Data Mining and Matrices

06 – Non-Negative Matrix Factorization

Rainer Gemulla, Pauli Miettinen

May 23, 2013

Non-Negative Datasets

Some datasets are intrinsically non-negative:

- Counters (e.g., no. occurrences of each word in a text document)
- Quantities (e.g., amount of each ingredient in a chemical experiment)
- Intensities (e.g., intensity of each color in an image)

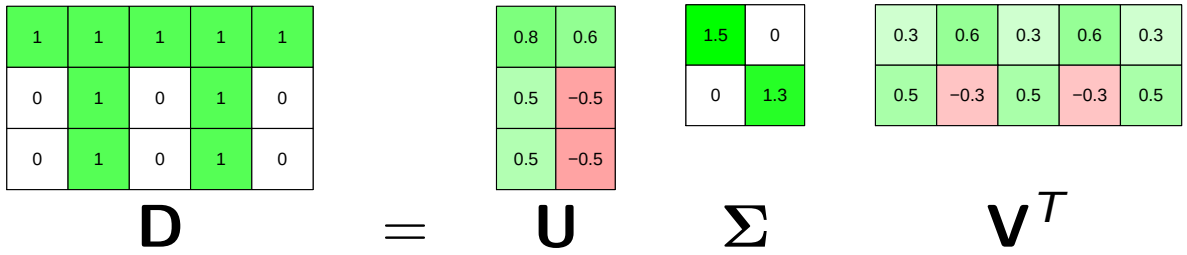
The corresponding data matrix $\mathbf{D}$ has only non-negative values.

- Decompositions such as SVD and SDD may involve negative values in factors and components
- Negative values describe the absence of something
- Often no natural interpretation

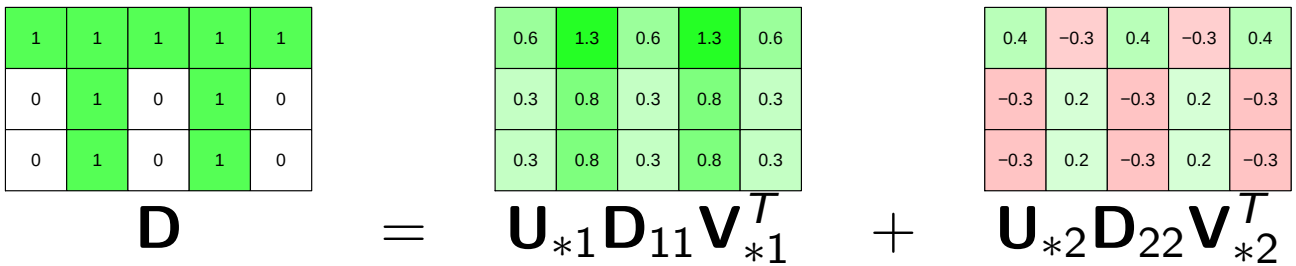
Can we find a decomposition that is more natural to non-negative data?

Example (SVD)

Consider the following "bridge" matrix and its truncated SVD:



Here are the corresponding components:



Negative values make interpretation unnatural or difficult.

Outline

Non-Negative Matrix Factorization (NMF)

Definition (Non-negative matrix factorization, basic form)

Given a non-negative matrix $\mathbf{D} \in \mathbb{R}_+^{m \times n}$, a *non-negative matrix factorization* of rank k is

$$\mathbf{D} \approx \mathbf{LR},$$

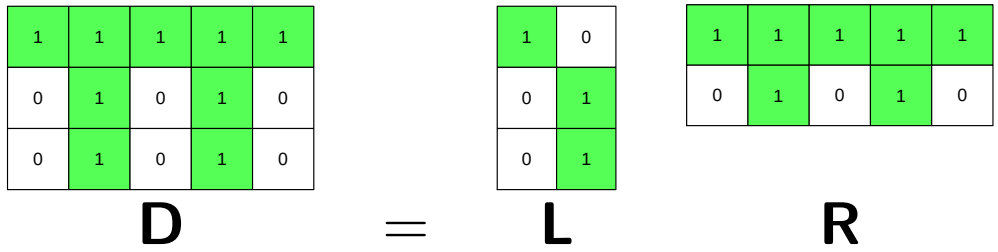
where $\mathbf{L} \in \mathbb{R}_+^{m \times r}$ and $\mathbf{R} \in \mathbb{R}_+^{r \times n}$ are both non-negative.

- Additive decomposition: factors and components non-negative
  $\rightarrow$ No cancellation effects
- Rows of $\mathbf{R}$ can be thought as "parts"
- Row of $\mathbf{D}$ obtained by mixing (or "assembling") parts in $\mathbf{L}$
- Smallest r such that $\mathbf{D} = \mathbf{LR}$ exists is called *non-negative rank* of $\mathbf{D}$

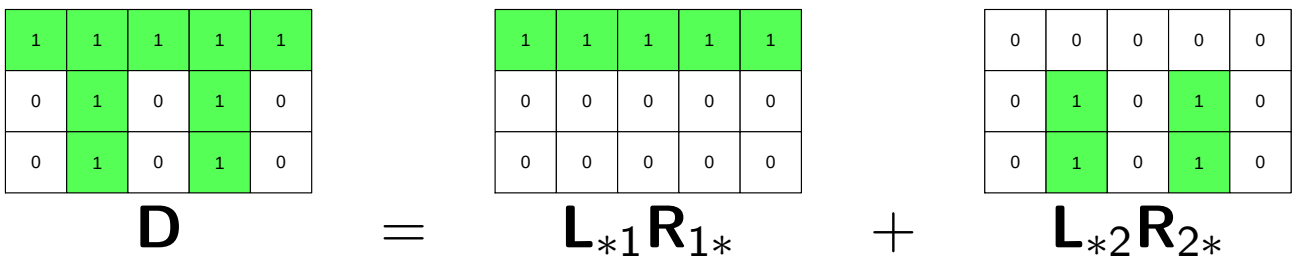
$$\operatorname{rank}(\mathbf{D}) \leq \operatorname{rank}_+(\mathbf{D}) \leq \min\{m, n\}$$

