

# MY360/459 Quantitative Text Analysis: Probabilistic Topic Models

Friedrich Geiecke

March 11, 2024

Course website: [lse-my459.github.io](https://lse-my459.github.io)

1. Overview and Fundamentals
2. Descriptive Statistical Methods for Text Analysis
3. Automated Dictionary Methods
4. Machine Learning for Texts
5. Supervised Scaling Models for Texts
6. *Reading Week*
7. Unsupervised Models for Scaling Texts
8. Similarity and Clustering Methods
9. Probabilistic Topic Models
10. Word Embedding Fundamentals
11. Neural Network Based Language Models

# Today

- ▶ Introduction
- ▶ Latent Dirichlet Allocation (LDA)
- ▶ Extensions
- ▶ Model selection and evaluation
- ▶ Implementations in R
- ▶ Alternative topic modelling approaches
- ▶ Coding

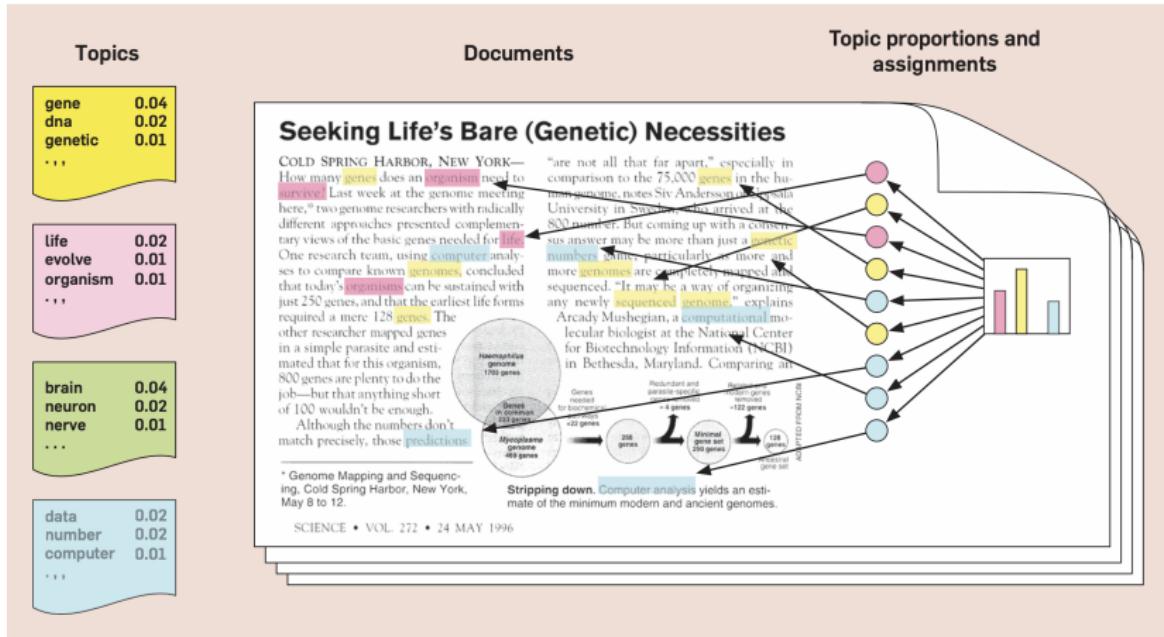
# Outline

- ▶ Introduction
- ▶ Latent Dirichlet Allocation (LDA)
- ▶ Extensions
- ▶ Model selection and evaluation
- ▶ Implementations in R
- ▶ Alternative topic modelling approaches
- ▶ Coding

## Topic models

- ▶ Topic models offer an automated procedure to discover *themes* in an unstructured corpus of texts
- ▶ Can be used to understand and organise large collections of documents according to the discovered themes
- ▶ Require no labelled data – only a set of documents
- ▶ Latent Dirichlet Allocation (LDA) is one fundamental approach (Blei et al., 2003)
- ▶ It is a mixture model: Documents can contain multiple topics; words can belong to multiple topics
- ▶ Alternative approaches e.g.: Clustering of transformer-based embeddings (also see discussion in last week of course)

