Topic Extraction and Categorization using LDA

Clint P. George

Overview

- Feature extraction from the Web
- Term Frequency – Inverse Document Frequency (TF-IDF)
- Latent Semantic Indexing (LSI)
- Unigram, Mixture of Unigrams, and Probabilistic LSI
- Latent Dirichlet Allocation (LDA)
  - Graphical view
  - Geometrical Interpretation
- LDA for dimensionality reduction
- LDA for document classification

Feature Extraction

Extract tokens or terms from the document text and represent them in a machine readable format

Methods:

- Tokenization
- Lemmatization (e.g. appeared → appear)
  - Different forms of a term into a common form
- Stemming (e.g. walking to walk)
  - Remove grammatical markings
- Removing noise from the documents
- Represent terms and documents to vector space models, e.g. $d_i = \{w_j\}_{j=1,\ldots,N}$
- Features are term-frequencies in a document

TF-IDF

- Reduces each document into a vector of TF-IDF values
- $tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$
  - n_{ij} represents the number of occurrences of a term i in the document j
  - $\sum_k n_{kj}$ is the sum of number of occurrences of all terms in document; to prevent a bias towards longer documents
- $idf_i = \ln \frac{|D|}{|\{d: t_i € d\}|}$
  - |D| represents total number of docs in a corpus
  - Denominator represents number of documents where the term t_i appears

Reference: Wikipedia

TF-IDF

- $\boldsymbol{(TF - IDF)_{ij} = tf_{ij} \; * \; idf_i}$
  - Results in a $V \; X \; D$ matrix of real numbers
  - V – vocabulary size
- Helps to remove **common terms** in a corpus
- Limitations
  - No dimensionality reduction
  - Reveals a little about the inter-or intra-document statistical structure

Reference: Wikipedia

Latent Semantic Indexing (LSI)

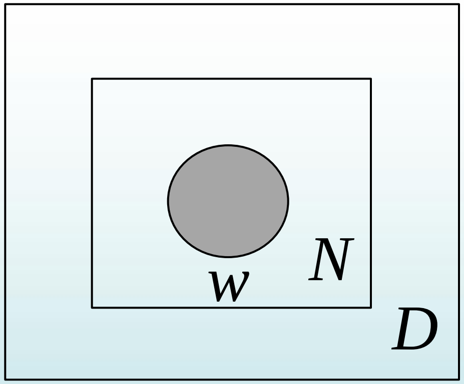
- Assume we have a $\mathrm{V\ X\ D}$ matrix X of TF-IDF values
- LSI does ***singular value decomposition*** of X

$$X = U\,\Sigma\,S^T$$

  - U is a $V\ X\ V$ unitary matrix; *Eigen vectors* of X^TX
  - $\sum$ is a $V\ X\ D$ diagonal matrix w/ non-negative real values, called as **singular values**
  - S is a $D\ X\ D$ unitary matrix; *Eigen vectors* of XX^T

- Identifies a linear subspace in the space of TF-IDF features
- Achieves significant compression in large collections

Unigram Model

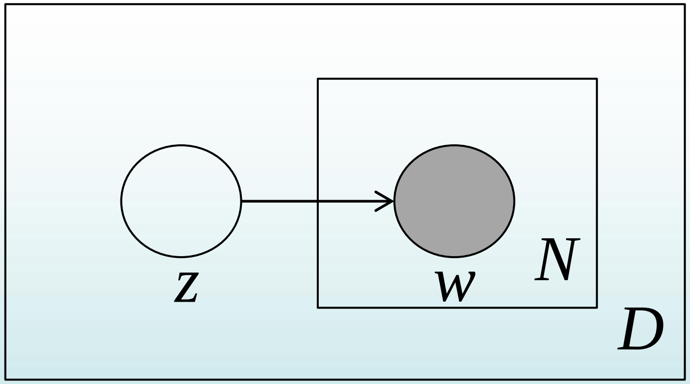


The generative process:

- Words in a document is considered as an outcome of independent multinomial draws
- Thus, the probability of generating a document $\boldsymbol{d}$ is:

$$\boldsymbol{P}\left(\boldsymbol{w}_{\mathbf{1}} \ldots \boldsymbol{w}_{\boldsymbol{N}_{\boldsymbol{d}}}\right) = \prod_{\boldsymbol{n}=\mathbf{1}}^{\boldsymbol{N}_{\boldsymbol{d}}} \boldsymbol{p}(\boldsymbol{w}_{\boldsymbol{n}})$$