Example (NMF)

Consider the following "bridge" matrix and its rank-2 NMF:



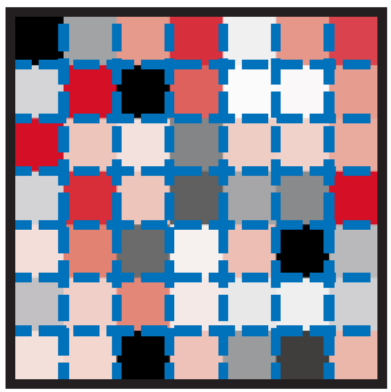
Here are the corresponding components:

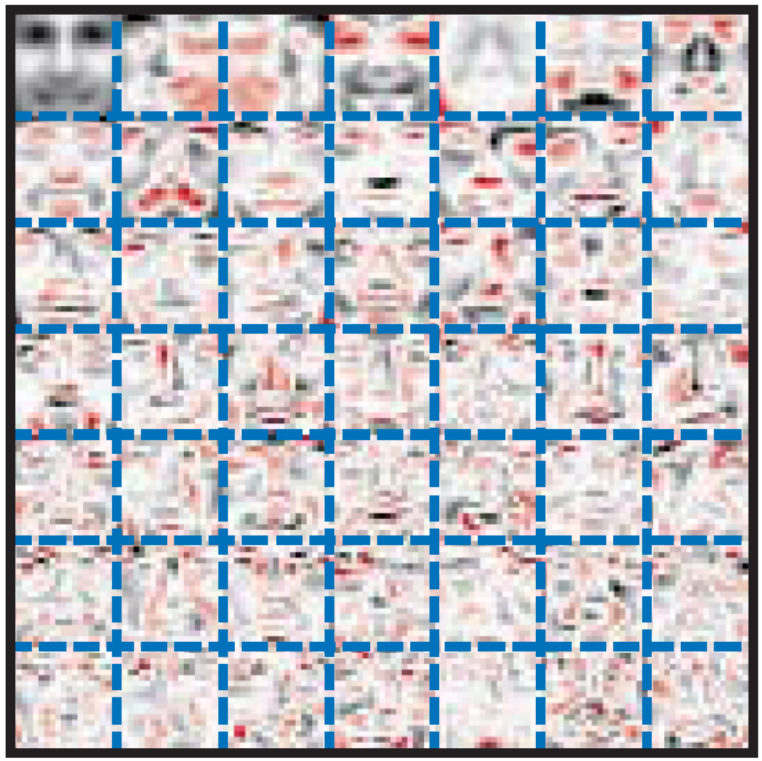


Non-negative matrix decomposition encourage a more natural, part-based representation and (sometimes) sparsity.

Decomposing faces (PCA)

$\mathbf{D}_{i*}$ (original)





$$[\mathbf{LR}]_{i*} = \mathbf{L}_{i*} \quad \mathbf{R}$$

$$[\mathbf{U\Sigma V}^T]_{i*} = \mathbf{U}_{i*}\mathbf{\Sigma} \quad \mathbf{V}^T$$

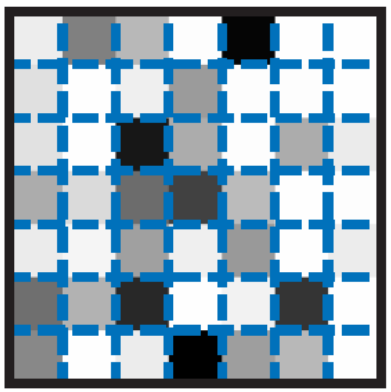
PCA factors are hard to interpret.

Lee and Seung, 1999.

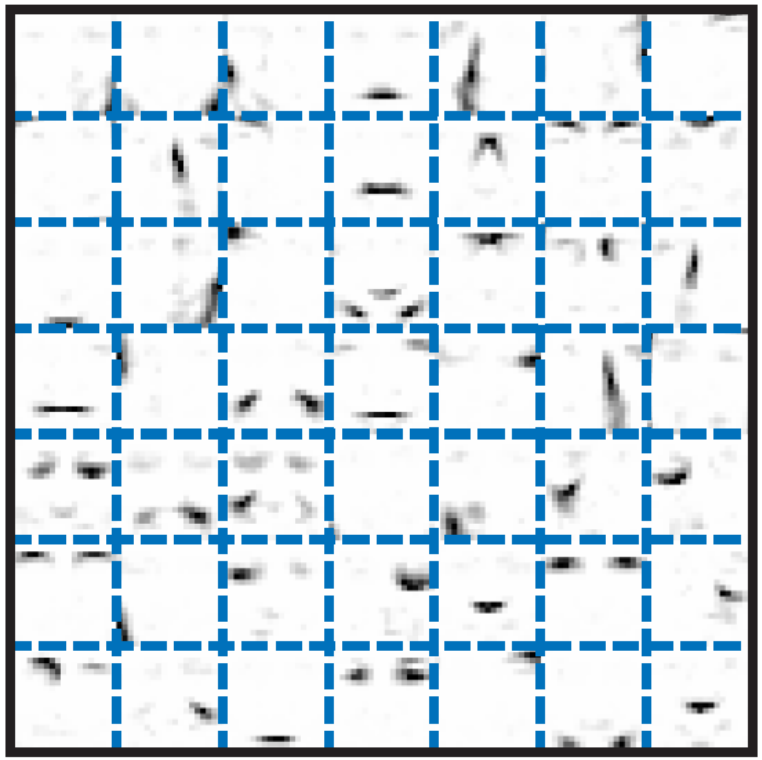
Decomposing faces (NMF)

$\mathbf{D}_{i*}$ (original)

$[\mathbf{LR}]_{i*}$ $=$



$\mathbf{L}_{i*}$



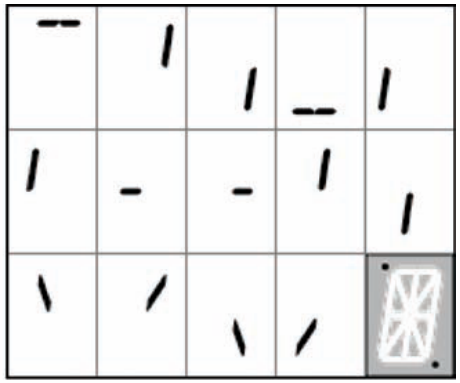
$\mathbf{R}$

NMF factors correspond to parts of faces.

