Text as Data: Homework 5

Jeff Ziegler

August 23, 2017

1 Using the `Structural Topic Model` in `R`

Using the preprocessed data based on the NYT Json file and the subsequent document term matrix for analysis in `stm`.

a) Download the `stm` package for `R` from `CRAN`

b) Convert the document-term matrix to the appropriate format. To do this, create a list in `R` where each component of the list corresponds to an individual document. Store in each component of the list a two rom matrix. The number of columns corresponds to the number of non-zero entries for the document in the document-term matrix. The first row will describe the words used in the document (the columns with the non-zero entry). The second row will correspond to a count of each of the words in the document (they should all be non-zero)

c) Following the help file in `STM` fit a model with 8 topics that conditions on the `desk` of origin for topic prevalence

d) Use `labelTopics` to label each of the topics

e) Compare the 8 topic proportions for each document to the 8 topic proportions without conditioning on `desk` (in vanilla LDA). How do the results differ?

```
# Problem 1

# load libraries and .csv files
library(rjson); library(stm)
NYTjson <- fromJSON(file="~/Documents/Git/WUSTL_textAnalysis/nyt_ac.json")

# fit a model with 8 topics that conditions on the desk of origin
# prep documents for STM
stmData <- as.data.frame(cbind(sapply(lapply(NYTjson, `[[`, c('meta', 'dsk')),
    paste0, collapse=""),
                                sapply(lapply(NYTjson, `[[`, c('body', 'body_
  text')), paste0, collapse="")))
```

```r
names(stmData) <- c("desk", "documents"); stmData$documents <- as.character(stmData$documents)
# create a function to only take first 25 words to preview docs
# visualize labels
makeShortDoc <- function(x) {
  ul = unlist(strsplit(x, split = "\\s+"))[1:25]
  paste(ul,collapse=" ")
}
stmData$shortdoc <- unlist(lapply(stmData$documents, makeShortDoc))

#stmData$desk <- as.factor(stmData$desk)
processedSTM <- textProcessor(stmData$documents, metadata = stmData)
outSTM <- prepDocuments(processedSTM$documents, processedSTM$vocab, processedSTM$meta)

# fit STM w/ 8 topics
# using the default settings found in the help package
stmModel <- stm(documents = outSTM$documents, vocab = outSTM$vocab, K = 8,
     prevalence =~desk, max.em.its = 75, data = outSTM$meta,
     init.type = "Spectral")

# d) label each topic
labelTopics(stmModel, c(1:8))

# create visualization of topics by previewing docs
pdf("~/Documents/Git/WUSTL_textAnalysis/HW5topicReview.pdf")
par(mfrow = c(2, 4), mar = c(.5, .5, 1, .5))
for(i in 1:8){
  plotQuote(findThoughts(stmModel, texts = stmData$shortdoc, n = 2, topics = i)$docs[[1]],
            width = 30, main = paste0("Topic", i, sep=" "))
}
dev.off()

# e) run vanilla LDA
# using the default settings found in the help package
ldaModel <- stm(documents = outSTM$documents, vocab = outSTM$vocab, K = 8,
                max.em.its = 75, data = outSTM$meta, init.type = "Spectral")
# plot LDA and STM topic proportions
pdf("~/Documents/Git/WUSTL_textAnalysis/HW5topicProportionsSTM.pdf")
plot(stmModel, type = "hist", xlim = c(0, .5), labeltype="frex")
dev.off()

pdf("~/Documents/Git/WUSTL_textAnalysis/HW5topicProportionsLDA.pdf")
plot(ldaModel, type = "hist", xlim = c(0, .5), labeltype="frex")
dev.off()

# not much difference between the two estimated topic proportions
```

Figure 1: Example documents highly associated with topics.

