Text Mining

Itamar Caspi

May 26, 2019 (updated: 2019-05-26)

Packages

```r
if (!require("pacman"))
  install.packages("pacman")

pacman::p_load(quanteda,
               tidyverse,
               knitr,
               xaringan,
               RefManageR)
```

The `quanteda` package is a Swiss army knife for handling text with R. More on that later.

Outline

1. Representing Text as Data
2. Text Regressions
3. Dictionary-based Methods
4. Topic Modeling
5. Hands-on

Representing Text as Data

Where to Start

A great introduction to this topic, along with many empirical examples from the social sciences appear in Gentzkow, Kelly, and Taddy's **"Text as Data"** (JEL forthcoming).

Basic notation

Definitions:

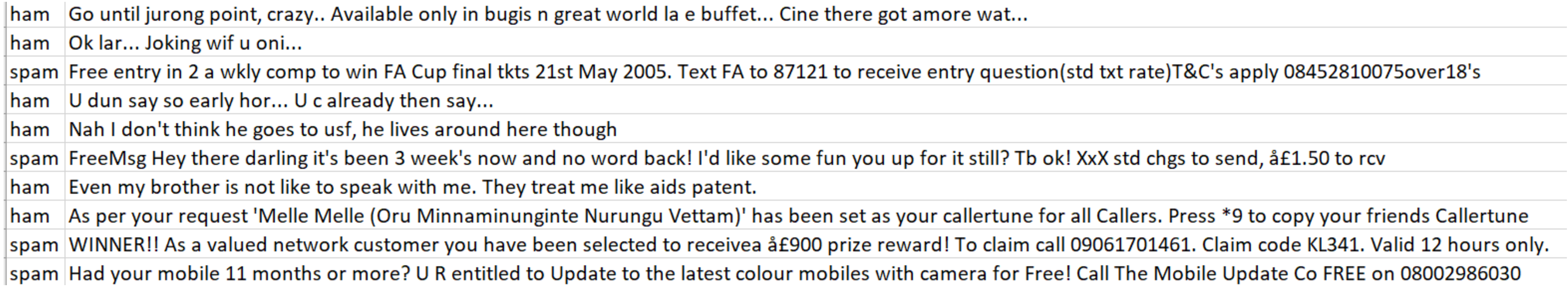
- A *corpus* is a collection of D documents (emails, tweets, speeches, articles, etc.)
- A *vocabulary* is a complete list of unique words that appear in the corpus.
- $\mathbf{X}$ is a numerical array representation of text. Rows correspond to documents $i = 1, \ldots, D$ and columns to words $j = 1, \ldots, N$.
- Y is a vector of predicted outcome (e.g., spam/ham, trump/not trump, etc.), one outcome per document.
- F is a low-dimensional representation of $\mathbf{X}$.

Document term matrix (DTM)

In most applications raw text is represented as a numerical array $\mathbf{X}$ where the elements of the array, X_{ij}, are counts of words (or more generally, *tokens*. More on that later.)

Illustration: Spam vs. Ham

Consider the task of spam detection:



ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
ham	Ok lar... Joking wif u oni...
spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
ham	U dun say so early hor... U c already then say...
ham	Nah I don't think he goes to usf, he lives around here though
spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, å£1.50 to rcv
ham	Even my brother is not like to speak with me. They treat me like aids patent.
ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
spam	WINNER!! As a valued network customer you have been selected to receivea å£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030

In this case

- Documents are emails.
- Vocabulary includes words that appear in *each and every* emails.

NOTE: Spam detection is clearly a supervised learning task where $Y_i = \{\text{spam}, \text{ham}\}$.

Transforming a corpus to a DTM

Consider the following corpus ($M=3$):

```
txt <- c(doc1 = "Shipment of gold damaged in a fire",
         doc2 = "Delivery of silver arrived in a silver truck",
         doc3 = "Shipment of gold arrived in a truck" )

txt %>% quanteda::dfm() # transform text as a document term matrix
```

```
## Document-feature matrix of: 3 documents, 11 features (36.4% sparse).
## 3 x 11 sparse Matrix of class "dfm"
##       features
## docs   shipment of gold damaged in a fire delivery silver arrived truck
##   doc1        1  1    1       1  1 1    1        0      0       0     0
##   doc2        0  1    0       0  1 1    0        1      2       1     1
##   doc3        1  1    1       0  1 1    0        0      0       1     1
```

- `nrow(dfm_txt)` returns the number of documents in the corpus.
- `ncol(dfm_txt)` returns the number of words in the vocabulary.
- `colnames(dfm_txt)` returns the number of words in the vocabulary

Does every words matter? ¯_(ツ)_/¯

We can significantly reduce the dimension of $\mathbf{X}$ by

- filtering out very common ("stop words") and uncommon words.
- dropping numbers and punctuation.
- stemming, i.e., replacing words by their root. (*economi* instead of *economics*, *economists*, *economy*)
- convert to lower case

WARNING: Use text prepossessing steps with care. These steps should be application specific.

Illustration of some common preprocessing steps

Here, we remove stop words, punctuation, numbers, and stem words:

```r
txt <- c(doc1 = "Shipment of gold damaged in a fire",
         doc2 = "Delivery of silver, arrived in a silver truck",
         doc3 = "Shipment of 2 Karats of gold arrived in a truck" )

txt %>% dfm(remove = stopwords("english"),
            remove_punct = TRUE,
            remove_numbers = TRUE,
            stem = TRUE)
```

```
## Document-feature matrix of: 3 documents, 9 features (51.9% sparse).
## 3 x 9 sparse Matrix of class "dfm"
##       features
## docs   shipment gold damag fire deliveri silver arriv truck karat
##   doc1        1    1     1    1        0      0     0     0     0
##   doc2        0    0     0    0        1      2     1     1     0
##   doc3        1    1     0    0        0      0     1     1     1
```

Words are generally not independent

Sometimes we might care about multiword expressions, e.g., "not guilty", "labor market", etc.

We can define tokens (the basic element of text) as n-gram - a sequence of n words from a given sample of text.

NOTE: Using n-gram with n>2 is typically impractical due to the rapid growth in the dimensions of $\mathbf{X}$.

Example

Here is our sample text (just 2 "documents" in this example), where tokens are defined as *bi-gram* (a sequence of two words):

```
txt <- c(doc1 = "Shipment of gold damaged in a fire",
         doc2 = "Delivery of silver arrived in a silver truck")

txt %>% dfm(ngrams = 2L)
```

```
## Document-feature matrix of: 2 documents, 12 features (45.8% sparse).
## 2 x 12 sparse Matrix of class "dfm"
##       features
## docs   shipment_of delivery_of of_gold of_silver gold_damaged
##   doc1           1           0       1         0            1
##   doc2           0           1       0         1            0
##       features
## docs   silver_arrived damaged_in arrived_in in_a a_silver a_fire
##   doc1              0          1          0    1        0      1
##   doc2              1          0          1    1        1      0
##       features
```

The (social science) textmining play book

1. Collect text and generate a corpus.

2. Represent corpus as a DTM $\mathbf{X}$.

3. Then, proceed according to one of the following steps:

   - Use $\mathbf{X}$ to predict an outcome Y using high dimensional methods (e.g., lasso, Ridge, etc.). In some cases, proceed with $\hat{Y}$ to subsequent analysis.

   - Apply dimensionality reduction techniques (dictionary, PCA, LDA, etc.) to $\mathbf{X}$ and proceed with the output $\mathbf{F}$ to subsequent analysis.

