Natural Language Processing

#L8

Latent Semantic Analysis

袁彩霞

yuancx@bupt.edu.cn

人工智能学院 智能科学与技术中心

Outline

- The problem
- Motivation
- LSA algorithm
- Probablistic LSA
- Latent Dirichlet Allocation
- Discussions

The Problem

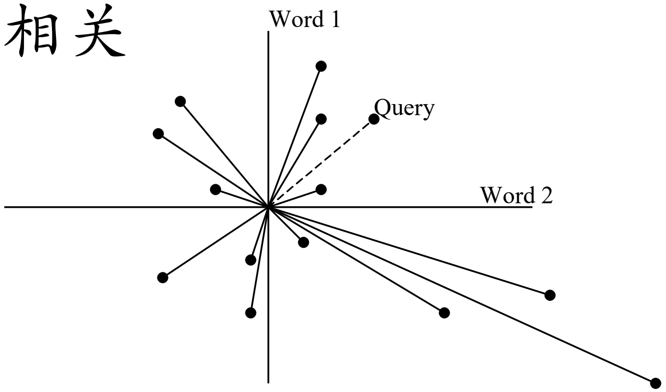
- 文本表示：
  - 向量空间方法：
    - 基于共现关系，得到词条文档矩阵
    - 理论上假设词条之间统计独立，即文本是由词构成的词集合（词袋，bag-of-words ）
    - 每个文档都被表示为一个向量（行）

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \cdots & \cdots & a_{NM} \end{pmatrix}$$

» a_{ik}: 词条k在文档i中的权重

The Problem

- E.g., 早期的信息检索(1980s)
  - 给定一个document集合，检索与给定query相关的文档
  - document中的词条 与 query中的词条相匹配
  - 采用Cosine来度量两个向量(query 和 documents)的距离：
    - 小的夹角 → 大的cosine值 → 相关
    - 大的夹角 → 小的cosine值 → 不相关

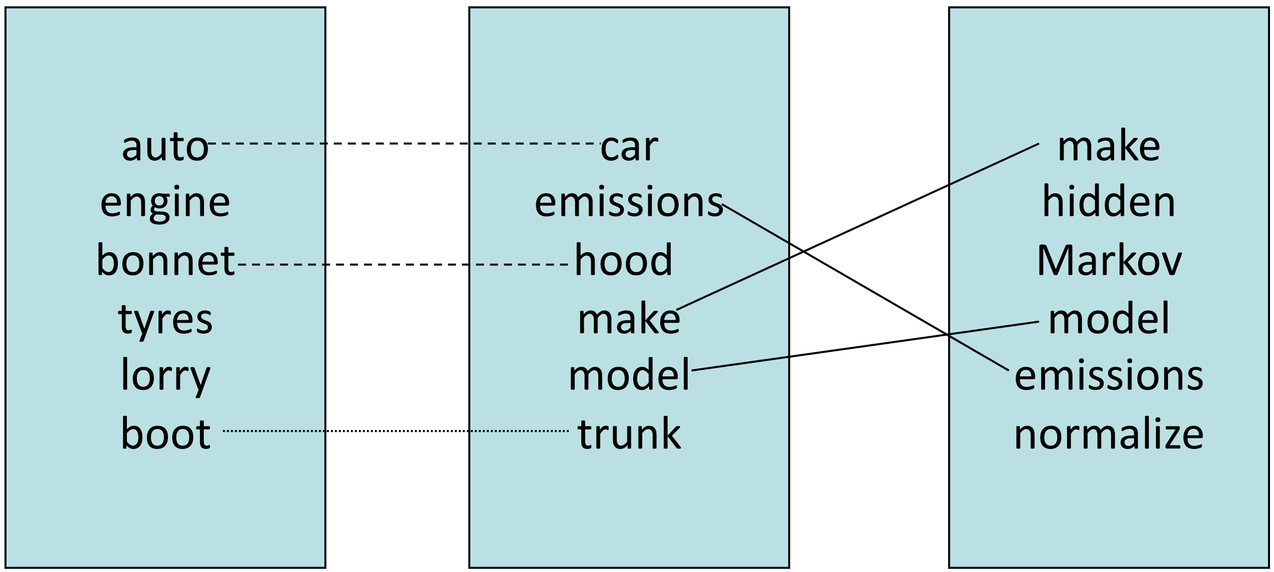


The Problem

- 向量空间模型的两个问题：
  - 同义词 (Synonymy)： e.g. car, automobile
    - $sim(d, q) < cos(\angle(d, q))$
    - 导致较低的召回率
  - 多义词(Polysemy): e.g. model, python, chip, bank
    - $sim(d, q) > cos(\angle(d, q))$
    - 导致较低的准确率
- Why?
  - Meanings/Concepts/Topics和words之间没有关联！

The Problem

- Example: Vector Space Model
  - (from Lillian Lee)



Synonymy

Will have small cosine

but are related

Polysemy

Will have large cosine

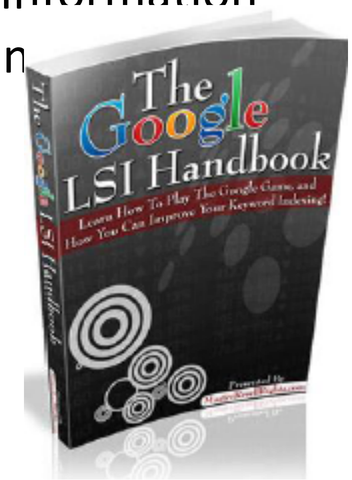
but not truly related

The Problem

- Solution: 采用隐藏的潜在概念表示documents (和queries)
- 潜在语义索引(Latent Semantic Indexing, LSI)
  - 最早为信息检索任务提出
    - T. Landauer, P. Foltz & S. Dumais, et al. , in the 1990s, at the Univ. of Colorado
  - Latent – “present but not evident, hidden”
  - Semantic – “meaning”