# Illustration

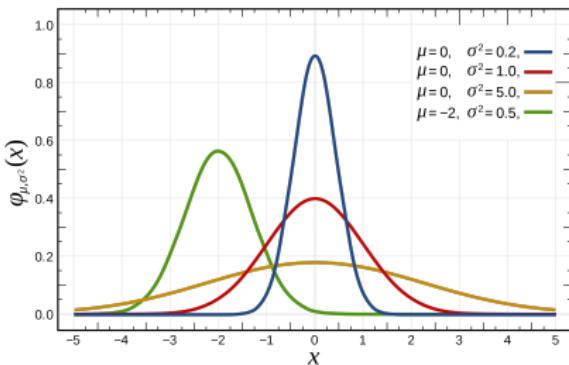


From: Probabilistic topic models, David Blei, 2012

# Outline

- ▶ Introduction
- ▶ Latent Dirichlet Allocation (LDA)
- ▶ Extensions
- ▶ Model selection and evaluation
- ▶ Implementations in R
- ▶ Coding

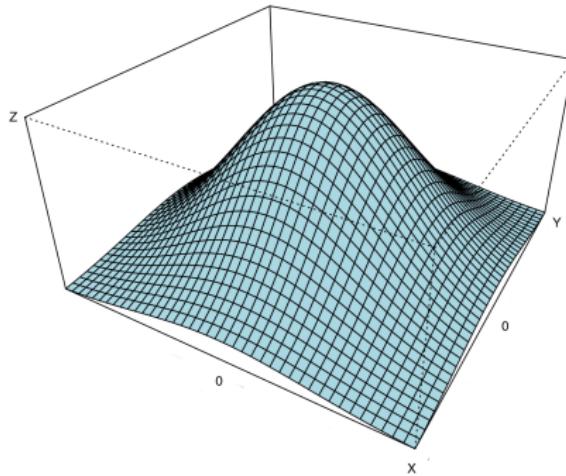
## Review: Univariate probability density function



Source: Wikipedia

- ▶ Different parameter values (e.g.  $\mu$  and  $\sigma$  for the normal distribution) change the distributions' shape
- ▶ The notation " $x \sim N(0, 1)$ " denotes to sample or draw "x" from a standard normal distribution. This draw could e.g. return  $x = -1.124$

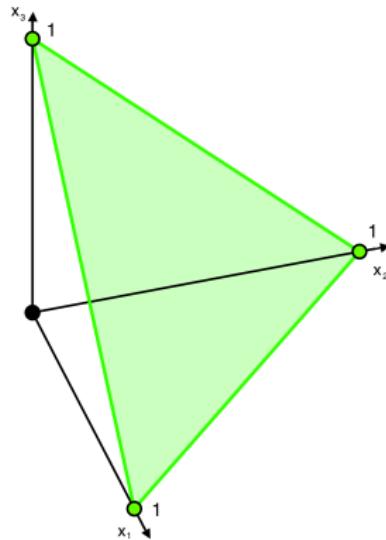
## Review: Multivariate probability density function



Source: Wikipedia

A draw  $a \sim N(\mu, \Sigma)$  from this multivariate normal distribution could e.g. return  $a = (-0.12, 1.2)$

