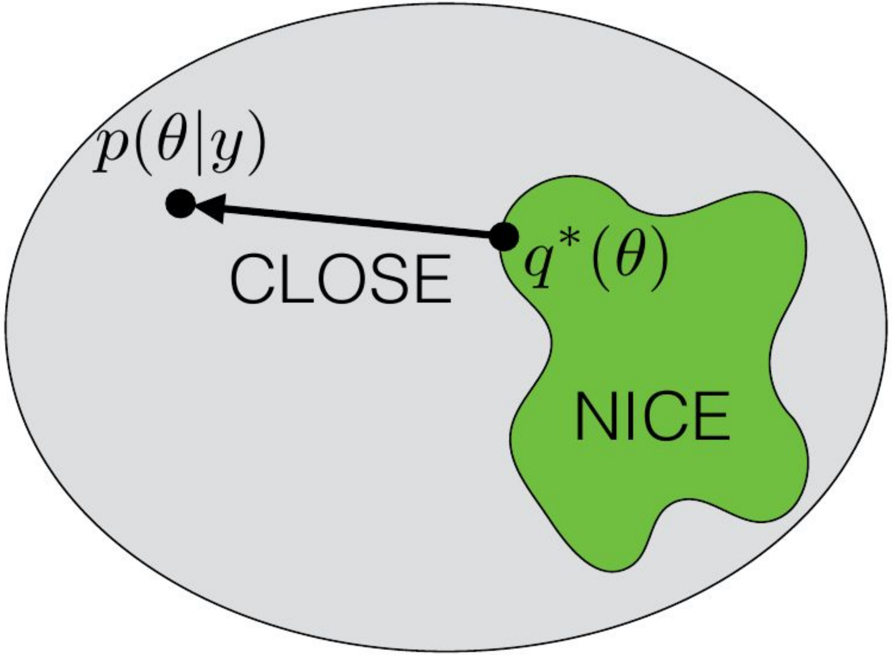
Example of Probabilistic Graphical Models - LDA

Tomasz Kajdanowicz

The presentation in based on the
D. Blei, A. Ng, M. Jordan, Latent Dirichlet Allocation, Journal of Machine Learning Research 3 (2003) 993-1022
and
Lettier, Your Guide to Latent Dirichlet Allocation, Medium

Approximate Bayesian Inference



Instead: an optimization approach

- Approximate posterior with q^*

$$q^* = \text{argmin}_{q \in Q} f(q(\cdot), p(\cdot|y))$$

- Variational Bayes (VB): *f* is Kullback-Leibler divergence

$$KL(q(\cdot)||p(\cdot|y))$$

q^* - what is its form?

- Selection of exponential distributions
- Mean-field variational Bayes

$$Q_{MFVB} := \left\{ q : q(\theta) = \prod_{j=1}^{J} q_j(\theta_j) \right\}$$

Approximate Bayesian inference

Use q^* to approximate $p(\cdot|y)$

Optimization

$$q^* = \text{argmin}_{q \in Q} f(q(\cdot), p(\cdot|y))$$

Variational Bayes

$$q^* = \text{argmin}_{q \in Q} KL(q(\cdot)||p(\cdot|y))$$

Mean-field variational Bayes

$$q^* = \text{argmin}_{q \in Q_{\text{MFVB}}} KL(q(\cdot)||p(\cdot|y))$$

- Coordinate descent
- Stochastic variational inference (SVI) [Hoffman et al 2013]
- Automatic differentiation variational inference (ADVI) [Kucukelbir et al 2015, 2017]

Latent Dirichlet Allocation (LDA)

- generative probabilistic model
- topic modelling
- the *composites*: documents, the *parts*: words and/or phrases
- Possible application:
  - DNA and nucleotides,
  - pizzas and toppings,
  - molecules and atoms,
  - employees and skills

How does the model look like?

