Theoretical Foundations of Machine Learning for Economists: Lecture 4

Sonan Memon
Lecturer, Institute of Business Administration, Karachi

15th April 2021

Computational Linguistics: Text as Data

i. Lots of human interaction, communication, and culture is recorded as digital text due to use of internet, social media etc.

ii. Recent years have seen an explosion of empirical economics research using text as data.

iii. Economic Applications:
   - In finance, text from financial news, social media, and company filings is used to predict asset prices.
   - In macroeconomics, text is used to forecast variation in inflation and unemployment, and estimate the effects of policy uncertainty.
   - In political economy, text from politicians; speeches is used to study the dynamics of political agendas and debate.

Applications

i. In media economics, text from news and social media is used to study the drivers and effects of political slant.

ii. Applications of **Topic Modeling** in particular include:
    - Discovery of overlapping communities in social networks Airoldi et al., 2008.
    - Dynamic topic models Blei and Lafferty, 2006, which have been used to analyze the progression of Science articles from 1880 to 2000.
    - Collaborative topic models Wang and Blei, 2011, which are used for content recommendation at the New York Times.
    - Population analysis of 2 billion genetic measurements Gopalan et al., 2016.

Applications of Computational Linguistics In General

i. Translating and summarizing text.
ii. Retrieving text that relates to a specific topic.
iii. Analyzing text or spoken language for context, sentiment or other affective qualities.
iv. Answering questions, including those that require inference and descriptive or discursive answers.
v. Hiring and Recruitment.
vi. Creating chatbots capable of passing the Turing Test.

Readings for Lecture 4

i. Gentzkow, M., Kelly, B. T., and Taddy, M. (2019). Text as data. Journal of Economic Literature.

ii. I Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, 3(Jan):993–1022.

iii. Hansen, S., & McMahon, M. (2016). Shocking language: Understanding the macroeconomic effects of central bank communication. Journal of International Economics, 99, S114-S133.

iv. Larsen, V. H., Thorsrud, L. A., & Zhulanova, J. (2021). News-driven inflation expectations and information rigidities. Journal of Monetary Economics, 117, 507-520.

Computational Linguistics: Text as Data

i. The most important way that text differs from the kinds of data: inherently high dimensional.

ii. A sample of thirty-word Twitter messages that use only the one thousand most common words in the English language, for example, has roughly as many dimensions as there are atoms in the universe.

iii. Feature extraction.

iv. Ad hoc dictionary methods versus more sophisticated methods such as LDA.

Motivation

i. One big contribution of machine learning methods to econometrics is that they make new forms of data amenable to quantitative analysis: Text, images, ...

ii. We discuss some methods for turning text into data. Key steps:
   - Converting corpus of documents into numerical arrays.
   - Extracting some compact representation of each document.
   - Using this representation for further analysis.

iii. Two approaches for step 2:
   - Supervised: e.g Lasso prediction of outcomes based on word counts.
   - Unsupervised: e.g., topic models, "latent dirichlet allocation (LDA)".

Motivation

i. Many sources of digital text for social scientists: political news, social media, political speeches, financial news, company filings, advertisements, product reviews etc.

ii. Very high dimensional: For a document of N words from a vocabulary of size V, there are V^N possibilities.

iii. Three steps:
   - Represent text as numerical array $\boldsymbol{w}$. (Drop punctuation and rare words, count words or phrases.)
   - Map array to an estimate of a latent variable. (Predicted outcome or classification to topics.)
   - Use the resulting estimates for further analysis. (Causal or other.)

Overview

i. To make text (or other high-dimensional discrete data) amenable to statistical analysis, we need to generate low-dimensional summaries.

ii. Supervised approach:
   - Regress observed outcome Y on high-dimensional description $\boldsymbol{w}$. Use appropriate regularization and tuning.
   - Impute predicted $\widehat{Y}$ for new realizations $\boldsymbol{w}$.

iii. Unsupervised approach:
   - Assume texts are generated from distributions corresponding to topics.
   - Impute unobserved topics.

iv. Topic models are a special case of hierarchical models. These are useful in many settings.

