

# Text Analysis

---

Jesús Fernández-Villaverde<sup>1</sup>

August 7, 2023

<sup>1</sup>University of Pennsylvania

# Text is the new data

- Important for economics:
  1. Statements by policy makers.
  2. Political manifestos.
  3. Legal documents (court decisions, criminal records).
  4. Companies earning reports.
  5. Customer complaints.
  6. Documents in libraries and archives.
  7. News, news commentary, and interviews.
  8. Verbal surveys.
  9. Opinion mining and sentiment analysis from social media.

# How do we handle text?

- How do we use text in economic and statistical methods?
- Historically: reading the documents (or interviewing the authors)! But too slow, prone to errors and biases, and hard to replicate.
- Basic statistics: **Inference in an Authorship Problem** by **Mosteller and Wallace (1963)**.
- Machine learning can help to extend the scope of text analysis.

*For Mr Church from his sister  
Elizabeth THE Hamilton*

FEDERALIST;

A COLLECTION

OF

E S S A Y S,

WRITTEN IN FAVOUR OF THE

NEW CONSTITUTION,

AS AGREED UPON BY THE FEDERAL CONVENTION,  
SEPTEMBER 17, 1787.

IN TWO VOLUMES.

VOL. I.

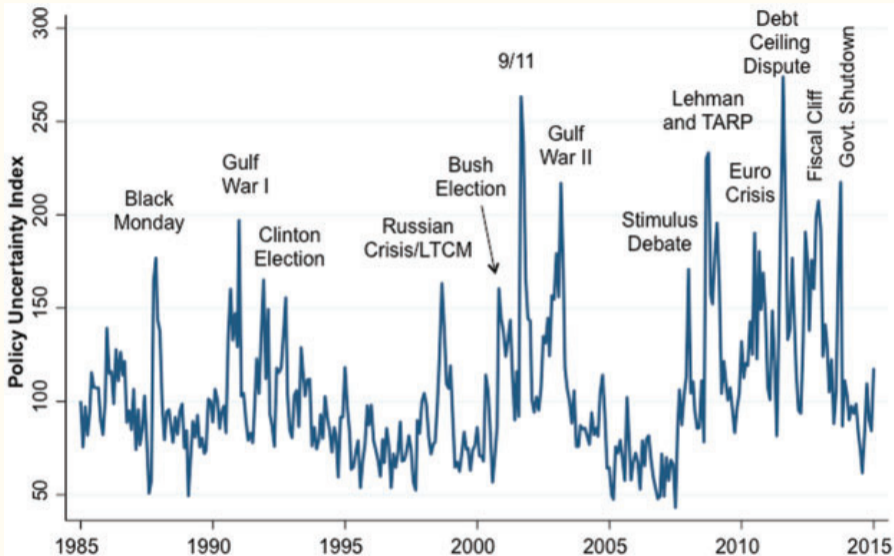


NEW-YORK:

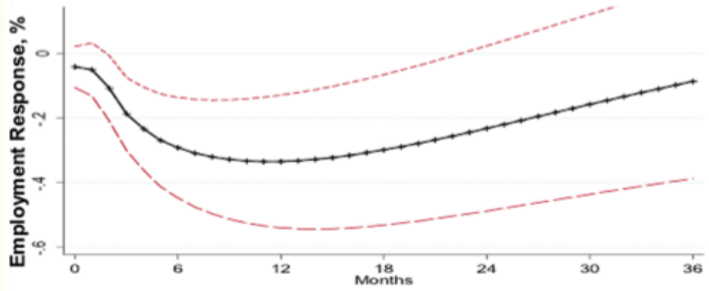
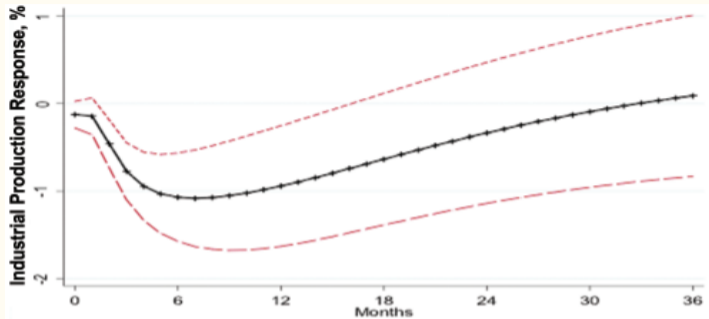
PRINTED AND SOLD BY J. AND A. McLEAN,  
No. 41, HANOVER-SQUARE.  
M,DCC,LXXXVIII.

*Mr Jefferson's copy*

- Large area with many other applications in economics:
  1. Measurement.
  2. Prediction.
  3. Causality.

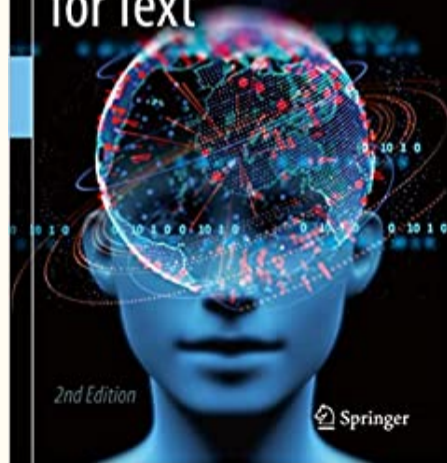


Index reflects scaled monthly counts of articles containing 'uncertain' or 'uncertainty', 'economic' or 'economy', and one or more policy relevant terms: 'regulation', 'federal reserve', 'deficit', 'congress', 'legislation', or 'white house'. The series is normalized to mean 100 from 1985-2009 and based on queries run on 2 February, 2015 for the USA Today, Miami Herald, Chicago Tribune, Washington Post, LA Times, Boston Globe, SF Chronicle, Dallas Morning News, NY Times, and the Wall Street Journal.




Charu C. Aggarwal

# Machine Learning for Text



*2nd Edition*

 Springer



AMONG THE NUMEROUS AS  
TENDENCY TO BREAK AND C  
FOR THEIR CHARACTER AND  
DUE VALUE ON ANY PLAN V  
TIC, AND  
RENDS HA  
DRENTH THE  
BOTH AN  
D THAT THE  
FROM OUR  
IS THAT OF  
ES ARE TOO  
HURRIED  
E OF KIND  
SITUATION  
RENDS EL  
AND, POSIT  
DED FROM  
S WITH MY  
S A PLEASU

# A New Framework for Machine Learning and the Social Sciences

Justin Grimmer | Margaret E. Roberts | Brandon M. Stewart

# DELIBERATING AMERICAN MONETARY POLICY

A TEXTUAL ANALYSIS

CHERYL SCHONHARDT-BAILEY

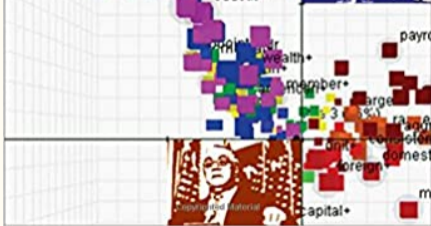


Copyrighted Material

Copyrighted Material

Copyrighted Material

Copyrighted Material



## Formalizing text

---

## Some terminology

- *Corpus*: the dataset under consideration (e.g., corporate reports, political speeches, statements, court decisions, newspaper articles, tweets, ...).
  - Third-declension neutral noun in Latin: nominative plural *corpora*.
- *Document*: each of the components of the corpus.
- *Terms*: each of the components of a document (usually words).
- *Ngrams*: Adjacent terms that we may want to handle together (“United States,” “high unemployment”).
- *Metadata*: covariates associated with each document (not always present).