The probabilistic topic model estimated by LDA consists of:

- a table that describes the probability or chance of selecting a particular **word** when sampling a particular **topic**
- a table that describes the chance of selecting a particular **topic** when sampling a particular **document** or composite

	Topic 0	Topic 1	Topic 2
✳	0.000	1.000	0.000
🎩	0.000	0.000	0.559
😺	1.000	0.000	0.441
	Topic 0	Topic 1	Topic 2

	Topic 0	Topic 1	Topic 2
Document 0	0.486	0.116	0.399
Document 1	0.094	0.638	0.268
Document 2	0.377	0.616	0.007
Document 3	0.007	0.899	0.094
	Topic 0	Topic 1	Topic 2

Demo

https://lettier.com/projects/lda-topic-modeling/

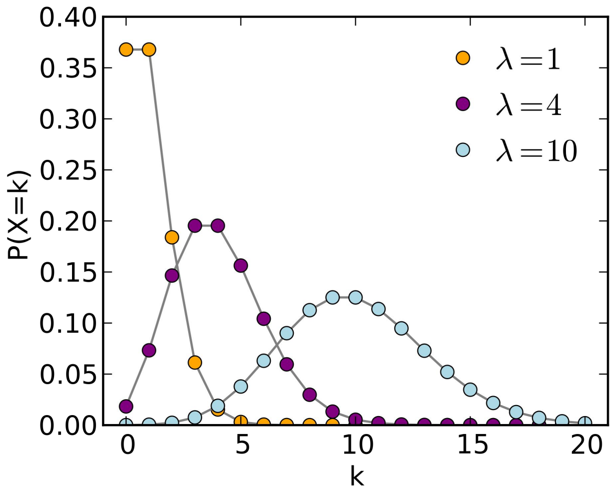
LDA generative procedure

1. Pick:
   a. your unique set of **words**
   b. how many **documents**
   c. how many **words** per **document** (sample from a Poisson distribution) (*N*)
   d. how many **topics**
2. Set: $\alpha \in (0, \infty)$, $\beta \in (0, \infty)$
3. Build:
   a. **‘words** vs. **topics’** table (sample from Dirichlet distribution using Beta as the input)
   b. **‘documents** vs. **topics’** table (sample from Dirichlet distribution using □ as the input
   c. documents
      i. sample a **topic** based on the probabilities for particular **document**
      ii. sample a **word** based on the probabilities for the **topic** sampled
      iii. repeat until you’ve reached how many **words** this **document** was set to have

Poisson distribution

- expresses the probability of a given number of events occurring in a fixed interval of time or space

$$P(k \text{ events in interval}) = e^{-\lambda} \frac{\lambda^k}{k!}$$



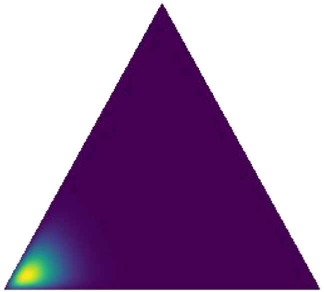
k - the number of occurrences
λ - expected number of occurrences

Dirichlet distribution

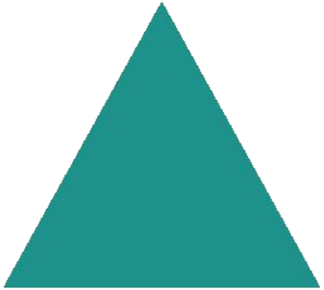
- K - way categorical events
- α - number of observed outcomes
- multivariate generalization of the Beta distribution

$$Dir(x|\alpha) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{k=1}^{K} x_k^{\alpha_k - 1}$$

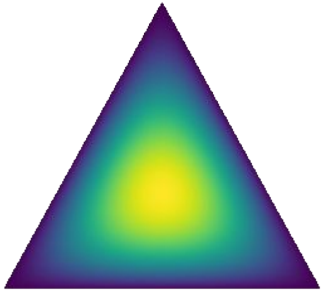
$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)}, \qquad \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K).$$



