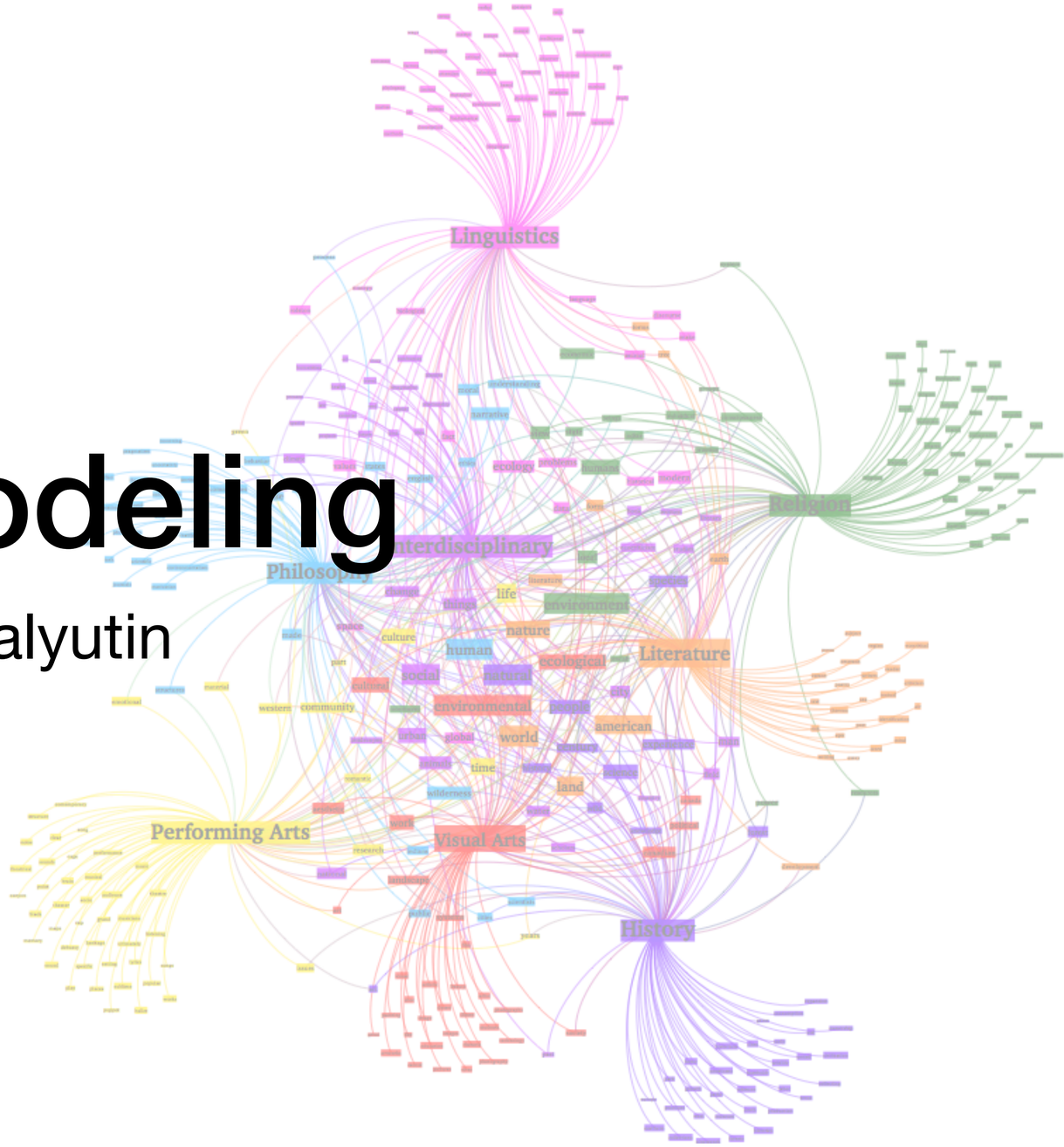
Topic modeling

Eugeny Malyutin



Topic modeling in a minute:

Topic model — text documents collection model determining which topics are present in every collection's document

The training algorithms receives an **unannotated texts** collection as input. The output of the algorithm are vectors for every document determining **the extent to which that document corresponds to each of the topics**.

The size of the vector (a number of topics) can either be a model's parameter or be inferred automatically by the mode. Always we can treat it as fuzzy clustering.

Topic modeling example

Topic modeling: motivation

- News streams analysis and aggregation
- Documents, images,videos, music rubrication
- Recommendation services (collaborative filtration)
- Scientific information exploratory search
- Experts, reviewers, projects search
- Trends and research directions analysis
- Genome analysis (?!)

Formal task:

Given:

- **W** - vocabulary, set of words (or **terms**)
- **D** - collection of text documents
- **n_dw** - a counter «How many times did the term **w** occur in document **D**»

Task:

- Find topics for each document
- Which terms define this topic?

Additional tasks:

- How many topics in collection?
- Find a hierarchy in a topics;
- Track an evolution of a topic through time
- Find topics distribution for linked objects: pic’s, authors, programs etc.

