



IFC – Bank Indonesia International Workshop and Seminar on *“Big Data for Central Bank Policies / Building Pathways for Policy Making with Big Data”*

Bali, Indonesia, 23-26 July 2018

## Introduction to text mining<sup>1</sup>

Stephen Hansen,  
University of Oxford

---

<sup>1</sup> This presentation was prepared for the meeting. The views expressed are those of the author and do not necessarily reflect the views of the BIS, the IFC or the central banks and other institutions represented at the meeting.

# Introduction to Text Mining

## Bank Indonesia—IFC Workshop

Stephen Hansen  
University of Oxford

# Introduction

Most empirical work in economics relies on inherently quantitative data: prices, demand, votes, etc.

But a large amount of unstructured text is also generated in economic environments: company reports, policy committee deliberations, media articles, political speeches, etc.

One can use such data qualitatively, but increasing interest in treating text quantitatively.

My lectures will review how economists have done this until recently, and also more modern machine learning approaches.

# Textual Databases

A single observation in a textual database is called a *document*.

The set of documents that make up the dataset is called a *corpus*.

We often have covariates associated with each document that are sometimes called *metadata*.

# Example

In “Transparency and Deliberation” we use a corpus of verbatim FOMC transcripts from the era of Alan Greenspan:

- ▶ 149 meetings from August 1987 through January 2006.
- ▶ A document is a single statement by a speaker in a meeting (46,502).
- ▶ Associated metadata: speaker biographical information, macroeconomic conditions, etc.

# Data Sources

There are many potential sources for text data, such as:

1. PDF files or other non-editable formats
2. Word documents or other editable formats
3. Web pages
4. Application Programming Interfaces (API) for web applications.

# From Files to Databases

Turning raw text files into structured databases is often a challenge:

1. Separate metadata from text
2. Identify relevant portions of text (paragraphs, sections, etc)
3. Remove graphs and charts

First step for non-editable files is conversion to editable format, usually with optical character recognition software.

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, relatively cheap and reliable services exist for manual extraction.

# What is Text?

At an abstract level, text is simply a string of characters.

Some of these may be from the Latin alphabet—‘a’, ‘A’, ‘p’ and so on—but there may also be:

1. Decorated Latin letters (e.g. ö)
2. Non-Latin alphabetic characters (e.g. Chinese and Arabic)
3. Punctuation (e.g. ‘!’)
4. White spaces, tabs, newlines
5. Numbers
6. Non-alphanumeric characters (e.g. ‘@’)

**Key Question:** How can we obtain an informative, quantitative representation of these character strings? This is the goal of text mining.

First step is to *pre-process* strings to obtain a cleaner representation.



# Pre-Processing I: Tokenization

Tokenization is the splitting of 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 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>.

# 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 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:

- ▶ *Stemming*: Deterministic algorithm for removing suffixes. Porter stemmer is popular.  
Stem need not be an English word: Porter stemmer maps 'inflation' to 'inflat'.
- ▶ *Lemmatizing*: Tag each token with its part of speech, then look up each (word, POS) pair in a dictionary to find linguistic root.  
E.g. 'saw' tagged as verb would be converted to 'see', 'saw' tagged as noun left unchanged.

A related transformation is *case-folding* each alphabetic token into lowercase. Not without ambiguity, e.g. 'US' and 'us' each mapped into 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, 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.

## Example from NYT Corpus

$C(w^1 w^2)$	$w^1$	$w^2$	tag pattern
11487	New	York	A N
7261	United	States	A N
5412	Los	Angeles	N N
3301	last	year	A N
3191	Saudi	Arabia	N N
2699	last	week	A N
2514	vice	president	A N
2378	Persian	Gulf	A N
2161	San	Francisco	N N
2106	President	Bush	N N
2001	Middle	East	A N
1942	Saddam	Hussein	N N
1867	Soviet	Union	A N
1850	White	House	A N
1633	United	Nations	A N
1337	York	City	N N
1328	oil	prices	N N

# Pre-Processing of FOMC Corpus

	All terms	Alpha terms	No stopwords	Stems
# total terms	6249776	5519606	2505261	2505261
# unique terms	26030	24801	24611	13734



# 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 then 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:

1. 'stephen is nice'
2. 'john is also nice'
3. 'george is mean'

We can consider the set of unique terms as  $\{\text{stephen, is, nice, john, also, george, mean}\}$  so that  $V = 7$ .

Construct the following index:

stephen	is	nice	john	also	george	mean
1	2	3	4	5	6	7

We then have  $\mathbf{w}_1 = (1, 2, 3)$ ;  $\mathbf{w}_2 = (4, 2, 5, 3)$ ;  $\mathbf{w}_3 = (6, 2, 7)$ .

Moreover  $x_{1,1} = 1$ ,  $x_{2,1} = 0$ ,  $x_{3,1} = 0$ , etc.

# Document-Term Matrix

A popular quantitative representation of text is the *document-term matrix*  $\mathbf{X}$ , which collects the counts  $x_{d,v}$  into a  $D \times V$  matrix.

In the previous example, we have

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

The key characteristics of the document-term matrix are its:

1. High dimensionality
2. Sparsity

# Ngram Models

In the above example, we made an explicit choice to count individual terms, which destroys all information on word order.

In some contexts, this may be sufficient for our information needs, but in others we might lose valuable information.

We could alternatively have counted all adjacent two-term phrases, called bigrams or, more generally, all adjacent  $N$ -term phrases, called Ngrams.

This is perfectly consistent with the model described above, where  $v$  now indexes unique bigrams rather than unique unigrams:

stephen.is	is.nice	john.is	is.also	also.nice	george.is	is.mean
1	2	3	4	5	6	7

We then have  $\mathbf{w}_1 = (1, 2)$ ;  $\mathbf{w}_2 = (3, 4, 5)$ ;  $\mathbf{w}_3 = (6, 7)$ .