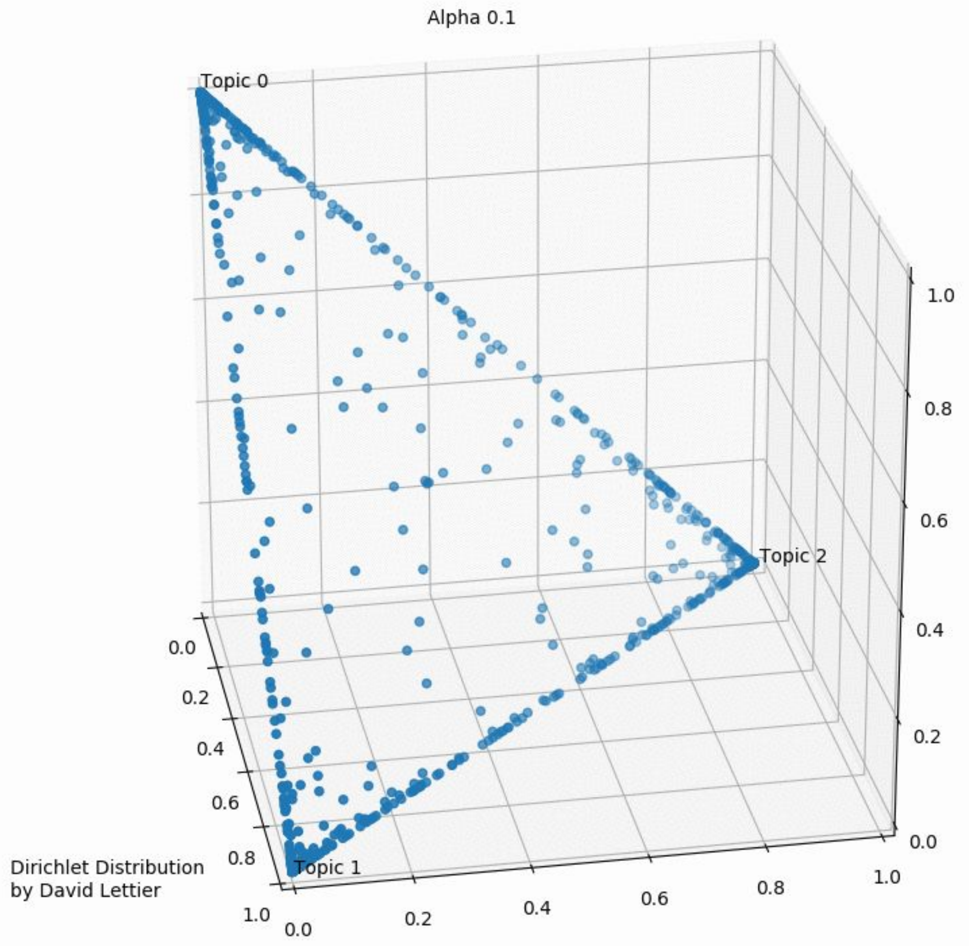
Dir(20,2,2)



Dir(1,1,1)



Dir(2,2,2)



Alpha 0.1
Topic 0
Topic 1
Topic 2
0.0
0.2
0.4
0.6
0.8
1.0
Dirichlet Distribution
by David Lettier

Parameters of LDA

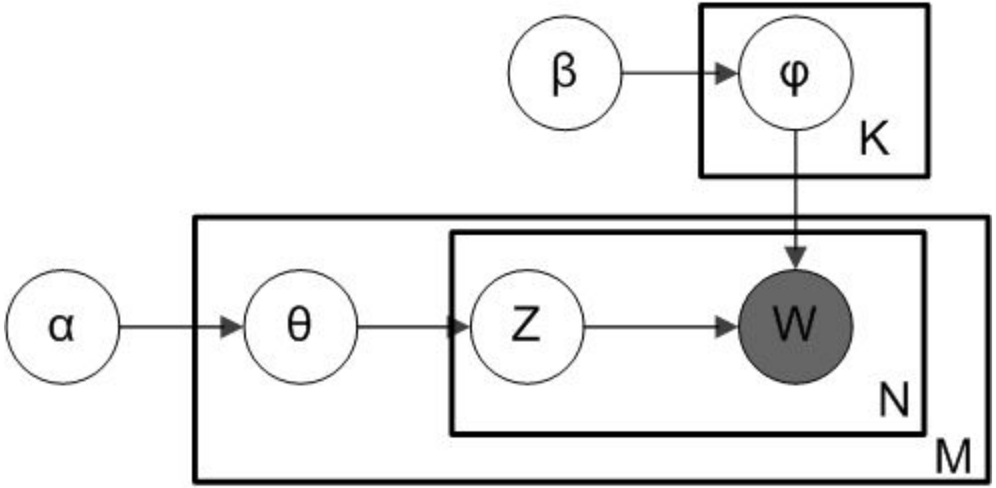
- α - controls the mixture of topics for any given document
- β - controls the distribution of words per topic
- both typically set below one:
    - we want our documents to be made up of only a few topics
    - words should belong to only some of the topics

Why to use LDA?

