Computational Linguistics

Latent Spaces and Matrix Factorization

Dietrich Klakow

FR 4.7 Allgemeine Linguistik (Computerlinguistik)
Universität des Saarlandes
Summer 2012

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

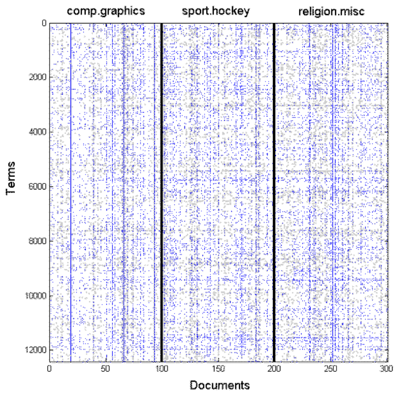
Word clustering

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

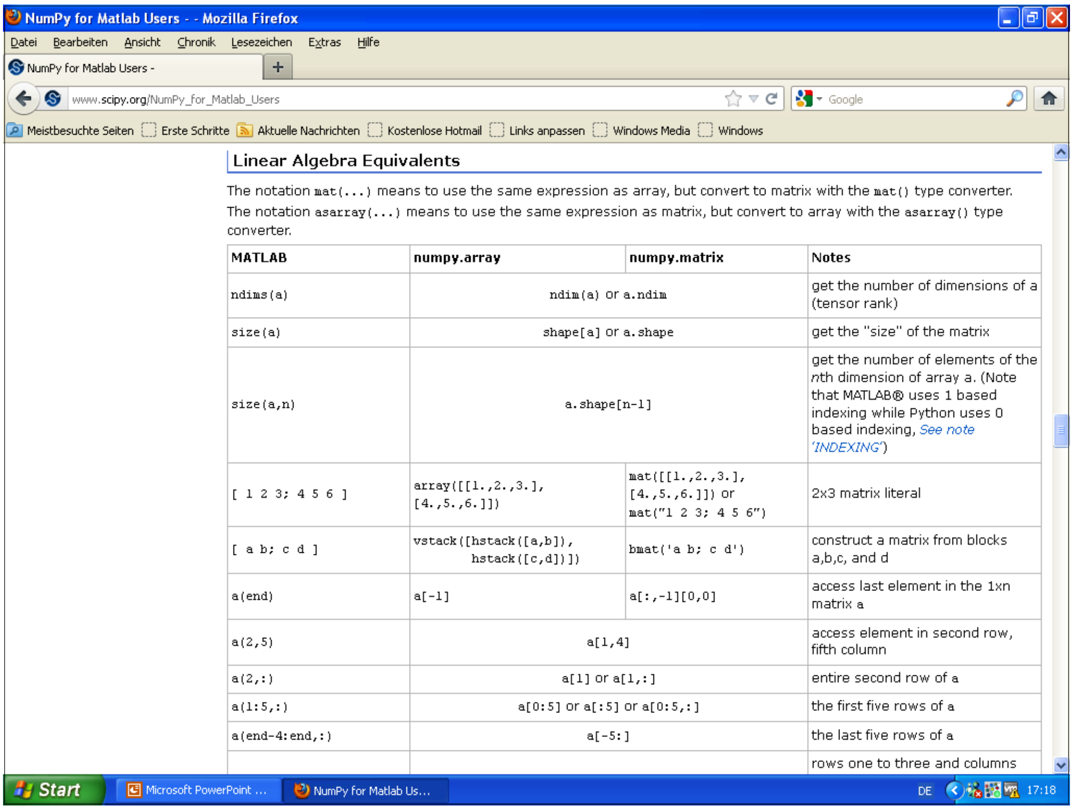
Orthogonal matrices

a matrix A is orthogonal if

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

Is the following matrix orthogonal:

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



NumPy for Matlab Users - - Mozilla Firefox

www.scipy.org/NumPy_for_Matlab_Users

Linear Algebra Equivalents

The notation mat(...) means to use the same expression as array, but convert to matrix with the mat() type converter.
The notation asarray(...) means to use the same expression as matrix, but convert to array with the asarray() type converter.

