Quantitative Content Analysis: Lecture 12

Matthias Haber

03 May 2017

Today's outline

- Final project expectations
- Topic Models
- Wrapping up

Final project expectations

- Develop an interesting (non-trivial) research question **(10)**
    - Trivial: Did the CDU move to the left under Angela Merkel?
    - Less trivial: Did the CDU move to the left on every issue under Angela Merkel?
    - Non-trivial: Is there a general shift to the left among conservative parties in Europe on all policy issues?
    - The RQ does not need to contain a causal link (but it can)
    - Motivate your RQ
- Collect appropriate (text) data to answer your research question **(25)**
    - Data needs to fit the research question (explain why it does)
    - Trivial: Use build-in datasets in R (e.g. inauguration speeches)
    - Less trivial: Download or access pre-existing datasets (e.g. manifestos)
    - Non-trivial: Construct your own data set (e.g. Twitter, News Media, Press releases)

Final project expectations (II)

- Prepare your data set for analysis **(15)**
  - Create a corpus
  - Perform the necessary pre-processing steps and create a dfm
- Specify and apply a computarized text analysis method **(25)**
  - must be a variant of a dictionary, wordscore, wordfish, or topic modelling
- Present and discuss your results **(25)**
  - Appropriate graphical presentations of results (watch out for the quality of graphs)

Final project expectations (III)

Deliverables

- ~8 pages (more is not an issue) detailing research question, data set, model specification, analysis and results
- Replication code and data set
- team-work is encouraged
- Due 15 May

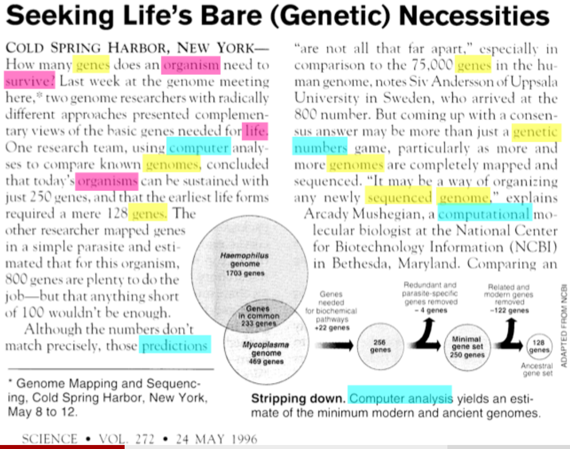
Topic models: basic idea

We often have collections of documents that we'd like to divide into natural groups so that we can understand them separately. Topic modeling is a method for unsupervised classification of such documents, which finds natural groups of items even when we're not sure what we're looking for.

- Topic models are exploratory probability models that
  - weaken the contraints required in dictionary based content analysis
  - have been intensively studied in the computer science literature
- Topic models work best with large amounts of text with a thematic structure

Topic models: LDA

Latent Dirichlet allocation (LDA) is a popular method for fitting a topic model. It treats each document as a mixture of topics, and each topic as a mixture of words. LDA estimats both of these at the same time.



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

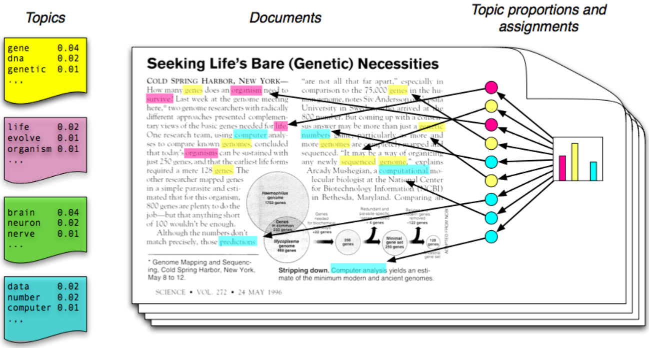
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed − 4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

SCIENCE • VOL. 272 • 24 MAY 1996

Topic models: LDA (II)


Topics
Documents
Topic proportions and assignments
gene 0.04
dna 0.02
genetic 0.01
...
life 0.02
evolve 0.01
organism 0.01
...
brain 0.04
neuron 0.02
nerve 0.01
...
data 0.02
number 0.02
computer 0.01
...
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed −4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set
ADAPTED FROM NCBI

