Wrapping up

Gov 51: Section 10

Sima Biondi

Spring 2025

Overview

1 Housekeeping

2 Text as Data: PageRank

3 Summary

4 Concluding Thoughts

Housekeeping

Deadlines

- April 24th → Final draft of poster

Housekeeping

Deadlines

- April 24th → Final draft of poster
- April 29th → Poster session

Housekeeping

Deadlines

- April 24th → Final draft of poster
- April 29th → Poster session
- April 30th → Problem Set III due

Housekeeping

Deadlines

- April 24th → Final draft of poster
- April 29th → Poster session
- April 30th → Problem Set III due

Housekeeping

Deadlines

- April 24th → Final draft of poster
- April 29th → Poster session
- April 30th → Problem Set III due

Course evaluations

- Please fill out your course evaluations!

Housekeeping

Deadlines

- April 24th → Final draft of poster
- April 29th → Poster session
- April 30th → Problem Set III due

Course evaluations

- Please fill out your course evaluations!
- They help us improve the course for future students

Housekeeping

Deadlines

- April 24th → Final draft of poster
- April 29th → Poster session
- April 30th → Problem Set III due

Course evaluations

- Please fill out your course evaluations!
- They help us improve the course for future students
- Documented systematic differences in teaching evaluations independent of instructor quality (Peterson et al. 2019, 2)

PageRank intuition

What is PageRank?

- Originally developed for ranking websites (Google Search).

PageRank intuition

What is PageRank?

- Originally developed for ranking websites (Google Search).
- Based on a *random surfer model*

PageRank intuition

What is PageRank?

- Originally developed for ranking websites (Google Search).
- Based on a *random surfer model*
- **Basic idea**: importance based on linked connections

PageRank intuition

What is PageRank?

- Originally developed for ranking websites (Google Search).
- Based on a *random surfer model*
- **Basic idea**: importance based on linked connections
- More information: how you sort matters (`https://www.toptal.com/developers/sorting-algorithms`)

PageRank intuition

What is PageRank?

- Originally developed for ranking websites (Google Search).
- Based on a *random surfer model*
- **Basic idea**: importance based on linked connections
- More information: how you sort matters (`https://www.toptal.com/developers/sorting-algorithms`)

PageRank intuition

What is PageRank?

- Originally developed for ranking websites (Google Search).
- Based on a *random surfer model*
- **Basic idea**: importance based on linked connections
- More information: how you sort matters (`https://www.toptal.com/developers/sorting-algorithms`)

Why should we care?

↪ We can repurpose this for analyzing text similarity between documents

Steps: applying PageRank to professor bios

1. Prep the corpus

Steps: applying PageRank to professor bios

1. Prep the corpus
2. Transform corpus into DTM

Steps: applying PageRank to professor bios

1. Prep the corpus
2. Transform corpus into DTM
3. Calculate similarities between documents using cosine similarity function

Steps: applying PageRank to professor bios

1. Prep the corpus
2. Transform corpus into DTM
3. Calculate similarities between documents using cosine similarity function
4. Display results

Prepare corpus

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

df <- read.csv("data/harvardgov.csv")
corpus <- Corpus(VectorSource(df$bio))

corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords,
    stopwords("english"))
corpus <- tm_map(corpus, stemDocument)
corpus <- tm_map(corpus, removePunctuation)
```

Create DTM and TF-IDF

```
dtm <- DocumentTermMatrix(corpus)
dtm.mat <- as.matrix(dtm)
rownames(dtm.mat) <- df$prof

tfidf <- weightTfIdf(dtm, normalize = TRUE)
tfidf.mat <- as.matrix(tfidf)
rownames(tfidf.mat) <- df$prof
```

Create DTM and TF-IDF

```
dtm <- DocumentTermMatrix(corpus)
dtm.mat <- as.matrix(dtm)
rownames(dtm.mat) <- df$prof

tfidf <- weightTfIdf(dtm, normalize = TRUE)
tfidf.mat <- as.matrix(tfidf)
rownames(tfidf.mat) <- df$prof
```

```
tfidf
```

Create DTM and TF-IDF

```r
dtm <- DocumentTermMatrix(corpus)
dtm.mat <- as.matrix(dtm)
rownames(dtm.mat) <- df$prof

tfidf <- weightTfIdf(dtm, normalize = TRUE)
tfidf.mat <- as.matrix(tfidf)
rownames(tfidf.mat) <- df$prof
```

```r
tfidf
```

```
<<DocumentTermMatrix (documents: 48, terms: 1684)>>
Non-/sparse entries: 3767/77065
Sparsity           : 95%
Maximal term length: 37
Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

Sparseness of DTM

How sparse is this matrix overall?

