TOPIC MODELING

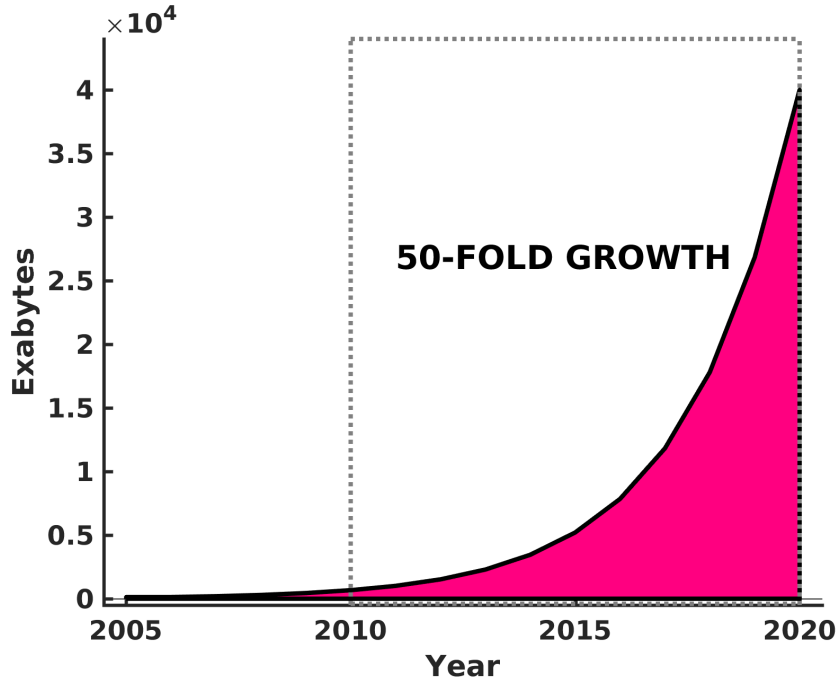
Text Analytics - a Shortcut to Linguistic Evidence

@Center for Language Technology|University of Copenhagen

Kristoffer L Nielbo
knielbo@sdu.dk
knielbo.github.io

Department of History|University of Southern Denmark

SDU



With the current **data deluge**, we want to replace manual modeling with a method that we can throw ++documents at and then let it sort things out automatically

preferably, the output should exhibit some degree of similarity with human text comprehension

"Thou shall know a word by the company it keeps" (Firth, 1975)

Words acquire meaning from their mutual dependencies to or **co-occurrence** with other words (distributional semantics)

```
[1] "it hath been said thou shalt love thy neighbour and hate
thine enemy but i say unto you love your enemies"
[2] "no man can serve two masters for either he will hate the
one and love the other or else he will"
```

Assigning probability to a document is one of the primary objectives of language modeling:

$$P(D) = P(w_1, w_2, w_3, ..., w_n) \tag{1}$$

In generative probabilistic models, documents are assumed to be generated at random

1. generate each word based on n previous words
2. generate each word based on a set of latent variables

`ex.1`

$$hello\ world\ in\ Python$$

$$P(hello, world, in, Python)$$

$$P(hello \mid \langle s \rangle)P(world \mid hello)P(hello \mid in)P(in \mid Python)P(Python \mid \langle /s \rangle) \text{ , for } n = 2$$

Unigram language models - each word w in a document D is assumed to be randomly generated and independent:

$$P(D) = P(w_1)P(w_2)P(w_3)...P(w_n) \tag{2}$$

N-gram language models - a continuous sequence of N items from a documents:

$$P(D) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2)...P(w_n \mid w_1, ...w_{n-1}) \tag{3}$$

in reality documents are not randomly generated AND exhibits long-range dependencies

Vector space models represent each document as an array of features
Representation types:

