Unsupervised Learning methods (i.e. Topicmodels)

Loren Collingwood

Unsupervised learning has grown tremendously in popularity because users do not need a corpus of pre-labeled documents. Instead, topic modeling can sort documents into distinct categories. The user pre-specifies the number of topics in advance. However, topicmodels can be messy and hard to interpret, so much care and thought must go into theoretical development and also into pre-processing.

```
options(scipen = 999, digits = 4)

########################
#       Packages       #
########################
#install.packages("topicmodels")
library(topicmodels)
library(quanteda)
```

```
## Package version: 2.1.1
## Parallel computing: 2 of 8 threads used.
## See https://quanteda.io for tutorials and examples.
##
## Attaching package: 'quanteda'
## The following object is masked from 'package:utils':
##
##     View
```

```
library(tm)
```

```
## Loading required package: NLP
##
## Attaching package: 'NLP'
## The following objects are masked from 'package:quanteda':
##
##     meta, meta<-
##
## Attaching package: 'tm'
## The following objects are masked from 'package:quanteda':
##
##     as.DocumentTermMatrix, stopwords
```

```
library(descr)
```

Step 1

Load pre-existing data Loren has previously gathered (real good)

```
#################
# Set Directory #
#################

setwd("~/Dropbox/collingwood_research/posc_fall_20/POSC-207/data"); list.files()
```

```
## [1] "7G16-VOTER_ID_ACT Exposed.pdf"    "cca_links.csv"
## [3] "data_corpus_germanifestos.rds"    "mj_nyt.Rdata"
## [5] "news_coverage_WordfishReady.xlsx" "test"
```

```
############################
# Loading & Pre-processing #
############################

load("mj_nyt.Rdata") #loads "final_out" object, which is the dataframe object from above loop
objects()
```

```
## [1] "final_out"
```

```
dim(final_out)
```

```
## [1] 12863     5
```

```
########################
# Convert to Dataframe #
########################

final_out <- as.data.frame(final_out)

##############################
# Take Random Sample for now #
# (ease of computing)        #
##############################

set.seed(492847)
samp <- sample(1:nrow(final_out), 3000)
final_out <- final_out[samp,]

#####################################
# Subset to relevant items (for now) #
#####################################

final_qe <- final_out[,c("uniq_id", "year", "texts")]
dim(final_qe)
```

```
## [1] 3000    3
```

Step 2

Then convert the text column into a corpus

```
############################
# Text Mining Create Corpus #
############################

nyt_corpus <- corpus(final_qe, text_field="texts")

############################
# Label Document Variables #
############################

docnames(nyt_corpus) <- final_qe$uniq_id
head(summary(nyt_corpus))
```

```
##                  Text Types Tokens Sentences             uniq_id year
## 1  TheNewYo1994032051  2650  11898       633  TheNewYo1994032051 1994
## 2 TheNewYo20000911333   333    803        25 TheNewYo20000911333 2000
## 3 TheNewYo20080707216   564   1297        53 TheNewYo20080707216 2008
## 4 TheNewYo20000322109   438   1227        43 TheNewYo20000322109 2000
## 5 TheNewYo19960629119   104    156         6 TheNewYo19960629119 1996
## 6 TheNewYo20100812278   392    983        43 TheNewYo20100812278 2010
```

```
##############################
# Remove Non-alpha Characters #
##############################

nyt_corpus <- tokens(nyt_corpus,
                     remove_punct=T,
                     remove_numbers = T)
```