Sparseness of DTM

How sparse is this matrix overall?

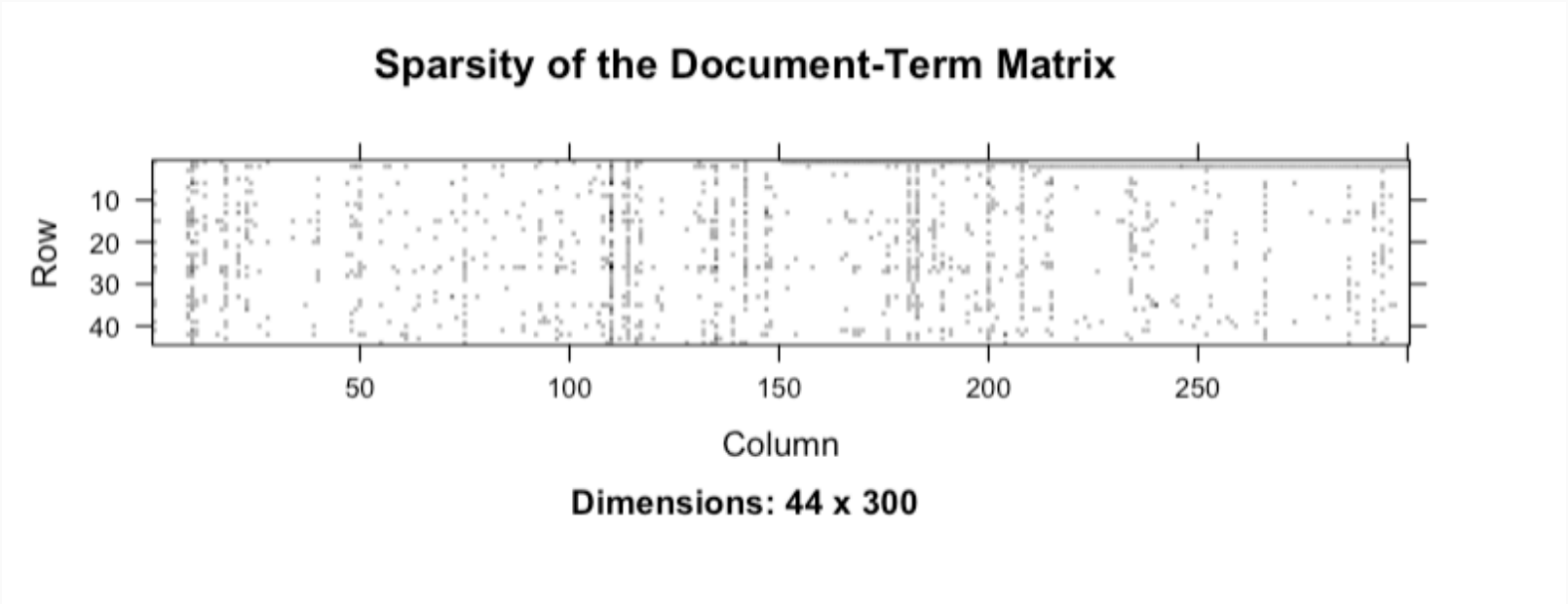


Figure 1: Snapshot of normalized DTM

Sparseness of DTM

Zooming in further, terms appear with different frequencies within each document

Sparseness of DTM

Zooming in further, terms appear with different frequencies within each document

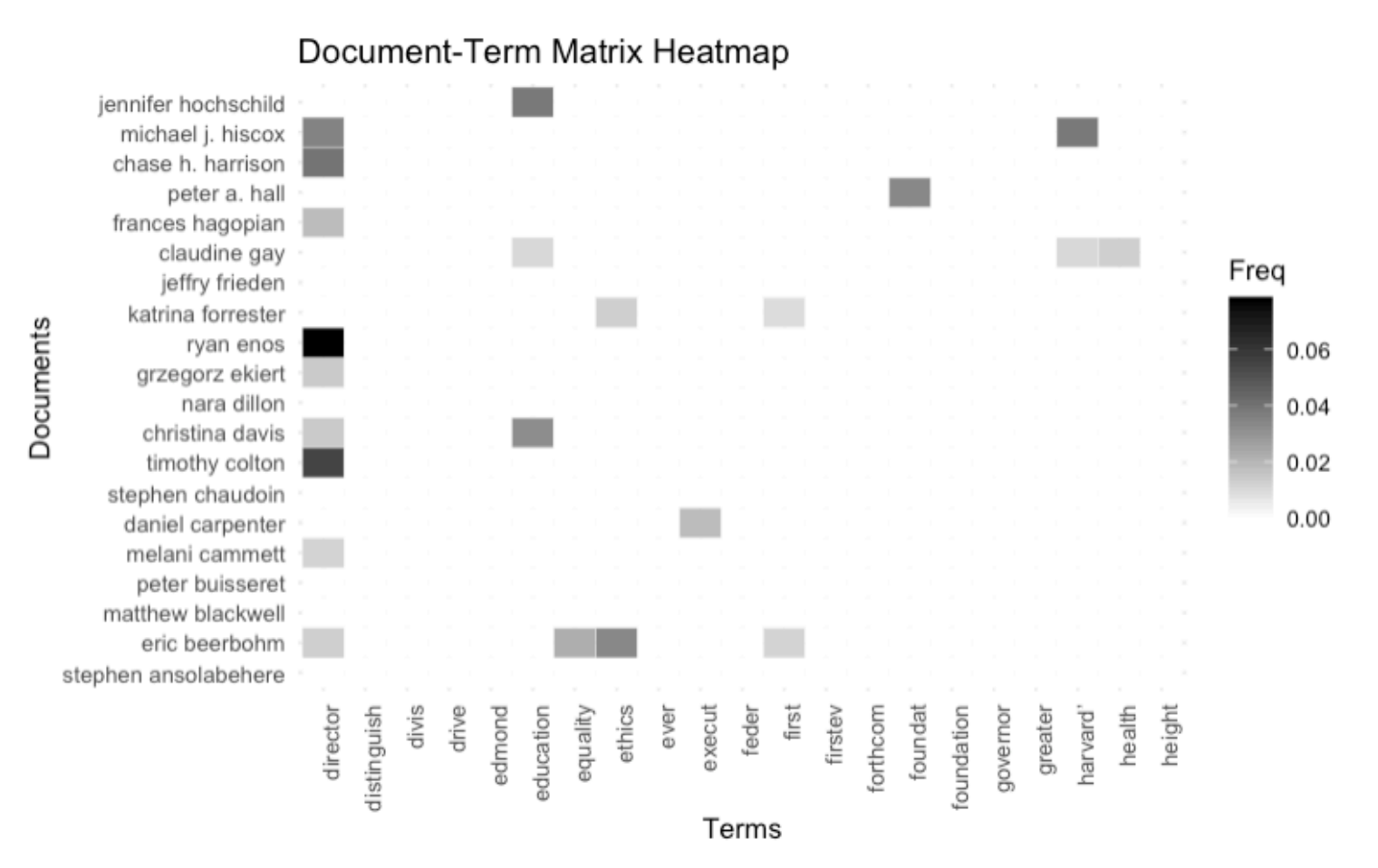


Figure 2: Heatmap of selected terms in document-term matrix

Cosine Similarity function

```r
cosine <- function(a, b) {
numer <- apply(a * t(b), 2, sum)