# Dimensionality Reduction through Keywords

The first approach to handling  $\mathbf{X}$  is to limit attention to a subset of columns of interest.

In the natural language context, this is equivalent to representing text using the distribution of keywords across documents.

One can either look at the incidence of keywords (Boolean search), or else their frequency (dictionary methods).

The researcher must decide in advance which are the keywords of interest.

# Application

The recent work of Baker, Bloom, and Davis on measuring economic policy uncertainty (<http://www.policyuncertainty.com/>) is largely based on a media index constructed via Boolean searches of US and European newspapers.

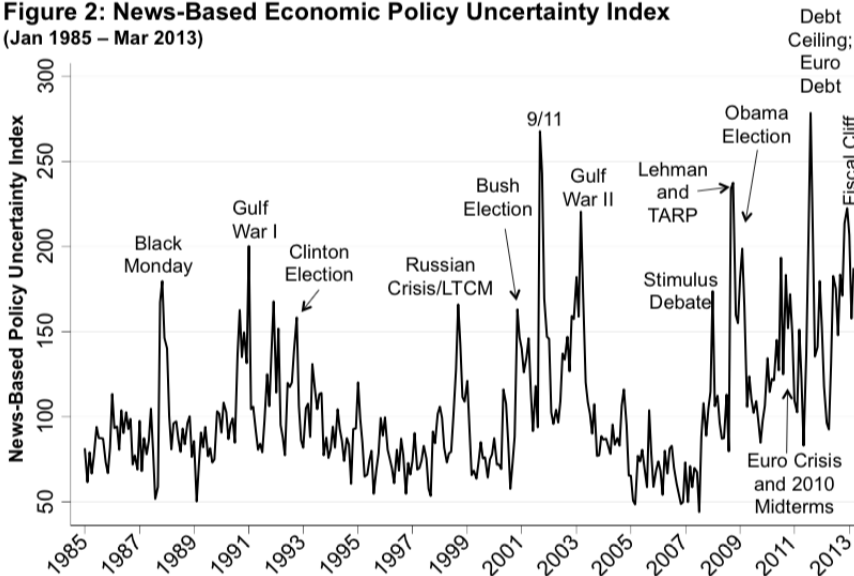
For each paper on each day since 1985, identify articles that contain:

1. “uncertain” OR “uncertainty”, AND
2. “economic” OR “economy”, AND
3. “congress” OR “deficit” OR “federal reserve” OR “legislation” OR “regulation” OR “white house”

Normalize resulting article counts by total newspaper articles that month.

# Results

**Figure 2: News-Based Economic Policy Uncertainty Index**  
(Jan 1985 – Mar 2013)



# Why Text?

VIX is an asset-based measure of uncertainty: implied S&P 500 volatility at 30-day horizon using option prices.

So what does text add to this?

1. Focus on broader type of uncertainty besides equity prices.
2. Much richer historical time series.
3. Cross-country measures.



# Term Weighting

Dictionary methods are based on raw counts of words.

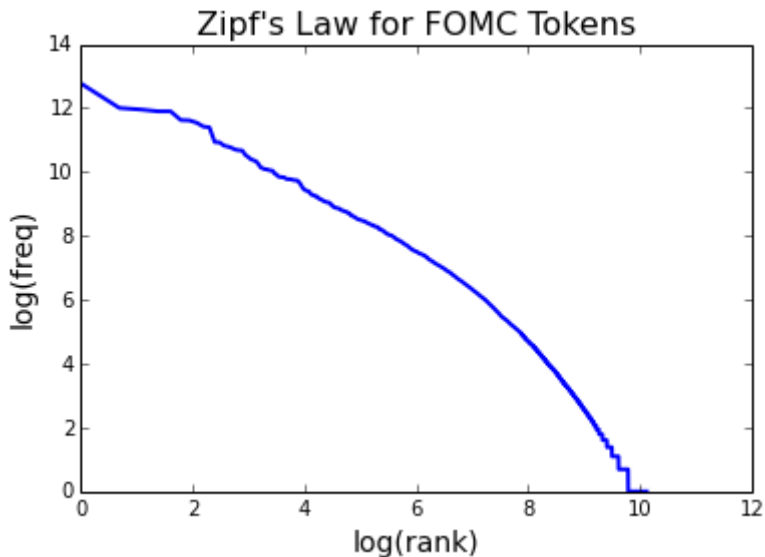
But the frequency of words in natural language can distort raw counts.

Zipf's Law is an empirical regularity for most natural languages that maintains that the frequency of a particular term is inversely proportional to its rank.

Means that a few terms will have very large counts, many terms have small counts.

Example of a *power law*.

# Zipf's Law in FOMC Transcript Data



# Rescaling Counts

Let  $x_{d,v}$  be the count of the  $v$ th term in document  $d$ .

To dampen the power-law effect can express counts as

$$tf_{d,v} = \begin{cases} 0 & \text{if } x_{d,v} = 0 \\ 1 + \log(x_{d,v}) & \text{otherwise} \end{cases}$$

which is the *term frequency* of  $v$  in  $d$ .

# Thought Experiment

Consider a two-term dictionary  $\mathfrak{D} = \{v', v''\}$ .

Suppose two documents  $d'$  and  $d''$  are such that:

$$x_{d',v'} > x_{d'',v'} \text{ and } x_{d',v''} < x_{d'',v''}.$$

Now suppose that no other document uses term  $v'$  but every other document uses term  $v''$ .

Which document is “more about” the theme the dictionary captures?

# Inverse Document Frequency

Let  $df_v$  be the number of documents that contain the term  $v$ .

The *inverse document frequency* is

$$\text{idf}_v = \log \left( \frac{D}{df_v} \right),$$

where  $D$  is the number of documents.

