Computational Social Science

Topic Modeling

Dr. Thomas Davidson

Rutgers University

March 23, 2022

Plan

1. Course updates
2. Structural Topic Modeling (STM)

Course updates

Homework 3

- Homework 2 grades and comments have been released
- Homework 3 on NLP will be released this evening
- Project feedback also released this evening

Structural Topic Modeling

Background

- LDA assumes *topic prevalence* (frequency topic is mentioned) and *topic content* (the words used to discuss a topic) are constant across documents
  - e.g. In the previous example, we assume that *NYT* and *WSJ* devote equal coverage to topics and discuss topics in the same way.
- STM extends LDA by "allowing for the inclusion of covariates of interest into the prior distributions for document-topic proportions and topic-word distributions" (Roberts et al. 2014).
  - This allows analysis of how topics vary according to other factors, for example the treatment in a survey experiment may alter open responses.

Structural Topic Modeling

Topic prevalence

- Prevalence refers to the frequency distribution of a topic across documents
- As social scientists, we often want to see how a topic varies by some categorical variable of interest
  - Author (person, organization, publisher, political party, etc.)
  - Time (day, month, year, decade, etc.)
  - Demographics (age group, gender, race, ethnicity, etc.)

Structural Topic Modeling

Topic prevalence

- Example 1: How does topic prevalence vary between the *New York Times* and the *Wall Street Journal*?
  - Potential hypotheses:
    - WSJ focues more on topics including business and the economy
    - NYT focuses more on cultural issues
- Example 2: How does prevalence vary over time?
  - Potential hypotheses:
    - Topic prevalence ebbs and flows following the news cycle

Structural Topic Modeling

Topic content

- Content refers to the way different topics are dicussed
- As social scientists, we might expect different groups to use different kinds of language

Structural Topic Modeling

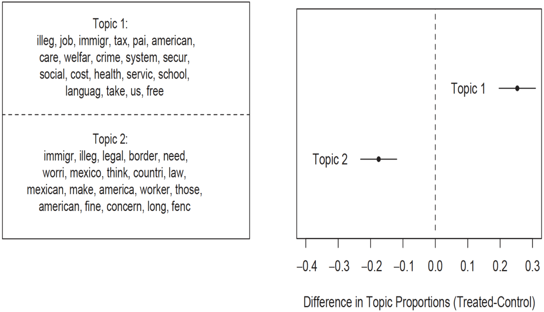
Topic content

- Example: How is the issue of the economy described differently by the *New York Times* and the *Wall Street Journal*?
  - Hypothesis 1: The *WSJ* uses more "jargon" words because it is targeted towards a more knowledgeable audience
  - Hypothesis 2: The *NYT* is more critical of capitalism than the *WSJ*
- These hypotheses would require careful identification of relevant topics and analysis of how language varies across publications.

Structural Topic Modeling

Analyzing open-ended survey responses using an STM

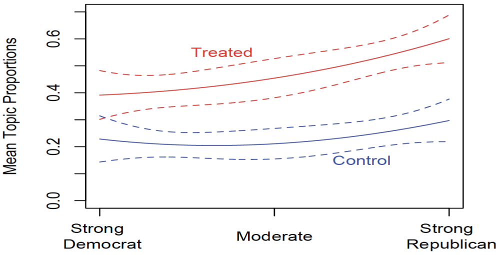
FIGURE 7 Words and Treatment Effect Associated with Topic 1



Structural Topic Modeling

Analyzing open-ended survey responses using an STM

FIGURE 8 Party Identification, Treatment, and the Predicted Proportion in Topic 1



Training an LDA topic model in R

Loading the corpus

Loading a corpus of most recent 1000 tweets by the New York Times and the Wall Street Journal. What do you notice about the file? Are there any problems with the way the data have been stored?

```
library(tidyverse)
library(lubridate)
data <- as_tibble(read_csv("data/nytimes_wsj_2000_statuses_march2122.cs
data <- data %>%
    mutate(text = gsub("#[A-Za-z0-9]+|@[A-Za-z0-9]", "", text)) %>%
    mutate(text = gsub("(http[^ ]*)|(www.[^ ]*)", "", text)) %>%
    distinct(text, .keep_all =TRUE) # Removing duplicates

data <- data %>% filter(month(created_at) == 3)
```

Structural Topic Modeling

Running an STM

The first step is to select the relevant metadata. In this case we are going to use the screen_name and the day of the month.

```
min(data$created_at)
max(data$created_at)
data$day <- data$created_at %>% day()
meta <- data %>% select(screen_name, day)
```

