Chapter 3

Non-Negative Matrix Factorization

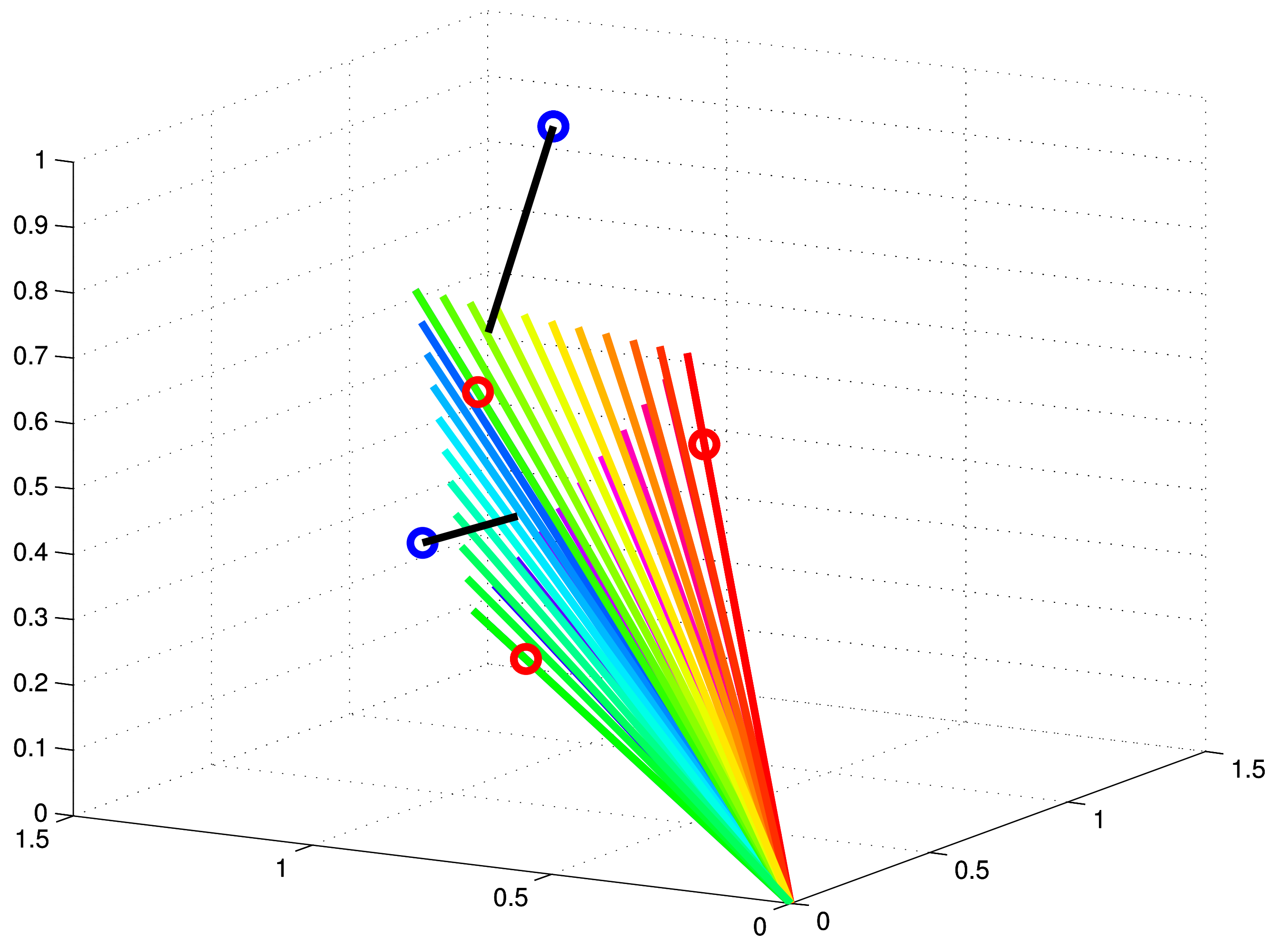
Part 2: Variations & applications

max planck institut
informatik

Geometry of NMF

Geometry of NMF

NMF factors
Data points
Convex cone
Projections



Sparsity in NMF

Skillicorn chapter 8; Berry et al. (2007)