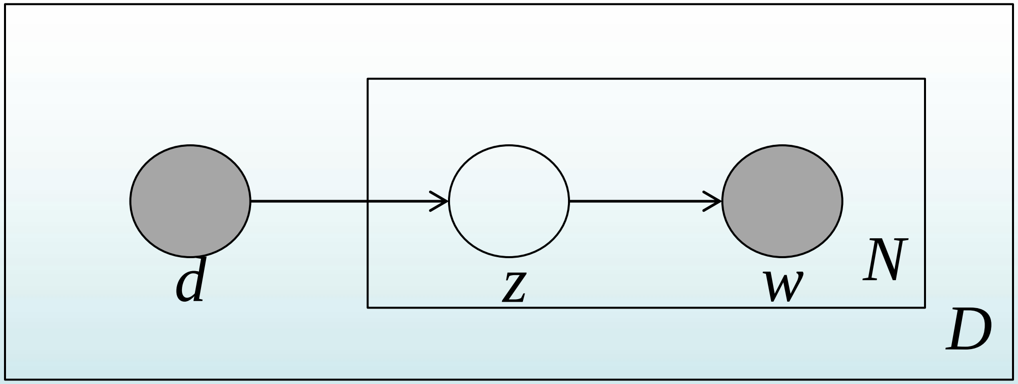
Mixture of Unigrams



- The <u>generative process</u>:
  - Pick a *hidden topic* z with probability $\boldsymbol{P(z_d)}$ for each document $\boldsymbol{d}$
  - Generate each word $\boldsymbol{w}_{dn}$ in a document with probability $\boldsymbol{P(w_{dn} \mid z_d)}$
- Thus, the probability of generating a document $\boldsymbol{d}$ is:

$$\boldsymbol{P\left(w_1 \dots w_{N_d}\right) = \sum_{z_d} p(z_d) \prod_{n=1}^{N_d} p(w_{dn}|z_d)}$$

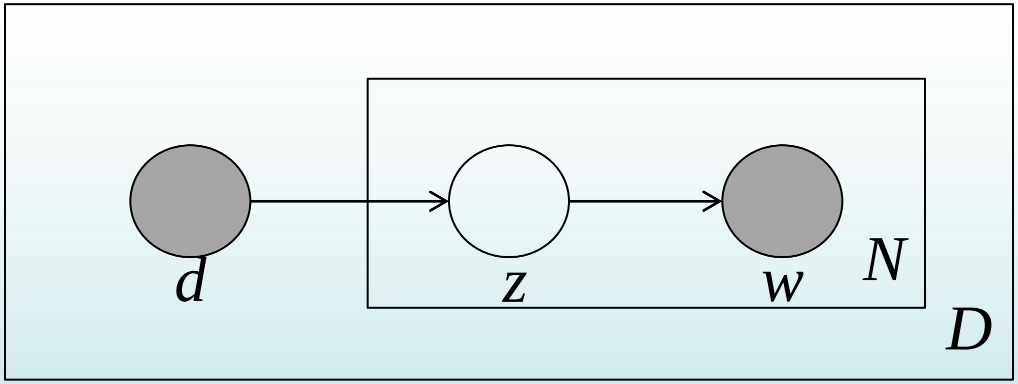
Probabilistic LSI



- The generative process:
  - Pick a *topic mixture* for each document $\boldsymbol{d}$
  - Pick a *hidden topic* z_n with probability $\boldsymbol{P(z_n|d)}$ for each term $\boldsymbol{w_n}$
  - Generate a word $\boldsymbol{w_n}$ with probability $\boldsymbol{P(w_n|z_n)}$
- Thus, the probability of generating a document $\boldsymbol{d}$ is:

$$\boldsymbol{P\left(w_1 \dots w_{N_d}\right) = \prod_{n=1}^{N_d} \sum_{i=1}^{K} P(w_n \mid z_{ni})\, P(z_{ni} \mid d)}$$