Topic modeling

Hypotesis:

- Bag of docs - the order of the documents in the collection is not important
- Bag of words - the order of the words in the document is not important
- Words found in almost all documents are not important.
- A word in different forms is the same word.
- The document consists of a small number of topics.
- The topic is defined by a small number of terms.

Preprocessing:

- Stemming and/or lemmatisation
- Term (keyphrase) extraction
- Stop-words

Probabilistic mode on:

Assumptions:

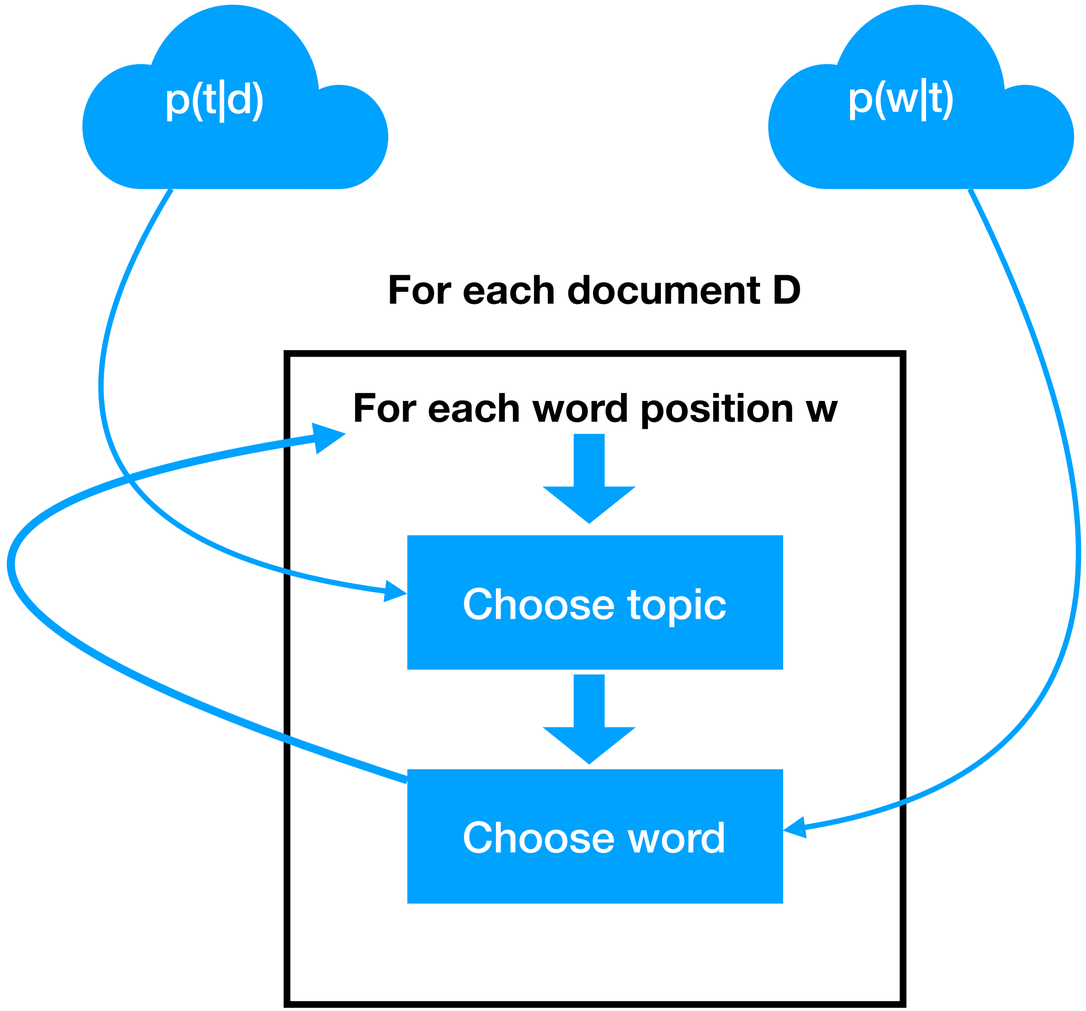
- Every word is linked with some topic **t** from **T**
- **D** - i.i.d (d, w), **p(d,w,t)** from **D x W x T**
- Conditional independence: p(w|d,t) = p(w|t)

Generative model:

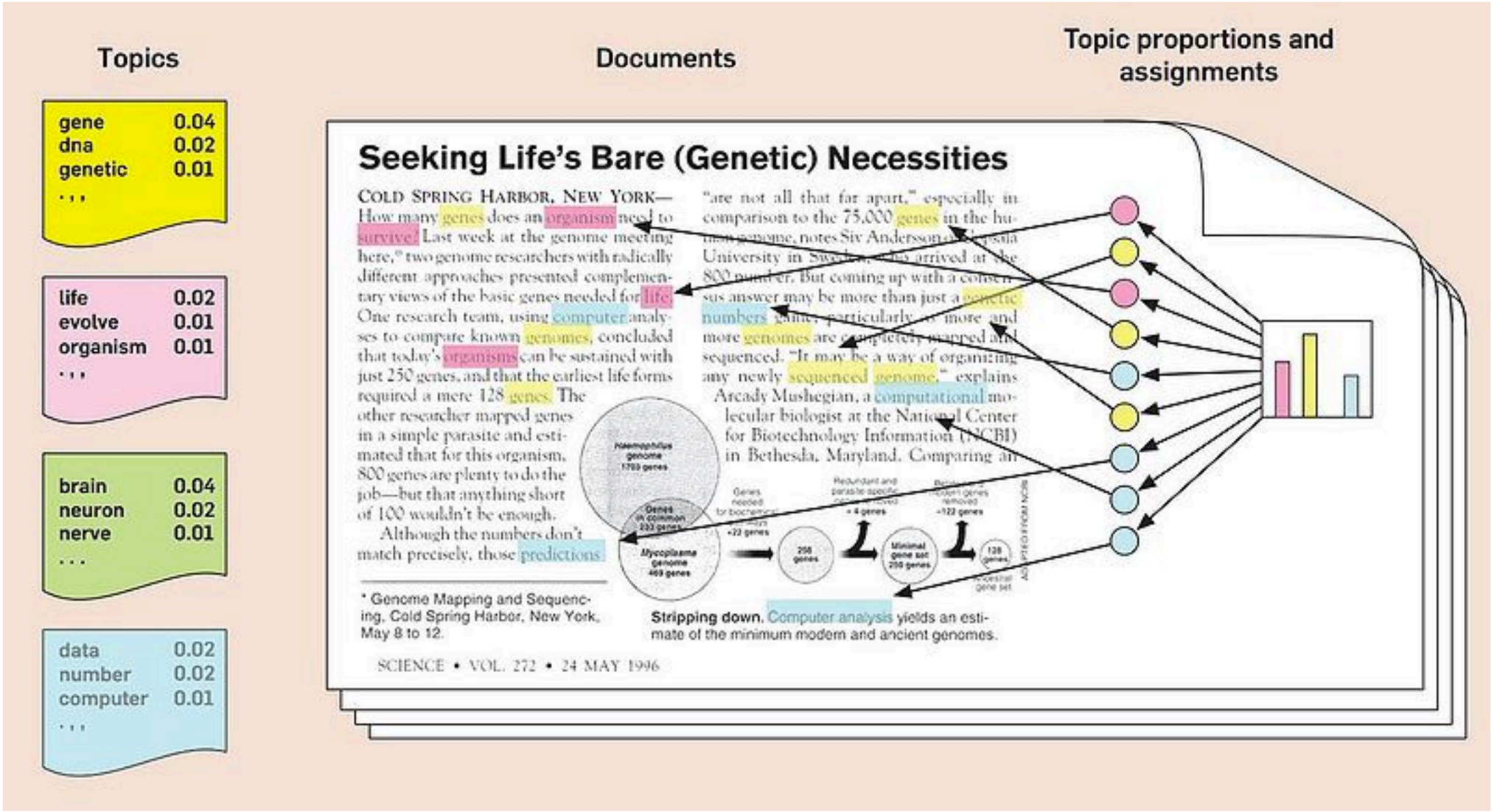
$$p(w,d) = \sum_{t \in T} p(w \mid d,t)p(t \mid d) \approx \sum_{t \in T} p(w \mid t)p(t \mid d)$$

Need to find:

- p(w | t) — terms distribution for all topics
- p(t | d) — topics distribution for all documents



Topic modeling example:



PLSA - probabilistic latent semantic analysis:

$$Maximize\ likelihood\ by\ \phi_{wt} = p(w \mid t)\ \theta_{td} = p(t \mid d)$$

Likelihood:

$$\sum_{d \in D} \sum_{w \in d} n_{dw} p(w \mid d) = \sum_{d \in D} \sum_{w \in D} n_{dw} ln \sum_{t \in T} \phi_{wt} \theta_{dt} \to \max_{\Phi, \Theta}$$

With given constraint (nonnegative and normalisation):

$$\phi_{wt} \geq 0 \quad \sum_{w \in W} \phi_{wt} = 1 \qquad \theta_{td} \geq 0 \quad \sum_{d \in D} \theta_{td} = 1$$

Interpretation 1: Minimisation total (by docs) Kullback-Leibler divergence between topic model p(m|d) (our topic model) and unigram model

$$KL(\hat{p} \mid\mid p) = \sum_{d \in D} \sum_{w \in d} \hat{p}(w \mid d)\ ln \frac{\hat{p}(w \mid d)}{p(w \mid d)} \to min$$

PLSA - probabilistic latent semantic analysis:

$$Maximize\ likelihood\ by\ \phi_{wt} = p(w \mid t)\ \theta_{td} = p(t \mid d)$$

Likelihood:

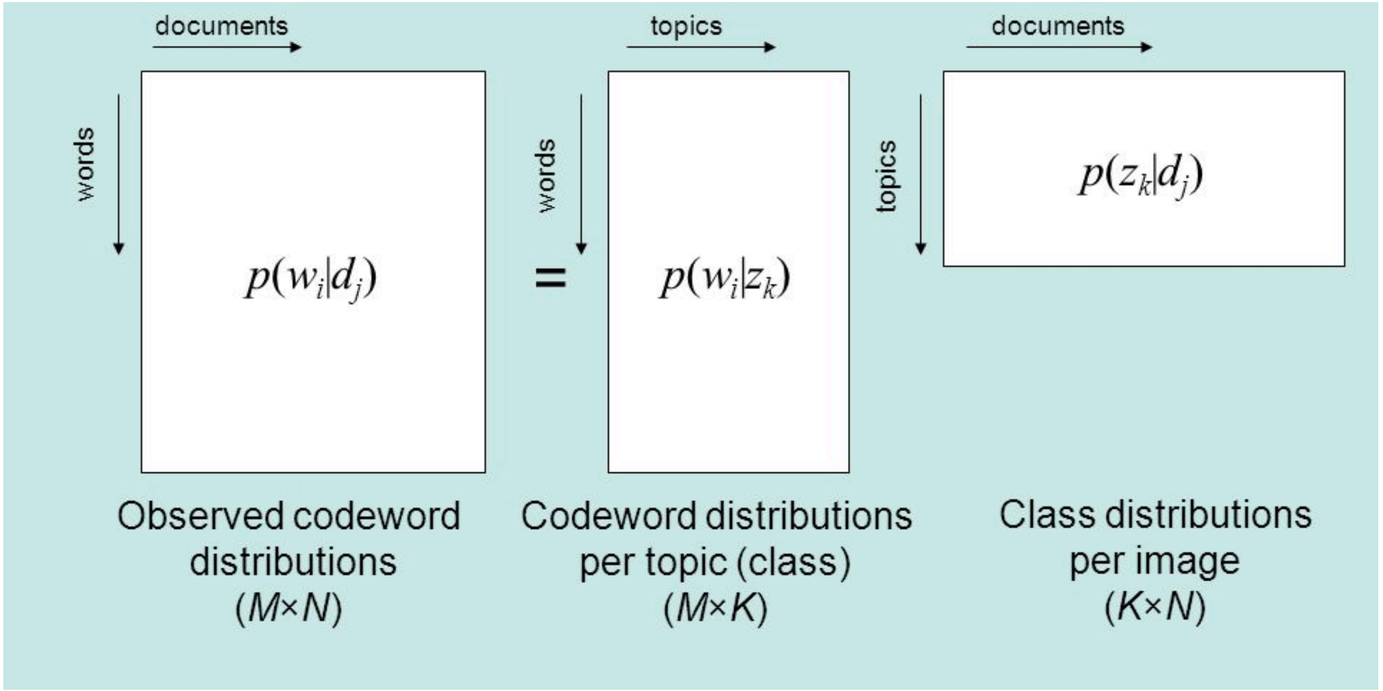
$$\sum_{d \in D} \sum_{w \in d} n_{dw} p(w \mid d) = \sum_{d \in D} \sum_{w \in D} n_{dw} ln \sum_{t \in T} \phi_{wt} \theta_{dt} \to \max_{\Phi, \Theta}$$

Interpretation 2: non-negative matrix factorisation $F \approx \Phi\Theta$

$F = (\hat{p}(w \mid d))_{W \times D}$ - known data matrix

$\Phi = (\phi_{wt})_{W \times T}$ - unknown matrix term/ topic

$\Theta = (\theta_{td})_{T \times D}$ - unknown matrix topic/ document



PLSA with EM-algorithm:

E-step: with **known** ϕ and θ we can infer conditional prob for all topics t by given (d,w):

$$H_{dwt} = p(t \mid d, w) = \frac{p(w, t \mid d)}{p(w, d)} = \frac{p(w \mid t)p(t \mid d)}{p(w \mid d)} = \frac{\phi_{wt}\theta_{td}}{\sum_s \phi_{ws}\theta_{sd}}$$

M-step: with **known H_{dwt}** we can easily write out all needed probability estimations:

$$\phi_{wt} = \frac{\hat{n}_{wt}}{\hat{n}_t} \quad \hat{n}_{wt} = \sum_{d \in D} n_{dw} H_{dwt} \quad \hat{n}_t = \sum_{w \in W} \hat{n}_{wt}$$

$$\theta_{td} = \frac{\hat{n}_{dt}}{\hat{n}_d} \quad \hat{n}_{dt} = \sum_{w \in d} n_{dw} H_{dwt} \quad \hat{n}_d = \sum_{t \in T} \hat{n}_{dt}$$

pLSA discussions:

- Slow convergence on long text collections;
- PLSA don’t sparse H_{dwt} = p(t | d,w) matrix; and we need to store 3-D data matrix in memory
- No way to reduce (or adjust) Phi and Theta sparsity
- the more documents there are, the larger the number of parameters => we overfit easily (however, after the removal of rare words, things are not that bad)
- if we see a new document d, we can’t estimate p(t|d) without retraining the model
- stochastic matrix decomposition is an **ill-posed** problem, which means it can have an infinite number of solutions, which leads to the instability of ‘recovered’ matrices phi and theta (this is not only pLSA’s problem, however)

«Rational» EM-algorithm:

Algorithm 2.1: The rational EM-algorithm for PLSA.