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK— How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

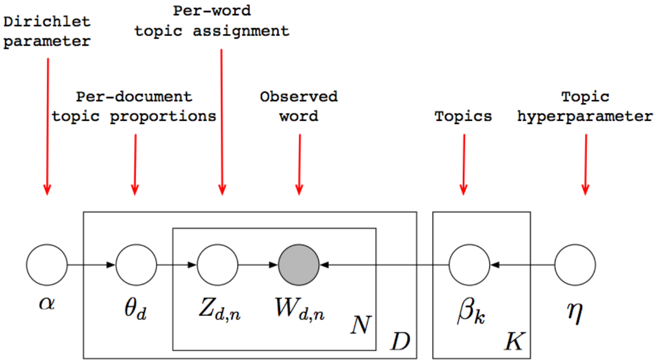
Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

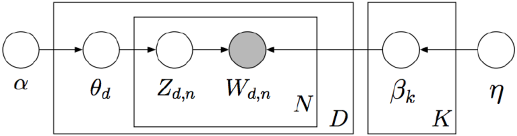
SCIENCE • VOL. 272 • 24 MAY 1996

Topic models: LDA (III)



Each piece of the structure is a random variable.

Topic models: LDA (IV)



1. Draw each topic $\beta_k \sim \mathrm{Dir}(\eta)$, for $k \in \{1, \ldots, K\}$.
2. For each document:
    1. Draw topic proportions $\theta_d \sim \mathrm{Dir}(\alpha)$.
    2. For each word:
        1. Draw $Z_{d,n} \sim \mathrm{Mult}(\theta_d)$.
        2. Draw $W_{d,n} \sim \mathrm{Mult}(\beta_{z_{d,n}})$.

Topic model: LDA (V)

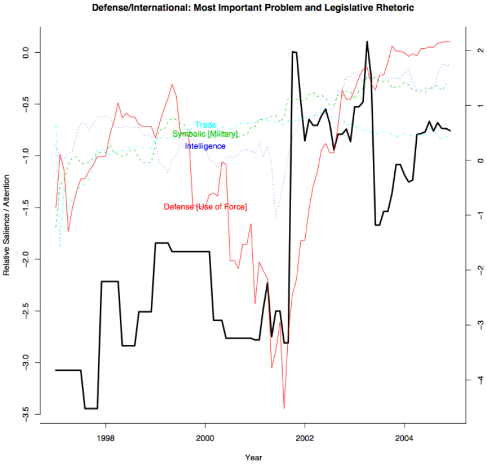
- Topic models giveth:
  - a probabilistic view of the relationship between W, Z and θ
  - a full statistical framework for learning most aspects of the relationship
- and taketh away:
  - substantive control: You do not get to assert what the topics mean (inevitable when the Z and θ are both unobserved)

Topic model: LDA (VI)

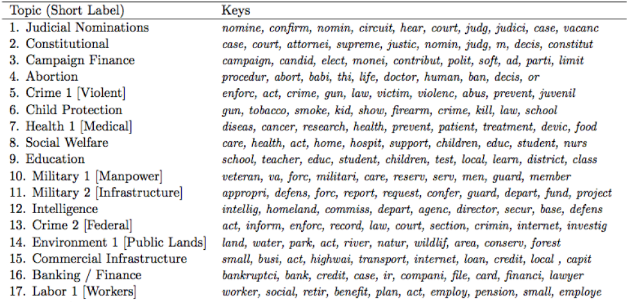
- What control is left?
  - α the parameter of a Dirichlet prior for the distributions (not number) of topics. The closer α is to 0 the more each document will tend to contain instances of fewer rather than more topics
  - η the parameter of a Dirichlet prior over the distributions of words in topics. The closer η is to 0 the more a topic will generate fewer words with high probability.
- Roughly, larger values allow more variation and less sparse representations
- Topic models are admixtures: mixtures of mixtures

Application: policy agenda

- Quinn et al. analyze 118,065 congressional speeches from 1997-2004.