Notation

i. Basic unit of vocabulary, indexed by $v \in \{1, 2, 3, ..., V\}$, where there are V possible words in the vocabulary. vth word in vocabulary is represented by unit vector w such that $w^v = 1$ and $w^u = 0, \forall u \neq v$.

ii. Document is a sequence of n words, $\boldsymbol{w} = (w_1, w_2, ..., w_N)$.

iii. Corpus is collection of M documents: $\boldsymbol{D} = (\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_M)$.

Representing Text as Data

i. Language is very complex. Context, grammar, ...

ii. Quantitative text analysis discards most of this information.

iii. **Data preparation steps:**

1. Divide corpus D into documents j, such as
   - the news of a day, individual news articles,
   - all the speeches of a politician, single speeches, ...
2. Pre-process documents:
   - Remove punctuation and tags,
   - remove very common words ("the, a," "and, or," "to be," ...),
   - remove very rare words (occurring less than k times),
   - stem words, replacing them by their root.

Representing Text as Data

3. Next, convert resulting documents into numerical arrays $\boldsymbol{w}$.
   - Simplest version: Bag of words. Ignore sequence. w_v is the count of word v, for every v in the vocabulary.
   - Somewhat more complex: $w_{vv'}$ is the count of ordered occurrence of the words v, v' for every such “bigram”.
   - Can extend this to N-grams, i.e., sequences of N words. But $N > 2$ tends to be too unwieldy in practice.

Dimension Reduction

i. Goal: Represent high-dimensional $\boldsymbol{w}$ by some low-dimensional summary.

ii. 4 alternative approaches:
   - Dictonary-based: Just define a mapping $g(\boldsymbol{w})$.
   - Predict observed outcome Y based on w. Use predicted $\widehat{Y}$ as summary: Supervised learning.
   - Predict $\boldsymbol{w}$ based on unobserved latent θ. Topic models. Impute $\widehat{\theta}$ and use as summary: Unsupervised learning.

Text Regression

i. Suppose we observe outcomes Y for a subset of documents. We want to estimate $\mathbb{E}[Y|\boldsymbol{w}]$ for this outcome so we can impute $\widehat{Y} = \mathbb{E}[Y|\boldsymbol{w}]$ for new draws $\boldsymbol{w}$.

ii. $\boldsymbol{w}$ is very high dimensional, so we cannot just run OLS, so we used regularization:

$$\hat{\beta} = \underset{\beta}{argmin} \sum_j \left(Y_j - \boldsymbol{w}_j \beta\right)^2 + \lambda \sum_v |w_v|^p$$

$$Y_j = \boldsymbol{w_j}\beta.$$

iii. $p = 1$ leads to LASSO (Lec 1) and $p = 2$ leads to Ridge Regression.

iv. λ is chosen using cross-validation.

Nonlinear Regression

i. For binary outcomes, we may use multinomial logit with regularization.

ii. Neural Networks could also be used as predictors. Recurrent Neural Networks are used in natural language processing and speech recognition for instance.

Generative Language Models

i. Generative models give a probability distribution over documents.

ii. Let us start with a very simple model. Unigram model: The words of every document are drawn independently from a single multinomial distribution.

iii. The probability of a document is: $p(\boldsymbol{w}) = \prod_n p(w_n)$.

iv. The vector of probabilities $\tau = (p(w_{\alpha_1}), p(w_{\alpha_2}), ..., p(w_{\alpha_v}))$ sum to 1 and is a point in the simplex spanned by the v words. $v \leq V$ and $\alpha_i \in \{1, 2, ..., V\}, \forall i$.

v. In the unigram model, each document is generated based on the same vector of words τ.