Properties:

1. Higher weight for words in fewer documents.
2. Log dampens effect of weighting.

# TF-IDF Weighting

Combining the two observations from above allows us to express the *term frequency - inverse document frequency* of term  $v$  in document  $d$  as

$$\text{tf-idf}_{d,v} = \text{tf}_{d,v} \times \text{idf}_v.$$

Gives prominence to words that occur many times in few documents.

Can now score each document as  $s_d = \sum_{v \in \mathcal{V}} \text{tf-idf}_{d,v}$  and then compare.

In practice, this can provide better results than simple counts.

# Data-Driven Stopwords

Stopword lists are useful for generic language, but there are also context-specific frequently used words.

For example, in a corpus of court proceedings, words like ‘lawyer’, ‘law’, ‘justice’ will show up a lot.

Can also define term-frequency across entire corpus as

$$tf_v = 1 + \log \left( \sum_d x_{d,v} \right).$$

One can then rank each term in the corpus according to  $tf_v \times idf_v$ , and choose a threshold below which to drop terms.

This provides a means for data-driven stopwords selection.

# Stem Rankings in FOMC Transcript Data

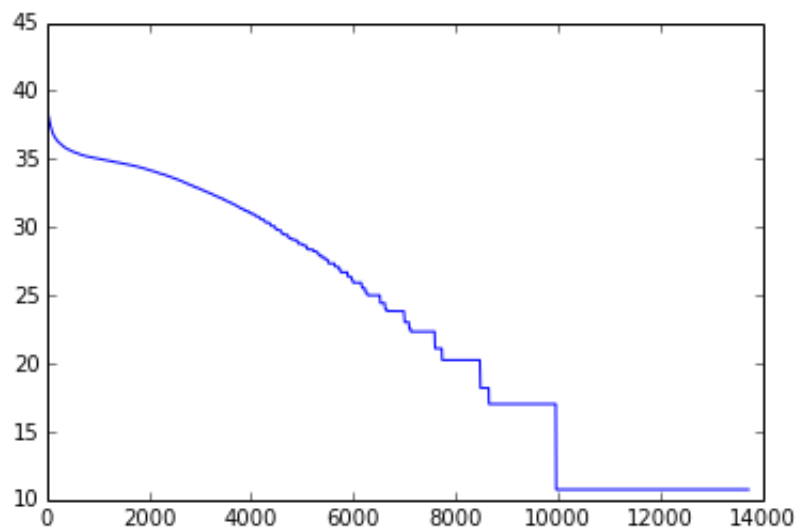
R1 = collection frequency ranking

R2 = tf-idf-weighted ranking

Rank	1	2	3	4	5	6	7	8	9
R1	rate	think	year	will	market	growth	inflat	price	percent
R2	panel	katrina	graph	fedex	wal	mart	mbs	mfp	euro



## Ranking of All FOMC Stems



# Vector Space Model

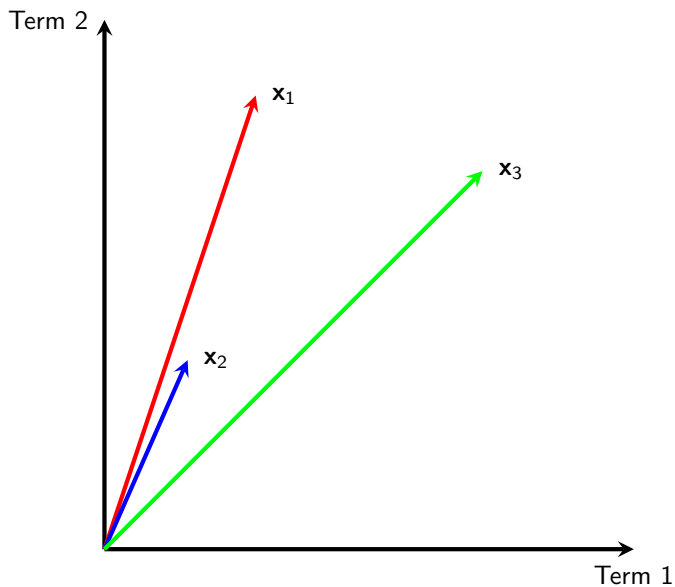
One can also view rows of document-term matrix as vectors lying in a  $V$ -dimensional space.

Tf-idf weighting usually used, but not necessary.

The question of interest is how to measure the similarity of two documents in the vector space.

Initial instinct might be to use Euclidean distance  $\sqrt{\sum_v (x_{i,v} - x_{j,v})^2}$ .

# Three Documents



# Problem with Euclidean Distance

Semantically speaking, documents 1 and 2 are very close, and document 3 is an outlier.

But the Euclidean distance between 1 and 2 is high due to differences in document length.

What we really care about is whether vectors point in same direction.

# Cosine Similarity

Define the cosine similarity between documents  $i$  and  $j$  as

$$CS(i, j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|}$$

1. Since document vectors have no negative elements  $CS(i, j) \in [0, 1]$ .
2.  $\mathbf{x}_i / \|\mathbf{x}_i\|$  is unit-length, correction for different distances.

# Application

An important theoretical concept in industrial organization is location on a product space.

Industry classification measures are quite crude proxies of this.

Hoberg and Phillips (2010) take product descriptions from 49,408 10-K filings and use the vector space model (with bit vectors defined by dictionaries) to compute similarity between firms.

Data available from <http://alex2.umd.edu/industrydata/>.

# Towards Machine Learning

Dictionary methods focus on variation across observations along a limited number of dimensions and ignore the rest.

Ideally we would use variation across *all* dimensions to describe documents.

This obviously provides a richer description of the data, but a deeper point relevant for many high-dimensional datasets is that economic theory does not tell us which dimensions are important.

At the same time, incorporating thousands of independent dimensions of variation in empirical work is difficult.