Remember:

> "*Text information is usually best as part of a larger system. Use text data to fill in the cracks around what you know. Don't ignore good variables with stronger signal than text!*" (Matt Taddy)

Text Regression

Sounds familiar...

- We are interested in predicting some Y using $\mathbf{X}$.
- Clearly, with text as data, we are facing the high-dimensionality problem. $\mathbf{X}$ has $M \times N$ elements.
- Classical methods such as OLS won't do the trick $\Rightarrow$ need the ML touch.
- An obvious choice would be penalized linear/non-linear regression (e.g. Lasso, ridge, etc.). Other methods such as random forest can work too.

EXAMPLE: Lasso text regression `glmnet(Y, X)` where

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^N}{\mathbf{argmin}} \sum_{i=1}^{N} (Y_i - X_i \beta)^2 + \lambda ||\boldsymbol{\beta}||_1$$

Can be easily extended to binary / categorical Y, e.g.`glmnet(X, Y, family = "binomial")`

Practical notes about penalized text regression

- Typically, DTM entries count the number of times word i appears in document d. This provide "intuitive" interpretation for regression coefficients.

-

- Nevertheless, as usual, beware of giving causal interpretation to the Lasso's coefficients. (Recall the irrepresentability condition.)

Dictionary-based Methods

Reducing dimensionality using a dictionary

- Dictionary-based methods provide a low-dimensional representation of high-dimensional text.

- Essentially, Think of F as an unobserved characteristic of the text that we are trying to estimate. A Dictionary-based methods provides a mapping from $\mathbf{X}$ onto F:

$$g : \mathbf{X} \rightarrow F$$

Example: Sentiment analysis

- A prominent example of dictionary-based methods is sentiment analysis
- The latent factor we are trying to estimate is the writer's attitude towards the topic in question.
- The most common approach is based on prespecified dictionaries that classify word according to some predefined sentiment class (e.g. "positive", "negative", and "neutral".)
- Typically, the sentiment *score* of each document is a function of the relative frequencies of positive, negative, neutral, etc., words.

REMARK: Sentiment analysis can be supervised as well. E.g., the availability of labeled movie reviews (1-5 stars) can be used to train a model and use its predictions to classify unlabeled reviews.

Example: Loughran and McDonald financial sentiment dictionary

A random list of words from the Loughran and McDonald (2011) financial sentiment dictionary (positive/negative/litigious/uncertainty/constraining):

```r
library(tidytext)
sample_n(get_sentiments("loughran"),8)
```

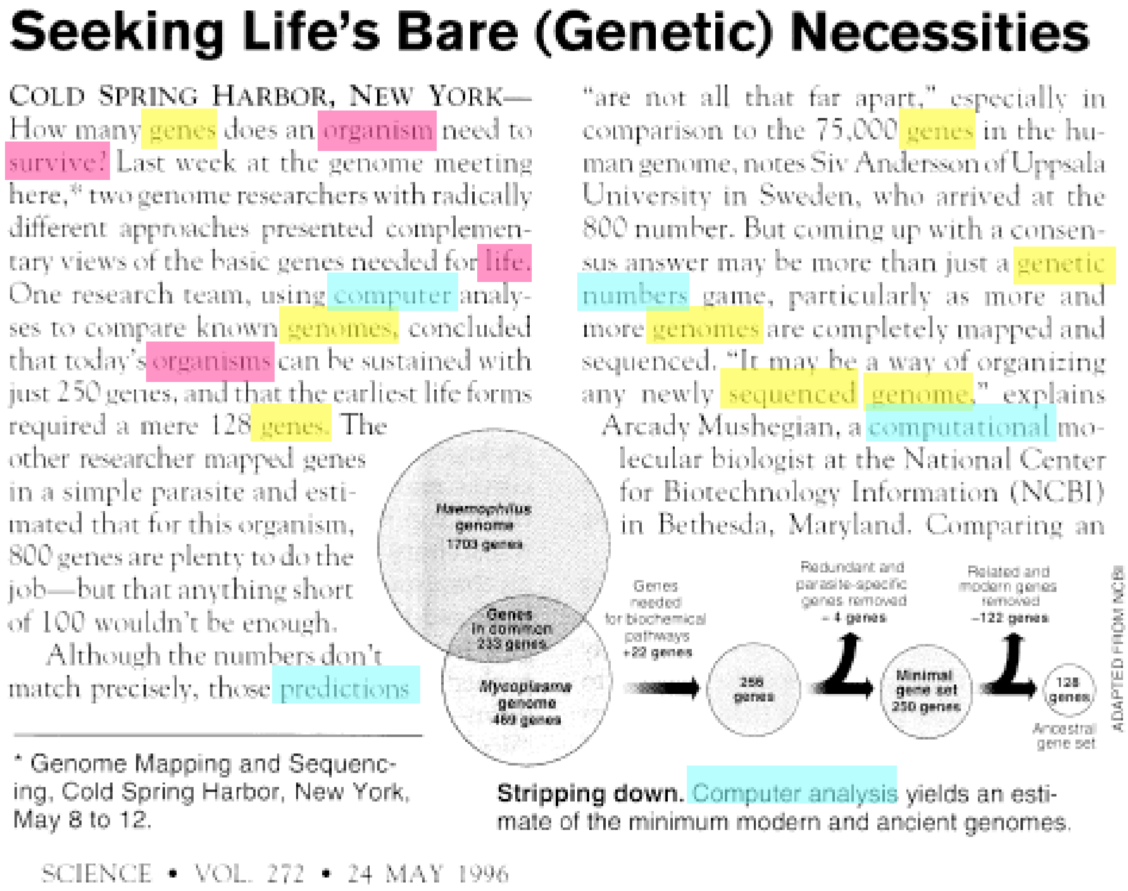
```
## # A tibble: 8 x 2
##   word         sentiment
##   <chr>        <chr>
## 1 disapproving negative
## 2 exculpated   negative
## 3 rehearings   litigious
## 4 somewhere    uncertainty
## 5 overbuild    negative
## 6 consent      litigious
## 7 unfavorable  negative
## 8 variants     uncertainty
```

Topic Modeling

Topic models

- Topic models extend unsupervised learning methods to text data.
- Topic modeling classifies documents and words to latent topics and is often followed by more conventional empirical methods.
- The workhorse of topic modeling is the Latent Dirichlet Allocation model (Blei, Ng, and Jordan, 2003), or LDA for short.