Mixture of Unigrams

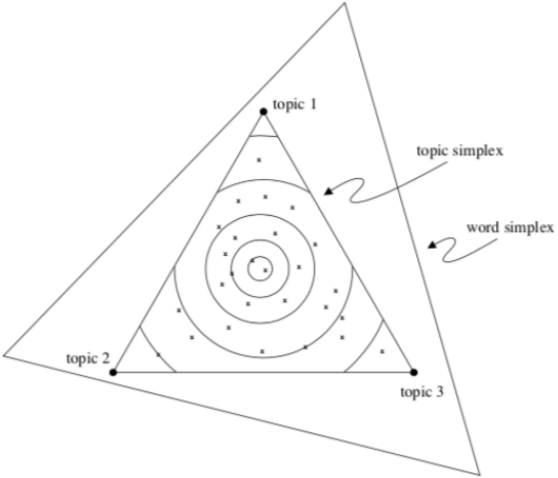
i. A more complicated model is the mixture of unigrams model, which assumes that each document has an unobserved topic z.

ii. Conditional on z, words are sampled from a multinomial distribution with parameter vector θ_z.

iii. **Mixture of Unigrams:** The probability of a document is $p(\boldsymbol{w}) = \sum_z p(z) \prod_n p(w_n|z)$, where $p(w_n|z) = \beta_{z,w_n}$.

iv. The vector of probabilities β_z is a point in the simplex, spanned by the V words in the vocabulary.

Word and Topic Simplex



Graphical Representation of Hierarchical Models

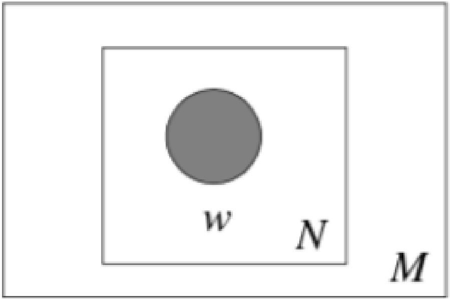
i. The mixture of unigrams model is a simple case of a hierarchical model.

ii. Hierarchical models are defined by a sequence of conditional distributions. Not all variables in these models need to be observed.

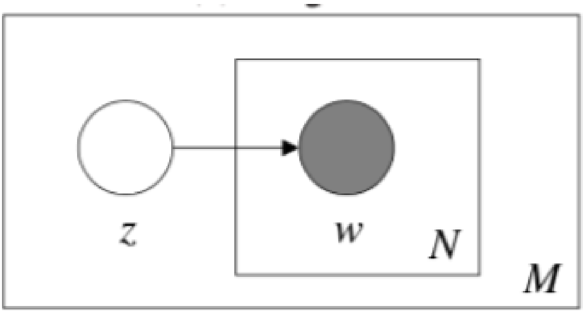
iii. Hierarchical models are often represented graphically:

   - Observed variables are shaded circles, unobserved variables are empty circles.
   - Arrows represent conditional distributions.
   - Boxes are "plates" representing replicates.
   - Replicates are conditionally independent repeated draws.
   - In the upcoming slides, the outer plate represents documents.
   - The inner plate represents the repeated choice of words within a document.

Unigram



Mixture of Unigrams



Latent Dirichlet Allocation

i. LDA is a very popular generative, language model and is a generalization of the mixture of unigrams model Blei et al. (2003): 10,000 + citations in around 10 years!

ii. For modeling text corpora and other collections of discrete data.

iii. Goal: Find short descriptions of the members of a collection.

iv. Applications of **LDA** include:

- Discovery of overlapping communities in social networks Airoldi et al., 2008.
- Dynamic topic models Blei and Lafferty, 2006, which have been used to analyze the progression of Science articles from 1880 to 2000.
- Analysis of Central Bank Communication.
- Policy Uncertainty Baker et al (2016).

Exchangeability and De Fenetti's Theorem

i. **Exchangeability:** $p(z_1, z_2, ...z_n) = p(z_{\pi_1}, z_{\pi_2}, ..., z_{\pi_n})$.

ii. De Finetti's representation theorem states that the joint distribution of an infinitely exchangeable sequence of random variables is as if a random parameter were drawn from some distribution and then the random variables in question were independent and identically distributed, conditioned on that parameter.

Latent Dirichlet Model

i. Condition on document lengths N.

ii. Draw θ:

$$\theta \sim Dirichlet(\alpha).$$

iii. Given θ, for each of the N words in the document, draw topic

$$z_n \sim Multinomial(\theta).$$

iv. Given θ and z_n, draw a word w_n from the conditional distribution

$$w_n \sim p(w_n | z, \beta)$$

where β is $k \times V$ matrix with entries $p(w^j = 1 | z^i = 1)$.