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

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

- How to define sparsity?
- Naïve: density(***W***) = 1 – nnz(***W***)/size(***W***)
  - Non-convex, non-nice to optimize directly

Frobenius regularizer

- A.k.a. Tikhonov regularizer
- Minimize $\frac{1}{2}\left(\|\boldsymbol{A}-\boldsymbol{W}\boldsymbol{H}\|_F^2+\alpha\|\boldsymbol{W}\|_F^2+\beta\|\boldsymbol{H}\|_F^2\right)$
  - Doesn’t help much with sparsity
  - Used to impose smoothness

ALS-NMF with Frobenius regularizers

- The update rules for ALS with Frobenius regularizer are

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

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

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

L_1 (Lasso) regularizer

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

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

  - 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

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

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

  - $\mathbf{1}_{n\times k}$ is n-by-k matrix of all 1s
  - Requires columns of $\boldsymbol{W}$ to be normalized to unit L_1 after each update

$$\boldsymbol{W}_{ij} \leftarrow \boldsymbol{W}_{ij} / \sum_i \boldsymbol{W}_{ij}$$

Hoyer's sparse NMF

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

$$\text{sparsity}(\boldsymbol{x}) = \frac{\sqrt{n} - \|\boldsymbol{x}\|_1 / \|\boldsymbol{x}\|_2}{\sqrt{n} - 1}$$

  - sparsity($\boldsymbol{x}$) = 1 iff nnz($\boldsymbol{x}$) = 1
  - sparsity($\boldsymbol{x}$) = 0 iff $|\boldsymbol{x}_i| = |\boldsymbol{x}_j|$ for all i, j

Hoyer 2004

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 $\boldsymbol{W}$ and rows of $\boldsymbol{H}$ 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 $\boldsymbol{s} = \boldsymbol{x} + (L_1 - \sum_i \boldsymbol{x}_i)/\mathrm{dim}(\boldsymbol{x})$ and Z = {}

  Fix L_1

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

    Fix L_2

  - Set $\boldsymbol{s} = \boldsymbol{m} + \alpha(\boldsymbol{s} - \boldsymbol{m})$ where α is s.t. $||\boldsymbol{s}||_2 = L_2$
  - If $\boldsymbol{s} \geq 0$, **return** $\boldsymbol{s}$

    Are we done?

  - Set $Z = Z \cup \{i : \boldsymbol{s}_i < 0\}$ and $\boldsymbol{s}_i = 0$ for all $i \in Z$
  - Set $c = (\sum_i s_i - L_1)/(\mathrm{dim}(\boldsymbol{x}) - \mathrm{size}(Z))$
  - Set $s_i = s_i - c$ for all $i \notin Z$

    Truncate negative values and fix L_1 again

Other forms of NMF

Normalized NMF

- Columns of $\boldsymbol{W}$ (and/or rows of $\boldsymbol{H}$) should be normalized to sum to unity
  - Stability of the solution and interpretability
- If only $\boldsymbol{W}$ (or $\boldsymbol{H}$) is normalized, the weights can be pushed to the other matrix
- To normalize both, use k-by-k diagonal $\boldsymbol{\Sigma}$ s.t. $\sigma_{ii} = ||\boldsymbol{W}_i||_1 \times ||(\boldsymbol{H}^T)_i||_1$
  - Normalized NMF: $\boldsymbol{W\Sigma H}$

