# High-level structure in texts as sets of words - II

Anton Alekseev,
Steklov Mathematical Institute in St Petersburg

NRU ITMO, St Petersburg, 2019
anton.m.alexeyev+itmo@gmail.com

# Plan

1. ~~Clustering~~
2. ~~Finding similar items~~
    a. ~~Task and motivation~~
    b. ~~Document as a set of shingles~~
    c. ~~MinHash: compressed document representation~~
    d. ~~A look at LSH~~
3. Topic modeling (~~in a fast pace~~)
    a. Task and motivation
    b. Matrix factorization as a topic model
    c. Probabilistic topic modeling
        i. pLSA
        ii. LDA
        iii. ARTM
    d. Topic modeling quality evaluation

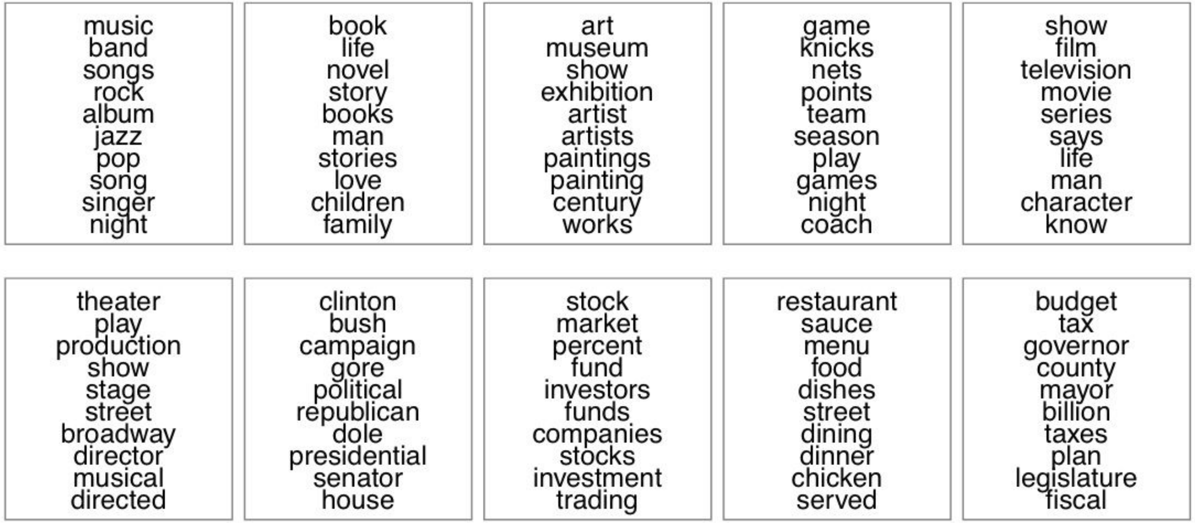
# Topic modeling [fast-paced review]

**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 model.

Taken from http://machinelearning.ru, Тематическое моделирование

# Topic modeling: example



|  |  |  |  |  |
|---|---|---|---|---|
| music | book | art | game | show |
| band | life | museum | knicks | film |
| songs | novel | show | nets | television |
| rock | story | exhibition | points | movie |
| album | books | artist | team | series |
| jazz | man | artists | season | says |
| pop | stories | paintings | play | life |
| song | love | painting | games | man |
| singer | children | century | night | character |
| night | family | works | coach | know |
| theater | clinton | stock | restaurant | budget |
| play | bush | market | sauce | tax |
| production | campaign | percent | menu | governor |
| show | gore | fund | food | county |
| stage | political | investors | dishes | mayor |
| street | republican | funds | street | billion |
| broadway | dole | companies | dining | taxes |
| director | presidential | stocks | dinner | plan |
| musical | senator | investment | chicken | legislature |
| directed | house | trading | served | fiscal |

Some of the topics found by analyzing 1.8 million articles from the New York Times. Each panel illustrates a set of tightly co-occurring terms in the collection. Hoffman, M., Blei, D. Wang, C. and Paisley, J. “Stochastic variational inference.” Journal of Machine Learning Research.

http://journalofdigitalhumanities.org/2-1/topic-modeling-and-digital-humanities-by-david-m-blei/

# 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

# Topic modeling: the task

Text collection D, each document **d** from **D** is made up of terms ($w_0$,...$w_{n\_d}$).
We suppose that every document can have one or more topics

Determine

- number of topics
- the extent to which every document satisfies each topic,
- the importance of each word for every topic

This can be treated as a task of (fuzzy) **biclustering**:
joint clustering of words and documents into the same set of topics (clusters)