# What is text?

- Formally, a text is an ordered string of characters.
- Some of these may be from the Latin alphabet – ‘a’, ‘A’ – but there may also be:
  1. Decorated Latin letters (e.g., ú).
  2. Non-Latin alphabetic characters (e.g., Chinese, Arabic, Hebrew).
  3. Punctuation (e.g., ‘!’).
  4. White spaces, tabs, newlines.
  5. Numbers.
  6. Non-alphanumeric characters (e.g., ‘@’).

## Text wrangling

---

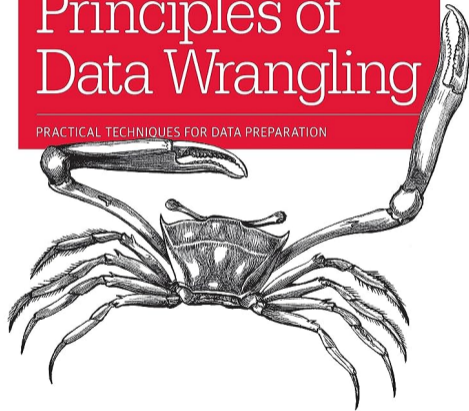
# From files to databases, I

- First step is to *pre-process* strings to obtain a cleaner representation.
- This is often the “secret sauce of LLM.”
- **Rattenbury *et al.*, (2017)** claim that between 50% and 80% of real-life data analysis is spent with data wrangling.
- Turning raw text files into structured databases is often a challenge:
  1. Separate metadata from text.
  2. Identify relevant portions of the text (paragraphs, sections, etc).
  3. Remove graphs and charts.
  4. Often, concerns about copyright, consent, safety, and privacy considerations.

O'REILLY®

# Principles of Data Wrangling

PRACTICAL TECHNIQUES FOR DATA PREPARATION



Tye Rattenbury, Joe Hellerstein,  
Jeffrey Heer, Sean Kandel & Connor Carreras

## From files to databases, II

- First step for non-editable files is conversion to an editable format, usually with optical character recognition (OCR) software.
- This is another potential application of deep learning.
- Check, for example: [Shen \*et al.\* \(2021\), LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis.](#)
- With raw text files, we can use regular expressions to identify relevant patterns.
- HTML and XML pages provide structure through tagging.
- If all else fails, manual extraction.



A NEW YORK TIMES NOTABLE BOOK

"A stupendous achievement, a triumph of historical research and imagination. No serious historian can write about the politics, diplomacy and economics of the 19th century in the same way."

—Robert Skidelsky, *The New York Review of Books*



# THE HOUSE OF ROTHSCHILD

*The World's Banker*  
1849 - 1999

—  
NIALL  
FERGUSON

## Raw text files

The Quartz guide to bad data,

<https://qz.com/572338/the-quartz-guide-to-bad-data/>

I once acquired the complete dog licensing database for Cook County, Illinois. Instead of requiring the person registering their dog to choose a breed from a list, the creators of the system had simply given them a text field to type into. As a result this database contained at least 250 spellings of Chihuahua.

- Issues:
  1. Inconsistent spelling and historical changes.
  2. N/A, blank, or null values.
  3. 0 values (or -1 or dates 1900, 1904, 1969, or 1970).
  4. Text is garbled.
  5. Lines ends are garbled.
  6. Text comes from OCR.

WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.

OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!



BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!



IT'S HOPELESS!

EVERYBODY STAND BACK.



I KNOW REGULAR EXPRESSIONS.



# Regular expressions I

- Regular expressions: sequence of characters that specifies a search pattern.
- You need to learn a programming language that manipulates regular expressions efficiently.
- About regular expressions in general:
  1. Tutorial: <https://www.regular-expressions.info/reference.html>.
  2. Online trial: <https://regexr.com/>.

## Regular expressions II

- Modern programming languages have powerful regular expressions capabilities.
- In Python: [https://www.tutorialspoint.com/python/python\\_reg\\_expressions.htm](https://www.tutorialspoint.com/python/python_reg_expressions.htm).
- In R: [https://evoldyn.gitlab.io/evomics-2018/ref-sheets/R\\_strings.pdf](https://evoldyn.gitlab.io/evomics-2018/ref-sheets/R_strings.pdf).
  1. Key packages: `dplyr`, `stringr`, and `tidyr` part of `tidyverse`.
  2. In particular, learn to use the piping command from `dplyr` to make code more readable.
  3. Look also at <https://www.tidytextmining.com/> for text mining.

# Pre-processing

---

## Pre-processing I: tokenization

- Tokenization is splitting a raw character string into individual elements of interest.
- Often, these elements are words, but we may also want to keep numbers or punctuation as well.
- Simple rules work well, but not perfectly. For example, splitting on white space and punctuation will separate hyphenated phrases, as in 'risk-averse agent' and contractions, as in 'aren't'.
- In practice, you should (probably) use a specialized library for tokenization.

## Pre-processing II: stopword removal

- The frequency distribution of words in natural languages is highly skewed, with a few dozen words accounting for the bulk of a text.
- These *stopwords* are typically stripped out of the tokenized representation of text as they take up memory but do not help distinguish one document from another.
- Examples from English are 'a', 'the', 'to', 'for,' and so on.
- No definitive list, but example on <http://snowball.tartarus.org/algorithms/english/stop.txt>.
- Also common to drop rare words, for example those that appear in less than some fixed percentage of documents.



## Pre-processing III: linguistic roots

- For many applications, the relevant information in tokens is their linguistic root, not their grammatical form. We may want to treat 'prefer', 'prefers', 'preferences' as equivalent tokens.
- Two options:
  1. *Stemming*: Deterministic algorithm for removing suffixes. Porter stemmer is popular.
  2. Stem need not be an English word: Porter stemmer maps 'inflation' to 'inflat'.
  3. *Lemmatizing*: Tag each token with its part of speech, then look up each (word, POS) pair in a dictionary to find the linguistic root.
  4. E.g., 'saw' tagged as a verb would be converted to 'see', 'saw' tagged as a noun left unchanged.
- A related transformation is *case-folding* each alphabetic token into lowercase. Not without ambiguity, e.g., 'US' and 'us' are each mapped into the same token.

## Pre-processing IV: multi-word phrases

- Sometimes groups of individual tokens like “Bank Indonesia” or “text mining” have a specific meaning.
- One ad-hoc strategy is to tabulate the frequency of all unique two-token (bigram) or three-token (trigram) phrases in the data and convert the most common into a single token.
- In FOMC data, the most common bigrams include ‘interest rate’, ‘labor market’, ‘basis point’; most common trigrams include ‘federal fund rate’, ‘real interest rate’, ‘real gdp growth’, ‘unit labor cost’.

## More systematic approach

- Some phrases have meaning because they stand in for specific names, like “Bank Indonesia”. One can use named-entity recognition software applied to raw, tokenized text data to identify these.
- Other phrases have meaning because they denote a recurring concept, like “housing bubble”. To find these, one can apply part-of-speech tagging, then tabulate the frequency of the following tag patterns:

AN/NN/AAN/ANN/NAN/NNN/NPN.

- See chapter on collocations in *Manning and Schütze's Foundations of Statistical Natural Language Processing* for more details.

# Notation

- The corpus is composed of  $D$  documents indexed by  $d$ .
- After pre-processing, each document is a finite, length- $N_d$  list of terms  $\mathbf{w}_d = (w_{d,1}, \dots, w_{d,N_d})$  with generic element  $w_{d,n}$ .
- Let  $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_D)$  be a list of all terms in the corpus, and let  $N \equiv \sum_d N_d$  be the total number of terms in the corpus.
- Suppose there are  $V$  **unique** terms in  $\mathbf{w}$ , where  $1 \leq V \leq N$ , each indexed by  $v$ .
- We can map each term in the corpus into this index so that  $w_{d,n} \in \{1, \dots, V\}$ .
- Let  $x_{d,v} \equiv \sum_n \mathbb{1}(w_{d,n} = v)$  be the count of term  $v$  in document  $d$ .