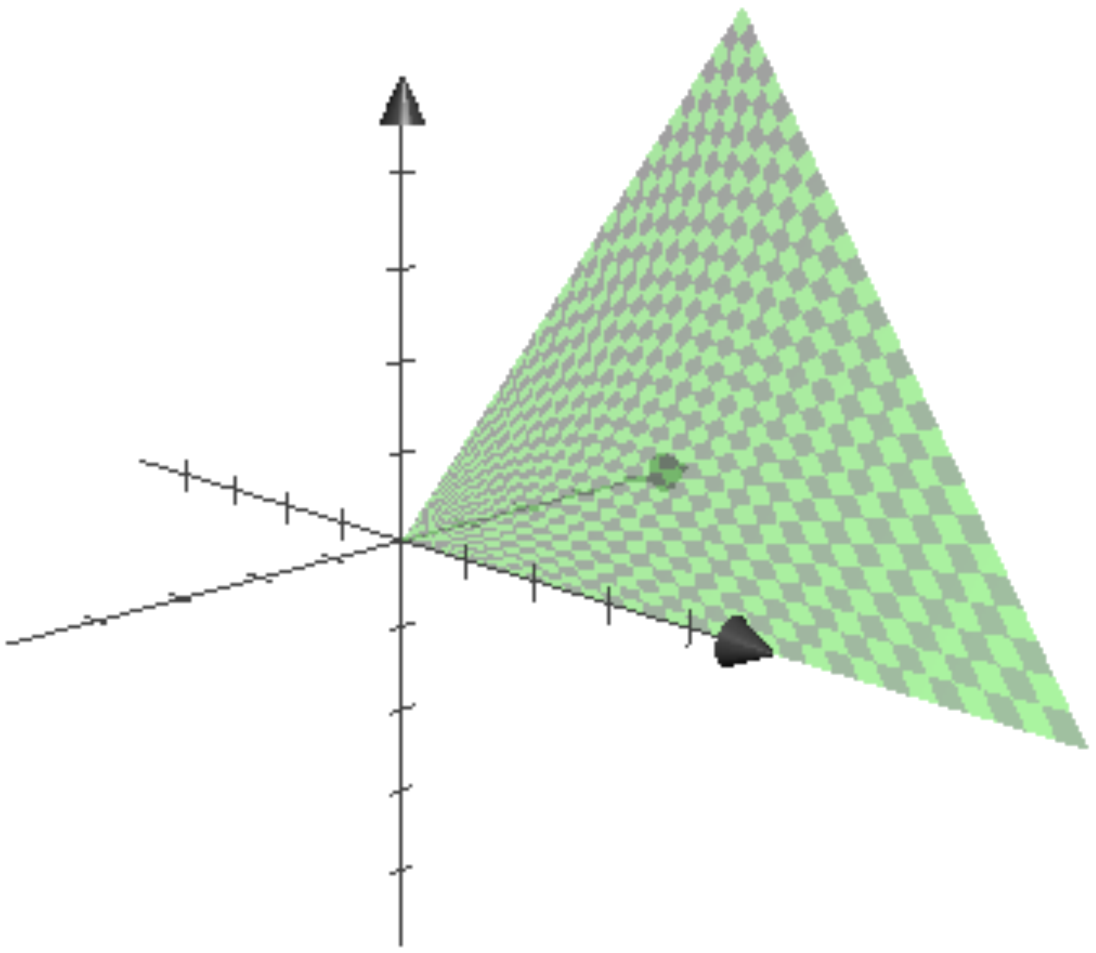
Semi-orthogonal NMF

- In **semi-orthogonal NMF** we restrict $\boldsymbol{H}$ 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

Ding et al. 2006

Geometry of semi-orthogonal NMF

The orthogonal factors span a cone



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: $\|\boldsymbol{A}-\boldsymbol{G}\boldsymbol{M}\|_F^2$
    - $\boldsymbol{M}$ is k-by-m containing the centroids as its rows