**sharp guy** **fuzzy guy**

# You’ll definitely see this picture again

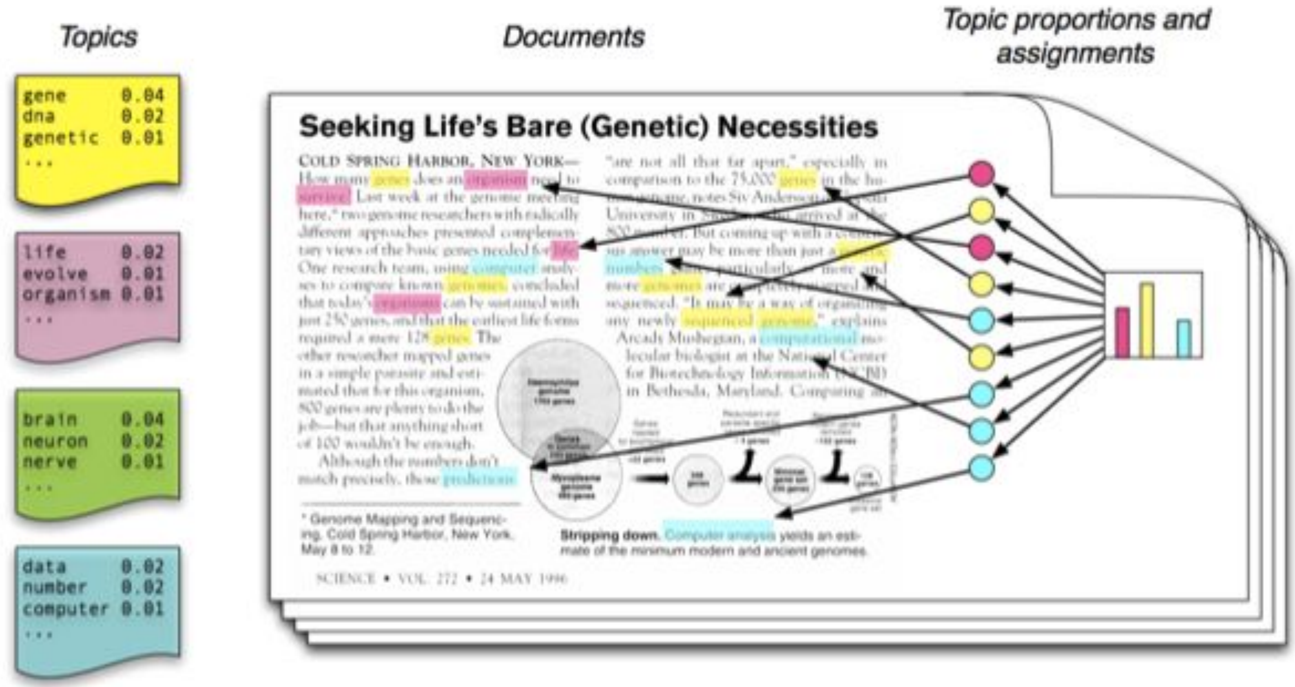
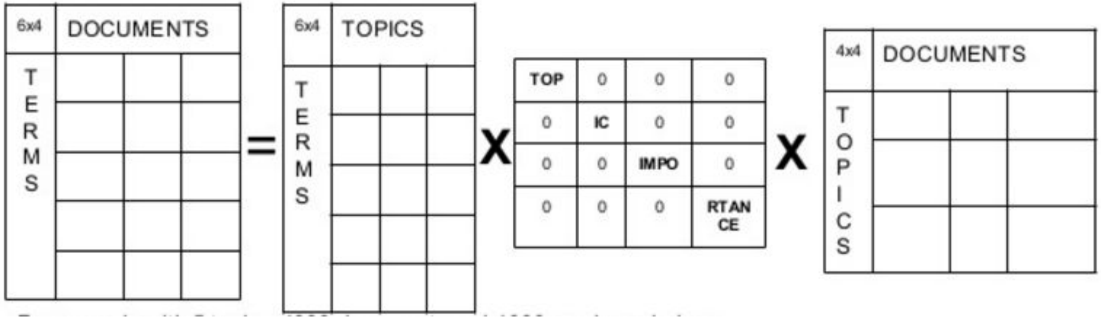

Topics
gene 0.04
dna 0.02
genetic 0.01
...
life 0.02
evolve 0.01
organism 0.01
...
brain 0.04
neuron 0.02
nerve 0.01
...
data 0.02
number 0.02
computer 0.01
...
Documents
Seeking Life’s Bare (Genetic) Necessities
SCIENCE • VOL. 272 • 24 MAY 1996
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
Topic proportions and assignments

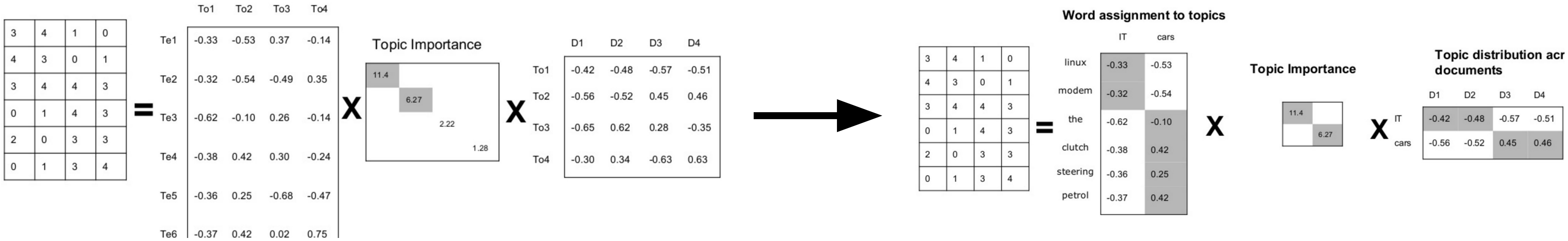
Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