Dirichlet Distribution

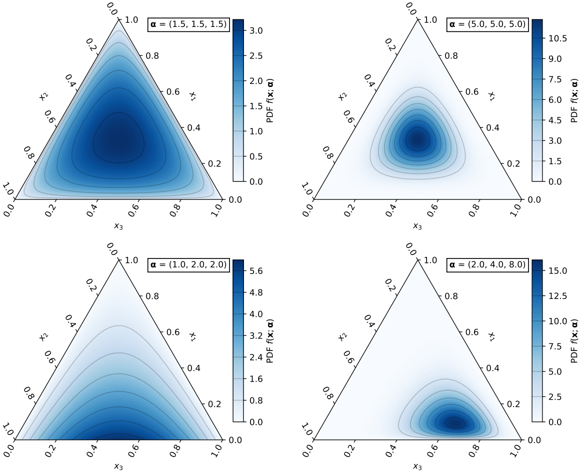
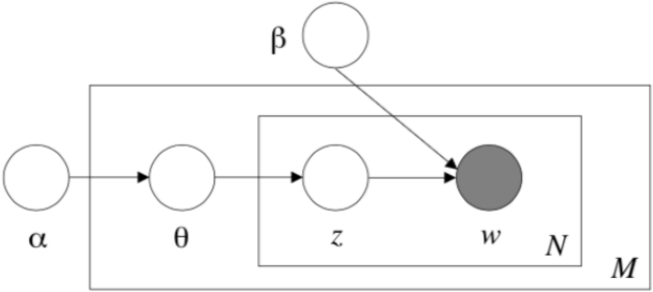
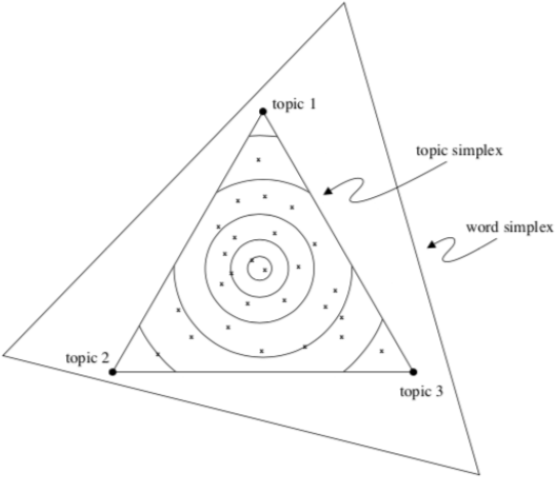


Figure: Dirichlet Probability Density

Graphical Representation of LDA



Word and Topic Simplex



Likelihood

i. Dirichlet distribution of topic mixtures:
$p(\theta|\alpha) = const. \prod_{j=1}^{k} \theta_j^{\alpha_j - 1}$.

ii. Joint distribution of topic mixture θ, a set of N topics $\boldsymbol{z}$ and a set of N words $\boldsymbol{w}$:
$p(\theta, \boldsymbol{z}, \boldsymbol{w}) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta)$.

iii. What is the probability of a given document $\boldsymbol{w}$ and probability of corpus $\boldsymbol{D}$?

Key Probabilities

i. Probability of document $\boldsymbol{w}$:

$$p(\boldsymbol{w}|\alpha,\beta) = \int p(\theta|\alpha)\left(\prod_n \sum_{z_n} p(z_n|\theta)p(w_n|z_n,\beta)\right) d\theta$$

ii. Probability of corpus $\boldsymbol{D}$:

$$p(\boldsymbol{D}|\alpha,\beta) =$$
$$\prod_d \left[\int p(\theta_d|\alpha)\left(\prod_n \sum_{z_n} p(z_{dn}|\theta)p(w_{dn}|z_{dn},\beta)\right) d\theta_d\right]$$

Estimation

i. Closed form likelihoods are not available: Blei et al (2003) combine variational inference (maximizing lower bound of likelihood) with expectation maximization algorithm (EM).

ii. Alternative is Markov Chain Monte Carlo, in particular Gibbs Sampling based algorithms.

iii. R packages mentioned later in slides allow for both kinds of estimation, based on "VEM" as well as on Gibbs sampling.

iv. Useful tool is **Stan**: `http://mc-stan.org/rstan/`. General purpose environment for sampling from posteriors for hierarchical models.