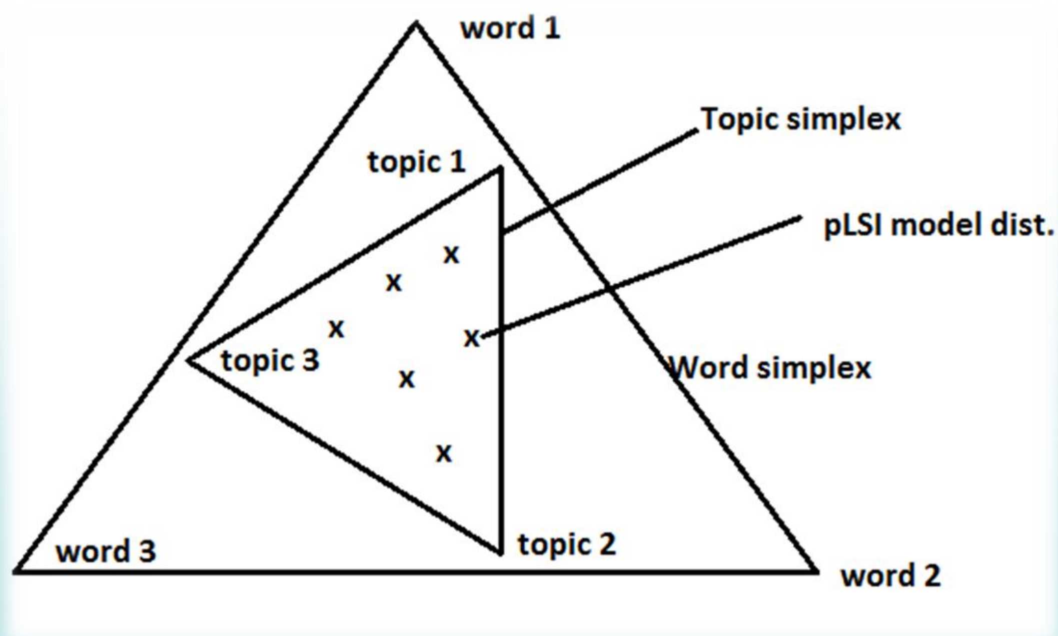
Probabilistic LSI



$$P\left(d, w_1 \ldots w_{N_d}\right) = p(d) \prod_{i=1}^{N_d} \sum_{z=1}^{K} P(w_i|z)P(z|d)$$

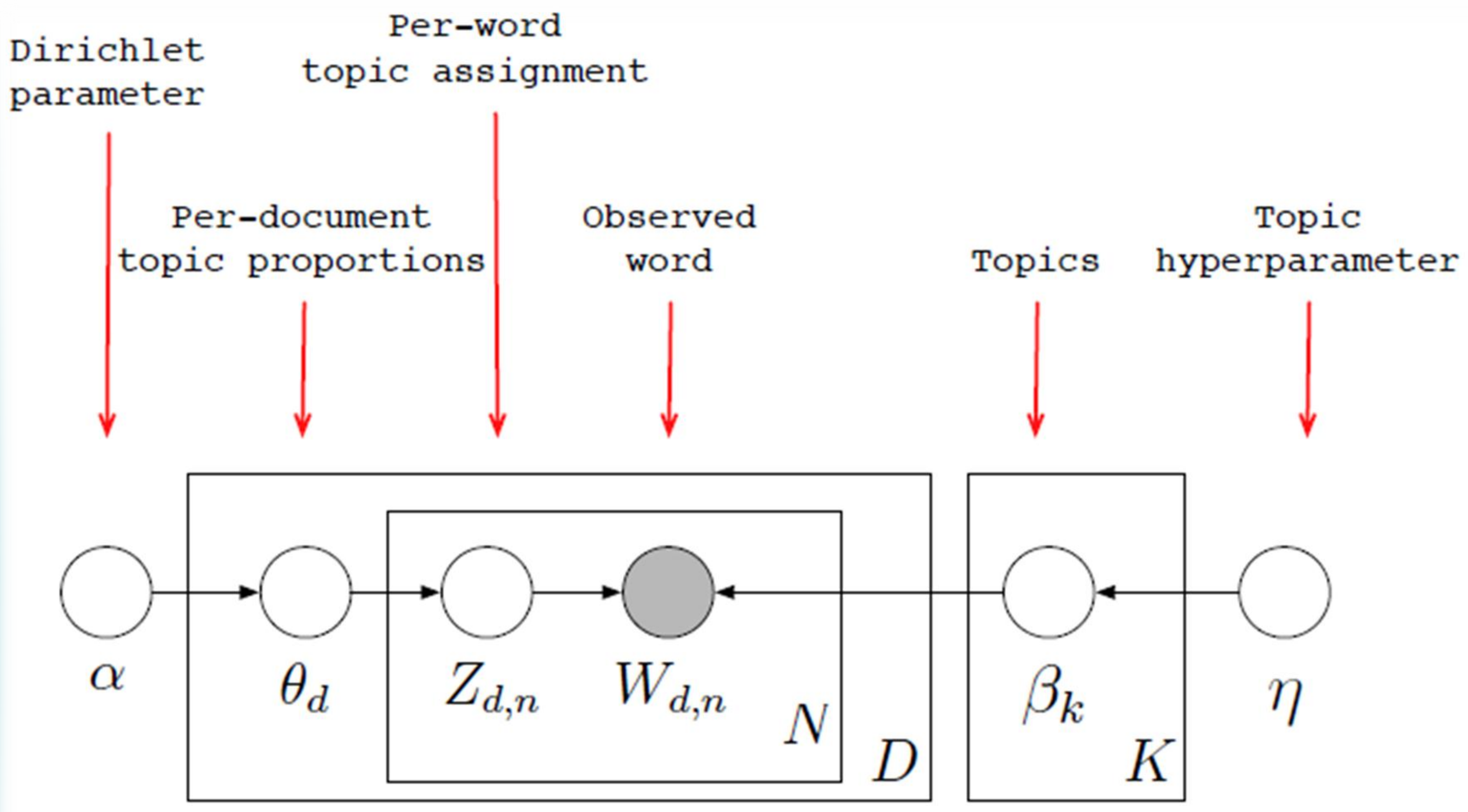
- Limitations
  - No probabilistic model at the level of documents
  - Number of parameters of the model increases with the size of the corpus

pLSI – Geometric Interpretation



- The corners of the **word simplex** represent three distributions, where **each word** has probability one
- The corners of the **topic simplex** represent three distributions, where **each topic** has probability one