## Example

- Consider three documents:

- 'En un lugar'
- 'Muchos años después'
- 'Después del lugar'

- Set of  $V = 7$  unique terms:

{en, un, lugar, muchos, años, después, del}

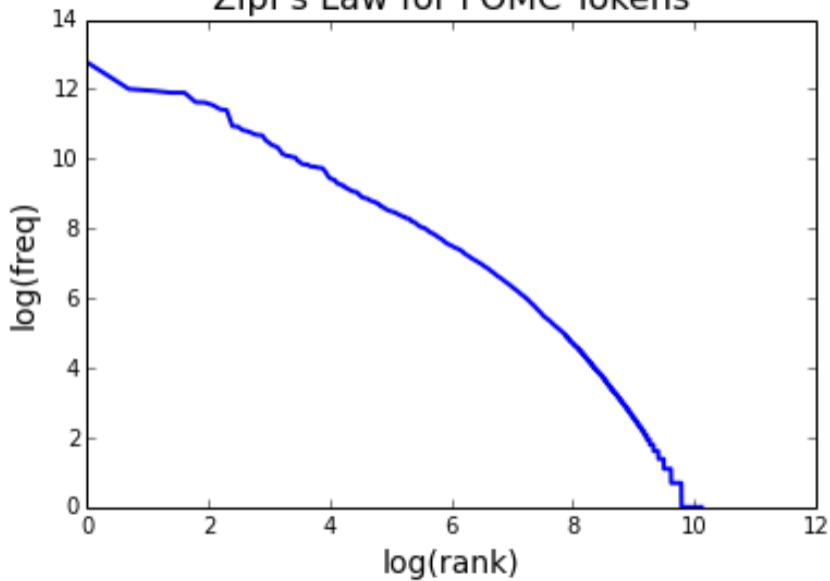
- Index:

	en	un	lugar	muchos	años	después	del
	1	2	3	4	5	6	7

- We then have  $\mathbf{w}_1 = (1, 2, 3)$ ;  $\mathbf{w}_2 = (4, 5, 6)$ ;  $\mathbf{w}_3 = (6, 2, 3)$ .
- Moreover  $x_{1,1} = 1$ ,  $x_{2,1} = 0$ ,  $x_{3,1} = 0$ , etc.

- We need to reduce the dimensionality of the document-term matrix to incorporate it into the empirical analysis.
- Popular strategy: define a list of terms that capture the content of interest, and then express documents in terms of (perhaps normalized by Zipf's Law) terms counts.
- Strategy is referred to as dictionary methods.
- Where do the dictionaries come from?
  1. Pre-defined lists.
  2. Domain expertise.
  3. Ability to predict objective label.
- Limitations.

## Zipf's Law for FOMC Tokens



## Probabilistic thinking

---



# Multinomial distribution I

- Multivariate generalization of the Bernoulli, categorical, and binomial distributions.
- An experiment with  $n$  independent trials over order  $K \geq 2$  and probability parameters  $\beta_1, \beta_2, \dots, \beta_K$  has a probability mass function:

$$f(x_1, x_2, \dots, x_K | n, \beta_1, \beta_2, \dots, \beta_K) = \frac{n!}{x_1! x_2! \dots x_K!} \prod_{i=1}^K \beta_i^{x_i}$$

where

$$\sum_{i=1}^K x_i = n$$

$$\sum_{i=1}^K \beta_i = 1$$

## Multinomial distribution II

- Note alternative, yet equivalent, form of the pdf:

$$f(x_1, x_2, \dots, x_K | n, \beta_1, \beta_2, \dots, \beta_K) = \frac{\Gamma\left(\sum_{i=1}^K x_i + 1\right)}{\prod_{i=1}^K \Gamma(x_i + 1)} \prod_{i=1}^K \beta_i^{x_i}$$

- However, this normalization constant is not very important for inference.
- Moments:

$$\mathbb{E}(X_i) = n\beta_i$$

$$\text{Var}(X_i) = n\beta_i(1 - \beta_i)$$

$$\text{Cov}(X_i, X_j) = -n\beta_i\beta_j \text{ for } i \neq j$$

# Dirichlet distribution I

- Multivariate generalization of the beta distribution.
- An experiment with order  $K \geq 2$  and concentration parameters  $\beta_1, \beta_2, \dots, \beta_K$  has pdf in the  $K - 1$  simplex:

$$f(x_1, x_2, \dots, x_K | \beta_1, \beta_2, \dots, \beta_K) = \frac{1}{\mathcal{B}(\beta_1, \beta_2, \dots, \beta_K)} \prod_{i=1}^K x_i^{\beta_i - 1}$$

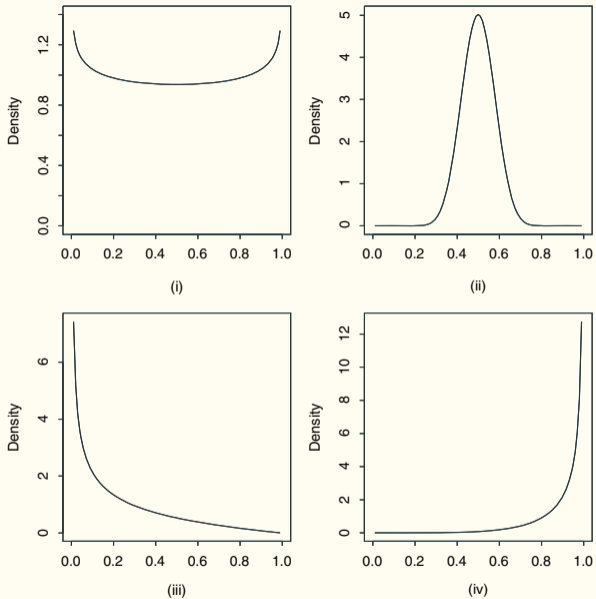
where

$$\sum_{i=1}^K x_i = 1 \text{ and } x_i \geq 0 \text{ for all } i = 1, \dots, K$$

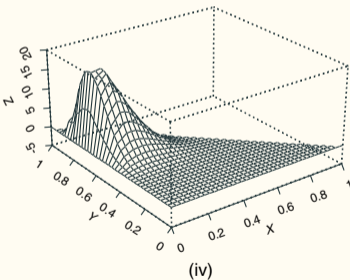
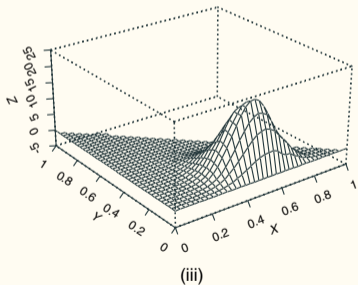
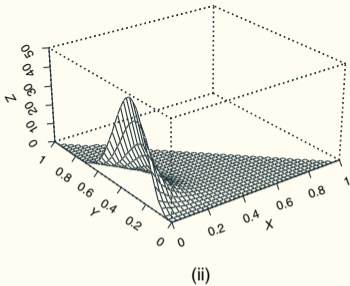
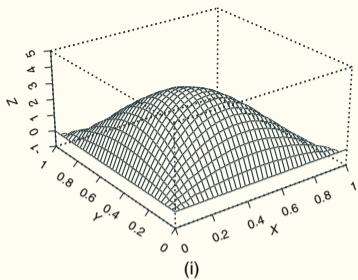
$$\mathcal{B}(\beta_1, \beta_2, \dots, \beta_K) = \frac{\prod_{i=1}^K \Gamma(\beta_i)}{\Gamma(\sum_{i=1}^K \beta_i)}$$

$$\Gamma(\beta) = \int_0^{\infty} x^{\beta-1} e^{-x} dx$$

- Each order is often called a category.