VEM Algorithm

i. $p(\theta, \boldsymbol{z} | \boldsymbol{w}, \alpha, \beta) = \frac{p(\theta, \boldsymbol{z}, \boldsymbol{w} | \alpha, \beta)}{p(\boldsymbol{w} | \alpha, \beta)}$ is the posterior and denominator is intractable to compute.

ii. Approximate variational inference is used: consider a family of lower bounds, indexed by a set of variational parameters ϕ, which are chosen to find the tightest possible lower bound Blei et al (2003).

iii. Find approximate empirical Bayes estimates via an alternating variational EM procedure that maximizes a lower bound with respect to ϕ and then, for fixed values of ϕ, maximizes the lower bound with respect to the model parameters α and β.

Gibbs Sampling

i. Sample from conditional distribution of parameters since the joint distribution is intractable to sample from.

ii. Constructs Markov Chain in which samples converge to joint distribution.

iii. In general, suppose we want to sample θ_1, θ_2 from $p(\theta_1, \theta_2)$ but this is intractable so instead we sample from $p(\theta_1|\theta_2)$ and $p(\theta_2|\theta_1)$.

iv. Sample $\theta_1^j \sim p(\theta_1|\theta_2^{j-1})$ and then
Sample $\theta_2^j \sim\sim p(\theta_2|\theta_1^j)$.

v. $(\theta_1, \theta_2) \rightarrow \theta \sim p(\theta_1, \theta_2)$.

vi. See Heinrich, G. (2009) Parameter estimation for text analysis. Technical report, Fraunhofer IGD for detailed treatment of Gibbs Sampling for LDA.

Economic Application 1: Macroeconomic Effects of Central Bank Communication

Hansen and McMahon et al (2016) (Journal of International Economics)

i. Using tools from computational linguistics, they measure the information released by the FOMC on the state of economic conditions, as well as the guidance the FOMC provides about future monetary policy.

ii. Employing the measures in a FAVAR framework, they find that shocks to forward guidance are more important than Fed communication about current economic conditions.

iii. The corpus is the full history of 142 FOMC statements, up till March 2015.

iv. Remove stop words such as “the”, “a” and “and” and stem words, reducing them to a common linguistic root: “economy” and “economic” both become “economi”.

Economic Application 1: Macroeconomic Effects of Central Bank Communication

LDA Methodology Hansen and McMahon et al (2016)

i. Use 15 topic model and LDA algorithm with sentences as unit of analysis.
ii. The topic model estimates K topics, each of which is a distribution $\beta^K \in \Delta^V$ over the V words in vocabulary.
iii. They identify a sentence as being about topic k, if $\phi_{p,k,d} = \frac{n_{p,d,k}}{n_{p,d}} > \alpha$ (some critical fraction).
iv. $\phi_{p,k,d}$ is the fraction of words in sentence p and document d which are about topic k.
v. Use Gibbs Sampling to estimate LDA parameters.

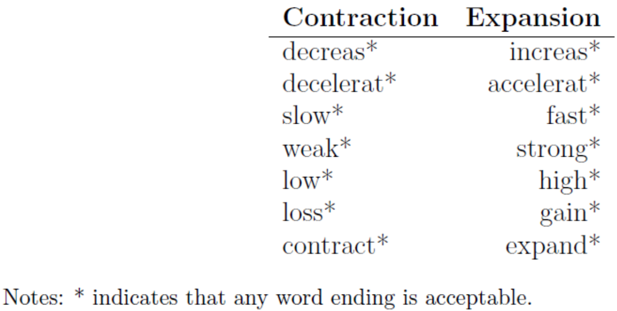
Economic Application 1: Macroeconomic Effects of Central Bank Communication

LDA and Dictionary Methods Hansen and McMahon et al (2016)

i. They combine dictionary methods with LDA topic allocation to get topic wise measures of tone.

ii. $EcSit_t = \frac{n_{Pos,t} - n_{Neg,t}}{Total\ Words_t^{EC}}$ is the balance measure of economic situation.