Latent Dirichlet Allocation - review



Reference: Slides on LDA by David Blei

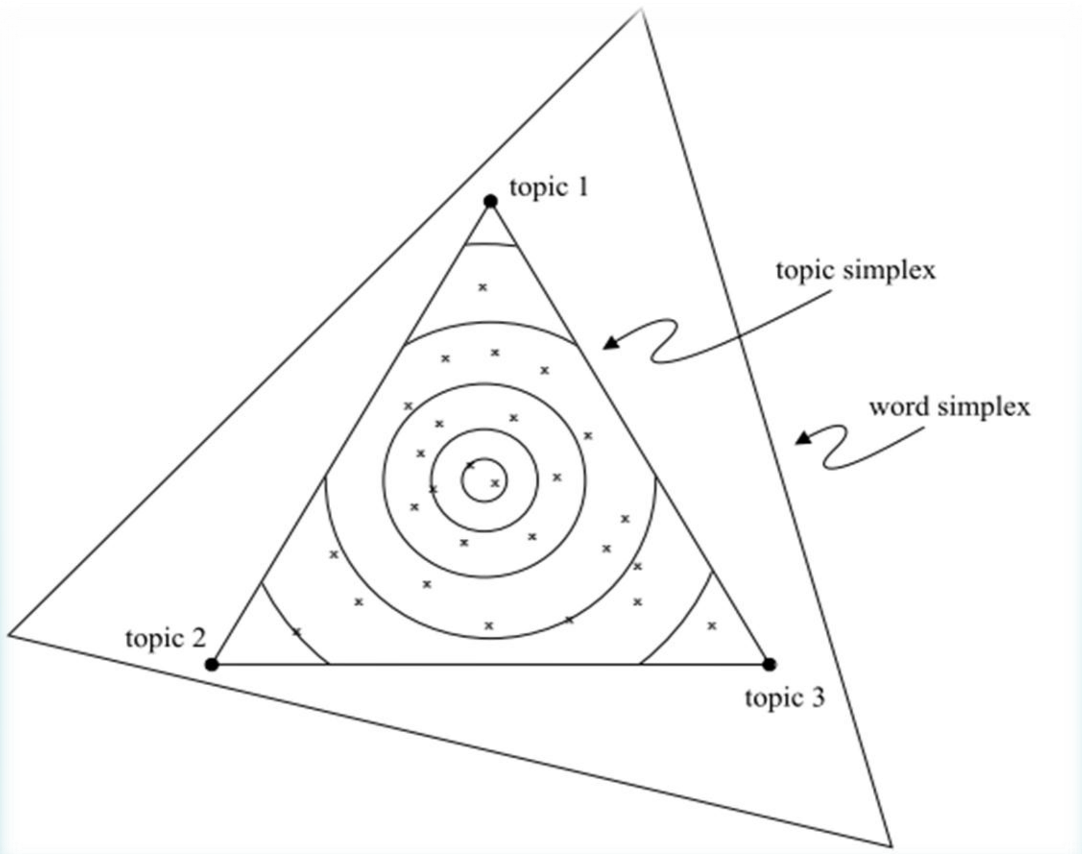
Latent Dirichlet Allocation - review

The generative process:

- Choose Θ from a Dirichlet distribution $Dir(\alpha)$
- Chose β from a Dirichlet distribution $Dir(\eta)$
- For each of the N_d words w_n in a document d
  - Choose a latent topic z_n from $Multinomial\ (\Theta)$
  - Choose a words w_n from a multinomial $p(w_n|z_n, \beta)$, conditioned on z_n
- *Thus, per-document* $posterior$ *is*