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

Orthogonal tri-factor NMF

- We can find NMF where both $\boldsymbol{W}$ and $\boldsymbol{H}$ are (column/row) orthogonal
  - Often too restrictive; cannot handle different scales
- In **orthogonal nonnegative tri-factorization** we add third non-negative matrix $\boldsymbol{S}$: minimize $\|\boldsymbol{A} - \boldsymbol{WSH}\|_F$ s.t. $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$, $\boldsymbol{HH}^T = \boldsymbol{I}$, 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 $\boldsymbol{H}$ has to be orthogonal, either
  - update as usual and set after every iteration $\boldsymbol{H} \leftarrow [\boldsymbol{H}\boldsymbol{H}^T]^{-1/2}\boldsymbol{H}$; or
  - update $\boldsymbol{H}_{ij} \leftarrow \boldsymbol{H}_{ij}\sqrt{\frac{(\boldsymbol{W}^T\boldsymbol{A})_{ij}}{(\boldsymbol{W}^T\boldsymbol{A}\boldsymbol{H}^T\boldsymbol{H})_{ij}}}$
- $\boldsymbol{W}$ is updated as usual (w/o constraints)
  - If $\boldsymbol{W}$ needs to be orthogonal, the update rules are changed accordingly

Computing the orthogonal tri-factorization

- The update rules for orthogonal tri-factorization are

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

$$\boldsymbol{W}_{ij} \leftarrow \boldsymbol{W}_{ij} \sqrt{\frac{(\boldsymbol{A}(\boldsymbol{SH})^T)_{ij}}{(\boldsymbol{W}\boldsymbol{W}^T\boldsymbol{A}(\boldsymbol{SH})^T)_{ij}}}$$

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

Other optimization functions

Kullback–Leibler divergence

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

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

  - 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:

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

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

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

KL v.s. GKL in NMF

- KL requires $\boldsymbol{A}$ to be considered as a probability distribution
  - $\sum_{i,j} \boldsymbol{A}_{i,j} = 1$ (or row/column normalization)
  - $\boldsymbol{WH}$ 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

$$\boldsymbol{H}_{kj} \leftarrow \boldsymbol{H}_{kj} \frac{\sum_{i=1}^{n} \boldsymbol{W}_{ik}(\boldsymbol{A}_{ij}/(\boldsymbol{W}\boldsymbol{H})_{ij})}{\sum_{i=1}^{n} \boldsymbol{W}_{ik}}$$

$$\boldsymbol{W}_{ik} \leftarrow \boldsymbol{W}_{ik} \frac{\sum_{j=1}^{m} (\boldsymbol{A}_{ij}/(\boldsymbol{W}\boldsymbol{H})_{ij})\boldsymbol{H}_{kj}}{\sum_{j=1}^{m} \boldsymbol{H}_{kj}}$$

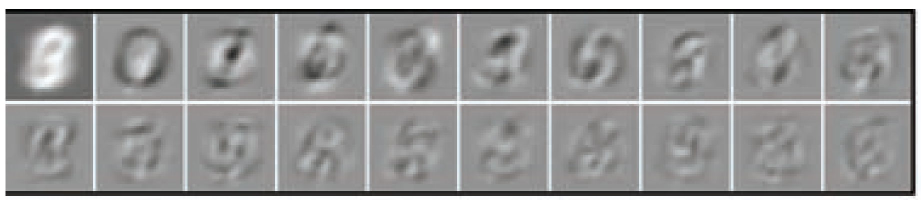
- The columns of $\boldsymbol{W}$ are normalized to sum to unity after every iteration

Applications of NMF

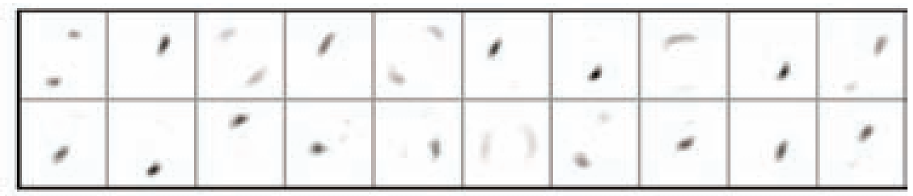
Component interpretation

- NMF's main "sales argument" is the component interpretation
  - $\boldsymbol{A} \approx \boldsymbol{w}_1\boldsymbol{h}^1 + \boldsymbol{w}_2\boldsymbol{h}^2 + \ldots + \boldsymbol{w}_k\boldsymbol{h}^k$
  - Each rank-1 component has "parts-of-whole" interpretation
    - Nothing is ever removed

