

Topic Models

Tom Paskhalis

31 January, 2019

```
library("readr")
library("dplyr")
library("quanteda")
library("topicmodels")
library("stm")
```

US Senate

To fit topic models, we will restrict our analysis to US Senate. First, it makes our corpus smaller and, thus, speed up estimation process. And, second, it contains some covariates that we might be interested in when fitting structural topic models. Let us first read in the datasets and combine them together as in the previous part.

```
# Senate
us_senate_2017 <- readr::read_csv("../data/us-senate-2017.csv.gz")
us_senate_2018 <- readr::read_csv("../data/us-senate-2018.csv.gz")

senate115 <- us_senate_2017 %>%
  dplyr::bind_rows(us_senate_2018)

nrow(senate115)

## [1] 38878

head(senate115, 10)

## # A tibble: 10 x 16
##   chamber speaker date      text first_name last_name party gender
##   <chr>   <chr>   <date>   <chr> <chr>      <chr>   <chr> <chr>
## 1 S      The VI~ 2017-01-03 The ~ <NA>      <NA>      <NA>   <NA>
## 2 S      Mr COR~ 2017-01-03 Mr. ~ John    cornyn    Repu~ M
## 3 S      The PR~ 2017-01-03 The ~ <NA>      <NA>      <NA>   <NA>
## 4 S      Mr DUR~ 2017-01-03 Mr. ~ Richard  durbin    Demo~ M
## 5 S      Mr MER~ 2017-01-03 Mr. ~ Jeff    merkley    Demo~ M
## 6 S      The PR~ 2017-01-03 The ~ <NA>      <NA>      <NA>   <NA>
## 7 S      Mr McC~ 2017-01-03 Mr. ~ Mitch    mcconnell Repu~ M
## 8 S      The PR~ 2017-01-03 With~ <NA>      <NA>      <NA>   <NA>
## 9 S      Mr DUR~ 2017-01-03 Mr. ~ Richard  durbin    Demo~ M
## 10 S     The PR~ 2017-01-03 The ~ <NA>      <NA>      <NA>   <NA>
## # ... with 8 more variables: birthday <date>, state <chr>, url <chr>,
## #   twitter <chr>, facebook <chr>, govtrack_id <dbl>, icpsr_id <dbl>,
## #   votesmart_id <dbl>
```

After inspecting the dataset, we can see that a lot of the rows contain procedural statements by presiding officers of the Senate. As we might be interested in the topical content of the speeches, rather than procedural discussion, we can remove those:

```
senate115 <- senate115 %>%
  dplyr::filter(!is.na(first_name))

nrow(senate115)
```

```
## [1] 20683
```

```
head(senate115, 10)
```

```
## # A tibble: 10 x 16
##   chamber speaker date      text first_name last_name party gender
##   <chr>   <chr>   <date>   <chr> <chr>      <chr>   <chr> <chr>
## 1 S      Mr COR~ 2017-01-03 Mr. ~ John      cornyn    Repu~ M
## 2 S      Mr DUR~ 2017-01-03 Mr. ~ Richard   durbin    Demo~ M
## 3 S      Mr MER~ 2017-01-03 Mr. ~ Jeff      merkley    Demo~ M
## 4 S      Mr McC~ 2017-01-03 Mr. ~ Mitch     mcconnell Repu~ M
## 5 S      Mr DUR~ 2017-01-03 Mr. ~ Richard   durbin    Demo~ M
## 6 S      Mr COR~ 2017-01-03 Mr. ~ Bob       corker     Repu~ M
## 7 S      Mr PET~ 2017-01-03 Mr. ~ Gary      peters     Demo~ M
## 8 S      Mr MOR~ 2017-01-03 Mr. ~ Jerry     moran      Repu~ M
## 9 S      Mr McC~ 2017-01-03 Mr. ~ Mitch     mcconnell Repu~ M
## 10 S     Mr McC~ 2017-01-03 Mr. ~ Mitch     mcconnell Repu~ M
## # ... with 8 more variables: birthday <date>, state <chr>, url <chr>,
## #   twitter <chr>, facebook <chr>, govtrack_id <dbl>, icpsr_id <dbl>,
## #   votesmart_id <dbl>
```

