

Occurrence-based text representations and topic modelling

Prof. Luca Cagliero
Dipartimento di Automatica e Informatica
Politecnico di Torino



**Politecnico
di Torino**

Lecture goal

- Occurrence-based text representation
- Topic modelling
 - Latent Semantic Indexing
 - Latent Dirichlet Allocation
 - The Author-Topic Model

Structured text representation

- Feature-based text representation
- Encode the information hidden in the text in a feature-value representation
- Two main strategies
 - Occurrence-based representation
 - textual features are computed on the frequencies of occurrences of the main textual units (e.g., words, n-grams, phrases) in larger text snippets
 - Distributed vector representations of text
 - high-dimensional text representation extracted by means of neural network training

Structured text representation

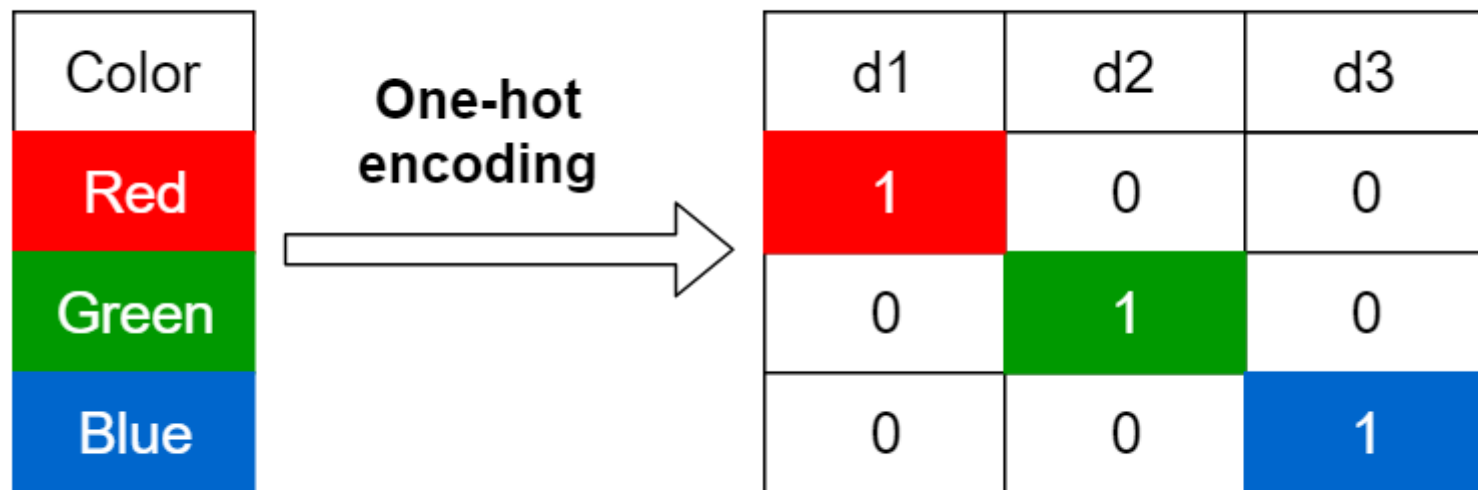
- Feature-based text representation
- Encode the information hidden in the text in a feature-value representation
- Two main strategies
 - Occurrence-based representation
 - textual features are computed on the frequencies of occurrences of the main textual units (e.g., words, n-grams, phrases) in larger text snippets
 - Distributed vector representations of text
 - high-dimensional text representation extracted by means of neural network training

Here we focus on occurrence-based text representations

One-hot encoding

- Each word in the dictionary is regarded as discrete symbol and mapped to a distinct feature
- Each document is represented as a tuple in a relation whose schema consists of the word-level features
- The feature takes binary values (0/1) indicating whether a particular word occurs in the given document

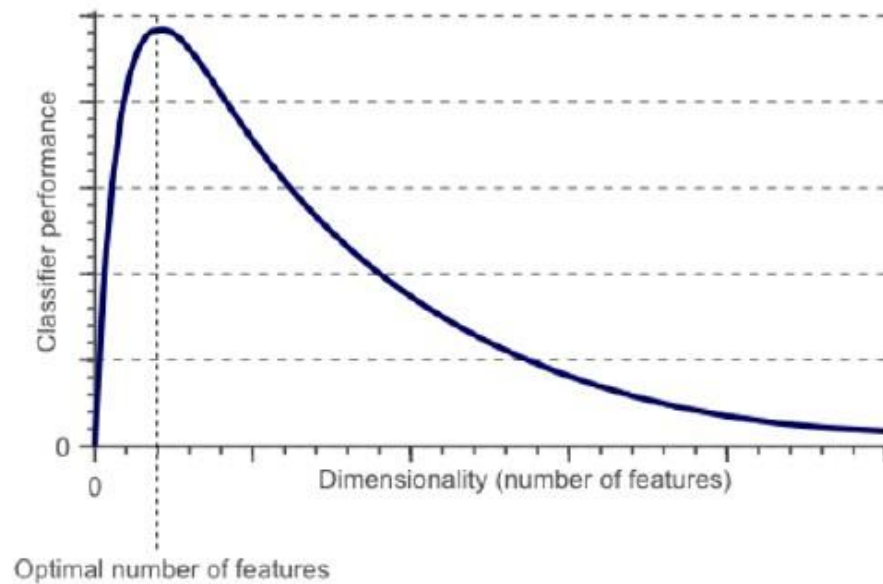
One-hot encoding



One-hot encoding: drawbacks

- High dimensionality
 - Number of words in the dictionary \gg Number of documents
- Data sparsity
 - Each document is likely to include a limited number of words in the dictionary

Curse of dimensionality



Term-frequency - inverse document frequency

- Established weighted representation of text
 - Counteract data sparsity
- For each combination of word and document it stores the term frequency-inverse document frequency (tf-idf) statistics
- $n_{i,j}$: number of occurrences of word i in document d_j
- d_j : number of words in document j
- $|D|$: number of documents
- Term frequency:

$$tf_{i,j} = \frac{n_{i,j}}{d_j}$$

- Inverse document frequency:

$$idf_i = \log_{10} \frac{|D|}{|\{d:i \in d\}|}$$

- Tf-idf statistics:

$$tf_{i,j} \cdot idf_i$$

Term-frequency - inverse document frequency

- Term frequency: observed frequency of occurrence within the specific document
 - Each word is representative of a local pattern
 - Term frequency positively contributes to the relative word importance
- Document frequency: number of documents in which the word occurs at least once
 - Word occurrences are spread over all documents
 - Document frequent negatively contributes to the relative word importance