denom <- sqrt(sum(a2)) * sqrt(apply(b2, 1, sum))
return(numer / denom)
}
```

Cosine Similarity function

```r
cosine <- function(a, b) {
numer <- apply(a * t(b), 2, sum)
denom <- sqrt(sum(a2)) * sqrt(apply(b2, 1, sum))
return(numer / denom)
}
```

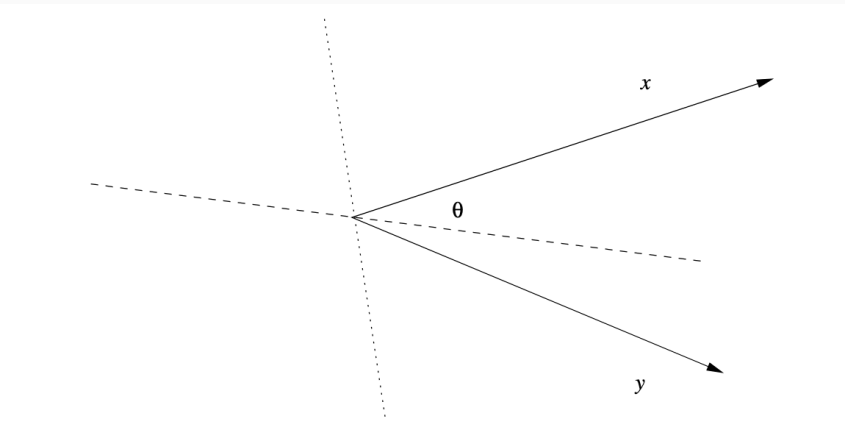


Figure 3: Two vectors make an angle θ

Create graph

Load library and initialize empty matrix

```
1 library(igraph)
2
3 cosine.adj <- matrix(0, nrow = nrow(tfidf.mat), ncol =
      nrow(tfidf.mat))
4 rownames(cosine.adj) <- colnames(cosine.adj) <-
      rownames(tfidf.mat)
```

