Computational Linguistics

Latent Spaces and Matrix Factorization

Stefan Thater & Dietrich Klakow

FR 4.7 Allgemeine Linguistik (Computerlinguistik)
Universität des Saarlandes

Summer 2013

Goal

Goal:

treat document clustering and word clustering on the same footing (same semantic space)

find low dimensional representations

The word document matrix

Clustering

Document clustering

describe each document by a vector containing the frequencies of the words

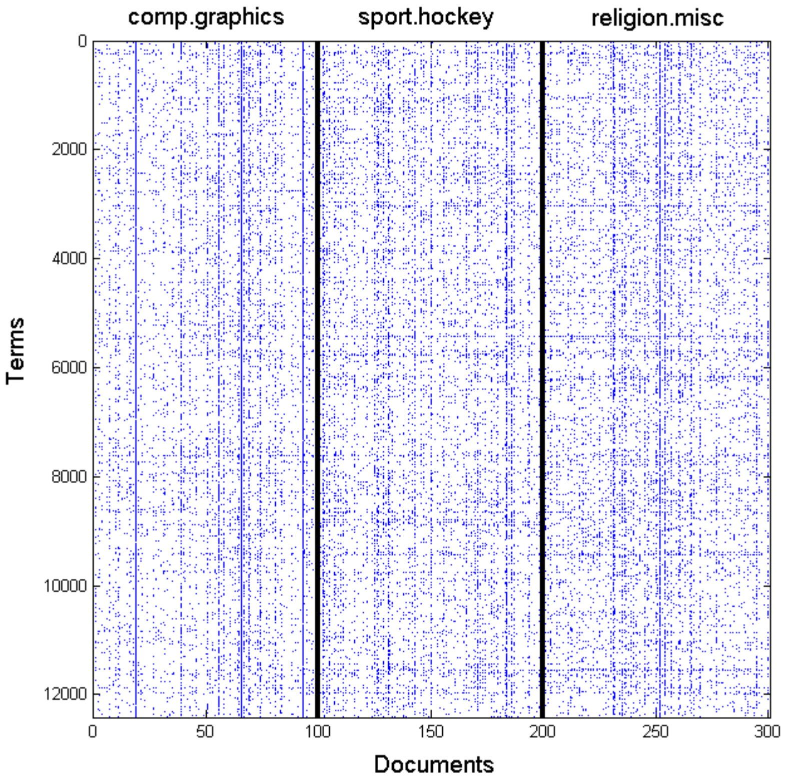
Word clustering

describe each word by a vector containing the frequencies of its occurance in different document

Joint word and document clustering

The word document matrix:

Enter frequency (or tf-idf) for each word and document in a square scheme of numbers (matrix)



Matrices

Matrices

A matrix is an array with two indices

e.g. in a python program this could be `A[i][j]` with i=1..N and j=1...M

When writing, often a subscript notation is used $a_{i,j}$

or a square scheme:

$$A = \begin{pmatrix} a_{1,1} & \dots & a_{1,M} \\ \dots & a_{i,j} & \dots \\ a_{N,1} & \dots & a_{N,M} \end{pmatrix}$$

Specific example of a 2x3 matrix

$$A = \begin{pmatrix} 2 & -5 & 0.5 \\ -2 & 0.1 & -8 \end{pmatrix}$$

The transpose of a matrix

The two indices are swapped

e.g. in a python program this could be `At[j][i]=A[i][j]` for i=1..N and j=1...M

for the matrices from the previous slide we have:

$$A^t = \begin{pmatrix} a_{1,1} & \dots & a_{1,N} \\ \dots & a_{j,i} & \dots \\ a_{M,1} & \dots & a_{M,N} \end{pmatrix}$$

Specific example of a 2x3 matrix

$$A = \begin{pmatrix} 2 & -5 & 0.5 \\ -2 & 0.1 & -8 \end{pmatrix}$$

What is A^t

Product of two matrices

The elements of a product matrix can be calculated in a python program by

```
for i in range(1,N+1):
    for j in range(1,M+1):
           for k in range(1,K+1):
                  C[i][j] = A[i][k]*B[k][j]
```

In math notation $C = A \cdot B$

with $$c_{i,j} = \sum_{k=1}^{K} a_{i,k} b_{k,j}$$

Example see black board

Unit matrix

Unit matrix: the element are the indicator function

$$a_{i,j} = \delta_{i,j}$$

Example:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Often the unit matrix is denoted by a 1