Some History

- The first papers about LSI
  - Dumais, S. T., Furnas, G. W., Landauer, T. K. and Deerwester, S. (1988), "Using latent semantic analysis to improve information retrieval." In Proceedings of CHI'88: Conference on Human Factors in Computing, New York: ACM, 281-285.
  - Deerwester, S., Dumais, S. T., Landauer, T. K., Furnas, G. W. and Harshman, R.A. (1990) "Indexing by latent semantic analysis." Journal of the Society for Information Science, 41(6), 391-407.
  - Foltz, P. W. (1990) "Using Latent Semantic Indexing for Information Filtering". In R. B. Allen (Ed.) Proceedings of the Conference on Office Information Systems, Cambridge, MA, 40-47.
  - …



Motivation

- 词条间的深层关系：
  - 不单单是邻接次数
  - 或者上下文中的共现次数
- 词条与上下文之间存在互相约束
- 如何捕捉这种约束？
  - 词条的上下文可替换性(contextual substitutability)： 词条A作为词条B在同一个上下文中使用的可能性
  - E.g., Doctor, nurse, patient, bedside

Latent Semantic Indexing

- LSI 通过词条在文档中的共现关系，找到词条的 “hidden meaning”
- LSI 将词条和文档映射到潜在语义空间 “latent semantic space”
- 在潜在语义空间中，同义词的相似性可以更好地体现

LSA vs. LSI

- But first:
- What is the difference between LSI and LSA?
  - LSI refers to using this technique for *indexing*, or information retrieval.
  - LSA refers to using it for everything else.
  - It’s the same technique, just different applications.

LSA Algorithm

- LSA is based on 3 steps:
  - 1) represent the text as a word $\times$ document matrix (word vectors are represented as rows with their frequency marked on each cell)
  - 2) Transform the cell entries by weighting them with a function expressing word importance and information rate
  - 3) Apply singular values decomposition to the matrix to reduce the dimensions of the vectors

Step 1: Representing the text as a matrix

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & \cdots & a_{2M} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \cdots & \cdots & a_{NM} \end{pmatrix}$$

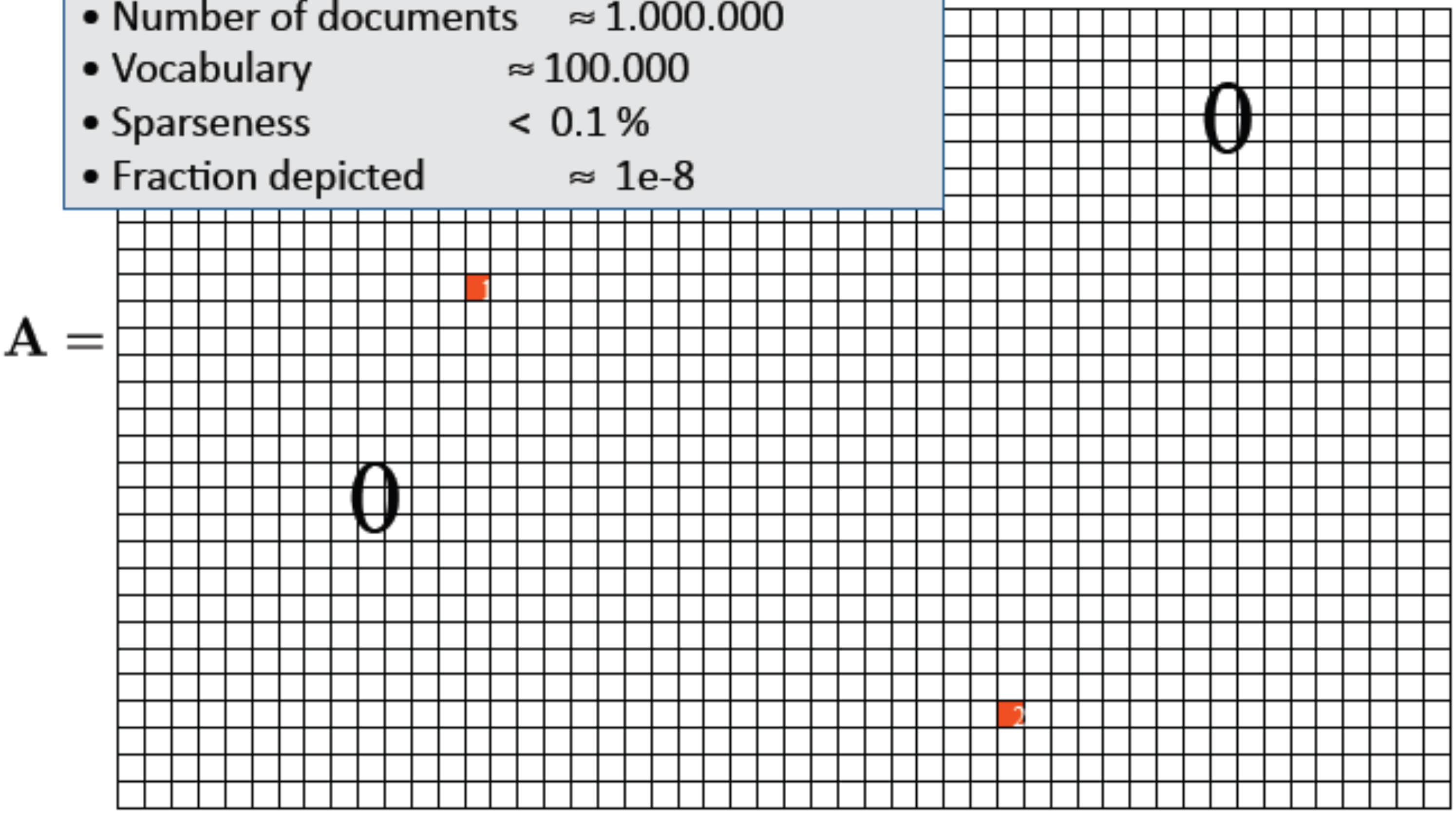
- Columns are words
- Rows are documents (or other context in which words occur)
- Cells contain the frequency with which the words appear in the documents

