

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 row 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?

```
1 # Problem 1
2
3 # load libraries and .csv files
4 library(rjson); library(stm); library(tm); library(stringr)
5 NYTjson <- fromJSON(file=~ /Documents/Git/WUSTL-textAnalysis/nyt-ac.json")
6
7 # fit a model with 8 topics that conditions on the desk of origin
8 # prep documents for STM
9 stmData <- as.data.frame(cbind(sapply(lapply(NYTjson, '[', c('meta', 'dsk'))),
10                                paste0, collapse=""),
11                           sapply(lapply(NYTjson, '[', c('body', 'body_
12                                text')), paste0, collapse=""))
```

```

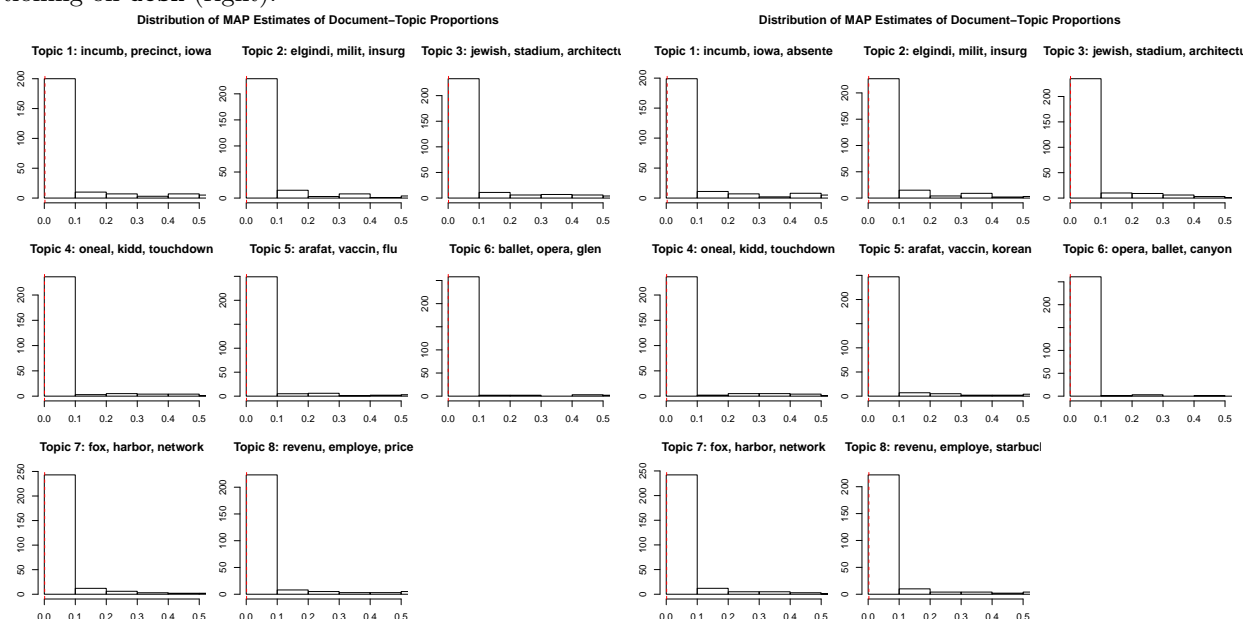
11 names(stmData) <- c("desk", "documents"); stmData$documents <- as.character(
    stmData$documents)
12 # create a function to only take first 25 words to preview docs
13 # visualize labels
14 makeShortDoc <- function(x) {
15   ul = unlist(strsplit(x, split = "\\s+"))[1:25]
16   paste(ul, collapse=" ")
17 }
18 stmData$shortdoc <- unlist(lapply(stmData$documents, makeShortDoc))
19
20 #stmData$desk <- as.factor(stmData$desk)
21 processedSTM <- textProcessor(stmData$documents, metadata = stmData)
22 outSTM <- prepDocuments(processedSTM$documents, processedSTM$vocab,
    processedSTM$meta)
23
24 # fit STM w/ 8 topics
25 # using the default settings found in the help package
26 stmModel <- stm(documents = outSTM$documents, vocab = outSTM$vocab, K = 8,
27   prevalence = ~desk, max.em.its = 75, data = outSTM$meta,
28   init.type = "Spectral")
29
30 # d) label each topic
31 labelTopics(stmModel, c(1:8))
32
33 # create visualization of topics by previewing docs
34 pdf("~/Documents/Git/WUSIL_textAnalysis/HW5topicReview.pdf")
35 par(mfrow = c(2, 4), mar = c(.5, .5, 1, .5))
36 for(i in 1:8){
37   plotQuote(findThoughts(stmModel, texts = stmData$shortdoc, n = 2, topics = i
    )$docs[[1]],
38     width = 30, main = paste0("Topic", i, sep=" "))
39 }
40 dev.off()
41
42 # e) run vanilla LDA
43 # using the default settings found in the help package
44 ldaModel <- stm(documents = outSTM$documents, vocab = outSTM$vocab, K = 8,
45   max.em.its = 75, data = outSTM$meta, init.type = "Spectral")
46 # plot LDA and STM topic proportions
47 pdf("~/Documents/Git/WUSIL_textAnalysis/HW5topicProportionsSTM.pdf")
48 plot(stmModel, type = "hist", xlim = c(0, .5), labeltype="frex")
49 dev.off()
50
51 pdf("~/Documents/Git/WUSIL_textAnalysis/HW5topicProportionsLDA.pdf")
52 plot(ldaModel, type = "hist", xlim = c(0, .5), labeltype="frex")
53 dev.off()
54 # not much difference between the two estimated topic proportions

