Topic modeling

A brief non-technical walk through the algorithm

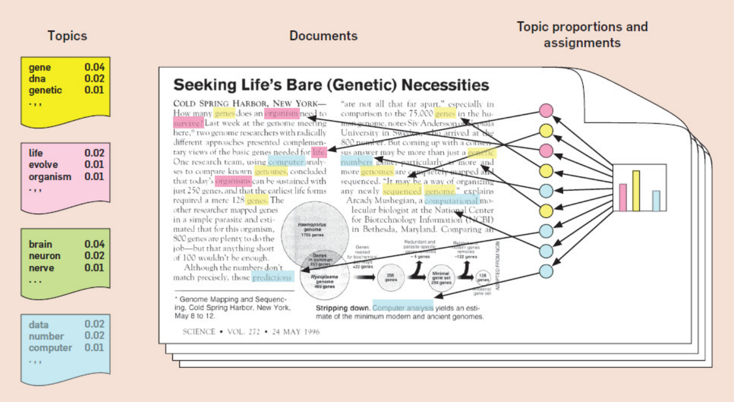
Norwegian Research School, NTNU, 5 February 2025

Gregory Ferguson-Cradler
Univesity of Inland Norway

Before topic models

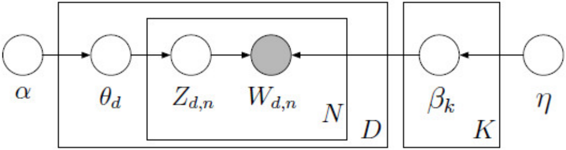
- From a CS perspective: too many/much text out there, need a way of getting an overview in an automated fashion
- Many algorithms before aimed at categorizing texts
- Topic modelling: probabalistic (assumed based on stochostic processes) and mixed membership (one document can have multiple topics) model.
- Problem behind the algorithm: how do we learn the underlying thematic structure (and probability distribution) that created a certain topic (called *posterior inference*).

The assumptions behind the topic model algorithm



Text is produced by chosing a distribtion of topics within the given document; then for every word a selection of topic based on the document-level distribution; finally a word from the corresponding topic (Blei 2012, 78). For my best attempt at a non-technical explanation of topic models, see (Ferguson-Cradler 2021).

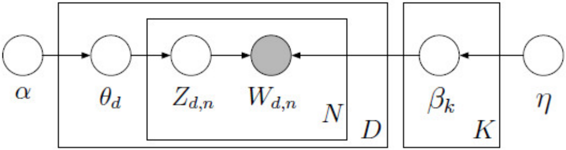
The document generating process



Interrelations of the probabalistic data generating process (Blei and Lafferty 2009, 78).

- $\vec{\beta}_k \sim \mathrm{Dir}_V(\eta) \rightarrow$ *distribution over the vocabulary*

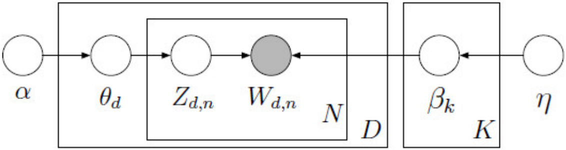
The document generating process



Interrelations of the probabalistic data generating process (78).

- $\vec{\beta}_k \sim \mathrm{Dir}_V(\eta) \rightarrow$ *distribution over the vocabulary*
- $\vec{\theta}_d \sim \mathrm{Dir}_k(\vec{\alpha}) \rightarrow$ *distribution over the topics*

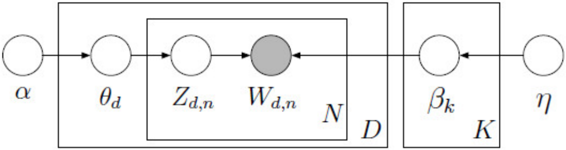
The document generating process



Interrelations of the probabalistic data generating process (78).

- $\vec{\beta}_k \sim \mathrm{Dir}_V(\eta) \rightarrow$ *distribution over the vocabulary*
- $\vec{\theta}_d \sim \mathrm{Dir}_k(\vec{\alpha}) \rightarrow$ *distribution over the topics*
- $Z_{d,n} \sim \mathrm{Mult}(\vec{\theta}), Z_{d,n} \in \{1, ..., K\}$

The document generating process



Interrelations of the probabalistic data generating process (78).

- $\vec{\beta}_k \sim \mathrm{Dir}_V(\eta) \rightarrow$ *distribution over the vocabulary*
- $\vec{\theta}_d \sim \mathrm{Dir}_k(\vec{\alpha}) \rightarrow$ *distribution over the topics*
- $Z_{d,n} \sim \mathrm{Mult}(\vec{\theta}), Z_{d,n} \in \{1, ..., K\}$
- $W_{d,n} \sim \mathrm{Mult}(\vec{\beta}_{Z_{d,n}}), W_{d,n} \in \{1, ..., V\}$

Fitting the model

First, randomly assign a topic to each word in the document. We can now compute θ and β distributions. Now, for every word, compute:

$$P(K|d,n) = \frac{tf_{K,n} + \eta}{tf_K} \cdot (tf_{K,d} + \alpha)$$

and reassign based on new most likely topic assignment.

- This is a process that does not seem much like the act of writing as we know it, but it *might* give interesting results.

Fitting the model

First, randomly assign a topic to each word in the document. We can now compute θ and β distributions. Now, for every word, compute:

$$P(K|d,n) = \frac{tf_{K,n} + \eta}{tf_K} \cdot (tf_{K,d} + \alpha)$$

and reassign based on new most likely topic assignment.

- This is a process that does not seem much like the act of writing as we know it, but it *might* give interesting results.
- One parameter we must set: k. Best practices recommend fiddling with it until you get a model fit that is coherent.

Fitting the model

First, randomly assign a topic to each word in the document. We can now compute θ and β distributions. Now, for every word, compute:

$$P(K|d,n) = \frac{tf_{K,n} + \eta}{tf_K} \cdot (tf_{K,d} + \alpha)$$

and reassign based on new most likely topic assignment.

- This is a process that does not seem much like the act of writing as we know it, but it *might* give interesting results.
- One parameter we must set: k. Best practices recommend fiddling with it until you get a model fit that is coherent.
- Concentration hyperparameters (η and α) – the higher they are the more even β and θ.

Fitting the model

First, randomly assign a topic to each word in the document. We can now compute θ and β distributions. Now, for every word, compute:

$$P(K|d,n) = \frac{tf_{K,n} + \eta}{tf_K} \cdot (tf_{K,d} + \alpha)$$

and reassign based on new most likely topic assignment.

- This is a process that does not seem much like the act of writing as we know it, but it *might* give interesting results.
- One parameter we must set: k. Best practices recommend fiddling with it until you get a model fit that is coherent.
- Concentration hyperparameters (η and α) – the higher they are the more even β and θ.
- Our two matrices of interest: θ and β.

Blei, David M. 2012. Probabilistic topic models. *Communications of the ACM* 55 (4): 77–84.

Blei, David M, and John D Lafferty. 2009. Topic models. *Text mining: classification, clustering, and applications* 10 (71): 34.

Ferguson-Cradler, Gregory. 2021. Narrative and computational text analysis in business and economic history. *Scandinavian Economic History Review*, 1–25.