Create graph

Load library and initialize empty matrix

```
1 library(igraph)
2
3 cosine.adj <- matrix(0, nrow = nrow(tfidf.mat), ncol =
      nrow(tfidf.mat))
4 rownames(cosine.adj) <- colnames(cosine.adj) <-
      rownames(tfidf.mat)
```

	danielle allen	stephen ansolabehere	eric beerbohm
danielle allen	0	0	0
stephen ansolabehere	0	0	0
eric beerbohm	0	0	0
matthew blackwell	0	0	0
peter buisseret	0	0	0
melani cammett	0	0	0

Create graph

Use for-loop to populate matrix with cosine similarity values

```
1
2 for (i in 1
3 (tfidf.mat)) {
4 cosine.adj[i, ] <- cosine(tfidf.mat[i,], tfidf.mat)
5 cosine.adj[i, dfphd[i]<dfphd] <- 0
6 }
```

	danielle allen	stephen ansolabehere	eric beerbohm
danielle allen	1.000000000	0.041332417	0.00000000
stephen ansolabehere	0.000000000	1.000000000	0.00000000
eric beerbohm	0.091361849	0.048064445	1.00000000
matthew blackwell	0.003588252	0.008361747	0.02377969
peter buisseret	0.008678968	0.052150952	0.02564076
melani cammett	0.015404712	0.009130659	0.00000000

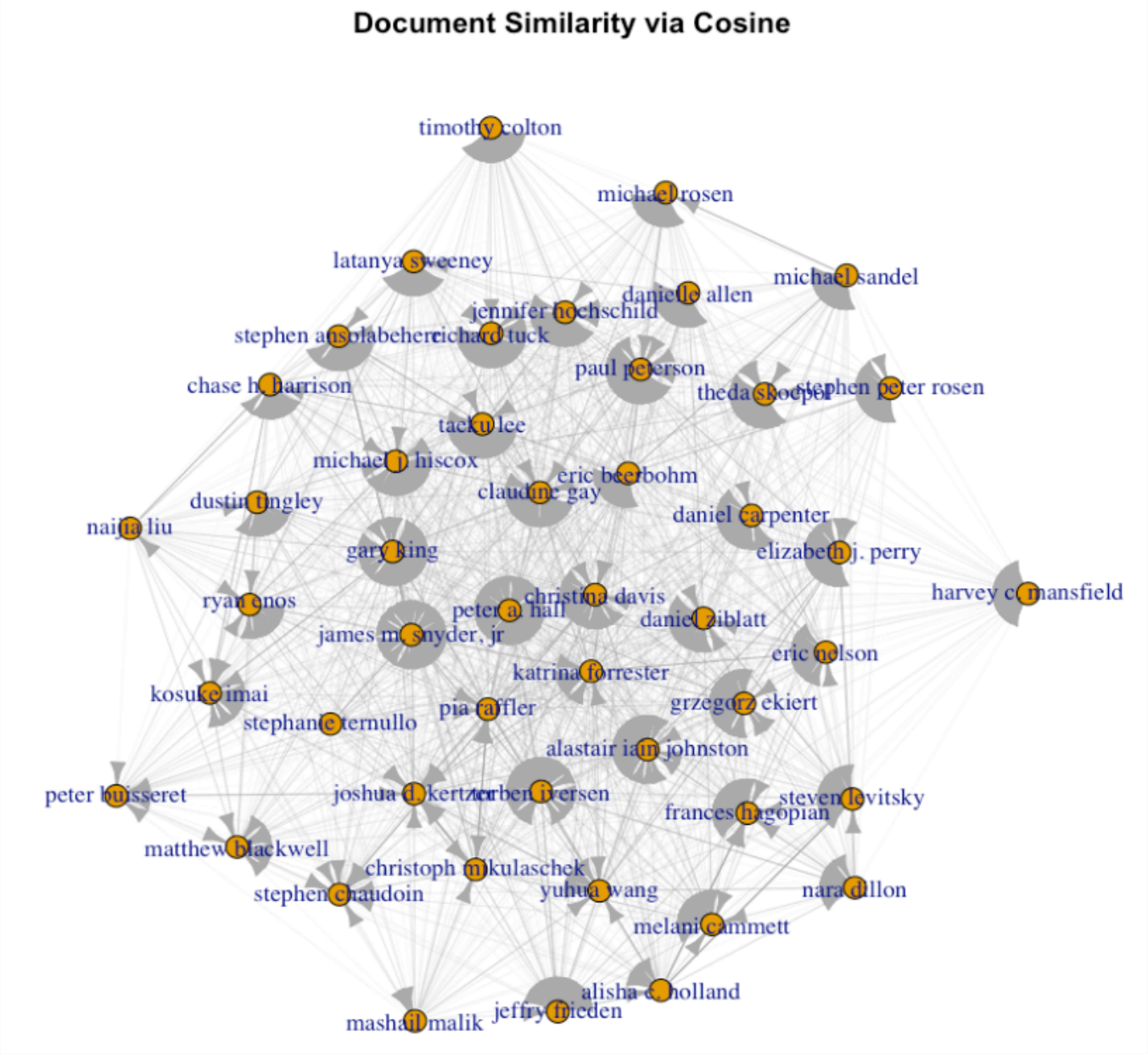
Create graph

```r
diag(cosine.adj) <- 0
cosine.graph <- graph_from_adjacency_matrix(cosine.adj,
    mode = "directed", weighted = TRUE)

set.seed(123)  # For consistent layout
layout_fr <- layout_with_fr(cosine.graph)

plot(cosine.graph,
     layout = layout_fr,
     vertex.size = 5,
     vertex.label = df$prof,
     edge.arrow.size = 0.3,
     edge.width = E(cosine.graph)$weight * 5,
     edge.color = "darkgray",
     main = "Document Similarity via Cosine")
```