Lee and Seung, 1999.

Decomposing digits (NMF)

D

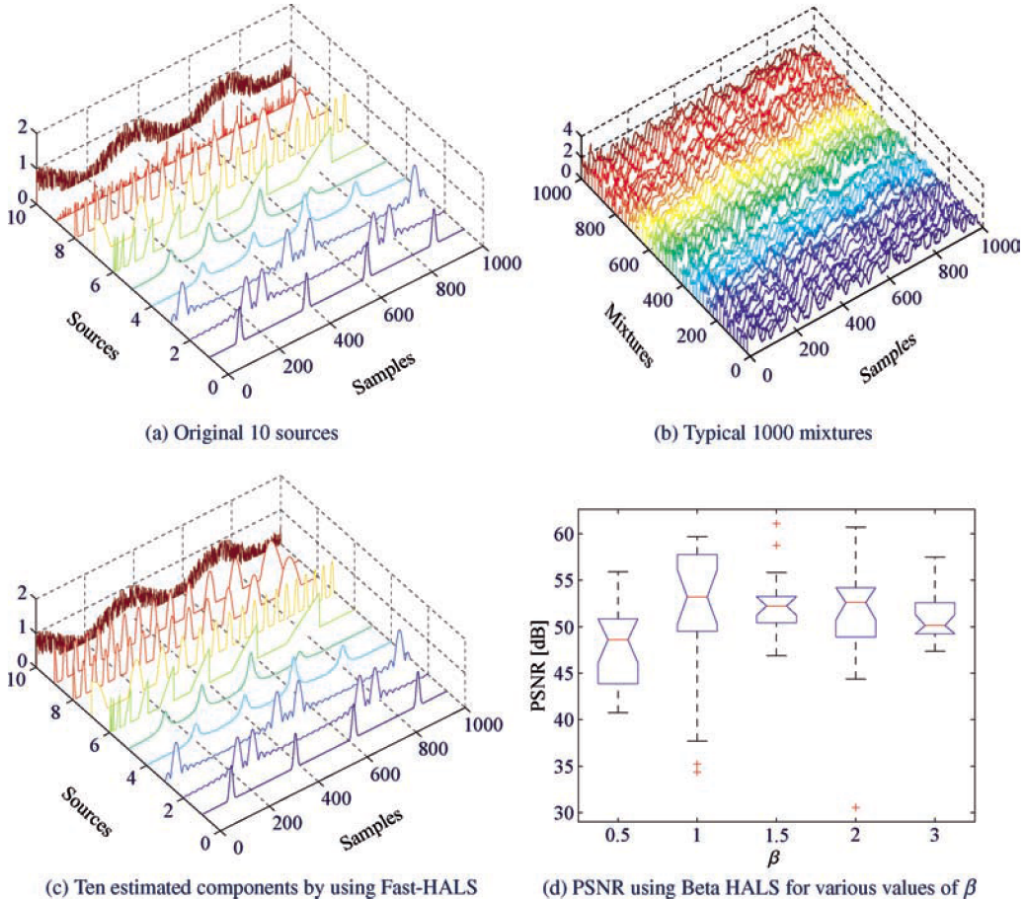


R

NMF factors correspond to parts of digits and "background".

Some applications

- Text mining (more later)
- Bioinformatics
- Microarray analysis
- Mineral exploration
- Neuroscience
- Image understanding
- Air pollution research
- Chemometrics
- Spectral data analysis
- Linear sparse coding
- Image classification
- Clustering
- Neural learning process
- Sound recognition
- Remote sensing
- Object characterization
- ...



(a) Original 10 sources (b) Typical 1000 mixtures

(c) Ten estimated components by using Fast-HALS (d) PSNR using Beta HALS for various values of β

Figure 4.8 Illustration for (a) benchmark used in large-scale experiments with 10 nonnegative sources; (b) Typical 1000 mixtures; (c) Ten estimated components by using FAST HALS NMF from the observations matrix $\mathbf{Y}$ of dimension 1000×1000. (d) Performance expressed via the PSNR using the Beta HALS NMF algorithm for $\beta = 0.5, 1, 1.5, 2$ and 3.

Cichocki et al., 2009.

Gaussian NMF

- Gaussian NMF is the most basic form of non-negative factorizations:

$$\begin{aligned} \text{minimize} \quad & \|\mathbf{D} - \mathbf{LR}\|_F^2 \\ \text{s. t.} \quad & \mathbf{L} \in \mathbb{R}_+^{m \times r} \\ & \mathbf{R} \in \mathbb{R}_+^{r \times n} \end{aligned}$$

- Truncated SVD minimizes the same objective (but without non-negativity constraints)
- Many other variants exist
  - Different objective functions (e.g., KL-divergence)
  - Additional regularizations (e.g., L_1-regularization)
  - Different constraints (e.g., orthogonality of $\mathbf{R}$)
  - Different compositions (e.g., 3 matrices)
  - multi-layer NMF, semi-NMF, sparse NMF, tri-NMF, symmetric NMF, orthogonal NMF, non-smooth NMF (nsNMF), overlapping NMF, convolutive NMF (CNMF), k-Means, ...