Step 3

Convert the corpus into a document term/frequency matrix

```
########################################
# Create Document Term/Frequency Matrix #
########################################

nyt_dfm <- dfm(nyt_corpus,
               stem=T,
               remove= stopwords("english"))

############################
# Look at top 20 features #
############################

topfeatures(nyt_dfm, n=20)
```

```
##      said        mr      drug       one      year       new marijuana      like
##     23244     19262      9133      8351      7920      7228      6777      6375
##     state      time     peopl       use       say     polic      will       two
##      6343      5766      5755      5502      5423      5051      4784      4733
##     offic      work      last        $
##      4496      4052      4038      3994
```

```
#################################
# Trim Matrix Based on Sparsity #
#################################

smalldfm <- dfm_trim(nyt_dfm,sparsity=.991)
smalldfm
```

```
## Document-feature matrix of: 3,000 documents, 4,836 features (94.5% sparse) and 2 docvars.
##                      features
## docs                  peopl wonder tobacco compani execut can live conclud must
##   TheNewYo1994032051     61      2      46     102     25  17   14       1   10
##   TheNewYo20000911333     1      0       0       0      0   0    0       0    0
##   TheNewYo20080707216     3      0       0       1      1   0    1       0    0
##   TheNewYo20000322109     1      0       0       0      0   0    0       0    0
##   TheNewYo19960629119     0      0       0       0      0   0    0       0    0
##   TheNewYo20100812278     4      1       0       1      0   0    0       0    0
##                      features
## docs                  denial
##   TheNewYo1994032051       7
##   TheNewYo20000911333      0
##   TheNewYo20080707216      0
##   TheNewYo20000322109      0
##   TheNewYo19960629119      0
##   TheNewYo20100812278      0
## [ reached max_ndoc ... 2,994 more documents, reached max_nfeat ... 4,826 more features ]
```

```
topfeatures(smalldfm, n=20)
```

```
##      said        mr      drug       one      year       new marijuana      like
##     23244     19262      9133      8351      7920      7228      6777      6375
##     state      time     peopl       use       say     polic      will       two
##      6343      5766      5755      5502      5423      5051      4784      4733
##     offic      work      last         $
##      4496      4052      4038      3994
```

```
########################################
# Convert to Matrix so can remove words #
########################################

sdfm_mat <- as.matrix(smalldfm) # Turn into matrix format for easier access
```

Step 4

Then clean it up manually nice and good.

```
#############################################################################
# Remove Words that don't tell us anything in Topic Model, but are frequent #
#############################################################################

remove <- c("play", "mr", "new", "show", "like", "first", "one",
            "will", "said", "say","two", "home", "get","go", "just","want",
            "use", "peopl", "think", "know", "time", "can", "make", "way", "thing", "now",
            "even", "place", "around", "ms", "includ", "charact", "also",
            "man", "ask", "come", "look", "back", "work", "see", "seem", "got", "day",
```

```
            "year", "call", "plan", "open", "room" ,"water", "men","last", "good",
            "never","us","talk", "much", "take", "road", "live", "s", "someth", "still",
            "lot", "tell", "s", "word", "well", "mani", "along", "told", "went", "tri",
            "live")

######################
# Remove those words #
######################

sdfm_mat <- sdfm_mat[,!colnames(sdfm_mat) %in% remove]

#################################################################
# Clear out text with none of the words, after the Trimming #
#################################################################

zeros <- apply(sdfm_mat, 1, function(x) ifelse(sum(x) == 0, FALSE, TRUE))
sdfm_mat <- sdfm_mat[zeros,] # Regular Matrix but seems to work with LDA
dim(sdfm_mat)
```

```
## [1] 2903 4765
```

```
#########################################################
# Calculate marijuana usage in text, for subsetting #
#########################################################

smalldfm <- as.data.frame(sdfm_mat)
table(smalldfm$marijuana)
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
##  153 1835  386  153   94   41   37   22   23   25   11   14   10   11   12    8
##   16   17   18   19   20   21   22   24   25   26   27   28   30   31   32   33
##    8    5    8    9    8    6    3    2    2    3    2    1    1    2    2    1
##   36   40   41   46
##    1    2    1    1
```

```
#################################################################
# Take only texts with word marijuana appearing at least once #
#################################################################

smalldfm <- smalldfm[smalldfm$marijuana > 0 ,]
dim(smalldfm)
```

```
## [1] 2750 4765
```

```
############################
# Convert Back to Matrix #
############################

sdfm_mat <- as.matrix(smalldfm)
dim(sdfm_mat)
```

```
## [1] 2750 4765
```