Although we lost some observations, it is still a quite sizeable dataset. Now, we can proceed with creating a corpus and dfm in the usual way.

```
corpus115 <- quanteda::corpus(senate115)
head(quanteda::docvars(corpus115), 10)
```

```
##      chamber      speaker      date first_name last_name      party
## text1         S    Mr CORNYN 2017-01-03      John      cornyn Republican
## text2         S    Mr DURBIN 2017-01-03    Richard      durbin  Democrat
## text3         S  Mr MERKLEY 2017-01-03      Jeff    merkley  Democrat
## text4         S Mr McCONNELL 2017-01-03      Mitch mcconnell Republican
## text5         S    Mr DURBIN 2017-01-03    Richard      durbin  Democrat
## text6         S    Mr CORKER 2017-01-03      Bob      corker Republican
## text7         S    Mr PETERS 2017-01-03      Gary    peters  Democrat
## text8         S    Mr MORAN 2017-01-03      Jerry     moran Republican
## text9         S Mr McCONNELL 2017-01-03      Mitch mcconnell Republican
## text10        S Mr McCONNELL 2017-01-03      Mitch mcconnell Republican
##      gender  birthday state      url
## text1      M 1952-02-02   TX https://www.cornyn.senate.gov
## text2      M 1944-11-21   IL https://www.durbin.senate.gov
## text3      M 1956-10-24   OR https://www.merkley.senate.gov
## text4      M 1942-02-20   KY https://www.mcconnell.senate.gov
## text5      M 1944-11-21   IL https://www.durbin.senate.gov
## text6      M 1952-08-24   TN https://www.corker.senate.gov
## text7      M 1958-12-01   MI https://www.peters.senate.gov
## text8      M 1954-05-29   KS https://www.moran.senate.gov
## text9      M 1942-02-20   KY https://www.mcconnell.senate.gov
## text10     M 1942-02-20   KY https://www.mcconnell.senate.gov
##      twitter      facebook govtrack_id icpsr_id votesmart_id
## text1 JohnCornyn sen.johncornyn      300027      40305      15375
```