**Figure 1.1** Plots of the densities of  $Z \sim \text{Beta}(a_1, a_2)$  with various parameter values (i)  $a_1 = a_2 = 0.9$ ; (ii)  $a_1 = a_2 = 20$ ; (iii)  $a_1 = 0.5$  and  $a_2 = 2$ ; (iv)  $a_1 = 5$  and  $a_2 = 0.4$ .



**Figure 2.1** Plots of the densities of  $(x_1, x_2)^T \sim \text{Dirichlet}(a_1, a_2; a_3)$  on  $\mathbb{V}_2$  with various parameter values: (i)  $a_1 = a_2 = a_3 = 2$ ; (ii)  $a_1 = 1, a_2 = 5, a_3 = 10$ ; (iii)  $a_1 = 10, a_2 = 3, a_3 = 8$ ; (iv)  $a_1 = 2, a_2 = 10, a_3 = 4$ .

- Moments:

$$\mathbb{E}(X_i) = \frac{\beta_i}{\sum_{i=1}^K \beta_i}$$

$$\text{Var}(X_i) = \frac{\beta_i \left( \sum_{i=1}^K \beta_i - \beta_i \right)}{\left( \sum_{i=1}^K \beta_i \right)^2 \left( \sum_{i=1}^K \beta_i + 1 \right)}$$

$$\text{Cov}(X_i, X_j) = \frac{-\beta_i \beta_j}{\left( \sum_{i=1}^K \beta_i \right)^2 \left( \sum_{i=1}^K \beta_i + 1 \right)} \text{ for } i \neq j$$

- Conjugate prior to the multinomial distribution.

## Simple probability model

- Consider the list of terms  $\mathbf{w} = (w_1, \dots, w_N)$  where  $w_n \in \{1, \dots, V\}$ .
- Suppose that each term is i.i.d., and that  $p(w_n = v) = \beta_v \in [0, 1]$ .
- Let  $\beta = (\beta_1, \dots, \beta_V) \in \Delta^{V-1}$  be the parameter vector we are interested in.
- The probability of the ordered data given the parameters is

$$p(\mathbf{w}|\beta) = \prod_n \sum_v \mathbb{1}(w_n = v) \beta_v = \prod_v \beta_v^{x_v}$$

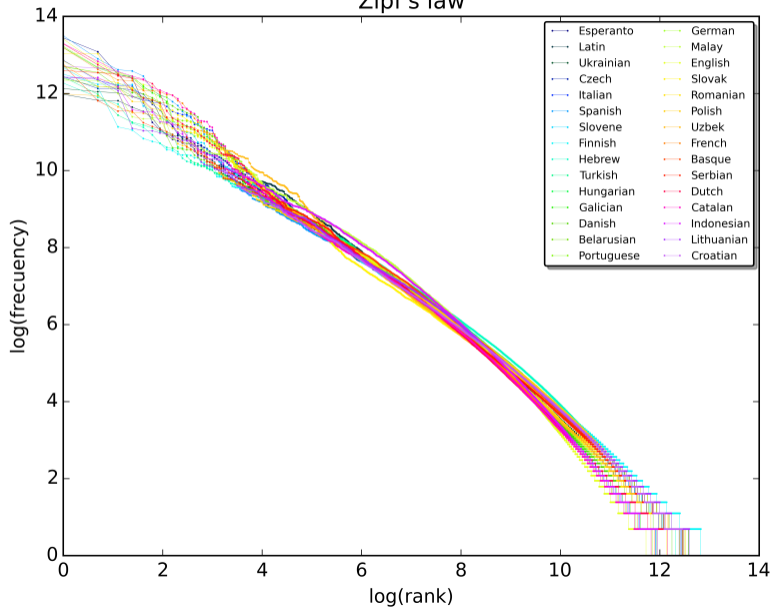
where  $x_v$  is the count of term  $v$  in  $\mathbf{w}$ .

- Notice that term counts are a sufficient statistic for  $\mathbf{w}$  in the estimation of  $\beta$ . The independence assumption provides statistical foundations for the bag-of-words model.

- Why?
- Highly parameterized model.
- Words are not uniformly distributed in texts.
- McMcs simplify inference.
- The Dirichlet is a great prior: conjugacy and interpretability.



## Zipf's law



## Posterior distribution

- We can add term counts to the prior distribution's parameters to form posterior:

$$p(\boldsymbol{\beta}|\mathbf{w}) \propto p(\mathbf{w}|\boldsymbol{\beta})p(\boldsymbol{\beta}) \propto \prod_{v=1}^V \beta_v^{x_v} \prod_{v=1}^V \beta_v^{\eta_v-1} = \prod_{v=1}^V \beta_v^{x_v+\eta_v-1}$$

- Posterior is a Dirichlet with parameters  $(\eta'_1, \dots, \eta'_V)$  where  $\eta'_v \equiv \eta_v + x_v$ .
- Dirichlet hyperparameters can be viewed as pseudo-counts.
- Thus, we obtain

$$\mathbb{E}[\beta_v|\mathbf{w}] = \frac{\eta_v + x_v}{\sum_v \eta_v + N}$$

which also corresponds to the predictive distribution  $p[w_{N+1} = v|\mathbf{w}]$ .

- MAP (mode) estimator of  $\beta_v$  is

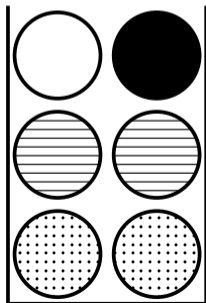
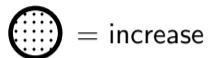
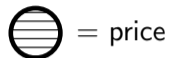
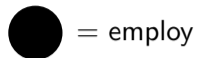
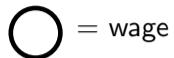
$$\mathbb{E}[\beta_v|\mathbf{w}] = \frac{\eta_v + x_v - 1}{\sum_v \eta_v + N - 2}$$

- Nice asymptotic properties.

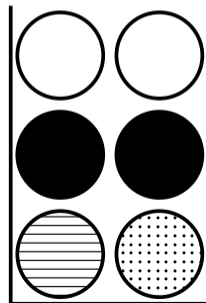
## Generative latent variable model (LVM)

- Documents in a corpus might have a more complex structure.
- $k$  separate categorical distributions (“topics”), each with parameter vector  $\beta_k$ .
- $\theta_{d,k}$  is the share of topic  $k$  in document  $d$ .
- Thus, each document is represented on a space of topics with  $\theta_d \in \Delta^{K-1}$  instead of a raw vocabulary space.
- The probability that topic  $k$  generate term  $v$  is  $\beta_{k,v}$ .
- General probabilistic structure:  $\mathbf{x}_d \sim \text{MN}(\sum_k \theta_{d,k} \beta_k, N_d)$ .
- Let  $\beta = (\beta_1, \dots, \beta_K)$  and  $\theta = (\theta_1, \dots, \theta_D)$ .

# Topics as urns



"Inflation" Topic



"Labor" Topic

# How do we model topics?

- Two approaches:

1. Mixture models: every document belongs to single category  $z_d \in \{1, \dots, K\}$ , which is independent across documents and drawn from  $p(z_d = k) = \rho_k$ .