Intuition behind LDA



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

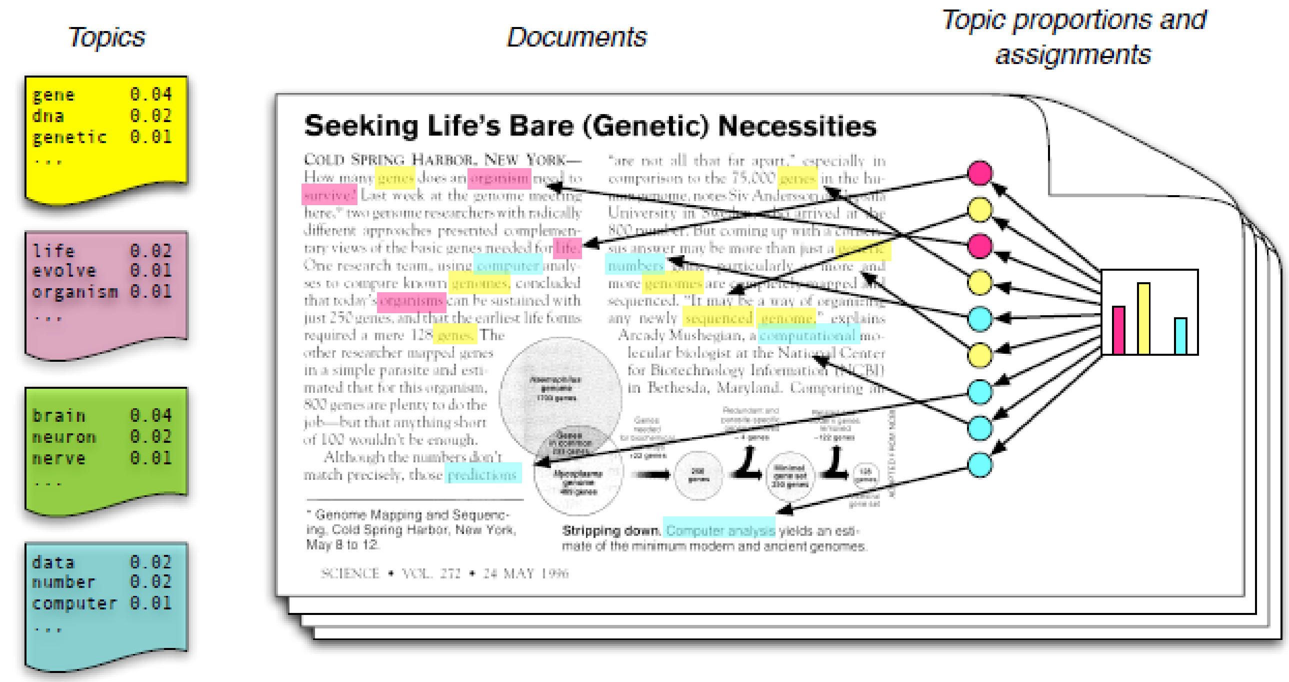
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways +22 genes
256 genes
Redundant and parasite-specific genes removed −4 genes
Minimal gene set 250 genes
Related and modern genes removed −122 genes
128 genes
Ancestral gene set
ADAPTED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

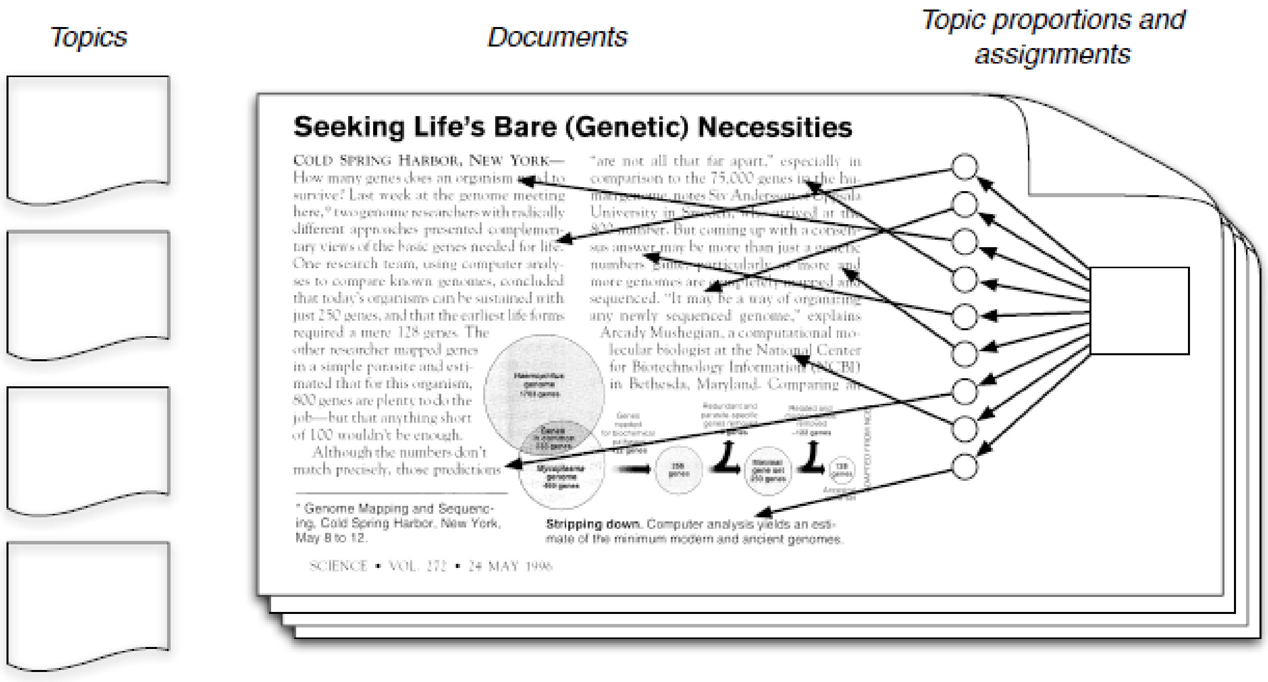
SCIENCE • VOL. 272 • 24 MAY 1996

Intuition behind LDA


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
...
Seeking Life's Bare (Genetic) Necessities
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
SCIENCE • VOL. 272 • 24 MAY 1996

- A topic is a distribution over *all* the words included in a *fixed* vocabulary.
- A word can have non-zero (yet different) probability multiple topics (e.g., bank)
- Each document is a mixture of topics
- Each word is drawn from one of the topics.

Intuition behind LDA


Topics
Documents
Topic proportions and assignments
Haemophilus genome 1703 genes
Genes in common 233 genes
Mycoplasma genome 469 genes
Genes needed for biochemical pathways
256 genes
Redundant and parasite-specific genes removed
Minimal gene set 250 genes
Related and duplicate genes removed
128 genes
Ancestral
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

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

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

SCIENCE • VOL. 272 • 24 MAY 1996

QUESTION: How realistic is the LDA setup? Does it matter? What's our goal here anyway?

Notation

- A *vocabulary* is a collection of words represented by the vector $\{1, \ldots, V\}$
- Each *word* is represented by a unit vector $\boldsymbol{\delta}_v = (0, \ldots, v, \ldots, 0)'$
- A *document* is a sequence of N words denoted by $\mathbf{w} = (w_1, \ldots, w_N)$.
- A *corpus* is a collection of M documents denoted by $\mathcal{D} = (\mathbf{w}_1, \ldots, \mathbf{w}_M)$.