## text2	SenatorDurbin	SenatorDurbin	300038	15021	26847
## text3	SenJeffMerkley	jeffmerkley	412325	40908	23644
## text4	McConnellPress	mitchmcconnell	300072	14921	53298
## text5	SenatorDurbin	SenatorDurbin	300038	15021	26847
## text6	SenBobCorker	bobcorker	412248	40705	65905
## text7	SenGaryPeters	SenGaryPeters	412305	20923	8749
## text8	JerryMoran	jerrymoran	400284	29722	542
## text9	McConnellPress	mitchmcconnell	300072	14921	53298
## text10	McConnellPress	mitchmcconnell	300072	14921	53298

```
summary(corpus115, 10)
```

```
## Corpus consisting of 20683 documents, showing 10 documents:
```

```
##
##      Text Types Tokens Sentences chamber      speaker      date first_name
##      text1   592   1917         69      S      Mr CORNYN 2017-01-03      John
##      text2   781   2542        137      S      Mr DURBIN 2017-01-03      Richard
##      text3   635   2334         91      S      Mr MERKLEY 2017-01-03      Jeff
##      text4    16     18          1      S Mr McCONNELL 2017-01-03      Mitch
##      text5     64    102          7      S      Mr DURBIN 2017-01-03      Richard
##      text6  1450   6749        258      S      Mr CORKER 2017-01-03      Bob
##      text7   303   667          30      S      Mr PETERS 2017-01-03      Gary
##      text8   288   640          26      S      Mr MORAN 2017-01-03      Jerry
##      text9    41     57           2      S Mr McCONNELL 2017-01-03      Mitch
##      text10   51     70           1      S Mr McCONNELL 2017-01-03      Mitch
##      last_name      party gender  birthday state
##      cornyn Republican      M 1952-02-02    TX
##      durbin  Democrat      M 1944-11-21    IL
##      merkley Democrat      M 1956-10-24    OR
##      mcconnell Republican      M 1942-02-20    KY
##      durbin  Democrat      M 1944-11-21    IL
##      corker Republican      M 1952-08-24    TN
##      peters  Democrat      M 1958-12-01    MI
##      moran Republican      M 1954-05-29    KS
##      mcconnell Republican      M 1942-02-20    KY
##      mcconnell Republican      M 1942-02-20    KY
##
##      url      twitter      facebook
##      https://www.cornyn.senate.gov      JohnCornyn sen.johncornyn
##      https://www.durbin.senate.gov      SenatorDurbin SenatorDurbin
##      https://www.merkley.senate.gov      SenJeffMerkley jeffmerkley
##      https://www.mcconnell.senate.gov      McConnellPress mitchmcconnell
##      https://www.durbin.senate.gov      SenatorDurbin SenatorDurbin
##      https://www.corker.senate.gov      SenBobCorker      bobcorker
##      https://www.peters.senate.gov      SenGaryPeters      SenGaryPeters
##      https://www.moran.senate.gov      JerryMoran      jerrymoran
##      https://www.mcconnell.senate.gov      McConnellPress mitchmcconnell
##      https://www.mcconnell.senate.gov      McConnellPress mitchmcconnell
##      govtrack_id icpsr_id votesmart_id
##      300027      40305      15375
##      300038      15021      26847
##      412325      40908      23644
##      300072      14921      53298
##      300038      15021      26847
##      412248      40705      65905
##      412305      20923      8749
```

```
##          400284      29722          542
##          300072      14921         53298
##          300072      14921         53298
##
## Source: /home/tpaskhalis/Decrypted/Git/VAM_Text_Analysis/code/* on x86_64 by tpaskhalis
## Created: Thu Jan 31 13:58:29 2019
## Notes:
```

As some speeches might be very short and not very informative, let us first trim the corpus by applying `corpus_trim()` function.

```
pre <- quanteda::ndoc(corpus115)

corpus115 <- corpus115 %>%
  quanteda::corpus_trim(what = "documents", min_ntoken = 10)

post <- quanteda::ndoc(corpus115)
c(pre, post, pre-post)
```

```
## [1] 20683 18955 1728
```

To make the model less computationally expensive, we will reduce the number of features by stemming the tokens.

```
dfm115 <- quanteda::dfm(corpus115,
  tolower = TRUE,
  stem = TRUE,
  remove = stopwords("english"),
  remove_punct = TRUE)
```

Before fitting the model, let us further trim the dataset by removing infrequent tokens. To do that, we will be using `dfm_trim()` function. There are several options to trim the dfm. One, which we are using here is to specify the minimum number of documents in which a given token occurs (`min_docfreq`). Another would be to specify the minimum number of times a token should be used across all the documents (`min_termfreq`) to remain in the dfm.

```
dfm115 <- quanteda::dfm_trim(dfm115, min_docfreq = 2)
```

Latent Dirichlet Allocation (LDA)

Let us start with the original implementation of topic models, also called Latent Dirichlet Allocation (or LDA for short). Another way to think about a topic model is as Bayesian mixed-membership. If you have encountered mixture models before, where each observed unit (say, an individual) belongs to a latent class, here we allow each observed unit (document) to belong to multiple classes.

We will be using the package `topicmodels` and function `LDA()`. This is essentially an R wrapper around C code, implemented by the authors of LDA.