## Review: Graphical intuition of a standard simplex



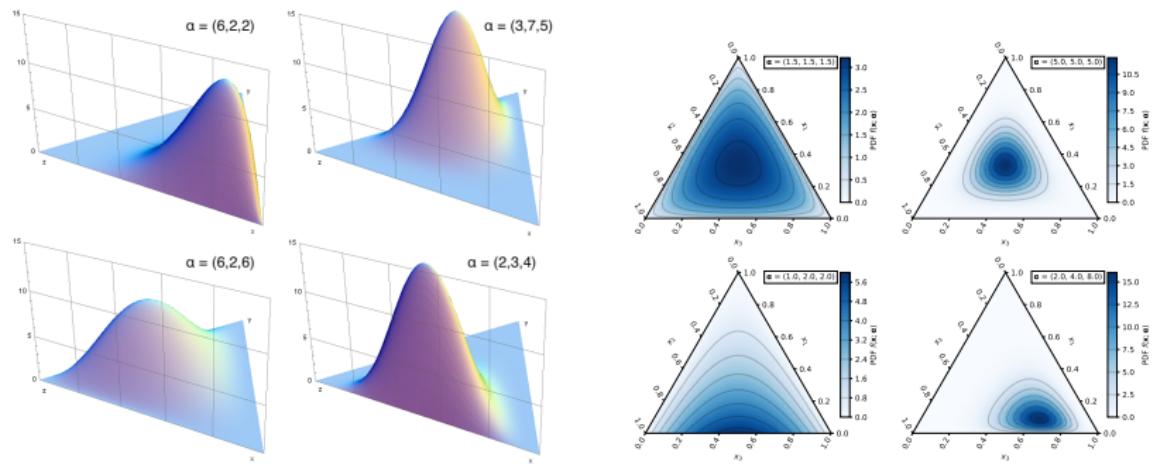
Source: Wikipedia

A point on the triangle is  $x_1, x_2, x_3$ , with  $x_1, x_2, x_3 \in [0, 1]$  and  $x_1 + x_2 + x_3 = 1$ . For example,  $(0.05, 0.8, 0.15)$

Generalises to higher dimensions with  $x_1, \dots, x_K \in [0, 1]$  and  $\sum x_k = 1$

# Key distribution 1: The Dirichlet distribution

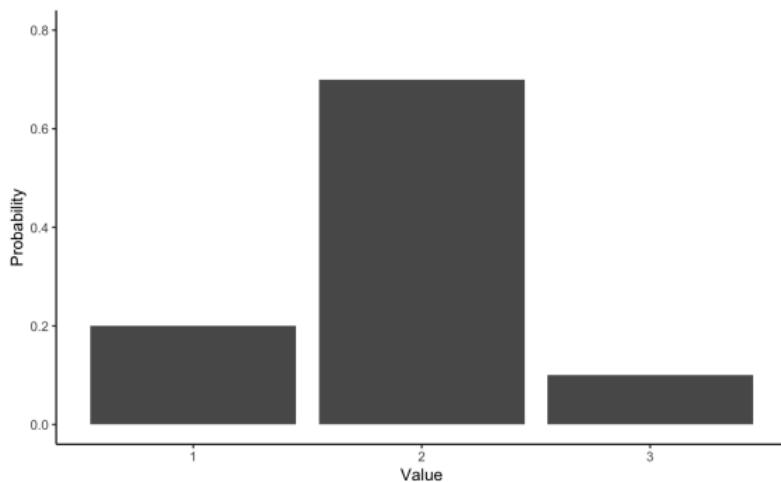
Dirichlet distribution: Probability distribution over a simplex



Source: Wikipedia

- ▶ A draw  $b \sim Dir(\alpha)$  from this distribution could e.g. return  $b = (0.2, 0.7, 0.1)$
- ▶ Hence, we can think of the draw from a Dirichlet distribution being itself a multinomial distribution (next slide)

## Key distribution 2: Multinomial distribution



The multinomial distribution depicted has probabilities  $[0.2, 0.7, 0.1]$ . A draw  $c \sim \text{Multinomial}([0.2, 0.7, 0.1])$  could e.g. return  $c = 2$

## LDA: Generative model

- ▶ Now that we have reviewed some key concepts, we can study how the LDA assumes our documents have been generated
- ▶ We consider a corpus of  $D$  documents, each with  $N_d$  words
- ▶ Idea: Assume a statistical model that generated the observed documents, then estimate the model and recover latent (unobserved) quantities, i.e. topics, under the assumptions made
- ▶ Each document is assumed to contain weights of topics, and each topic is assumed to contain weights of words
- ▶ For simplicity, let us first imagine a stylised corpus with only  $D = 5$  documents,  $N = 8$  words per document (and also 8 unique words in the corpus in total), and  $K = 3$  topics

## LDA: Generative model - stylised example

Assume the documents have been generated according to the following process:

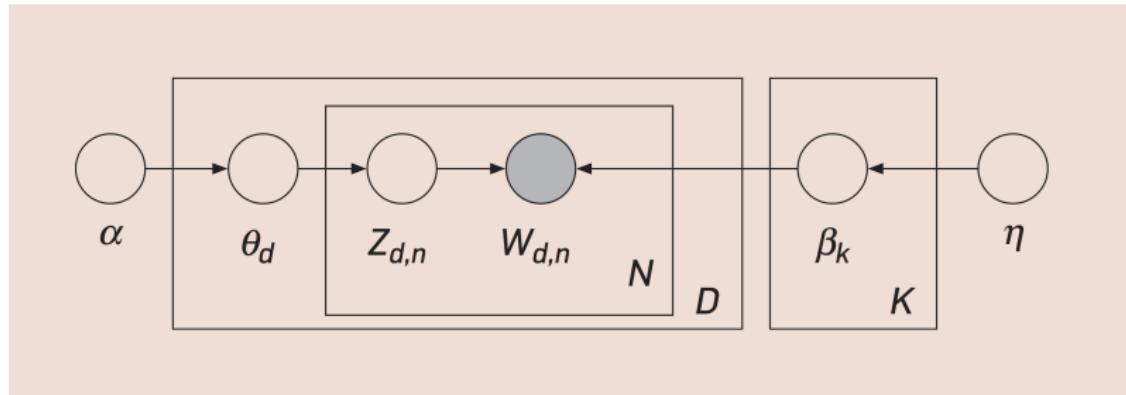
- ▶ For each of the 5 documents, draw its topic shares from a Dirichlet distribution  $\theta_d \sim Dir(\alpha)$ . For document 1 this could e.g. yield  $\theta_1 = (0.1, 0.8, 0.1)$ , i.e. document 1 consists predominantly of topic 2
- ▶ For each of the 3 topics, draw its word shares from a second Dirichlet distribution  $\beta_k \sim Dir(\eta)$ . For topic 2 this could e.g. yield  $\beta_2 = (0, 0, 0, 0.1, 0, 0, 0.4, 0.5)$ , i.e. topic 2 consists predominantly of words 7 and 8
- ▶ Now it is possible to fill each document with words. For each document-word position  $d, n$ 
  - ▶ Choose the topic  $z_{d,n}$  of the document-word position by taking a draw from document  $d$ 's topic distribution:  $z_{d,n} \sim \text{Multinomial}(\theta_d)$ . For example, for the first word in document 1 we might draw topic  $z_{1,1} = 2$
  - ▶ Choose the word  $w_{d,n}$  by taking a draw from the corresponding topic distribution:  $w_{d,n} \sim \text{Multinomial}(\beta_{z_{d,n}})$ . Since we are drawing from topic 2, we might draw word 8. This fills the word at position 1, 1

## LDA: Generative model - general case

Assume the documents have been generated according to the following process:

- ▶ For each document  $d \in \{1, \dots, D\}$ , draw its topic shares from a Dirichlet distribution  $\theta_d \sim Dir(\alpha)$
- ▶ For each topic  $k \in \{1, \dots, K\}$ , draw its word shares from a second Dirichlet distribution  $\beta_k \sim Dir(\eta)$
- ▶ Fill each document with words. For each document-word position  $d, n$  in  $d \in \{1, \dots, D\}$  and  $n \in \{1, \dots, N_d\}$ 
  - ▶ Choose the topic  $z_{d,n}$  of the document-word position by taking a draw from document  $d$ 's topic distribution:  $z_{d,n} \sim \text{Multinomial}(\theta_d)$
  - ▶ Choose the word  $w_{d,n}$  by taking a draw from the corresponding topic distribution:  $w_{d,n} \sim \text{Multinomial}(\beta_{z_{d,n}})$

## In plate notation



From: “Probabilistic Topic Models” by David Blei, 2012

## Estimation

- ▶ The assumed generative model is of course at best only a sketch of true language generation
- ▶ Yet, assuming the documents have been generated in such a way, makes it possible in return to back out the shares of topics within documents and the share of words within topics