- soft-clustering of documents and words
  - number of topics as a number of clusters, probabilities are the proportion of cluster membership
- reducing the dimensionality
  - the number of topics to be less than the documents
- in general: uncover the themes in the data

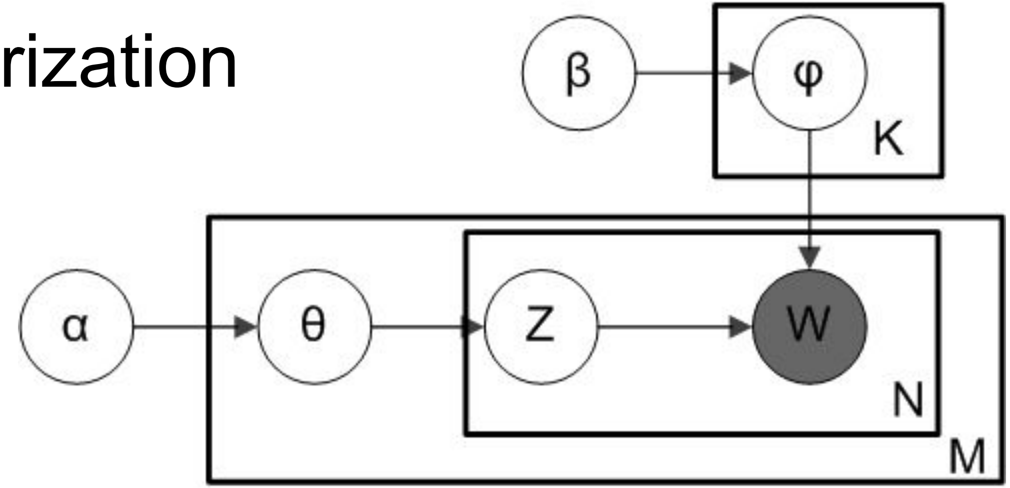
Graphical model representation of LDA

M - documents
N - words in a document
K - number if topics
φ - mixture of words
θ - mixture of topics
W - words
Z - topics



Params:
α - controls the mixture of topics
β - controls the distribution of words per topic

Graphical model factorization



M - documents
N - words in a document
K - number if topics
φ - mixture of words
θ - mixture of topics
W - words
Z - topics

$$P(\boldsymbol{W},\boldsymbol{Z},\boldsymbol{\theta},\boldsymbol{\varphi};\alpha,\beta)=\prod_{i=1}^{K}P(\varphi_i;\beta)\prod_{j=1}^{M}P(\theta_j;\alpha)\prod_{t=1}^{N}P(Z_{j,t}\mid\theta_j)P(W_{j,t}\mid\varphi_{Z_{j,t}})$$

Params:
α - controls the mixture of topics
β - controls the distribution of words per topic

Probability of document

- Integrating over θ and summing over z, we obtain the **marginal distribution of a document**

$$p(\mathbf{w}\,|\,\alpha,\beta) = \int p(\theta\,|\,\alpha)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n\,|\,\theta)p(w_n\,|\,z_n,\beta)\right)d\theta$$

- taking the product of the marginal probabilities of single documents, we obtain the p**robability of a corpus *D*** (set of documents)

$$p(D\,|\,\alpha,\beta) = \prod_{d=1}^{M}\int p(\theta_d\,|\,\alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{dn}} p(z_{dn}\,|\,\theta_d)p(w_{dn}\,|\,z_{dn},\beta)\right)d\theta_d$$