# Plan

1. ~~Clustering~~
2. ~~Finding similar items~~
    a. ~~Task and motivation~~
    b. ~~Document as a set of shingles~~
    c. ~~MinHash: compressed document representation~~
    d. ~~A look at LSH~~
3. ~~Topic modeling (in a fast pace)~~
    a. ~~Task and motivation~~
    b. Matrix factorization as a topic model
    c. Probabilistic topic modeling
        i. pLSA
        ii. LDA
        iii. ARTM
    d. Topic modeling quality evaluation

# Old friend: LSA





|   |   |   |   |
|---|---|---|---|
| 3 | 4 | 1 | 0 |
| 4 | 3 | 0 | 1 |
| 3 | 4 | 4 | 3 |
| 0 | 1 | 4 | 3 |
| 2 | 0 | 3 | 3 |
| 0 | 1 | 3 | 4 |

|  | To1 | To2 | To3 | To4 |
|---|---|---|---|---|
| Te1 | -0.33 | -0.53 | 0.37 | -0.14 |
| Te2 | -0.32 | -0.54 | -0.49 | 0.35 |
| Te3 | -0.62 | -0.10 | 0.26 | -0.14 |
| Te4 | -0.38 | 0.42 | 0.30 | -0.24 |
| Te5 | -0.36 | 0.25 | -0.68 | -0.47 |
| Te6 | -0.37 | 0.42 | 0.02 | 0.75 |

Topic Importance

|   |   |   |   |
|---|---|---|---|
| 11.4 |  |  |  |
|  | 6.27 |  |  |
|  |  | 2.22 |  |
|  |  |  | 1.28 |

|  | D1 | D2 | D3 | D4 |
|---|---|---|---|---|
| To1 | -0.42 | -0.48 | -0.57 | -0.51 |
| To2 | -0.56 | -0.52 | 0.45 | 0.46 |
| To3 | -0.65 | 0.62 | 0.28 | -0.35 |
| To4 | -0.30 | 0.34 | -0.63 | 0.63 |

|   |   |   |   |
|---|---|---|---|
| 3 | 4 | 1 | 0 |
| 4 | 3 | 0 | 1 |
| 3 | 4 | 4 | 3 |
| 0 | 1 | 4 | 3 |
| 2 | 0 | 3 | 3 |
| 0 | 1 | 3 | 4 |

**Word assignment to topics**

|  | IT | cars |
|---|---|---|
| linux | -0.33 | -0.53 |
| modem | -0.32 | -0.54 |
| the | -0.62 | -0.10 |
| clutch | -0.38 | 0.42 |
| steering | -0.36 | 0.25 |
| petrol | -0.37 | 0.42 |

**Topic Importance**

|   |   |
|---|---|
| 11.4 |  |
|  | 6.27 |

**Topic distribution acr documents**

|  | D1 | D2 | D3 | D4 |
|---|---|---|---|---|
| IT | -0.42 | -0.48 | -0.57 | -0.51 |
| cars | -0.56 | -0.52 | 0.45 | 0.46 |

https://www.slideshare.net/vitomirkovanovic/topic-modeling-for-learning-analytics-researchers-lak15-tutorial

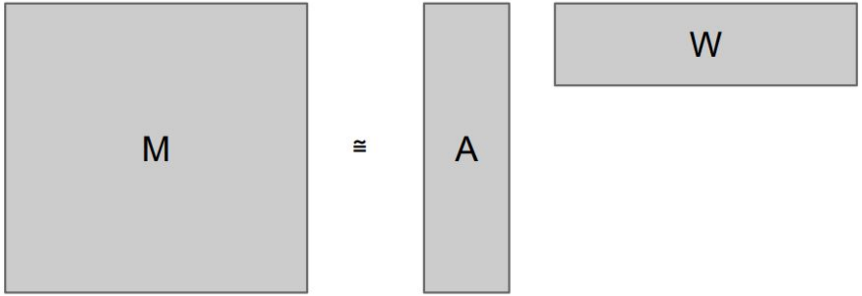
# Similar story: NMF (non-negative matrix factorization)