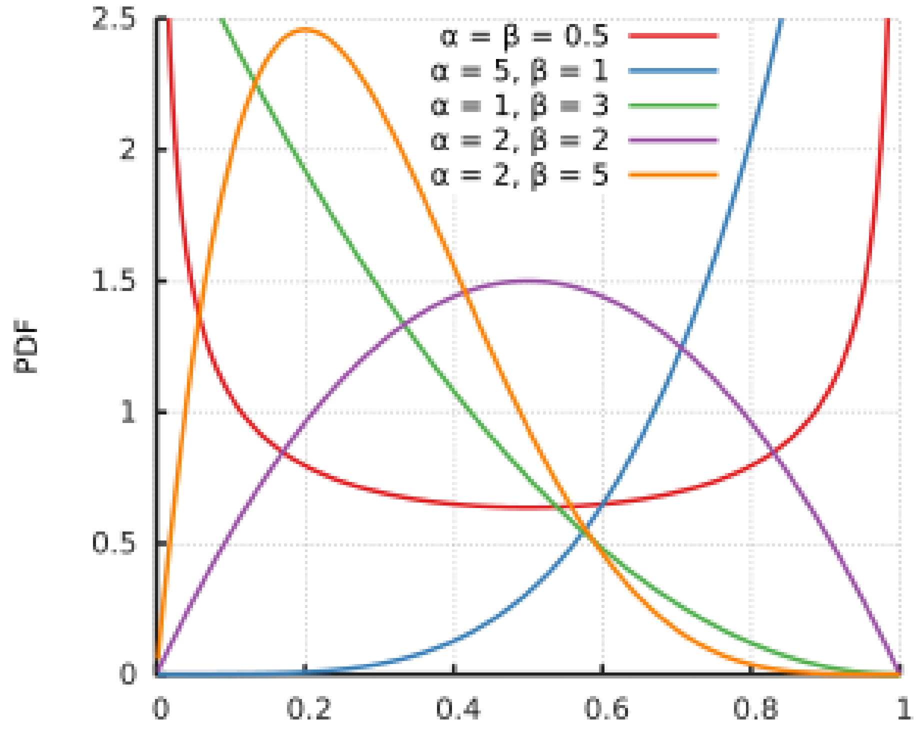
Prerequisite: The Beta distribution

The PDF for the Beta distribution, denoted as $B(\alpha, \beta)$ is

$$p(\theta|\alpha, \beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

for $\theta \in [0, 1]$ and $\alpha, \beta > 0$.

Because its properties, the Beta distribution is useful as a prior for probabilities.



The Dirichlet distribution

The Dirichlet distribution, denoted as $\mathbf{Dir}(\boldsymbol{\alpha})$ is a multivariate generalization of the Beta distribution.

Let $\theta = (\theta_1, \theta_2, \ldots, \theta_K) \sim \mathbf{Dir}(\boldsymbol{\alpha})$.

The PDF for a K-dimensional Dirichlet distribution is

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \propto \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}$$

where $K \geq 2$ is the number of categories, $\alpha_i > 0$ and $\theta_i \in (0, 1)$ for all i and $\sum_{i=1}^{K} \theta_i = 1$.

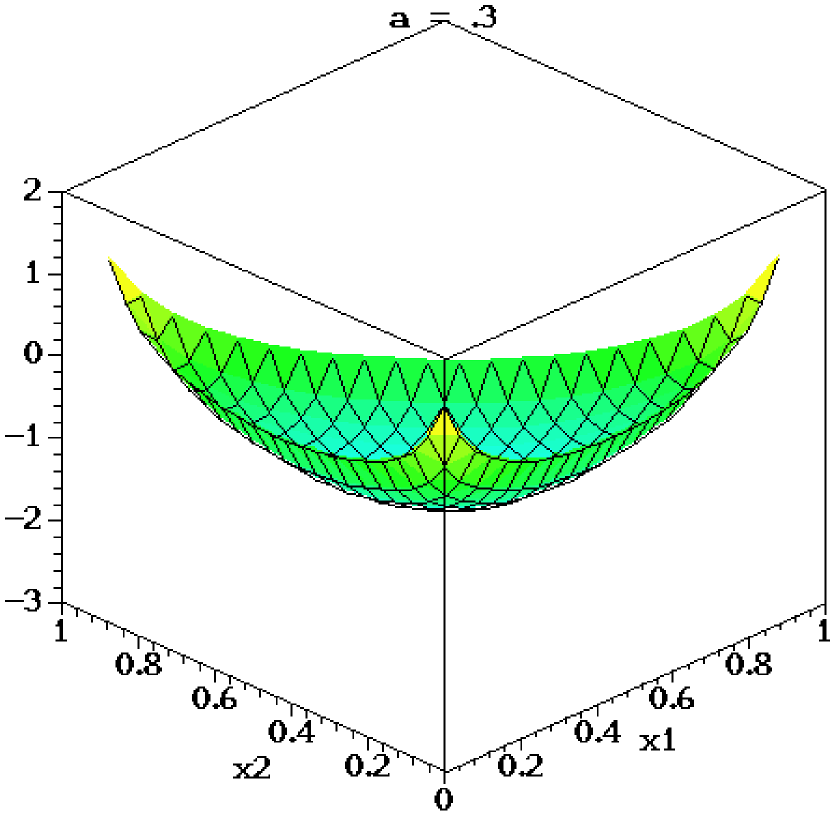
REMARK: The parameter $\boldsymbol{\alpha}$ controls the sparsity of $\boldsymbol{\theta}$

BOTTOM LINE: Vectors drawn from a Dirichlet distribution represent probabilities.

On the right:

The change in the density function $(K = 3)$ as the vector $\boldsymbol{\alpha}$ changes from $\boldsymbol{\alpha} = (0.3, 0.3, 0.3)$ to $(2.0, 2.0, 2.0)$, while keeping $\alpha_1 = \alpha_2 = \alpha_3$.

REMARK: Placing $\boldsymbol{\alpha} = (1, 1, 1)$ results in a uniform distribution over the simplex.



By Initial version by Panos Ipeirotis, later modified by Love Sun and Dreams - [1], CC BY 3.0, Link

The data generating process behind LDA

Assumption: The number of topics K and the size of the vocabulary V are fixed.

The DGP:

For each document $d = 1, \ldots, \mathcal{D}$

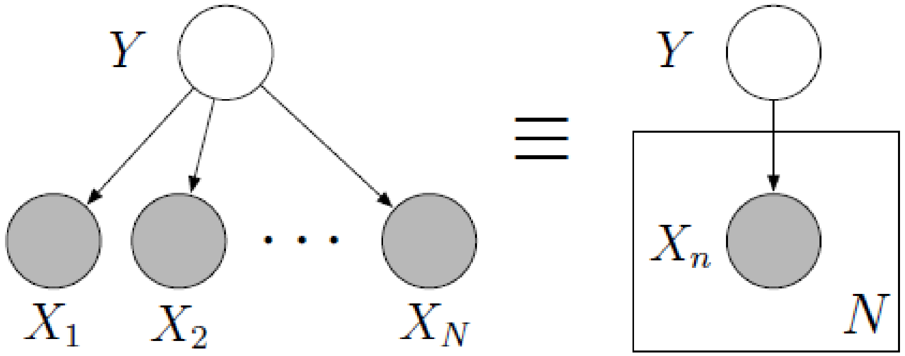
1. Choose topic proportions $\theta_d \sim \mathrm{Dir}(\boldsymbol{\alpha})$
2. For each word $n = 1, \ldots, N$

    2.1. Choose a topic assignment $Z_{dn} \sim \mathrm{Mult}(\theta_d)$.

    2.2. Choose a word $W_{dn} \sim \mathrm{Mult}(\beta_{z_{dn}})$.

REMARK: Note the "factor model" aspects of LDA, where topics are factors and word probabilities are loadings, and both affect the probability of choosing a word.