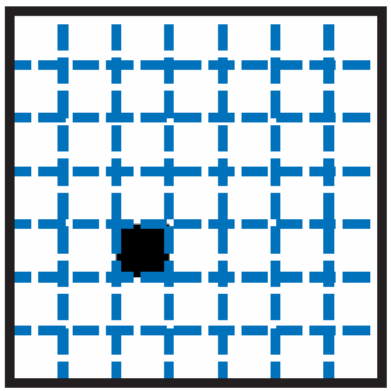
k-Means can be seen as a variant of NMF

$\mathbf{D}_{i*}$ (original)

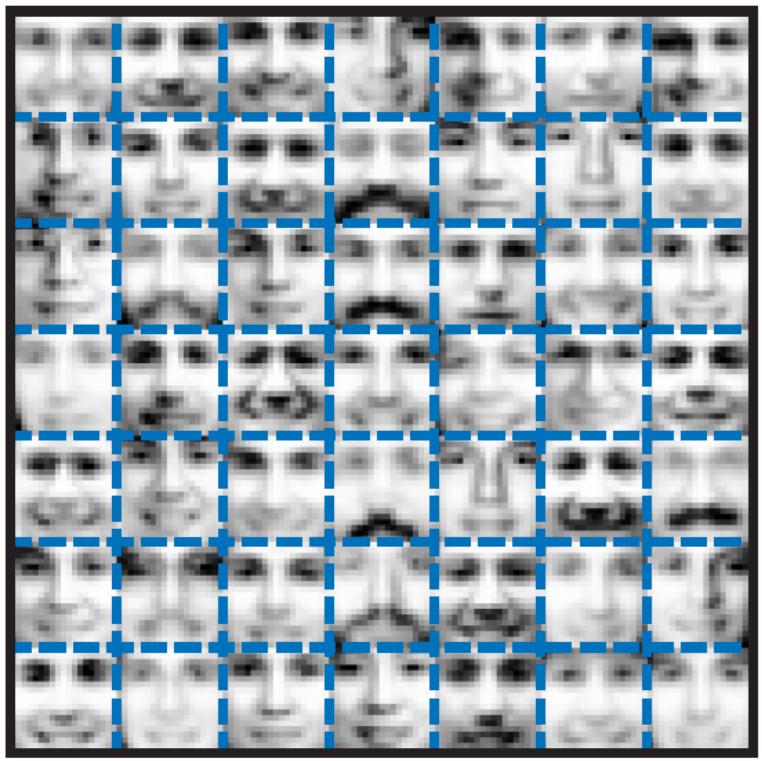
Additional constraint: $\mathbf{L}$ contains exactly one 1 in each row, rest 0

$[\mathbf{LR}]_{i*}$

$=$



$\mathbf{L}_{i*}$



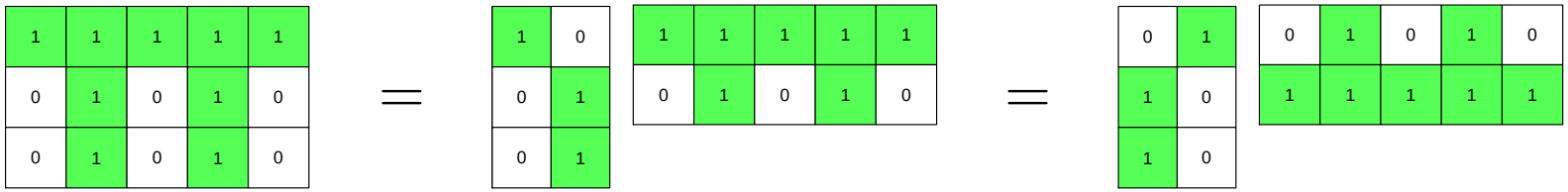
$\mathbf{R}$

k-Means factors correspond to prototypical faces.

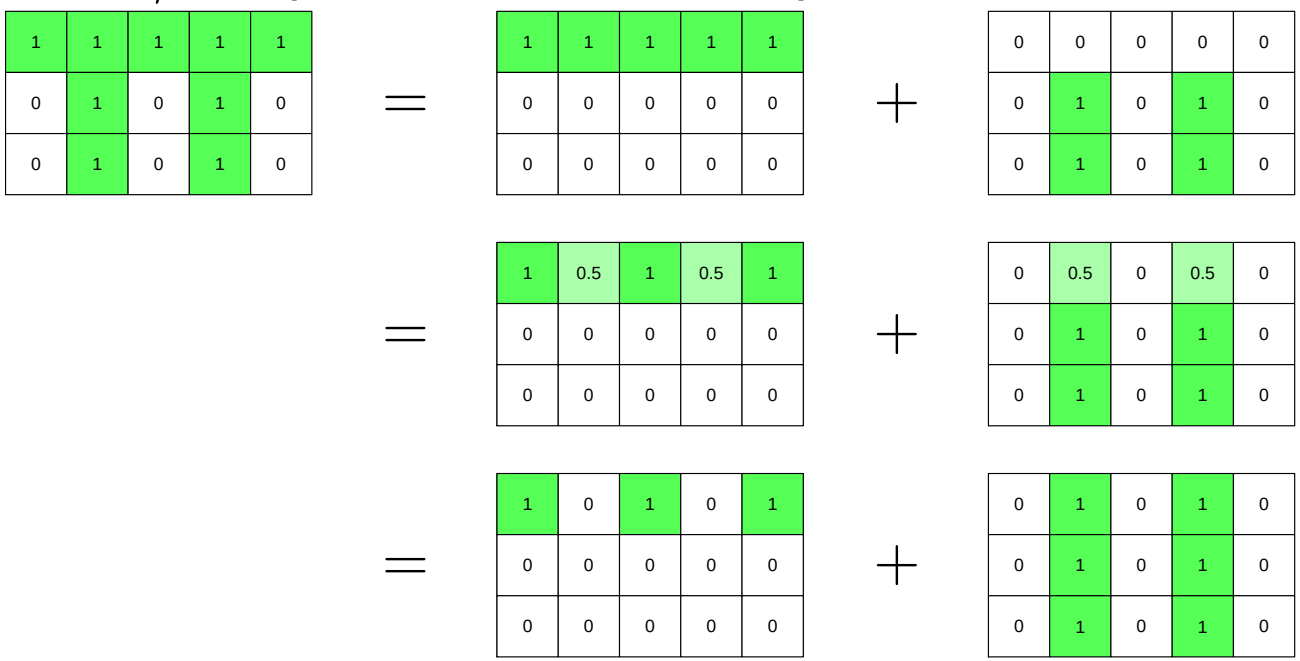
Lee and Seung, 1999.