Orthogonal matrices

a matrix A is orthogonal if

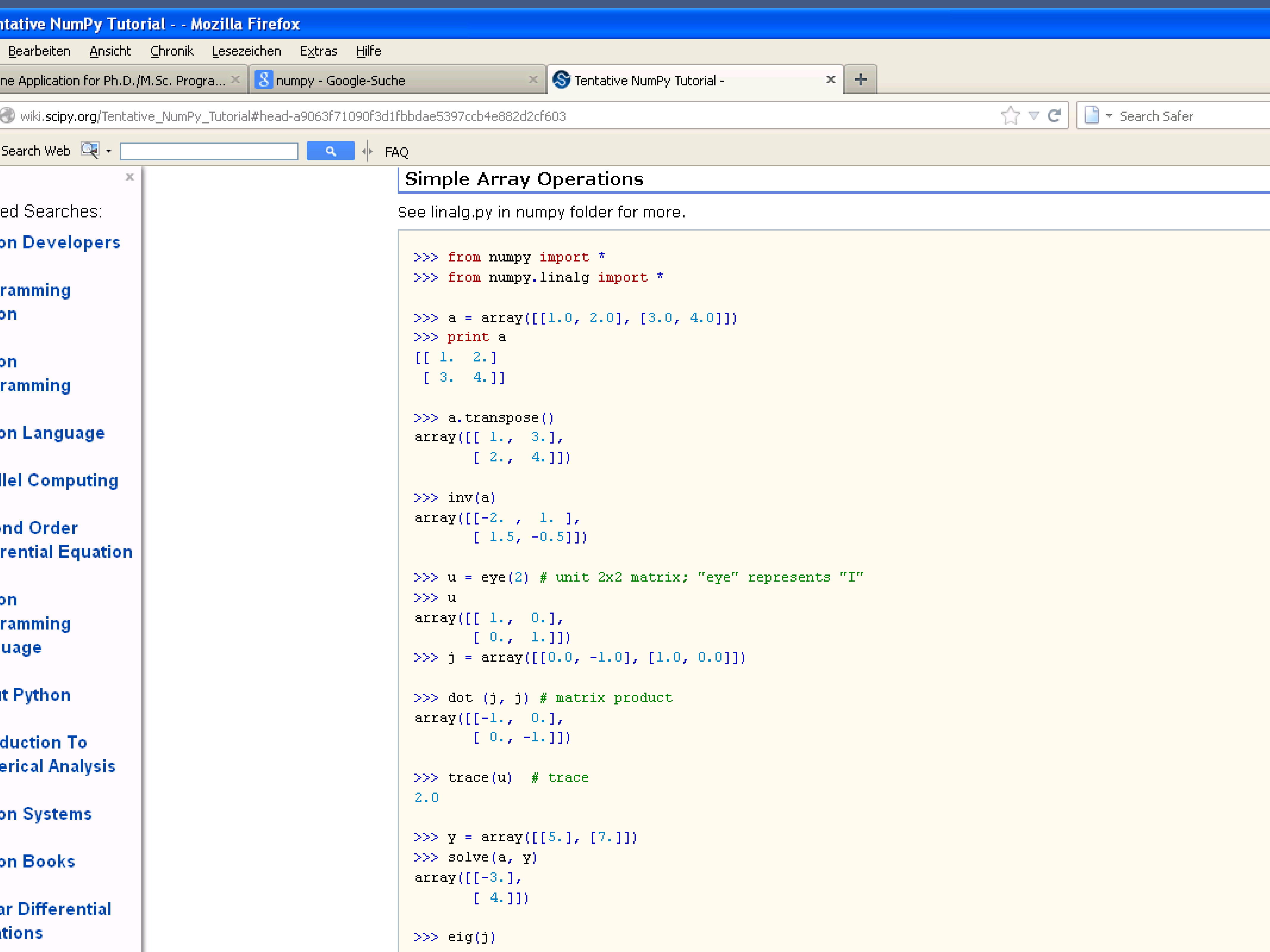
$$1 = A^t \cdot A$$

Is the following matrix orthogonal:

$$A = \begin{pmatrix} 0.96 & -0.28 \\ 0.28 & 0.96 \end{pmatrix}$$

Matrices in python

See http://wiki.scipy.org/Tentative_NumPy_Tutorial#head-a9063f71090f3d1fbbdae5397ccb4e882d2cf603



Simple Array Operations

See linalg.py in numpy folder for more.

```
>>> from numpy import *
>>> from numpy.linalg import *

>>> a = array([[1.0, 2.0], [3.0, 4.0]])
>>> print a
[[ 1.  2.]
 [ 3.  4.]]

>>> a.transpose()
array([[ 1.,  3.],
       [ 2.,  4.]])

>>> inv(a)
array([[-2. ,  1. ],
       [ 1.5, -0.5]])

>>> u = eye(2) # unit 2x2 matrix; "eye" represents "I"
>>> u
array([[ 1.,  0.],
       [ 0.,  1.]])
>>> j = array([[0.0, -1.0], [1.0, 0.0]])

>>> dot (j, j) # matrix product
array([[-1.,  0.],
       [ 0., -1.]])

>>> trace(u)  # trace
2.0

>>> y = array([[5.], [7.]])
>>> solve(a, y)
array([[-3.],
       [ 4.]])

>>> eig(j)
```

Latent Semantic Analysis (LSA)

This section mostly follows Manning and Schütze Chapter 15

Singular Value Decomposition

Decompose A such that

$$\tilde{A} = TSD^t$$

With $|\tilde{A} - A|^2$ minimal

and

$$T^t \cdot T = 1 \qquad D^t \cdot D = 1$$

A a t by d matrix T a t by n matrix

S a n by n matrix D a d by n matrix

An artificial Example of Singular Value Decomposition

Is

$$T = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{pmatrix} \qquad S = \begin{pmatrix} 2\sqrt{2} \end{pmatrix} \qquad D = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ -\frac{1}{2} \\ -\frac{1}{2} \end{pmatrix}$$

An SVD of

$$A = \begin{pmatrix} 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \end{pmatrix}$$

More realistic Example

(from Manning and Schütze)

Decompose

$$A = \left(\begin{array}{l|cccccc} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline \text{cosmonaut} & 1 & 0 & 1 & 0 & 0 & 0 \\ \text{astronaut} & 0 & 1 & 0 & 0 & 0 & 0 \\ \text{moon} & 1 & 1 & 0 & 0 & 0 & 0 \\ \text{car} & 1 & 0 & 0 & 1 & 1 & 0 \end{array}\right)$$

More realistic Example

(from Manning and Schütze)

$$D^{t} = \left(\begin{array}{l|rrrrrr} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline \text{Dimension 1} & -0.75 & -0.28 & -0.20 & -0.45 & -0.33 & -0.12 \\ \text{Dimension 2} & -0.29 & -0.53 & -0.19 & 0.63 & 0.22 & 0.41 \\ \text{Dimension 3} & 0.28 & -0.75 & 0.45 & -0.20 & 0.12 & -0.33 \\ \text{Dimension 4} & 0.00 & 0.00 & 0.58 & 0.00 & -0.58 & 0.58 \\ \text{Dimension 5} & -0.53 & 0.29 & 0.63 & 0.19 & 0.41 & -0.22 \end{array}\right)$$

$$T^{t} = \left(\begin{array}{l|rrrrr} & \text{cosm.} & \text{astr.} & \text{moon} & \text{car} & \text{truck} \\ \hline \text{Dimension 1} & -0.44 & -0.13 & -0.48 & -0.70 & -0.26 \\ \text{Dimension 2} & -0.30 & -0.33 & -0.51 & 0.35 & 0.65 \\ \text{Dimension 3} & 0.57 & -0.59 & -0.37 & 0.15 & -0.41 \\ \text{Dimension 4} & 0.58 & 0.00 & 0.00 & -0.58 & 0.58 \\ \text{Dimension 5} & 0.25 & 0.73 & -0.61 & 0.16 & -0.09 \end{array}\right)$$

More realistic Example

(from Manning and Schütze)

$$S = \begin{pmatrix} 2.16 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 1.59 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 1.28 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.39 \end{pmatrix}$$