Suitable for analyzing heterogeneous document collections

Term frequency - document frequency

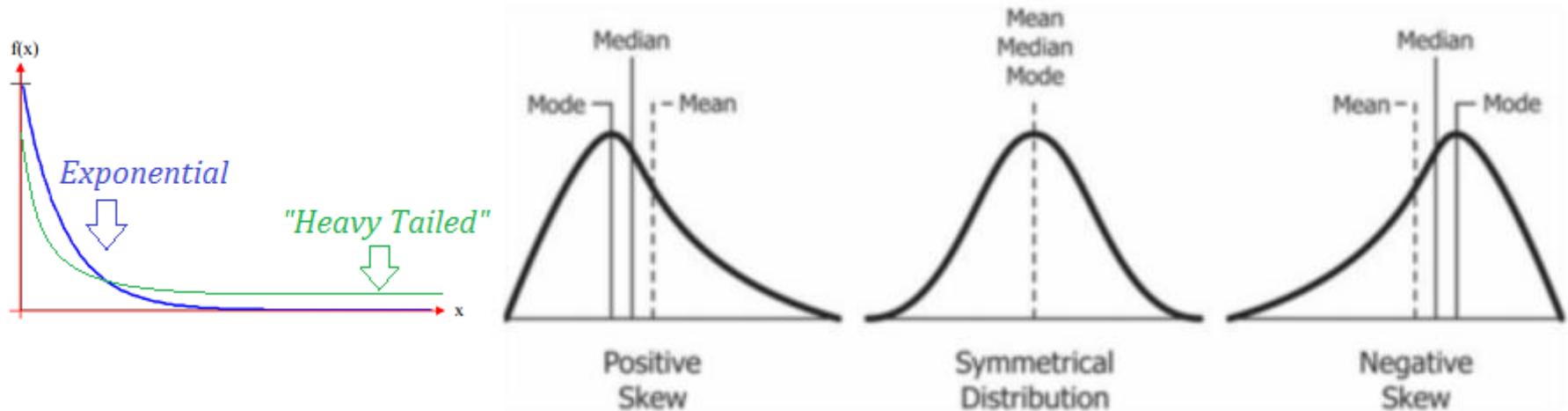
- Replace the inverse document frequency with the document frequency in the tf-idf statistics
- Word occurrences spread over all the documents are rewarded

Suitable for analyzing homogeneous document collections

Elena Baralis, Luca Cagliero, Alessandro Fiori, and Paolo Garza. 2015. MWI-Sum: A Multilingual Summarizer Based on Frequent Weighted Itemsets. ACM Trans. Inf. Syst. 34, 1, Article 5 (October 2015), 35 pages. DOI: <https://doi.org/10.1145/2809786>

Best Matching 25

- Overcome the issues related to unbounded term frequency values
 - Heavy tailed or skewed distributions of tf-idf values may come out
- It defines a family of scoring functions
 - Each of them contains slightly different components and parameters



Best Matching 25

- Key idea: introduce corpus-level statistics in the tf-idf formulation
 - *Avgdl*: average document length (expressed as number of words) over all the documents in the collection
 - k_1 : term frequency saturation (free parameter)
 - b : penalty score associated with the document length (free parameter)

$$\text{BM25}_{i,j} = \text{idf}_i \cdot \frac{tf_{i,j} \cdot (k_1 + 1)}{tf_{i,j} + k_1 \cdot (1 - b + b \cdot \frac{d_j}{\text{avgdl}})}$$

Alternative scoring functions: term frequency weight

weighting scheme	tf weight
binary	0, 1
raw count	$f_{t,d}$
term frequency	$f_{t,d} / \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K) \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Alternative scoring functions: document frequency weight

weighting scheme	idf weight ($n_t = \{d \in D : t \in d\} $)
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log \left(\frac{N}{1 + n_t} \right) + 1$
inverse document frequency max	$\log \left(\frac{\max_{\{t' \in d\}} n_{t'}}{1 + n_t} \right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

Alternative scoring functions

- Log normalization
 - To scale the skewed weight distributions
- Frequency max
 - To bound the range of variation of the frequency values
- Double normalization
 - To set a unique free parameter K

Additional reading on term weighting



- Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008.
 - <https://nlp.stanford.edu/IR-book/information-retrieval-book.html>
- Download and read chapter 6
 - <https://nlp.stanford.edu/IR-book/pdf/06vect.pdf>

Limitations of occurrence-based text representations

- Suitable for evaluating syntactical text similarities
- Unsuitable for capturing semantic text similarities

king = [0000000000010000]

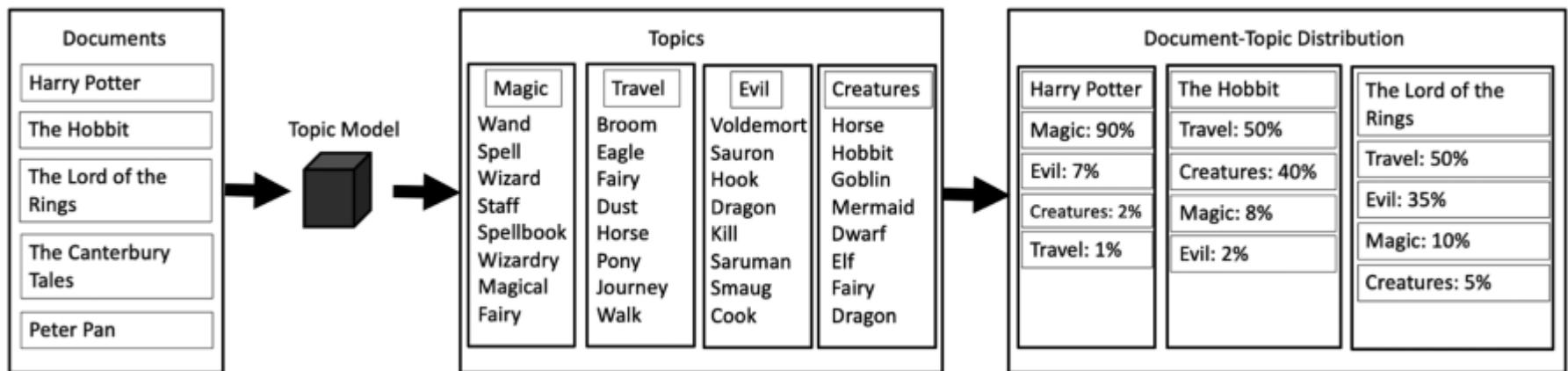
queen = [0000000010000000]

Curse of dimensionality: countermeasures

- Feature selection
 - Discard the less relevant features
- Dimensionality reduction
 - Derive a lower-dimensional representation

Topic modelling

- Established NLP application
 - Mainly unsupervised
- Focus on capturing latent topics in a large document corpus
- Provide end-users with a topic-level description of the analyzed documents



Topic modelling and dimensionality reduction

- The number of topics covered by a document is likely small
- Some embedding techniques and dimensionality reduction methods inherently provide a synthetic description of the most salient document topics
 - Vector representations of text allow clustering text into homogenous groups
 - Occurrence-based representations are high-dimensional can be reduced to low-dimensional spaces

Topic modeling techniques based on embedding techniques
will be covered later on in this course

Topic modelling: traditional approaches

- Based on occurrence-based text representation
- The distribution of words in each document is expressed as a weighted combination of concepts derived from the occurrence-based text representation
- Latent Semantic Indexing (LSI)
 - The underlying concepts and the corresponding weights are derived from the Singular Value Decomposition
- Latent Dirichlet Allocation (LDA)
 - Derive document-topics and topic-terms probabilities distributions using a generative model

Latent Semantic Indexing

- Mathematical approach used to represent occurrence-based text representations in a lower dimensional latent space
- Based on Singular Value Decomposition
 - Matrix factorization

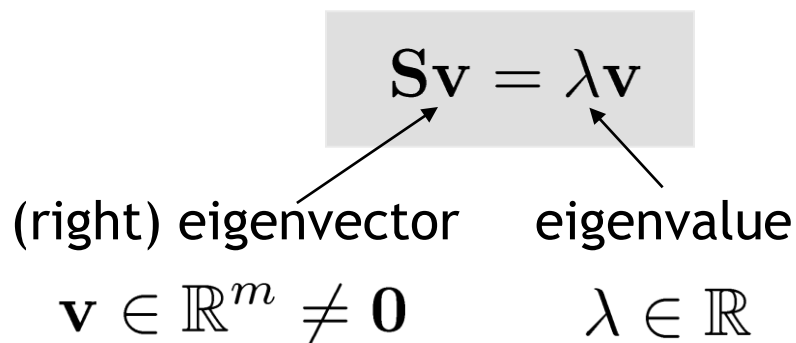
Singular Value Decomposition: preliminaries

- Eigenvectors and eigenvalues of $M \times N$ matrix A

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{v}$$

(right) eigenvector eigenvalue

$\mathbf{v} \in \mathbb{R}^m \neq \mathbf{0}$ $\lambda \in \mathbb{R}$



- \mathbf{S} is a square matrix
- Is this the case of a document-word matrix? Not actually..