NMF is not unique

- Factors are not "ordered"



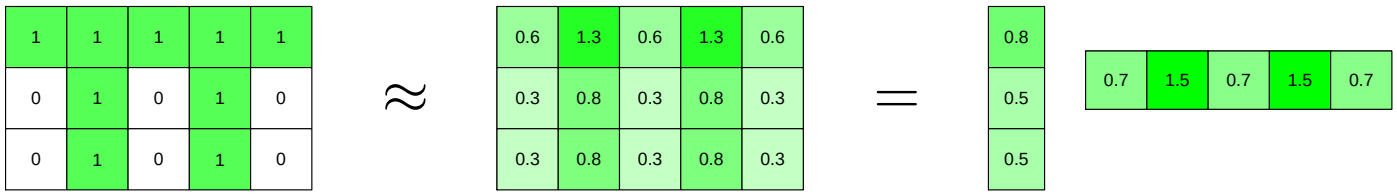
- One way of ordering: decreasing Frobenius norm of components (i.e., order by $\|\mathbf{L}_{*k}\mathbf{R}_{k*}\|_F$)
- Factors/components are not unique



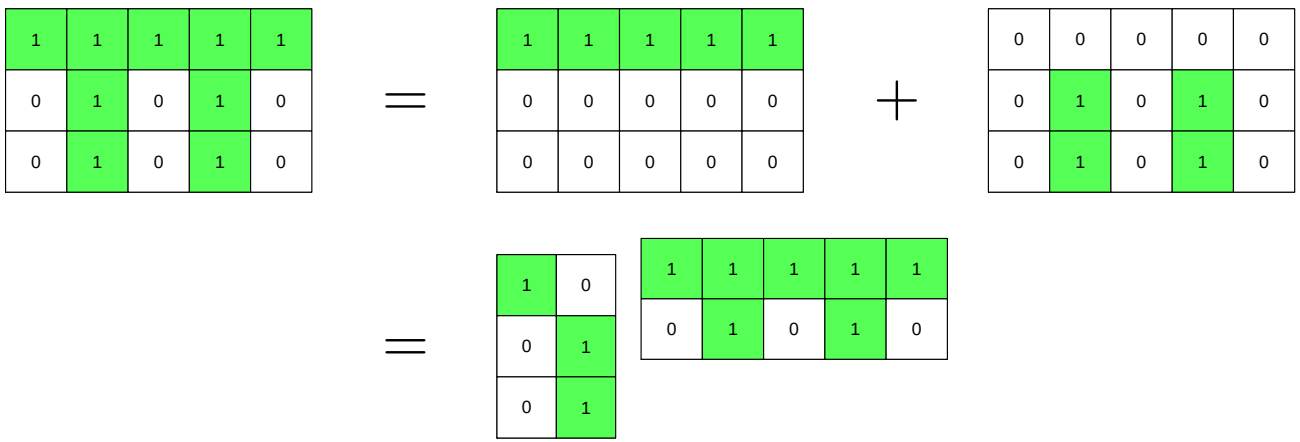
Additional constraints or regularization can encourage uniqueness.

NMF is not hierarchical

- Rank-1 NMF



- Rank-2 NMF



- Best rank-k approximation may differ significantly from best rank-$(k-1)$ approximation
- Rank influences sparsity, interpretability, and statistical fidelity
- Optimum choice of rank is not well-studied (often requires experimentation)

Outline

NMF is difficult

We focus on minimizing $L(\mathbf{L}, \mathbf{R}) = \|\mathbf{D} - \mathbf{LR}\|_F^2$.

- For varying m, n, and r, problem is NP-hard
- When $\text{rank}(\mathbf{D}) = 1$ (or $r = 1$), can be solved in polynomial time
  1. Take first non-zero column of $\mathbf{D}$ as $\mathbf{L}_{m\times 1}$
  2. Determine $\mathbf{R}_{1\times n}$ entry by entry (using the fact that $\mathbf{D}_{*j} = \mathbf{LR}_{1j}$)
- Problem is not convex
  - Local optimum may not correspond to global optimum
  - Generally little hope to find global optimum
- But: Problem is biconvex
  - For fixed $\mathbf{R}$, $f(\mathbf{L}) = \|\mathbf{D} - \mathbf{LR}\|_F^2$ is convex

$$
\begin{aligned}
f(\mathbf{L}) &= \textstyle\sum_i \|\mathbf{D}_{i*} - \mathbf{L}_{i*}\mathbf{R}\|_F^2 && \text{(chain rule)} \\
\nabla_{\mathbf{L}_{ik}} f(\mathbf{L}) &= -2(\mathbf{D}_{i*} - \mathbf{L}_{i*}\mathbf{R})\mathbf{R}_{k*}^T && \text{(product rule)} \\
\nabla^2_{\mathbf{L}_{ik}} f(\mathbf{L}) &= 2\mathbf{R}_{k*}\mathbf{R}_{k*}^T \geq 0 && \text{(does not depend on } \mathbf{L}\text{)}
\end{aligned}
$$