Step 2: Data transformation

- Weighting the terms, e.g.,
  - The word frequency in each cell is converted to its *log*
  - The entropy of each word is computed as *plogp* over all entries in its column
  - Each cell entry is divided by the column entropy value
  - This transformation weights each word-type occurrence by estimating its importance in the passage

A 100 millionth of a typical term-document matrix

Typical:
- Number of documents ≈ 1.000.000
- Vocabulary ≈ 100.000
- Sparseness < 0.1 %
- Fraction depicted ≈ 1e-8



Step 3: Singular Value Decomposition

- The document-word matrix A is decomposed into the product of 3 other matrices

  $A = U\Sigma V'$

  - U, V' are orthonormal matrices
    - an orthogonal matrix is a square matrix with real entries whose columns and rows are orthogonal unit vectors (i.e., orthonormal vectors)
  - Σ = diagonal matrix containing singular values ordered by size
    - Has non-zero entries on one of its main diagonals
    - The values of the main diagonal of Σ after SVD are called singular values of A and they are ordered from greatest to least along the main diagonal of Σ

Step 3: Singular Value Decomposition

- The document-word matrix A is decomposed into the product of 3 other matrices

$$A = U\Sigma V'$$

$$A_{n\times m} = U_{n\times n}\Sigma_{n\times m}V'_{m\times m}$$

- Keep only k eigenvalues from Σ, we get a *low-rank approximation* of A:

$$A_{n\times m} = U_{n\times k}\Sigma_{k\times k}V'_{k\times m}$$

- Convert terms and documents to points in k -dimensional space

Step 3: Singular Value Decomposition

- For an arbitrary matrix A there exists a factorization (singular value decomposition, SVD) as follows

$$A = U\Sigma V' \in \mathbb{R}^{n\times m}$$

- Where

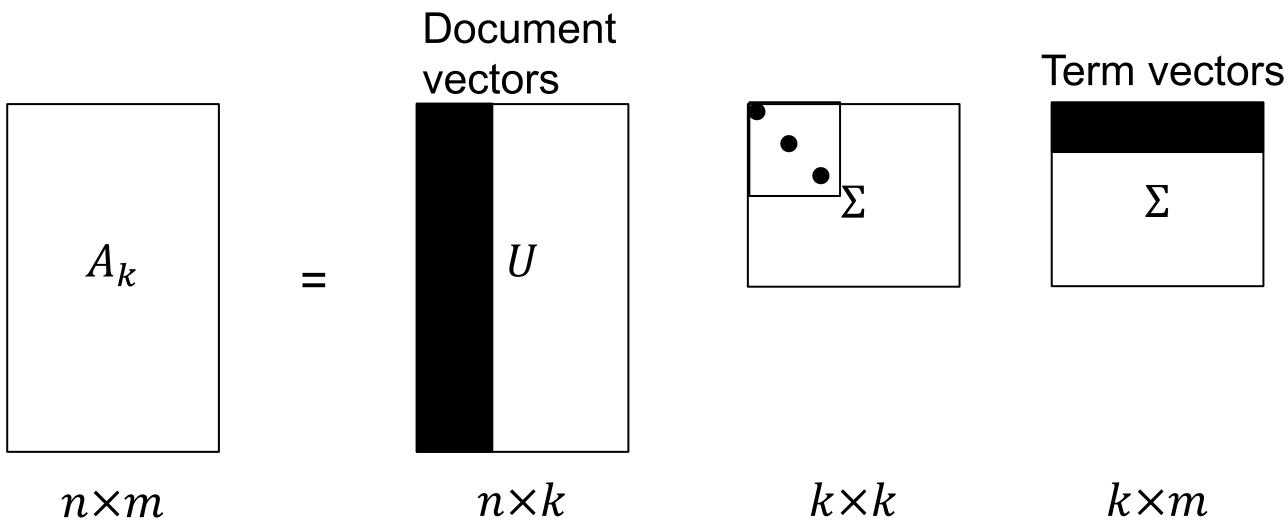
(i) $\mathbf{U} \in \mathbb{R}^{n\times k}$ $\quad \mathbf{\Sigma} \in \mathbb{R}^{k\times k}$ $\quad \mathbf{V} \in \mathbb{R}^{m\times k}$

(ii) $\mathbf{U}'\mathbf{U} = \mathbf{I}$ $\quad \mathbf{V}'\mathbf{V} = \mathbf{I}$ orthonormal columns