Singular Value Decomposition: preliminaries

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{v} \iff (\mathbf{S} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$$

only has a non-zero solution if $|\mathbf{S} - \lambda\mathbf{I}| = 0$

Finding the m distinct solutions (i.e., the roots of the characteristic polynomial) can be complex even though \mathbf{S} is real.

Singular Value Decomposition

- For an $M \times N$ matrix \mathbf{A} of rank r there exists a factorization as follows:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

The diagram illustrates the dimensions of the matrices in the SVD equation $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. Below the equation, three boxes are arranged horizontally. The first box, labeled $M \times M$, has an arrow pointing to the matrix \mathbf{U} . The second box, labeled $M \times N$, has an arrow pointing to the matrix \mathbf{S} . The third box, labeled $V \text{ is } N \times N$, has an arrow pointing to the matrix \mathbf{V}^T .

Singular Value Decomposition

$$A = USV^T$$

Diagram illustrating the dimensions of the matrices in the SVD equation $A = USV^T$:

- U is $M \times M$
- S is $M \times N$
- V is $N \times N$

- $AA^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V\Sigma U^T) = U\Sigma^2 U^T$
- The columns of U are orthogonal eigenvectors of AA^T
- The columns of V are orthogonal eigenvectors of $A^T A$

Singular Value Decomposition

$$A = USV^T$$

Diagram illustrating the dimensions of the matrices in the SVD equation $A = USV^T$:

- U is $M \times M$
- S is $M \times N$
- V is $N \times N$

Eigenvalues $\lambda_1 \dots \lambda_r$ of AA^T are the eigenvalues of $A^T A$

$$s_i = \sqrt{l_i}$$

$$S = \text{diag}(s_1 \dots s_r)$$

$$\sigma_{i+1} \leq \sigma_i$$

Christopher D. Manning and Pandu Nayak. Introduction to Information Retrieval. CS276

Low-rank approximation

- We exploit SVD to compute the optimal low-rank approximation
 - A lower dimensional representation of the occurrence-based text representation
- Approximation problem: find A_k of rank k such that it minimizes the Frobenius norm

$$A_k = \min_{X: \text{rank}(X)=k} \|A - X\|_F$$

$$\|A\|_F \equiv \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}.$$

- A_k and X are $M \times N$ matrices
- The rank $r \gg K$ reflects the underlying dimensionality of the data