(Unsupervised) machine learning approaches exploit all dimensions of variation to estimate a lower-dimensional set of types for each documents.

# Unsupervised Learning for Text

The implicit assumption of dictionary methods is that the set of words in the dictionary map back into an underlying theme of interest.

For example we might have that

$\mathcal{D} = \{\text{school, university, college, teacher, professor}\} \rightarrow \text{education}.$

Latent variable models formalize the idea that documents are formed by hidden variables that generate correlations among observed words.

In a natural language context, these variables can be thought of as topics; other applications will have other interpretations.



# Information Retrieval

Latent variable representations can more accurately identify document similarity.

The problem of *synonymy* is that several different words can be associated with the same topic. Cosine similarity between following documents?

school	university	college	teacher	professor
0	5	5	0	2
school	university	college	teacher	professor
10	0	0	4	0

The problem of *polysemy* is that the same word can have multiple meanings. Cosine similarity between following documents?

tank	seal	frog	animal	navy	war
10	10	3	2	0	0
tank	seal	frog	animal	navy	war
10	10	0	0	4	3

If we correctly map words into topics, comparisons become more accurate.

# Mixture versus Mixed-Membership Models

An important distinction in modeling documents is whether they are associated with one or more latent variables.

In traditional cluster analysis, and in mixture models, observations are represented as coming from a single latent category.

In fact, we might imagine that documents cover more than one topic: monetary policy speeches discuss inflation and growth.

Models that associated observations with more than one latent variable are called *mixed-membership* models. Also relevant outside of text mining: in models of group formation, agents can be associated with different latent communities (sports team, workplace, church, etc).

# Latent Semantic Analysis

One of the first mixed-membership models in text mining was the Latent Semantic Analysis/Indexing model of Deerwester et. al. (1990).

A linear algebra rather than probabilistic approach that applies a singular value decomposition to document-term matrix.

Closely related to classical principal components analysis.

Examples in economics: Boukus and Rosenberg (2006); Hendry and Madeley (2010); Acosta (2014); Waldinger et. al. (2018).

# Singular Value Decomposition

The document-term matrix  $\mathbf{X}$  is not square, but we can decompose it using a generalization of the eigenvector decomposition called the *singular value decomposition*.

## Proposition

*The document-term matrix can be written  $\mathbf{X} = \mathbf{A}\mathbf{\Sigma}\mathbf{B}^T$  where  $\mathbf{A}$  is a  $D \times D$  orthogonal matrix,  $\mathbf{B}$  is a  $V \times V$  orthogonal matrix, and  $\mathbf{\Sigma}$  is a  $D \times V$  matrix where  $\Sigma_{ij} = \sigma_i$  with  $\sigma_i \geq \sigma_{i+1}$  and  $\Sigma_{ij} = 0$  for all  $i \neq j$ .*

# Approximating the Document-Term Matrix

We can obtain a rank  $k$  approximation of the document-term matrix  $\mathbf{X}_k$  by constructing  $\mathbf{X}_k = \mathbf{A}\mathbf{\Sigma}_k\mathbf{B}^T$ , where  $\mathbf{\Sigma}_k$  is the diagonal matrix formed by replacing  $\Sigma_{ii} = 0$  for  $i > k$ .

The idea is to keep the “content” dimensions that explain common variation across terms and documents and drop “noise” dimensions that represent idiosyncratic variation.

Often  $k$  is selected to explain a fixed portion  $p$  of variance in the data. In this case  $k$  is the smallest value that satisfies  $\sum_{i=1}^k \sigma_i^2 / \sum_i \sigma_i^2 \geq p$ .

We can then perform the same operations on  $\mathbf{X}_k$  as on  $\mathbf{X}$ , e.g. cosine similarity.

## Example

Suppose the document-term matrix is given by

$$\mathbf{X} = \begin{array}{ccccc} & \text{car} & \text{automobile} & \text{ship} & \text{boat} \\ \begin{array}{l} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{array} & \left[ \begin{array}{cccc} 10 & 0 & 1 & 0 \\ 5 & 5 & 1 & 1 \\ 0 & 14 & 0 & 0 \\ 0 & 2 & 10 & 5 \\ 1 & 0 & 20 & 21 \\ 0 & 0 & 2 & 7 \end{array} \right] \end{array}$$

# Matrix of Cosine Similarities

$$\begin{array}{c} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{array} \begin{bmatrix} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0.70 & 1 & \cdot & \cdot & \cdot & \cdot \\ 0.00 & 0.69 & 1 & \cdot & \cdot & \cdot \\ 0.08 & 0.30 & 0.17 & 1 & \cdot & \cdot \\ 0.10 & 0.21 & 0.00 & 0.92 & 1 & \cdot \\ 0.02 & 0.17 & 0.00 & 0.66 & 0.88 & 1 \end{bmatrix}$$

# SVD

The singular values are (31.61, 15.14, 10.90, 5.03).

$$\mathbf{A} = \begin{bmatrix} 0.0381 & 0.1435 & -0.8931 & -0.02301 & 0.3765 & 0.1947 \\ 0.0586 & 0.3888 & -0.3392 & 0.0856 & -0.7868 & -0.3222 \\ 0.0168 & 0.9000 & 0.2848 & 0.0808 & 0.3173 & 0.0359 \\ 0.3367 & 0.1047 & 0.0631 & -0.7069 & -0.2542 & 0.5542 \\ 0.9169 & -0.0792 & 0.0215 & 0.1021 & 0.1688 & -0.3368 \\ 0.2014 & -0.0298 & 0.0404 & 0.6894 & -0.2126 & 0.6605 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0.0503 & 0.2178 & -0.9728 & 0.0595 \\ 0.0380 & 0.9739 & 0.2218 & 0.0291 \\ 0.7024 & -0.0043 & -0.0081 & -0.7116 \\ 0.7088 & -0.0634 & 0.0653 & 0.6994 \end{bmatrix}$$