```

Figure 1: Example documents highly associated with topics.

Topic1	Topic2	Topic3	Topic4
<p>CONNECTICUT In the most watched race in the state, Representative Christopher Shays, a nine-term incumbent Republican, pulled off a victory over Diane Farrell, the first</p> <p>-----</p> <p>ALABAMA President Bush, who won this state by 14 percentage points in 2000, carried it again by a similar margin, and Senator Richard C. Shelby</p>	<p>Insurgents blew up a northern oil export pipeline on Tuesday, dealing a severe blow to the national economy, even as car bombs and gun battles</p> <p>-----</p> <p>He shaved his beard to appear less conspicuously religious and then slipped into Iraq through Syria, willing to die to defeat the Americans. Soon, the</p>	<p>Ximude, the director of the Jerim League Museum in Tongliao, a city in Inner Mongolia, was perplexed by the American visitor's strange interests. Why was</p> <p>-----</p> <p>You can't help feeling sorry for the Jets. Their only moment of glory was the Joe Namath era. And for decades, they have suffered the</p>	<p>Eastern Conference TITLE CONTENDER 1. DETROIT PISTONS --- The defending champion Pistons changed little in the off-season, which is why they should conquer the East</p> <p>-----</p> <p>What hangover? After spending the week facing questions on how they would rebound from their disappointing loss to New England, the Jets turned in a</p>
Topic5	Topic6	Topic7	Topic8
<p>An experimental vaccine to prevent cervical cancer, first proved effective in preliminary testing two years ago, has continued to provide protection against the disease, researchers</p> <p>-----</p> <p>A new study undermines the long-held belief among obstetricians that oxygen deprivation, or hypoxia, is the main cause of cerebral palsy in premature infants. The</p>	<p>As far as eclectic evenings go, a program on Saturday night billed as the gala concert of the weeklong Russian Nights Festival was more scattershot</p> <p>-----</p> <p>In the early 1960's, the nation's environmental movement cut its baby teeth on a fierce battle to stop construction of dams along the Colorado River.</p>	<p>Nightmare on Elms It may be true that great oaks from little acorns grow, but the problem with British elm trees, apparently, is that they</p> <p>-----</p> <p>After the Cassini spacecraft's close-up photography and radar imaging of Saturn's largest moon, the first parting of its veil of dense smog, scientists know this</p>	<p>When Infosys Technologies began scouting for an alternative to India as a source of unlimited, low-cost human resources, the fast-growing company came up with one</p> <p>-----</p> <p>How much will it cost Californians to buy cooler cars? The Golden State's roads are known for vintage T-birds, customized muscle cars and the Bentleys</p>

Figure 2: Expected distribution of topic proportions across the documents, vanilla LDA (left) and conditioning on **desk** (right).