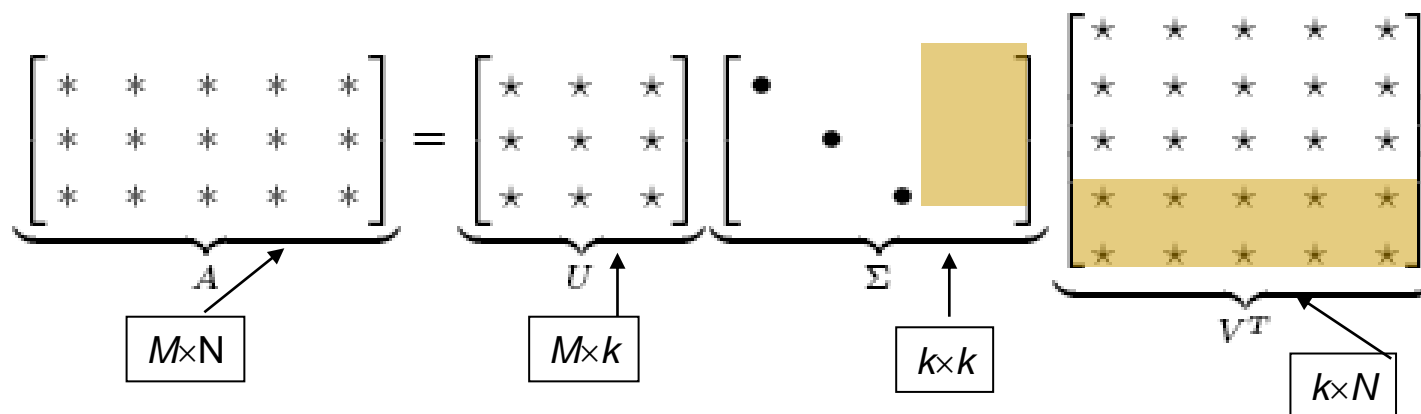
Low-rank approximation

- The approximation, in terms of the Frobenius norm of the error, is

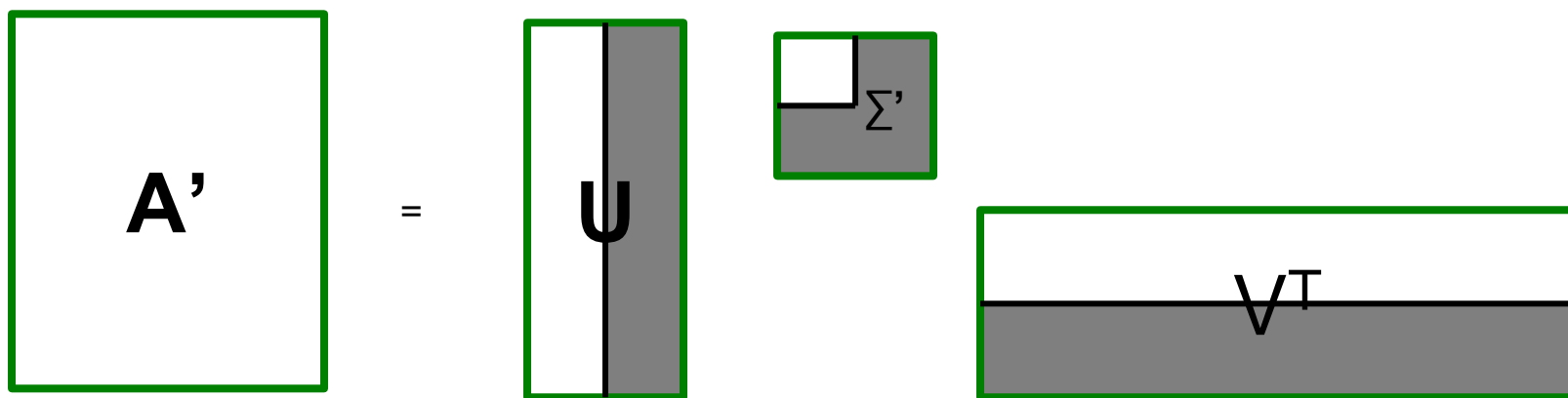
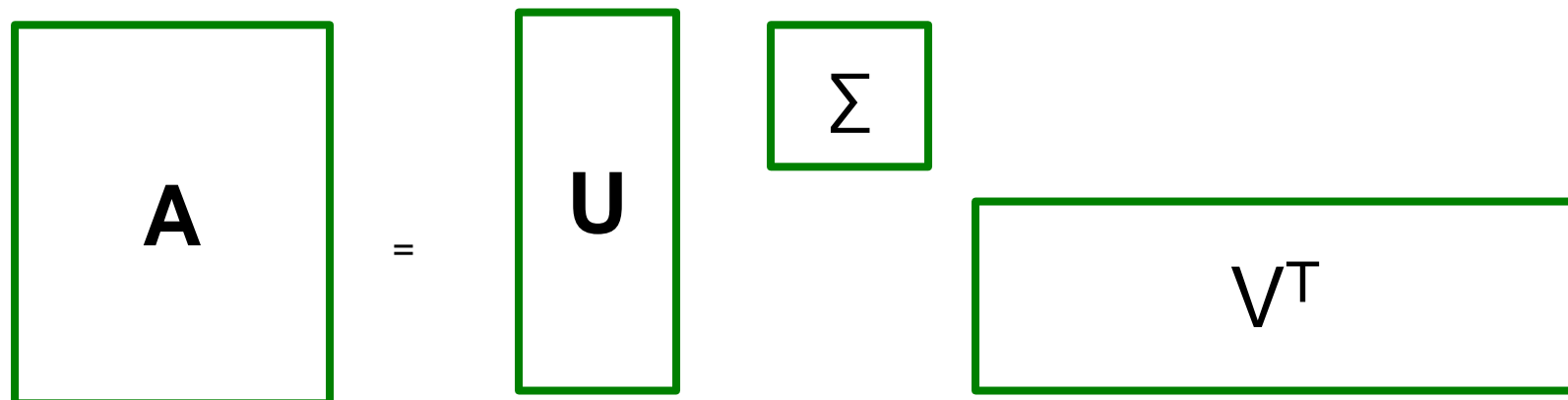
$$\min_{X: \text{rank}(X)=k} \|A - X\|_F = \|A - A_k\|_F = \sigma_{k+1}$$

Low-rank approximation

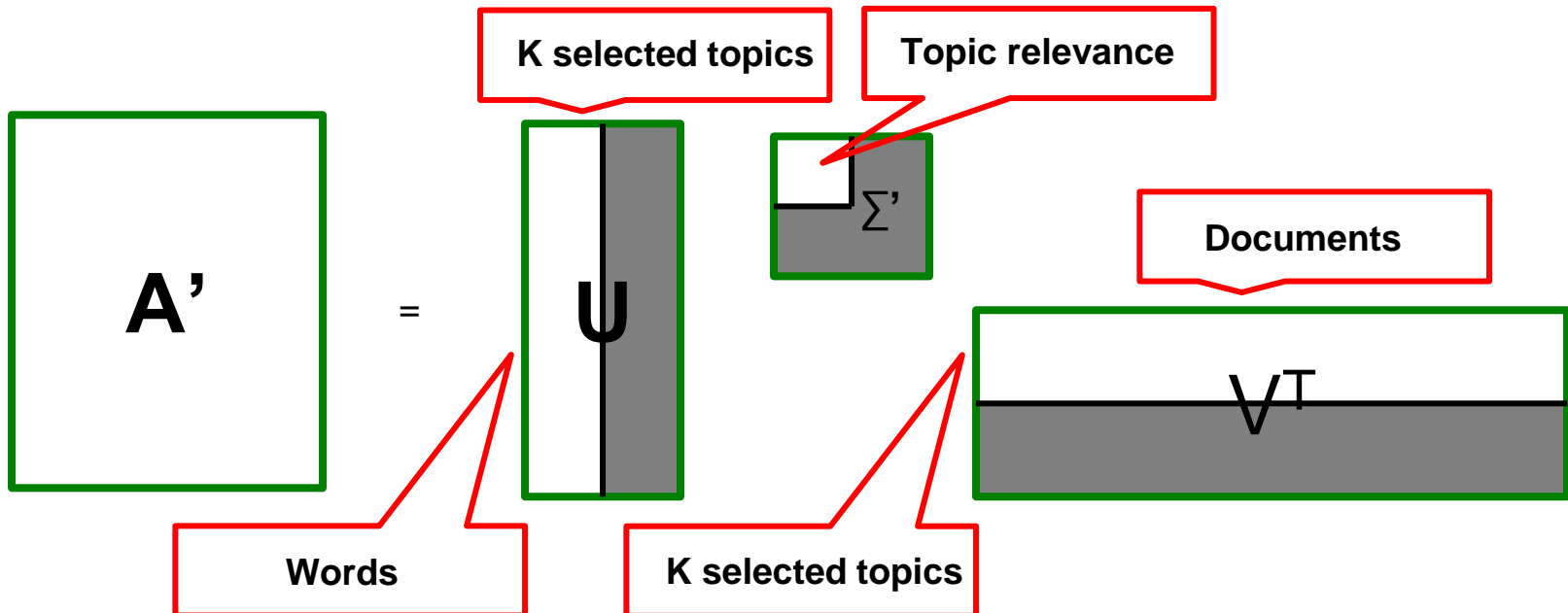
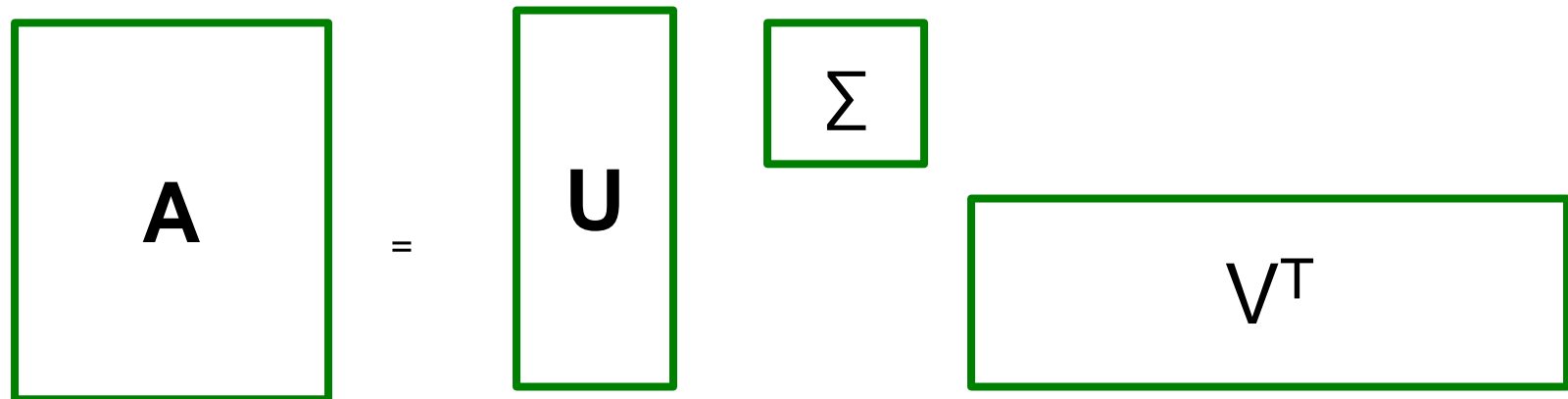
- We retain only the top- k singular values
 - All the other are reduced to zero
- We get a **reduced SVD**