Aside: Plate notation

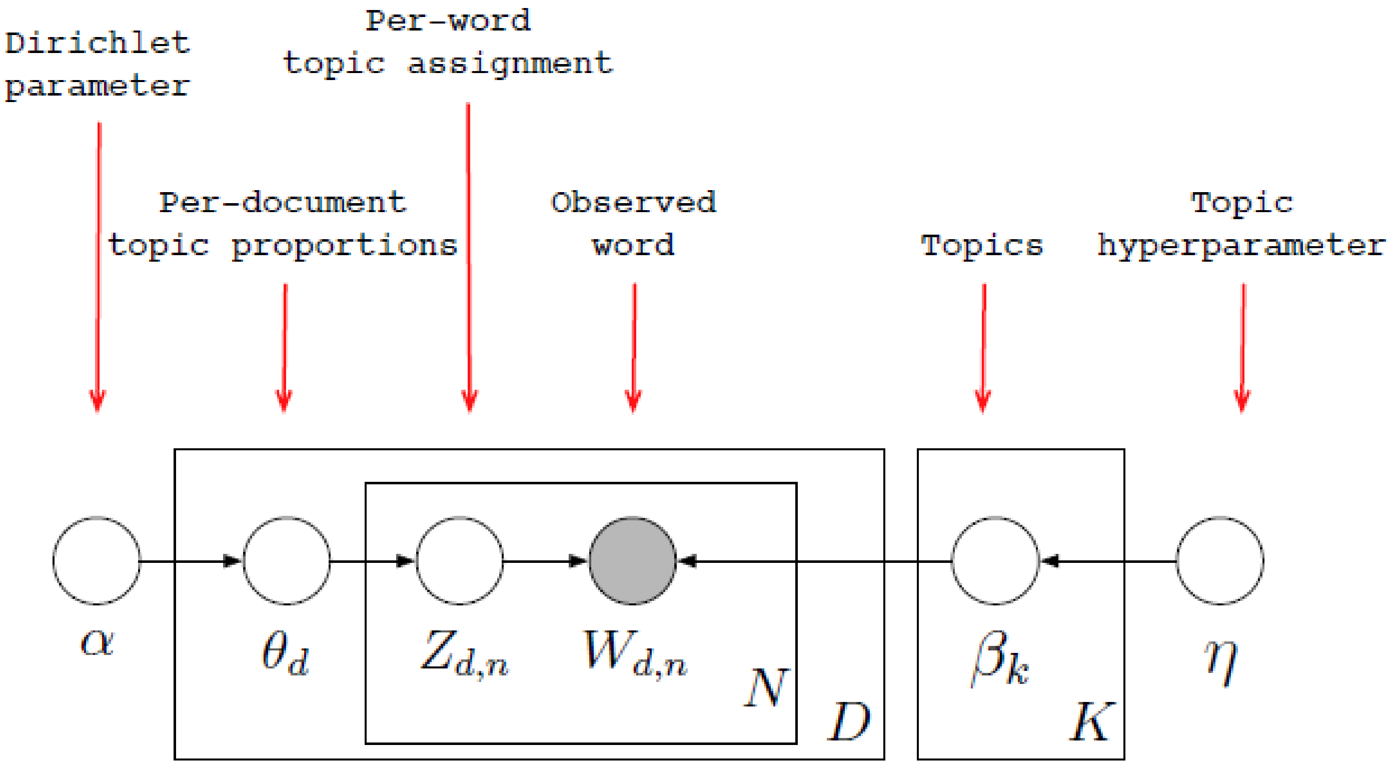


- Each *node* is a random variable
- *Shaded* nodes are observables
- *Edges* denote dependence
- *plates* denote replicated structures

The above graph corresponds to

$$p\left(y, x_1, \ldots, x_N\right) = p(y) \prod_{n=1}^{N} p\left(x_n \mid y\right)$$

LDA in plate notation



Source: http://videolectures.net/mlss09uk_blei_tm/#.

Aside: Conjugate priors

The Dirichlet distribution is a conjugate prior for the Multinomial.

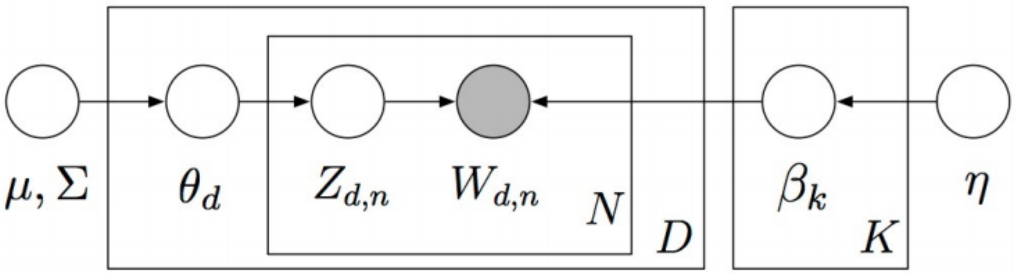
Let $n(Z_i)$ denote the count of topic i.

$$\boldsymbol{\theta} | Z_{1,...,N} \sim \mathrm{Dir}(\boldsymbol{\alpha} + n(Z_{1,...,N}))$$

i.e., as the number of times we see topic i increases, our posterior becomes "peakier" at its i^{th} component.

Extension #1: Correlated topic models (Lafferty and Blei, 2005)

- LDA assumes that topics independently cooccure in documents.
- This is clearly wrong.
- For example, a document about *economics* is more likely to also be about *politics* than it is to be about *cooking*.
- Lafferty and Blei relax independence by drawing topic proportions from a logistic normal, which allows correlations between topic proportions:



where μ and Σ are priors for the logistic normal distribution.

Extension #2: Dynamic LDA (Blei and Lafferty, 2006)

Dynamic topic modeling takes into account the ordering of the documents and gives a richer posterior topical structure than LDA

In dynamic topic modeling, a topic is a *sequence* of distributions over words. Topics evolve systematically over time. In particular, the vector of parameters for topic k in period t evolves with a Gaussian noise:

$$\beta_{t,k} | \beta_{t-1,k} \sim \mathcal{N}\left(\beta_{t-1,k}, \sigma^2 I\right).$$