## Estimation output

- ▶ Say we have a corpus with  $D = 1,000$  documents, a vocabulary of  $V = 10,000$  total unique words/n-grams and chose  $K = 3$  topics. The final output after estimating the topic model could be

$$\theta = \underbrace{\begin{pmatrix} \theta_{1,1} & \theta_{1,2} & \theta_{1,3} \\ \theta_{2,1} & \theta_{2,2} & \theta_{2,3} \\ \dots & \dots & \dots \\ \theta_{1000,1} & \theta_{1000,2} & \theta_{1000,3} \end{pmatrix}}_{1000 \times 3} = \underbrace{\begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 0.3 & 0.1 & 0.6 \\ \dots & \dots & \dots \\ 0.1 & 0.8 & 0.1 \end{pmatrix}}_{1000 \times 3}$$

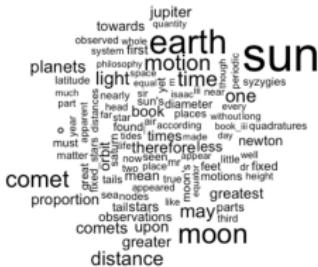
$$\beta = \underbrace{\begin{pmatrix} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,10000} \\ \beta_{2,1} & \beta_{2,2} & \dots & \beta_{2,10000} \\ \beta_{3,1} & \beta_{3,2} & \dots & \beta_{3,10000} \end{pmatrix}}_{3 \times 10000} = \underbrace{\begin{pmatrix} 0.04 & 0.01 & \dots & 0.0001 \\ 0.00002 & 0.001 & \dots & 0.05 \\ 0.00001 & 0.03 & \dots & 0.0001 \end{pmatrix}}_{3 \times 10000}$$

## Estimation intuition

- ▶ Estimating the topic models is done in a Bayesian framework
- ▶ Our  $Dir(\alpha)$  and  $Dir(\eta)$  are the so called *prior distributions* of the  $\theta_d$  and  $\beta_k$
- ▶ With Bayes' rule, and with the help of our data and the model, we update these prior distributions to obtain a new so called *posterior distribution* for each  $\theta_d$  and  $\beta_k$
- ▶ The means of these posterior distributions are the rows in the above matrices commonly outputted by statistical packages and referred to as  $\theta$  and  $\beta$

# Topics

- ▶ What is referred to as a “topic” is a row vector in  $\beta$  with a probability for every word in the corpus to belong to that topic
- ▶ For example, the second topic is the third row vector  
 $\beta_2 = (\beta_{2,1}, \beta_{2,2}, \dots, \beta_{2,10000}) = (0.00002, 0.001, \dots, 0.05)$
- ▶ These topics only include word frequencies, names which summarise them are given by researchers
- ▶ A topic is often visualised via a word cloud where the size of words corresponds to their frequency in the vector



A topic which could be termed “astronomy”

## Algorithms to obtain posterior distributions

- ▶ Not the focus of this lecture
- ▶ Common algorithm to obtain posterior distributions for the  $\theta_d$  and  $\beta_k$  are from two main groups
  1. Sampling algorithms such as *Gibbs sampling* approximate the posterior distribution with an empirical distribution by drawing a sequence of samples in a way that ensure its limit distribution is the true posterior distribution
  2. Variational methods approximate the posterior distribution with a parametrised family of distributions. The error of this approximation is minimised and obtaining posteriors thus becomes an optimisation problem
- ▶ See “Probabilistic topic models” by Blei (2012) for links to further resources

# Outline

- ▶ Introduction
- ▶ Latent Dirichlet Allocation (LDA)
- ▶ Extensions
- ▶ Model selection and evaluation
- ▶ Implementations in R
- ▶ Alternative topic modelling approaches
- ▶ Coding