Estimation

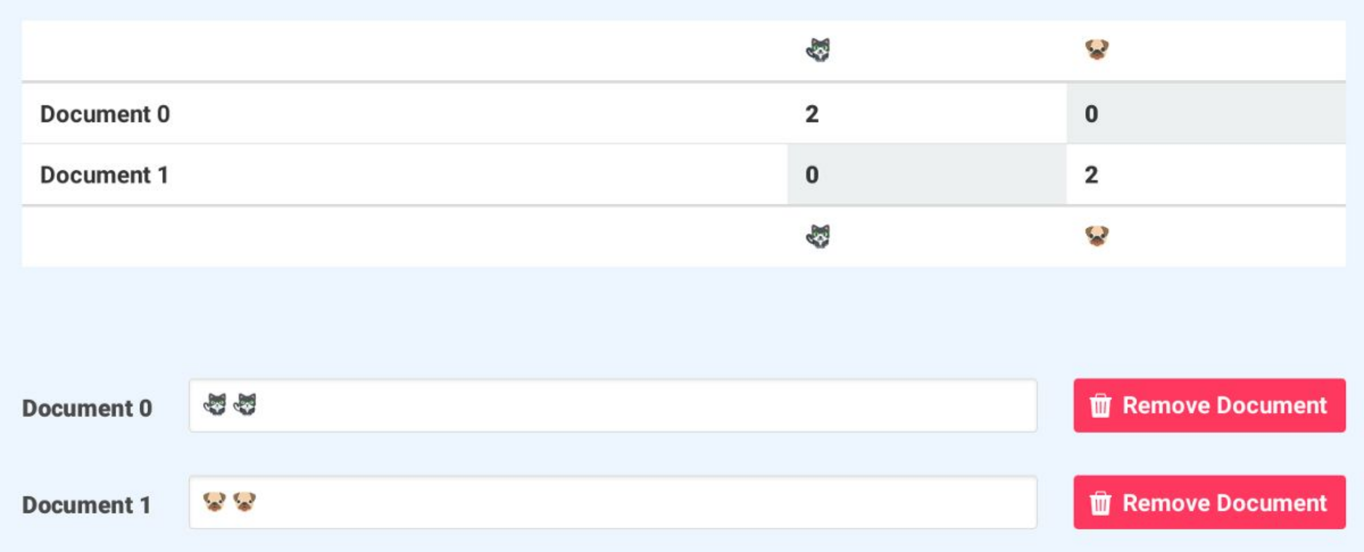
Classically: Gibbs sampling - estimate the topic assignments for each of words

Algorithm 1 Gibbs sampler

Initialize $x^{(0)} \sim q(x)$
for iteration $i = 1, 2, \ldots$ **do**
$x_1^{(i)} \sim p(X_1 = x_1 | X_2 = x_2^{(i-1)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
$x_2^{(i)} \sim p(X_2 = x_2 | X_1 = x_1^{(i)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
$\vdots$
$x_D^{(i)} \sim p(X_D = x_D | X_1 = x_1^{(i)}, X_2 = x_2^{(i)}, \ldots, X_D = x_{D-1}^{(i)})$
end for

Working example

- alpha = 0.5
- beta = 0.01
- ‘topics’ = 2
- ‘iterations’ = 1



sampling from a uniform distribution

randomly assign:

to the first cat emoji ‘Topic 0’,
the second cat ‘Topic 1’,
the first dog emoji ‘Topic 1’,
the second dog ‘Topic 0’

current topic assignment per each emoji:

	Cat 0	Cat 1	Dog 0	Dog 1
Topic 0	*			*
Topic 1		*	*	

Working example

current emoji versus topic counts

	Topic 0	Topic 1
Cat	1	1
Dog	1	1

current document versus topic counts

	Topic 0	Topic 1
Document 0	1	1
Document 1	1	1

Working example

Now update the topic assignment for the first cat:

- subtract one from the emoji versus topic counts for Cat 0
- subtract one from the document versus topic counts for Cat 0
- calculate the probability of Topic 0 and 1 for Cat 0
- flip a biased coin (sample from a categorical distribution) and update the assignment and counts

```
t0 = ((cat emoji with Topic 0 + beta) / (emoji with Topic 0 + unique emoji * beta))
*
((emoji in Document 0 with Topic 0 + alpha) / (emoji in Document 0 with a topic +
number of topics * alpha)) =

((0 + 0.01) / (1 + 2 * 0.01)) * ((0 + 0.5) / (1 + 2 * 0.5)) = 0.0024509803921568627


t1 = ((1 + 0.01) / (2 + 2 * 0.01)) * ((1 + 0.5) / (1 + 2 * 0.5)) = 0.375

p(Cat 0 = Topic 0 | *) = t0 / (t0 + t1) = 0.006493506493506494

p(Cat 0 = Topic 1 | *) = t1 / (t0 + t1) = 0.9935064935064936
```

Now do the same for Cat 1, Dog 0 and Dog 1

for each row-column cell in the ‘emoji-versus-topic’ count matrix

```
Phi row column = (emoji row with topic column + beta) /
                   (all emoji with topic column + unique emoji * beta)
```

for each row-column cell in the document versus topic count matrix

```
Theta row column =  (emoji in document row with topic column + alpha) /
 (emoji in document row + number of topics * alpha)
```

	Topic 0	Topic 1
	0.995	0.005
	0.005	0.995
	Topic 0	Topic 1

	Topic 0	Topic 1
Document 0	0.833	0.167
Document 1	0.167	0.833
	Topic 0	Topic 1