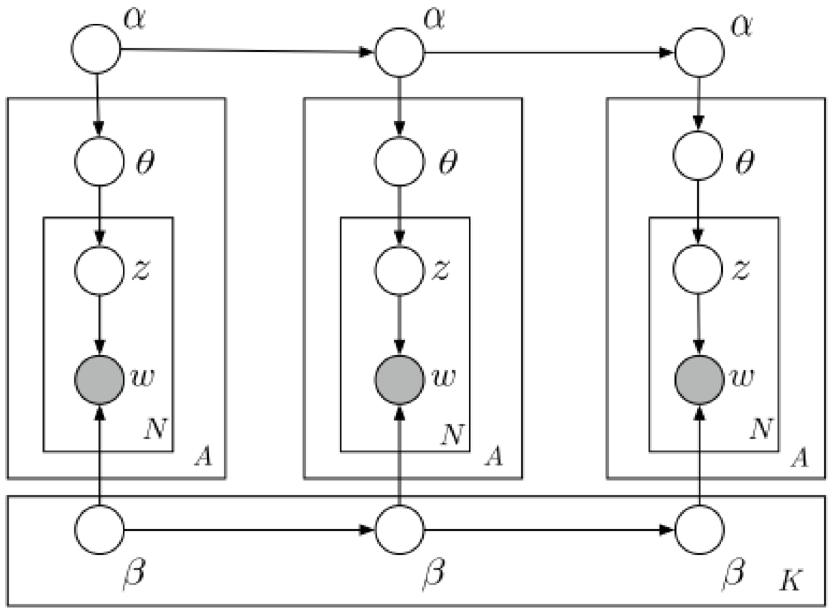
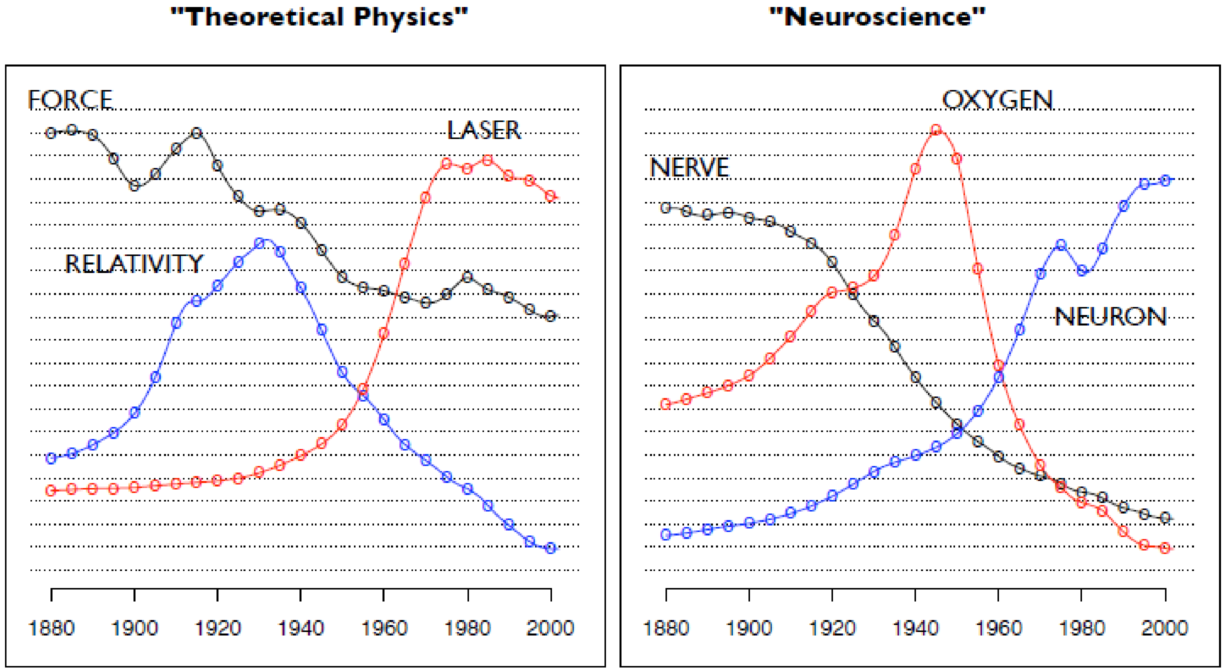


Figure 1. Graphical representation of a dynamic topic model (for three time slices). Each topic's natural parameters $\beta_{t,k}$ evolve over time, together with the mean parameters α_t of the logistic normal distribution for the topic proportions.

Dynamic LDA: Science 1881-1999

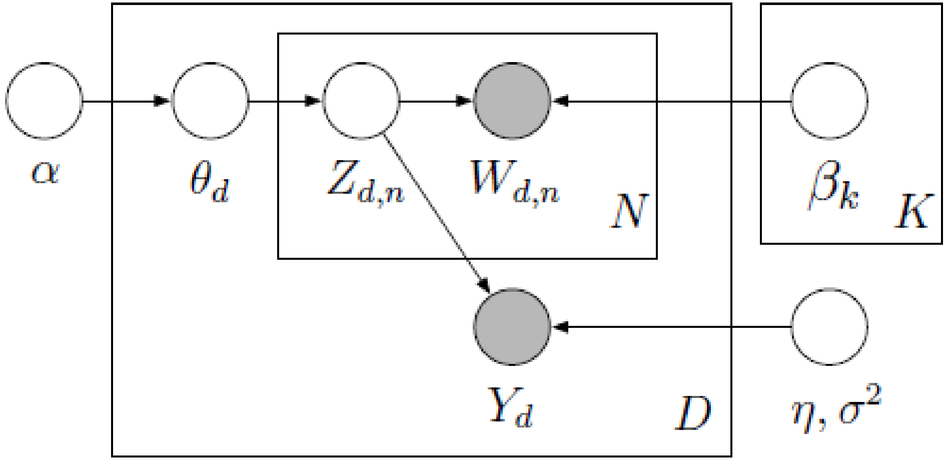
The posterior estimate of the frequency as a function of year of several words from the two topics: "Theoretical Physics" and "Neuroscience":



Source: Blei and Lafferty (2006).

Extension #3: Supervised Topic Model (McAuliffe and Blei, 2008)

add an extra connection between Z_{dn} to some observable attribute Y_d:



Source: McAuliffe and Blei (2008).

Structural Topic Models (Roberts, Stewart, and Tingley)

About the Structural Topic Model (STM):

> *"The Structural Topic Model is a general framework for topic modeling with document-level covariate information. The covariates can improve inference and qualitative interpretability and are allowed to affect topical prevalence, topical content or both."*

In STM, topics are drawn from the following logistic normal distribution,

$$\boldsymbol{\theta}_d | X_d\gamma, \Sigma \sim \text{ LogisticNormal } (\mu = X_d\gamma, \Sigma)$$

where X_d is a vector of observed document covariates.

REMARK: In the case of no covariates, the STM reduces to a (fast) implementation of the Correlated Topic Model (Blei and Lafferty, 2007).

`stm`: R package for structural topic models

Roberts, Stewart, and Tingley (JSS, 2014)

About the `stm` R package:

> *"The software package implements the estimation algorithms for the model and also includes tools for every stage of a standard workflow from reading in and processing raw text through making publication quality figures."*

The package is available on CRAN and can be installed using:

```
install.packages("stm")
```

To get started, see the vignette which includes several example analyses.

Applying topic models to measure the effect of transparency

Hansen, McMahon, and Prat (QJE 2017) study the effect of increasing Federal Open Market Committee (FOMC) transparency on debate during FOMC meetings.

- FOMC meetings have been tape recorded since the 1970s to prepare minutes.
- Committee members believed that these tapes were erased afterward.
- In October 1993, Fed chair Alan Greenspan ,discovered and revealed that before being erased the tapes had been transcribed and stored in archives all along.
- Following Greenspan's revelation The Fed agreed to publish all past transcripts and extended that policy to cover all future transcripts with a five-year lag.
- This gives Hansen et al. access to periods both when policy makers did and did not believe their deliberations would be public.

Topic modeling of FOMC meeting transcripts

Data:

- 149 FOMC meeting transcripts during, Alan Greenspan's tenure, before and after 1993.
- The unit of observation is a member-meeting.
- The outcomes of interest are
    - the proportion of words devoted to the K different topics
    - the concentration of these topic weights
    - the frequency of data citation.