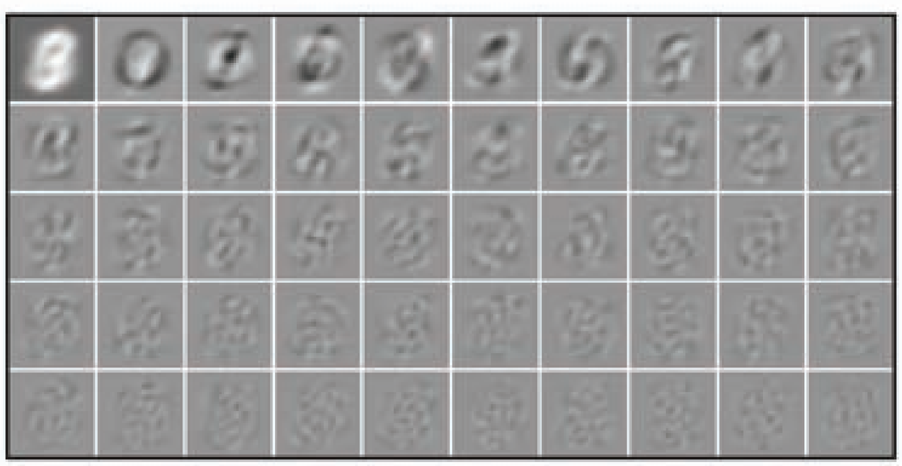
Hand-written digits



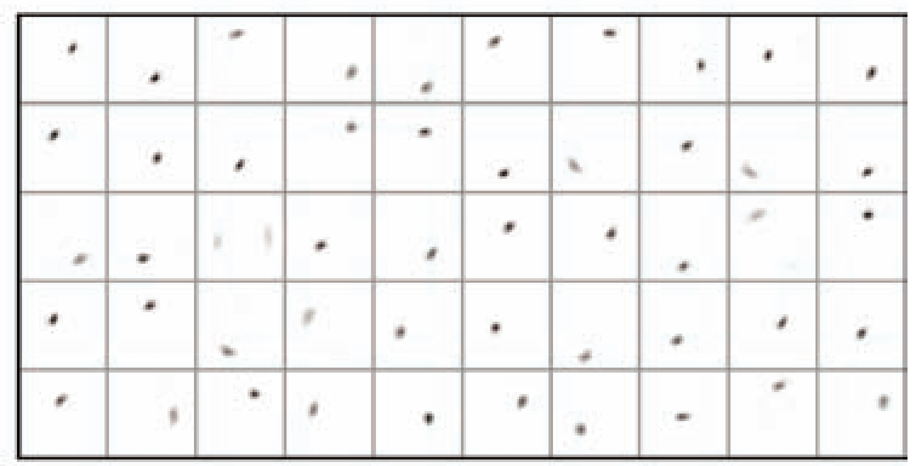
(a) PCA - $J = 20$



(b) NMF - $J = 20$



(c) PCA - $J = 50$



(d) NMF - $J = 50$

Cichocki et al. 2009

Factor interpretation

- NMF can be seen as a nonnegative mixture of nonnegative factors
    - The factors can capture underlying nonnegative phenomena
    - The nonnegative coefficients potentially help with the interpretation

Geometric interpretation

- NMF factors are not (generally) orthogonal
  - They do not create a coordinate system
  - Span a convex cone
- Projection to the space spanned by the factors can yield odd results
  - Points that are far away in the original space get close in the cone and vice versa