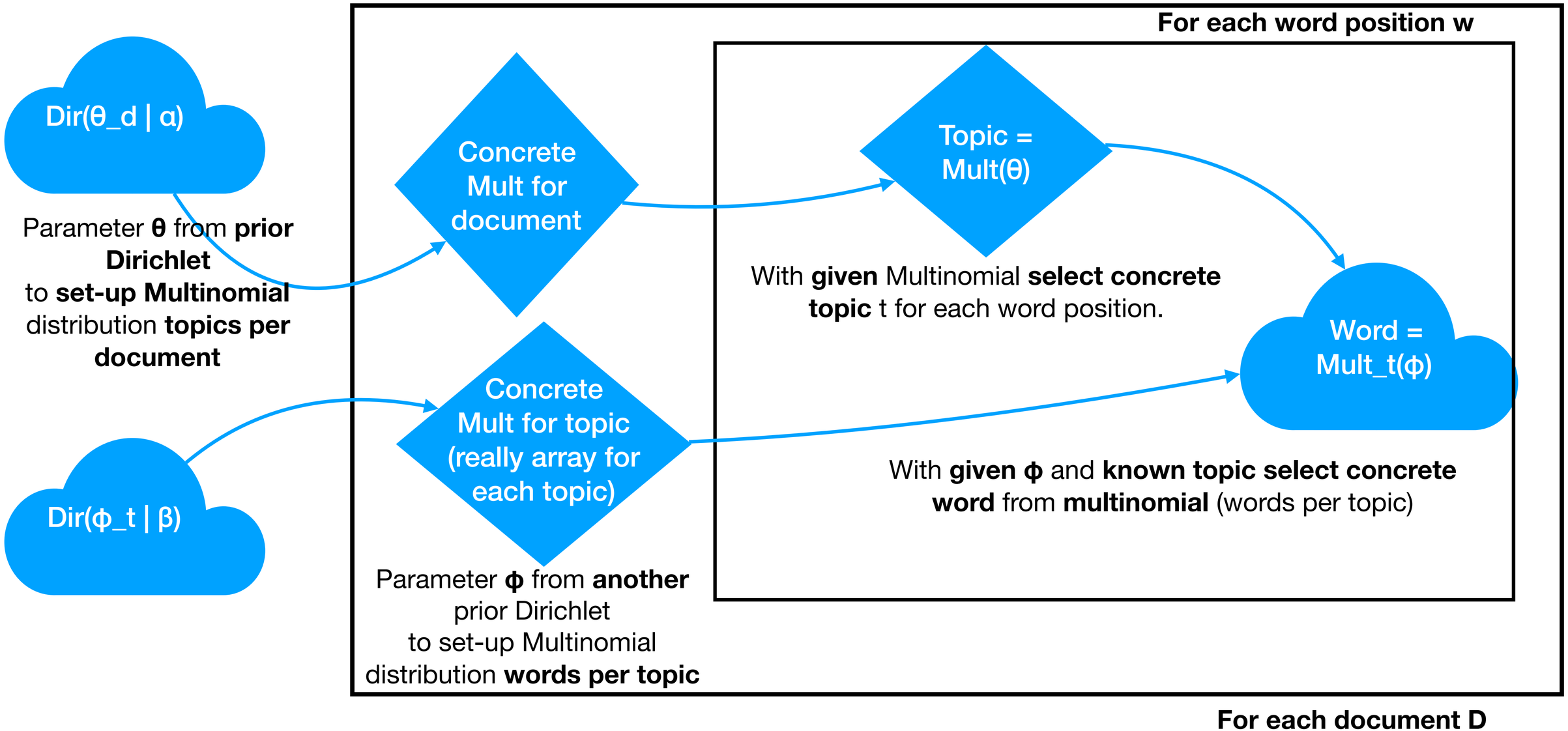
Input: document collection D, number of topics $|T|$, initialized Φ, Θ;
Output: Φ, Θ;

repeat
  zeroize n_{wt}, n_{dt}, n_t, n_d for all $d \in D,\ w \in W,\ t \in T$;
  for all $d \in D,\ w \in d$
    $Z := \sum_{t \in T} \phi_{wt}\theta_{td}$;
    for all $t \in T$: $\phi_{wt}\theta_{td} > 0$
      increase n_{wt}, n_{dt}, n_t, n_d by $\delta = n_{dw}\phi_{wt}\theta_{td}/Z$;
  $\phi_{wt} := n_{wt}/n_t$ for all $w \in W,\ t \in T$;
  $\theta_{td} := n_{dt}/n_d$ for all $d \in D,\ t \in T$;
until Φ *and* Θ *converge*;

Implicit probabilities

Matrices updates

LDA - Latent Dirichlet Allocation



LDA - Latent Dirichlet Allocation

Generative model: $p(w \mid d) = \sum_{t \in T} p(w \mid t) p(t \mid d)$

Hypothesis (of prior Dirichlet allocation):

- $\theta_d = (\theta_{td})_{t \in T} \in R^{|T|}$ - random vector from Dirichlet allocation (topics in docs):

