Chapter 2
Non-Negative Matrix Factorization
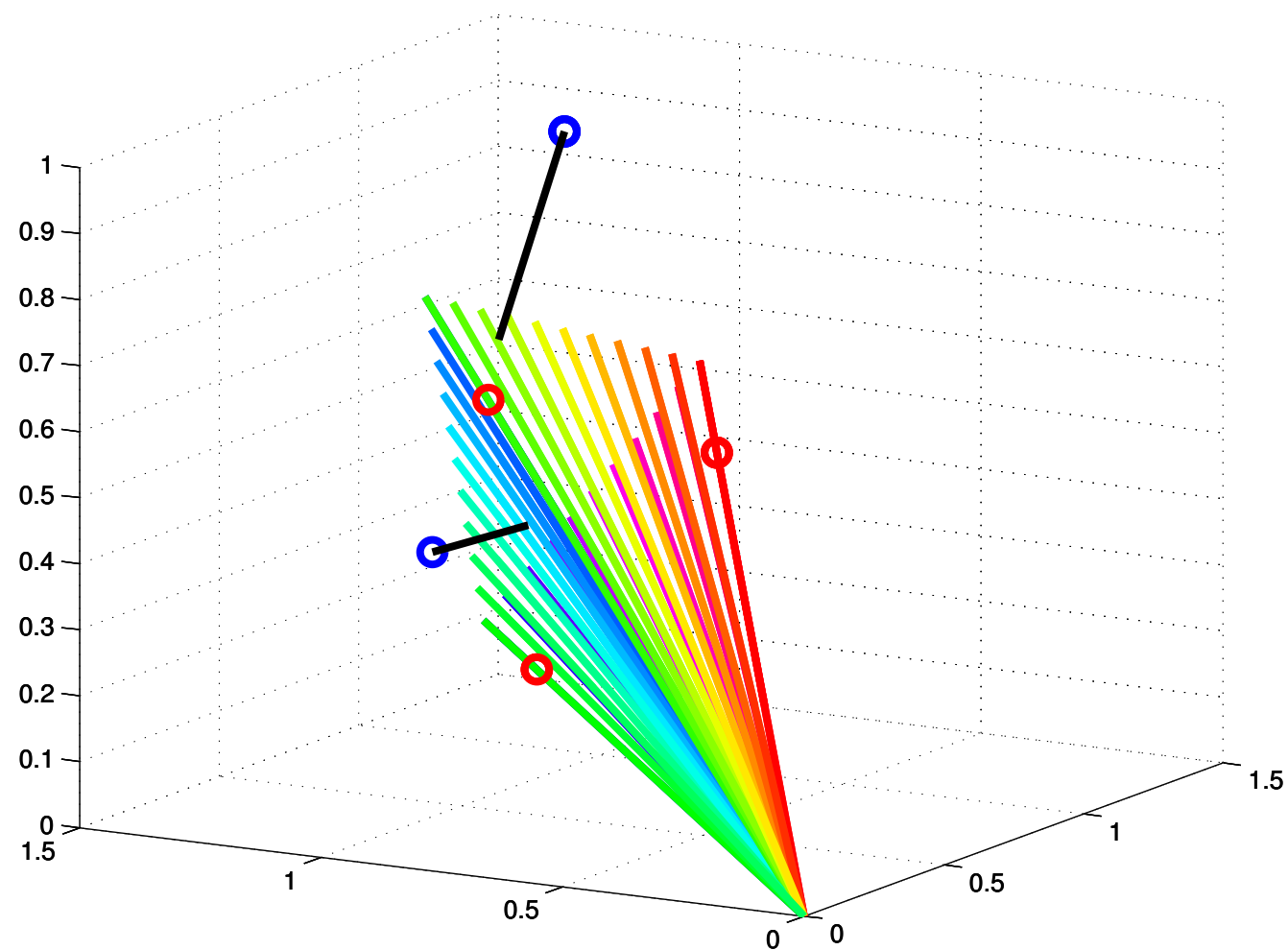
Part 2: Variations & applications

max planck institut
informatik

Geometry of NMF

Pauli Miettinen

Geometry of NMF

NMF factors
Data points
Convex cone
Projections

Sparsity in NMF

Sparsity: desiderata

- Sparse factor matrices are often preferred

  - Simpler to interpret (zeroes can be ignored)

  - Agrees with our intuition on parts of whole

  - Faster computations, less space

- NMF is sometimes claimed to automatically yield sparse factors

  - In practice, this is often not the case

Enforcing sparsity

- A common solution: change the target function to minimize

Parameters Regularizers

$$\frac{1}{2}\|\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}\|_F^2 + \alpha \cdot \text{sparsity}(\boldsymbol{W}) + \beta \cdot \text{sparsity}(\boldsymbol{H})$$

  - How to define sparsity?