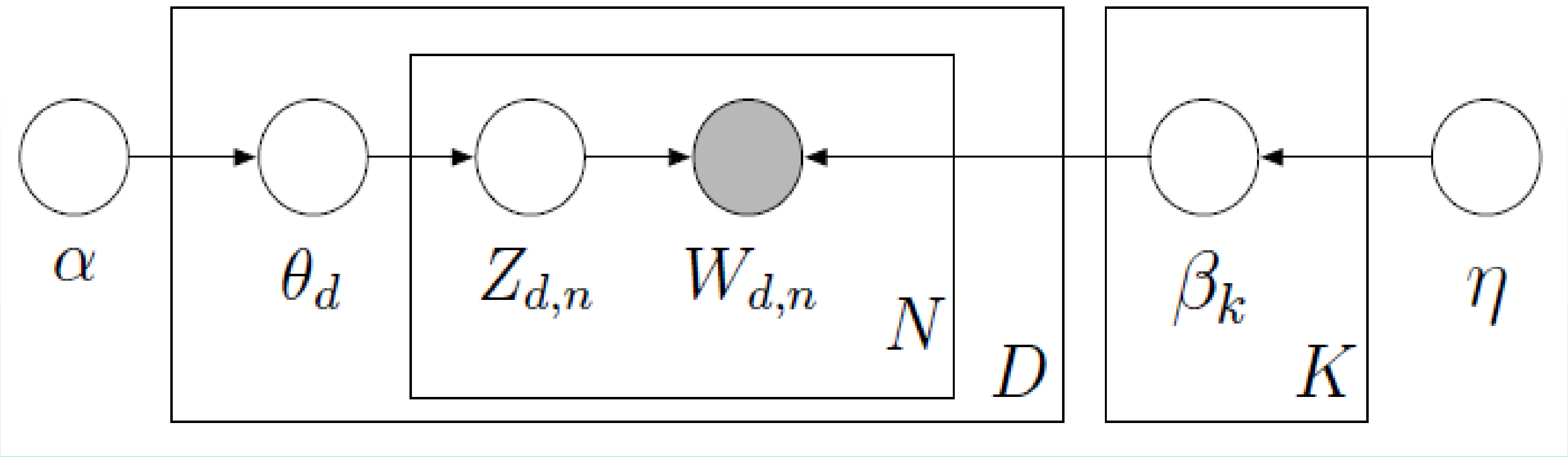
$$\frac{p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta_{1:K})}{\int_\theta p(\theta \mid \alpha) \prod_{n=1}^{N} \sum_{z=1}^{K} p(z_n \mid \theta) p(w_n \mid z_n, \beta_{1:K})}$$

LDA Geometric Interpretation



- Each word is generated by a randomly chosen topic which is drawn from **a smooth distribution** with a randomly chosen parameter (ά)

LDA Inference – review



LDA Inference gives

- per-word topic assignment (Z_{dn})
- per-document topic proportions (θ), K X D matrix
- per-corpus topic distributions (β), K X V matrix