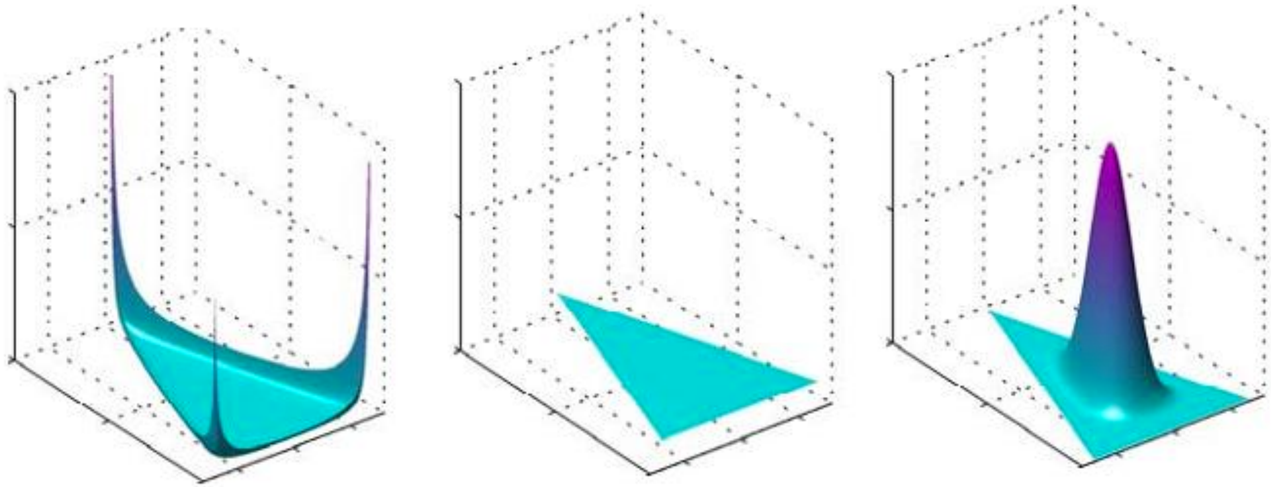
$$Dir(\theta_d \mid \alpha) = \frac{\Gamma(\alpha_0)}{\prod \Gamma(\alpha_t)} \prod_t \theta_{td}^{\alpha_i - 1}, \quad \alpha_0 = \sum_t \alpha_i, \quad \sum_t \theta_t = 1 \quad \alpha \in R^{|T|}$$

- $\phi_t = (\phi_{wt})_{w \in W} \in R^{|W|}$ - random vector from Dirichlet allocation (words in topics):

$$Dir(\phi_t \mid \beta) = \frac{\Gamma(\beta_0)}{\prod \Gamma(\beta_t)} \prod_w \phi_{wt}^{\beta_w - 1}, \quad \phi_0 = \sum_w \phi_i, \quad \sum_w \phi_w = 1 \quad \beta \in R^{|W|}$$

LDA - Why Dirichlet?

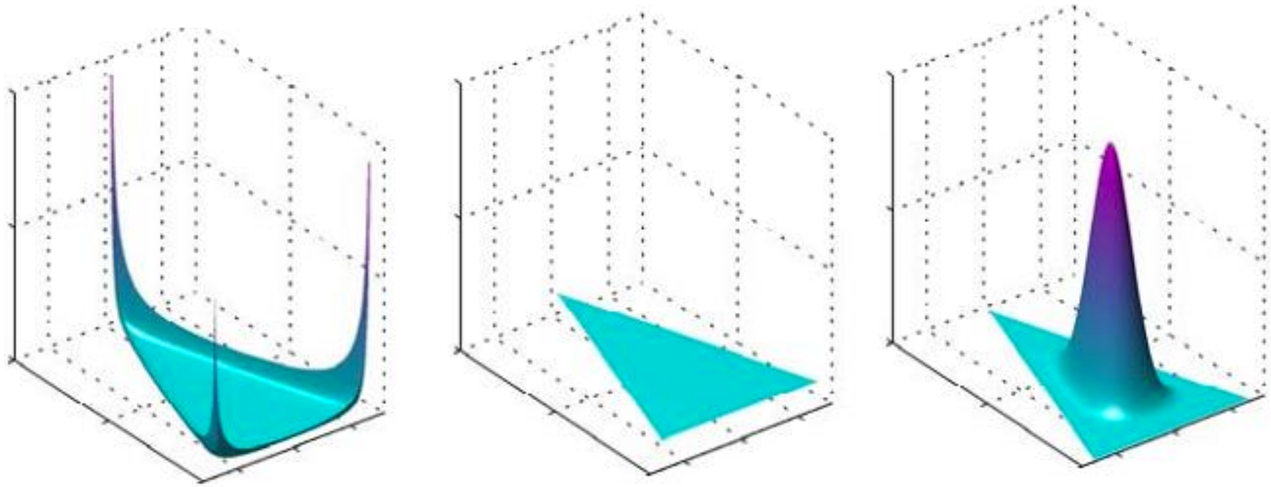
- Dirichlet models cluster structure of multinomial distributions;
- Dirichlet allows us to model sparse distributions (a = 0.1). But there is no way to set concrete component to zero (only close to zero)



LDA - Why Dirichlet?