MATLAB	numpy.array	numpy.matrix	Notes
ndims(a)	ndim(a) or a.ndim		get the number of dimensions of a (tensor rank)
size(a)	shape[a] or a.shape		get the "size" of the matrix
size(a,n)	a.shape[n-1]		get the number of elements of the nth dimension of array a. (Note that MATLAB® uses 1 based indexing while Python uses 0 based indexing, See note 'INDEXING')
[1 2 3; 4 5 6]	array([[1.,2.,3.], [4.,5.,6.]])	mat([[1.,2.,3.], [4.,5.,6.]]) or mat("1 2 3; 4 5 6")	2x3 matrix literal
[a b; c d]	vstack([hstack([a,b]), hstack([c,d])])	bmat('a b; c d')	construct a matrix from blocks a,b,c, and d
a(end)	a[-1]	a[:,-1][0,0]	access last element in the 1xn matrix a
a(2,5)	a[1,4]		access element in second row, fifth column
a(2,:)	a[1] or a[1,:]		entire second row of a
a(1:5,:)	a[0:5] or a[:5] or a[0:5,:]		the first five rows of a
a(end-4:end,:)	a[-5:]		the last five rows of a
			rows one to three and columns

Latent Semantic Analysis (LSA)

This section mostly follows Manning and Schütze Chapter 1

Singular Value Decomposition

Decompose A such that

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

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

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

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

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

An SVD of

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

More realistic Example

(from Manning and Schütze)

Decompose

$A =$

	d_1	d_2	d_3	d_4	d_5	d_6
cosmonaut	1	0	1	0	0	0
astronaut	0	1	0	0	0	0
moon	1	1	0	0	0	0
car	1	0	0	1	1	0

More realistic Example
(from Manning and Schütze)

$D^t =$

	d_1	d_2	d_3	d_4	d_5	d_6
Dimension 1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
Dimension 2	−0.29	−0.53	−0.19	0.63	0.22	0.41
Dimension 3	0.28	−0.75	0.45	−0.20	0.12	−0.33
Dimension 4	0.00	0.00	0.58	0.00	−0.58	0.58
Dimension 5	−0.53	0.29	0.63	0.19	0.41	−0.22

$T^t =$

	cosm.	astr.	moon	car	truck
Dimension 1	−0.44	−0.13	−0.48	−0.70	−0.26
Dimension 2	−0.30	−0.33	−0.51	0.35	0.65
Dimension 3	0.57	−0.59	−0.37	0.15	−0.41
Dimension 4	0.58	0.00	0.00	−0.58	0.58
Dimension 5	0.25	0.73	−0.61	0.16	−0.09

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

$A = (\vec{d}_1 \quad \vec{d}_2 \quad \dots \quad \vec{d}_d)$

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

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

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

$= DS^t SD^t$

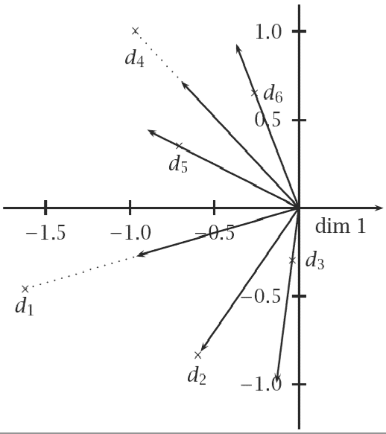
$= (SD^t)^t SD^t$

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

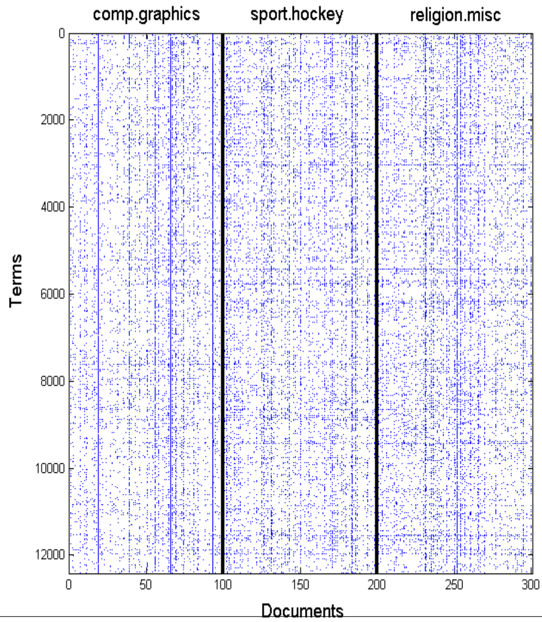


More realistic Example
(from Manning and Schütze)