# Rank-2 Approximation

$$\mathbf{x}_2 = \begin{matrix} & \text{car} & \text{automobile} & \text{ship} & \text{boat} \\ \begin{matrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{matrix} & \left[ \begin{array}{cccc} 0.5343 & 2.1632 & 0.8378 & 0.7169 \\ 1.3765 & 5.8077 & 1.2765 & 0.9399 \\ 2.9969 & 13.2992 & 0.3153 & 0.4877 \\ 0.8817 & 1.9509 & 7.4715 & 7.4456 \\ 1.1978 & 0.0670 & 20.3682 & 20.6246 \\ 0.2219 & 0.1988 & 4.4748 & 4.5423 \end{array} \right] \end{matrix}$$

# Matrix of Cosine Similarities

$$\begin{array}{c} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{array} \begin{bmatrix} d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0.97 & 1 & \cdot & \cdot & \cdot & \cdot \\ 0.91 & 0.97 & 1 & \cdot & \cdot & \cdot \\ 0.60 & 0.43 & 0.23 & 1 & \cdot & \cdot \\ 0.45 & 0.26 & 0.05 & 0.98 & 1 & \cdot \\ 0.47 & 0.29 & 0.07 & 0.98 & 0.99 & 1 \end{bmatrix}$$

# Application: Transparency

How transparent should a public organization be?

Benefit of transparency: accountability.

Costs of transparency:

1. Direct costs
2. Privacy
3. Security
4. Worse behavior → “chilling effect”

# Transparency and Monetary Policy

**Mario Draghi (2013):** “It would be wise to have a richer communication about the rationale behind the decisions that the governing council takes.”

**Table:** Disclosure Policies as of 2014

	Fed	BoE	ECB
Minutes?	✓	✓	X
Transcripts?	✓	X	X

# Natural Experiment

FOMC meetings were recorded and transcribed from at least the mid-1970's in order to assist with the preparation of the minutes.

Committee members unaware that transcripts were stored prior to October 1993.

Greenspan then acknowledged the transcripts' existence to the Senate Banking Committee, and the Fed agreed:

1. To begin publishing them with a five-year lag.
2. To publish the back data.

"All the News  
That's Fit to Print"

# The New York Times

VOL. CLXIII . . . No. 56,420

© 2014 The New York Times

SATURDAY, FEBRUARY 22, 2014

## ***Fed Misread Fiscal Crisis, Records Show***

***After Caution in 2008,  
Series of Bold Steps***

By BINYAMIN APPELBAUM

WASHINGTON — On the morning after Lehman Brothers filed for bankruptcy in 2008, most Federal Reserve officials still believed that the American economy would keep growing despite the metastasizing financial crisis.

The Fed's policy-making committee voted unanimously against bolstering the economy by cutting interest rates, and several officials praised what they described as the decision to let Lehman fail, saying it would help to restore a sense of accountability on Wall Street.

James Bullard, president of the Federal Reserve Bank of St. Louis, urged his colleagues "to wait for some time to assess the impact of the Lehman bankruptcy filing, if any, on the national economy," according to transcripts of the Fed's 2008 meetings that it published on Friday.

## **DETROIT OUTLINES MAP TO SOLVENCY, STRESSING REPAIR**

**WAY OUT OF BANKRUPTCY**

**Balancing Act Worries  
Banks and Angers  
Retirees in City**

By MONICA DAVEY  
and MARY WILLIAMS WALSH

DETROIT — Seven months after this city entered bankruptcy, its leaders on Friday presented a federal judge with the first official road map to Detroit's future — documents designed to show how it aims to settle its \$18 billion debt to creditors and make itself livable again.

But the proposal is less a vision for a brand-new city than a repair estimate for the old one. It is a document designed by lawyers and bankruptcy experts to find ways to pay off more than 100,000 creditors and then budget money over a period of years to create a

## ***Deal Signed in Ukraine, but Shows Strain***



## Greenspan's View on Transparency

**“A considerable amount of free discussion and probing questioning by the participants of each other and of key FOMC staff members takes place.** In the wide-ranging debate, new ideas are often tested, many of which are rejected ... **The prevailing views of many participants change as evidence and insights emerge.** This process has proven to be a very effective procedure for gaining a consensus ... It could not function effectively if participants had to be concerned that their half-thought-through, but nonetheless potentially valuable, notions would soon be made public. **I fear in such a situation the public record would be a sterile set of bland pronouncements scarcely capturing the necessary debates which are required of monetary policymaking.”**

# Measuring Disagreement

Acosta (2014) uses LSA to measure disagreement before and after transparency.

For each member  $i$  in each meeting  $t$ , let  $\vec{d}_{it}$  be member  $i$ 's words.

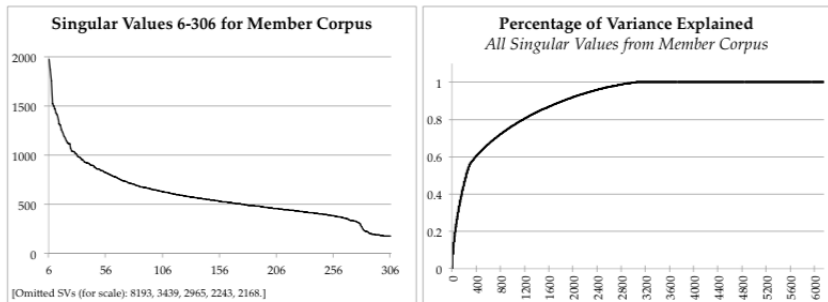
Let  $\vec{d}_{-i,t} = \sum_j \vec{d}_{jt} - \vec{d}_{it}$  be all other members' words.

Quantity of interest is the similarity between  $\vec{d}_{it}$  and  $\vec{d}_{-i,t}$ .