The crucial analytical decision to be made when fitting a topic model is to specify a number of topics (k). Here, we will just pick 10 as a starting value and then come back to diagnostics at a later stage. To run MCMC sampler we specify the `burnin` period of 100 iterations, that are discarded from the analysis of resultant chains and keep the remaining 500 (It is a relatively low number and in real-life analysis, it is better to have a few thousand iterations). The parameter `verbose` is just an integer indicating the number of iterations after which the output gets updated while the model is running.

```
k <- 10
lda <- topicmodels::LDA(dfm115,
```

```

k = k,
method = "Gibbs",
control = list(verbose=25L,
               seed = 123,
               burnin = 100,
               iter = 500))

```

Instead of using more traditional Gibbs sampling for Bayesian estimation, we can also try variational inference (VEM). Experiment with this. Mind that corpus is still considerably large. It might take some time for this model to converge!

```

k <- 10
lda <- topicmodels::LDA(dfm115,
                       k = k,
                       method = "VEM")

```

After fitting the model, we can inspect the top *n* terms from the model with `get_terms()` function and predict top *k* topics for each document with `get_topics()` function.

```
topicmodels::terms(lda, 10)
```

```

##      Topic 1  Topic 2 Topic 3  Topic 4  Topic 5  Topic 6
## [1,] "presid"  "go"    "tax"    "defens" "senat"  "act"
## [2,] "judg"    "peopl" "american" "nation" "mr"     "section"
## [3,] "court"   "get"   "busi"    "secur"  "presid" "1"
## [4,] "senat"   "want"  "bill"    "support" "committe" "state"
## [5,] "law"     "say"   "job"     "militari" "ask"     "2"
## [6,] "nomin"   "know"  "compani" "u."     "unanim"  "shall"
## [7,] "nomine"  "one"   "percent" "unit"   "consent" "b"
## [8,] "justic"  "just"  "make"    "system" "order"   "includ"
## [9,] "confirm" "think" "year"    "state"  "session" "may"
## [10,] "vote"   "us"    "work"    "forc"   "vote"    "committe"
##      Topic 7  Topic 8  Topic 9  Topic 10
## [1,] "school"  "state"  "countri" "care"
## [2,] "educ"    "nation" "law"     "health"
## [3,] "serv"    "water"  "state"   "insur"
## [4,] "year"    "chang"  "american" "healthcar"
## [5,] "work"    "climat" "right"   "bill"
## [6,] "student" "energi" "protect" "peopl"
## [7,] "communiti" "industri" "presid" "afford"
## [8,] "state"    "protect" "peopl"  "american"
## [9,] "servic"   "year"    "children" "state"
## [10,] "public"  "administr" "unit"   "act"

```

```
head(topicmodels::topics(lda, 1), 10)
```

```

##  text1  text2  text3  text4  text5  text6  text7  text8  text9 text10
##      2      2      2      5      2      4      7      4      5      5

```

Structural Topic Models (STM)

The original approach for topic modelling did not allow for the topical content to depend on any of the document covariates. Structural topic models introduced the possibility to incorporate this metadata into the estimation process. Here we will be using `stm` package and the function with the same name: `stm()`. Let us start with incorporating gender as a covariate.

```
stm115 <- stm::stm(dfm115, K = k, data = docvars(dfm115), prevalence = ~ gender)
```