Singular Value Decomposition



Singular Value Decomposition



Latent Semantic Indexing via SVD

- Compute a low-rank approximation of the document-word matrix
- Reduce the data dimensionality
 - Maps terms and documents to a lower dimensional representation
 - Preserve pairwise word associations conveyed by the occurrence-based model

$$\min_{X: \text{rank}(X)=k} \|A - X\|_F = \|A - A_k\|_F = S_{k+1}$$

Singular Value Decomposition

- Document similarity is computed based on the inner product in the low-dimensional representation
- The cosine similarity between the original document vectors can be computed as $A'^T A' = V' \Sigma^T \Sigma' V'^T$

Topic modeling using LSA

- LSA is instrumental for topic detection
- Each eigenvalue in A' is representative of a salient document concept
 - The i -th concept corresponding to σ_i is likely to be more relevant than those associated with σ_{i+1}

Topic modeling using LSA

- The first k rows of V^T represent the relevance of each document with respect to the shortlisted top- k concepts
 - We multiply the diagonal matrix Σ' by a specific orthogonal eigenvector in V^T
- The first k columns of U represent the relevance of each word with respect to the shortlisted top- k concepts
 - We multiply the specific orthogonal eigenvector in U by the diagonal matrix Σ'

Topic modeling using LSA: example

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

Topic detection based on LSI: example

U	1	2	3	4	5	
ship	−0.44	−0.30	0.00	0.00	0.00	
boat	−0.13	−0.33	0.00	0.00	0.00	
ocean	−0.48	−0.51	0.00	0.00	0.00	
wood	−0.70	0.35	0.00	0.00	0.00	
tree	−0.26	0.65	0.00	0.00	0.00	
Σ_2	1	2	3	4	5	
1	2.16	0.00	0.00	0.00	0.00	
2	0.00	1.59	0.00	0.00	0.00	
3	0.00	0.00	0.00	0.00	0.00	
4	0.00	0.00	0.00	0.00	0.00	
5	0.00	0.00	0.00	0.00	0.00	
V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

How can we set up the number of topics?

- The value K for truncated SVD is analyst-provided
- Heuristic approach
 - define σ_k as the smallest singular value that is above the half of the highest one (i.e., $\sigma_k > \sigma_1$)

Josef Steinberger and Karel Ježek. 2005. Text Summarization and Singular Value Decomposition. Springer, Berlin, Berlin, 245–254. DOI:http://dx.doi.org/10.1007/978-3-540-30198-1_25

SVD complexity

- $O(\min(NM^2, MN^2))$
- The complexity is lower when
 - the matrix is sparse
 - we consider only the $k \ll M$ singular vectors (reduced SVD)
 - Typical situation while coping with textual data
- Efficient SVD implementation are available, e.g.,
 - MATLAB (<https://www.mathworks.com/help/>)
 - SK-Learn (<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>)
 - Hadoop Spark (<https://spark.apache.org/docs/2.2.0/mllib-dimensionality-reduction.html>)
 - R (<https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/svd>)

LSA – Exercise 1

Let us consider the following SVD decomposition of a word-document matrix R (i.e., rows are words w_1, w_2, \dots, w_6 whereas columns are documents d_1, d_2, \dots, d_5).

Find the document-level topic relevance scores

$$\begin{array}{c} R \\ \begin{bmatrix} 5 & 5 & 0 & 0 & 1 \\ 4 & 5 & 1 & 1 & 0 \\ 5 & 4 & 1 & 1 & 0 \\ 0 & 0 & 4 & 4 & 4 \\ 0 & 0 & 5 & 5 & 5 \\ 1 & 1 & 4 & 4 & 4 \end{bmatrix} \end{array} \approx \begin{array}{c} U \\ \begin{bmatrix} -0,27 & 0,55 & -0,78 & 0 \\ -0,29 & 0,47 & 0,44 & -0,71 \\ -0,29 & 0,47 & 0,44 & 0,71 \\ -0,45 & -0,29 & -0,01 & 0 \\ -0,56 & -0,36 & -0,02 & 0 \\ -0,50 & -0,18 & -0,05 & 0 \end{bmatrix} \end{array} \begin{array}{c} \Sigma \\ \begin{bmatrix} 13,74 & 0 & 0 & 0 \\ 0 & 10,88 & 0 & 0 \\ 0 & 0 & 1,36 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \begin{array}{c} V^T \\ \begin{bmatrix} -0,32 & -0,32 & -0,52 & -0,52 & -0,5 \\ 0,63 & 0,63 & -0,25 & -0,25 & -0,29 \\ -0,02 & -0,02 & 0,41 & 0,41 & -0,82 \\ 0,71 & -0,71 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Solution

$$\begin{matrix} & R & & U & & \Sigma & & C_1 & C_2 & & V^T \\ \begin{bmatrix} 5 & 5 & 0 & 0 & 1 \\ 4 & 5 & 1 & 1 & 0 \\ 5 & 4 & 1 & 1 & 0 \\ 0 & 0 & 4 & 4 & 4 \\ 0 & 0 & 5 & 5 & 5 \\ 1 & 1 & 4 & 4 & 4 \end{bmatrix} & \approx & \begin{bmatrix} -0,27 & 0,55 & -0,78 & 0 \\ -0,29 & 0,47 & 0,44 & -0,71 \\ -0,29 & 0,47 & 0,44 & 0,71 \\ -0,45 & -0,29 & -0,01 & 0 \\ -0,56 & -0,36 & -0,02 & 0 \\ -0,50 & -0,18 & -0,05 & 0 \end{bmatrix} & \begin{bmatrix} 13,74 & 0 & 0 & 0 \\ 0 & 10,88 & 0 & 0 \\ 0 & 0 & 1,36 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -0,32 & -0,32 \\ 0,63 & 0,63 \\ -0,02 & -0,02 \\ 0,71 & -0,71 \end{bmatrix} & \begin{bmatrix} -0,52 & -0,52 & -0,5 \\ -0,25 & -0,25 & -0,29 \\ 0,41 & 0,41 & -0,82 \\ 0 & 0 & 0 \end{bmatrix}
 \end{matrix}$$

- How can we choose the optimal value of K?
 - Heuristic approach -> K = 2
- Example: d_1 topic relevance vector
 - Topic #1
 - $r_1 \times c_1 = -4.31$
 - Topic #2
 - $r_2 \times c_1 = 6.85$

LSA – Exercise 2

Let us consider the following SVD decomposition of a word-document matrix R (i.e., rows are words w_1, w_2, \dots, w_6 , columns are documents d_1, d_2, \dots, d_5).