Step 5

Now estimate the LDA topic model. This may take some time. Note you need to set the number of topics to estimate in advance. This can be a bit subjective although there are ways to estimate the number of topics in the data by looking at perplexity scores.

```
##########################################
# Set up Parameters for LDA/Gibbs Sampling #
# 15 Topic Model                         #
##########################################

burnin = 1000
iter = 1000
keep = 50
thin <- 500
k <- 15
alpha <- 1/k # This improves the probability separation; very important
seed <- 48790 # Seed for replication #

##################
# Estimate Model #
##################
# may take some time to run #

fitted <- LDA(sdfm_mat, k = k, method = "Gibbs",
              control = list(alpha=alpha, burnin = burnin,
                             iter = iter, keep = keep, seed=seed) )
```

Step 6

Now take a gander at the topics. This is where all the preprocessing becomes useful because you want the topics to jump out at you so to speak. If you find yourself really working hard to interpret, you should be careful.

```
##########
# Assess #
##########

get_terms(fitted, k=10)
```

```
##       Topic 1     Topic 2  Topic 3  Topic 4    Topic 5      Topic 6
##  [1,] "film"      "polic"  "life"   "drug"     "case"       "presid"
##  [2,] "movi"      "offic"  "might"  "state"    "court"      "clinton"
##  [3,] "book"      "citi"   "realli" "unit"     "prison"     "senat"
##  [4,] "stori"     "arrest" "feel"   "offici"   "charg"      "polit"
##  [5,] "direct"    "york"   "person" "govern"   "lawyer"     "democrat"
##  [6,] "star"      "drug"   "littl"  "american" "judg"       "republican"
##  [7,] "page"      "street" "anoth"  "mexico"   "convict"    "campaign"
##  [8,] "seri"      "depart" "turn"   "countri"  "sentenc"    "parti"
##  [9,] "york"      "mayor"  "made"   "border"   "state"      "support"
## [10,] "marijuana" "black"  "long"   "mexican"  "prosecutor" "state"
##       Topic 7     Topic 8      Topic 9  Topic 10    Topic 11   Topic 12
##  [1,] "player"    "marijuana"  "music"  "drug"      "famili"   "$"
```

```
##  [2,] "game"      "state"      "art"    "test"      "children" "compani"
##  [3,] "team"      "law"        "street" "dr"        "mother"   "million"
##  [4,] "season"    "legal"      "museum" "medic"     "school"   "busi"
##  [5,] "coach"     "feder"      "song"   "patient"   "friend"   "money"
##  [6,] "leagu"     "court"      "artist" "doctor"    "hous"     "tax"
##  [7,] "point"     "medic"      "record" "studi"     "father"   "state"
##  [8,] "footbal"   "justic"     "paint"  "marijuana" "parent"   "percent"
##  [9,] "marijuana" "california" "west"   "health"    "son"      "york"
## [10,] "test"      "rule"       "album"  "research"  "life"     "market"
##       Topic 13 Topic 14 Topic 15
##  [1,] "park"   "polic"  "drug"
##  [2,] "citi"   "shot"   "school"
##  [3,] "town"   "kill"   "student"
##  [4,] "hous"   "gun"    "percent"
##  [5,] "island" "shoot"  "alcohol"
##  [6,] "$"      "offic"  "high"
##  [7,] "near"   "night"  "marijuana"
##  [8,] "away"   "car"    "heroin"
##  [9,] "long"   "death"  "program"
## [10,] "start"  "victim" "univers"
```

