(AssociatedPress))
```

```
## # A tibble: 6 x 3
##   document term      count
##      <int> <chr>     <dbl>
## 1        1 adding        1
## 2        1 adult         2
## 3        1 ago           1
## 4        1 alcohol       1
## 5        1 allegedly     1
## 6        1 allen         1
```

Function

- `k`: number of topics we want to specify
- `control`: setting a seed because probability distributions entail randomness

```
# set a seed so that the output of the model is predictabl
ap_lda <- LDA(AssociatedPress,
              k = 2,
              control = list(seed = 02138))
ap_lda
```

```
## A LDA_VEM topic model with 2 topics.
```

Results

- `beta` refers to the probability that a given word is related to a topic
- Let's see which topic "harvard" is associated with

```r
ap_topics <- tidy(ap_lda,
                  matrix = "beta")

ap_topics |>
  filter(term == "harvard")
```

```
## # A tibble: 2 x 3
##   topic term         beta
##   <int> <chr>       <dbl>
## 1     1 harvard 0.0000402
## 2     2 harvard 0.000120
```

Visualization Prep

- Instead of looking for specific words, let's visualize the most likely terms per topic

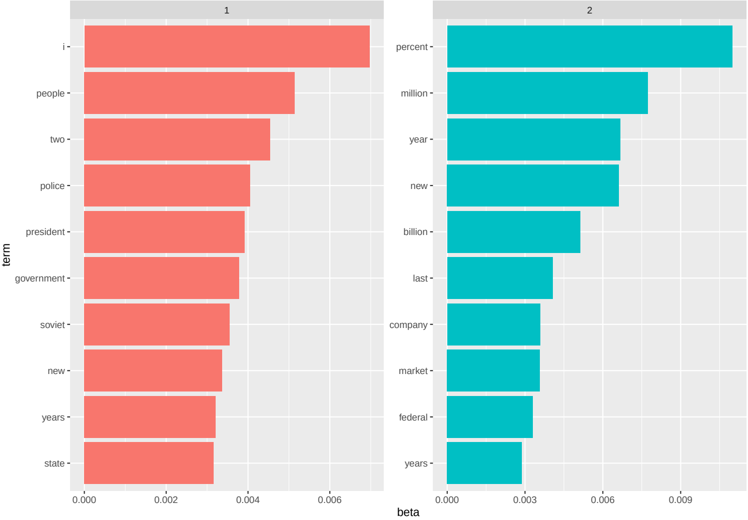
```r
top_terms <- ap_topics |>
  group_by(topic) |>
  slice_max(beta, n = 10) |>
  ungroup() |>
  arrange(topic, -beta)

top_terms <- top_terms  |>
  mutate(term = reorder_within(term, beta, topic))
```

Visualization

```r
ggplot(top_terms, aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  scale_y_reordered()
```

Visualization



Document Level Visualization

- How about topic likelihoods at the document level?
- gamma gives us the likelihood of a topic given the words in a document

```
ap_documents <- tidy(ap_lda, matrix = "gamma") |>
  arrange(document, topic)
head(ap_documents)
```

```
## # A tibble: 6 x 3
##   document topic    gamma
##      <int> <int>    <dbl>
## 1        1     1 0.999
## 2        1     2 0.000677
## 3        2     1 0.514
## 4        2     2 0.486
## 5        3     1 0.962
## 6        3     2 0.0380
```

So, the motivating question

1. Who is publishing in top political science journals? Are their ascriptive disparities?
2. **What is being published? Are some topics avoided?**

So, the motivating question

1. Who is publishing in top political science journals? Are their ascriptive disparities?
2. **What is being published? Are some topics avoided?**

Saraceno (2020) did an analysis of publications in The Journal of Politics

Back to Saraceno (2020)

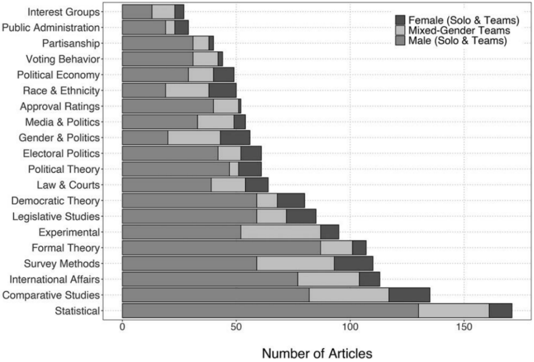


Figure 5. Topic by authorship type, 2000–2019. Bars represent the number of articles published from each topic. Each bar is shaded using the proportion of articles written exclusively by females, exclusively by males, and by mixed-gender teams.

Saraceno (2022)

Table 2. Topic Model Output

Topic Label	High-Frequency Words
Interest Groups	Groups, interest, members, organizations, activity, influence
Political Theory	Man, life, human, nature, thought, Hobbes, philosophy
Gender and Politics	Women, social, participation, gender, female, men, politics
Political Economy	Economic, income, fiscal, growth, welfare, market, inequality
Race and Ethnicity	Black, white, racial, ethnic, school, minority, representation
Democratic Theory	Rights, democratic, moral, public, liberal, justice, freedom
The South	States, southern, county, Negro, governor, Virginia, Carolina
Electoral Politics	Elections, candidates, district, incumbent, office, primary
Survey Methods	Respondents, information, survey, attitudes, treatment
Postwar Politics	Soviet, war, world, national, social, united, communist
International Affairs	Conflict, foreign, military, international, war, aid, civil
Judicial Politics	Court, supreme, cases, judicial, law, courts, justice, legal
Formal Theory	Model, decision, equilibrium, preferences, choice, probability
Public Administration	Local, government, service, agencies, city, administer, board
Statistical	Models, data, variables, effect, results, analysis, significant
Legislative Politics	Congress, House, legislative, committee, Senate, majority, bill
Comparative Politics	Countries, international, institution, regime, corrupt, Latin
Partisanship	Issues, ideological, opinion, liberal, conservative, polarization
Media and Politics	Media, coverage, exposure, negative, television, advertising
Voting Behavior	Voter, campaign, candidate , turnout, electorate, ballot
Approval Ratings	President, economic, support, approval, performance
Experimental Methods	Experimental, subjects, control, condition, assigned, effect

Topic Modelling, LDA, and Notes

- Reiterating caveats at the end of lecture
- Topic modelling is **not** a panacea!
  - Similar to other methods, it relies on assumptions, particularly about the DGP of text
  - Uncertainty is also a part of predictions - similar in respect to regression predictions which have their own standard errors

Packages and Preamble