Document-Document Similarity

Rewrite A

$$A = \begin{pmatrix} \vec{d}_1 & \vec{d}_2 & \ldots & \vec{d}_d \end{pmatrix}$$

with $\vec{d}_j$ a vector

with word frequencies of the j-th document

Similarity of i-th document with j-th document $\vec{d}_i^{\,t}\vec{d}_j$

All document-document similarities $A^t A$

Document-Document Similarity

Rewrite $\tilde{A}^t \tilde{A} =$

$$= (TSD^t)^t TSD^t$$

$$= DS^t T^t TSD^t$$

$$= DS^t SD^t$$

$$= (SD^t)^t SD^t$$

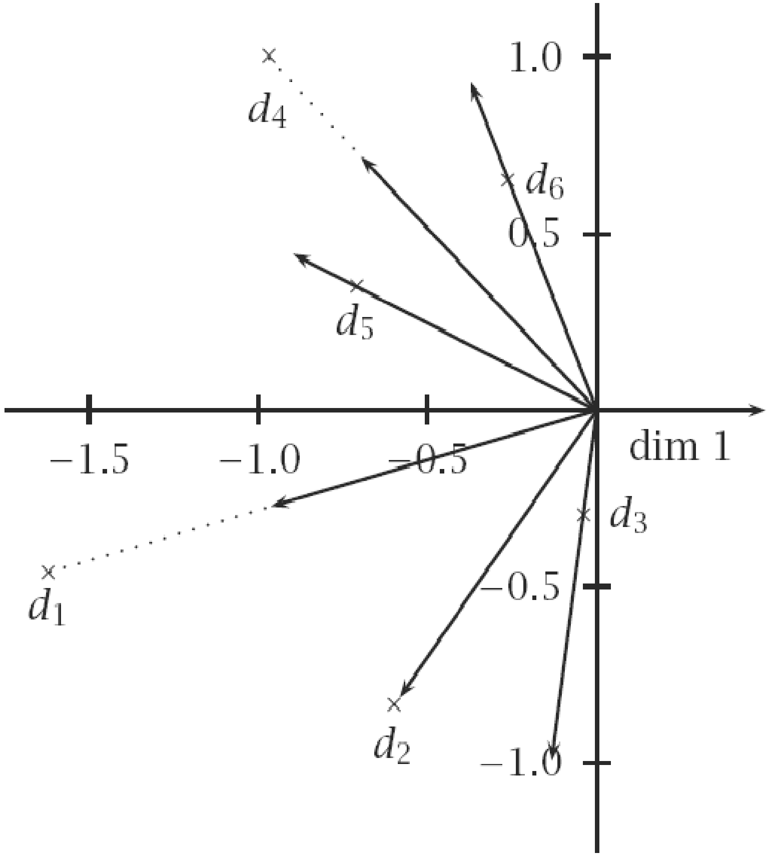
Measure similarity in subspace defined by SD^t

More realistic Example

(from Manning and Schütze)

Result for SD^t

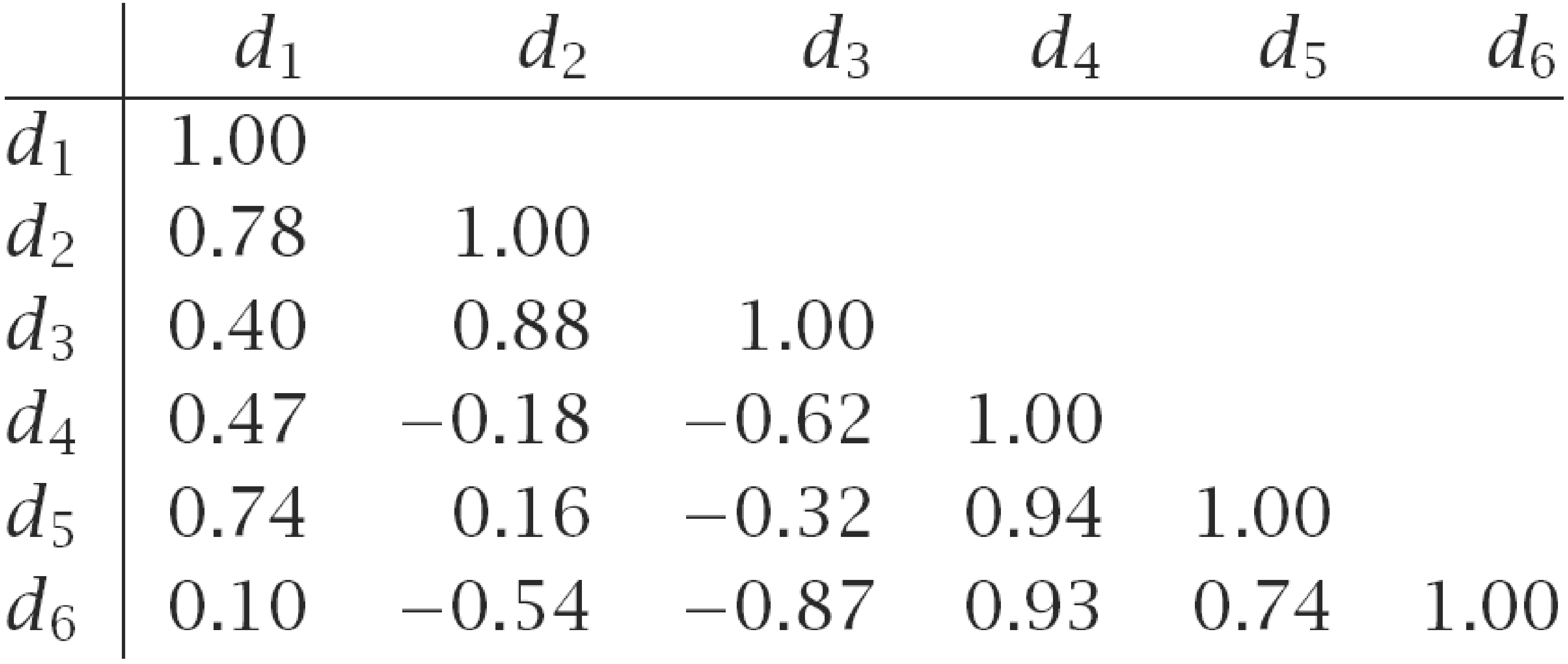
	d_1	d_2	d_3	d_4	d_5	d_6
Dimension 1	−1.62	−0.60	−0.04	−0.97	−0.71	−0.26
Dimension 2	−0.46	−0.84	−0.30	1.00	0.35	0.65