- Dirichlet distribution conjugate to multinomial

$$p(\Theta_d|X_d) = \frac{p(X_d|\Theta_d)Dir(\theta_d|\alpha)}{\int p(X_d|\theta)Dir(\theta|\alpha)d\theta} \approx \prod_t \theta_{td}^{n_{td}}\theta_{td}^{\alpha_i-1} = Dir(\theta_{td}, \alpha')$$



$\{\alpha_k\} = 0.1$ $\{\alpha_k\} = 1$ $\{\alpha_k\} = 10$

LDA discussion:

- **no linguistic clues** for using Dirichlet distribution
- **smoothing instead of sparsification** (naturally, most topics are usually NOT PRESENT in the document)
- there are **numerous LDA extensions** for taking into account extra constraints and for solving other tasks; however, most of the times their preparation is a complex mathematical task
- if dataset is large enough, there is **not much difference** between LDA and pLSA

LDA vs pLSA

- pLSA — unbiased MLE for p(w|t) and p(t|d)

$$\phi_{wt} = \frac{n_{wt}}{n_t} \quad \theta_{td} = \frac{n_{td}}{n_t}$$

- LDA — biased(smoothed) estimation:

$$\phi_{wt} = \frac{n_{wt} + \beta_w}{n_t + \beta_0} \quad \theta_{td} = \frac{n_{td} + \alpha_w}{n_t + \alpha_0}$$

pLSA:

- Maximize likelihood: $$\sum_{d\in D}\sum_{w\in D} n_{dw} ln \sum_{t\in T} \phi_{wt}\theta_{dt} \to \max_{\Phi,\Theta}$$

  With constraints: $$\phi_{wt} \geq 0 \quad \sum_{w\in W} \phi_{wt} = 1 \qquad \theta_{td} \geq 0 \quad \sum_{d\in D} \theta_{td} = 1$$

  $$F \approx \Phi\Theta$$

- Known word-document data: $F = (\hat{p}(w \mid d))_{W\times D}$
- Unknown word-topic matrix: $\Phi = (\phi_{wt})_{W\times T}$
- Unknown topic-doc matrix: $\Theta = (\theta_{td})_{T\times D}$

pLSA EM-algorithm:

E-step

$$H_{dwt} = p(t \mid d, w) = \frac{p(w, t \mid d)}{p(w, d)} = \frac{p(w \mid t)p(t \mid d)}{p(w \mid d)} = \frac{\phi_{wt}\theta_{td}}{\sum_s \phi_{ws}\theta_{sd}}$$

M-step

$$\phi_{wt} = \frac{\hat{n}_{wt}}{\hat{n}_t} \quad \hat{n}_{wt} = \sum_{d \in D} n_{dw} H_{dwt} \quad \hat{n}_t = \sum_{w \in W} \hat{n}_{wt}$$

$$\theta_{td} = \frac{\hat{n}_{dt}}{\hat{n}_d} \quad \hat{n}_{dt} = \sum_{w \in d} n_{dw} H_{dwt} \quad \hat{n}_d = \sum_{t \in T} \hat{n}_{dt}$$

Theorem: MLE optimal point point should satisfy this system of equations;

But: There are a lot (approx inf.) of optimal solutions;

PLSA problems

- Unstable soultion (due to ill-posed task)
- Could overfit on small collections
- We can’t control sparsity
- There is no un-specific terms extraction

PLSA problems

- Unstable soultion (due to ill-posed task)
  Regularisation — find the best solution with additional constraints
- Could overfit on small collections
  Regularisation — smooth/sparse/dimensionality reduction
- We can’t control sparsity
  Regularisation — progressive sparsification
- There is no un-specific terms extraction
  Regularisation — add and smooth background topics

PLSA additive regularisation

- Maximise likelihood with additional regularisers

$$\sum_{d\in D}\sum_{w\in D} n_{dw} ln \sum_{t\in T} \phi_{wt}\theta_{dt} + \sum_{i=1}^{n} \tau_i R(\Phi,\Theta) \to \max_{\Phi,\Theta}$$

with good old constraints:

$$\phi_{wt} \geq 0 \quad \sum_{w\in W} \phi_{wt} = 1 \qquad \theta_{td} \geq 0 \quad \sum_{d\in D} \theta_{td} = 1$$

where $\tau_i \geq 0$ — regularisers weights;

EM with regularisers

$$H_{dwt} = p(t \mid d, w) = \frac{p(w, t \mid d)}{p(w, d)} = \frac{p(w \mid t)p(t \mid d)}{p(w \mid d)} = \frac{\phi_{wt}\theta_{td}}{\sum_s \phi_{ws}\theta_{sd}}$$

$$\phi_{wt} = \frac{\hat{n}_{wt}}{\hat{n}_t} \quad \hat{n}_{wt} = (\sum_{d \in D} n_{dw} H_{dwt} + \phi_{wt}\frac{dR}{d\phi_{wt}})_+ \quad \hat{n}_t = \sum_{w \in W} \hat{n}_{wt}$$

$$\theta_{td} = \frac{\hat{n}_{dt}}{\hat{n}_d} \quad \hat{n}_{dt} = (\sum_{w \in d} n_{dw} H_{dwt} + \theta_{td}(\frac{dR}{d\theta_{td}}))_+ \quad \hat{n}_d = \sum_{t \in T} \hat{n}_{dt}$$