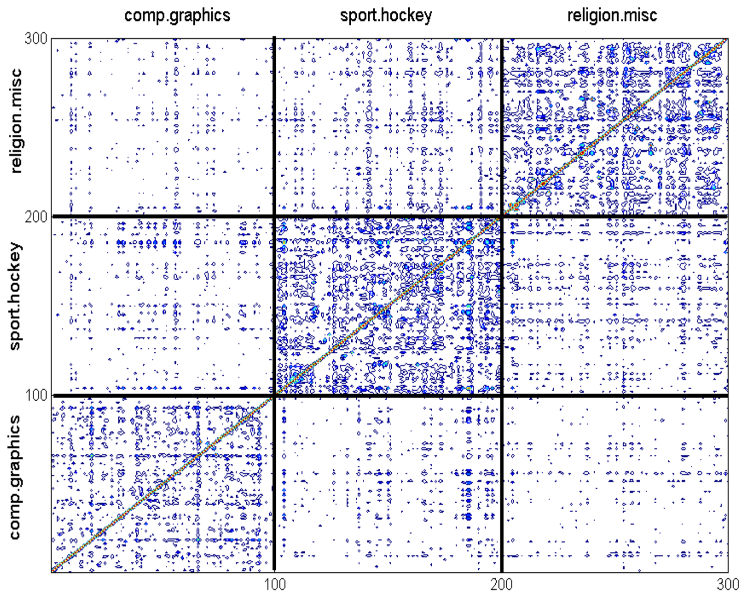
Decompose A such that

	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

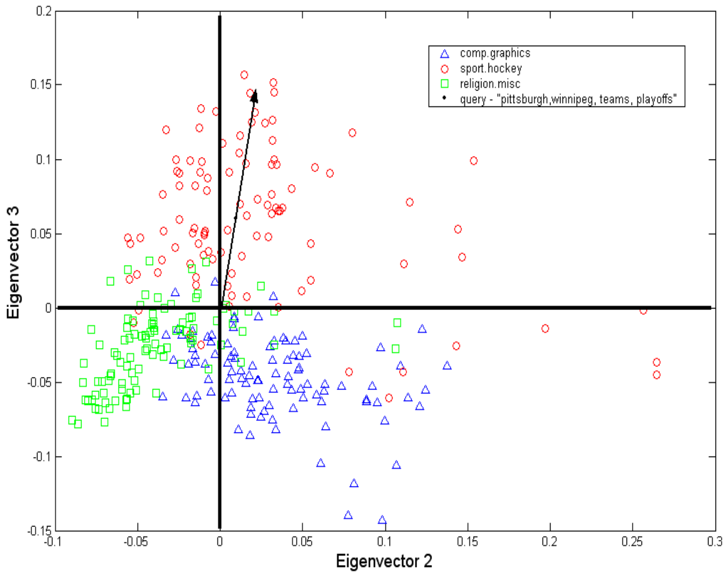
An even more realistic example



An even more realistic example
Document-Document Similarity



Representation for Documents in 2 dimensional Subspace



Term-Term Similarity

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

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

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

$$= TS^t ST^t$$

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

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.