  - For fixed $\mathbf{L}$, $f(\mathbf{R}) = \|\mathbf{D} - \mathbf{LR}\|_F^2$ is convex
  - Allows for efficient algorithms

General framework

- Gradient descent generally slow
- Stochastic gradient descent inappropriate
- Key approach: alternating minimization

1: Pick starting point $\mathbf{L}_0$ and $\mathbf{R}_0$
2: **while** not converged **do**
3: Keep $\mathbf{R}$ fixed, optimize $\mathbf{L}$
4: Keep $\mathbf{L}$ fixed, optimize $\mathbf{R}$
5: **end while**

- Update steps 3 and 4 easier than full problem
- Also called *alternating projections* or *(block) coordinate descent*
- Starting point
  - Random
  - Multi-start initialization: try multiple random starting points, run a few epochs, continue with best
  - Based on SVD
  - ...

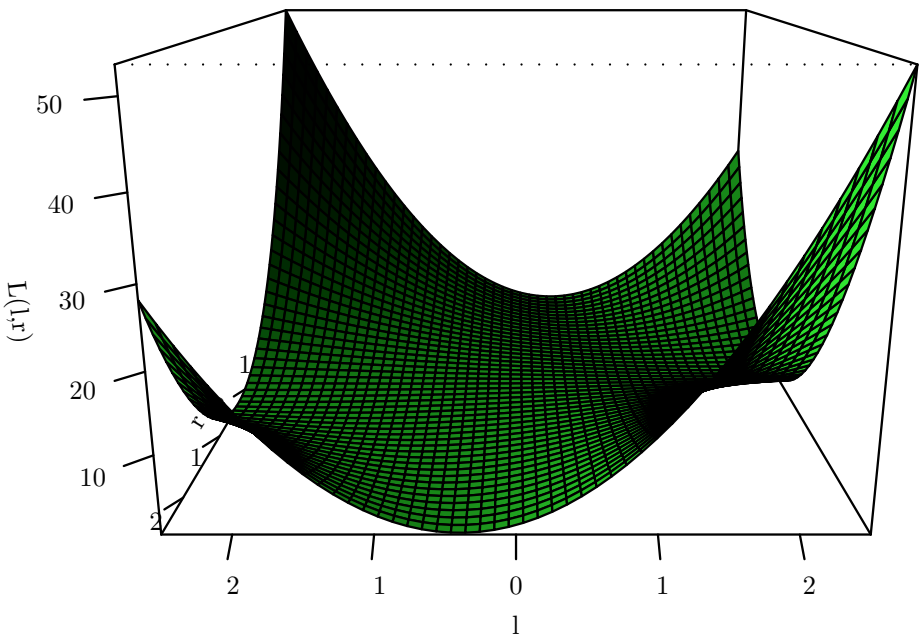
Example

Ignore non-negativity for now. Consider the regularized least-square error:

$$L(\mathbf{L}, \mathbf{R}) = \|\mathbf{D} - \mathbf{LR}\|_F^2 + \lambda(\|\mathbf{L}\|_F^2 + \|\mathbf{R}\|_F^2)$$

By setting $m = n = r = 1$, $\mathbf{D} = (1)$ and $\lambda = 0.05$, we obtain

$$L(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2)$$



$$\nabla_l f(l) = -2r(1 - lr) + 0.1l$$
$$\nabla_r f(r) = -2l(1 - lr) + 0.1r$$

Local optima:

$$\left(\sqrt{\frac{19}{20}}, \sqrt{\frac{19}{20}}\right), \left(-\sqrt{\frac{19}{20}}, -\sqrt{\frac{19}{20}}\right)$$

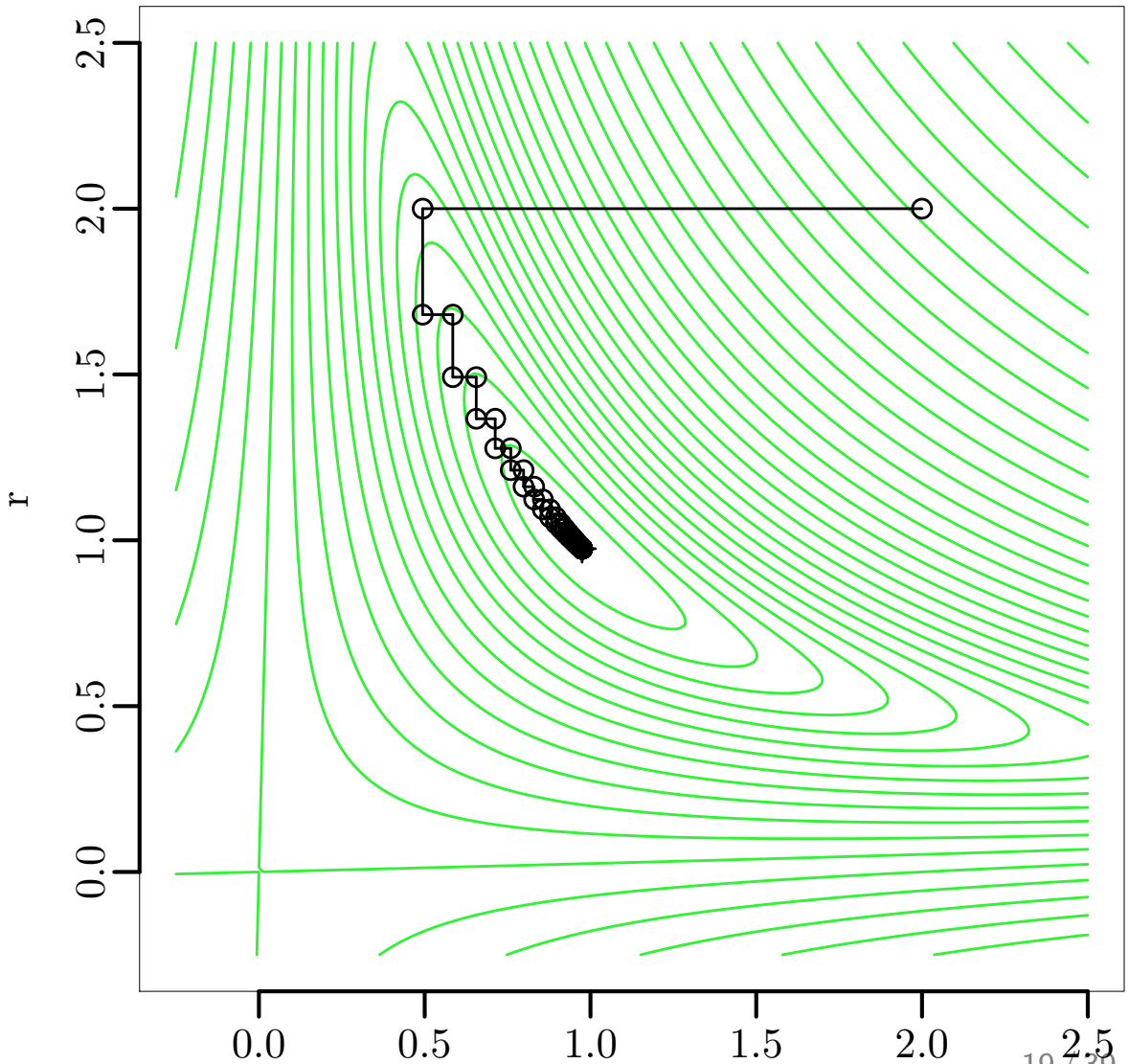
Stationary point: (0,0)