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

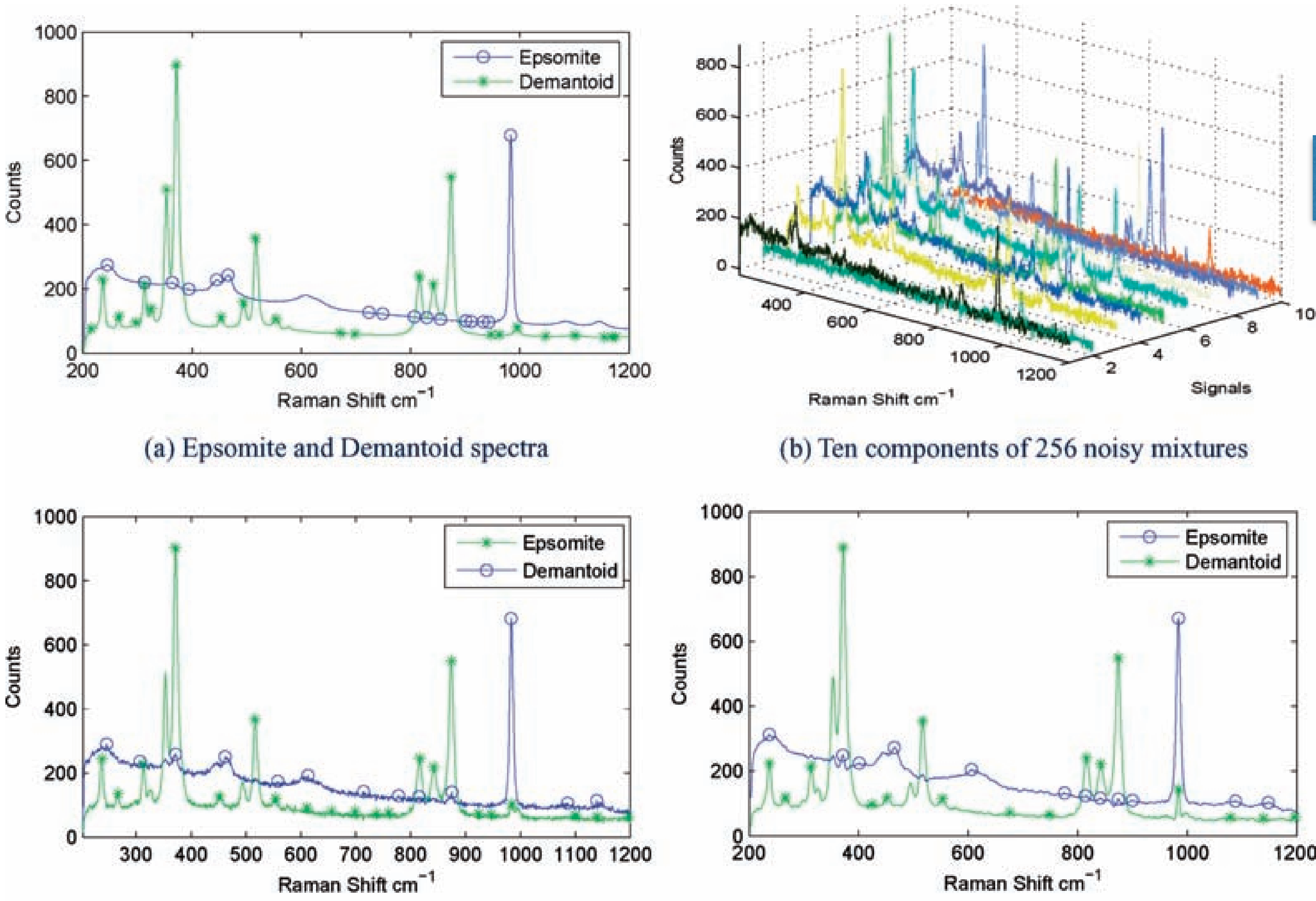
Raman spectroscopy

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

Can we find these topics automatically?