More realistic Example

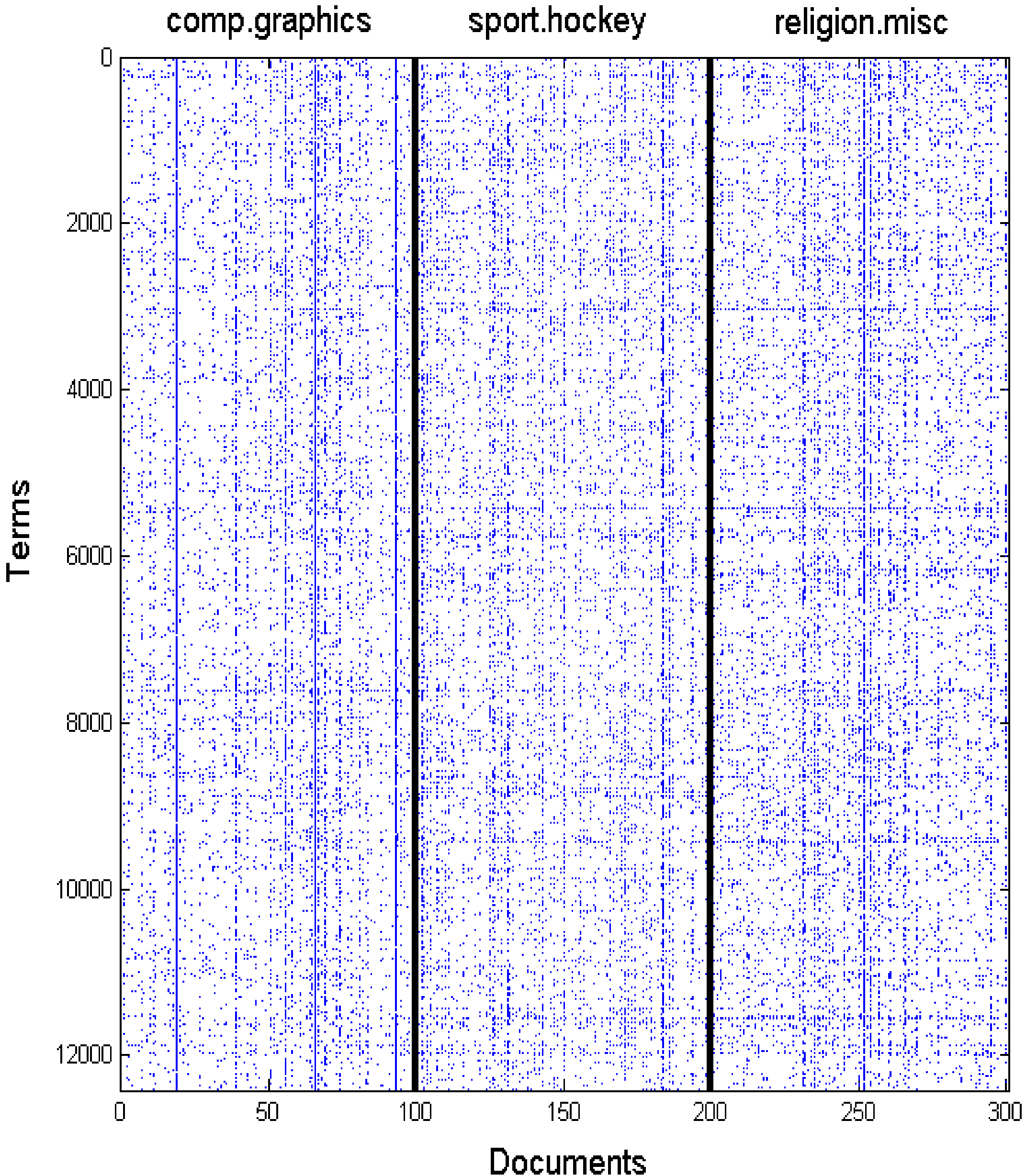
(from Manning and Schütze)

Decompose A such that

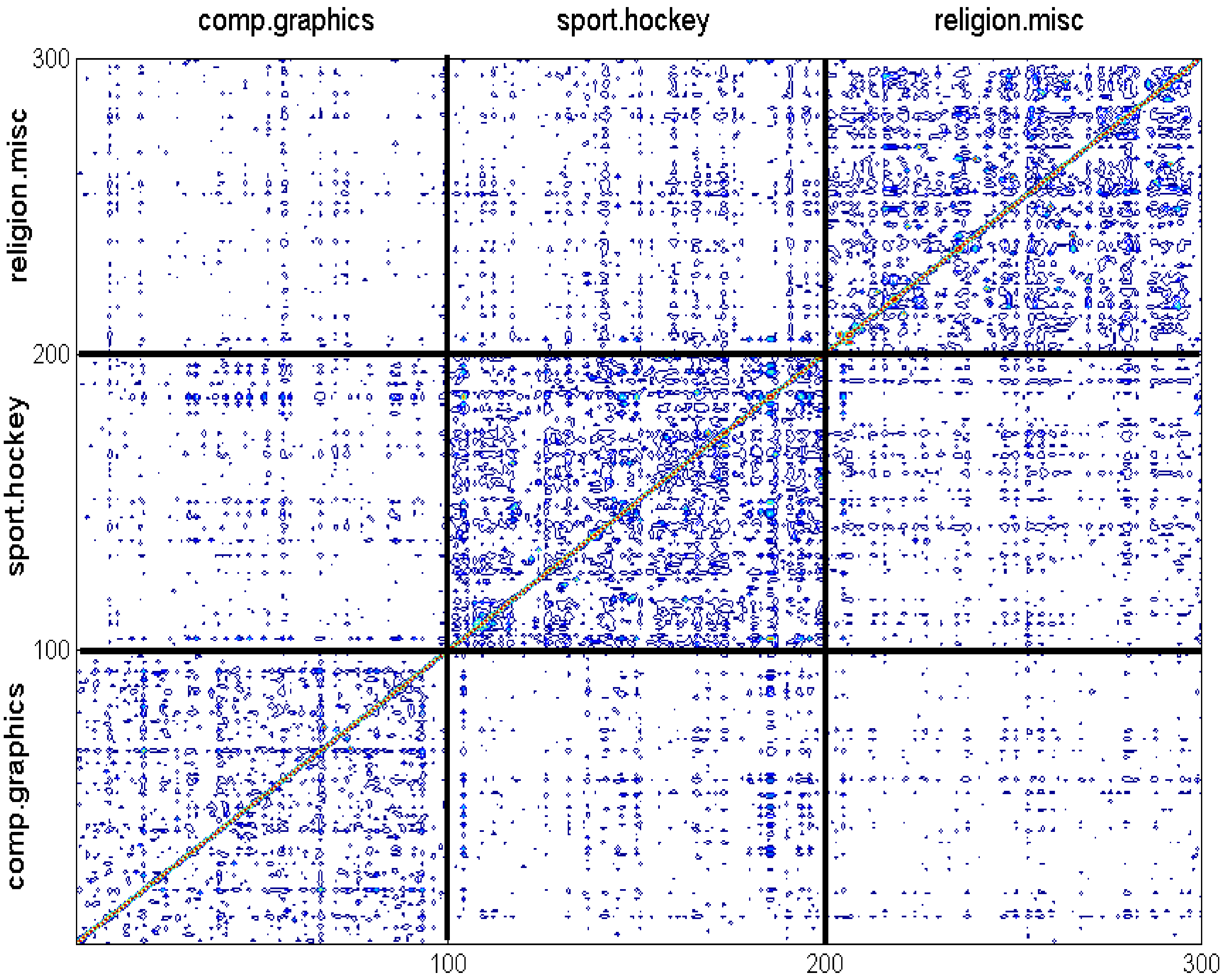


	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1.00					
d_2	0.78	1.00				
d_3	0.40	0.88	1.00			
d_4	0.47	−0.18	−0.62	1.00		
d_5	0.74	0.16	−0.32	0.94	1.00	
d_6	0.10	−0.54	−0.87	0.93	0.74	1.00