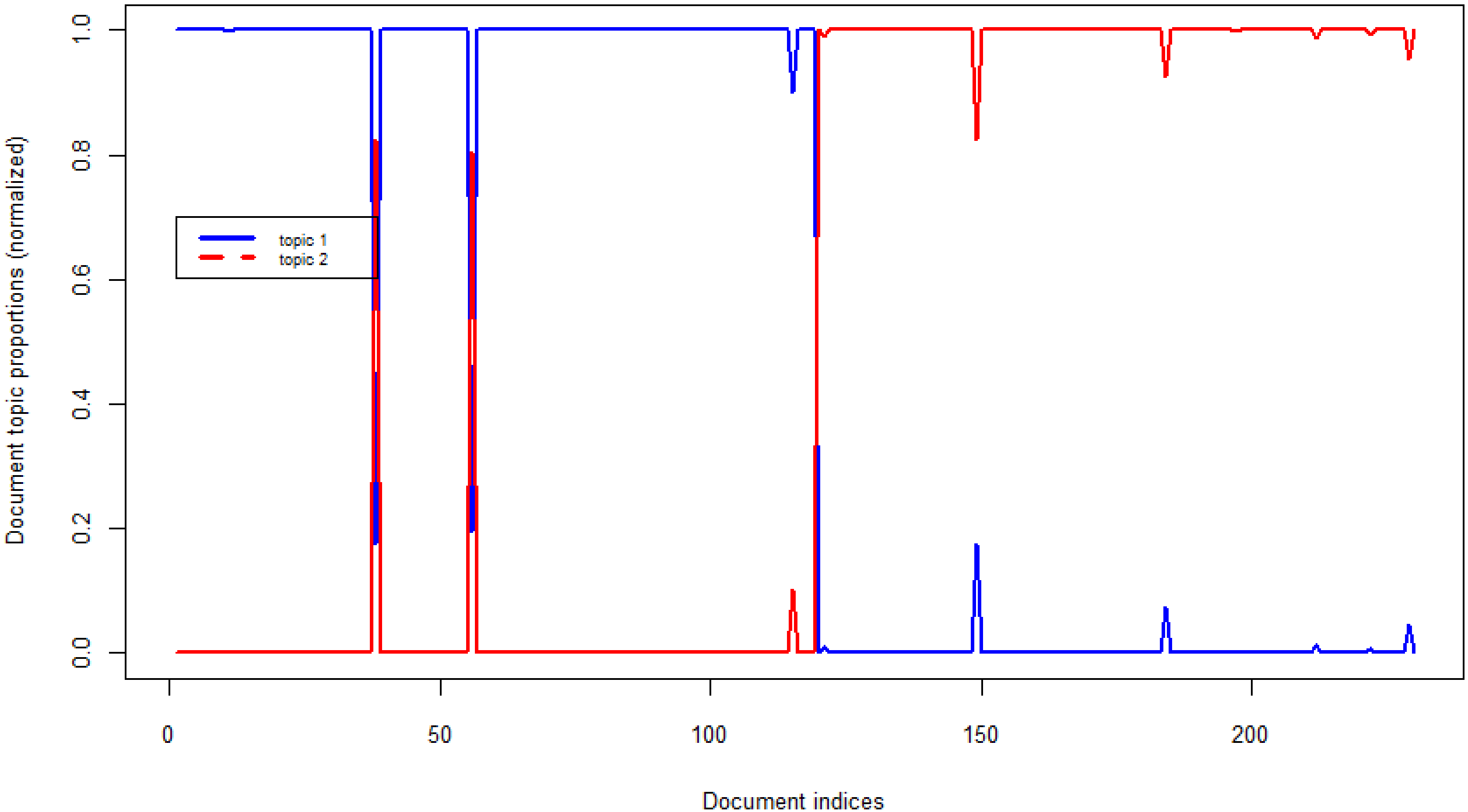
Reference: Slides on LDA by David Blei

LDA for Topic Extraction

Data set: Wikipedia pages from **whales** and **tires** domain; two given topics

	[Topic 1]	[Topic 2]
[1,]	"whale"	"tire"
[2,]	"dolphin"	"tyre"
[3,]	"species"	"wheel"
[4,]	"sea"	"rubber"
[5,]	"ship"	"vehicle"
[6,]	"killer"	"tread"
[7,]	"iwc"	"car"
[8,]	"orca"	"pressure"
[9,]	"population"	"wear"
[10,]	"animal"	"system"

Topic-words are listed in the non-increasing order of ***p(topic|term)***



First 119 docs are from **whales** domain and last 111 from **tires** domain. K = 2 (input)

Observations

- If the training documents are from completely different domains, LDA finds document topic mixtures that can classify the corpus documents
  - E.g. whales are tires
  - E.g. Case w/ similar domains subdomains of Whale_Products, Killer_Whale, Whaling, Baleen_Whale, and Toothed_Whale

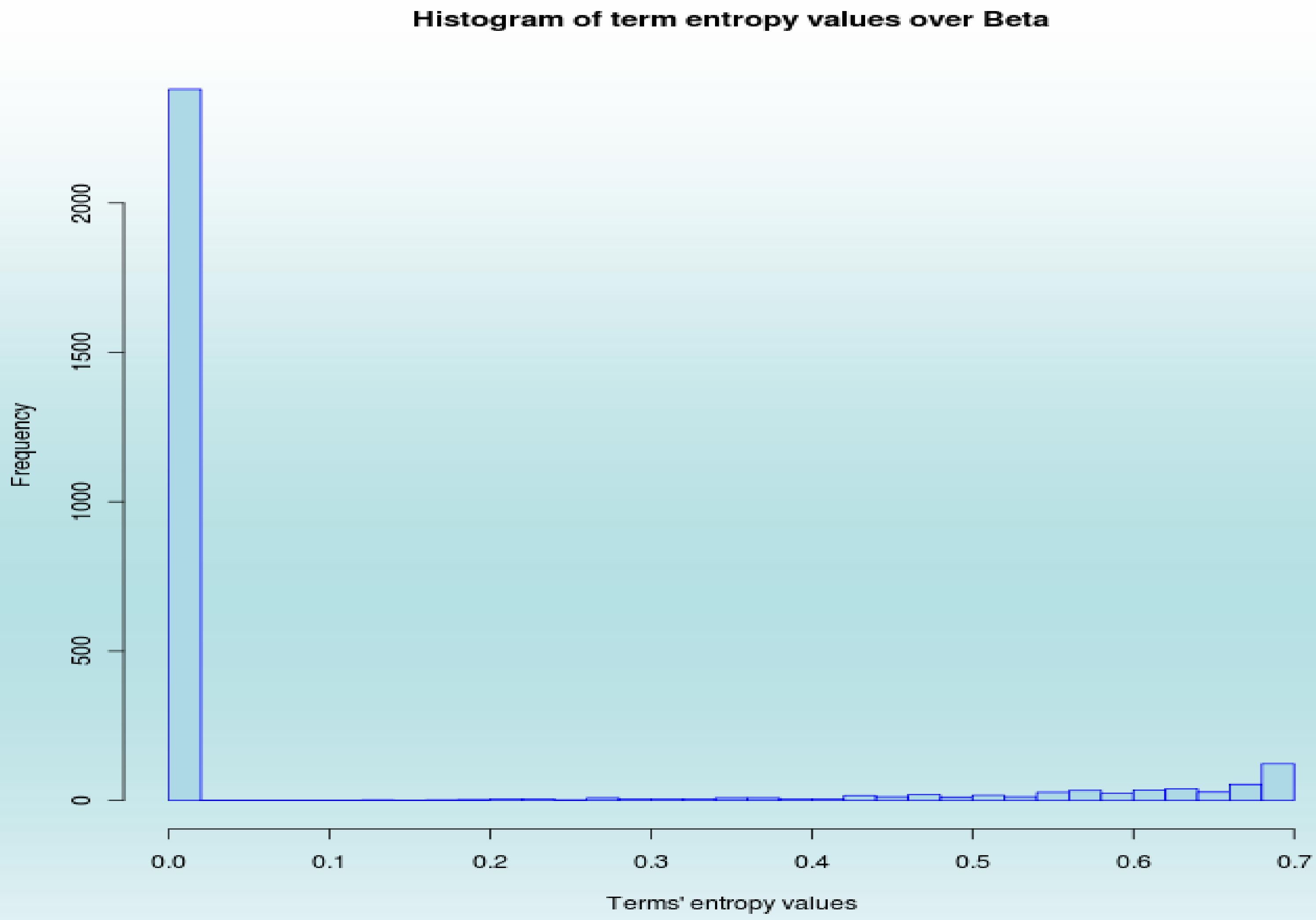
- This is supported by the path-similarity defined in the WordNet hierarchy