Approximate factorization -- a product of two matrices

- terms-topics
- topics-documents

so that values in A and W **non-negative**

- has the same computational complexity as KMeans (both are NP-hard)
- some algorithms that converge in practice are suggested (~ EM)
- It is shown in which cases (close to the real word problems) there exists a polynomial-time algorithm



https://www.cs.duke.edu/courses/fall15/compsci590.7/lecture5.pdf

Stephen A. Vavasis On the complexity of nonnegative matrix factorization https://arxiv.org/abs/0708.4149
Arora, S., Ge, R., Kannan, R., and Moitra, A. Computing a nonnegative matrix factorization – provably. In STOC, pp. 145–162, 2012a

# Plan

1. ~~Clustering~~
2. ~~Finding similar items~~
    1. ~~Task and motivation~~
    2. ~~Document as a set of shingles~~
    3. ~~MinHash: compressed document representation~~
    4. ~~A look at LSH~~
3. ~~Topic modeling (in a fast pace)~~
    1. ~~Task and motivation~~
    2. ~~Matrix factorization as a topic model~~
    3. Probabilistic topic modeling
        1. pLSA
        2. LDA
        3. ARTM
    4. Topic modeling quality evaluation

# Probabilistic topic models

Suppose that people generate texts this way, they

1) open the document
2) think about a **certain topic**
   (taking it from the preset document's topics distribution)
3) think about a certain **word**
   (taking in from the word distribution for the chosen topic)
4) write down the word
5) repeat 2-4, until they are tired or something

The goal of PTM training is to **choose such distributions parameters** that the **probability of the generation of the collection is MAX**.

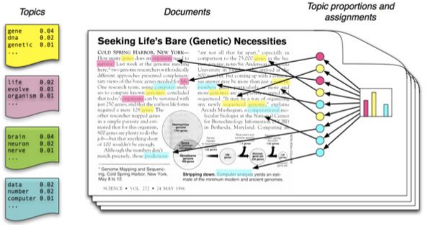


Figure source: Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77-84.

https://alessandrogirola.files.wordpress.com/2012/06/typing-monkey.jpg

# PTM: formulae

- word order is not important, the collection can be treated as a set of pairs

$$(d,w),\ d \in D,\ w \in W_d$$

- a total of **T** topics

- the topic is essentially the distribution **phi**

$$p(w|t) \longrightarrow \Phi = \|p(w|t)\|$$

- the document is essentially the distribution **theta**

$$p(t|d) \longrightarrow \Theta = \|p(t|d)\|$$

- important assumption, allowing us to infer the following

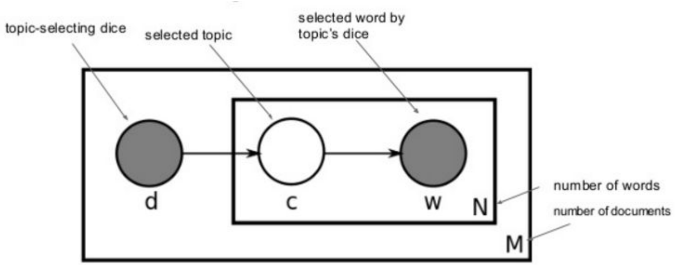
$$p(w|t,d) = p(w|t)$$

$$p(w|d) = \sum_{t \in T} p(w|d,t)\, p(t|d) = \boxed{\sum_{t \in T} p(w|t)\, p(t|d)}$$

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

# pLSA

Term-document matrix, the goal is to maximize likelihood:



$$\ln \prod_{i=1}^{n} p(d_i, w_i) = \sum_{d \in D} \sum_{w \in d} n_{dw} \ln p(w \mid d) + \sum_{d \in D} n_d \ln p(d) \to \max,$$

**at the same time**

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

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

$$\sum_{w \in W} \phi_{wt} = 1, \quad \phi_{wt} \geq 0; \qquad \sum_{t \in T} \theta_{td} = 1, \quad \theta_{td} \geq 0.$$

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '99). ACM, New York, NY, USA, 50-57.