Structural Topic Modeling

Running an STM

The `stm` library has its own set of functions for processing data. `textProcessor` takes a corpus, plus metadata, and conducts pre-processing tasks. `prepDocuments` then converts the documents into the appropriate format.

```
library(stm)
# install.packages("stm")
processed.docs <- textProcessor(data$text, metadata = meta)
output <- prepDocuments(processed.docs$documents, processed.docs$vocab,
```

Structural Topic Modeling

Finding K

The STM package can calculate some heuristics for finding the "best" value of `K`. This can take a while as it must run each of the models specified in the vector passed to the `K` parameter.

```
library(parallel)
search.results <- searchK(output$documents, output$vocab,
                          K = c(20,30,40,60,80),
                          data = output$meta,
                          proportion=0.1, # proportion of docs held-out
                          cores=detectCores() # use maximum number of availabl
)
```

See https://juliasilge.com/blog/evaluating-stm/ for an alternative approach that enables some more post-estimation evaluation.

Structural Topic Modeling

Selecting K

```
plot(search.results)
```

See Mimno, David, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. "Optimizing Semantic Coherence in Topic Models." In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 262–72. ACL for discussion of the semantic coherence measure.

Structural Topic Modeling

Fitting a model with k=40. The screen_name field is used as a covariate for both prevalence and content. This means that we allow both the prevalence and content of topics to vary depending on whether the tweet was from the *NYT* or the *WSJ*. Day is used as a covariate for prevalence, where s() is a non-linear spline function.

```
K=40
fit <- stm(documents = output$documents, vocab = output$vocab, K=K,
           data = output$meta,
           prevalence = ~ screen_name + s(day), # s takes a non-linear
           content = ~ screen_name, # content can only contain one vari
           verbose = TRUE
           )
```

Structural Topic Modeling

Loading data

I stored the image of this workspace and uploaded it to Github. You can load the trained model and all other files in this script by running this line.

```
#save.image(file = "data/news_stm.RData")
#load(file = "data/news_stm.RData")
```

Structural Topic Modeling

Plotting the results

We can directly plot the proportions to show how frequent different topics are. Here are the first 20.

```
plot(fit, type = "summary", topics = 1:20)
```

Structural Topic Modeling

Plotting the results

We can directly plot the proportions to show how frequent different topics are. Here are the first 20.

```
plot(fit, type = "summary", topics = 21:40)
```

Structural Topic Modeling

Inspecting topics

```
labelTopics(fit, topics=20, n=10)
```

Structural Topic Modeling

Inspecting topics

We can use `findThoughts` to identify documents with a high weight in a given topic. Note that the original texts column does not work, I have to use the index for the metadata file to identify relevant columns.

```
t=20
thoughts <- findThoughts(fit, texts = as.character(data[as.numeric(rown
for (i in unlist(thoughts$docs)) {print(i)}
```

Structural Topic Modeling

Inspecting topics

```
t=30
thoughts <- findThoughts(fit, texts = as.character(data[as.numeric(rown
for (i in unlist(thoughts$docs)) {print(i)}
```

Structural Topic Modeling

Inspecting topics

```
t=38
thoughts <- findThoughts(fit, texts = as.character(data[as.numeric(rown
for (i in unlist(thoughts$docs)) {print(i)}
```

Structural Topic Modeling

Estimating relationship between topic prevalence and metadata

```
prep <- estimateEffect(~ screen_name + s(day), fit, meta = output$meta)
summary(prep, topics = c(20, 30,38)) # show results for selected topics
```

Structural Topic Modeling

Topic prevalence by publication

We can see how different topics vary in prevalence according to the publication. The horizontal lines indicate 95% confidence intervals.

Structural Topic Modeling

Prevalence over time

We can use the `day` variable to track how prevalence changes over time.

Structural Topic Modeling

Content by publication

We can also see how the topic content varies according to the publication. Let's take a look at each topic

Structural Topic Modeling

Content by publication

Structural Topic Modeling

Content by publication

Structural Topic Modeling

Topic correlations

We can also measure the correlations between different topics.
Topics are connected if the correlation exceeds a threshold.

Structural Topic Modeling

- Resources
  - The STM website contains information on various tools and research papers that use the approach
    - There are several packages including `stmBrowser`, `stmCorrViz` and `stminsights` that enable more interactive visualization.
  - The vignette provides a closer description of the methodology and a hands-on guide to using the `stm` package.

Summary

- Topic modeling is an inductive approach for the summary of large text corpora
  - Analysis of topic models involves the interpretation of topics
  - A key challenge is selecting an appropriate number of topics
- LDA algorithm summarize as corpus into K topics
  - Each document is composed of a mixture of topics
  - Each topic is a mixture of words
- STM improves on LDA by allowing topic prevalence and content to vary by covariates
  - This is particularly useful for social scientific applications