	Tires	Killer_whale	Baleen_whale	Toothed_whale
Tires	1	0.0526	0.0588	0.0588
Killer_whale	0.0526	1	0.2000	0.3333
Baleen_whale	0.0588	0.2000	1	0.3333
Toothed_whale	0.0588	0.3333	0.3333	1

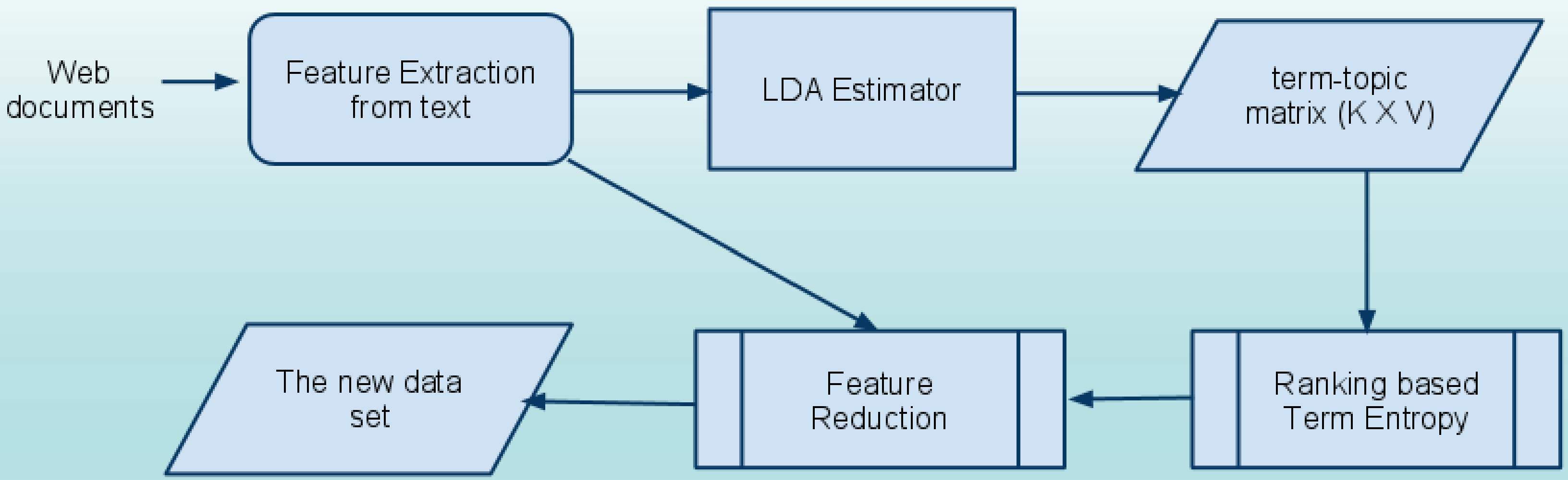
WordNet - facts

- A manually build lexical data base of English
- English words are grouped into synsets or sets of synonyms
  - Contains nouns, verbs, adverbs
- Synsets are connected to other synsets by semantic relations such as
  - Hypernyms e.g. canine is a hypernym of dog
  - hyponyms e.g. dog is a hyponym of canine
- Path similarity is based on the path distance defined on the semantic hierarchy of synsets