Environmet

$\boldsymbol{A}$ =

	air	water	pollution	democrat	republican
doc 1	3	2	8	0	0
doc 2	1	4	12	0	0
doc 3	0	0	0	10	11
doc 4	0	0	0	8	5
doc 5	1	1	1	1	1

Politics

The idea

- Normalized $\boldsymbol{A}'$ that sums to 1 can be considered as a probability distribution

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

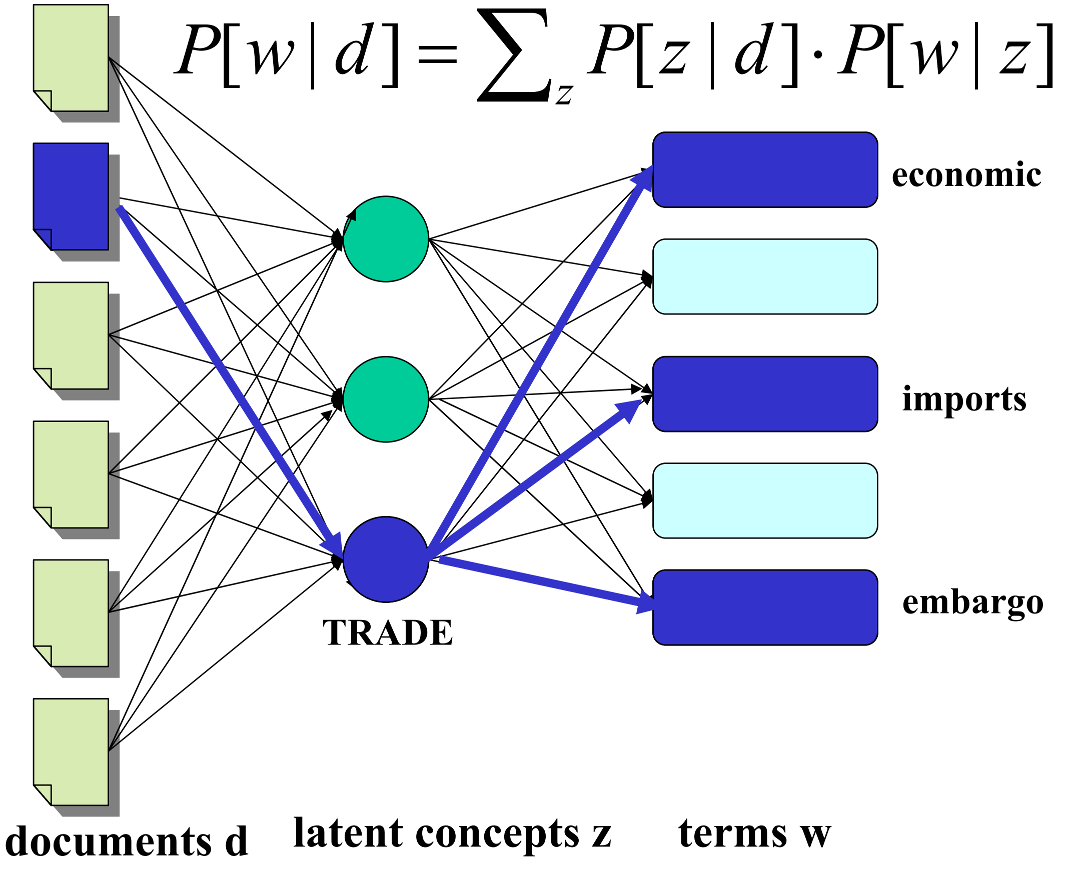
$$P(d, w) = \sum_k P(k)P(d \mid k)P(w \mid k)$$

- Model with topics:

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

Generative process

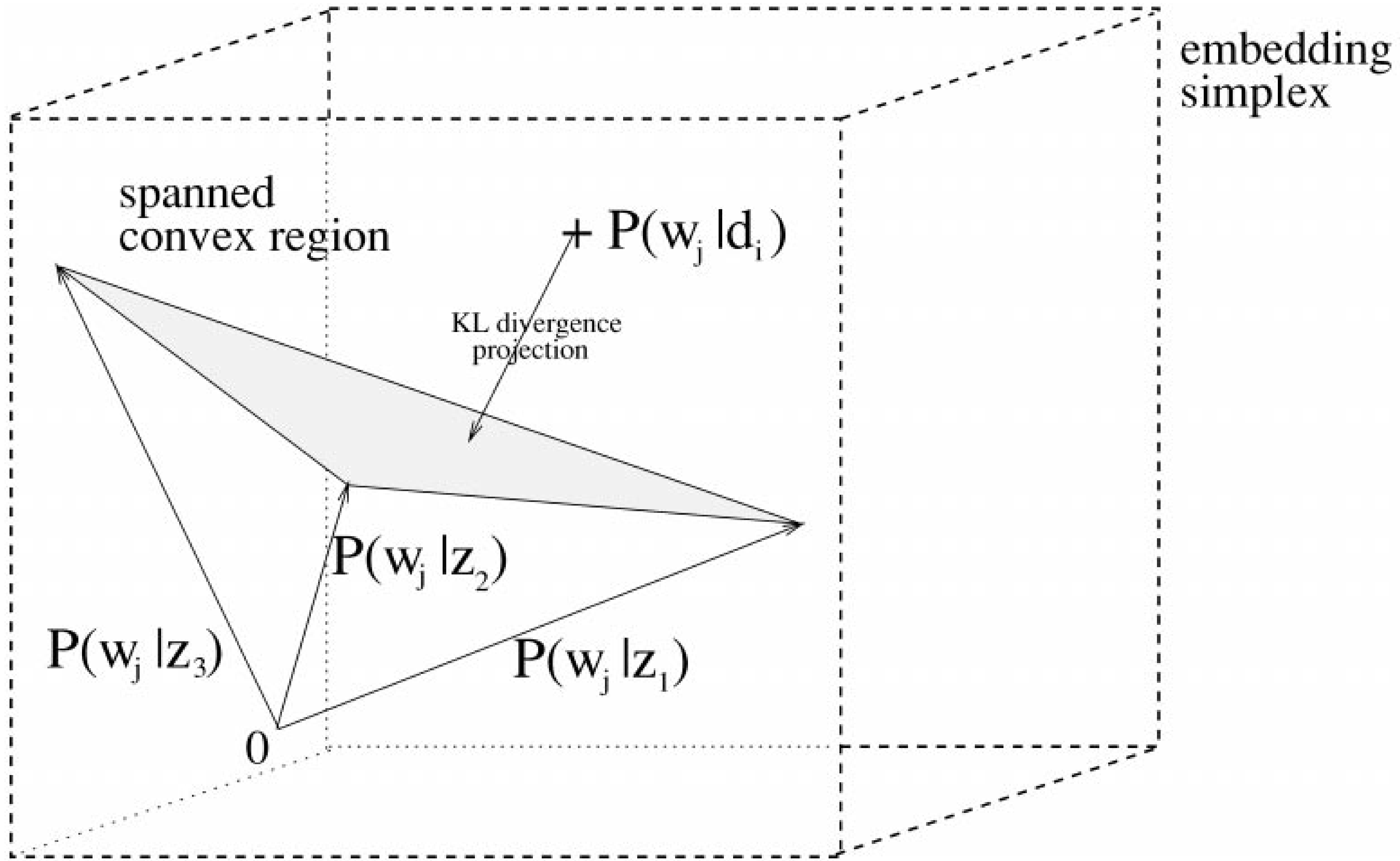
- Pick a document according to $P(d)$
- Select a topic according to $P(z \mid d)$
- Select a word according to $P(w \mid z)$



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 $D_{\text{GKL}}(\boldsymbol{A} \,||\, \boldsymbol{W\Sigma H})$

Geometry of pLSA



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

pLSA example

A

	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

Here, A is normalized

W

0.39	0
0.52	0
0	0.58
0	0.36
0.09	0.06

How strong the topic is in the document?

Σ

0.48	0
0	0.52

Overall frequency

H

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

How strong the word is in the topic?

NMF algorithm for pLSA

- Compute $\boldsymbol{W}$ and $\boldsymbol{H}$ using GKL NMF algorithm
- Normalize columns (rows) of $\boldsymbol{W}$ ($\boldsymbol{H}$) and put the multipliers to $\boldsymbol{\Sigma}$
  - Normalize $\boldsymbol{\Sigma}$ 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	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

P(w|z)

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

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