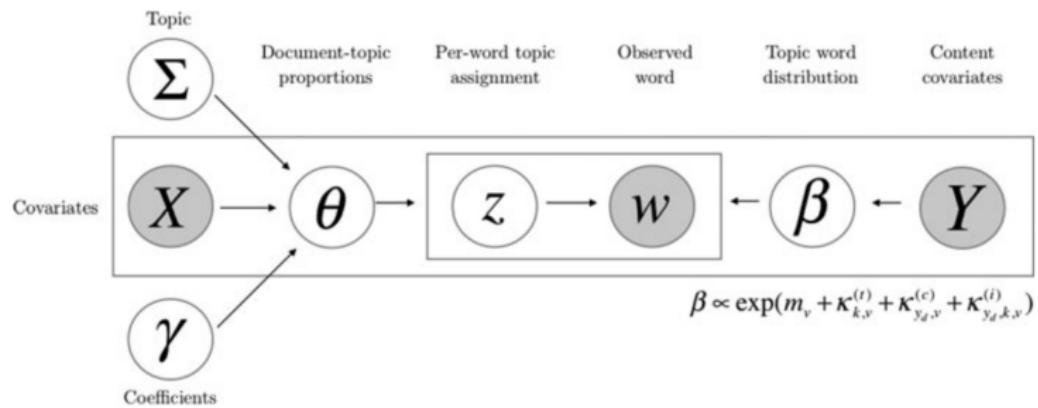
## Correlated topic model

- ▶ Blei and Lafferty (2005, 2007) developed the correlated topic model (CTM)
- ▶ It swaps the Dirichlet distribution of topic shares in documents ( $\theta$ ) with a logistic normal normal distribution
- ▶ The logistic normal distribution is also defined over a simplex, however, unlike the Dirichlet distribution, it also allows to model correlations (between topics) through a covariance matrix
- ▶ This can return estimates of correlations between the topics and can also lead to a better fit

## Structural topic model

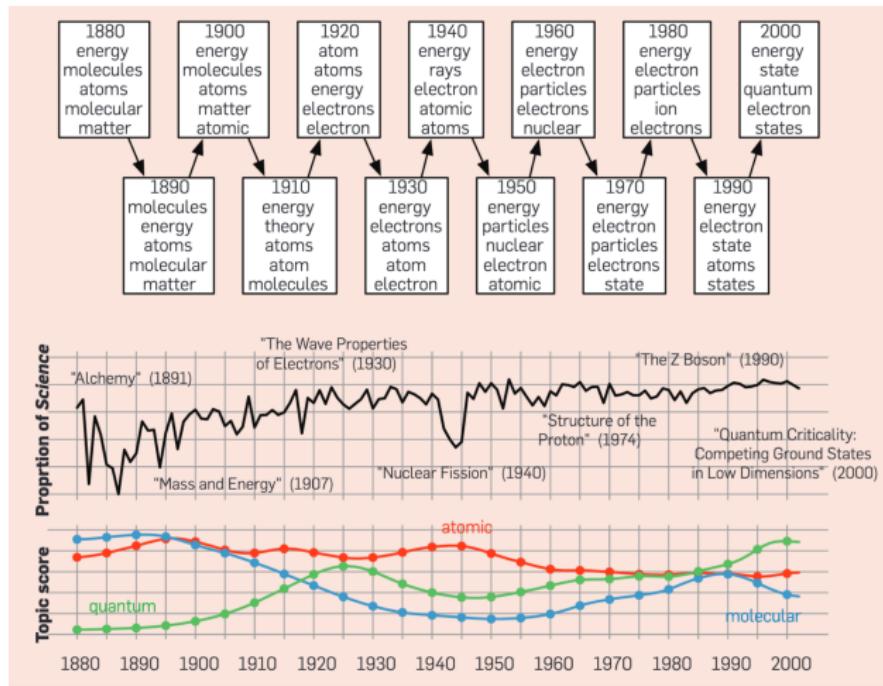
- ▶ The structural topic model was introduced by Roberts et al. (2013). It includes
- ▶ *Topic prevalence* covariates: Topic proportions within documents can vary through covariates (e.g. social media posts by Republican politicians might have different topic proportions than those posted by Democrats)
- ▶ *Topical content* covariates: Word proportions within topics can vary through covariates (e.g. when talking about a health care topic, Republican politicians might use different words than Democrats)
- ▶ Without any covariates supplied, the model reduce to the correlated topic model

# Structural topic model



From: "A Model of Text for Experimentation in the Social Sciences" by  
Roberts, Stewart, and Airoldi, 2016

# Dynamic topic model



A “physics” topics from a dynamic topic model that was fit to Science from 1880 to 2002. From: “Probabilistic Topic Models” by David Blei, 2012

# Outline

- ▶ Introduction
- ▶ Latent Dirichlet Allocation (LDA)
- ▶ Extensions
- ▶ Model selection and evaluation
- ▶ Implementations in R
- ▶ Coding