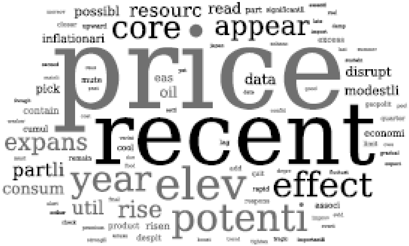
Dictionary Method Based Classification

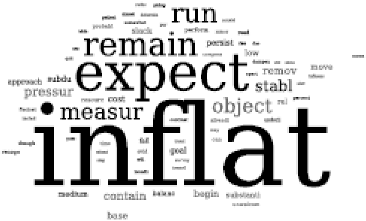


Contraction	Expansion
decreas*	increas*
decelerat*	accelerat*
slow*	fast*
weak*	strong*
low*	high*
loss*	gain*
contract*	expand*

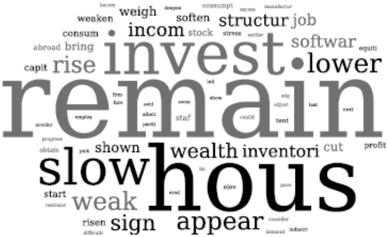
Notes: * indicates that any word ending is acceptable.

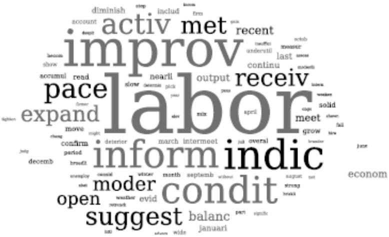
FOMC Topics Regarding Inflation and Prices



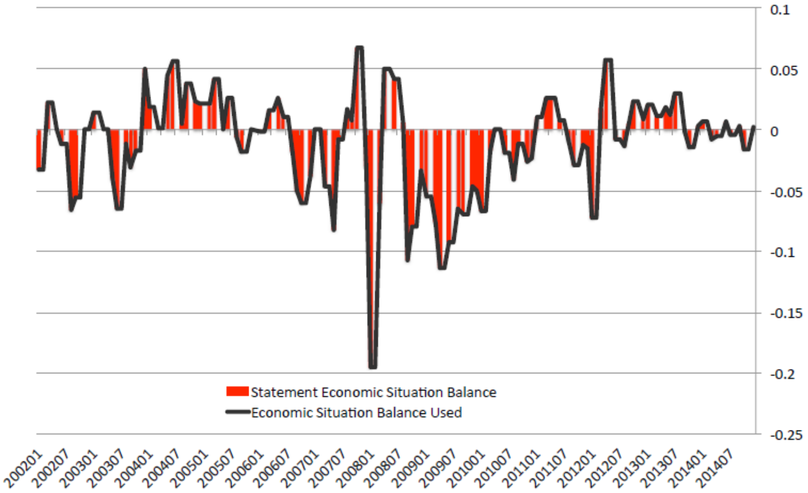


FOMC Topics Covering Demand Side and Labor Market Issues





Economic Situation Index



Economic Application 1: Macroeconomic Effects of Central Bank Communication

Forward Guidance Measurement Hansen and McMahon et al (2016)
Manually identify statements about forward guidance such as "To support continued progress toward maximum employment and price stability, the Committee today reaffirmed its view that a highly accommodative stance of monetary policy will remain appropriate for a considerable time after the asset purchase program ends and the economic recovery strengthens".

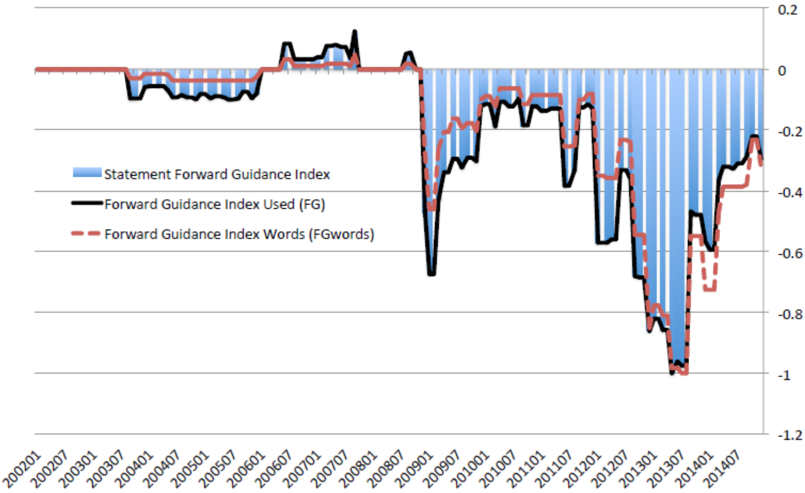
Economic Application 1: Macroeconomic Effects of Central Bank Communication

