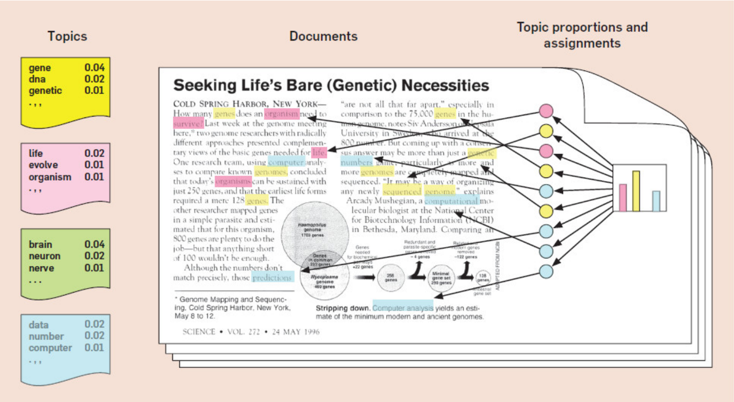
Topic modeling

A brief (semi-technical) walk through the algorithm)

NTNU, Trondheim, 12.-13. August 2021

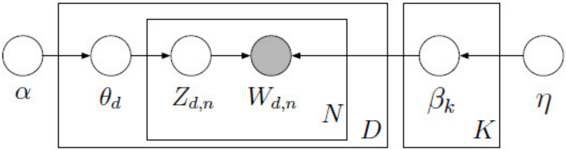
Gregory Ferguson-Cradler
Institutt for rettsvitenskap, filosofi og internasjonale studier
Høgskolen i Innlandet, Lillehammer

The assumptions behind the topic model algorithm



Text is produced by chosing a distribtion of topics within the given document; the for every word a selection of topic based on the document-level distribution; finally a word from the corresponding topic (Blei 2012, 78).

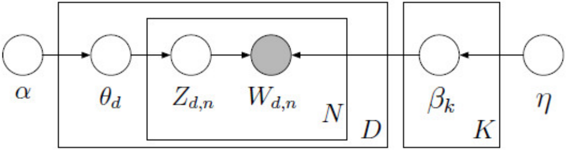
The document generating process



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

- $\vec{\beta}_k \sim \mathrm{Dir}_V(\eta)$

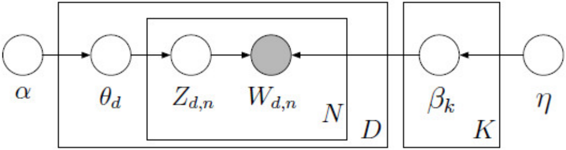
The document generating process



Interrelations of the probabalistic data generating process (78).

- $\vec{\beta}_k \sim \mathrm{Dir}_V(\eta)$
- $\vec{\theta}_d \sim \mathrm{Dir}_k(\vec{\alpha})$

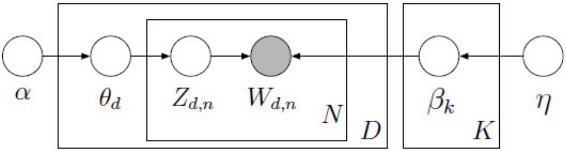
The document generating process



Interrelations of the probabalistic data generating process (78).

- $\vec{\beta}_k \sim \mathrm{Dir}_V(\eta)$
- $\vec{\theta}_d \sim \mathrm{Dir}_k(\vec{\alpha})$
- $Z_{d,n} \sim \mathrm{Mult}(\vec{\theta}), Z_{d,n} \in \{1, ..., K\}$

The document generating process



Interrelations of the probabalistic data generating process (78).

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

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.

Practice

Enough Greek letters, let's see how to do this in practice.