(iii) $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_k),\ \sigma_i \geq \sigma_{i+1}$ singular values (ordered)

(iv) $k = \mathrm{rank}(\mathbf{A})$

LSA decomposition via SVD



More on SVD

- SVD
  - tool for dimension reduction
  - similarity measure based on co-occurrence
  - finds optimal projection into low-dimensional space
  - can be viewed as a method for rotating the axes in n-dimensional space, so that the first axis runs along the direction of the largest variation among the documents
    - the second dimension runs along the direction with the second largest variation
    - and so on
  - generalized least-squares method

A Small Example

Technical Memo Titles

c1: *Human* machine *interface* for ABC *computer* applications

c2: A *survey* of *user* opinion of *computer system response time*

c3: The *EPS user interface* management *system*

c4: *System* and *human system* engineering testing of *EPS*

c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*

m2: The intersection *graph* of paths in *trees*

m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering

m4: *Graph minors*: A *survey*

A Small Example

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

- Compute Spearman correlation coefficient:

r (human, user) = -0.38, r (human, minors) =0.29

A Small Example

- Singular Value Decomposition

$$A = U\Sigma V'$$

- Selecting the k largest singular values, and corresponding singular vectors from *U* and *V*, you get the rank k approximation to *A with the smallest error (Frobenius norm).*

- Dimension Reduction

$$\tilde{A} = \widetilde{U}\tilde{\Sigma}\widetilde{V}'$$

A Small Example

- $\boldsymbol{U}$ =

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

A Small Example

- $\Sigma =$

$$
\begin{matrix}
3.34 & & & & & & & & \\
 & 2.54 & & & & & & & \\
 & & 2.35 & & & & & & \\
 & & & 1.64 & & & & & \\
 & & & & 1.50 & & & & \\
 & & & & & 1.31 & & & \\
 & & & & & & 0.85 & & \\
 & & & & & & & 0.56 & \\
 & & & & & & & & 0.36
\end{matrix}
$$

A Small Example

- V =

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

A Small Example

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

r (human, user) = 0.94,   r (human, minors) = -0.83

LSA Titles example:
Correlations between titles in raw data

	c1	*c2*	*c3*	*c4*	*c5*	*m1*	*m2*	*m3*
c2	-0.19							
c3	0.00	0.00						
c4	0.00	0.00	0.47					
c5	-0.33	0.58	0.00	-0.31				
m1	-0.17	-0.30	-0.21	-0.16	-0.17			
m2	-0.26	-0.45	-0.32	-0.24	-0.26	0.67		
m3	-0.33	-0.58	-0.41	-0.31	-0.33	0.52	0.77	
m4	-0.33	-0.19	-0.41	-0.31	-0.33	-0.17	0.26	0.56

0.02	
-0.30	**0.44**

Correlations in first-two dimension space

c2	0.91							
c3	1.00	0.91						
c4	1.00	0.88	1.00					
c5	0.85	0.99	0.85	0.81				
m1	-0.85	-0.56	-0.85	-0.88	-0.45			
m2	-0.85	-0.56	-0.85	-0.88	-0.44	1.00		
m3	-0.85	-0.56	-0.85	-0.88	-0.44	1.00	1.00	
m4	-0.81	-0.50	-0.81	-0.84	-0.37	1.00	1.00	1.00

0.92	
-0.72	1.00

Fundamental Comparison Quantities from the SVD Model

- Comparing Two Terms: the dot product between two row vectors of ***U*** reflects the extent to which two terms have a similar pattern of occurrence across the set of document.
- Comparing Two Documents: dot product between two column vectors of ***V′***
- Comparing a query and a Document: view query as a mini document, and compare it to your documents in the concept space.

Summary

- LSI puts documents together even if they don’t have common words if
  - The docs share frequently co-occurring terms

- Disadvantages:
  - Statistical foundation is missing
  - Context of terms is not taken into account (BOW)
  - Direction in latent space are hard to interpret

Summary

- Some Issues
  - SVD Algorithm complexity $O(n^2k^3)$
    - n = number of terms
    - k = number of dimensions in semantic space (typically small ~50 to 350)
    - for stable document collection, only have to run once
    - dynamic document collections: might need to rerun SVD, but can also "fold in" new documents

Summary

- Some issues
  - SVD assumes normally distributed data
    - term occurrence is not normally distributed (泊松分布)
    - matrix entries are weights, not counts, which may be normally distributed even when counts are not