```
library(tm)
library(SnowballC)
```

- In what ways can we categorize and divide the Harvard Government faculty?
- Let's say we have a **corpus** with three variables in `.csv` form

1. `prof` - name of faculty member
2. `phd` - year of phd attainment
3. `bio` - biography on website

Pre-pre-processing

```r
# Loading in the data
df <- read.csv("data/harvardgov.csv")

# Converting the .csv to document term matrix form
corpus <- Corpus(VectorSource(df$bio))
```

Pre-processing

```
# make everything lowercase
corpus <- tm_map(corpus, content_transformer(tolower))

# remove white space (e.g. spaces)
corpus <- tm_map(corpus, stripWhitespace)

# remove numbers
corpus <- tm_map(corpus, removeNumbers)

# remove stopwords
corpus <- tm_map(corpus, removeWords, stopwords("english"))

# stem words (e.g. remove "ing")
corpus <- tm_map(corpus, stemDocument)
```

Conversion to DtM

```r
# Turning into a document term matrix
dtm <- DocumentTermMatrix(corpus)
dtm.mat <- as.matrix(dtm)

# Adding labels to each document
rownames(dtm.mat) <- df$prof
```

```r
# Normalize by document size
tfidf <- weightTfIdf(dtm, normalize = TRUE)
tfidf.mat <- as.matrix(tfidf, normalize = TRUE)

# Adding labels to each document
rownames(tfidf.mat) <- df$prof
```

Visualizing

```r
par(cex = 1.25)
library("wordcloud")
```

```
## Loading required package: RColorBrewer
```

```r
liu <- dtm.mat["naijia liu",]
liu.tfidf <- tfidf.mat["naijia liu",]
```

Visualizing Example

```
wordcloud(colnames(dtm.mat), liu,
          min.freq = min(liu[liu > 0]))
```

Descriptive Stats

```r
sort(liu, decreasing = TRUE)[1:5]
```

```
##      polit  research   current    includ professor
##          2         2         1         1         1
```

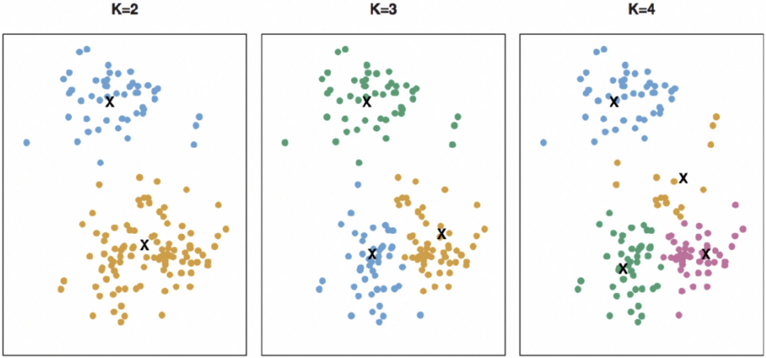
```r
sort(liu.tfidf, decreasing = T)[1:3]
```

```
##  demographi    facebook imputation,
##   0.1745301   0.1745301   0.1745301
```

K-Means Algorithm

- K-Means clustering is simply an exercise in partitioning n observations into k clusters
  - In a text context, this means comparing the similarity of words between documents
- Here, we are looking to cluster professors based on similar word usage in their bios!
- This is an iterative process - the algorithm does initial groupings, then sees whether it can minimize error by another permutation

K-Means Algorithm



K-Means Algorithm

```
# Need to standardize so that each row sums to a unit leng
tfidf.unit <- tfidf.mat / sqrt(rowSums(tfidf.mat^2))

set.seed(1234)
# centers indicates the number of clusters (e.g. k)
kconfour.out <- kmeans(tfidf.unit,
                       centers = 5)
```

Descriptive Stats