To view the top terms by various statistics we can use `laelTopics()` function:

```
stm::labelTopics(stm115, n = 10)
```

```
## Topic 1 Top Words:
```

```
## Highest Prob: school, year, work, serv, educ, state, student, communiti, servic, famili
## FREX: selfless, championship, ywca, devo, 1943, patterson, monson, thad, museum, smithsonian
## Lift: 1,000-mile, 1,177, 1.45, 10,000th, 105th, 114-265, 116th, 125th, 130th, 14-15
## Score: school, student, devo, love, educ, veteran, teacher, betsi, mani, graduat
```

```
## Topic 2 Top Words:
```

```
## Highest Prob: senat, mr, presid, ask, unanim, consent, order, amend, committe, motion
## FREX: adjourn, 1628, bloc, yea, nay, reconsid, motion, rescind, unanim, consent
## Lift: 1628, 1007, 1032, 1033, 1038, 1039, 1055, 1057, 1065, 1082
## Score: consent, unanim, motion, rescind, p.m, 1628, adjourn, h.r, session, reconsid
```

```
## Topic 3 Top Words:
```

```
## Highest Prob: section, shall, committe, 1, act, state, author, b, 2, unit
## FREX: subsect, u.s.c, subparagraph, outlay, p.l, n.a, seq, paragraph, sec, subclaus
## Lift: 1351, 1396a, 2,280,970, 2,281,616, 2017-2026, 303, 715,835, prereleas, subclaus, p.l
## Score: subsect, subparagraph, shall, u.s.c, section, b, paragraph, sec, outlay, p.l
```

```
## Topic 4 Top Words:
```

```
## Highest Prob: state, water, climat, energi, year, chang, epa, nation, industri, just
## FREX: epa, pruit, mercuri, solar, greenhous, dioxid, coal, fossil, climat, oil
## Lift: 111,000, 18.7, 20-to-1, 2075, 222nd, 36.5, 999, absorpt, agronomi, airboat
## Score: epa, pruit, climat, pollut, fossil, carbon, farmer, wildlif, farm, emiss
```

```
## Topic 5 Top Words:
```

```
## Highest Prob: judg, court, senat, presid, nomine, nomin, law, vote, justic, suprem
## FREX: kavanaugh, gorsuch, suprem, judg, nomine, court, scalia, ford, circuit, judici
## Lift: 101-year, 102,000, 182,000, 228-year, 230-year, 290-plus, 30-plus-year-old, 4-4, 4-to-4,
## Score: judg, kavanaugh, gorsuch, court, nomine, suprem, justic, nomin, judici, circuit
```

```
## Topic 6 Top Words:
```

```
## Highest Prob: committe, investig, general, member, inform, attorney, intellig, report, elect,
## FREX: comey, haspel, rosenstein, cia, fda, russian, interfer, mueller, transcript, investig
## Lift: 39-minut, 514.110, 6,700-page, 90-9, abd, al-nashiri, alfa-bank, anada, anda, archibald
## Score: russian, investig, fda, russia, fbi, comey, intellig, attorney, cia, trump
```

```
## Topic 7 Top Words:
```

```
## Highest Prob: presid, peopl, countri, state, us, one, unit, american, senat, go
## FREX: daca, dreamer, iran, sanction, zte, china, syrian, putin, backpack, refuge
## Lift: 1,200-mile, 10-day, 2,342, 2,370, 32-year-old, 51-49, 62-page, 790,000, 800-percent, 846
## Score: daca, putin, dreamer, peopl, immigr, trump, just, say, russia, iran
```

```
## Topic 8 Top Words:
```

```
## Highest Prob: defens, support, system, militari, internet, propos, servic, sale, u., forc
## FREX: hardwar, mde, non-md, herewith, warhead, mk, launcher, ajit, isp, low-yield
## Lift: 1.06, 1.3b, 100.0, 10514, 15-70, 1b, 2,500-6,000, 22m, 24-channel, 250-lb
## Score: missil, mde, non-md, herewith, transmitt, fcc, internet, aircraft, vii, softwar
```

```
## Topic 9 Top Words:
```

```
## Highest Prob: bill, provid, feder, program, support, state, busi, work, act, legisl
## FREX: cfpb, loan, dodd-frank, bank, lender, consum, onewest, financi, cra, workforc
## Lift: assigne, osha, piowar, onewest, 12.50, 13.2, 1504, 2009-2011, 2216, 23.688
## Score: bank, worker, regul, program, financi, consum, loan, cfpb, veteran, dodd-frank
```