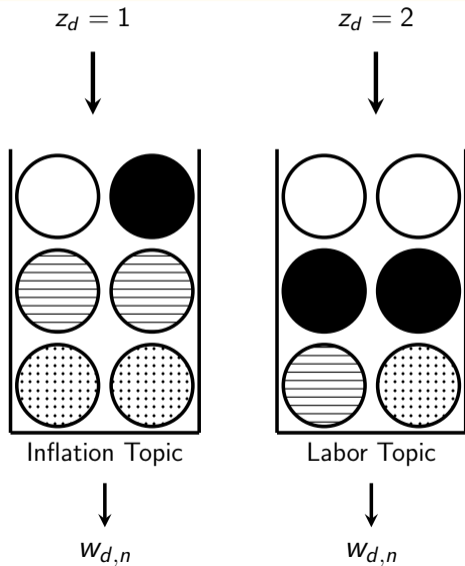
We then have

$$\theta_{d,k} = \begin{cases} 1 & \text{if } z_d = k \\ 0 & \text{otherwise} \end{cases} .$$

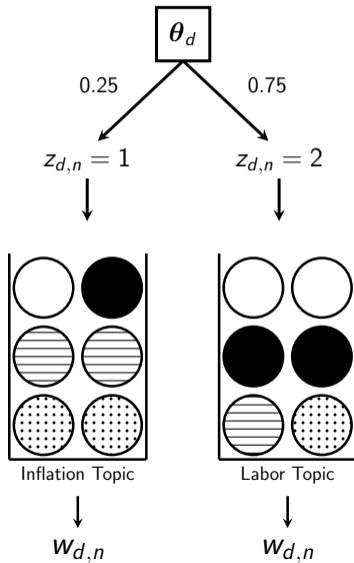
2. Mixed-membership models: we can assign each word in each document to a topic.

Let  $z_{d,n} \in \{1, \dots, K\}$  be the topic assignment of  $w_{d,n}$ ;  $\mathbf{z}_d = (z_{d,1}, \dots, z_{d,N_d})$ ; and  $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_D)$ .

# Mixture model



# Mixed-membership model



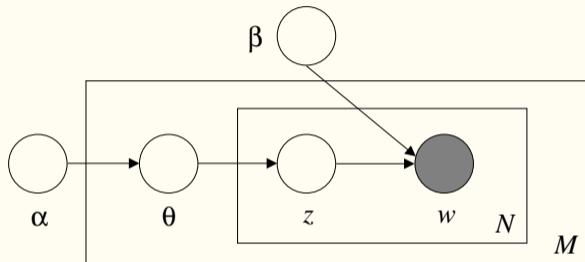
# A canonical model

- Latent Dirichlet allocation (Blei, Ng, and Jordan 2003).
- Extremely popular and the base for more general models.
- Structure:
  1. Draw  $\theta_d$  independently for  $d = 1, \dots, D$  from  $\text{Dirichlet}(\alpha)$ .
  2. Each word  $w_{d,n}$  in document  $d$  is generated from a two-step process:
    - 2.1 Draw topic assignment  $z_{d,n}$  from  $\theta_d$ .
    - 2.2 Draw  $w_{d,n}$  from  $\beta_{z_{d,n}}$ .
- Estimate hyperparameters  $\alpha$  and term probabilities  $\beta_1, \dots, \beta_K$ .

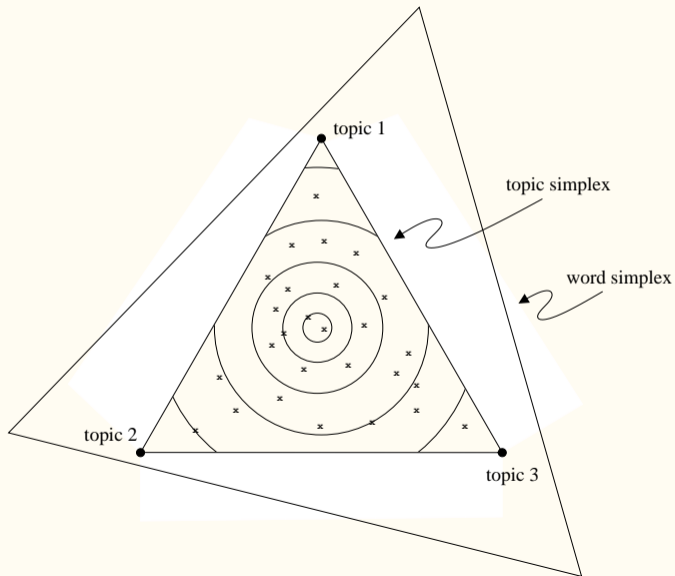


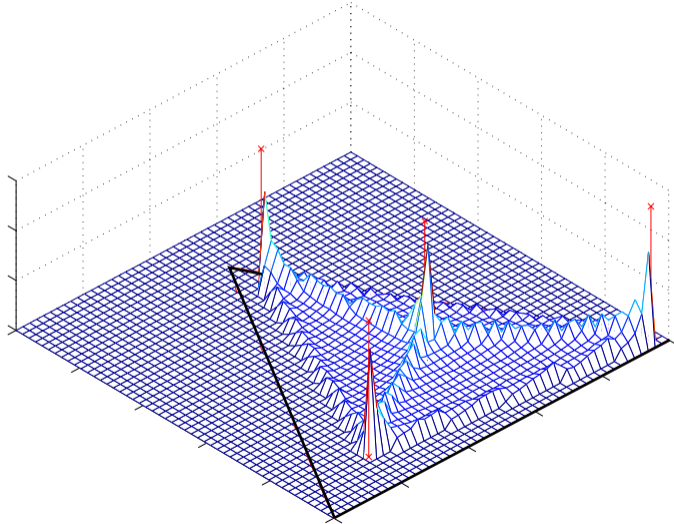
# A modification of the Latent Dirichlet allocation

- We can slightly modify the previous model to ease, later on, the implementation of a Gibbs sampler.
- Structure:
  1. Draw  $\theta_d$  independently for  $d = 1, \dots, D$  from  $\text{Dirichlet}(\alpha)$ .
  2. Draw  $\beta_k$  independently for  $k = 1, \dots, K$  from  $\text{Dirichlet}(\eta)$ .
  3. Each word  $w_{d,n}$  in document  $d$  is generated from a two-step process:
    - 3.1 Draw topic assignment  $z_{d,n}$  from  $\theta_d$ .
    - 3.2 Draw  $w_{d,n}$  from  $\beta_{z_{d,n}}$ .
- Fix scalar values for  $\alpha$  and  $\eta$ .



Graphical model representation of LDA. The boxes are “plates” representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.





## Example statement: Yellen, March 2006, #51

Raw Data → Remove Stop Words → Stemming → Multi-word tokens = Bag of Words

We have noticed a change in the relationship between the core CPI and the chained core CPI, which suggested to us that maybe something is going on relating to substitution bias at the upper level of the index. You focused on the nonmarket component of the PCE, and I wondered if something unusual might be happening with the core CPI relative to other measures.

## Example statement: Yellen, March 2006, #51

Raw Data → Remove Stop Words → Stemming → Multi-word tokens = Bag of Words

We have noticed a change in the relationship between the core CPI and the chained core CPI which suggested to us that maybe something is going on relating to substitution bias at the upper level of the index. You focused on the nonmarket component of the PCE and I wondered if something unusual might be happening with the core CPI relative to other measures.