Summary

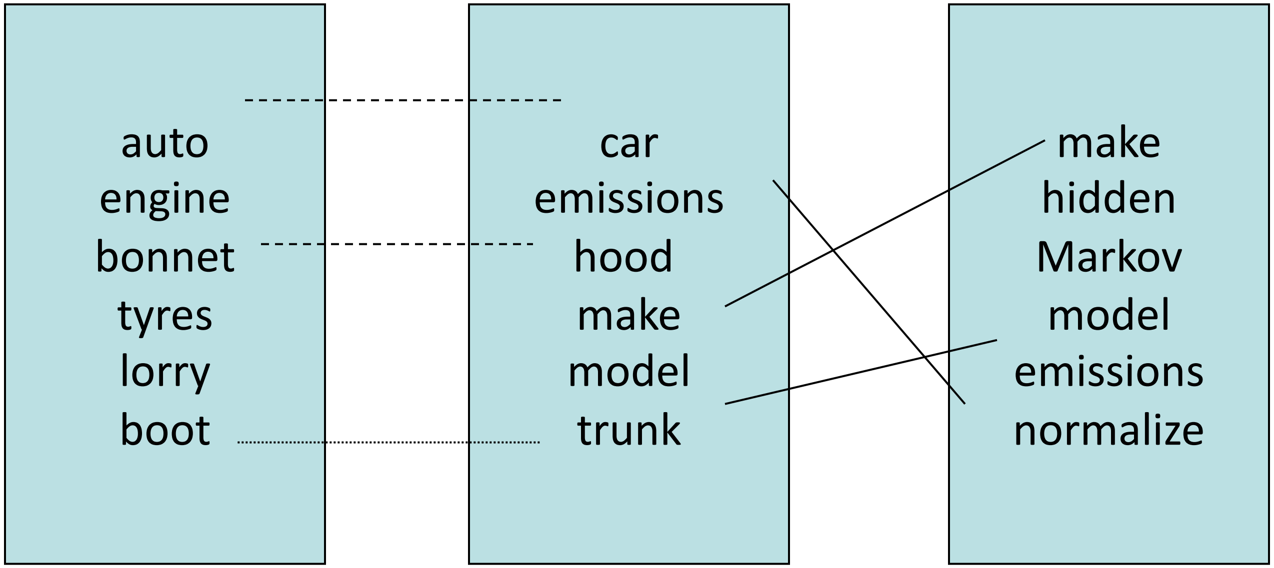
- Some issues
  - Finding optimal dimension for semantic space
    - precision-recall improve as dimension is increased until hits optimal, then slowly decreases until it hits standard vector model
    - run SVD once with big dimension, say k = 1000
      - then can test dimensions <= k
    - in many tasks 150-350 works well, still room for research

Summary

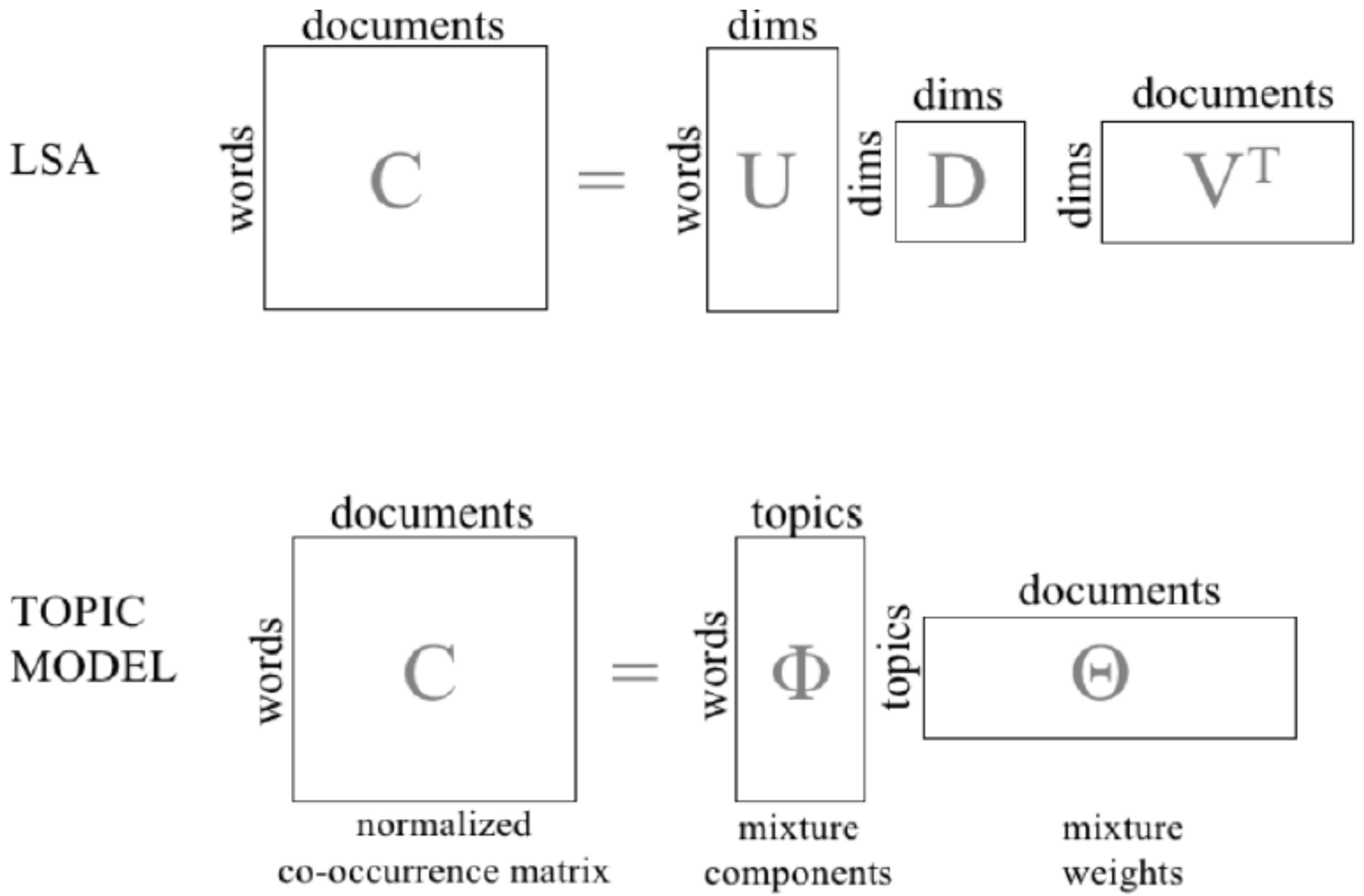
- Has proved to be a valuable tool in many areas of NLP as well as IR
  - summarization
  - cross-language IR
  - topics segmentation
  - text classification
  - question answering
  - And more……