nclusion: 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

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

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

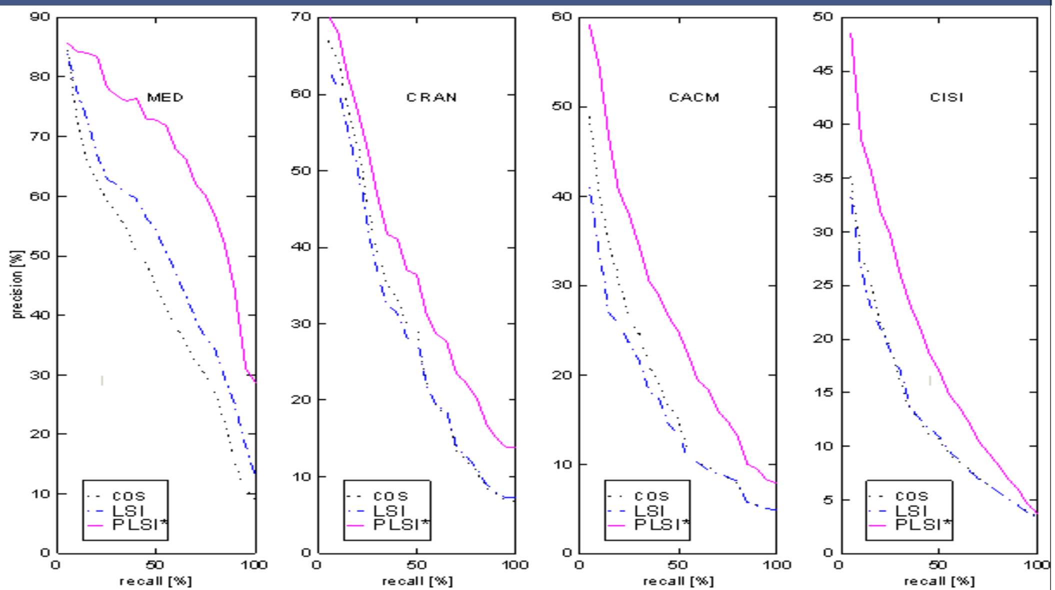
PLSA

P(t|k) for some

universe	0.0439	drug	0.0672	cells	0.0675	sequence	0.0818	years	0.156
galaxies	0.0375	patients	0.0493	stem	0.0478	sequences	0.0493	million	0.0556
clusters	0.0279	drugs	0.0444	human	0.0421	genome	0.033	ago	0.045
matter	0.0233	clinical	0.0346	cell	0.0309	dna	0.0257	time	0.0317
galaxy	0.0232	treatment	0.028	gene	0.025	sequencing	0.0172	age	0.0243
cluster	0.0214	trials	0.0277	tissue	0.0185	map	0.0123	year	0.024
cosmic	0.0137	therapy	0.0213	cloning	0.0169	genes	0.0122	record	0.0238
dark	0.0131	trial	0.0164	transfer	0.0155	chromosome	0.0119	early	0.0233
light	0.0109	disease	0.0157	blood	0.0113	regions	0.0119	billion	0.0177
density	0.01	medical	0.00997	embryos	0.0111	human	0.0111	history	0.0148

bacteria	0.0983	male	0.0558	theory	0.0811	immune	0.0909	stars	0.0524
bacterial	0.0561	females	0.0541	physics	0.0782	response	0.0375	star	0.0458
resistance	0.0431	female	0.0529	physicists	0.0146	system	0.0358	astrophys	0.0237
coli	0.0381	males	0.0477	einstein	0.0142	responses	0.0322	mass	0.021
strains	0.025	sex	0.0339	university	0.013	antigen	0.0263	disk	0.0173
microbiol	0.0214	reproductive	0.0172	gravity	0.013	antigens	0.0184	black	0.0161
microbial	0.0196	offspring	0.0168	black	0.0127	immunity	0.0176	gas	0.0149
strain	0.0165	sexual	0.0166	theories	0.01	immunology	0.0145	stellar	0.0127
salmonella	0.0163	reproduction	0.0143	aps	0.00987	antibody	0.014	astron	0.0125
resistant	0.0145	eggs	0.0138	matter	0.00954	autoimmune	0.0128	hole	0.00824

Comparison LSA and PLSA



From Th. Hofmann, 2000

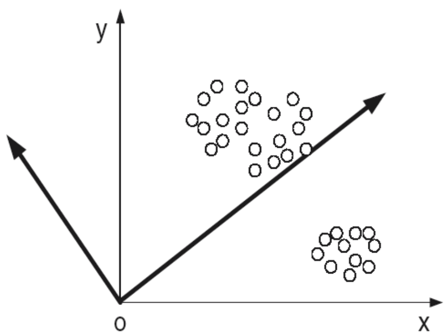
Non-negative Matrix Factorization

See: **Document Clustering Based On Non-negative Matrix Factorization**

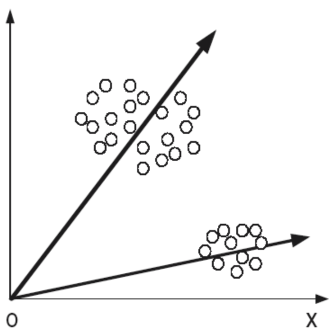
Wei Xu, Xin Liu, Yihong Gong
NEC Laboratories America, Inc.
10080 North Wolfe Road, SW3-350
Cupertino, CA 95014, U.S.A.
{xw,xliu,ygong}@ccrl.sj.nec.com

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 $D(A \mid WH)$
  - 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