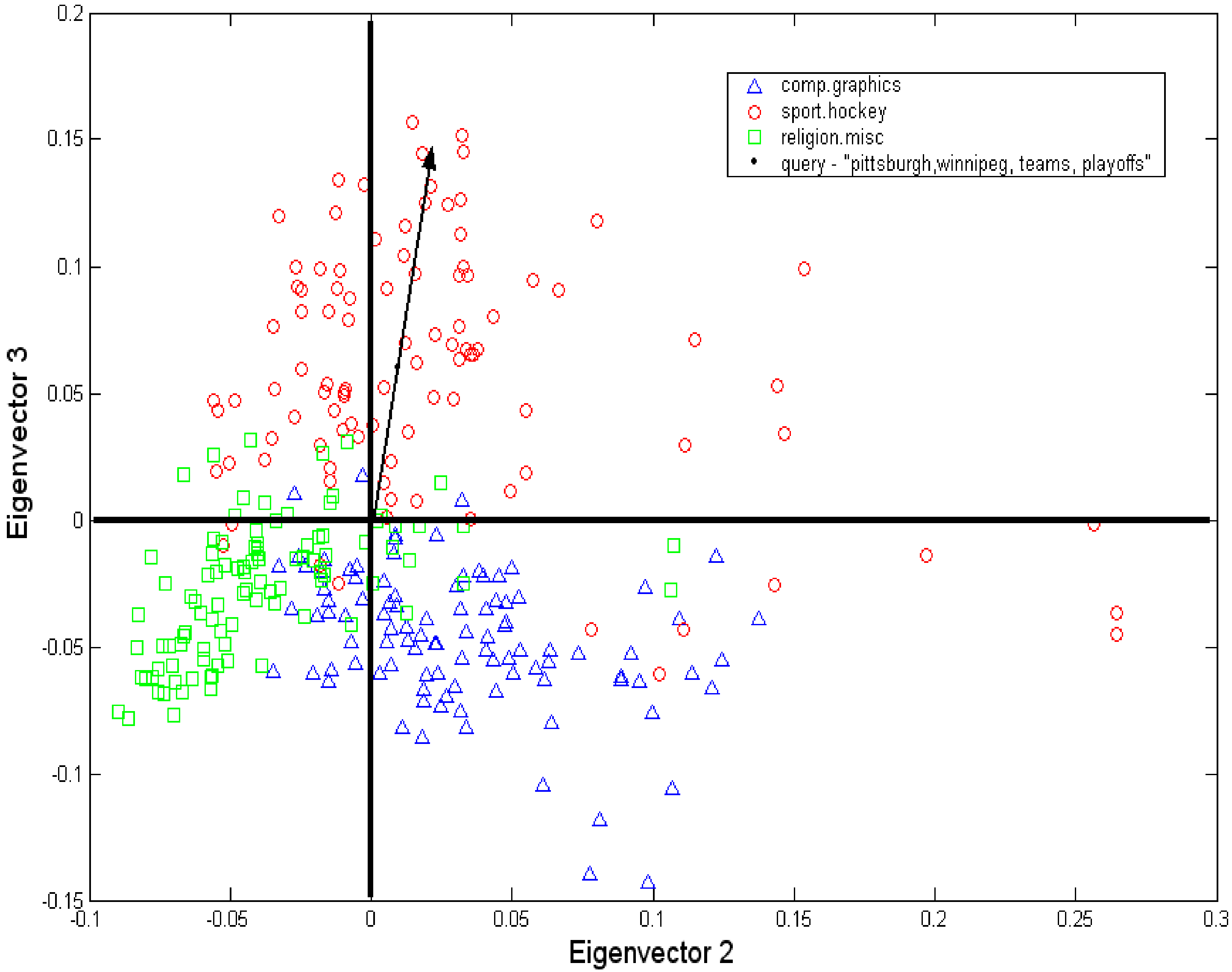
An even more realistic example



An even more realistic example
Document-Document Similarity



Representation for Documents in 2 dimensional Subspace



Term-Term Similarity

Rewrite $\tilde{A}\tilde{A}^t =$

$$= (TSD^t)(TSD^t)^t$$

$$= TSD^tDS^tT^t$$

$$= TS^tST^t$$

$$= (TS)(TS)^t$$

Measure similarity in subspace defined by TS

Task

How does your programming language support SVD

Do some internet search (~10 minutes)

Report your findings

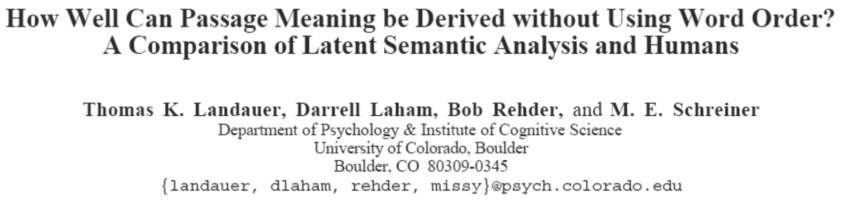
Homework

See sheet

LSA Performance

- LSA consistently improves recall on standard test collections (precision/recall generally improved)
- Variable performance on larger TREC collections
- Dimensionality of Latent Space – a magic number – 300 – 1000 seems to work fine – no satisfactory way of assessing value.
- Computational cost high

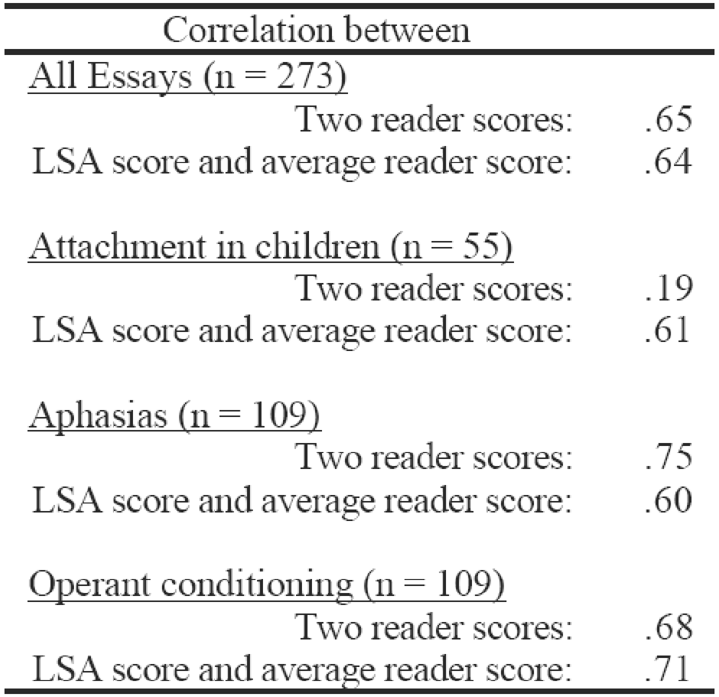
Application (by Landauer et. Al)