Rank the input documents in order of decreasing relevance to topic #2

What are the most representative words for that topic?

$$\begin{matrix} R & U & \Sigma & V^T \\ \begin{bmatrix} 5 & 5 & 0 & 0 & 1 \\ 4 & 5 & 1 & 1 & 0 \\ 5 & 4 & 1 & 1 & 0 \\ 0 & 0 & 4 & 4 & 4 \\ 0 & 0 & 5 & 5 & 5 \\ 1 & 1 & 4 & 4 & 4 \end{bmatrix} & \approx \begin{bmatrix} -0,27 & 0,55 & -0,78 & 0 \\ -0,29 & 0,47 & 0,44 & -0,71 \\ -0,29 & 0,47 & 0,44 & 0,71 \\ -0,45 & -0,29 & -0,01 & 0 \\ -0,56 & -0,36 & -0,02 & 0 \\ -0,50 & -0,18 & -0,05 & 0 \end{bmatrix} & \begin{bmatrix} 13,74 & 0 & 0 & 0 \\ 0 & 10,88 & 0 & 0 \\ 0 & 0 & 1,36 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -0,32 & -0,32 & -0,52 & -0,52 & -0,5 \\ 0,63 & 0,63 & -0,25 & -0,25 & -0,29 \\ -0,02 & -0,02 & 0,41 & 0,41 & -0,82 \\ 0,71 & -0,71 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

LSA - Ex. 2 solution

$$\begin{matrix} & R & & & U & & \Sigma & & V^T \\ & & & r_1 & & & c_2 & & \\ \begin{bmatrix} 5 & 5 & 0 & 0 & 1 \\ 4 & 5 & 1 & 1 & 0 \\ 5 & 4 & 1 & 1 & 0 \\ 0 & 0 & 4 & 4 & 4 \\ 0 & 0 & 5 & 5 & 5 \\ 1 & 1 & 4 & 4 & 4 \end{bmatrix} & \approx & \begin{bmatrix} -0,27 & 0,55 & -0,78 & 0 \\ -0,29 & 0,47 & 0,44 & -0,71 \\ -0,29 & 0,47 & 0,44 & 0,71 \\ -0,45 & -0,29 & -0,01 & 0 \\ -0,56 & -0,36 & -0,02 & 0 \\ -0,50 & -0,18 & -0,05 & 0 \end{bmatrix} & \begin{bmatrix} 13,74 & 0 & 0 & 0 \\ 0 & 10,88 & 0 & 0 \\ 0 & 0 & 1,36 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -0,32 & -0,32 & -0,52 & -0,52 & -0,5 \\ 0,63 & 0,63 & -0,25 & -0,25 & -0,29 \\ -0,02 & -0,02 & 0,41 & 0,41 & -0,82 \\ 0,71 & -0,71 & 0 & 0 & 0 \end{bmatrix}
 \end{matrix}$$

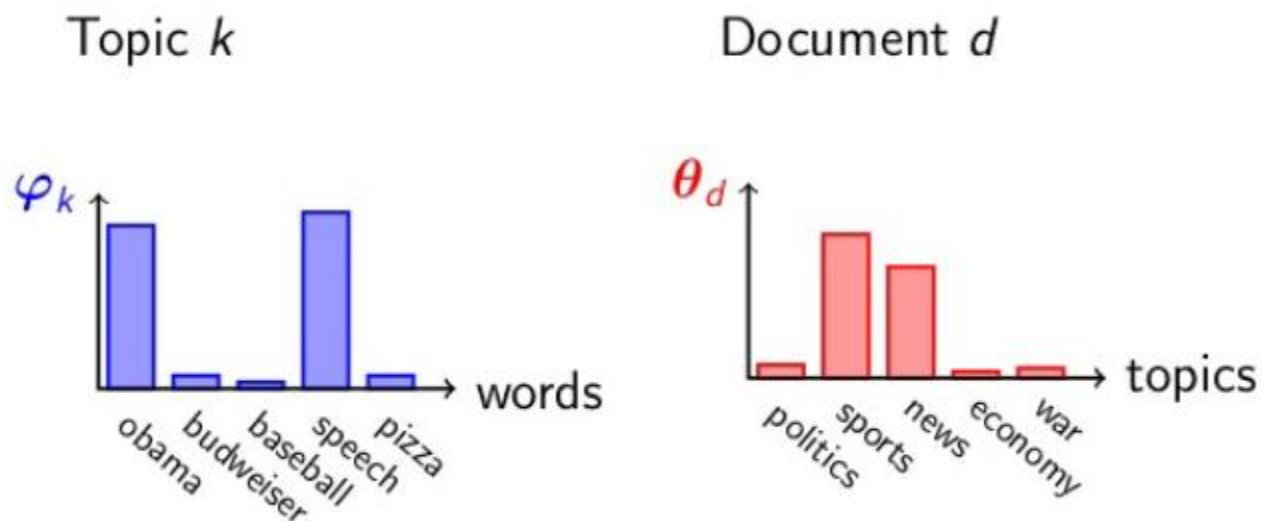
- Topic relevance vectors separately for each document
 - $d_1 = [-4.31, 6.85]$
 - $d_2 = [-4.31, 6.85]$
 - $d_3 = [-7.14, -2.72]$
 - $d_4 = [-7.14, -2.72]$
 - $d_5 = [-6.87, -3.16]$
- Documents d_1 and d_2 are mostly relevant to topic #2
- Topic #2 word relevance scores separately for each word
 - $W_1: 5.984 (= r_1 \times c_2)$
 - $W_2: 5.11$
 - $W_3: 5.11$
 - $W_4: -3.16$
 - $W_5: -3.92$
 - $W_6: -1.96$

Latent Dirichlet Allocation

- Probabilistic topic model
- Key idea: documents are mixtures of multiple topics
- Topics are defined as distributions over a fixed vocabulary
- Documents are defined as distributions over the set of different topics
 - They range over multiple topics in different proportions

Generating Summary Keywords for Emails Using Topics. Mark Dredze, Hanna M. Wallach, Danny Puller, , Fernando Pereira. IUI'08. ACM. 2008

Latent Dirichlet Allocation



David M. Blei, Andrew Y. Ng, Micheal I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (2003) 993-1022

Latent Dirichlet Allocation

- Generative topic model
- Each word w in a document d is assumed to have been generated
 - by sampling a topic from a document-specific distribution over topics and
 - by sampling a word from the distribution over words that characterizes that topic
- The distributions over topics and words are drawn from conjugate Dirichlet priors $\text{Dir}(\alpha)$ and $\text{Dir}(\beta)$

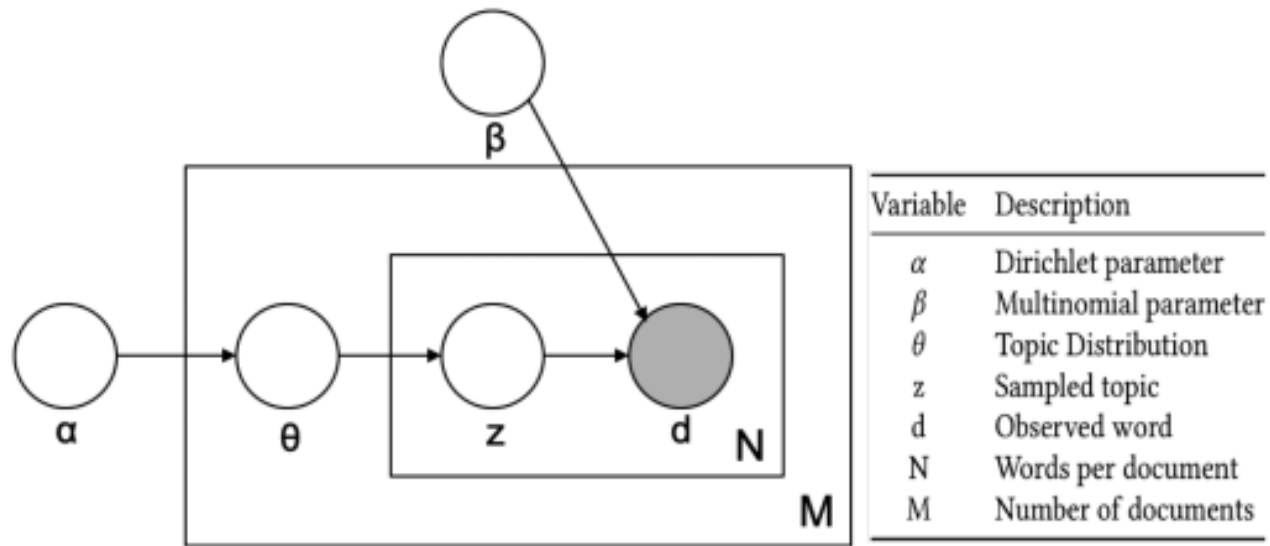
$$g(\theta_1, \theta_2, \dots, \theta_k | (x_1, x_2, \dots, x_k)) = \text{Dir}_k(\alpha_1 + x_1, \alpha_2 + x_2, \dots, \alpha_k + x_k)$$