  - Naïve: sparsity($\boldsymbol{W}$) = nnz($\boldsymbol{W}$)/size($\boldsymbol{W}$)

    - Non-convex, non-nice to optimize directly

Frobenius regularizer

- A.k.a. Tikhonov regularizer

- Minimize $\frac{1}{2}\left(\|\boldsymbol{A} - \boldsymbol{WH}\|_F^2 + \alpha\|\boldsymbol{W}\|_F^2 + \beta\|\boldsymbol{H}\|_F^2\right)$

  - Doesn't help much with sparsity

  - Used to impose smoothness

ALS-NMF with Frobenius regularizers

- The update rules for ALS with Frobenius regularizer are

$$W \leftarrow \left[(AH^T)(HH^T + \alpha I)^{-1}\right]_+$$

$$H \leftarrow \left[(W^T W + \beta I)^{-1}(W^T A)\right]_+$$

- Uses the fact that $X^+ = (X^T X)^{-1} X^T$ if X has full column rank (homework)

L_1 (Lasso) regularizer

- Using L_1 based instead of L_2 based regularizer helps obtaining sparse solutions

$$\frac{1}{2} \|\boldsymbol{A} - \boldsymbol{WH}\|_F^2 + \alpha \sum_{i,j} \boldsymbol{W}_{ij} + \beta \sum_{i,j} \boldsymbol{H}_{ij}$$

- Larger values of α and β yield sparse solutions (e.g. $\alpha, \beta \in [0.01, 0.5]$)

- Still no guarantees on sparsity

ALS-NMF with Lasso

- The update rules are

$$\mathbf{W} \leftarrow \left[(\mathbf{A}\mathbf{H}^T - \alpha\mathbf{1}_{n \times k})(\mathbf{H}\mathbf{H}^T)^{-1}\right]_+$$

$$\mathbf{H} \leftarrow \left[(\mathbf{W}^T\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{A} - \beta\mathbf{1}_{k \times m})\right]_+$$

- $\mathbf{1}_{n \times k}$ is n-by-k matrix of all 1s

- Requires columns of $\mathbf{W}$ to be normalized to unit L_1 after each update

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij}/\textstyle\sum_i \mathbf{W}_{ij}$$

Hoyer's sparse NMF

- Hoyer (2004) considers the following sparsity function for n-dimensional vector $\boldsymbol{x}$

$$\text{sparsity}(\boldsymbol{x}) = \frac{\sqrt{n} - \left(\sum_i |x_i|\right)/\sqrt{\sum_i x_i^2}}{\sqrt{n} - 1}$$

- sparsity$(\boldsymbol{x}) = 1$ iff nnz$(\boldsymbol{x}) = 1$

- sparsity$(\boldsymbol{x}) = 0$ iff $|\boldsymbol{x}_i| = |\boldsymbol{x}_j|$ for all i, j

Pauli Miettinen

Hoyer's sparse NMF

- Hoyer's algorithm obtains an NMF using factor matrices with user-defined level of sparsity

  - After every update, the columns of **W** and rows of **H** are updated s.t. their L_2 is constant and L_1 is set to desired level of sparsity

Getting the desired level of sparsity

- Set $s = x + (L_1 - \sum_i x_i)/\dim(x)$ and $Z = \{\}$ — Fix L_1

- **repeat**

  - Set $m_i = L_1/(\dim(x) - \text{size}(Z))$ for $i \notin Z$ and $m_i = 0$ o/w

    Fix L_2

  - Set $s = m + \alpha(s - m)$ where α is s.t. $||s||_2 = L_2$

  - If $s \geq 0$, **return** s — Are we done?

  - Set $Z = Z \cup \{i : s_i < 0\}$ and $s_i = 0$ for all $i \in Z$

  - Set $c = (\sum_i s_i - L_1)/(\dim(x) - \text{size}(Z))$

    Truncate negative values and fix L_1 again

  - Set $s_i = s_i - c$ for all $i \notin Z$

Other forms of NMF

Normalized NMF

- Columns of **W** (and/or rows of **H**) should be normalized to sum to unity

  - Stability of the solution and interpretability

- If only **W** (or **H**) is normalized, the weights can be pushed to the other matrix

- To normalize both, use k-by-k diagonal **Σ** s.t.
  $\sigma_{ii} = ||\boldsymbol{W}_i||_1 \times ||(\boldsymbol{H}^T)_i||_1$

