Estimating and visualizing an LDA topic model

14 May 2019

This document gives some examples of how to estimate and evaluate LDA in R. For theses example, you'll use the (English) speeches of EP group leaders that are part of the EUSpeech dataset.

NB: Use setwd() to set the working directory to the folder that contains English speeches in the file speeches_ep.csv. You will also need to download the `topicmodels` library using the `install.packages()` function:

```
Sys.setlocale(locale = "en_US.UTF-8")
```

```
## [1] "en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8"
```

```
#load libraries
library(dplyr)
library(readtext)
library(stringr)
library(topicmodels)
library(quanteda)
```

```
## Warning: package 'quanteda' was built under R version 3.5.2
```

```
library(ggplot2)

#read in the EP speeches
speeches <- read.csv(file = "speeches_ep.csv",
                     header = TRUE,
                     stringsAsFactors = FALSE,
                     sep = ",",
                     encoding = "UTF-8")

#let's do a bit of manual cleaning to remove some boiler plate terms
speeches$text <- str_replace_all(speeches$text, "ladies and gentlemen", " ")
speeches$text <- str_replace_all(speeches$text, "President", " ")
speeches$text <- str_replace_all(speeches$text, "Mr", " ")
speeches$text <- str_replace_all(speeches$text, "Council", " ")
speeches$text <- str_replace_all(speeches$text, "Commission", " ")

#concatenate the speeches
speeches <- speeches %>%
  group_by(speaker) %>%
  summarise(text = paste(text, collapse = " ")) %>%
  ungroup()

#create corpus object
speeches <- corpus(speeches)

#create a dfm
speeches.dfm <- dfm(speeches, stem = TRUE,
                    remove=stopwords("english"),
                    remove_punct=TRUE,
                    ngrams = 1,
                    remove_numbers = TRUE)
```

```r
#include only thoses features that occur in at least 5 documents
speeches.dfm <- dfm_trim(speeches.dfm, min_docfreq = 5)
```

Estimating an LDA topic model

```r
#convert the speeches dfm to a format that can be read in by the topicmodels library
speeches.lda.dfm <- convert(speeches.dfm, to = "topicmodels")

#set the seed to make the results replicable, since topic models are probabilistic
set.seed(2)

#estimate two topic models, one with 5 topics and one with 10 topics.
#This may take a few minutes, depending on your system
#Gibbs refers to the Gibbs sampler, a Bayesian approach to obtaining posterior parameter values
#k refers to the number of topics to be estimated; this is a parameter determined by the researcher
speeches.lda.5 <- LDA(speeches.lda.dfm,
                      method = "Gibbs",
                      k = 5)

speeches.lda.10 <- LDA(speeches.lda.dfm,
                       method = "Gibbs",
                       k = 10)
```

Take a look at the output of the topic model with 5 topics. For example, we can take a look at the 10 highest-loading terms for each of k topics.

```r
terms(speeches.lda.5, 10)
```

```
##       Topic 1   Topic 2      Topic 3    Topic 4    Topic 5
##  [1,] "eu"      "also"       "peopl"    "european" "european"
##  [2,] "report"  "like"       "european" "europ"    "need"
##  [3,] "countri" "european"   "want"     "must"     "think"
##  [4,] "vote"    "parliament" "eu"       "state"    "can"
##  [5,] "peopl"   "howev"      "europ"    "group"    "union"
##  [6,] "import"  "want"       "now"      "us"       "crisi"
##  [7,] "mani"    "europ"      "us"       "member"   "say"
##  [8,] "right"   "social"     "say"      "polit"    "let"
##  [9,] "union"   "say"        "go"       "treati"   "problem"
## [10,] "support" "must"       "union"    "market"   "also"
```