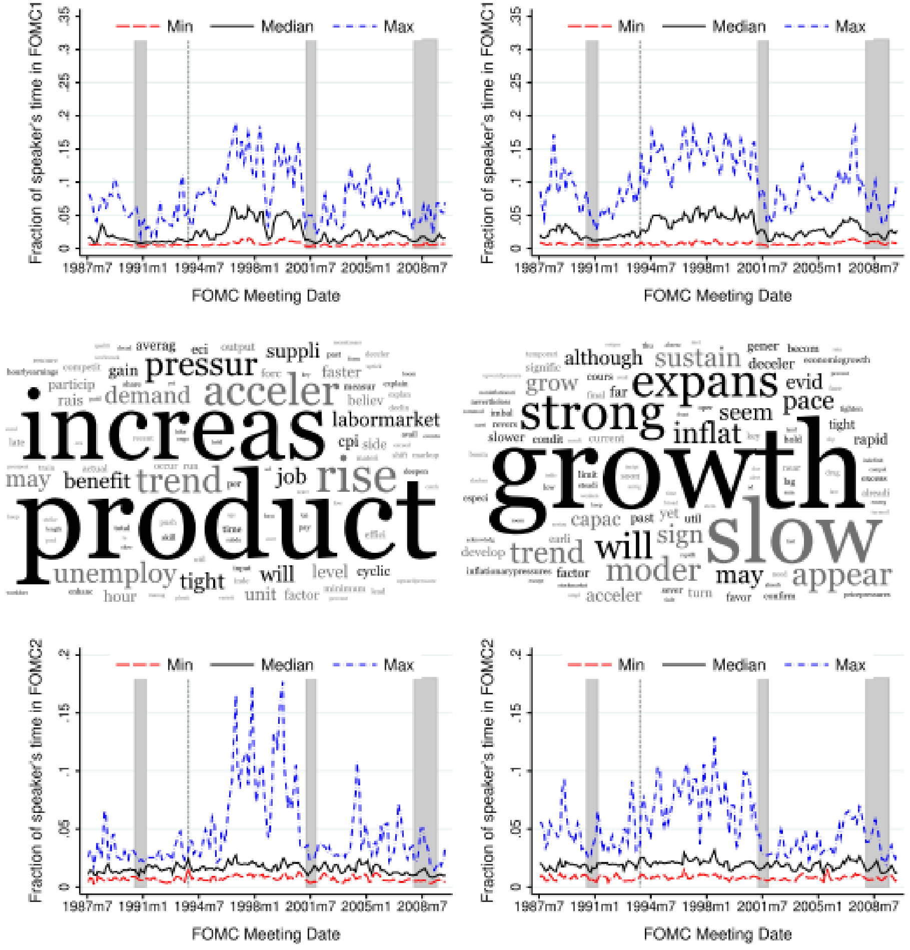
Estimation

- Estimate topics using LDA.
- Use LDA's output to construct outcomes of interest
- Difference / Difference-in-differences regressions that estimate the effects of the change in transparency on outcomes. For example, Hansen et al. estimate

$$y_{it} = \alpha_i + \gamma D(\mathbf{Trans})_t + \lambda X_t + \varepsilon_{it}$$

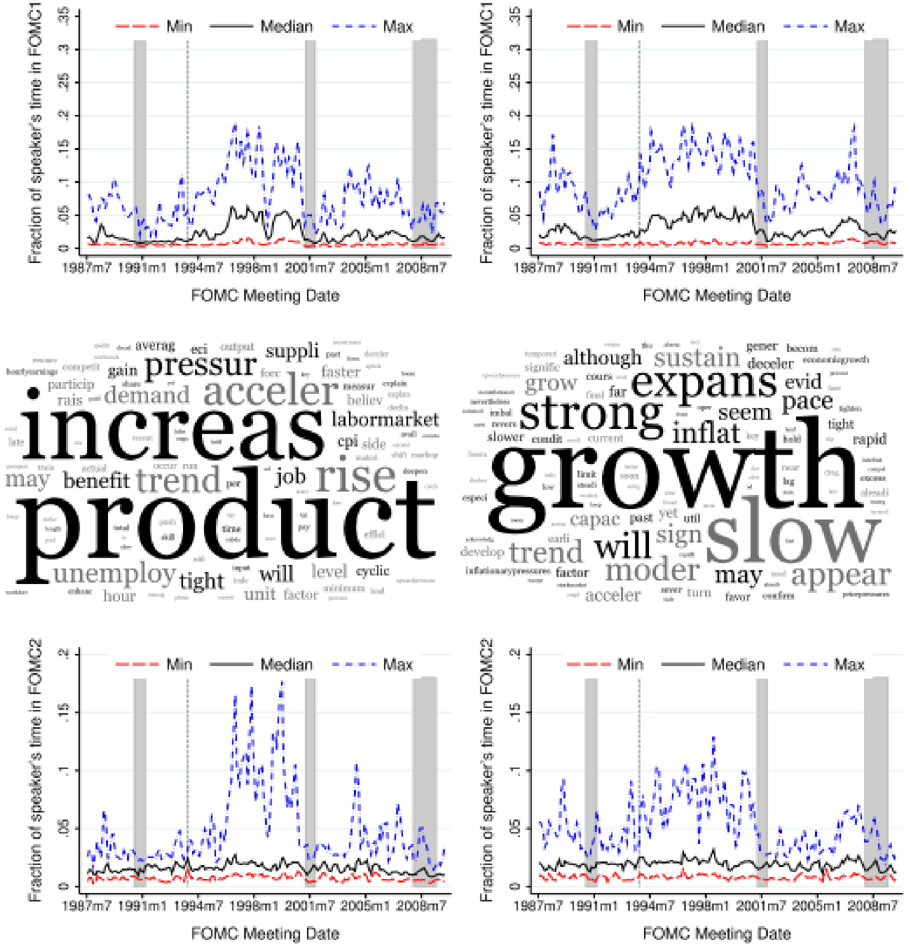
- where
  - y_{it} represents any of the communication measures for member i in time t.
  - $D(\mathbf{Trans})$ is an indicator for being in the transparency regime (1 after November 1993, 0 before).
  - X_t is a vector of macro controls for the meeting at time t.

Pro-cyclical topics



Source: Hansen, McMahon, and Prat (QJE 2017).

Counter-cyclical topics



Source: Hansen, McMahon, and Prat (QJE 2017).