  - Normalized NMF: **WΣH**

Semi-orthogonal NMF

- In **semi-orthogonal NMF** we restrict **H** to row-orthogonal:

  minimize $||\boldsymbol{A} - \boldsymbol{WH}||_F$ s.t. $\boldsymbol{HH}^T = \boldsymbol{I}$ and $\boldsymbol{W}$ and $\boldsymbol{H}$ are nonnegative

  - Solutions are unique (up to permutations)

  - The problem is "equivalent" to k-means

    - In the sense that the optimal solutions have the same value

Pauli Miettinen

NMF and clustering

- In k-means, we minimize

$$\sum_{j=1}^{k} \sum_{i \in C_j} \left\| \boldsymbol{a}_i - \boldsymbol{\mu}_j \right\|_2^2 = \sum_{j=1}^{k} \sum_{i=1}^{n} \boldsymbol{G}_{ij} \left\| \boldsymbol{a}_i - \boldsymbol{\mu}_j \right\|_2^2$$

  - $\boldsymbol{\mu}_j$ is the centroid of the jth cluster C_j

  - $\boldsymbol{G}$ is n-by-k **cluster assignment matrix**

    - $\boldsymbol{G}_{ij} = 1$ if $i \in C_j$ and 0 otherwise

  - Equivalently: $\left\| \boldsymbol{A} - \boldsymbol{GM} \right\|_F^2$

    Type of NMF if $\boldsymbol{A}$ is nonnegative!

    - $\boldsymbol{M}$ is k-by-m containing the centroids as its rows

Orthogonal tri-factor NMF

- We can find NMF where both W and H are (column/row) orthogonal

  - Often too restrictive; cannot handle different scales

- In **orthogonal nonnegative tri-factorization** we add third non-negative matrix S:

  minimize $||A - WSH||_F$ s.t. $W^T W = I$, $HH^T = I$, and all matrices are non-negative

More on tri-factorization

- **S** does not have to be square

  - **W** is n-by-k, **S** is k-by-l, **H** is l-by-m

    - Different number of row and column factors

- If orthogonal NMF "clusters" columns of **A**, this "bi-clusters" rows and columns simultaneously

Computing semi-orthogonal NMF

- If H has to be orthogonal, either

  - update as usual and set after every iteration
    $H \leftarrow [HH^T]^{-1/2} H$; or

  - update $H_{ij} \leftarrow H_{ij} \sqrt{\dfrac{(W^T A)_{ij}}{(W^T A H^T H)_{ij}}}$

- W is updated as usual (w/o constraints)

  - If W needs to be orthogonal, the update rules are changed accordingly

Computing the orthogonal tri-factorization

- The update rules for orthogonal tri-factorization are

$$H_{ij} \leftarrow H_{ij} \sqrt{\frac{\left((WS)^T A\right)_{ij}}{\left((WS)^T A H^T H\right)_{ij}}}$$

$$W_{ij} \leftarrow W_{ij} \sqrt{\frac{(A(SH)^T)_{ij}}{(WW^T A(SH)^T)_{ij}}}$$

$$S_{ij} \leftarrow S_{ij} \sqrt{\frac{(W^T A H^T)_{ij}}{(W^T W S H H^T)_{ij}}}$$

Other optimization functions

Kullback–Leibler divergence

- The **Kullback–Leibler divergence** of Q from P, $D_{KL}(P\|Q)$, measures the expected number of **extra** bits required to code samples from P when using a code optimized for Q

$$D_{KL}(P\|Q) = \sum_i P(i) \ln \frac{P(i)}{Q(i)}$$

- P and Q are probability distributions

- Non-negative and **non-symmetric**

Generalized KL-divergence and matrix factorizations

- The standard KL-divergence requires P and Q be probability distributions (e.g. $\sum_i P(i) = 1$)

  - The **generalized KL-divergence** (or **I-divergence**) removes this requirement:

  $$D_{\mathsf{GKL}}(P \| Q) = \sum_i \left(P(i) \ln \frac{P(i)}{Q(i)} - P(i) + Q(i) \right)$$

- In NMF, $P = \boldsymbol{A}$ and $Q = \boldsymbol{WH}$:

$$D_{\mathsf{GKL}}(\boldsymbol{A} \| \boldsymbol{WH}) = \sum_{i,j} \left(\boldsymbol{A}_{ij} \ln \frac{\boldsymbol{A}_{ij}}{(\boldsymbol{WH})_{ij}} - \boldsymbol{A}_{ij} + (\boldsymbol{WH})_{ij} \right)$$