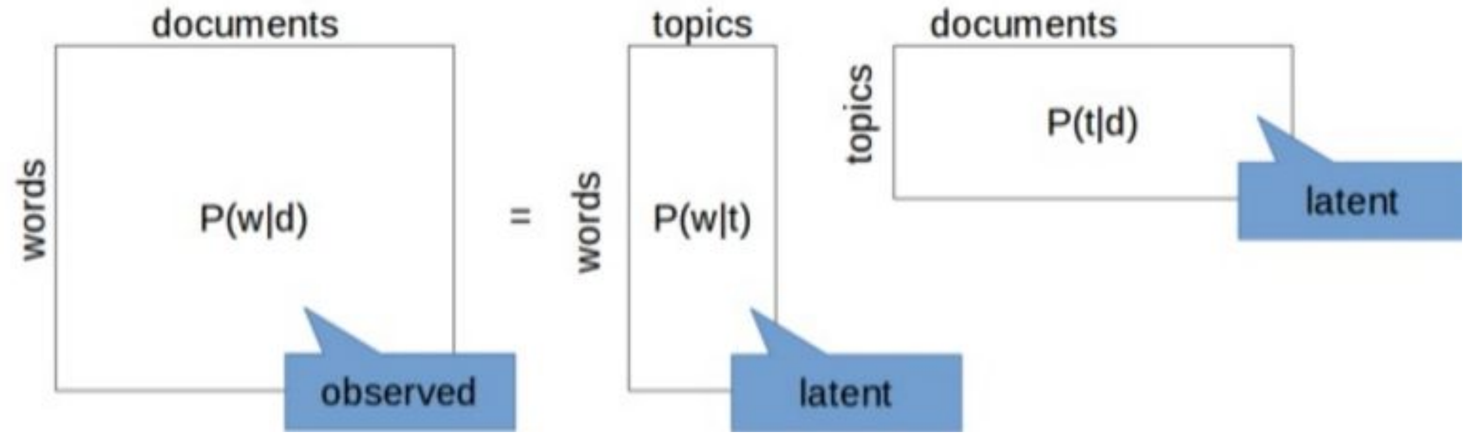
# pLSA: what it looks like

Attentive students may have noticed that the posed task reminds us of another one

# pLSA: what it looks like

NNMF of course

$$P(w|d)=\sum_t P(t|d)P(w|t)$$



# pLSA: how to train

So we want to train two matrices $\Phi, \Theta$

Until they stop changing —
**Step E**

$$n_{dwt} = n_{dw} p(t \,|\, d, w), \quad p(t \,|\, d, w) = \frac{\phi_{wt}\theta_{td}}{\sum_{s\in T} \phi_{ws}\theta_{sd}}$$

**Step M** (based on fixed n$_{dwt}$ estimated above!)

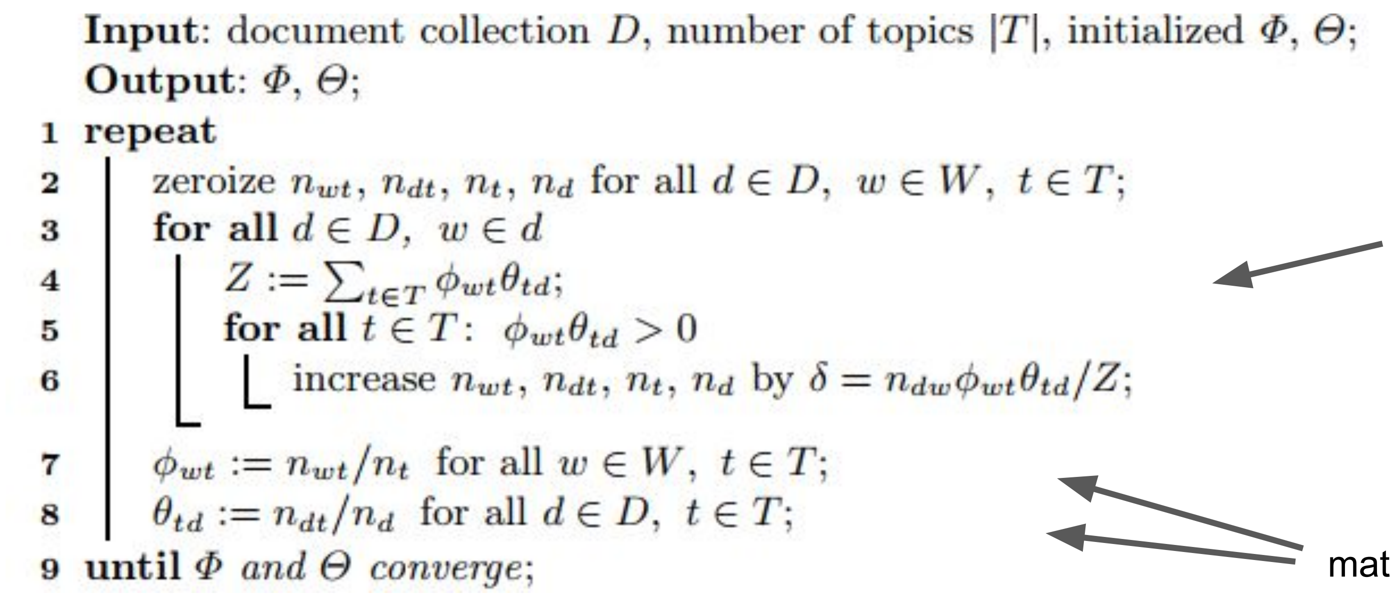
$$\phi_{wt} = \frac{n_{wt}}{n_t}, \qquad n_{wt} = \sum_{d\in D} n_{dwt}, \qquad n_t = \sum_{w\in W} n_{wt},$$

$$\theta_{td} = \frac{n_{dt}}{n_d}, \qquad n_{dt} = \sum_{w\in d} n_{dwt}, \qquad n_d = \sum_{t\in T} n_{dt},$$

http://www.machinelearning.ru/wiki/images/1/1f/Voron14aist.pdf

# pLSA: how to train

“Rational” algorithm



```
Input: document collection $D$, number of topics $|T|$, initialized $\Phi$, $\Theta$;
Output: $\Phi$, $\Theta$;
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 $\Phi$ and $\Theta$ converge;
```

$$p(t\,|\,d,w) = \frac{\phi_{wt}\theta_{td}}{\sum_{s\in T}\phi_{ws}\theta_{sd}}$$

implicit topic probabilities estimation

matrices re-estimation

http://www.machinelearning.ru/wiki/images/1/1f/Voron14aist.pdf

# pLSA: drawbacks

- stochastic matrix decomposition is an **ill-posed problem**, which means it can have an infinite number of solutions, which leads to the **instability of the ‘recovered’ matrices phi and theta**
  (this is not only pLSA’s problem, however)