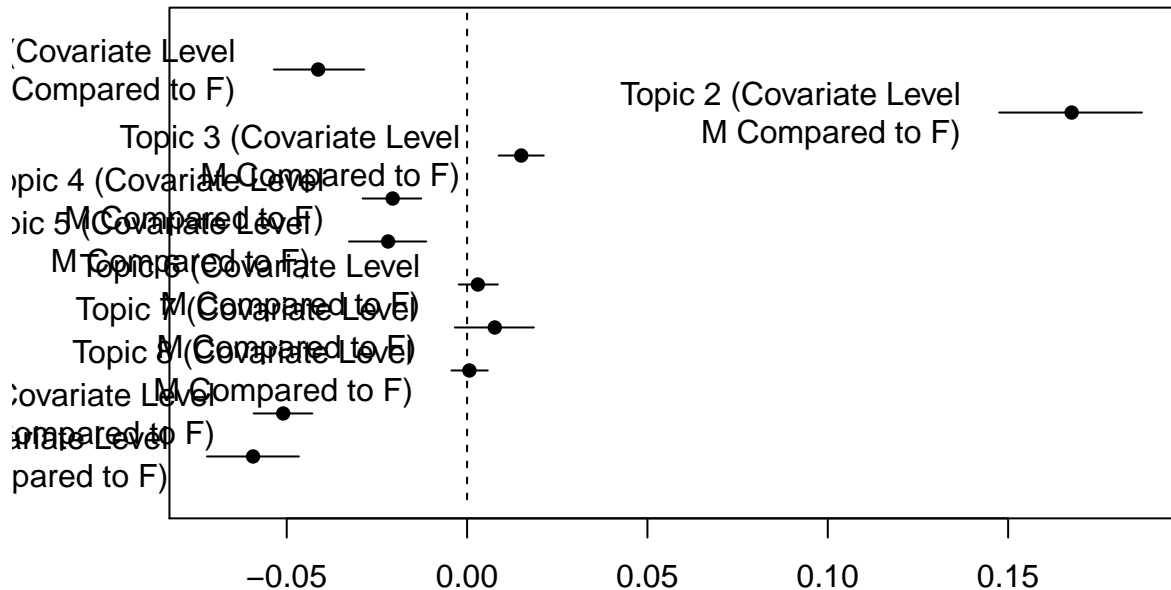
```
## Topic 10 Top Words:
```

```
## Highest Prob: peopl, tax, go, bill, get, american, care, health, year, insur
## FREX: obamacar, trumpcar, medicaid, tax, healthcar, insur, premium, afford, medicar, cut
```

```
## Lift: 0-percent, 1.9-percent, 10.5-percent, 12,900, 12.9, 14,600, 14504, 16-bed, 16-percent, 1
## Score: tax, medicaid, insur, obamacare, healthcar, get, peopl, go, premium, medicar
```

To plot the estimated effect of gender on the topics, we can use `estimateEffect()` function from the `stm` package and an in-built plot method for the resultant object.

```
md115 <- stm::estimateEffect(1:10 ~ gender, stmobj = stm115, metadata = docvars(dfm115))
plot(md115, "gender", cov.value1 = "M", cov.value2 = "F", method = "difference")
```



A few other useful functions in the `stm` package are `searchK()` for the diagnostics of the number of topics, `topicQuality()` for assessing the quality of the model fit. See the examples below:

```
# Before we can proceed with using searchK, we need to prepare our dfm.
dfm115stm <- quanteda::convert(dfm115, to = "stm", docvars = docvars(dfm115))
kdiag <- searchK(documents = dfm115stm[["documents"]],
                 vocab = dfm115stm[["vocab"]],
                 K = seq(5,50,5))
plot(kdiag)

topicQuality(stm115, documents = quanteda::convert(dfm115, to = "stm"))
```

Challenge 3

Easy mode Experiment with LDA by fitting it with a different number of topics and observing how it affects the top terms.

Medium Calculate age in years for each senator and use it alongside gender as a covariate for topic models. Use `lubridate` package for calculating the age.

Advanced Produce a coefficients plot for the estimated model. Try `ggplot2` package to make it appear nicer.