Forward Guidance Measurement Hansen and McMahon et al (2016)

i. Combine amount, direction and certainty of guidance to get $FG_t = \frac{ShareFG_t \times DirectionFG_t}{Uncertainty_t}$ where direction is measured manually, uncertainty is measured using dictionary methods and share of words about *FG* are also identified manually.

ii. Negative and strong value in next slide indicates that a highly certain, strong, expansionary monetary policy stance about future is being communicated.

Forward Guidance Index



Economic Application 2: News Driven Inflation Expectations

Larsen et al (2021) (Journal of Monetary Economics).

i. Investigate the role played by the media in the expectations formation process of households.
ii. The news media corpus consists of roughly 5 million news articles from Dow Jones Newswires Archive (DJ) including WSJ for 1990 to 2016.
iii. Find that the news topics media report on are good predictors of both inflation and inflation expectations.
iv. The news corpus is cleaned by removing stop-words, conducting stemming etc.

Economic Application 2: News Driven Inflation Expectations

LDA Methodology

i. 80 different topics are extracted.

ii. The topic decomposition is transformed into time series, measuring how much each topic is written about at any given point in time.

iii. LASSO regression is run, in which 20 topics are selected and six are significant predictors of household inflation expectations.

iv. Distribution of words for a given topic is illustrated using word clouds.

v. Successively positive and strong time series values for a topic means that the topic is discussed with increasingly high frequency and positive tone.

Economic Application 2: News Driven Inflation Expectations

Tone Adjusted Topical Time Series

i. They first collapse all the articles for a particular day into one document and then compute using the estimated word distribution for each topic, the topic frequencies for this newly formed document.

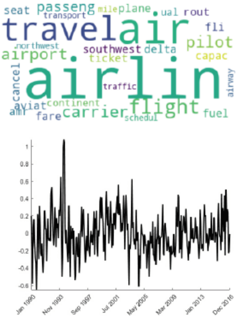
ii. This yields K daily time series. Then, for each day, we find the article that is best explained by each topic, and from that identify the tone of the topic, i.e., whether or not the news is positive or negative.

iii. This is done using an external word list and simple word counts. The word list classifies positive/negative words as defined by the Harvard IV-4 Psychological Dictionary.

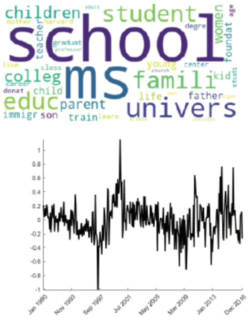
iv. Used Gibbs Sampling Based Algorithm.

Topics in Larsen et al (2021)

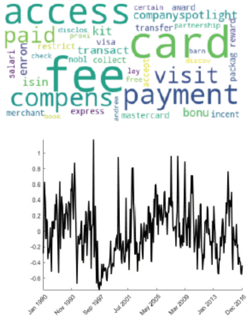
US T75: Aviation



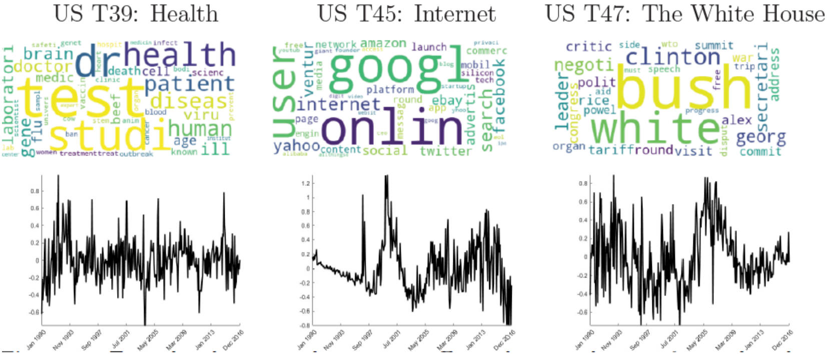
US T2: Education



US T77: Transactions



Topics in Larsen et al (2021)