- bag-of-words (unigram count), document is a sparse vector of size equals to the vocabulary size
- TF-IDF, like BOW bet penality for words that are frequent in multiple documents
- topic models, document is a low-rank dense vector

document similarity (or query-document similarity) can be expressed as either cosine distance:

$$cos(\theta) = \frac{A \cdot B}{\| A \|_2 \| B \|_2} \tag{4}$$

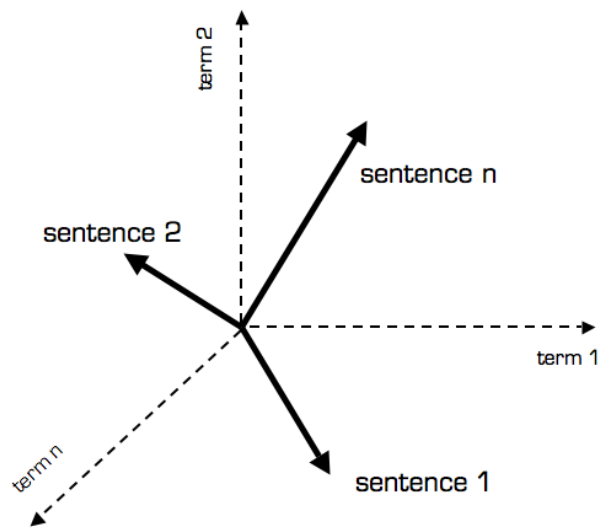
or Kullback-Leibler divergence:

$$D_{KL}(P \parallel Q) = \sum_{i=1}^{n} P(i) \times \log_2 \frac{P(i)}{Q(i)} \tag{5}$$

Any collection of *m* documents can be represented using vector space model by a **document-term (co-occurrence) matrix** of *m* documents and *n* terms

- a document vector with only one word is collinear to the vocabulary word axis
- a document vector that does not contain a specific word is orthogonal/perpendicular to the word axis
- two documents are identical if they contain the same words in a different order (BOW assumption)

Document space	t_1	t_2	t_3	...	t_n ← Term vector space
D_1	a_{11}	a_{12}	a_{13}	...	a_{1n}
D_2	a_{21}	a_{22}	a_{23}	...	a_{2n}
D_3	a_{31}	a_{32}	a_{33}	...	a_{3n}
...					
D_m	a_{m1}	a_{m2}	a_{m3}	...	a_{mn}
Q	b_1	b_2	b_3	...	b_n



Topic Modeling is a set of methods for analyzing the words (or other fine-grained features) of a large collection of documents in order to discover themes that run through them.

1. **Discover** thematic structure
2. **Annotate** documents
3. **Use** the annotations to visualize, organize, summarize, ...

– typically, the number of topics, k, is predefined

– like other **unsupervised techniques**, topic modeling does not require prior class information (annotations, lables).

– derive semantic information by capturing co-occurrence of words at the document level and focus on **heterogeneity** (i.e., that a document can exhibit multiple topics).

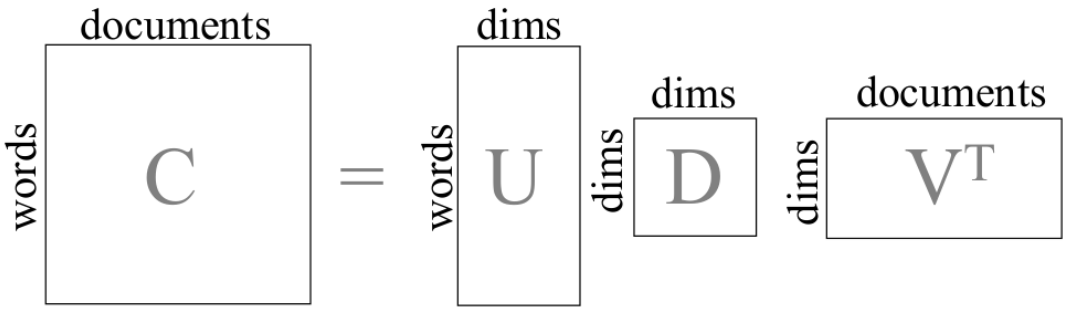
Approaches to topic modeling:

- geometrical Latent Semantic Indexing/Analysis
- probabilistic **Latent Dirichlet Allocation**

Latent Semantic Indexing (in one slide) downsizes a co-occurrence matrix using Singlur Value Decomposition (SVD)

$$C = UDV^T \tag{6}$$

C is co-occurrence matrix (normalized) and D is a diagonal matrix containing the singular values



LSI creates a low-rank approximation to the document-term matrix $\sim$ 200-300 topics

```
1 from gensim import corpora, models
2
3 corpus = corpora.MmCorpus('/tmp/deerwester.mm')
4 id2word = corpora.Dictionary.load('/tmp/deerwester.dict')
5
6 lsi = models.LsiModel(corpus, id2word=id2word, num_topics=200)
```

A **document** is a structured* or non-random collection of words

Who or what is the structuring agent?

How to **create a document**

1. Choose a distribution over topics
2. : For each word in the document:
   - Choose a topic from 1
   - Choose a word from the topic's distribution over the vocabulary

– Each word in a document is (randomly) selected from a (randomly) selected topic from a distribution of topics

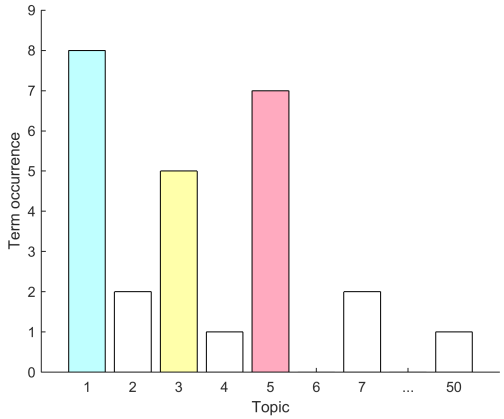
– Each document exhibits multiple topics in different proportions

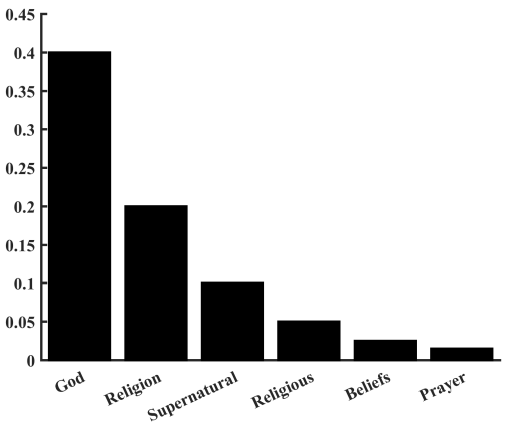
– For the reader, topic structure, per-document topic distributions and the per-document per-word topic assignments are latent, and have to be inferred from observed documents.

Topics create documents: *topics* $\Rightarrow$ *doc* & *words*

If we want to extract the **latent topics** that generated the documents, we have to reverse the process: *words* & *doc* $\Rightarrow$ *topics*

ex.2





ABSTRACT—*We present two studies aimed at resolving experimentally whether religion increases prosocial behavior in the anonymous dictator game. Subjects allocated more money to anonymous strangers when God concepts were implicitly activated than when neutral or no concepts were activated. This effect was at least as large as that obtained when concepts associated with secular moral institutions were primed. A trait measure of self-reported religiosity did not seem to be associated with prosocial behavior. We discuss different possible mechanisms that may underlie this effect, focusing on the hypotheses that the religious prime had an ideomotor effect on generosity or that it activated a felt presence of supernatural watchers. We then discuss implications for theories positing religion as a facilitator of the emergence of early large-scale societies of cooperators.*

Many theorists have suggested that the cognitive availability of omniscient and omnipresent supernatural agents has had a dramatic impact on the development of large-scale human societies. The imagined presence of such agents, along with emotional ritual and costly commitment to the social group they govern, may have been the major development that allowed genetically unrelated individuals to interact in cooperative ways (e.g., Atran & Norenzayan, 2004; Irons, 1991; Sosis & Ruffle, 2004). The research reported in this article experimentally investigated this link between two broad classes of culturally widespread phenomena of interest to social science—religious beliefs and cooperative behavior among unrelated strangers.