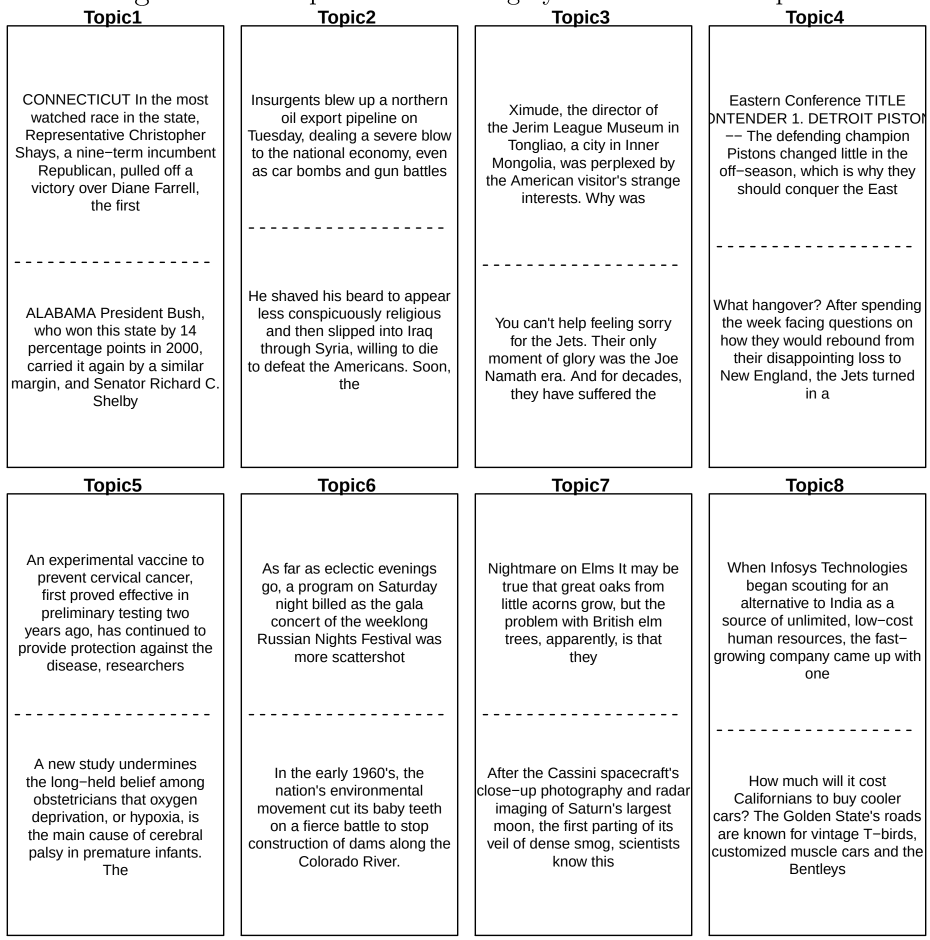
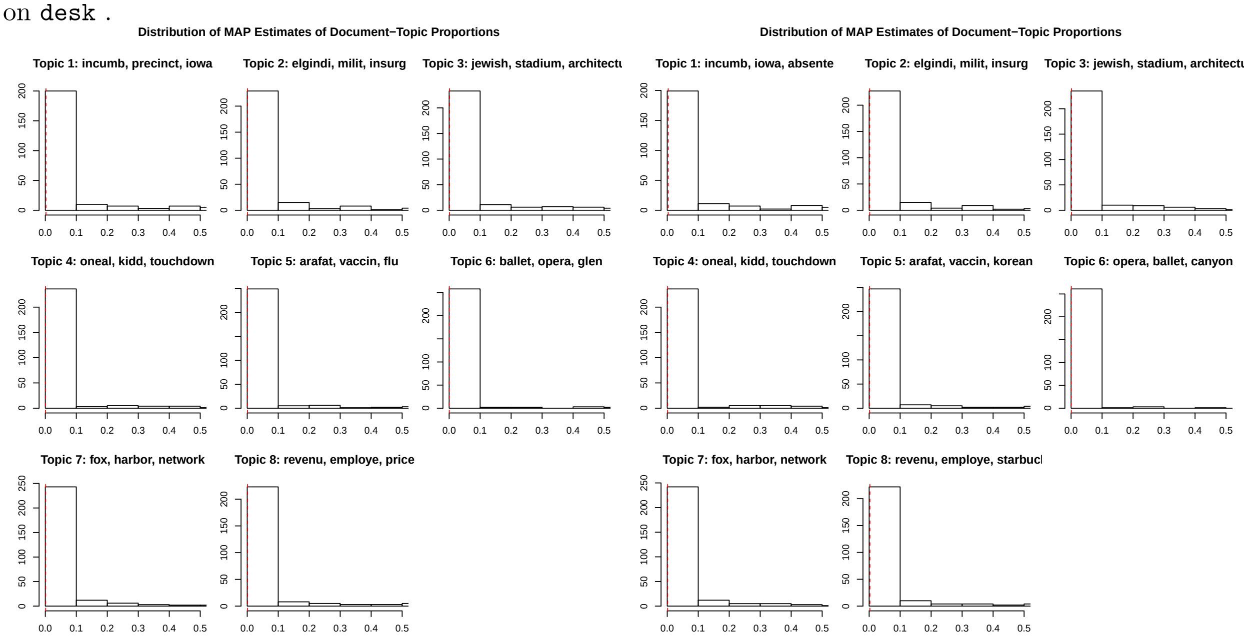


Figure 2: Expected distribution of topic proportions across the documents, vanilla LDA vs. conditioning on `desk` .