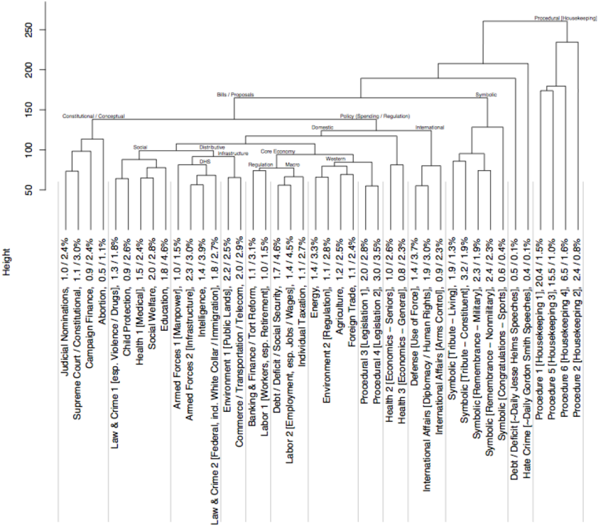
Defining topics



Topic (Short Label)	Keys
1. Judicial Nominations	*nomine, confirm, nomin, circuit, hear, court, judg, judici, case, vacanc*
2. Constitutional	*case, court, attornei, supreme, justic, nomin, judg, m, decis, constitut*
3. Campaign Finance	*campaign, candid, elect, monei, contribut, polit, soft, ad, parti, limit*
4. Abortion	*procedur, abort, babi, thi, life, doctor, human, ban, decis, or*
5. Crime 1 [Violent]	*enforc, act, crime, gun, law, victim, violenc, abus, prevent, juvenil*
6. Child Protection	*gun, tobacco, smoke, kid, show, firearm, crime, kill, law, school*
7. Health 1 [Medical]	*diseas, cancer, research, health, prevent, patient, treatment, devic, food*
8. Social Welfare	*care, health, act, home, hospit, support, children, educ, student, nurs*
9. Education	*school, teacher, educ, student, children, test, local, learn, district, class*
10. Military 1 [Manpower]	*veteran, va, forc, militari, care, reserv, serv, men, guard, member*
11. Military 2 [Infrastructure]	*appropri, defens, forc, report, request, confer, guard, depart, fund, project*
12. Intelligence	*intellig, homeland, commiss, depart, agenc, director, secur, base, defens*
13. Crime 2 [Federal]	*act, inform, enforc, record, law, court, section, crimin, internet, investig*
14. Environment 1 [Public Lands]	*land, water, park, act, river, natur, wildlif, area, conserv, forest*
15. Commercial Infrastructure	*small, busi, act, highwai, transport, internet, loan, credit, local , capit*
16. Banking / Finance	*bankruptci, bank, credit, case, ir, compani, file, card, financi, lawyer*
17. Labor 1 [Workers]	*worker, social, retir, benefit, plan, act, employ, pension, small, employe*

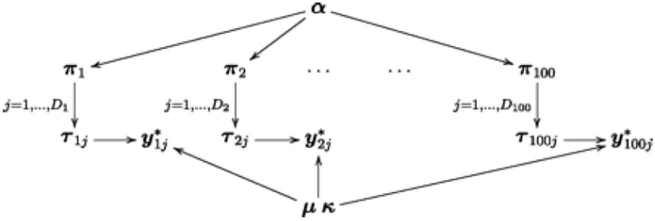
Defining topics (II)

Agglomerative Clustering of 42 Topic Model



Variations: expressed agenda model

In a simpler variation on LDA, Grimmer (2009) defines an expressed agenda model as



- Here there are not multiple topics per press release, but there are observed authors drawn from a population
- R: `install_github("christophergandrud/ExpAgenda")`

Variations: correlated topic models

- The Dirichlet multinomial assumptions hide a constraint about topic covariation
  - LDA cannot represent free covariation of topic proportions
  - The correlated topic model can
- Replace the Dirichlet with a Logistic Normal structure (Aitchison, 1986) with arbitrary covariance matrix
- R: `topicmodels`

Topic model example (from tidytextmining.com)