Although anecdotes documenting religion's prosocial and antisocial effects abound, the empirical literature has produced mixed results regarding religion's role in prosocial behavior.

[redacted]

Sosis and Ruffle (2004) examined levels of generosity in an experimental cooperative pool game in religious and secular kibbutzim in Israel and found higher levels of cooperation in the religious ones, and the highest levels among religious men who engaged in daily communal prayer. Batson and his colleagues (Batson et al., 1989; Batson, Schoenrade, & Ventis, 1993) have shown that although religious people report more explicit willingness to care for others than do nonreligious people, controlled laboratory measures of altruistic behavior often fail to corroborate this difference. Furthermore, when studies demonstrate that helpfulness is higher among more devoted people, this finding is typically better explained by egoistic motives such as seeking praise or avoiding guilt, rather than by higher levels of compassion or by a stronger motivation to benefit other people.

However insightful these findings are, research on religion and prosocial behavior has been limited by its overwhelming reliance on correlational designs. If religiosity and prosocial behavior are found to be correlated, it is just as likely that having a prosocial disposition causes one to be religious, or that some third variable such as guilt proneness or dispositional empathy causes both cooperative behavior and religiosity, as that religious beliefs somehow cause prosocial behavior. Only rarely have studies induced supernatural beliefs to examine them as a causal factor. Bering (2003, 2006) inhibited 3-year-old children's tendencies to cheat (i.e., open a "forbidden box") by telling them that an invisible agent ("Princess Alice") was in the room with them. In a different study, college students who were casually told that the ghost of a dead graduate student had been spotted in their private testing room were less willing to cheat on a computerized spatial-reasoning task than were those told nothing (Bering, McLeod, & Shackelford, 2005). These studies suggest that explicit thoughts of supernatural agents curb cheating behavior.

In the research reported here, we examined the effect of God concepts specifically on selfish and prosocial behavior. Our research design was novel in two ways. First, we introduced an

Latent Dirichlet Allocation (LDA) is a simple class of topic models

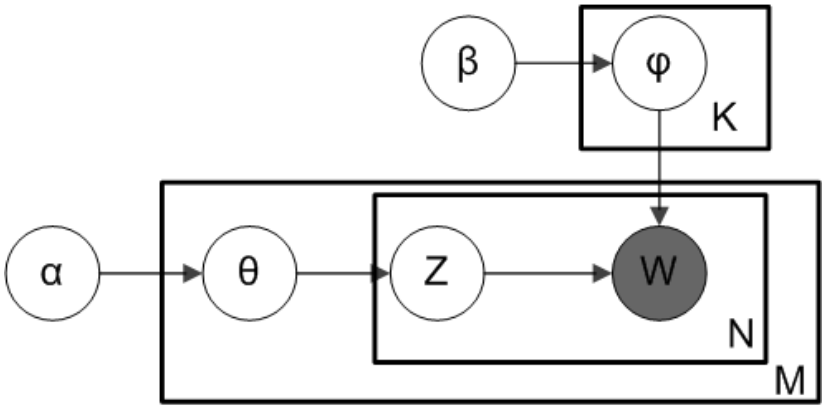
Competing sparsity (goals of LDA)

1. For each *document*, allocate its words to as *few topics* as possible
2. For each *topic*, assign high probability to as *few terms* as possible

These *goals are at odds*, because putting one topic in a single documents complicates 2 and putting few words in each topic complicates 1