How Well Can Passage Meaning be Derived without Using Word Order?
A Comparison of Latent Semantic Analysis and Humans

Thomas K. Landauer, Darrell Laham, Bob Rehder, and **M. E. Schreiner**
Department of Psychology & Institute of Cognitive Science
University of Colorado, Boulder
Boulder, CO 80309-0345
{landauer, dlaham, rehder, missy}@psych.colorado.edu

Rate essay by similarity to existing ones
Measure correlation with human rating



Correlation between	
All Essays (n = 273)	
Two reader scores:	.65
LSA score and average reader score:	.64
Attachment in children (n = 55)	
Two reader scores:	.19
LSA score and average reader score:	.61
Aphasias (n = 109)	
Two reader scores:	.75
LSA score and average reader score:	.60
Operant conditioning (n = 109)	
Two reader scores:	.68
LSA score and average reader score:	.71

Table 2: Psychology essay results.

Conclusion: drop the right key-words and you are set

Probabilistic Latent Semantic Analysis (PLSA)

Motivation

- Does orthogonally matter?
- Wouldn’t a sound statistical foundation be better?

PLSA

Likelihood of document

$$P(doc) = P(term_1 \mid doc)P(term_2 \mid doc)\ldots P(term_L \mid doc)$$

Introduce term-frequency matrix X

$$\prod_{l=1}^{L} P(term_l \mid doc) = \prod_{t=1}^{T} P(term_t \mid doc)^{A(term_t\ ,doc)}$$

PLSA

Introduce hidden topic

$$P(term_t \mid doc) = \sum_{k=1}^{K} P(term_t \mid topic_k)P(topic_k \mid doc)$$

Shorthand t=term_t

$$P(t \mid doc) = \sum_{k=1}^{K} P(t \mid k)P(k \mid doc)$$

Relation to LSA?

Likelihood of document

$$P(doc) = \prod_{t=1}^{T} \left\{ \sum_{k=1}^{K} P(t \mid k)P(k \mid doc) \right\}^{A(t,doc)}$$

PLSA: training

Training objective function

$$\sum_{d=1}^{N} \log P(d) = \sum_{d=1}^{N} \sum_{t=1}^{T} A(t,d) \log \sum_{k=1}^{K} P(t \mid k) P(k \mid d)$$

which is to be maximised w.r.t. parameters $P(t \mid k)$ and then also $P(k \mid d)$,

subject to the constraints that $\sum_{t=1}^{T} P(t \mid k) = 1$ and $\sum_{k=1}^{K} P(k \mid d) = 1$.

PLSA: training

Update term-topic matrix

$$P1(t,k) \leftarrow P1(t,k) \sum_{d=1}^{N} \frac{A(t,d)}{\sum_{k=1}^{K} P1(t,k) P2(k,d)} P2(k,d)$$

$$P1(t,k) \leftarrow \frac{P1(t,k)}{\sum_{t=1}^{T} P1(t,k)}$$

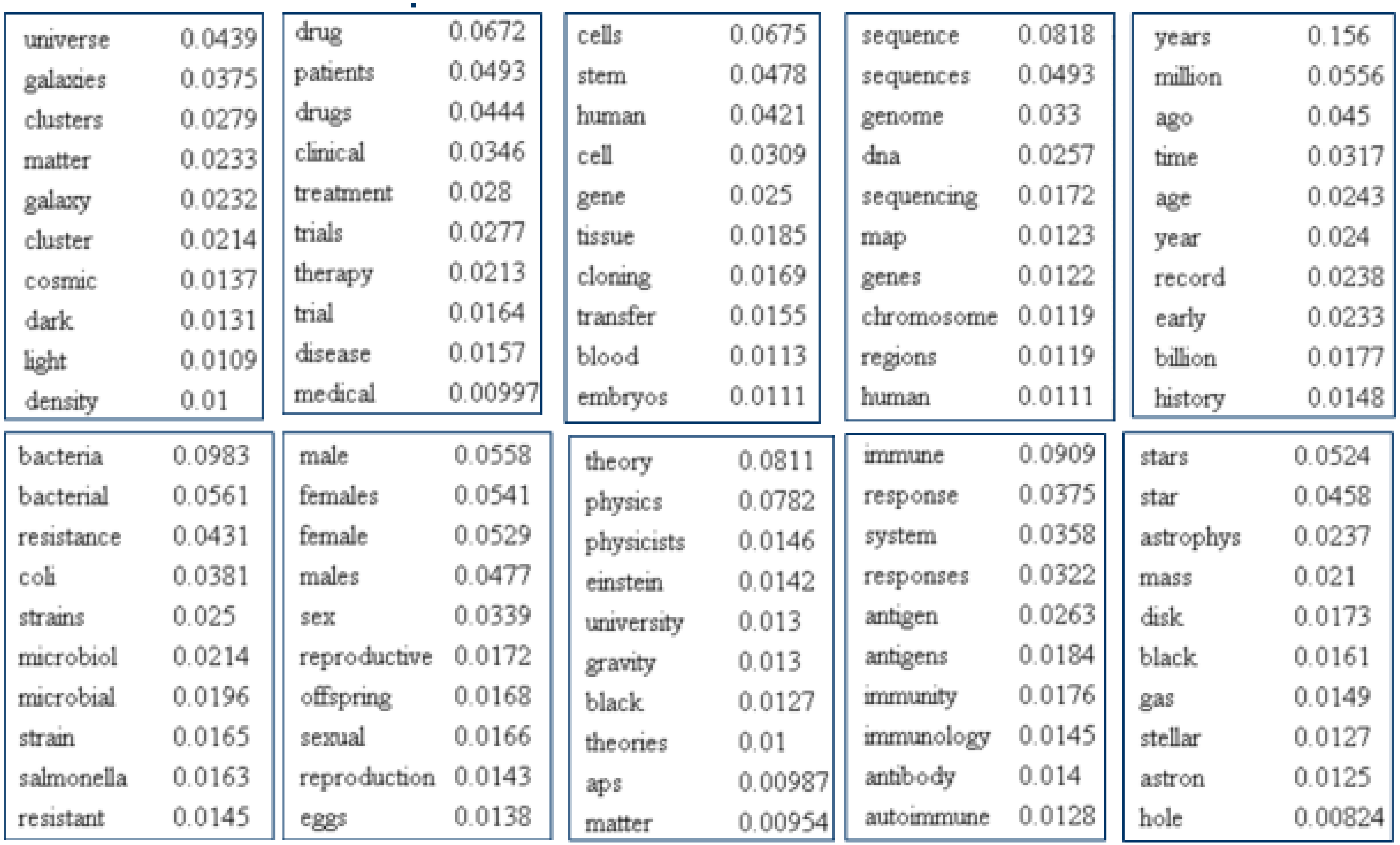
Update topic-document matrix

$$P2(k,d) \leftarrow P2(k,d) \sum_{t=1}^{T} \frac{A(t,d)}{\sum_{k=1}^{K} P1(t,k) P2(k,d)} P1(t,k)$$