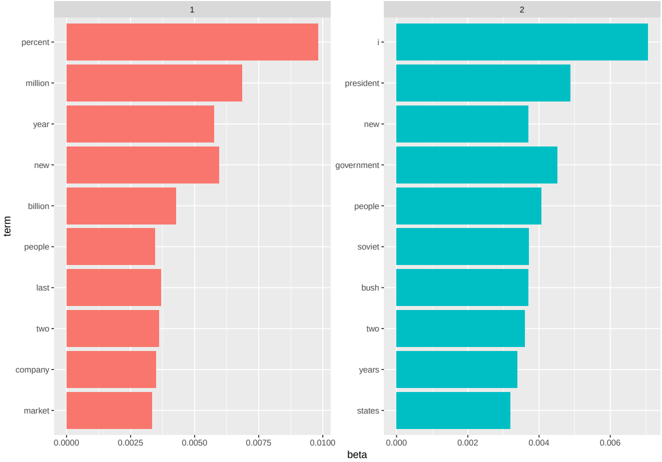
```r
library(topicmodels)
# browseVignettes('topicmodels')
data("AssociatedPress")
apLda <- LDA(AssociatedPress, control = list(seed = 1234), k = 2)
```

Word-topic probabilities

```r
# extract the per-topic-per-word probabilities ('beta')
library(tidytext)
library(ggplot2)
library(dplyr)
apTopics <- tidy(apLda, matrix = "beta")

# Top 10 terms that are most common within each topic
topTerms <- apTopics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
plotTerms <- topTerms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```

Word-topic probabilities (II)



Words that discriminate well

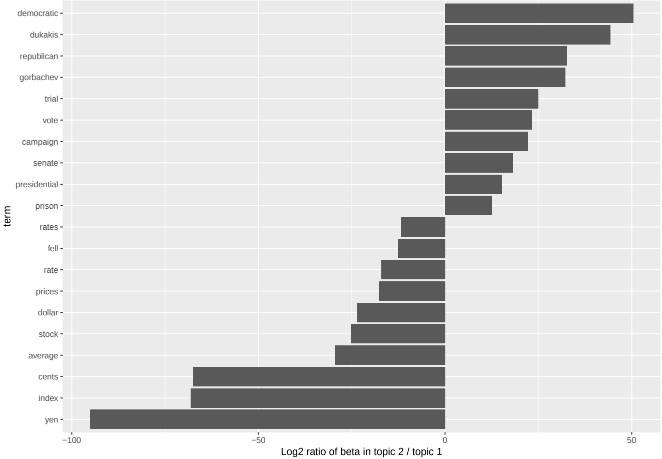
- Extract terms with the greatest difference in β between topic 1 and topic 2 using the log ratio $log_2(\frac{\beta_1}{\beta_2})$

```
library(tidyr)
betaSpread <- apTopics %>%
  mutate(topic = paste0("topic", topic)) %>%
  spread(topic, beta) %>%
  filter(topic1 > .001 | topic2 > .001) %>%
  mutate(log_ratio = log2(topic2 / topic1))
head(betaSpread)
## # A tibble: 6 × 4
##             term      topic1       topic2  log_ratio
##            <chr>       <dbl>        <dbl>      <dbl>
## 1 administration 4.309502e-04 0.0013822436  1.6814189
## 2            ago 1.065216e-03 0.0008421279 -0.3390353
## 3      agreement 6.714984e-04 0.0010390238  0.6297728
## 4            aid 4.759043e-05 0.0010459576  4.4580091
## 5            air 2.136933e-03 0.0002966593 -2.8486628
## 6       american 2.030497e-03 0.0016838838 -0.2700405
```

Words that discriminate well (II)

```
plotBetaSpread <- betaSpread %>%
  mutate(absratio = abs(log_ratio)) %>%
  group_by(direction = log_ratio > 0) %>%
  top_n(10, absratio) %>%
  ungroup() %>%
  mutate(term = reorder(term, log_ratio)) %>%
  ggplot(aes(term, log_ratio)) +
  geom_col() +
  labs(y = "Log2 ratio of beta in topic 2 / topic 1") +
  coord_flip()
```

Words that discriminate well (III)



Document-topic probabilities

- LDA models each document as a mixture of topics so we can examine the per-document-per-topic probabilities(γ)

```
apDocs <- tidy(apLda, matrix = "gamma")
head(apDocs)
## # A tibble: 6 × 3
##   document topic        gamma
##      <int> <int>        <dbl>
## 1        1     1 0.2480616686
## 2        2     1 0.3615485445
## 3        3     1 0.5265844180
## 4        4     1 0.3566530023
## 5        5     1 0.1812766762
## 6        6     1 0.0005883388
```

Document-topic probabilities (II)