LDA inputs

- Vocabulary of words W
- The number of topics k
- parameters α and β
- All variables that contribute to generate the corpus are independent of the number N of words in each document

Generating Summary Keywords for Emails Using Topics. Mark Dredze, Hanna M. Wallach, Danny Puller, , Fernando Pereira. IUI'08. ACM. 2008

Latent Dirichlet Allocation



Statistical inference

- Statistical inference algorithms are used to optimize the Dirichlet parameters and to infer the latent topics and document-specific topic mixtures
 - E.g., the Gibbs-Expectation Maximization (EM) algorithm is commonly used to do statistical inference of the posterior distribution of the latent variable for a given corpus
 - Gibbs-EM alternates between optimizing α and β and sampling a topic assignment for each word in the corpus from the distribution over topics for that word, conditioned on all other variables

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Latent Dirichlet Allocation

- For each document in the corpus and for each term
 - A topic is chosen accordingly to the document-topic distribution
- Words are extracted from the input vocabulary V by considering terms probabilities for each given topic of the documents' mixture

David M. Blei, Andrew Y. Ng, Micheal I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (2003) 993-1022

Generative process for LDA

- Key elements
 - the corpus
 - the documents
 - the terms
- Each topic z is sampled from every word w in the document
 - the same document can be described by multiple topics
- The variables θ_d are sampled once per document
- Parameters α and β are sampled once for the whole document generating process

David M. Blei, Andrew Y. Ng, Micheal I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (2003) 993-1022

Generative process for LDA

- Find the parameters of a topic-word distribution that maximize the likelihood of documents in the dataset over k topics
- Each document has a topic distribution specific to it that is proportional to the probability of each of its words in the topic-word distribution
- LDA uses expectation maximization to train its model and has two main parameters aside from k : α and β
 - α corresponds to topics-per-document ratio
 - Setting α higher results in more topics per document
 - β corresponds to words-per-topic ratio
 - Setting β lower results in fewer words per topic

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of machine Learning research, 3(Jan):993–1022, 2003.

Generative process for LDA

For each document d in a collection D :

- (1) Randomly draw the number of words N for d .
- (2) Randomly draw the topic distribution θ from the Dirichlet distribution, conditioned on the parameter α .
- (3) For each word w_i , $0 \leq i < N$:
 - (a) Draw a topic z_i from θ .
 - (b) Draw a word w_i based on the probability of w_i given the topic z_i and conditioned on the parameter β .

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Latent Dirichlet Allocation

- The probability of word w given document d and the Dirichlet priors' parameters α and β is given by

$$P(w|d, \alpha, \beta) = \sum_{t=1}^T P(w|t, \beta)P(t|d, \alpha)$$

- T is the number of latent topics

The LDA generative model

- The joint multivariate distribution of the topic mixture

$$p(\boldsymbol{\theta}, \mathbf{z}, \mathbf{w} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = p(\boldsymbol{\theta} | \boldsymbol{\alpha}) \prod_{n=1}^{N_d} p(z_n | \boldsymbol{\theta}) p(w_n | z_n, \boldsymbol{\beta})$$

- The probability of the whole corpus is computed by
 - integrating over the whole distribution θ
 - summing over all topics z
 - taking the product of the marginal probabilities of each document

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

LDA complexity

- High complexity on large document corpora
- Efficient implementations are available
 - MATLAB (<https://www.mathworks.com/help/textanalytics/ref/ldamodel.html>)
 - SK-Learn
(<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.LatentDirichletAllocation.html>)
 - Hadoop Spark
(<https://spark.apache.org/docs/2.3.1/api/java/org/apache/spark/mllib/clustering/LDA.html>)

Additional reading on LDA



- David M. Blei, Andrew Y. Ng, Micheal I. Jordan. Latent Dirichlet Allocation. Journal of Machine Learning Research 3 (2003) 993-1022
- Download and read the paper:
<https://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>

The Author-Topic Model

- Generative model for documents
- It extends the Latent Dirichlet Allocation to include authorship information
- It describes the topics according to the following facets:
 - Documents
 - Terms
 - Authors

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In Proceedings of the 20th conference on Uncertainty in artificial intelligence (UAI '04). AUAI Press, Arlington, Virginia, USA, 487–494.

The Author-Topic Model

- Example of applications
 - Who is the most authoritative author on a given topic?
 - What are the topic covered by a given author?
 - What is the most authoritative paper of an author?

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In Proceedings of the 20th conference on Uncertainty in artificial intelligence (UAI '04). AUAI Press, Arlington, Virginia, USA, 487–494.

Author-Topic Model

- A document d is a vector of N_d words w_d
- Each w_d is chosen from a vocabulary of size V
- A vector of A_d authors a_d is chosen from a set of authors of size A .
- A collection of D documents is defined by