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 \mathbf{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:

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

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

```

1 # Problem 2
2
3 # we'll make our DIM using the tm package primarily
4 # create a corpus and transform data
5 # read in files
6 # make corpus
7 machText <- Corpus(DirSource("~/Documents/Git/WUSTL_textAnalysis/MachText"),
8                     readerControl = list(language = "en"))
9 # remove punctuation and capitalization
10 machText <- tm_map(machText, removePunctuation)
11 machText <- tm_map(machText, content_transformer(tolower))
12 # we'll also remove stop words
13 machText <- tm_map(machText, removeWords, stopwords("english"))
14 # apply porter stemmer
15 machText <- tm_map(machText, stemDocument)
16 # create DIM of machTexts
17 machDTM <- DocumentTermMatrix(machText)
18 # reduce DIM to top 500 unigrams
19 machDTM <- machDTM[, c(501:2367)]
20 # normalize by term frequency - i.e. divide count of each word
21 # in document by total number of words in document
22 machDTM <- normalize(machDTM$j)
23
24 # reduce DIM to top 500 unigrams
25 machDTM <- machDTM[, c(501:2367)]
26 # show top 500 unigrams
27 # machDTM$dimnames$Terms
28 # see what DIM looks like
29 machDTM <- machDTM/rowSums(as.matrix(machDTM))
30 inspect(machDTM)

```

```

<<DocumentTermMatrix (documents: 188, terms: 500)>>
Non-/sparse entries: 6279/87721
Sparsity           : 93%
Maximal term length: 14
Weighting          : term frequency (tf)
Sample            :

```

Docs	Terms					
	alway	men	one	peopl	power	ruler
Mach_1.txt	0.00000000	0.01315789	0.00000000	0.00000000	0.00000000	0.01315789
Mach_101.txt	0.00000000	0.01086957	0.00000000	0.00000000	0.00000000	0.01086957
Mach_106.txt	0.01639344	0.00000000	0.01639344	0.00000000	0.00000000	0.01639344
Mach_110.txt	0.00000000	0.00000000	0.18000000	0.00000000	0.04000000	0.00000000
Mach_115.txt	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.05128205
Mach_126.txt	0.04761905	0.00000000	0.04761905	0.04761905	0.02380952	0.04761905
Mach_153.txt	0.03333333	0.06666667	0.00000000	0.00000000	0.03333333	0.06666667
Mach_168.txt	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.02500000
Mach_56.txt	0.00000000	0.00000000	0.02040816	0.02040816	0.00000000	0.02040816
Mach_76.txt	0.03921569	0.01960784	0.00000000	0.01960784	0.01960784	0.05882353

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 \mathbf{X} has variance-covariance matrix $\mathbf{\Sigma} = \frac{\mathbf{X}'\mathbf{X}}{N}$. And suppose that $\mathbf{\Sigma}$ has eigenvalues $\lambda_1 > \lambda_2 > \dots > \lambda_d > 0$. 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?)

Reducing the variance of the eigenvalues may or may not help improve PCA's ability to better summarize data, but reducing the sum of remaining eigenvalues reduces the error because the total variance explained = (sum of included eigenvalues)/(sum of all eigenvalues).

- 2) Apply the function `prcomp` to \mathbf{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`)

```

1 # Problem 3
2
3 # 2) apply the function prcomp
4 # scale data to ensure each column has unit variance
5 machPCA <- prcomp(machDTM, scale = T)
6
7 # a ) reate a plot of variance explained by each additional principal
  component
8 pdf("~/Documents/Git/WUSIL_textAnalysis/HW5screePlot.pdf")
9 plot(machPCA, type = "l")
10 dev.off()
11 # the "elbow rule" doesn't really apply, and we don't want to introduce
12 # more error for better fit at a certain point
13 # so we'll include 10 components
14
15 pdf("~/Documents/Git/WUSIL_textAnalysis/HW5pcaEmbedding.pdf")
16 plot(machPCA$rotation, pch='',
17       xlab="1st Principal Component", ylab="2nd Principal Component")
18 text(machPCA$rotation, labels=str_extract(machDTM$dimnames$Docs, "[[:digit:]]+"), cex= 0.7)
19 dev.off()
20
21 # c) looking at figure 4, it appears that the first two components
22 # are orthogonal to each other and that most documents tend
23 # toward zero along both dimensions
24 # find extreme docs (13, 112, 134; and 76, 85)
25 # first component seems to be advice to the ruler
26 # w/ examples that use Romans frequently
27 # second compotent is discussing protecting a ruler's state

```

Figure 3: Scree plot of the variance explained by the addition of each component.