KL v.s. GKL in NMF

- KL requires A to be considered as a probability distribution

  - $\sum_{i,j} A_{i,j} = 1$ (or row/column normalization)

  - WH should be normalized the same way

- GKL only requires non-negativity

  - But inherently assumes integer data

  - Looses a bit of the probability interpretation

NMF for GKL

- The update rules for multiplicative GKL NMF algorithm are

$$H_{kj} \leftarrow H_{kj} \frac{\sum_{i=1}^{n} W_{ik}(A_{ij}/(WH)_{ij})}{\sum_{i=1}^{n} W_{ik}}$$

$$W_{ik} \leftarrow W_{ik} \frac{\sum_{j=1}^{m} (A_{ij}/(WH)_{ij})H_{kj}}{\sum_{j=1}^{m} H_{kj}}$$
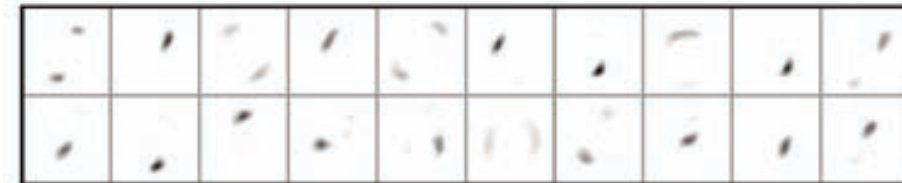
- The columns of W are normalized to sum to unity after every iteration

Applications of NMF

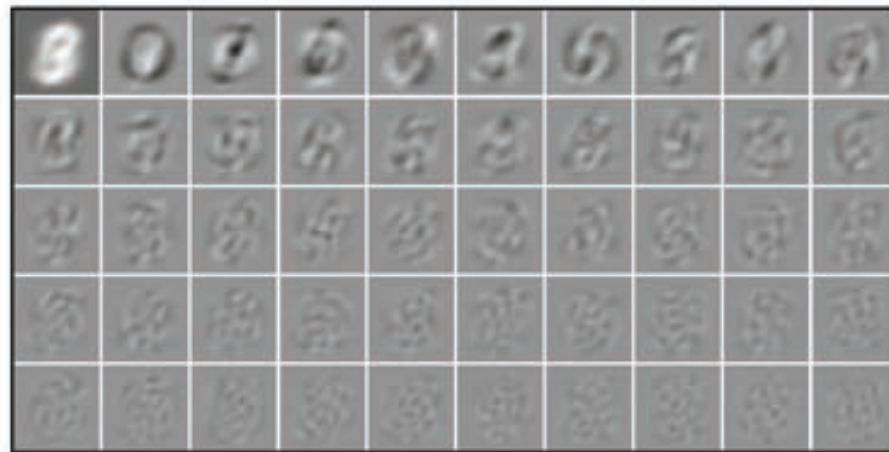# Hand-written digits

(a) PCA - $J = 20$

(b) NMF - $J = 20$

(c) PCA - $J = 50$

(d) NMF - $J = 50$

Cichocki et al. 2009
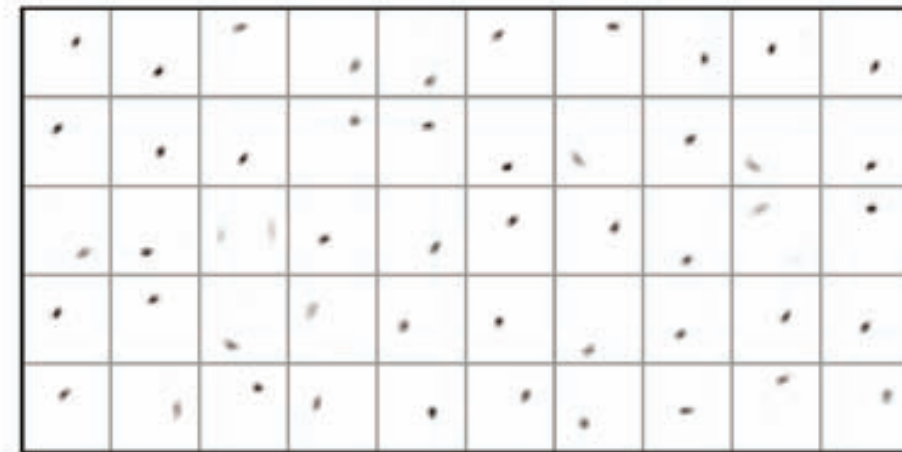
Separation of various spectra

- In spectroscopy imaging we often have multiple observations of signals over some spectrum