Total set of documents— $\vec{d}_{it}$  and  $\vec{d}_{-i,t}$  for all meetings and speakers—is 6,152.



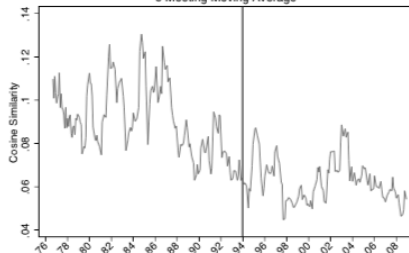
# Singular Values



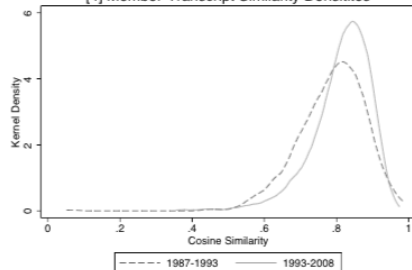
**Figure 11:** The left hand side shows the 6<sup>th</sup> through 306<sup>th</sup> singular values (the elements  $\sigma_i \in \Sigma$  from the SVD) from the member corpus. The right hand side graph show percentage of the variance explained by all 6152 singular values for the member corpus.

# Results

[3] Std. Dev. of Member-Transcript Similarities  
6 Meeting Moving Average



[4] Member-Transcript Similarity Densities



--- 1987-1993    — 1993-2008


# Probabilistic Models


LSA is an important development in machine learning approaches to text, but has some important limitations:


1. SVD is a linear algebra approach to dimensionality reduction, no underlying probability model.
2. The statistical foundations that do exist for SVD are not appropriate for text.
3. Difficult to interpret the components.
4. Difficult to extend LSA to incorporate additional dependencies of interest.


Explicit probability models for text address all of these.

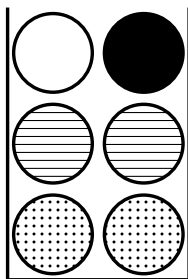
# Topics as Urns

 = wage

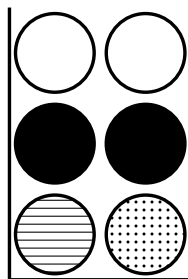
 = price

 = employ

 = increase

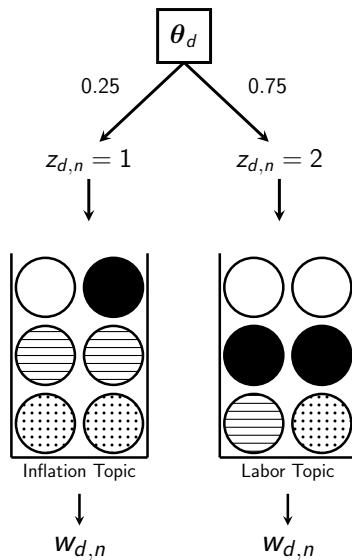


"Inflation" Topic



"Labor" Topic

# Mixed-Membership Model for Document



# Latent Dirichlet Allocation

The preceding model of documents has a huge number of parameters, so maximum likelihood estimation risks overfitting.

One solution is to adopt a Bayesian approach; the preceding model with a prior distribution on the document-specific mixing probabilities is called Latent Dirichlet Allocation (Blei, Ng, and Jordan 2003).

LDA is widely used within computer science and, increasingly, social sciences.

LDA forms the basis of many, more complicated mixed-membership models.

# Latent Dirichlet Allocation—Formal Model

1. Draw  $\beta_k$  independently for  $k = 1, \dots, K$  from  $\text{Dirichlet}(\eta)$ .
2. Draw  $\theta_d$  independently for  $d = 1, \dots, D$  from  $\text{Dirichlet}(\alpha)$ .
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  $\eta$  and  $\alpha$ .

## 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

noticed change relationship between core CPI  
chained core CPI suggested maybe something  
going relating substitution bias upper level index  
focused nonmarket component PCE wondered  
something unusual happening core CPI relative  
measures

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

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

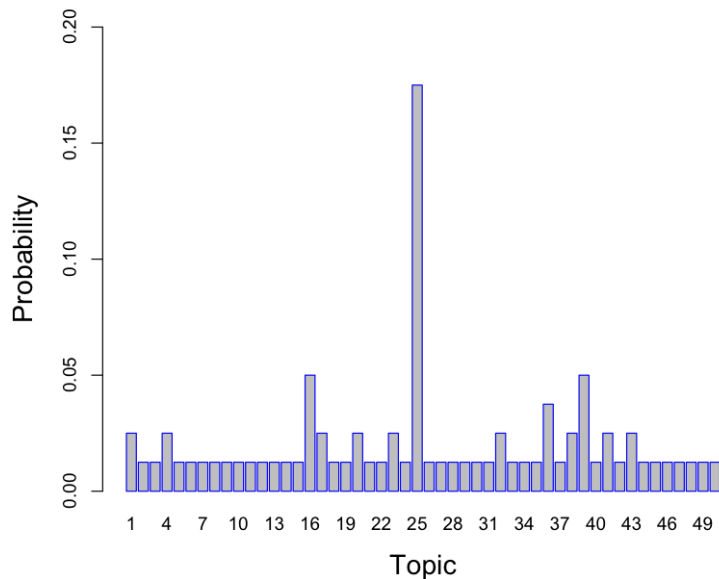
notic chang relationship between core CPI  
chain core CPI suggest mayb someth  
go relat substitut bia upper level index  
focus nonmarket compon PCE wonder  
someth unusu happen core CPI rel  
measur