Question: How would you interpret these topics? Do you think they are meaningful topics? Why yes or why no?

```r
#the topics function shows you which topics load highest in each document
topics(speeches.lda.5, 10)
```

```
##      text1 text2 text3 text4 text5 text6 text7 text8 text9 text10 text11
## [1,]     1     4     2     3     4     2     4     5     2      2      4
## [2,]     4     1     5     5     2     1     2     2     5      3      2
## [3,]     2     2     3     2     1     4     1     4     4      5      5
## [4,]     3     5     4     1     5     5     3     3     1      4      1
## [5,]     5     3     1     4     3     3     5     1     3      1      3
##      text12 text13 text14 text15 text16 text17 text18 text19 text20 text21
## [1,]      1      2      3      3      2      1      2      3      3      2
```

```
## [2,]      4      5      1      4      4      4      4      5      2      5
## [3,]      2      4      5      5      5      2      5      2      4      4
## [4,]      3      1      2      2      3      5      1      4      5      1
## [5,]      5      3      4      1      1      3      3      1      1      3
##       text22
## [1,]      3
## [2,]      5
## [3,]      4
## [4,]      1
## [5,]      2
```

```r
#topic proportions for each document in speeches.lda.5 are saved in posterior(speeches.lda.5)$topics, w
posterior(speeches.lda.5)$topics
```

```
##                 1          2          3          4          5
## text1  0.34298611 0.14833333 0.12256944 0.28840278 0.09770833
## text2  0.30276879 0.20699988 0.03499940 0.37444564 0.08078629
## text3  0.03501904 0.39129285 0.14457258 0.12309564 0.30601989
## text4  0.04083589 0.05246677 0.77594581 0.03917434 0.09157720
## text5  0.15120058 0.26656436 0.09216465 0.35972197 0.13034844
## text6  0.20449717 0.39022663 0.04868980 0.19458215 0.16200425
## text7  0.16898263 0.16978908 0.14900744 0.43188586 0.08033499
## text8  0.02331377 0.13415316 0.07923421 0.10948854 0.65381032
## text9  0.06830179 0.56227945 0.06465630 0.12190498 0.18285748
## text10 0.11570976 0.28214160 0.21081367 0.19355407 0.19778091
## text11 0.06103380 0.23287773 0.04463221 0.53841948 0.12303678
## text12 0.57409038 0.14150528 0.09272300 0.14531984 0.04636150
## text13 0.12621661 0.43145579 0.05342721 0.18916960 0.19973079
## text14 0.23148148 0.15740741 0.24074074 0.14814815 0.22222222
## text15 0.10520621 0.13327393 0.41598778 0.19055499 0.15497709
## text16 0.05514899 0.50154497 0.07475470 0.23315444 0.13539690
## text17 0.50083565 0.18394530 0.02476576 0.20906559 0.08138769
## text18 0.11783321 0.41260238 0.06450112 0.26535741 0.13970588
## text19 0.04399415 0.07273867 0.73515772 0.05055358 0.09755588
## text20 0.04797048 0.21771218 0.32287823 0.21586716 0.19557196
## text21 0.08195054 0.47998073 0.06755165 0.13697677 0.23354031
## text22 0.10046624 0.06707776 0.33117762 0.23417055 0.26710784
```

```r
#confirm that the topic proportions add up to 1 for each document:
rowSums(posterior(speeches.lda.5)$topics)
```

```
##  text1  text2  text3  text4  text5  text6  text7  text8  text9 text10
##      1      1      1      1      1      1      1      1      1      1
## text11 text12 text13 text14 text15 text16 text17 text18 text19 text20
##      1      1      1      1      1      1      1      1      1      1
## text21 text22
##      1      1
```

Visualizing a LDA topic model

Let's say we are interested in a crisis topic. Let's measure this topic for each document by summing topic proportions of topics that contain the word `crisi` in the 10 topic LDA model:

```r
#locate in which topics `crisi` appears
crisis.topics <- which(terms(speeches.lda.10, 10) == 'crisi', arr.ind=TRUE)[,2]
```

```r
print(crisis.topics)
```

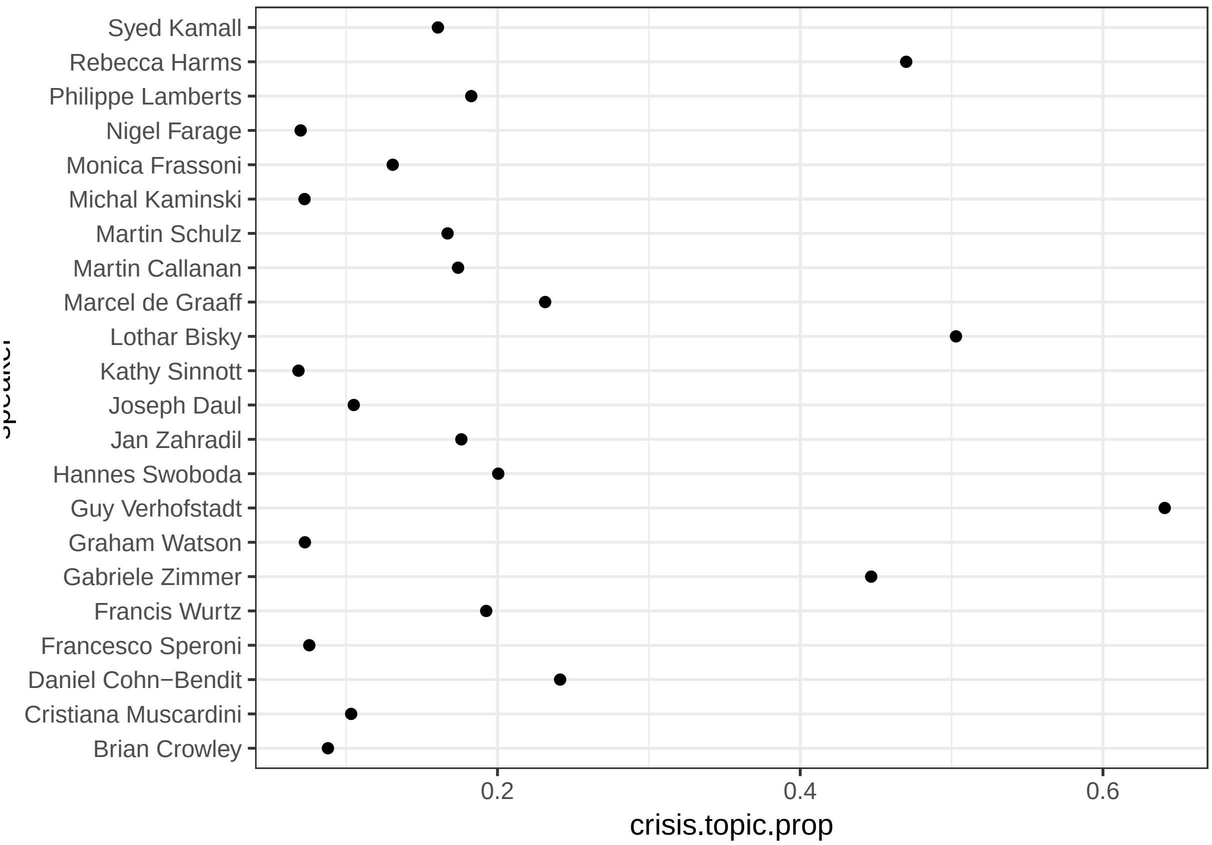
```
## [1] 7 8
```

```r
#add up topic proportions of crisis topics for each document, and save as docvar to the speeches.dfm ob
docvars(speeches.dfm, 'crisis.topic.prop') <- rowSums(posterior(speeches.lda.10)$topics[, crisis.topics
```