  - Observations-by-spectrum non-negative matrix

- The signals constitute an additive mixture of "pure" signals

Raman spectroscopy

Ground truth

Observation

Estimated
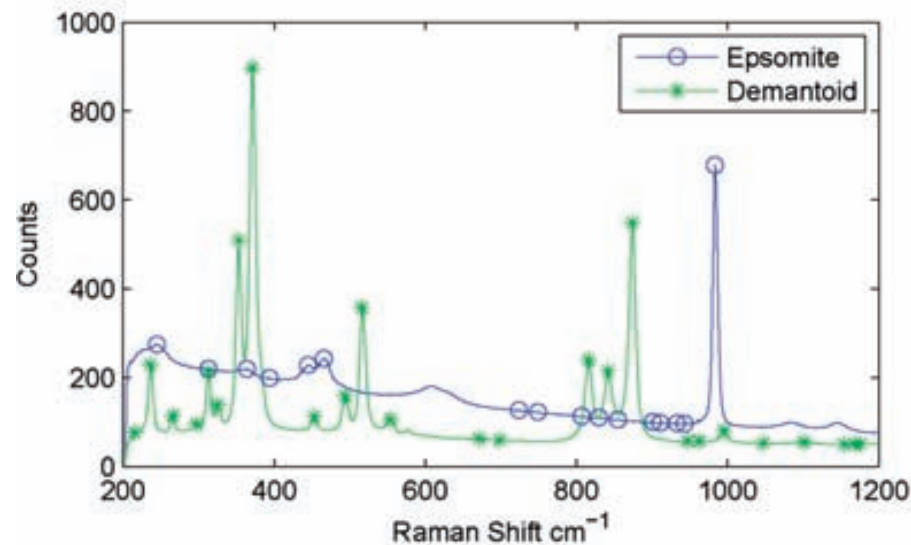
Estimated

(a) Epsomite and Demantoid spectra

(b) Ten components of 256 noisy mixtures
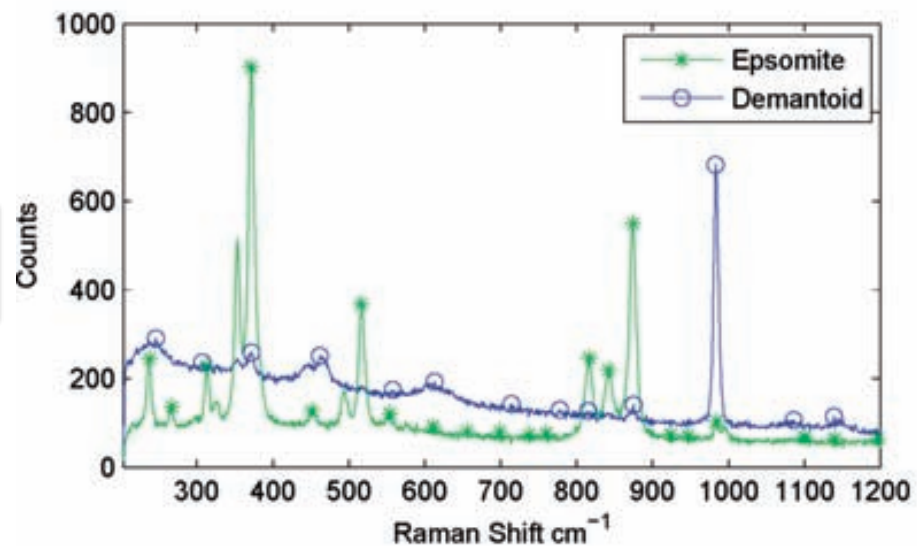
(c) Estimated Raman spectra
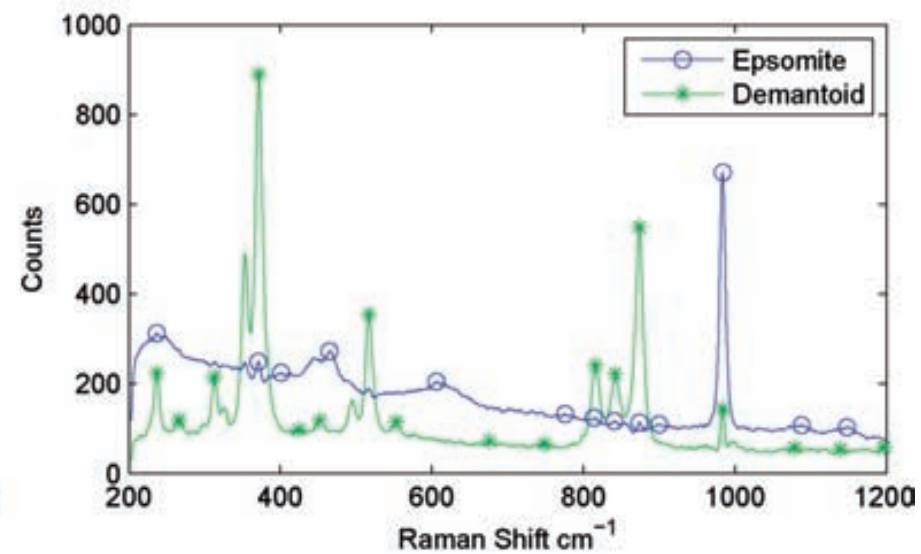
(d) Smoothed spectra