Summary

- Solves synonymy, but does not solve polysemy



Summary



LSA
documents
words
C
=
dims
words
U
dims
dims
D
documents
dims
V^T
TOPIC
MODEL
documents
words
C
=
topics
words
Φ
topics
documents
Θ
normalized
co-occurrence matrix
mixture
components
mixture
weights

Summary

- Ongoing research and extensions include
  - Probabilistic LSA (Hofmann)
  - Iterative Scaling (Ando and Lee)
  - Psychology
    - model of semantic knowledge representation
    - model of semantic word learning

Outline

- LSA algorithm
- Probablistic LSA
- Latent Dirichlet Allocation
- Discussions

probabilistic Latent Semantic Analysis

- pLSA evolved from Latent semantic analysis, adding a sounder probabilistic model
- It was introduced in 1999 by Thomas Hofmann (UAI'99)
- It is an aspect model
- It is related to non-negative matrix factorization (NMF)

probabilistic Latent Semantic Analysis

- Motivation
  - Documents are not related to a single cluster (i.e. aspect)
    - For each z, P(z|d) defines a specific mixture of factors
    - This offers more flexibility, and produces effective modeling
  - Latent Variable model for general co-occurrence data
    - Associate each observation (w, d) with a class variable $z \in Z\{z_1,\ldots,z_K\}$

Probabilities and Bayes rule

- To get the joint probability model

$$P(d, w) = P(d)P(w|d)$$

$$= P(d)\sum_z P(w|z)P(z|d)$$

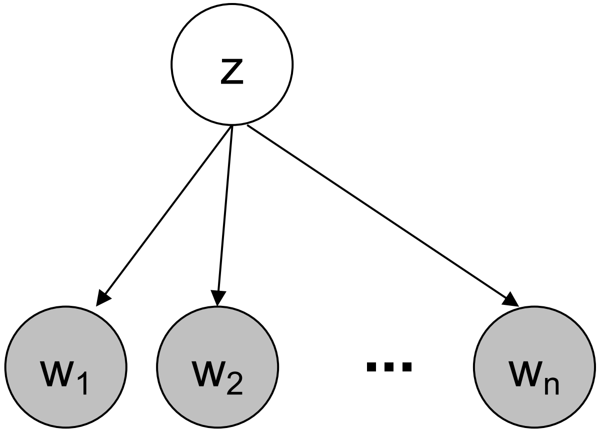
  - (d, w) – assumed to be independent
  - Each word token associated with hidden variable

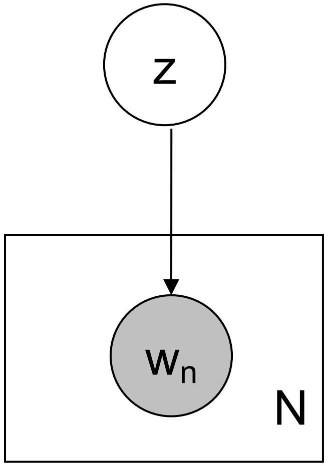
- or

$$P(d, w) = \sum_z P(w|z)P(d|z)P(z)$$

which will lead to different inference process

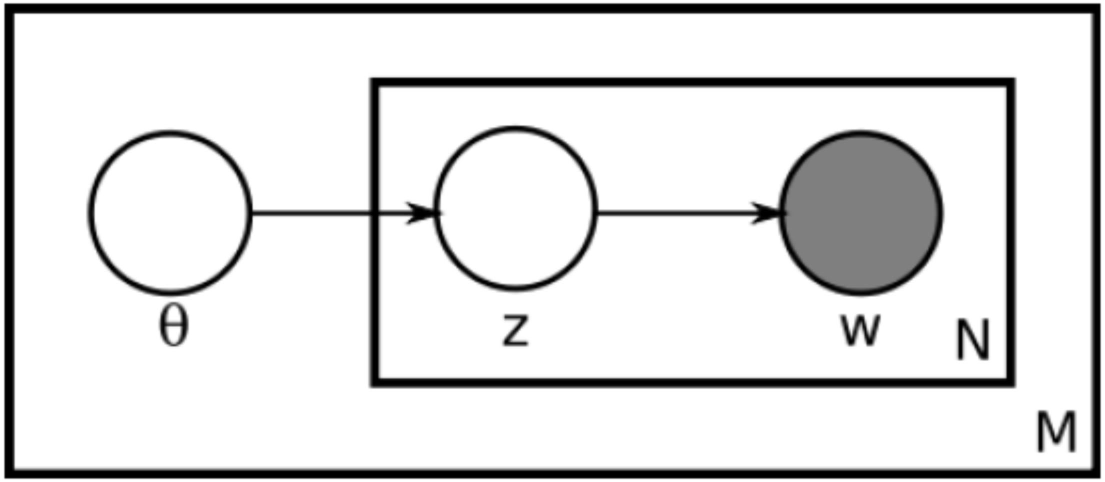
Graphic model





- Nodes are random variables
- Edges denote possible dependence
- Observed variables are shaded
- Plate denote replicated structure