Let's plot the crisis topic for each EP leader:

```r
#change the document names to the speaker names
docnames(speeches.dfm) <- docvars(speeches.dfm, "speaker")

topic.plot <- ggplot(docvars(speeches.dfm),
                     aes(x= crisis.topic.prop,
                         y = speaker))
topic.plot <- topic.plot + geom_point() + theme_bw()
print(topic.plot)
```



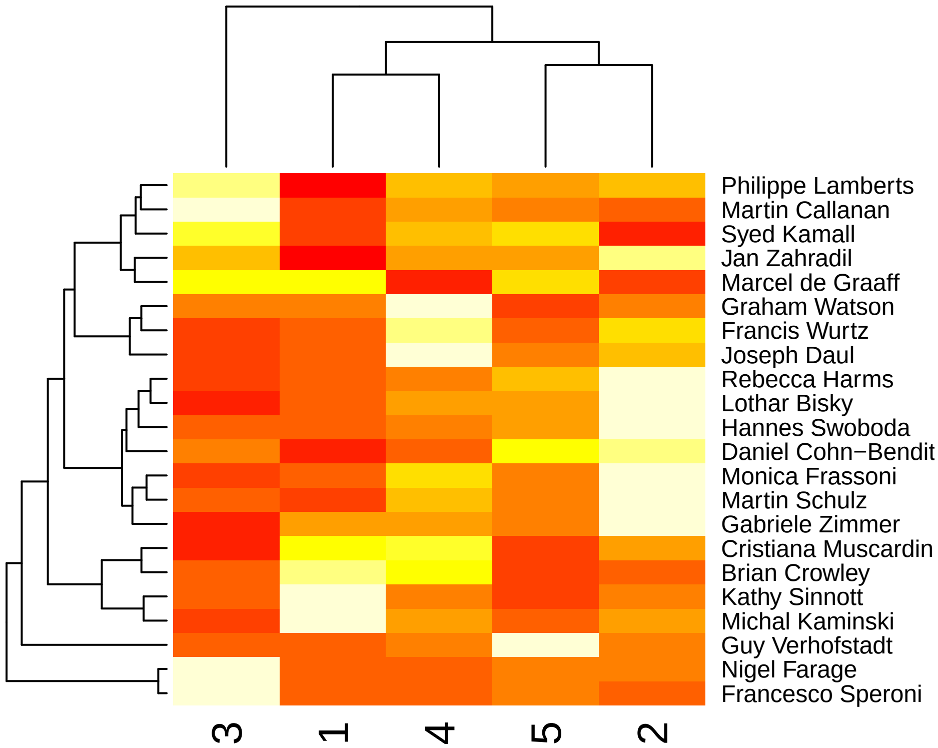
Question: Do you think this is a good way of summarizing a topic model? Why yes or why no?

Take a look at topic proportions for each speaker

```r
#append the topic proportions

topic.proportions <- posterior(speeches.lda.5)$topics
rownames(topic.proportions) <- rownames(speeches.dfm)

heatmap(as.matrix(topic.proportions[]))
```



In a heatmap, darker colors correspond with higher proportions, whereas lighter colors denote lower proportions. In addition, it displays a clustering of speakers and topics? How would you interpret this heatmap? Do you find this visualization useful?