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

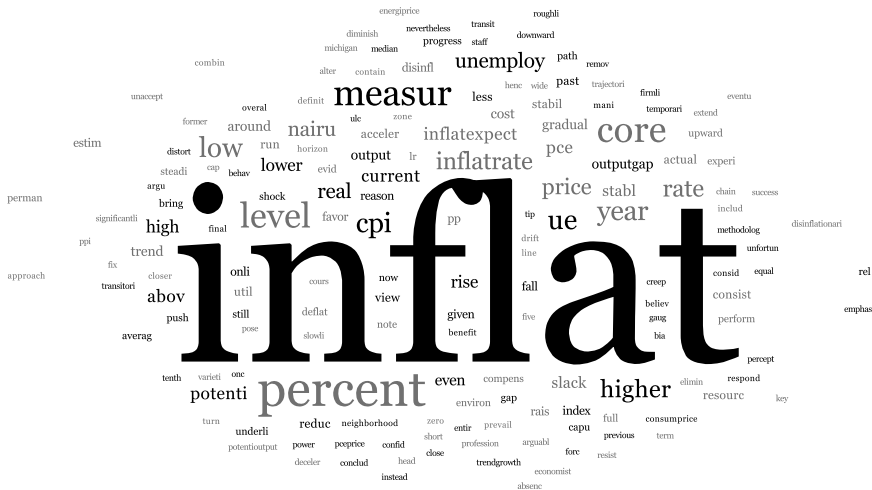
Allocation

	17	39		39	1	25	25
41	25	25	25		36	36	
38	43	25	20	25	25	39	16
23		25	25		25		32
38	16		4		25	25	16
25							

# Distribution of Attention



# Topic 25



# Advantage of Flexibility

'measur' has probability 0.026 in topic 25, and probability 0.021 in topic 11



# Advantage of Flexibility

'measur' has probability 0.026 in topic 25, and probability 0.021 in topic 11.

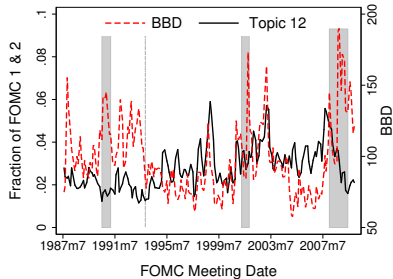
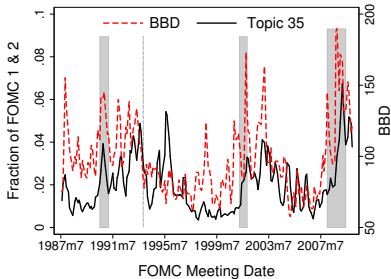
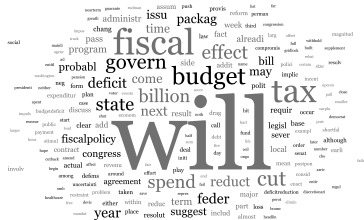
It gets assigned to 25 in this statement consistently due to the presence of other topic 25 words.

In statements containing words on evidence and numbers, it consistently gets assigned to 11.

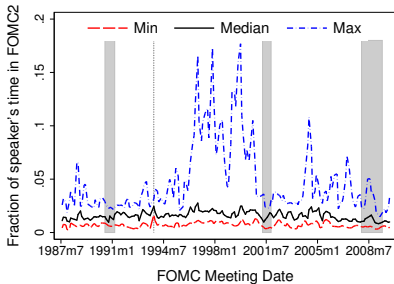
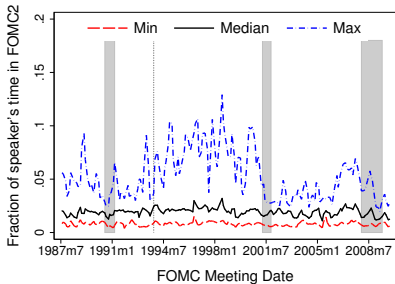
Sampling algorithm can help place words in their appropriate context.



## External Validation—BBD



## Pro-Cyclical Topics





# Applications of LDA

1. Forecasting: Mueller and Rauh (2017); Larsen and Thorsrud (2016).
2. Transparency: Hansen et. al. (2017).
3. Information Processing: Nimark and Pitschner (2017).
4. Basis for structural estimation?

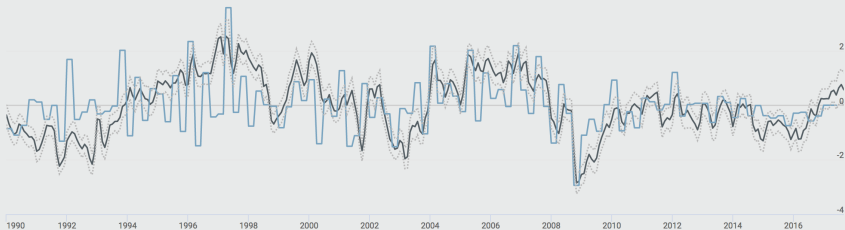
# Financial News Index

## FINANCIAL NEWS INDEX (FNI)

[ABOUT THE INDEX](#)[BACKGROUND](#)[RETRIEVER](#)[CAMP](#)[CONTACT/PRESS](#)

Zoom 6m YTD **All** 1y 5y 10y

From Jan 1, 1990 To Sep 30, 2017



# Intuition for Algorithm

Probability term  $n$  in document  $d$  is assigned to topic  $k$  is increasing in:

1. The number of other terms in document  $d$  that are currently assigned to  $k$ .
2. The number of other occurrences of the term  $w_{d,n}$  in the entire corpus that are currently assigned to  $k$ .

Both mean that terms that regularly co-occur in documents will be grouped together to form topics.

Property 1 means that terms within a document will tend to be grouped together into few topics rather than spread across many separate topics.

# Model Selection

There are three parameters to set to run the Gibbs sampling algorithm: number of topics  $K$  and hyperparameters  $\alpha, \eta$ .

Priors don't receive too much attention in 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).