$$P2(k,d) \leftarrow \frac{P2(k,d)}{\sum_{k=1}^{K} P2(k,d)}$$

PLSA

P(t|k) for some topics



universe	0.0439
galaxies	0.0375
clusters	0.0279
matter	0.0233
galaxy	0.0232
cluster	0.0214
cosmic	0.0137
dark	0.0131
light	0.0109
density	0.01

drug	0.0672
patients	0.0493
drugs	0.0444
clinical	0.0346
treatment	0.028
trials	0.0277
therapy	0.0213
trial	0.0164
disease	0.0157
medical	0.00997

cells	0.0675
stem	0.0478
human	0.0421
cell	0.0309
gene	0.025
tissue	0.0185
cloning	0.0169
transfer	0.0155
blood	0.0113
embryos	0.0111

sequence	0.0818
sequences	0.0493
genome	0.033
dna	0.0257
sequencing	0.0172
map	0.0123
genes	0.0122
chromosome	0.0119
regions	0.0119
human	0.0111

years	0.156
million	0.0556
ago	0.045
time	0.0317
age	0.0243
year	0.024
record	0.0238
early	0.0233
billion	0.0177
history	0.0148

bacteria	0.0983
bacterial	0.0561
resistance	0.0431
coli	0.0381
strains	0.025
microbiol	0.0214
microbial	0.0196
strain	0.0165
salmonella	0.0163
resistant	0.0145

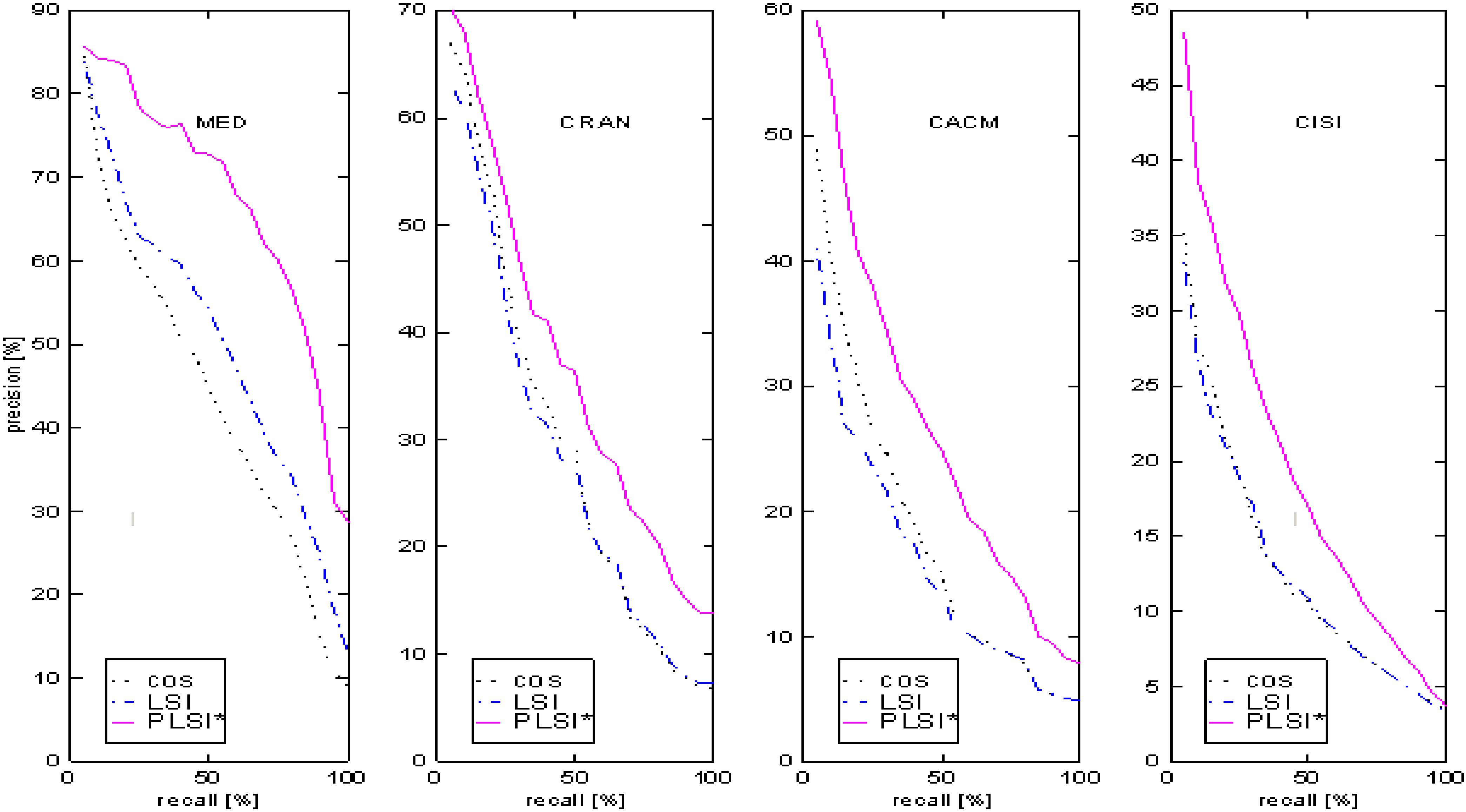
male	0.0558
females	0.0541
female	0.0529
males	0.0477
sex	0.0339
reproductive	0.0172
offspring	0.0168
sexual	0.0166
reproduction	0.0143
eggs	0.0138

theory	0.0811
physics	0.0782
physicists	0.0146
einstein	0.0142
university	0.013
gravity	0.013
black	0.0127
theories	0.01
aps	0.00987
matter	0.00954

immune	0.0909
response	0.0375
system	0.0358
responses	0.0322
antigen	0.0263
antigens	0.0184
immunity	0.0176
immunology	0.0145
antibody	0.014
autoimmune	0.0128

stars	0.0524
star	0.0458
astrophys	0.0237
mass	0.021
disk	0.0173
black	0.0161
gas	0.0149
stellar	0.0127
astron	0.0125
hole	0.00824