Cichocki et al. 2009

Text mining and pLSA

- Consider a document–term matrix **A**

- a_{ij} is the number of times term j appears in document i

Can we find these topics automatically?

Environmet

Politics

$$
\mathbf{A} = \begin{array}{c} \text{doc 1} \\ \text{doc 2} \\ \text{doc 3} \\ \text{doc 4} \\ \text{doc 5} \end{array}
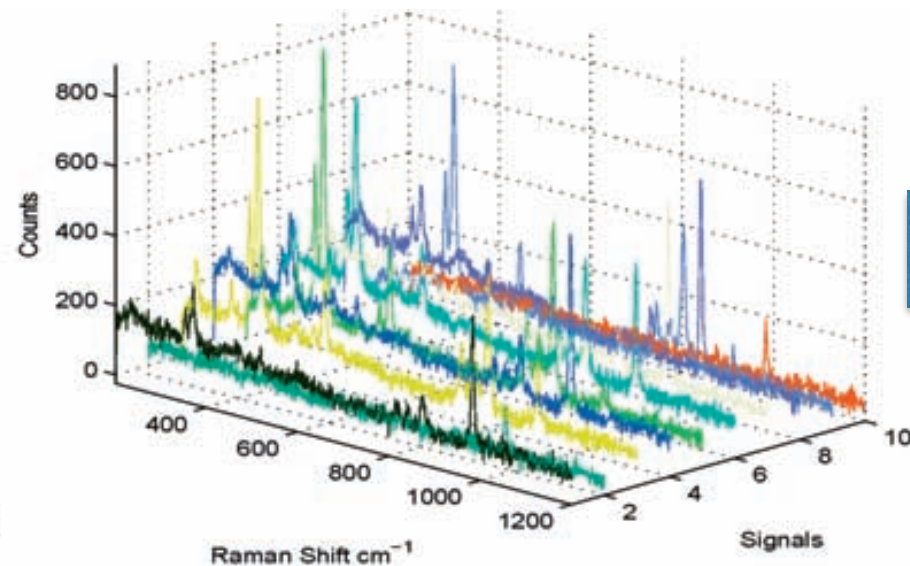\begin{array}{ccccc}
\text{air} & \text{water} & \text{pollution} & \text{democrat} & \text{republican} \\
3 & 2 & 8 & 0 & 0 \\
1 & 4 & 12 & 0 & 0 \\
0 & 0 & 0 & 10 & 11 \\
0 & 0 & 0 & 8 & 5 \\
1 & 1 & 1 & 1 & 1
\end{array}
$$

The idea

- Normalized **A**' that sums to 1 can be considered as a probability distribution

  $P(d, w) = \mathbf{A}'_{d,w}$

- Model with topics:

  $$P(d, w) = \sum_k P(k)P(d \mid k)P(w \mid k)$$

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

Generative process

- Pick a document according to $P(d)$

- Select a topic according to $P(z \mid d)$

- Select a word according to $P(w \mid z)$

$$P[w \mid d] = \sum_z P[z \mid d] \cdot P[w \mid z]$$

economic

imports

embargo

TRADE

documents d   latent concepts z   terms w
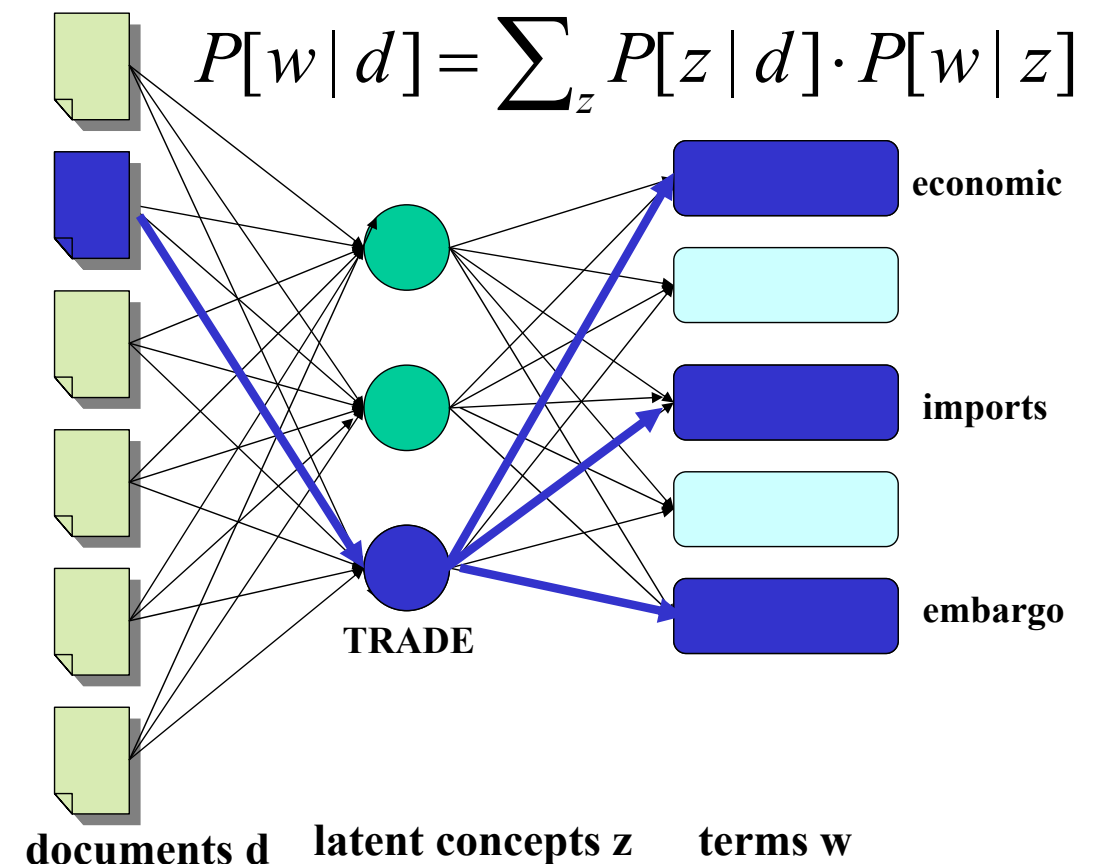
pLSA as NMF

- In the NMF version of the **probabilistic latent semantic analysis** (pLSA) we are given

  - documents-by-terms matrix $\boldsymbol{A}$ and rank r

- We have to find

  - n-by-r non-negative $\boldsymbol{W}$ (columns sum to unity)

  - r-by-r diagonal non-negative $\boldsymbol{\Sigma}$

  - r-by-m non-negative $\boldsymbol{H}$ (rows sum to unity)

- Minimizing $D_{\text{GKL}}(\boldsymbol{A} \mathbin{\|} \boldsymbol{W\Sigma H})$