## Example statement: Yellen, March 2006, #51

Raw Data → Remove Stop Words → **Stemming** → Multi-word tokens = Bag of Words

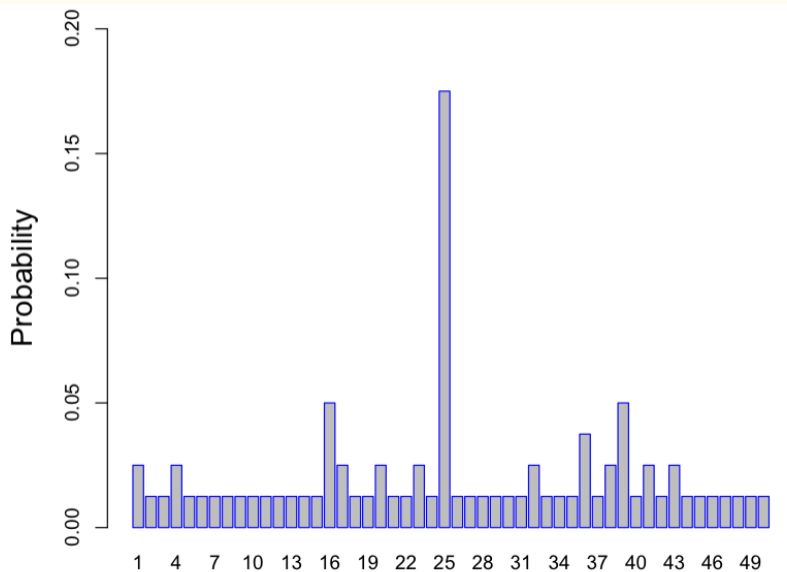
We have noticed a change in the relationship between the core CPI and the chained core CPI which suggested to us that maybe something is going on relating to substitution bias at the upper level of the index. You focused on the nonmarket component of the PCE and I wondered if something unusual might be happening with the core CPI relative to other measures.

## Example statement: Yellen, March 2006, #51

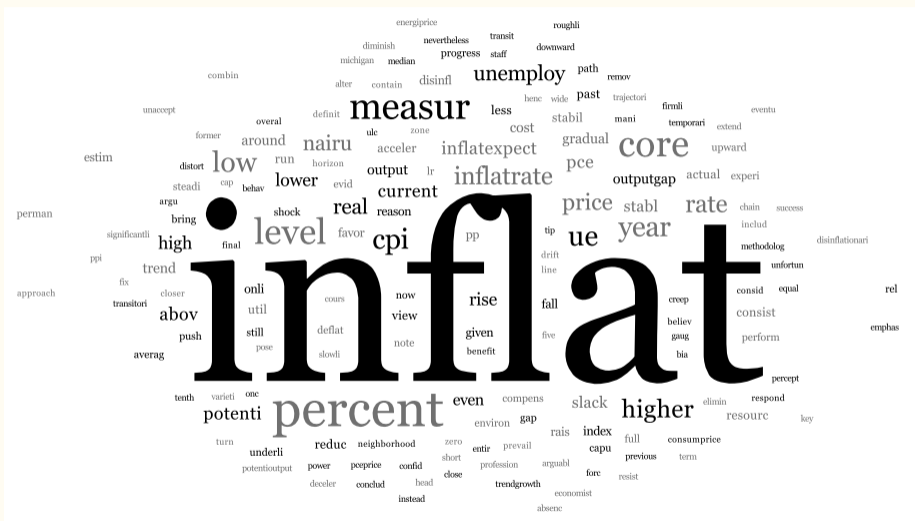
We have noticed a change in the relationship between the real interest rate and the real return on capital, which suggests to us that something is going on relating to institutions at the level of the real economy. You could use the market return component of the real return, and I considered if something unusual might be happening with the real return relative to other measures.

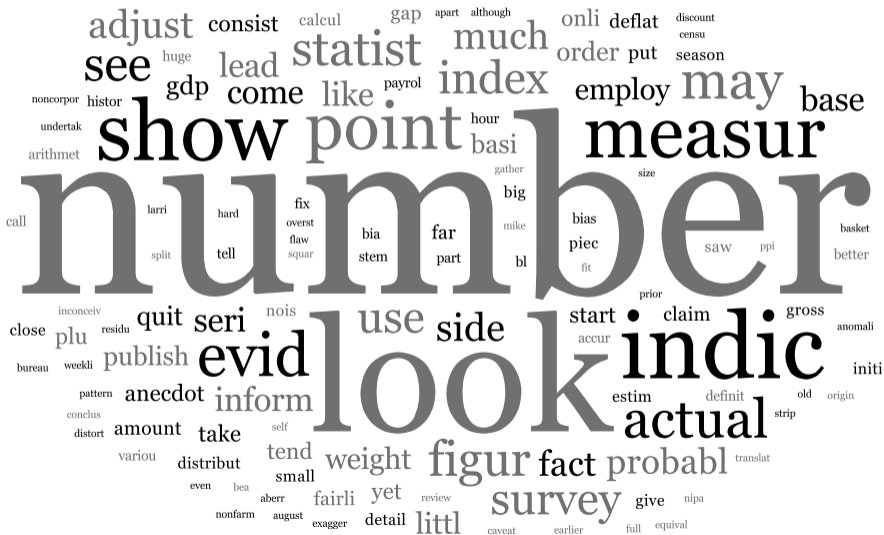


# Distribution of topics



# Topic 25

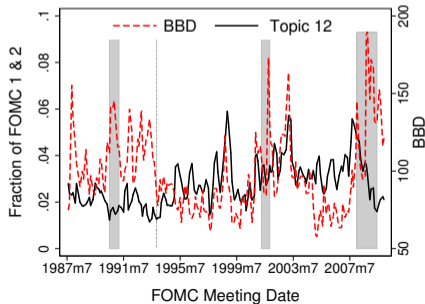
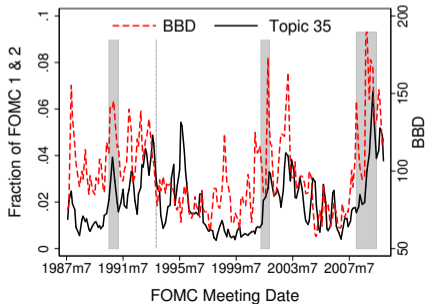
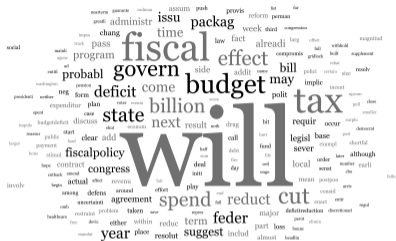




## Advantage of Flexibility

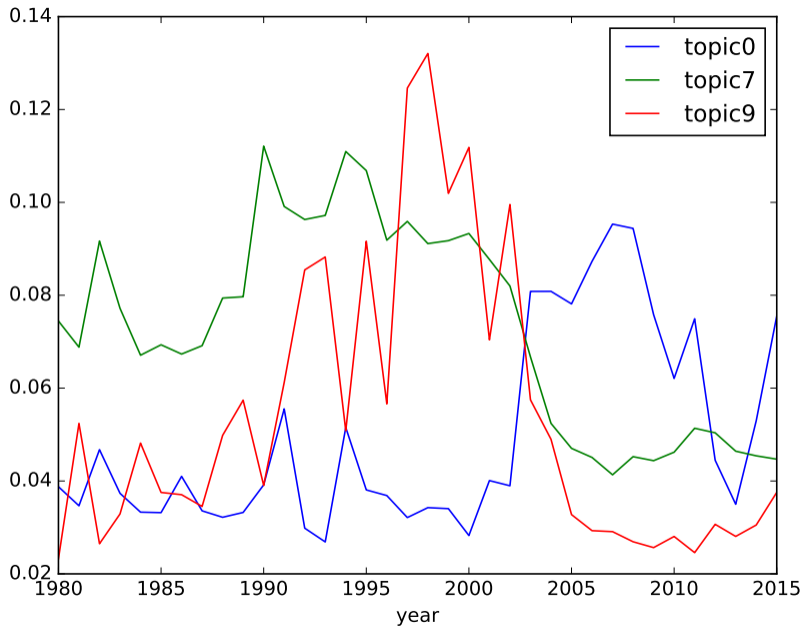
- 'measur' has probability 0.026 in topic 25, and probability 0.021 in topic 11.
- It gets assigned to 25 in this statement consistently due to the presence of other topic 25 words.
- In statements containing words on evidence and numbers, it consistently gets assigned to 11.
- Sampling algorithm can help place words in their appropriate context.