```
# Inspect document 6
tidy(AssociatedPress) %>%
  filter(document == 6) %>%
  arrange(desc(count))
## # A tibble: 287 × 3
##    document           term count
##       <int>          <chr> <dbl>
## 1         6        noriega    16
## 2         6         panama    12
## 3         6        jackson     6
## 4         6         powell     6
## 5         6 administration     5
## 6         6       economic     5
## 7         6        general     5
## 8         6              i     5
## 9         6     panamanian     5
## 10        6       american     4
## # ... with 277 more rows
```

Topic model exercise

We will take another look at the US Senate debate on partial birth abortion.

1. Download the speeches from Moddle (if you use Github you should have them inside the Week12 folder) and load them into R.
2. We dont have very large numbers of speeches, but we can do a topic analysis using paragraphs as documents. Load `quanteda`, construct a corpus object and use the `corpus_segment` function to split the documents into paragraphs.
3. Use `dfm` to construct a document-feature matrix out of the paragraphs. Convert everything to lowercase, remove numbers, symbols and english stopwords.
4. Use `convert` to convert the dfm to a topicmodel object.
5. Estimate a lda model using the `LDA` function from the `topicmodel` package and set the number of topics to 6 and α to 0.1.
6. Create a plot of the top 10 terms that are most common within each topic.

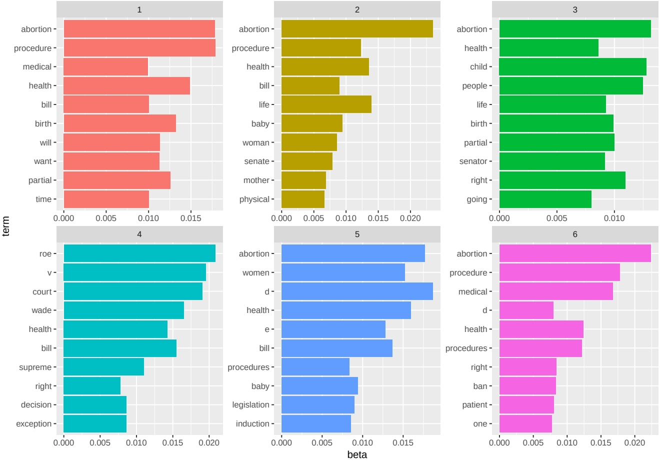
Topic model exercise solution

```r
library(quanteda)
library(readtext)
library(topicmodels)
wdir <- getwd()
speechCorpus <- corpus(readtext(paste0(wdir, "/usSenateDebate/*.txt")))
speechCorpusPara <- corpus_segment(speechCorpus, what = "paragraphs")
myDfm <- dfm(speechCorpusPara, tolower = T,
                removeNumbers = T, removeSymbols = T,
                remove = stopwords("english"))
ldaDfm <- convert(myDfm, to = "topicmodels")
speechLda <- LDA(ldaDfm, control = list(alpha = 0.1, seed = 1234), k = 6)
```

Topic model exercise solution (II)

```r
library(tidytext)
library(ggplot2)
library(dplyr)
apTopics <- tidy(speechLda, matrix = "beta")
topTerms <- apTopics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
plotTerms <- topTerms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```

Topic model exercise solution (III)



Quantitative Text Analysis in R

Take home messages

- There is an abundant amount of text data in the social sciences
- Generating inferences from these data frequently requires computers and algorithms
- Various computerized methods exists to quantify and analyze text
- Everything can be done in R using a handful of packages

Quantitative Text Analysis in R (II)

You should be able to

- . . . interact and manipulate text data in R
- . . . classify documents into categories
- . . . scale documents in a unidimensional space
- . . . understand the theory behind the text classification functions
- . . . create appealing visualizations
- . . . access interesting text data sources using APIs and other ways
- . . . (scrape data off the internet)

Quantitative Text Analysis in R (III)

What to improve

- Working alongside the slides
- Clearer expectations regarding the assignments
- Better balance between theory and practice
- Incorporate YOUR interests more
- . . .