- if we see a new document **d**, we can’t estimate **p(t|d)** without retraining the model

- 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)

# Plan

1. ~~Clustering~~
2. ~~Finding similar items~~
    a. ~~Task and motivation~~
    b. ~~Document as a set of shingles~~
    c. ~~MinHash: compressed document representation~~
    d. ~~A look at LSH~~
3. ~~Topic modeling (in a fast pace)~~
    a. ~~Task and motivation~~
    b. ~~Matrix factorization as a topic model~~
    c. ~~Probabilistic topic modeling~~
        i. ~~pLSA~~
        ii. LDA
        iii. ARTM
    d. Topic modeling quality evaluation

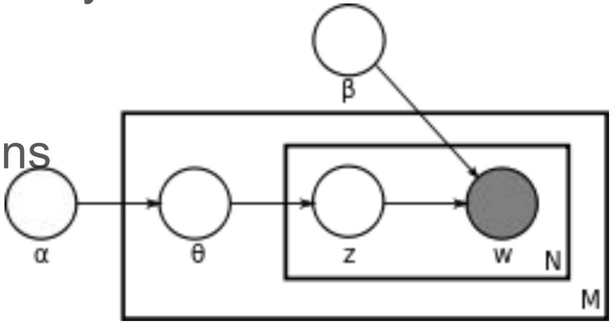
# LDA without too much detail

| CITED BY | YEAR |
|---|---|
| 20632 | 2003 |

Essentially the same model as pLSA, but we require that ‘topic vectors’ **p(w|t)** and ‘document vectors’ **p(t|d)** satisfy Dirichlet distribution

In short, the novelty of the work is in extra assumptions about the distributions of topics in documents and distributions of words in topics

- narrow down the solution space
- allow the model to work with new documents



Latent Dirichlet Allocation David M. Blei, Andrew Y. Ng, Michael I. Jordan; 3(Jan):993-1022, 2003.

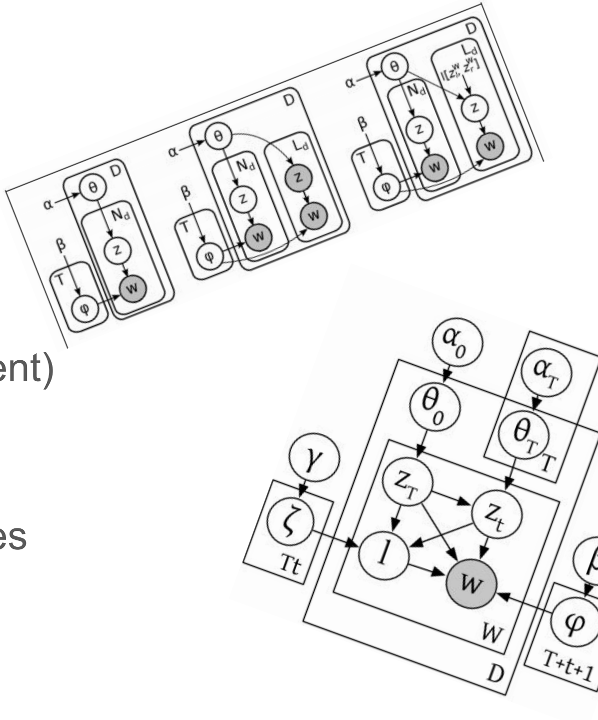
# LDA: discussion

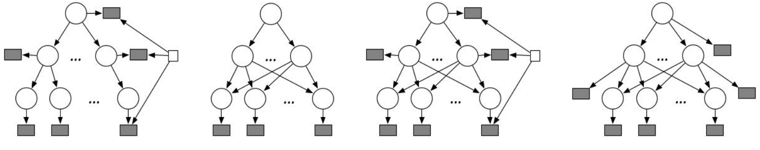
- no linguistic clues for using Dirichlet distribution

- smoothing instead of sparsification  
  (naturally, most topics are usually NOT PRESENT in the document)

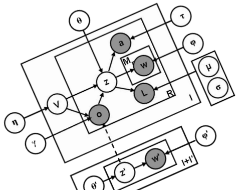
- there are numerous LDA extensions for taking into account extra constraints and for solving other tasks; however, most of the times their preparation is a complex mathematical task