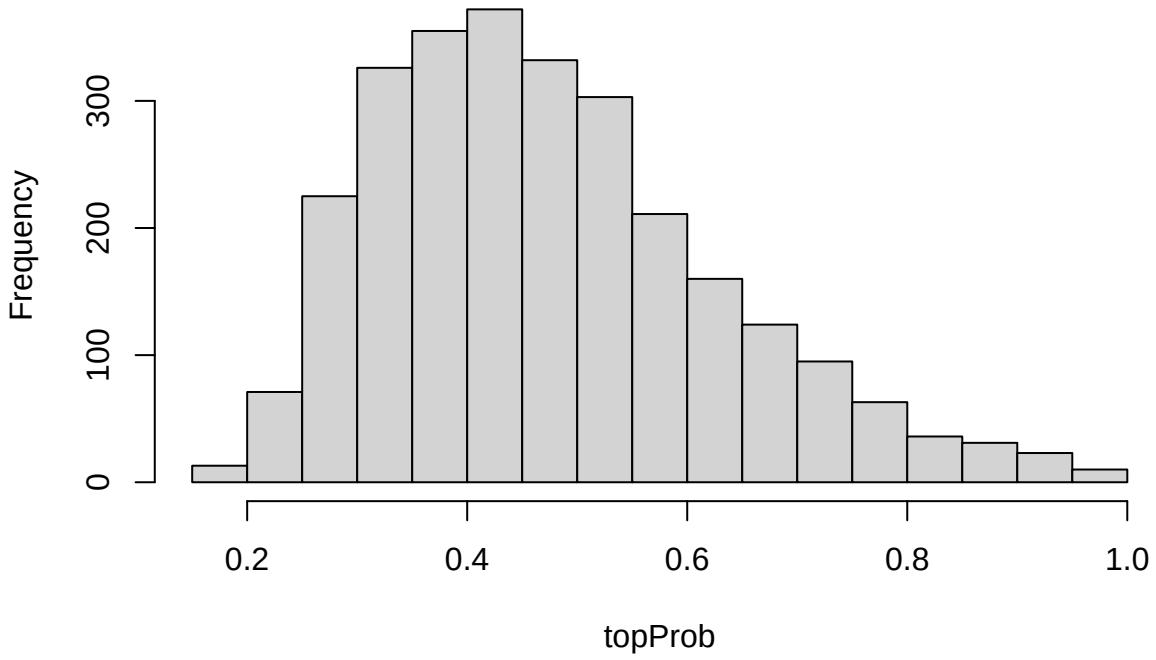
Step 7

Assign topics to each document then attach back onto original data.

```
##############################
# Gather Topic Probabilities #
##############################

topicProbabilities <- as.data.frame(fitted@gamma)

topProb <- apply(topicProbabilities, 1, max)
hist(topProb) # Decent
```




```
########################
# Extract Topics, etc. #
########################

ldaOut.topics <- as.data.frame(as.matrix(topics(fitted)))
ldaOut.topics$uniq_id <- row.names(ldaOut.topics)
colnames(ldaOut.topics)[1] <- "topic_15"
ldaOut.terms <- as.matrix(terms(fitted,6))

#####################################
# Merge Topic Model with Exist Datas  #
#####################################

final_out <- merge(final_out, ldaOut.topics, by.x="uniq_id", by.y="uniq_id", all.x=T)

table(final_out$topic_15)
```

```
##
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
## 155 177 306 192 296 234 237 210  80 136 122 122 145 155 183
```

```
############################
# Get Proportions by Year #
############################

tabs <- CrossTable(final_out$year, final_out$topic_15, prop.r=T, prop.c=F, prop.t=F, prop.chisq = F)$pr

# Clean #
tabs <- tabs[row.names(tabs)!="2007",]
```

Step 8

Plot it out over time.

```
############################
#     Initiate Plot        #
############################

plot(row.names(tabs), tabs[,5], type="n", ylim=c(0,.17), bty="n", lwd=3, # Legalization/medicinal
     ylab="Topic Percent of all articles",
     xlab= "Year",
     main = "Marijuana Newspaper Topic Model Across Time\n(NYT marijuana-related articles)")
lines(lowess(row.names(tabs), tabs[,7]), lty=1, lwd=3, col="blue") # Mexican Drug/Border
lines(lowess(row.names(tabs), tabs[,4]), lty=2, lwd=3, col="red") # Addiction
lines(lowess(row.names(tabs), tabs[,14]), lty=3, lwd=3, col="pink") # Courts
lines(lowess(row.names(tabs), tabs[,2]), lty=4, lwd=3, col="black") # Police/shooting/murder
lines(lowess(row.names(tabs), tabs[,9]), lty=5, lwd=3, col="green") # State Revenue/Tax
lines(lowess(row.names(tabs), tabs[,5]), lty=6, lwd=3, col="orange") # Legalization/Medicinal

legend("topright",
       bty="n",
       lty=1:6,
       lwd=3,
       cex=.7,
       legend=c("Mexico/Border", "Addiction", "Law and Courts",
                "Police/Shoot/Murder","State Revenue",
                "Legalization/Medicinal"),
       col=c("blue", "red", "pink", "black", "green",  "orange")
)
```