## Selecting K and beyond

- ▶ In particular selecting the amount of topics  $K$ , but also parameters, covariates, and potentially initialisations is a challenging exercise without a single solution
- ▶ Researchers typically consult a combination of quantitative metrics and human judgement

## Quantitative metrics

- ▶ Held-out likelihood or perplexity: For some held-out documents, how likely would the model have generated/predicted these documents
- ▶ Semantic coherence: For example, how likely do the most common words from a topic also co-occur in the same document?
- ▶ Exclusivity: Do words with high probability in one topic have low probabilities in others?
- ▶ Many automated metrics exist, see e.g. Grimmer and Stewart (2013), Mimno et al. (2011), Taddy (2012)

## On held-out likelihood metrics

- ▶ Held-out likelihood and perplexity allow to judge the predictive ability of the model
- ▶ Yet, choosing  $K$  such that it only achieves the highest held-out likelihood has serious limitations
- ▶ Rather than creating a prediction model or something similar, the purpose of topic modelling is most often to obtain coherent topics that tell a story
- ▶ In their paper “Reading tea leaves” (2009) Chang et al. contrast likelihood based metrics with human judgements about topic coherence
- ▶ Also see this short video by one of the authors

# Reading tea leaves

Word Intrusion

1 / 10	floppy	alphabet	computer	processor	memory	disk
2 / 10	molecule	education	study	university	school	student
3 / 10	linguistics	actor	film	comedy	director	movie
4 / 10	islands	island	bird	coast	portuguese	mainland

Topic Intrusion

6 / 10	DOUGLAS HOFSTADTER							
Douglas Richard Hofstadter (born February 15, 1945 in New York, New York) is an American academic whose research focuses on consciousness, thinking and creativity. He is best known for ", first published in								
<a href="#">Show entire excerpt</a>								
student	school	study	education	research	university	science	learn	
human	life	scientific	science	scientist	experiment	work	idea	
play	role	good	actor	star	career	show	performance	
write	work	book	publish	life	friend	influence	father	

Figure 2: Screenshots of our two human tasks. In the word intrusion task (left), subjects are presented with a set of words and asked to select the word which does not belong with the others. In the *topic intrusion* task (right), users are given a document's title and the first few sentences of the document. The users must select which of the four groups of words does not belong.

From: “Reading Tea Leaves: How Humans Interpret Topic Models” by Chang et al., 2009

- ▶ Likelihood based metrics were actually negatively correlated with human metrics about topic coherence

## Quantitative metrics

- ▶ For a discussion of model selection for (structural) topic models (e.g. different choices of  $K$ , initialisations, and covariates) and evaluation, see e.g. Section “3.4. Evaluate: Model selection and search” in the package vignette or the Section *Model Specification and Selection* in “Structural Topic Models for Open-Ended Survey Responses” by Roberts et al. (2014)
- ▶ In Roberts et al. (2014), the authors e.g. argue that “a semantically interpretable topic has two qualities: (1) it is cohesive in the sense that high-probability words for the topic tend to co-occur within documents, and (2) it is exclusive in the sense that the top words for that topic are unlikely to appear with in top words of other topics.”
- ▶ Semantic coherence and exclusivity, with many other quantitative metrics, are outputs of the function ‘searchK’ in the ‘stm’ package

## Takeways

- ▶ No quantitative metric can replace human judgement when selecting  $K$  or other model parameters, and evaluating the fit of a particular topic model more generally
- ▶ “The most effective method for assessing model fit is to carefully read documents that are closely associated with particular topics to verify that the semantic concept covered by the topic is reflected in the text.” from “A Model of Text for Experimentation in the Social Sciences” by Roberts et al. (2016)
- ▶ The ‘plotQuote’ function in ‘stm’ allows to plot documents highly associated with certain topics
- ▶ Key consideration: What is the goal of the current model?
- ▶ To generate coherent topics which describe themes? Combine careful human reading with quantitative metrics
- ▶ To use topics e.g. as input in a predictive model? Try to select  $K$  and other parameters such that they maximise whatever the objective function