Display graph



PageRank results

```
pr <- data.frame(name = colnames(cosine.adj),
year = dfphd,pagerank=page.rank(cosine.graph)vector)

arrange(pr, desc(pagerank))
```

	phdyear <int>	pagerank <dbl>
harvey c. mansfield	1961	0.154765821
paul peterson	1962	0.101931502
richard tuck	1973	0.082913801
theda skocpol	1975	0.042681307
elizabeth j. perry	1978	0.036496378
peter a. hall	1982	0.035537510
timothy colton	1974	0.034745156
jennifer hochschild	1979	0.033293136
michael rosen	1980	0.030915136
james m. snyder, jr	1984	0.029899312

Summary

- PageRank identifies influence based on connections or similarity

Summary

- PageRank identifies influence based on connections or similarity
- Unsupervised techniques help when data is high-dimensional or unstructured

Summary

- PageRank identifies influence based on connections or similarity
- Unsupervised techniques help when data is high-dimensional or unstructured
- This method bypasses manual interpretation and leverages structure in the data

Final Thoughts

- This course helps build an intuition around modeling, data cleaning, and analysis.

Final Thoughts

- This course helps build an intuition around modeling, data cleaning, and analysis.
- Once your TF, always your TF.

Final Thoughts

- This course helps build an intuition around modeling, data cleaning, and analysis.
- Once your TF, always your TF.
- Feel free to reach out (sbiondi@g.harvard.edu) if you need:

Final Thoughts

- This course helps build an intuition around modeling, data cleaning, and analysis.
- Once your TF, always your TF.
- Feel free to reach out (sbiondi@g.harvard.edu) if you need:
    - Grad school advice

Final Thoughts

- This course helps build an intuition around modeling, data cleaning, and analysis.
- Once your TF, always your TF.
- Feel free to reach out (sbiondi@g.harvard.edu) if you need:
    - Grad school advice
    - Letters of recommendation

Final Thoughts

- This course helps build an intuition around modeling, data cleaning, and analysis.
- Once your TF, always your TF.
- Feel free to reach out (sbiondi@g.harvard.edu) if you need:
    - Grad school advice
    - Letters of recommendation
    - Research help

Final Thoughts

- This course helps build an intuition around modeling, data cleaning, and analysis.
- Once your TF, always your TF.
- Feel free to reach out (sbiondi@g.harvard.edu) if you need:
  - Grad school advice
  - Letters of recommendation
  - Research help
- Don't forget to fill out your Q evaluations!