# Topics vs. BDD







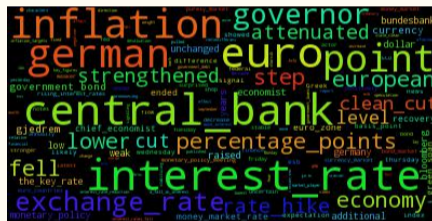


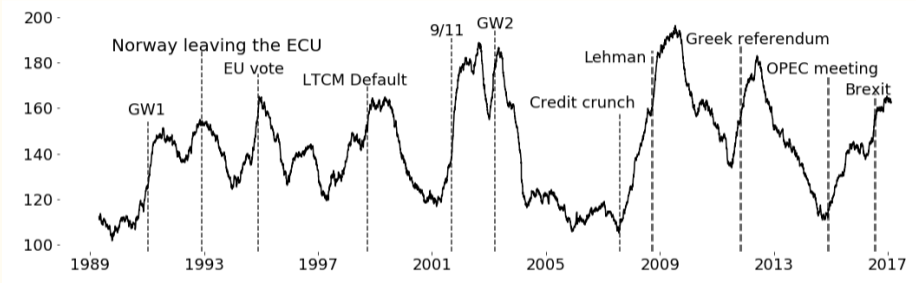


(a) Macroeconomics

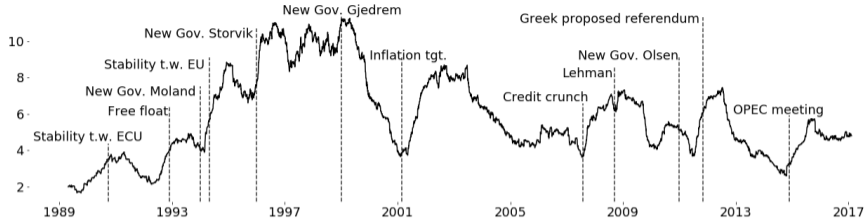


(b) Monetary policy

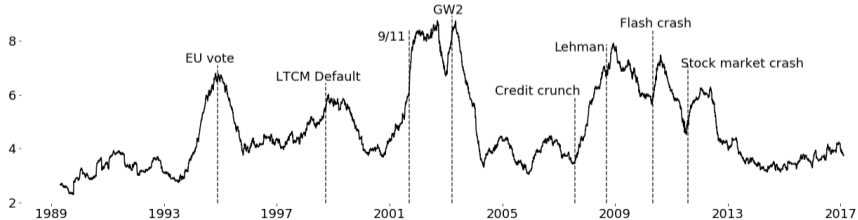




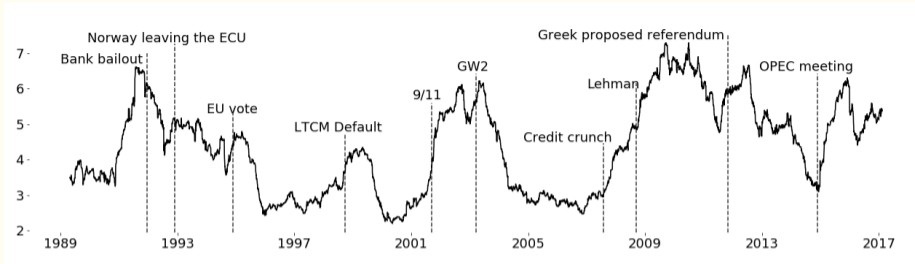
### (a) *Monetary policy uncertainty*



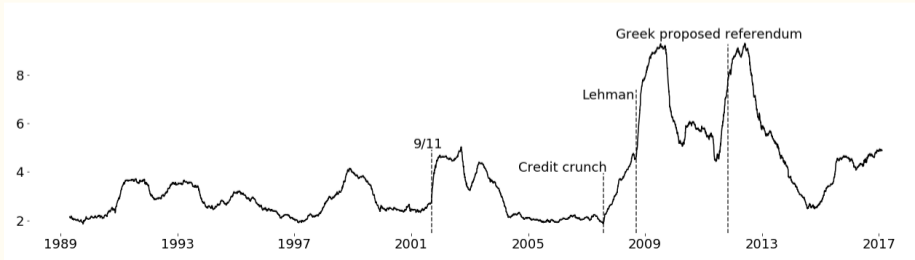
### (b) *Stock market uncertainty*

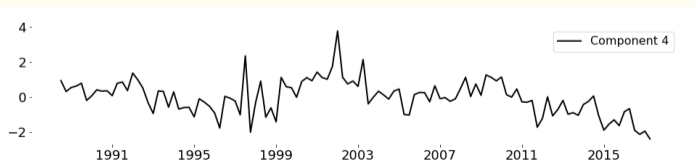
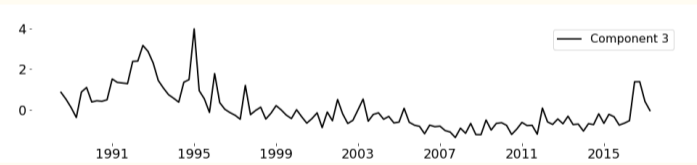
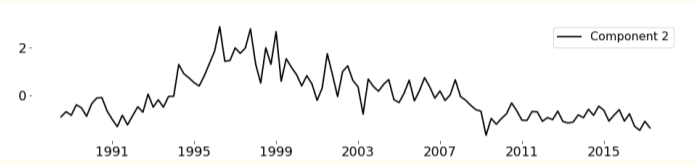
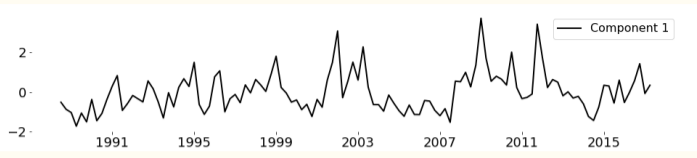


### (c) *Macroeconomics uncertainty*



### (d) *Fear uncertainty*





	Component 1	Component 2	Component 3	Component 4
Norway EPU	0.69	-0.16	0.25	-0.2
US VIX	0.66	-0.033	-0.24	0.33
US EPU	0.55	-0.59	-0.094	0.0053
JLN Finance	0.52	-0.06	-0.29	0.37
EU EPU	0.48	-0.41	-0.11	-0.4
JLN Macro	0.42	-0.28	-0.41	0.34
China EPU	0.41	-0.52	0.24	-0.32
RSMV	0.38	-0.33	-0.56	-0.19
UK EPU	0.27	-0.62	0.34	-0.49

## Posterior distribution

- The inference problem in LDA is to compute the posterior distribution over  $\mathbf{z}$ ,  $\theta$ , and  $\beta$  given the data  $\mathbf{w}$  and Dirichlet hyperparameters.
- The posterior distribution of the latent variables taking the parameters as given is:

$$p(\mathbf{z} = \mathbf{z}' | \mathbf{w}, \theta, \beta) = \frac{p(\mathbf{w} | \mathbf{z} = \mathbf{z}', \theta, \beta) p(\mathbf{z} = \mathbf{z}' | \theta, \beta)}{\sum_{\mathbf{z}'} p(\mathbf{w} | \mathbf{z} = \mathbf{z}', \theta, \beta) p(\mathbf{z} = \mathbf{z}' | \theta, \beta)}$$

- We can compute the numerator and each element of the denominator.
- But  $\mathbf{z}' \in \{1, \dots, K\}^N \Rightarrow$  there are  $K^N$  terms in the sum  $\Rightarrow$  intractable problem.
- For example, a 100-word corpus with 50 topics has  $\approx 7.88 \times 10^{169}$  terms.