All of the words in a document have a probability under each topic (1), but to cover a document's words, we must assign many topics to it (2)

Trade off between these goals results in tightly co-occurring words within each topic



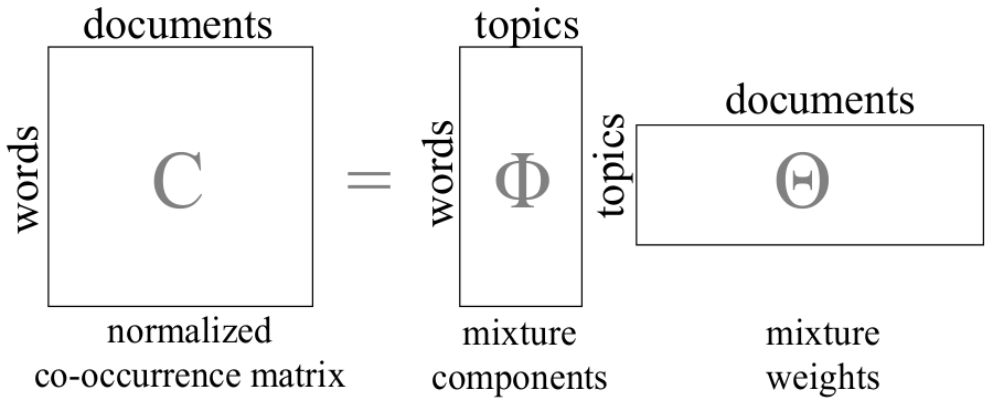
α	Dirichlet prior for per-doc topic dist - proportions parameter
β	Dirichlet prior for per-topic word dist - topic parameter
θ_i	word dist for topic - per-document topic proportions
ϕ_k	word dist for topic k - topics
Z_{ij}	topic for j^{th} word in doc i - per-word topic assignment
W_{ij}	the observed word

Procedure 1 Generative Model

1: **choose** $\theta_i \sim Dir(\alpha)$, i is a document
2: **choose** $\phi_k \sim Dir(\beta)$, k is a topic
3: **for** each word position **do**
4: **choose** a topic $z_{ij} \sim Multinomial(\theta_i)$
5: **choose** a word $w_{ij} \sim Multinomial(\phi_{z_{ij}})$
6: **end for**

The joint distribution defines a posterior probability: $P(\theta, z, \phi)$
use posterior to:
Train on a corpus: Bayesian inference on θ and ϕ
Train on a new documents d: fix $P(w \mid z)$ to infer $P(z \mid d)$
– Multiple inference algorithms available (expectation-maximization/VEM and Gibbs sampling/GIBBS)

Matrix decomposition interpretation of LDA



The document-term co-occurrence matrix C can be decomposed into a product of the topic matrix ϕ and the document matrix θ

Hyper-parameters are parameters that are not directly learnt within estimators

α is a proportions parameter that for low values places more weight on documents composed of only a few dominant topics (sparse document representations). High α values return more relatively dominant topics (dense document representations):

$$\alpha = \frac{50}{k} \tag{7}$$

β is a topic parameter that for low values places more weigth on topics composed of only a few dominant words*. High values of β return topics composed of many words (flatter distributions).

$$\beta = 0.01 \tag{8}$$

k determines the model's granularity, that is, its number of topics. Low values of k (few topics) return a model with a very general topical structure (low granularity). High values of k return a more specific topical structure (high granularity).
– k can be estimated using maximum likelihood estimation and is often reported in perplexity reduction

$$\mathcal{L}(W) = \log p(W \mid \phi, \alpha) = \sum_d \log p(W_d \mid \phi, \alpha) \tag{9}$$

$$perplexity(W) = \exp\{-\frac{\mathcal{L}(W)}{count\ of\ tokens}\} \tag{10}$$

THANK YOU

knielbo@sdu.dk
knielbo.github.io

& credits to
Jaimie Murdock and Karol Grzegorczyk