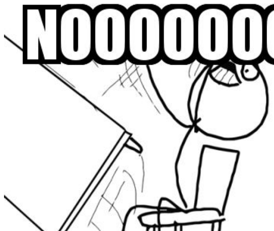
- if dataset is large enough, there is not much difference between LDA and pLSA





(a) hLDA (b) PAM (c) hPAM 1 (d) hPAM 2





# Plan

1. ~~Clustering~~
2. ~~Finding similar items~~
    a. ~~Task and motivation~~
    b. ~~Document as a set of shingles~~
    c. ~~MinHash: compressed document representation~~
    d. ~~A look at LSH~~
3. ~~Topic modeling (in a fast pace)~~
    a. ~~Task and motivation~~
    b. ~~Matrix factorization as a topic model~~
    c. ~~Probabilistic topic modeling~~
        i. ~~pLSA~~
        ii. ~~LDA~~
        iii. ARTM
    d. Topic modeling quality evaluation

# ARTM: Additive Regularized Topic Models

**Regularization** -- introduction of extra constraints on the model with the aim of narrowing down the solution space or as a way to make the model less prone to the possible overfitting

We could add extra summands to **pLSA** (which is an ill-posed problem!) at the training stage

$$R(\Phi, \Theta) = \sum_{i=1}^{r} \tau_i R_i(\Phi, \Theta), \qquad L(\Phi, \Theta) + R(\Phi, \Theta) \rightarrow \max_{\Phi, \Theta}$$

**R** should be continuously differentiable

One can use **R** for forcing the desired properties that **phi** and **theta** should hopefully have; algorithms, recipes, theorems are in the tutorial

Tutorial on Probabilistic Topic Modeling: Additive Regularization for Stochastic Matrix Factorization Konstantin Vorontsov and Anna Potapenko

# ARTM: regularizer example

If we need the distributions to be like that

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

we can set the **smoothing regularizer**

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

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

# ARTM: regularizer example

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 between 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**

# 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

chad topic modeler
prefers ARTM

# Plan

1. ~~Clustering~~
2. ~~Finding similar items~~
    a. ~~Task and motivation~~
    b. ~~Document as a set of shingles~~
    c. ~~MinHash: compressed document representation~~
    d. ~~A look at LSH~~
3. ~~Topic modeling (in a fast pace)~~
    a. ~~Task and motivation~~
    b. ~~Matrix factorization as a topic model~~
    c. ~~Probabilistic topic modeling~~
        i. ~~pLSA~~
        ii. ~~LDA~~
        iii. ~~ARTM~~
    d. Topic modeling quality evaluation

# Topic models evaluation

**Intrinsic evaluation. Method 1: perplexity**

This time the model of language is a word distribution

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

If it is uniform, then it is equal to number of words (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 models evaluation

**Intrinsic evaluation. Method 2**

Can the experts tag the topic with a title given its ‘top words’?

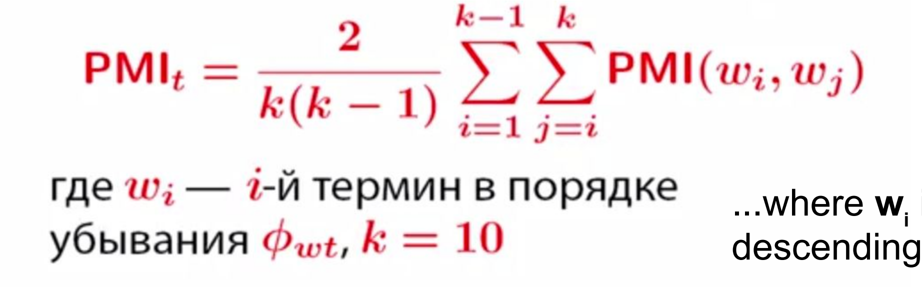
**Intrinsic evaluation. Method 2’**

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



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

где $w_i$ — $i$-й термин в порядке убывания $\phi_{wt}$, $k = 10$

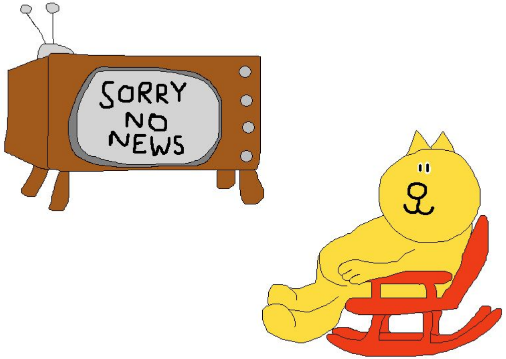
...where **w**$_i$ is the i-th term in the descending order of phi-s, k = 10

# Topic models evaluation

**Extrinsic evaluation**

we can use topic modeling for solving other tasks

For example, ranking and classification — and then compute quality evaluation metrics for those



http://www.jupiterjenkins.com/stress-printer-and-treadmill/

# Also see

- 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)

- **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 with arbitrary number of topics”

  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) and other neural approaches

  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.