- Different McMcs can handle this problem.
- For a basic implementation, a Gibbs sampler is easy to code and efficient.
- Outline:
  1. Sample from a multinomial distribution  $N$  times for the topic allocation variables.
  2. Sample from a Dirichlet  $D$  times for the document-specific mixing probabilities.
  3. Sample from a Dirichlet  $K$  times for the topic-specific term probabilities.
- We can improve upon the basic Gibbs sampler with collapsed sampling, i.e., analytically integrating out some variables in the joint likelihood and sampling the remainder ([Griths and Steyvers, 2004](#), and [Hansen, McMahan, and Prat, 2015](#)).



- By Bayes' Rule, we have  $p(\theta_d|\alpha, \mathbf{z}_d) \propto p(\mathbf{z}_d|\theta_d)p(\theta_d|\alpha)$ .
- Then

$$p(\mathbf{z}_d|\theta_d) = \prod_n \sum_k \mathbb{1}(z_{d,n} = k)\theta_{d,k} = \prod_k \theta_{d,k}^{n_{d,k}}.$$

- Putting this together, we arrive at

$$p(\theta_d|\alpha, \mathbf{z}_d) \propto \prod_k \theta_{d,k}^{n_{d,k}} \prod_k \theta_{d,k}^{\alpha-1} = \prod_k \theta_{d,k}^{n_{d,k}+\alpha-1}$$

which is a Dirichlet distribution.

## Sampling $\beta_k$

- By Bayes' Rule, we have  $p(\beta_k | \mathbf{z}, \mathbf{w}, \eta, \beta_{-k}) \propto p(\mathbf{z}, \mathbf{w} | \beta) p(\beta_k | \eta)$ .
- The likelihood function  $p(\mathbf{z}, \mathbf{w} | \beta)$  takes the form

$$\begin{aligned} p(\mathbf{z}, \mathbf{w} | \beta) &= \prod_d \prod_n \sum_v \sum_{k'} \mathbb{1}(w_{d,n} = v) \mathbb{1}(z_{d,n} = k') \beta_{k',v} \\ &= \prod_v \prod_{k'} \beta_{k',v}^{m_{k',v}} \\ &= \prod_v \beta_{k,v}^{m_{k,v}} \prod_v \prod_{k' \neq k} \beta_{k',v}^{m_{k',v}} \\ &\propto \prod_v \beta_{k,v}^{m_{k,v}} \end{aligned}$$

- Putting this together, we arrive at

$$p(\beta_k | \mathbf{z}, \mathbf{w}, \eta, \beta_{-k}) \propto \prod_v \beta_{k,v}^{m_{k,v}} \prod_v \beta_{k,v}^{\eta-1} = \prod_v \beta_{k,v}^{m_{k,v} + \eta - 1}$$

which is a Dirichlet distribution.

- Finally, note that:

$$\begin{aligned} & p(z_{d,n} = k | w_{d,n} = v, \theta_d, \beta) \\ = & \frac{p(w_{d,n} = v | z_{d,n} = k, \theta_d, \beta) p(z_{d,n} = k | \theta_d, \beta)}{\sum_k p(w_{d,n} = v | z_{d,n} = k, \theta_d, \beta) p(z_{d,n} = k | \theta_d, \beta)} \\ = & \frac{\theta_d^k \beta_k^v}{\sum_k \theta_d^k \beta_k^v} \end{aligned}$$

- There are three parameters to set to run the Gibbs sampling algorithm: number of topics  $K$  and hyperparameters  $\alpha, \eta$ .
- Priors are treated cavalierly in the literature.
- **Griffiths and Steyvers** recommend  $\eta = 200/V$  and  $\alpha = 50/K$ . Smaller values will tend to generate more concentrated distributions. See also **Wallach et al. (2009)**.
- Methods to choose  $K$ :
  1. Predict text well  $\rightarrow$  out-of-sample goodness-of-fit.
  2. Information criteria.
  3. Cohesion (focus on interpretability).

- Fit LDA on training data, obtain estimates of  $\hat{\beta}_1, \dots, \hat{\beta}_K$ .
- For test data, obtain  $\theta_d$  distributions via sampling as above, or else use uniform distribution.
- Compute log-likelihood of held-out data as

$$\ell(\mathbf{w} | \hat{\Theta}) = \sum_{d=1}^D \sum_{v=1}^V x_{d,v} \log \left( \sum_{k=1}^K \hat{\theta}_{d,k} \hat{\beta}_{k,v} \right)$$

- Higher values indicate better goodness-of-fit.

- Information criteria trade off goodness-of-fit with model complexity.
- There are various forms: AIC, BIC, DIC, etc.
- [Erosheva et al. \(2007\)](#) compare several in the context of an LDA-like model for survey data, and find that AICM is optimal.
- Let  $\mu_\ell = \frac{1}{S} \sum_s \ell(\mathbf{w} | \hat{\Theta}^s)$  be the average value of the log-likelihood across  $S$  draws of a Markov chain.
- Let  $\sigma_\ell^2 = \frac{1}{S} \sum_s \left( \ell(\mathbf{w} | \hat{\Theta}^s) - \mu_\ell \right)^2$  be the variance.
- The AICM is  $2(\mu_\ell - \sigma_\ell^2)$ .

# Formalizing interpretability

- Topics seem objectively interpretable in many contexts.
- Chang et al. (2009), in “Reading Tea Leaves: How Humans Interpret Topic Models,” propose an objective way of determining whether topics are interpretable.
- Two tests:
  1. *Word intrusion*. Form a set of top five words from topic  $k$  + word with low probability in topic  $k$ . Ask subjects to identify inserted word.
  2. *Topic intrusion*. Show subjects a snippet of a document + top three topics associated with it + randomly drawn other topic. Ask to identify the inserted topic.
- Estimate LDA and other topic models on NYT and Wikipedia articles for  $K = 50, 100, 150$ .

# Distribution of topics

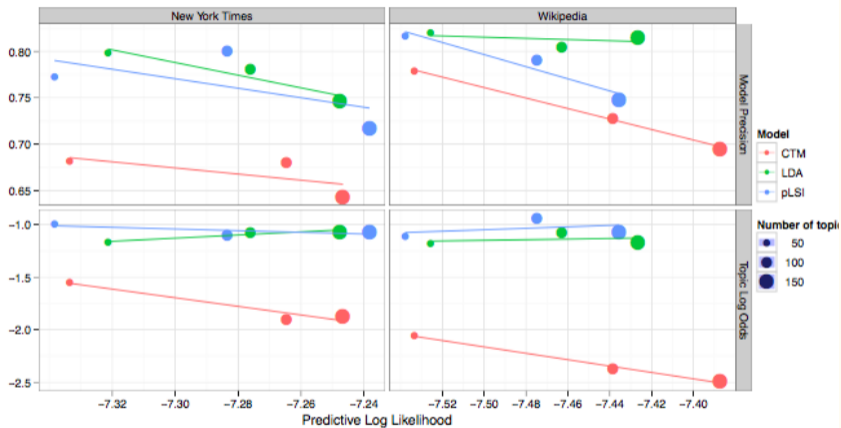


Figure 5: A scatter plot of model precision (top row) and topic log odds (bottom row) vs. predictive log likelihood. Each point is colored by model and sized according to the number of topics used to fit the model. Each model is accompanied by a regression line. Increasing likelihood does not increase the agreement between human subjects and the model for either task (as shown by the downward-sloping regression lines).