```r
table(kconfour.out$cluster)
```

```
##
##  1  2  3  4  5
## 11 14  5  7 11
```

K-Means Group 1

```
knitr::kable(df$prof[kconfour.out$cluster == 1])
```

x
stephen ansolabehere
nara dillon
grzegorz ekiert
peter a. hall
jennifer hochschild
alastair iain johnston
taeku lee
elizabeth j. perry
michael rosen
james m. snyder, jr
richard tuck

K-Means Group 2

```r
knitr::kable(df$prof[kconfour.out$cluster == 2])
```

x
danielle allen
eric beerbohm
daniel carpenter
timothy colton
katrina forrester
claudine gay
harvey c. mansfield
eric nelson
paul peterson
stephen peter rosen
michael sandel
theda skocpol
latanya sweeney
daniel ziblatt

K-Means Group 3

```
knitr::kable(df$prof[kconfour.out$cluster == 3])
```

x
jeffry frieden
frances hagopian
alisha c. holland
torben iversen
steven levitsky

K-Means Group 4

```
knitr::kable(df$prof[kconfour.out$cluster == 4])
```

x
melani cammett
stephen chaudoin
christina davis
joshua d. kertzer
christoph mikulaschek
pia raffler
yuhua wang

K-Means Group 5

```
knitr::kable(df$prof[kconfour.out$cluster == 5])
```

x
matthew blackwell
peter buisseret
ryan enos
chase h. harrison
michael j. hiscox
kosuke imai
gary king
naijia liu
mashail malik
stephanie ternullo
dustin tingley

Overtime Comparisons

- How similar are each bio to the professor with the earliest PhD, Harvey Mansfield?

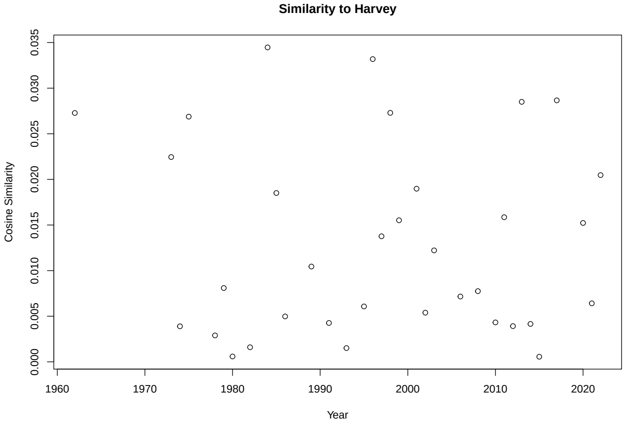
```
# Isolate Harvey
harvey <- as.data.frame(tfidf.mat["harvey c. mansfield",])
# Isolate non-Harveys
nonhm.tfidf <- as.data.frame(tfidf.mat[rownames(tfidf.mat)
                                 != "harvey c. mansfield", ])
# Sort everything chronologically by PhD attainment
nonhm.tfidf$year.index <-
    df$phd[df$prof!= "harvey c. mansfield"]
chron.tfidf <- nonhm.tfidf[order(nonhm.tfidf$year.index),]
years <- sort(unique(df$phd))
years <- years[-1]
```

For-Loop

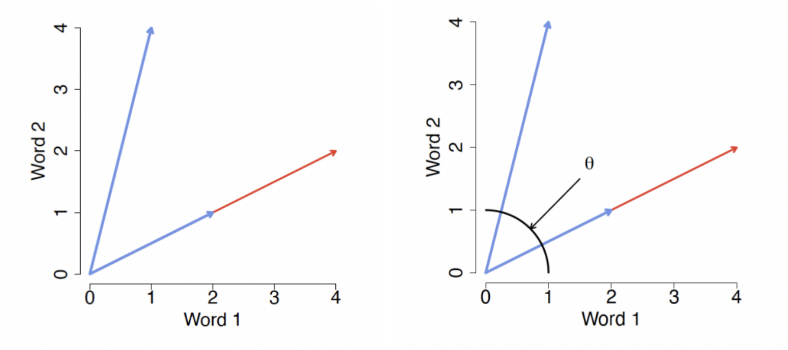
```r
avg.cosim <- rep(NA, length(years))
for (i in 1:length(years)) {
    decade <- subset(chron.tfidf,
                     (year.index == years[i]))
    decade <- decade[, names(decade) != "year.index"]
    similarity <- cosine(harvey, decade)
  avg.cosim[i] <- mean(similarity)
}
```

Plotting Similarity to Harvey Across Time

```r
plot(years, avg.cosim, main = "Similarity to Harvey",
     xlab = "Year", ylab = "Cosine Similarity")
```



Reminder: Cosine similarity vs. Euclidean length



Summary

- Pre-processing text in an easy-to-implement way
  - Also, pre-pre-processing when our data isn't already a document term matrix
- Learned one way to group text based on similarity (e.g. k-means algorithm)
- Using for-loops and our own cosine similarity function, we can plot similarity over time

Office Hours

- Bring all your questions!
- Happy to help on code, identification, etc!
- Happy to talk about final projects!