# Tools

1. **Gensim**
   (LSI, LDA, visualization tools)
2. **BigARTM**, github
   (ARTM with a few prepared regularizers, can be extended)
3. **Mallet** (Java / CLI)
4. Other

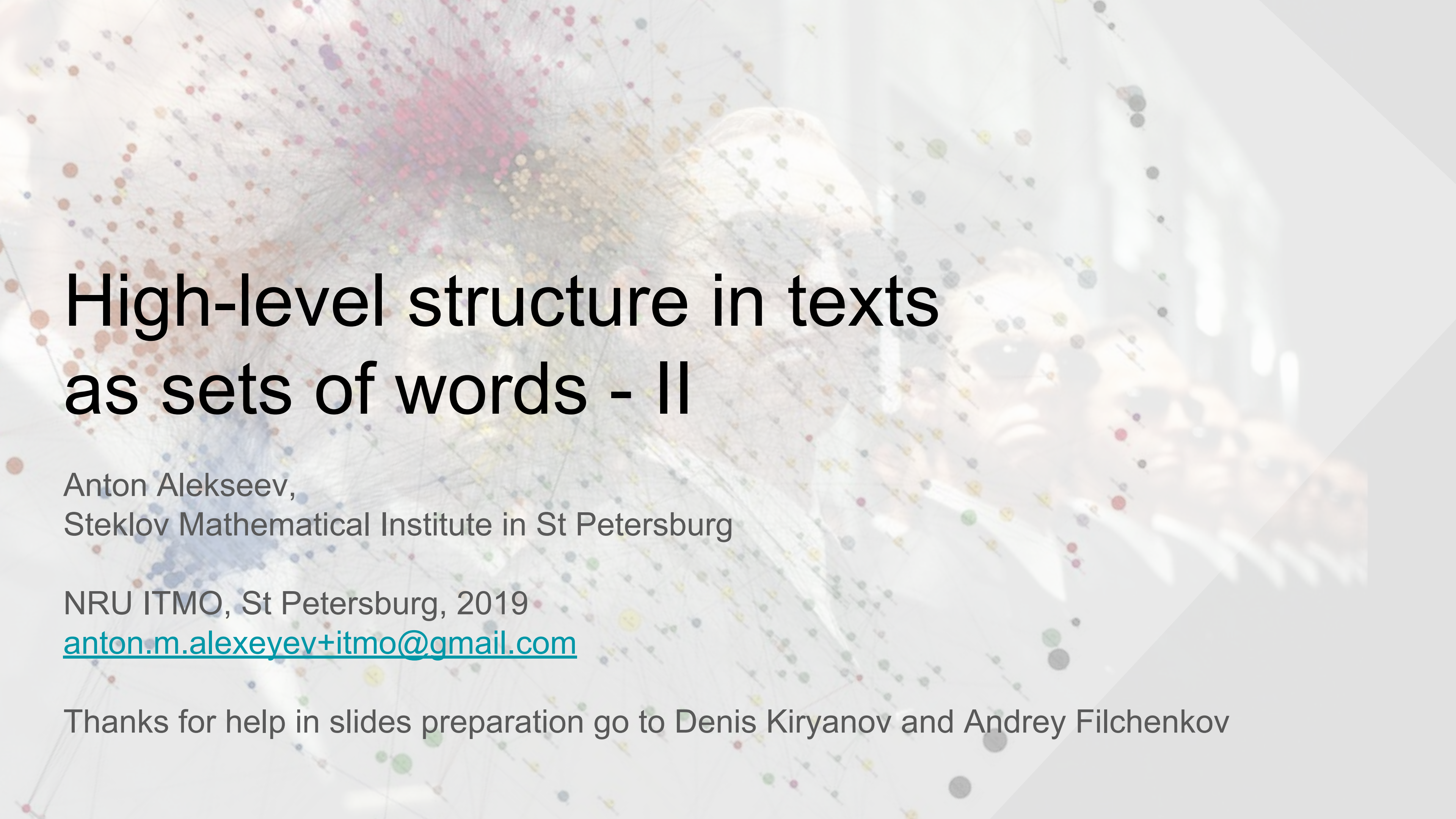
gensim
topic modelling for humans

BigARTM

MALLET
machine learning for language toolkit

# TM: used/recommended materials

1. Hanna Wallach, NIPS2009 tutorial
2. Course by Rong Ge в Duke University: Algorithimic Aspects of Machine Learning, монография Ankur Moitra
3. K.Vorontsov Additive regularization of topic models
4. Articles and tutorials on slides
5. Wikipedia
6. [Russian] Обзор К.В.Воронцова (может обновляться! см.), доклад об ARTM



# High-level structure in texts as sets of words - II

Anton Alekseev,
Steklov Mathematical Institute in St Petersburg

NRU ITMO, St Petersburg, 2019
anton.m.alexeyev+itmo@gmail.com

Thanks for help in slides preparation go to Denis Kiryanov and Andrey Filchenkov