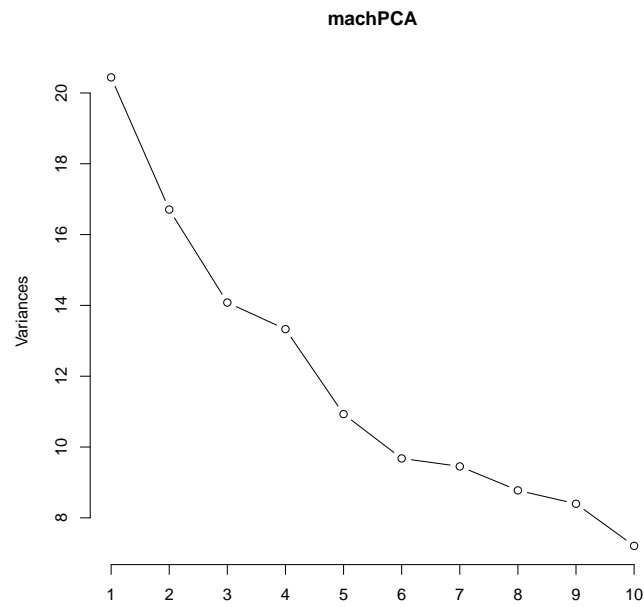
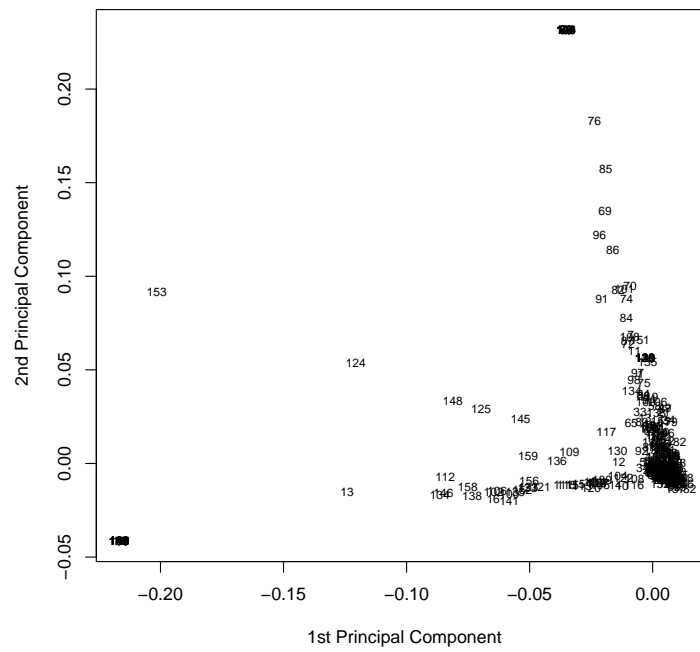


Figure 4: Projection of documents on the first two principal components.



- 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 \mathbf{X}^* . Call this matrix $\mathbf{D}(\mathbf{X}^*)$ (Hint: use R's built in function `dist`)
 - b) Apply the classic MDS to $\mathbf{D}(\mathbf{X}^*)$ using the R function `cmdscale`. That is, execute the code


```
mds_scale<- cmdscale(DISTANCE_MATRIX, k = 2)
```
 - c) Apply PCA to \mathbf{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?

```

1 # c) looking at figure 4, it appears that the first two components
2 # are orthogonal to each other and that most documents tend
3 # toward zero along both dimensions
4 # find extreme docs (13, 112, 134; and 76, 85)
5 # first component seems to be advice to the ruler
6 # w/ examples that use Romans frequently
7 # second component is discussing protecting a ruler's state
8
9 # 3 a) calculate the Euclidean distance of machDTM
10 euclideanMachDTM <- as.matrix(dist(machDTM[, -1], method = "euclidean"))
11 # b) apply the classic MDS
12 classicMDS <- cmdscale(euclideanMachDTM, k = 2)
13 # c) re-run PCA w/o scaling
14 machPCAunscaled <- prcomp(machDTM, scale = F)
15 # d) check correlation between embeddings
16 cor(classicMDS[, 1], machPCAunscaled$x[, 1])
17 # cor = 0.99
18 # e) create distance matrix using the manhattan metric
19 manhattanMachDTM <- as.matrix(dist(machDTM[, -1], method = "manhattan"))
20 # apply classic multidimensional scaling
21 manhattanMDS <- cmdscale(manhattanMachDTM, k = 2)
22 cor(manhattanMDS[, 1], machPCAunscaled$x[, 1])
23 # cor = 0.95
24
25 # when PCA minimizes dimensions, tries to preserve covariance of data

```

```
26 # when MDS minimizes dimensions, tries to preserve distance between data
    points
27 # so if covariance in data = distance (euclidean or manhattan) between data
    points they should be the same
```