- PLSA $R(\Phi, \Theta) = 0$
- LDA $R(\Phi, \Theta) = \sum_{tw} \beta_w \ ln\phi_{wt} + \sum_{dt} \alpha_t ln\theta_{td}$

//look at Karush — Kuhn — Tucker conditions

Example: pLSA to LDA casting

- If we want our distributions to look like this:

$$\sum_{t\in T}\mathrm{KL}_w(\beta_w\|\phi_{wt})\to\min_{\Phi},\qquad\sum_{d\in D}\mathrm{KL}_t(\alpha_t\|\theta_{td})\to\min_{\Theta}.$$

- We can set smoothing regularisers like this:

$$R(\Phi,\Theta)=\beta_0\sum_{t\in T}\sum_{w\in W}\beta_w\ln\phi_{wt}+\alpha_0\sum_{d\in D}\sum_{t\in T}\alpha_t\ln\theta_{td}\to\max$$

- Then if we write down EM-algorithm steps, we'll see that it has the same updates as LDA!

- That means **LDA is pLSA regularized** with minimization of KL-divergence between phi and beta, alpha and theta

https://github.com/bigartm/bigartm-book

Even more:

- Because of regularizing assumption about LDA's distributions it won't allow to set some vector values to zeros. However, that sometimes may be useful.

  For that purpose more complex LDA extensions are invented. In ARTM, one can easily sparsify vectors, maximizing distance betwee the trained and the preset distributions

$$R(\Phi, \Theta) = -\beta_0 \sum_{t \in T} \sum_{w \in W} \beta_w \ln \phi_{wt} - \alpha_0 \sum_{d \in D} \sum_{t \in T} \alpha_t \ln \theta_{td} \to \max$$

- e.g., if we make alpha and beta uniform (max entropy!), we'll get a sparsifying regularizer

https://github.com/bigartm/bigartm-book

ARTM discussion:

- easy to understand and adopt
- easy to extend without writing down integrals (for adding a regularizer one will just have to take one derivative)
- requires specific skills for regularizers weights tuning and setting their modifications strategies while training

Topic model evaluation:

- Intrinsic evaluation. **Method 1: perplexity**
  This time the model of language is a word distribution

$$\mathcal{P}(D) = \exp\left(-\frac{1}{n}\sum_{d\in D}\sum_{w\in d} n_{dw}\ln p(w|d)\right)$$

  If it is uniform, then it is equal number of word (seems legit, huh?)

  Problem: can't measure on training set.
  But the parameters are connected to the documents!

  Okay then: all parameters related to documents are estimated on the holdout set
  Even better: we split all holdout documents into two parts; parameters related to the documents, are estimated on the first part, the other part is used for computing perpexity

Topic model evaluation:

Intrinsic evaluation. Method 2
Can the experts tag the topic with a title given its ‘top words’? I

ntrinsic evaluation. Method 2’
Insert a ‘wrong’ word into the list of top topic’s words and check whether the experts can find it.
Write down the number of experts’ errors as a quality measure.

Intrinsic evaluation. Method 3 (correlates with way 2)

Topic coherence — mean PMI for topic’s top k words

$$\text{PMI}_t = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i}^{k} \text{PMI}(w_i, w_j)$$

w_i - the i-th word desc

Topic model evaluation:

Extrinsic evaluation

(the best one (!))

MOAR MODELS:

- Other PTMs training techniques: **Variational Inference**, **MCMC** (e.g. **Gibbs Sampling**)
- Topic models need visualization
  (e.g., LDAvis + some tricks in a videocourse by MIPT and Yandex https://www.coursera.org/lecture/unsupervised-learning/vizualizatsiia-tiematichieskikh-modieliei-93ASP)
- Pachinko allocation (PAM):
  PTM, taking correlation between topics into account Pachinko Allocation: DAG-Structured Mixture Models of Topic Correlations. (2006). Wei Li; Andrew McCallum, University of Massachusetts - Amherst.
- Hierarchical Dirichlet process (HDP): “LDA without explicit num toipics”
  Teh, Y. W.; Jordan, M. I.; Beal, M. J.; Blei, D. M. (2006). "Hierarchical Dirichlet Processes" (PDF). Journal of the American Statistical Association. 101: pp. 1566–1581.
- Neural Topic Model (NTM)
  Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. 2015. A novel neural topic model and its supervised extension. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15). AAAI Press 2210-2216.

MOAR TOOLS

- Gensim
  (LSI, LDA, visualization tools)
- BigARTM
  (ARTM with a few prepared regularizers, can be extended)
- Mallet (Java / CLI)
- Other (ton’s of them)