2 Machiavelli's Prince

In this part of the assignment we will analyze Machiavelli's *The Prince*. Download `Mach.tar` from the course website and expand the compressed folder. (This is relevant `http://xkcd.com/1168/`).

Each file represents a subset of the manuscript. We will analyze its contents using principal components, multidimensional scaling, and clustering methods.

Create a Document-Term Matrix

Using the sections from the Machiavelli text, create a document term matrix.

- Discard punctuation, capitalization
- Apply the porter stemmer to the documents
- Identify the 500 most common unigrams
- Create a $N \times 500$ document term matrix $\boldsymbol{X}$, where the columns count the unigrams and the rows are the documents

We will work with a normalized version of the term document matrix. That is we will divide each row by the total number of words in the top 500 unigrams used:

$$\boldsymbol{x}_i^* = \frac{\boldsymbol{x}_i}{\sum_{j=1}^{500} x_{ij}}$$

$$\boldsymbol{X}^* = \begin{pmatrix} \boldsymbol{x}_1^* \\ \boldsymbol{x}_2^* \\ \vdots \\ \boldsymbol{x}_N^* \end{pmatrix}$$

Low Dimensional Embeddings with Principal Components

1) Wise Will (WW), your friend with a weird name, notices you looking at the slides about principal component analysis (PCA). WW casually remarks that the variance of the eigenvalues of the variance-covariance matrix is a useful heuristic for knowing if PCA can be fruitfully applied to some document-term matrix. WW, completely unsolicited, explains that as the variance of the eigenvalues goes up, the more useful PCA will be. He then laughs and leaves your office. WW is kind of a jerk.

   Let's formalize WW's suggestion. Suppose document-term matrix $\boldsymbol{X}$ has variance-covariance matrix $\boldsymbol{\Sigma} = \frac{\boldsymbol{X}'\boldsymbol{X}}{N}$. And suppose that $\boldsymbol{\Sigma}$ has eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_d > 0$. Then we calculate the variance of the eigenvalues as

   $$\sigma^2 = \frac{1}{d}\sum_{j=1}^{d}(\lambda_j - \bar{\lambda})^2$$

   where $\bar{\lambda}$ is $\frac{1}{d}\sum_{i=1}^{d}\lambda_i$. WW is saying that as σ^2 gets bigger, a low-dimensional embedding via PCA will provide a better summary of our data.

   Does WW have a good point? Why or why not? (Hint: what do the eigenvalues represent?)

2) Apply the function `prcomp` to $\boldsymbol{X}^*$. Be sure to set use a scaled version of the data, by setting `scale = T`, which will ensure that each column has unit variance.

   a) Create a plot of variance explained by each additional principal component. What does this plot suggest about the number of components to include?
   b) Plot the two-dimensional embedding of the text documents. Label the texts with their number. (Each file is `Mach_XX.txt`, where `XX` is the chunk number)
   c) Label the two largest principal components. What does this embedding suggest about the primary variation this representation of the Prince? (Hint: if your `embed` is your object with principal components, examine `embed$rotation`)

3) An alternative method—discussed at the end of the seventh lecture—is multidimensional scaling (MDS). Classic MDS attempts to preserve distances between objects in a low dimensional scaling.

   a) Calculate the Euclidean distance between each document using $\boldsymbol{X}^*$. Call this matrix $\boldsymbol{D}(\boldsymbol{X}^*)$ (Hint: use `R`'s built in function `dist`)

   b) Apply the classic MDS to $\boldsymbol{D}(\boldsymbol{X}^*)$ using the `R` function `cmdscale`. That is, execute the code
   `mds_scale<- cmdscale(DISTANCE_MATRIX, k = 2)`

   c) Apply PCA to $\boldsymbol{X}^*$, but this time do not use `prcomp`'s scaling option. That is, use `prcomp` with `scale = F`.

   d) Compare the first dimension of the output from classic MDS to the first dimension of the embedding from principal components. What is the correlation between the embeddings?

   d) Now use `dist` to create a distance matrix using the `manhattan` metric, apply Classic multidimensional scaling to the distance matrix based on manhattan distance, and compare the first dimension of this embedding to the first dimension from PCA. What is the correlation?

   e) What do you conclude about the relationship between PCA and MDS?