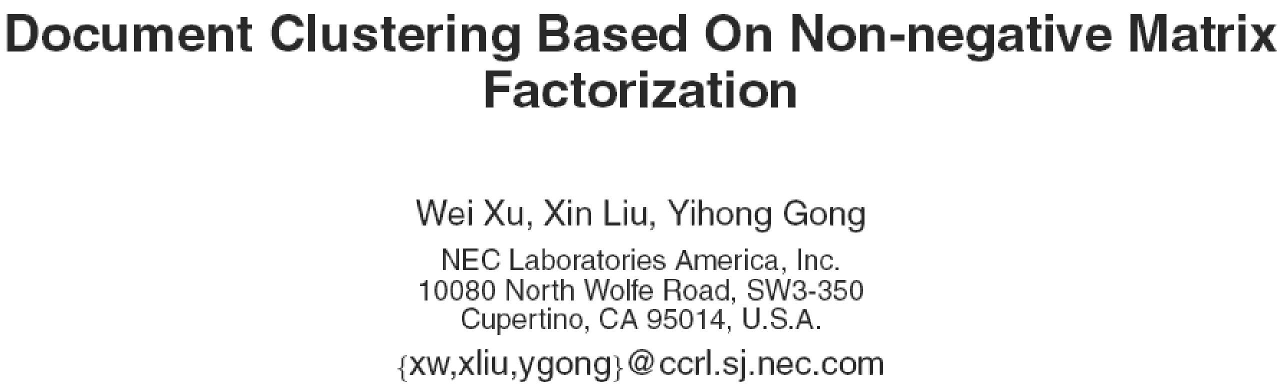
Comparison LSA and PLSA



From Th. Hofmann, 2000

Non-negative Matrix Factorization

See:



Document Clustering Based On Non-negative Matrix Factorization

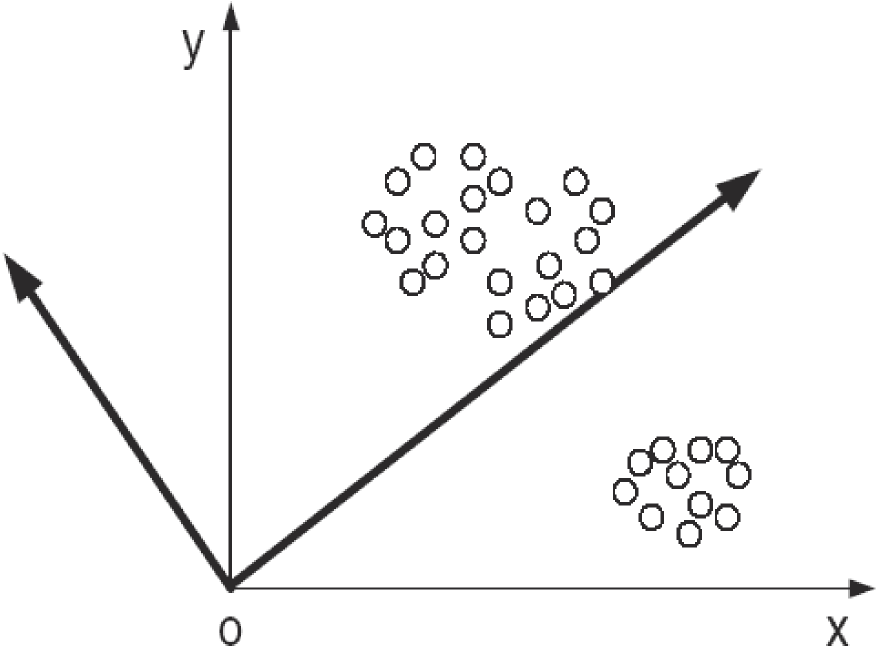
Wei Xu, Xin Liu, Yihong Gong

NEC Laboratories America, Inc.
10080 North Wolfe Road, SW3-350
Cupertino, CA 95014, U.S.A.

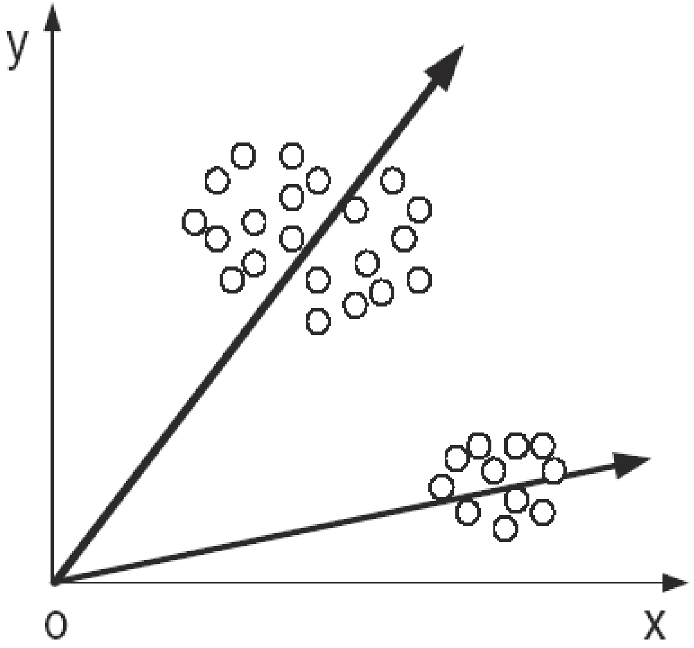
{xw,xliu,ygong}@ccrl.sj.nec.com

NMF: idea

- Find space that separates clusters better



Directions found by LSI



Directions found by NMF

NMF: the model

- Decompostion of a non-negaitve matrix X in two matrices W and H both non-negative

$$A = WH$$

- A: N x M – data matrix
- W: N x R – source matrix
- H: R x M – mixture matrix

NMF: the model

- Determine W and H such that the product WH is as close as possible to A

- W and H are bound to be non-negative values

- Possible metrics
  - Kullback-Leibler-Divergenz

    $$D(A|WH)$$

  - Frobenius-Norm

    $$\frac{1}{2}|A-WH|^2$$

NMF: training

Update

$$H_{ab} = H_{ab} \frac{\left(W^t A\right)_{ab}}{\left(W^t W H\right)_{ab} + \varepsilon}$$

$$W_{ab} = W_{ab} \frac{\left(A H^t\right)_{ab}}{\left(W H H^t\right)_{ab} + \varepsilon}$$

Relation to update
From PLSA?

In case the denominator vanishes, add a small number

Homework

Implement NMF for the matrix from the last lecture

Summary

Ways to find latent “semantic” spaces:

- LSA
- PLSA
- NMF

Similar factorizations

Different target functions and constraints