Example (ALS)

- $f(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2)$
- $l \leftarrow \min_l f(l) = \frac{2r}{2r^2+0.1}$
- $r \leftarrow \min_r f(r) = \frac{2l}{2l^2+0.1}$
-

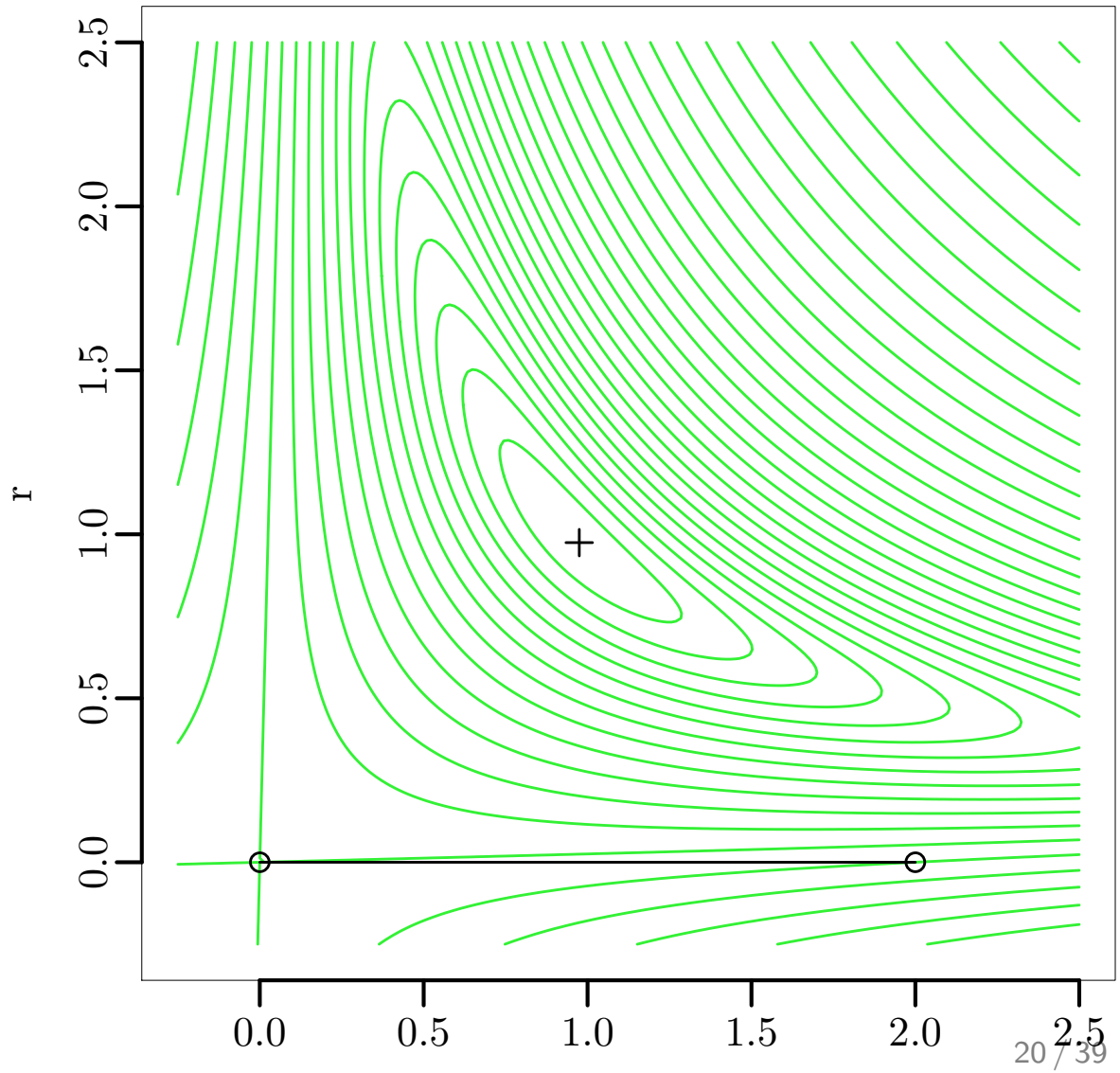
Step	l	r
0	2	2
1	0.49	2
2	0.49	1.68
3	0.58	1.68
4	0.58	1.49
⋮	⋮	⋮
100	0.97	0.97