## Example

- ▶ A carefully documented project with very good average topic coherence is e.g. “Leaders or Followers? Measuring Political Responsiveness in the U.S. Congress Using Social Media Data” by Barberá et al. (2014)
- ▶ Data: 651,116 tweets sent by US legislators from January 2013 to December 2014
- ▶ 2,920 documents =  $730 \text{ days} \times 2 \text{ chambers} \times 2 \text{ parties}$
- ▶ Why aggregating? Applications that aggregate by author or day outperform tweet-level analyses (Hong and Davidson, 2010)
- ▶ Sidenote: For short texts, such as also e.g. survey responses, topic models such as sparse additive generative models (SAGE) might create more coherent topics than e.g. LDA (see e.g. Bauer et al, Political Behavior, 2017)
- ▶  $K = 100$  topics
- ▶ Outcomes: <http://pablobarbera.com/congress-lda/>

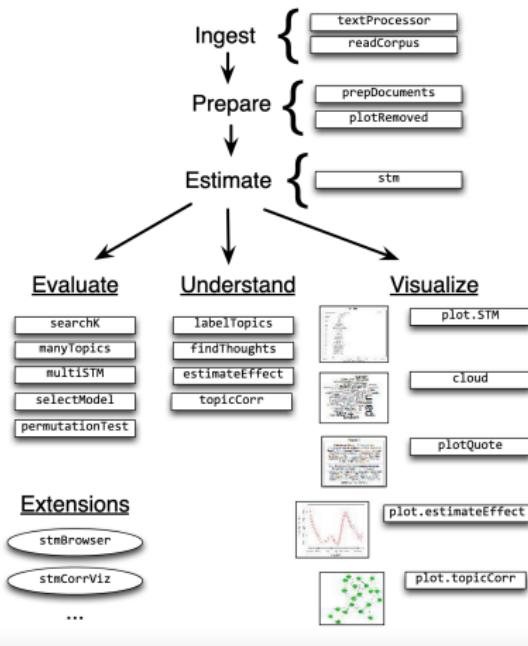
# Outline

- ▶ Introduction
- ▶ Latent Dirichlet Allocation (LDA)
- ▶ Extensions
- ▶ Model selection and evaluation
- ▶ **Implementations in R**
- ▶ Alternative topic modelling approaches
- ▶ Coding

## Implementations in R

- ▶ For an implementation of the LDA and CTM, see e.g. the package 'topicmodels'
- ▶ We will focus on the 'stm' package which offers a range of helpful functionalities
- ▶ Without added covariates, the 'stm' function also estimates a standard CTM, with covariates a structural topic model
- ▶ Further helpful package to visualise topic models are 'LDAvis' or, the stm-specific, 'stminsights'

# Functionalities 'stm' package



From the vignette "stm: RPackage for Structural Topic Models" by Roberts, Stewart, and Tingley

# Outline

- ▶ Introduction
- ▶ Latent Dirichlet Allocation (LDA)
- ▶ Extensions
- ▶ Model selection and evaluation
- ▶ Implementations in R
- ▶ Alternative topic modelling approaches
- ▶ Coding

## Some alternative approaches (more in last week of course)

- ▶ A simple alternative to the probabilistic topic models discussed here would have been to run a clustering algorithm on the row vectors of the dfm
- ▶ Limitations: 1) Bag-of-words approach, 2) each document is assigned to only one cluster/topic (i.e. no mixture model)
- ▶ Recent alternatives: For example, use transformer-based models to obtain better embeddings of documents, paragraphs, or sentences, etc., and cluster them with modern clustering algorithms such as e.g. HDBSCAN (primarily addresses limitation 1))
- ▶ Other options working in natural language directly: For example, send each document to an LLM chat model and ask it to return topics contained in the text (may be relatively slow/less scalable and tougher to standardise)

# Outline

- ▶ Introduction
- ▶ Latent Dirichlet Allocation (LDA)
- ▶ Extensions
- ▶ Model selection and evaluation
- ▶ Implementations in R
- ▶ Coding

# Coding

- ▶ 01-parsing-pdfs.Rmd
- ▶ 02-simple-topic-model.Rmd
- ▶ 03-structural-topic-model.Rmd