# Cross Validation

Fit LDA on training data, obtain estimates of  $\hat{\beta}_1, \dots, \hat{\beta}_K$ .

For test data, estimate  $\theta_d$  distributions, or else use uniform distribution.

Compute log-likelihood of held-out data as

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

Higher values indicate better goodness-of-fit.



# Information Criteria

Information criteria trade off goodness-of-fit with model complexity.

There are various forms: AIC, BIC, DIC, etc.

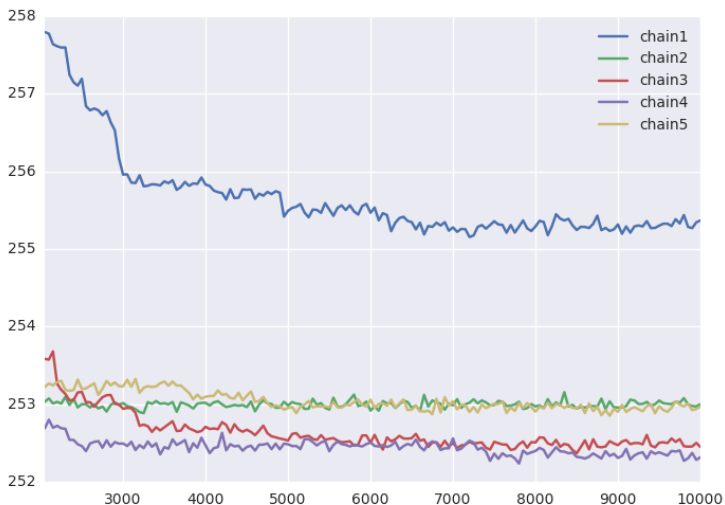
Erosheva et. al. (2007) compare several in the context of an LDA-like model, and find that AICM is optimal.

Let  $\mu_\ell = \frac{1}{S} \sum_s \ell(\mathbf{w} \mid \hat{\Theta}^s)$  be the average value of the log-likelihood across  $S$  draws of a Markov chain and

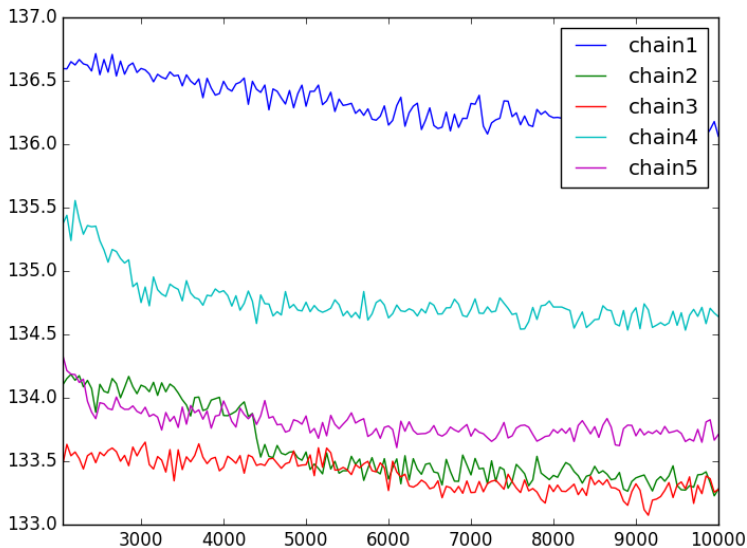
Let  $\sigma_\ell^2 = \frac{1}{S} \sum_s \left( \ell(\mathbf{w} \mid \hat{\Theta}^s) - \mu_\ell \right)^2$  be the variance.

The AICM is  $2(\mu_\ell - \sigma_\ell^2)$ .

## Goodness-of-Fit with $K = 2$



## Goodness-of-Fit with $K = 10$



# Formalizing Interpretability

Chang et. al. (2009) propose an objective way of determining whether topics are interpretable.

Two tests:

1. *Word intrusion*. Form set of words consisting 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 to it + randomly drawn other topic. Ask to identify inserted topic.

Estimate LDA and other topic models on NYT and Wikipedia articles for  $K = 50, 100, 150$ .

# Results

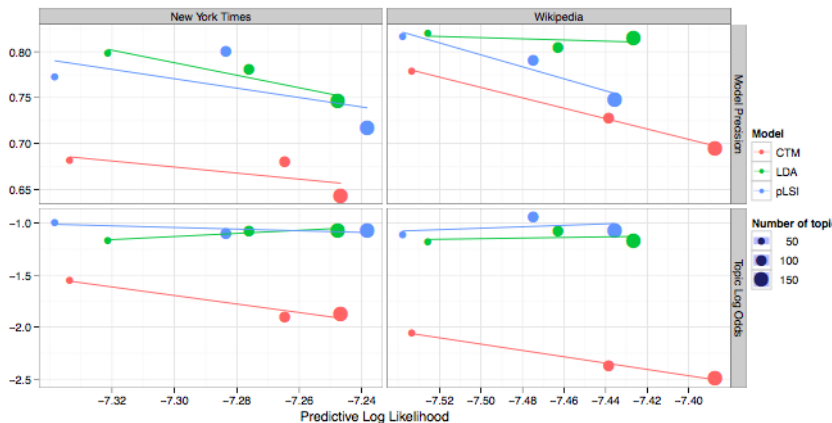


Figure 5: A scatter plot of model precision (top row) and topic log odds (bottom row) vs. predictive log likelihood. Each point is colored by model and sized according to the number of topics used to fit the model. Each model is accompanied by a regression line. Increasing likelihood does not increase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

# Takeaway

Topics seem objectively interpretable in many contexts.

Tradeoff between goodness-of-fit and interpretability, which is generally more important in social science.

Active area of research assessing LDA models in terms of topic coherence.

Newman et. al. (2010) propose a method based on mutual pointwise information between top words in topics as computed via co-occurrence in Wikipedia.