US T39: Health

US T45: Internet

US T47: The White House

Frontier: Word Emeddings and Natural Language Processing

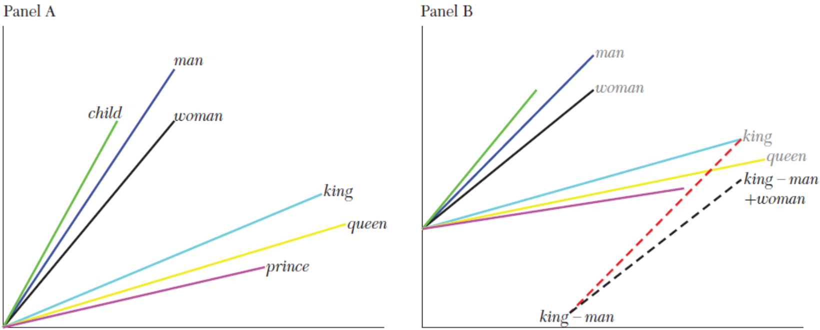
i. Methods discussed so far use token count vectors and abstract from any notion of similarity between words (such as run, runner, jogger) or syntactical richness.

ii. Other methods from computational linguistics exist that capture richer features of text but rarely used in social science.

iii. Rather than just treating documents as language tokens, treat them as ordered sequence of transitions between words e.g represent sentence of length s as matrix $\boldsymbol{S}$ which has dimension $s \times V$, where V is vocabulary length.

iv. Represent words in vector space $\mathbb{R}^K$ in which similar words are colocated. K is dimension of latent representation space.

Word Embedding Example



Implementation in R

i. Stan for hierarchical models.

ii. We can use the *LDA*() function from the `topicmodels` package in R and also the textmining package: `tt`.

iii. The `tidytext` package provides method for extracting the per-topic-per-word probabilities.

iv. See Text Mining with R by Julia Silge and David Robinson (2017) for text analysis using R.

Pseudo Code in R

i. library(topicmodels)

ii. data(“..”)

iii. ap_{lda} <- LDA(AssociatedPress, $k = 2$, control $=$ list(seed $=$ 1234))

iv. library(tidytext)

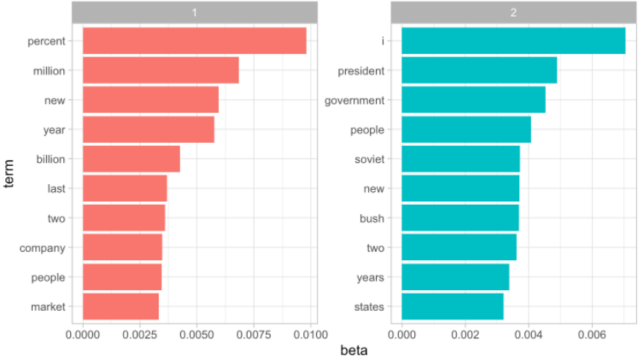
v. ap_{topics} <- tidy(ap_{lda}, matrix $=$ “beta”)

vi. The last line of code above computes the probability for each term/word of being generated from that topic.

Pseudo Code in R

i. library(ggplot2)
ii. library(dplyr)
iii. ap top terms <- ap-topics % > % group-by (topic) % > %
 slice max (beta, n = 10) % > % ungroup() % > %
 arrange(topic, −beta)
iv. ap top terms % > %
 mutate(term = reorder within (term, beta, topic)) %>%
 ggplot(aes(beta, term, fill = factor(topic))) +
 geom col(show.legend = FALSE) +
 facet wrap (˜topic, scales = “free”) +
 scale y reordered().

Top 10 Words by Topic



Thank you