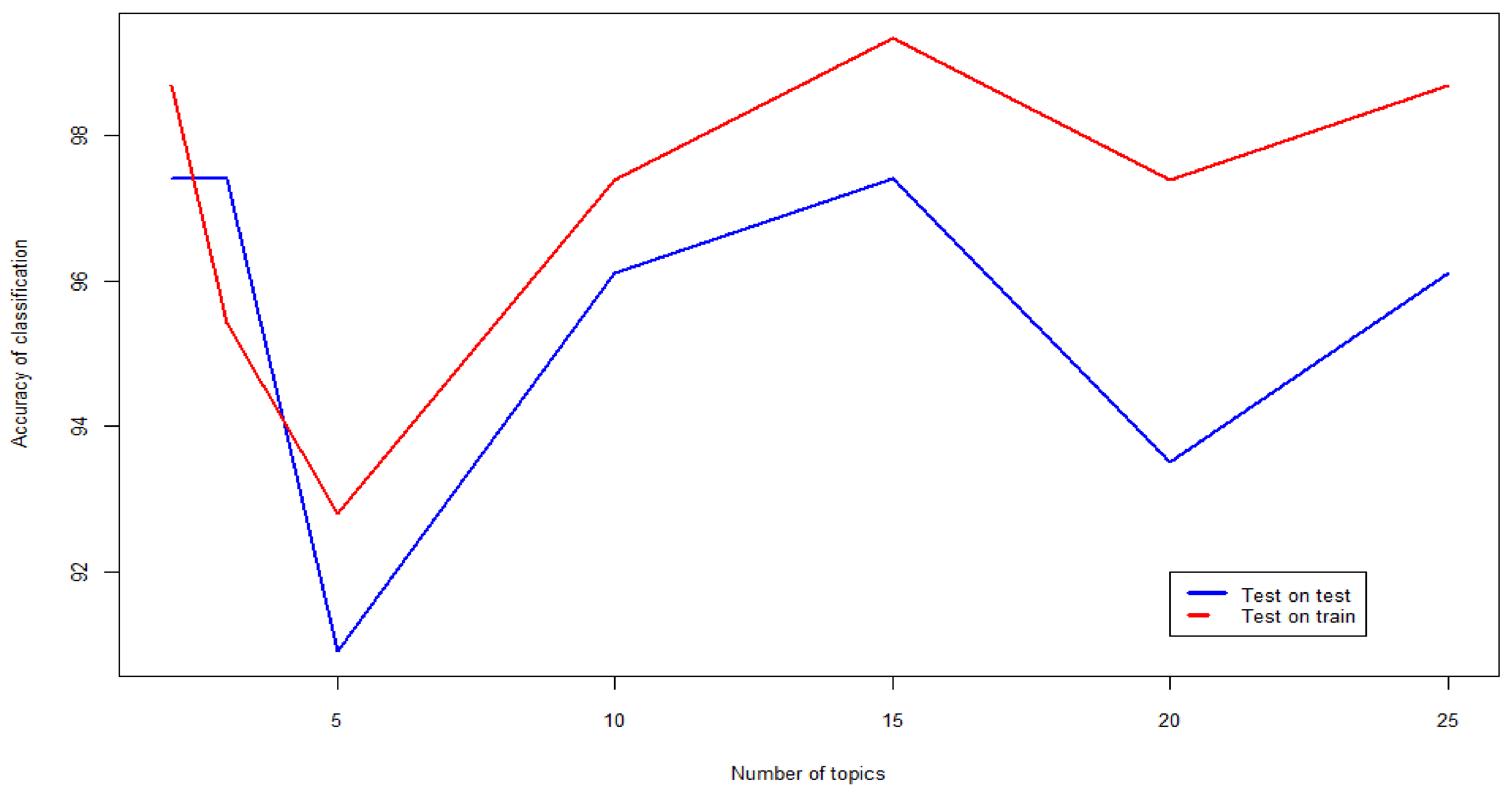
LDA for Dimensionality Reduction



LDA for Dimensionality Reduction



LDA output for classification



Limitations of LDA

- Inability to model topic correlation
  - E.g. a document about **genetics** is more likely to be about **disease**
  - Reason: Dirichlet distribution is used to model the topic variability
  - Correlated topic model solves this by using the logistic normal distribution
- Inability to capture the number of topics in a corpus
  - Hierarchical Dirichlet process

Conclusion

- Topic Modeling in general
- TF-IDF, LSI and pLSI advantages and disadvantages
- Latent Dirichlet allocation
  - A fully generative process even at the level of documents
  - Advantages over TF-IDF, LSI, and pLSI
  - Dimensionality reduction
  - Limitations