$$D = \{(w_1, a_1), \dots, (w_d, a_d)\}$$

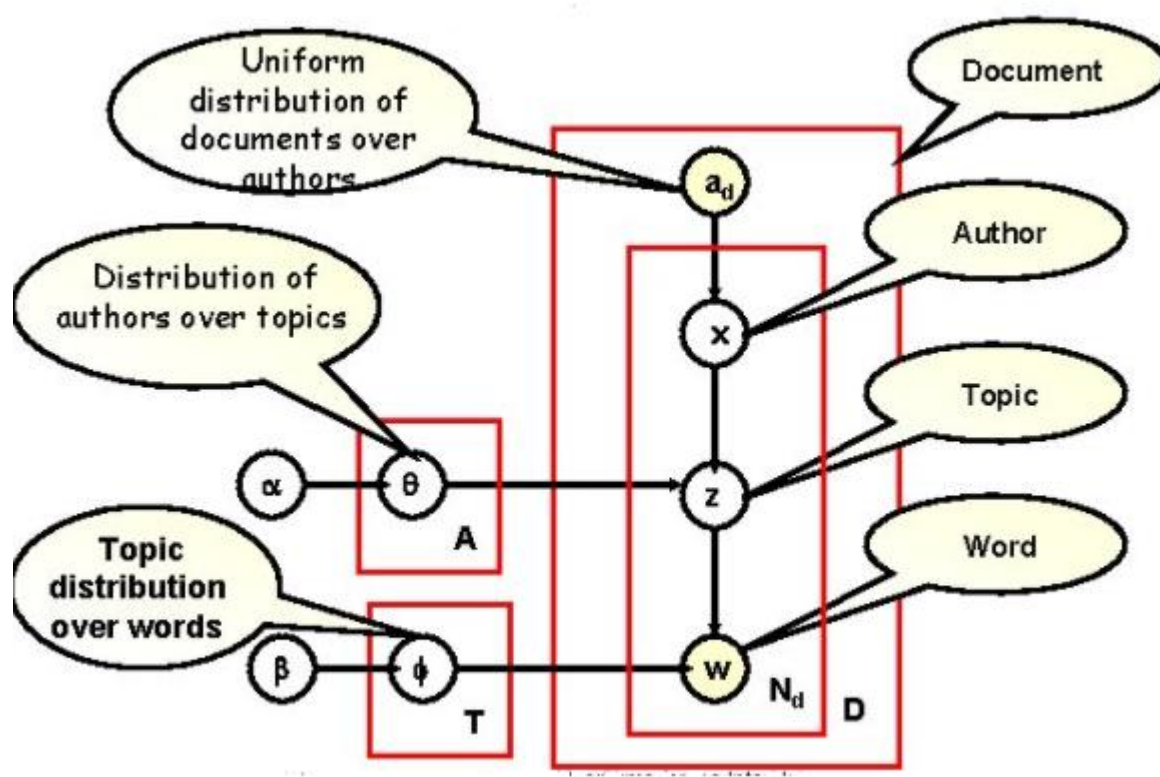
Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In Proceedings of the 20th conference on Uncertainty in artificial intelligence (UAI '04). AUAI Press, Arlington, Virginia, USA, 487–494.

The Author-Topic Model

- Each author is associated with a multinomial distribution over topics
- Each topic is associated with a multinomial distribution over words
- A document with multiple authors is modeled as a distribution over topics that is a mixture of the distributions associated with the authors

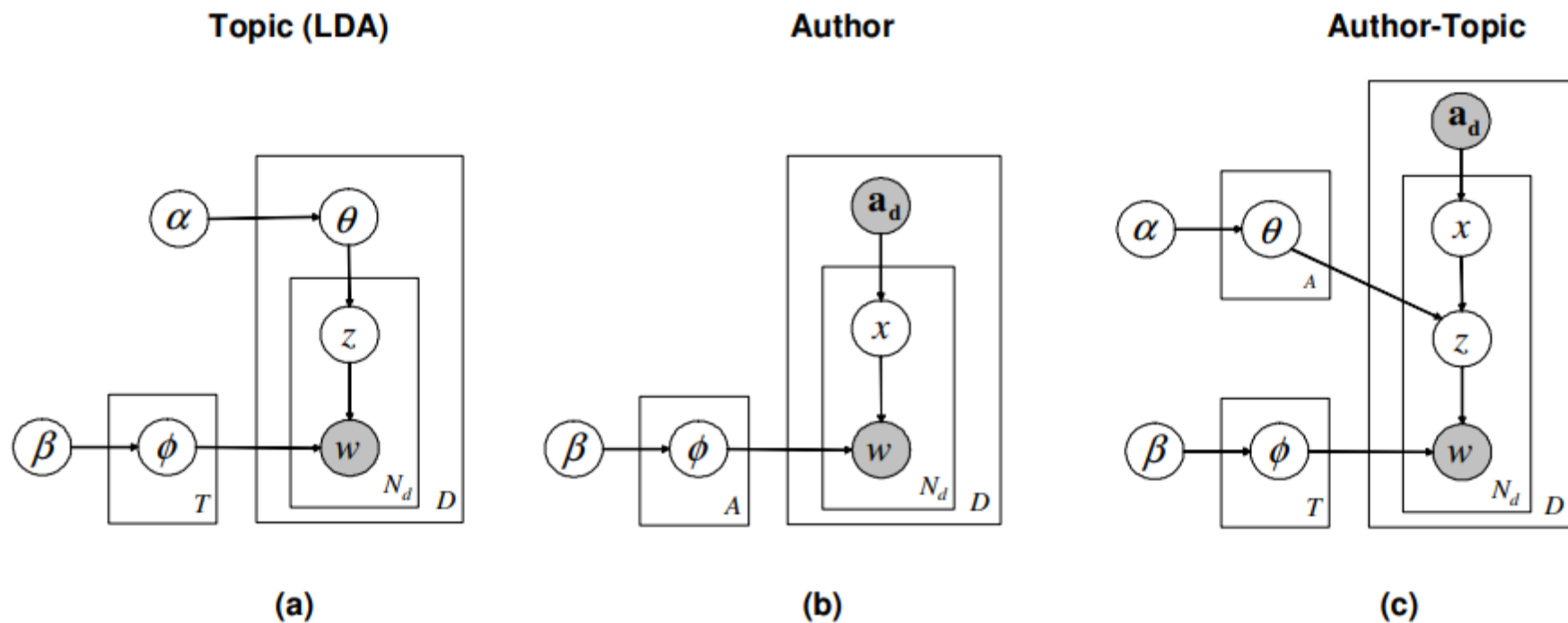
Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In Proceedings of the 20th conference on Uncertainty in artificial intelligence (UAI '04). AUAI Press, Arlington, Virginia, USA, 487–494.

The Author-Topic Model



Modeling Documents. Amruta Joshi. Stanford University. Department of Computer Science.

The Author-Topic Model



Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In Proceedings of the 20th conference on Uncertainty in artificial intelligence (UAI '04). AUAI Press, Arlington, Virginia, USA, 487–494.

Gibbs sampling for statistical inference

- Markov chain that converges to the posterior distribution on topics \mathbf{z} and authors \mathbf{x}

$$P(z_i = j, x_i = k | w_i = m, \mathbf{z}_{-i}, \mathbf{x}_{-i}, \mathbf{w}_{-i}, \mathbf{a}_d) \propto \frac{C_{mj}^{WT} + \beta}{\sum_{m'} C_{m'j}^{WT} + V\beta} \frac{C_{kj}^{AT} + \alpha}{\sum_{j'} C_{kj'}^{AT} + T\alpha}$$

- Use the results to infer the parameters
 - Θ : probability of a given word given a topic
 - φ : probability of a topic given an author

$$\phi_{mj} = \frac{C_{mj}^{WT} + \beta}{\sum_{m'} C_{m'j}^{WT} + V\beta}$$
$$\theta_{kj} = \frac{C_{kj}^{AT} + \alpha}{\sum_{j'} C_{kj'}^{AT} + T\alpha}$$

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In Proceedings of the 20th conference on Uncertainty in artificial intelligence (UAI '04). AUAI Press, Arlington, Virginia, USA, 487–494.

Additional reading on ATM



- Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In Proceedings of the 20th conference on Uncertainty in artificial intelligence (UAI '04). AUAI Press, Arlington, Virginia, USA, 487–494.
- Download and read the paper: <https://arxiv.org/abs/1207.4169>

Acknowledgements and copyright license

- Copyright licence

- Attribution + Noncommercial + NoDerivatives



- Acknowledgements

- I would like to thank Dr. Moreno La Quatra, who collaborated to the writing and revision of the teaching content

- Affiliation

- The author and his staff are currently members of the Database and Data Mining Group at Dipartimento di Automatica e Informatica (Politecnico di Torino) and of the SmartData interdepartmental centre
 - <https://dbdmg.polito.it>
 - <https://smartdata.polito.it>

Thank you!