Geometry of pLSA

T. Hofmann *Unsupervised learning by probabilistic latent semantic analysis*. 2001
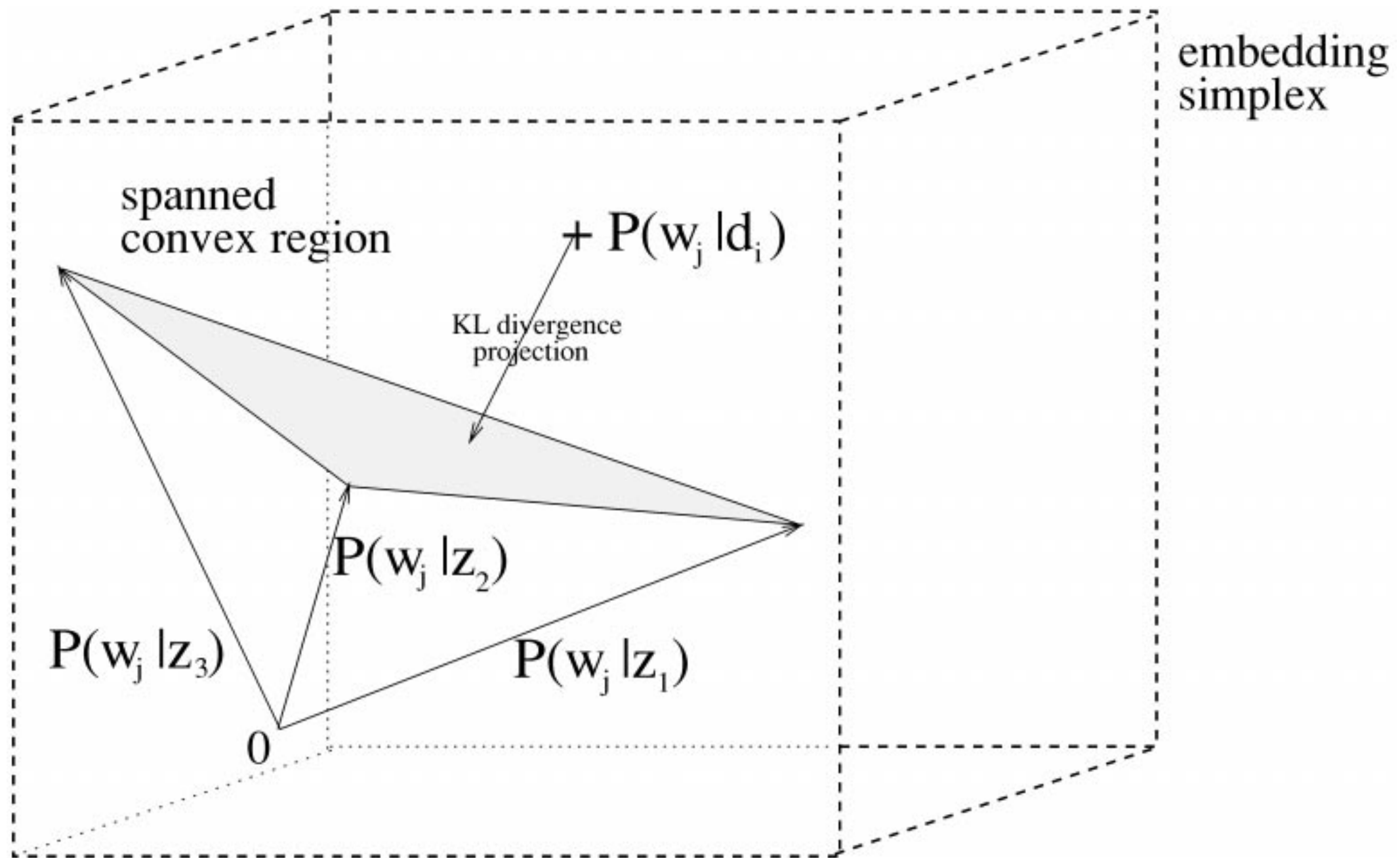
pLSA example

air wat pol dem rep

air	wat	pol	dem	rep
0.04	0.03	0.12	0	0
0.01	0.06	0.17	0	0
0	0	0	0.14	0.16
0	0	0	0.12	0.07
0.01	0.01	0.01	0.01	0.01

A

Here, A is normalized

0.39	0
0.52	0
0	0.58
0	0.36
0.09	0.06

W

How strong the topic is in the document?

0.48	0
0	0.52

Σ

Overall frequency

air wat pol dem rep

air	wat	pol	dem	rep
0.15	0.21	0.64	0	0
0	0	0	0.53	0.47

H

How strong the word is in the topic?

NMF algorithm for pLSA

- Compute **W** and **H** using GKL NMF algorithm

- Normalize columns (rows) of **W** (**H**) and put the multipliers to **Σ**

  - Normalize **Σ** to sum to unity

- Real implementations would require tempering to avoid over-fitting

pLSA example

- Concepts (10 of 128) extracted from Science Magazine articles (12K)

P(w|z)

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

P(w|z)

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

Source: Thomas Hofmann, Tutorial at ADFOCS 2004

pLSA applications

- Topic modeling

- Clustering documents and terms

- Information retrieval

  - Similar to LSA/LSI

- Generalizes better than LSA

  - But outperformed by **Latent Dirichlet Allocation** (LDA)

NMF summary

- Parts-of-whole interpretation

  - Often easier/more appropriate than SVD

- Hard to compute and non-unique

  - Local updates (multiplicative, gradient, ALS)

- Many applications and specific variations

- Still under very active research

Literature

- Berry, Browne, Langville, Pauca & Plemmons (2007): *Algorithms and applications for approximate nonnegative matrix factorization.* Comput. Stat. Data Anal. 52, pp. 155–173

- Hoyer (2004): *Non-negative matrix factorizations with sparseness constraints*. J. Mach. Learn. Res. 5, pp. 1457–1469

- Ding, Li, Peng & Park (2006): *Orthogonal nonnegative matrix tri-factorizations for clustering.* In KDD '06, pp. 126–135

- Cichocki, Zdunek, Phan & Amari (2009): *Nonnegative matrix and tensor factorizations.* John Wiley & Sons Chichester, UK