Increased accountability: More references to data

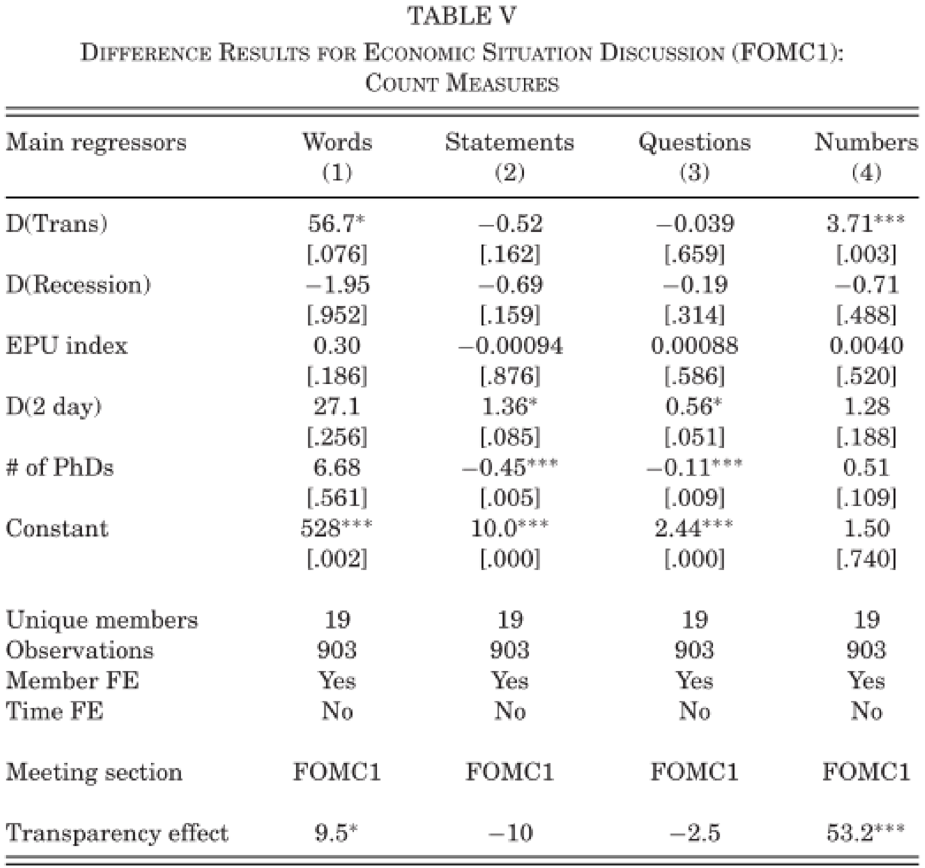


TABLE V

Difference Results for Economic Situation Discussion (FOMC1): Count Measures

Main regressors	Words (1)	Statements (2)	Questions (3)	Numbers (4)
D(Trans)	56.7*	−0.52	−0.039	3.71***
	[.076]	[.162]	[.659]	[.003]
D(Recession)	−1.95	−0.69	−0.19	−0.71
	[.952]	[.159]	[.314]	[.488]
EPU index	0.30	−0.00094	0.00088	0.0040
	[.186]	[.876]	[.586]	[.520]
D(2 day)	27.1	1.36*	0.56*	1.28
	[.256]	[.085]	[.051]	[.188]
# of PhDs	6.68	−0.45***	−0.11***	0.51
	[.561]	[.005]	[.009]	[.109]
Constant	528***	10.0***	2.44***	1.50
	[.002]	[.000]	[.000]	[.740]
Unique members	19	19	19	19
Observations	903	903	903	903
Member FE	Yes	Yes	Yes	Yes
Time FE	No	No	No	No
Meeting section	FOMC1	FOMC1	FOMC1	FOMC1
Transparency effect	9.5*	−10	−2.5	53.2***

Source: Hansen, McMahon, and Prat (QJE 2017).

Increased conformity: increase in similarity

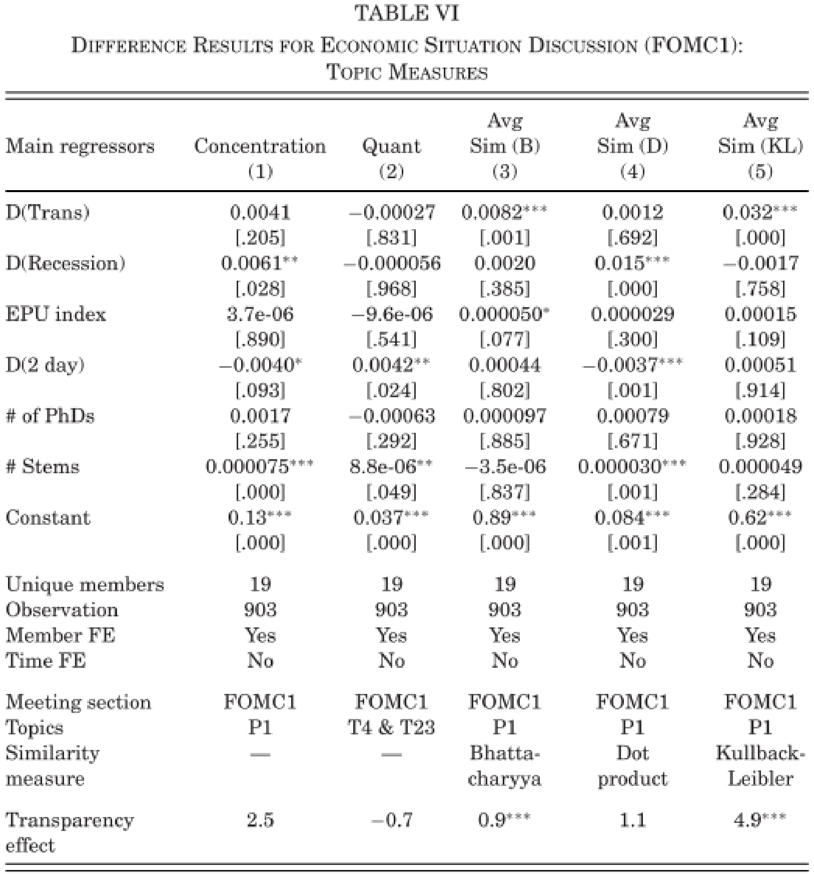


TABLE VI

DIFFERENCE RESULTS FOR ECONOMIC SITUATION DISCUSSION (FOMC1): TOPIC MEASURES

Main regressors	Concentration (1)	Quant (2)	Avg Sim (B) (3)	Avg Sim (D) (4)	Avg Sim (KL) (5)
D(Trans)	0.0041	−0.00027	0.0082^{***}	0.0012	0.032^{***}
	[.205]	[.831]	[.001]	[.692]	[.000]
D(Recession)	0.0061^{**}	−0.000056	0.0020	0.015^{***}	−0.0017
	[.028]	[.968]	[.385]	[.000]	[.758]
EPU index	3.7e-06	−9.6e-06	0.000050^{*}	0.000029	0.00015
	[.890]	[.541]	[.077]	[.300]	[.109]
D(2 day)	-0.0040^{*}	0.0042^{**}	0.00044	-0.0037^{***}	0.00051
	[.093]	[.024]	[.802]	[.001]	[.914]
# of PhDs	0.0017	−0.00063	0.000097	0.00079	0.00018
	[.255]	[.292]	[.885]	[.671]	[.928]
# Stems	0.000075^{***}	$8.8\text{e-}06^{**}$	−3.5e-06	0.000030^{***}	0.000049
	[.000]	[.049]	[.837]	[.001]	[.284]
Constant	0.13^{***}	0.037^{***}	0.89^{***}	0.084^{***}	0.62^{***}
	[.000]	[.000]	[.000]	[.001]	[.000]
Unique members	19	19	19	19	19
Observation	903	903	903	903	903
Member FE	Yes	Yes	Yes	Yes	Yes
Time FE	No	No	No	No	No
Meeting section	FOMC1	FOMC1	FOMC1	FOMC1	FOMC1
Topics	P1	T4 & T23	P1	P1	P1
Similarity measure	—	—	Bhattacharyya	Dot product	Kullback-Leibler
Transparency effect	2.5	−0.7	0.9^{***}	1.1	4.9^{***}

Source: Hansen, McMahon, and Prat (QJE 2017).

Hands-on

Go to:

`11-hands-on-text-as-data.Rmd`

```
slides %>% end()
```

Source code

Selected references

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan), 993-1022.

Blei, D. M., & Lafferty, J. D. (2006, June). Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning* (pp. 113-120). ACM.

Gentzkow, M., Kelly, B.T. and Taddy, M. (forthcoming). *The Quarterly Journal of Economics*.

Hansen, S., McMahon, M., & Prat, A. (2017). Transparency and Deliberation Within the FOMC: A Computational Linguistics Approach. *The Quarterly Journal of Economics*, 133(2), 801–870.

Lafferty, J. D., & Blei, D. M. (2006). Correlated topic models. In *Advances in neural information processing systems* (pp. 147-154).

Loughran, T. and McDonald, B., 2011. When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, 66(1), pp.35-65.

Selected references

Roberts, M.E., Stewart, B.M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S.K., Albertson, B. and Rand, D.G., 2014. Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4), pp.1064-1082.

Roberts, M.E., Stewart, B.M. and Tingley, D., 2014. stm: R package for structural topic models. *Journal of Statistical Software*, 10(2), pp.1-40.