Graphic model of pLSA



- P(z|d) is shared by all words in a document
- P(w|z) is shared by all documents in collection
- It is possible to derive the equations for computing these parameters by Maximum Likelihood

Maximum Likelihood

- The log likelihood of this model is the log probability of the entire collection:

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{j=1}^{M} n(d_i, w_j) \log P(d_i, w_j)$$
$$= \sum_{i=1}^{N} n(d_i) \left[\log P(d_i) + \sum_{j=1}^{M} \frac{n(d_i, w_j)}{n(d_i)} \log \sum_{k=1}^{K} P(w_j \mid z_k) P(z_k \mid d_i) \right]$$

- Which is to be maximized w.r.t. parameters P(z|d) and also P(w|z), subject to the constraints that $\sum_{j=1,\ldots,n} P(w_j|z)=1$ and $\sum_{k=1,\ldots,K} P(z_k|d)=1$

Maximum Likelihood

- Define ϕ_{kj} a distribution of word w_j on topic z_k

$$P(w_j|z_k) = \phi_{k,j}, \quad \sum_{w_j \in \mathcal{V}} \phi_{k,j} = 1$$

- And define θ_{ik} a distribution of topic z_k on document d_i

$$P(z_k|d_i) = \theta_{i,k}, \quad \sum_{z_k \in \mathcal{Z}} \theta_{i,k} = 1$$

- Two sets of parameters,

$$\Phi = [\phi_1, \cdots, \phi_K], \quad z_k \in \mathcal{Z}$$
$$\Theta = [\theta_i, \cdots, \theta_N], \quad d_i \in \mathcal{D}$$

Maximum Likelihood

- Then the log likelihood is

$$
\begin{aligned}
\ell(\Phi, \Theta) &= \sum_{i=1}^{N} \sum_{j=1}^{M} n(d_i, w_j) \log P(d_i, w_j) \\
&= \sum_{i=1}^{N} n(d_i) \left(\log P(d_i) + \sum_{j=1}^{M} \frac{n(d_i, w_j)}{n(d_i)} \log \sum_{k=1}^{K} P(w_j|z_k) P(z_k|d_i) \right) \\
&= \sum_{i=1}^{N} n(d_i) \left(\log P(d_i) + \sum_{j=1}^{M} \frac{n(d_i, w_j)}{n(d_i)} \log \sum_{k=1}^{K} \phi_{k,j} \theta_{i,k} \right)
\end{aligned}
$$

Recall: Expectation Maximization (EM)

- It is a process of iteration which consists of Expectation step and Maximization step with latent variables
- E-Step
  - Expectation step where expectation of the likelihood function is calculated with the current parameter values
- M-Step
  - Update the parameters with the calculated posterior probabilities
  - Find the parameters that maximizes the likelihood function

E Step

- For E step, simply using Bayes Rule, we can obtain

$$
\begin{aligned}
P(z_k|d_i,w_j) &= \frac{P(z_k,d_i,w_j)}{\sum_{l=1}^{K} P(z_l,d_i,w_j)} \\
&= \frac{P(w_j|d_i,z_k)P(z_k|d_i)P(d_i)}{\sum_{l=1}^{K} (P(w_j|d_i,z_l)P(z_l|d_i)P(d_i))} \\
&= \frac{P(w_j|z_k)P(z_k|d_i)}{\sum_{l=1}^{K} P(w_j|z_l)P(z_l|d_i)} \\
&= \frac{\phi_{k,j}\theta_{i,k}}{\sum_{l=1}^{K} \phi_{l,j}\theta_{i,l}}
\end{aligned}
$$

- Then,

$$\ell = \sum_{i=1}^{N}\sum_{j=1}^{M} n(d_i,w_j)\sum_{k=1}^{K} P(z_k|d_i,w_j)\log[\phi_{k,j}\theta_{i,k}]$$

M Step

- For M-step, we need to maximize *L*

$$\ell = \sum_{i=1}^{N} \sum_{j=1}^{M} n(d_i, w_j) \sum_{k=1}^{K} P(z_k|d_i, w_j) \log[\phi_{k,j}\theta_{i,k}]$$

- which needs to be incorporated with other constraints:

$$\sum_{j=1}^{M} \phi_{k,j} = 1$$

$$\sum_{k=1}^{K} \theta_{i,k} = 1$$

- By introducing Lagrange factors:

$$\mathcal{H} = \mathcal{L}^c + \sum_{k=1}^{K} \tau_k \left(1 - \sum_{j=1}^{M} \phi_{k,j}\right) + \sum_{i=1}^{N} \rho_i \left(1 - \sum_{k=1}^{K} \theta_{i,k}\right)$$

M Step

- Let all derivatives equal 0,

$$\sum_{i=1}^{N} n(d_i, w_j) P(z_k|d_i, w_j) - \tau_k \phi_{k,j} = 0, \quad 1 \leq j \leq M, 1 \leq k \leq K$$

$$\sum_{j=1}^{M} n(d_i, w_j) P(z_k|d_i, w_j) - \rho_i \theta_{i,k} = 0, \quad 1 \leq i \leq N, 1 \leq k \leq K$$

- Finally we get:

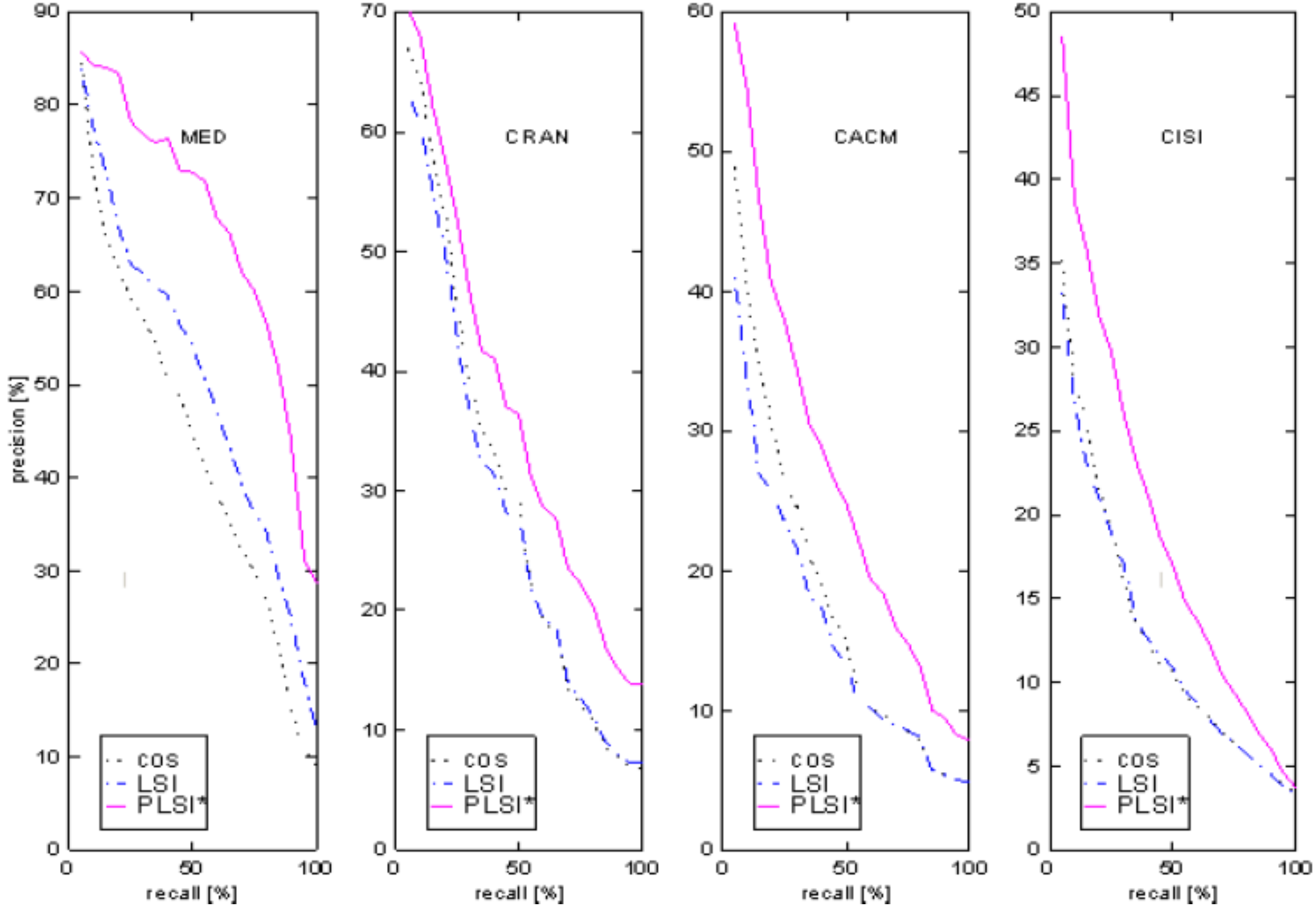
$$\phi_{k,j} = \frac{\sum_{i=1}^{N} n(d_i, w_j) P(z_k|d_i, w_j)}{\sum_{m=1}^{M} \sum_{i=1}^{N} n(d_i, w_m) P(z_k|d_i, w_m)}$$

$$\theta_{i,k} = \frac{\sum_{j=1}^{M} n(d_i, w_j) P(z_k|d_i, w_j)}{n(d_i)}$$

Comparing pLSA and LSA

- LSA and PLSA perform dimensionality reduction
  - In LSA, by keeping only K singular values
  - In pLSA, by having K aspects
- Comparison to SVD
  - U Matrix related to P(d|z) (doc to aspect)
  - V Matrix related to P(z|w) (aspect to term)
  - E Matrix related to P(z)   (aspect strength)
- The main difference is the way the approximation is done
  - pLSA generates a model (aspect model) and maximizes its predictive power
  - Selecting the proper value of K is heuristic in LSA
  - Model selection in statistics can determine optimal K in pLSA

- The performance of a retrieval system based on this model (PLSI) was found superior to that of both the vector space based similarity (cos) and a non-probabilistic latent semantic indexing (LSI) method. (From Th. Hofmann, 2000)



variations of pLSA

- Hierarchical extensions:
  - Asymmetric: MASHA ("Multinomial Asymmetric Hierarchical Analysis")
  - Symmetric: HPLSA ("Hierarchical Probabilistic Latent Semantic Analysis")
- Manifold regularizer:
  - Probabilistic Dyadic Data Analysis with Local and Global Consistency
- Generative models:
  - Latent Dirichlet allocation - adds a Dirichlet prior on the per-document topic distribution, trying to address an often-criticized shortcoming of PLSA, namely that it is not a proper generative model for new documents and at the same time avoid the overfitting problem.

- Handouts:
  - “Text representation and text modeling”
  - LDA and its mathematics