- Converges to local minimum



Example (ALS)

- $f(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2)$
- $l \leftarrow \min_l f(l) = \frac{2r}{2r^2+0.1}$
- $r \leftarrow \min_r f(r) = \frac{2l}{2l^2+0.1}$
Step	l	r
0	2	0
0	0	0
0	0	0
⋮	⋮	⋮
- Converges to stationary point



Alternating non-negative least squares (ANLS)

- Uses non-negative least squares approximation of $\mathbf{L}$ and $\mathbf{R}$:

$$\underset{\mathbf{L}\in\mathbb{R}_+^{m\times r}}{\operatorname{argmin}}\|\mathbf{D}-\mathbf{L}\mathbf{R}\|_F^2 \qquad \text{and} \qquad \underset{\mathbf{R}\in\mathbb{R}_+^{r\times n}}{\operatorname{argmin}}\|\mathbf{D}-\mathbf{L}\mathbf{R}\|_F^2$$

- Equivalently: find non-negative least squares solution to $\mathbf{L}\mathbf{R}=\mathbf{D}$
- Common approach: Solve unconstrained least squares problems and "remove" negative values. E.g., when columns (rows) of $\mathbf{L}$ ($\mathbf{R}$) are linearly independent, set

$$\mathbf{L}=[\mathbf{D}\mathbf{R}^\dagger]_\epsilon \qquad \text{and} \qquad \mathbf{R}=[\mathbf{L}^\dagger\mathbf{D}]_\epsilon$$

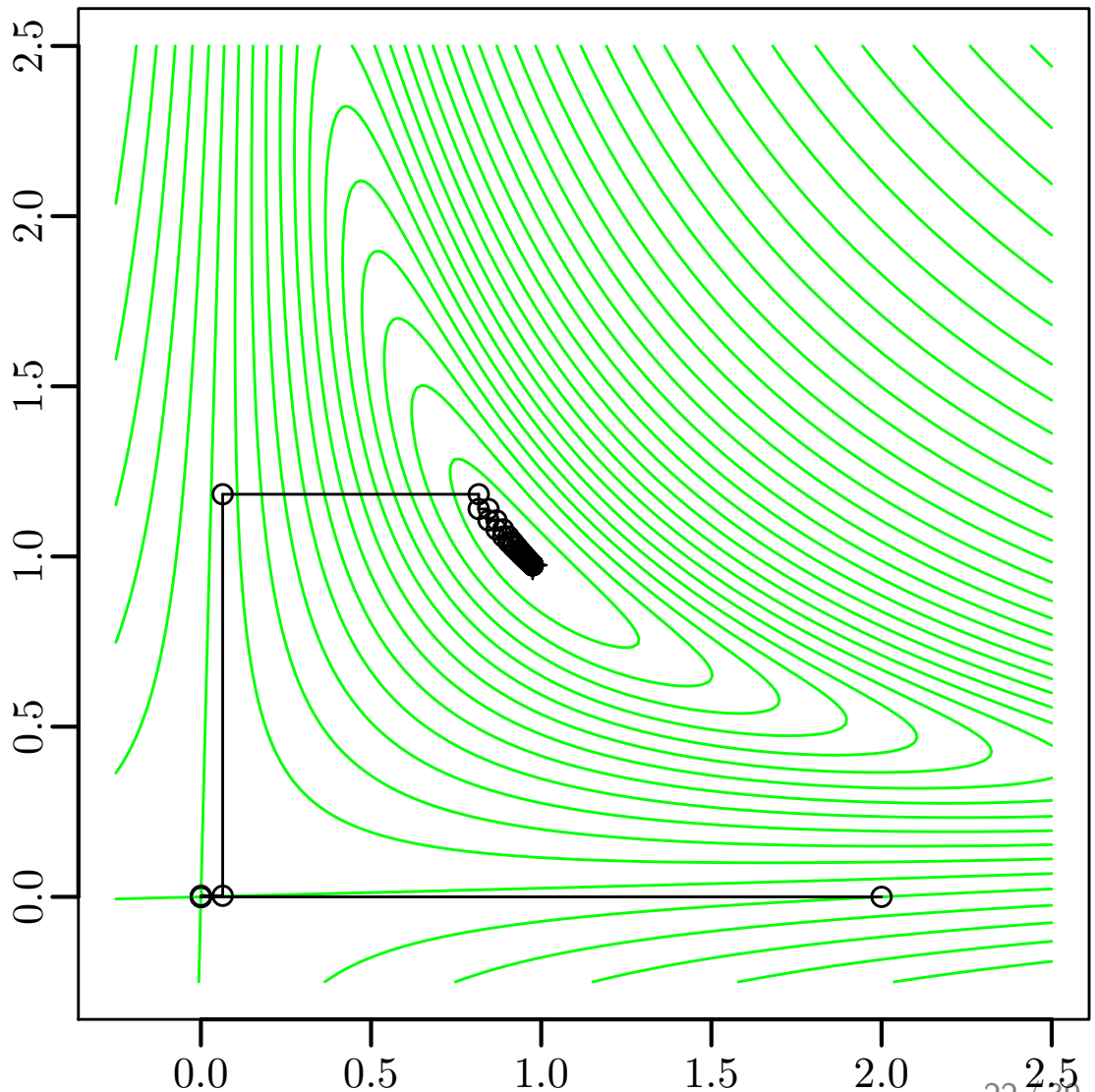
  where
  - $\mathbf{R}^\dagger=\mathbf{R}^T(\mathbf{R}\mathbf{R}^T)^{-1}$ is the right pseudo-inverse of $\mathbf{R}$
  - $\mathbf{L}^\dagger=(\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T$ is the left pseudo-inverse of $\mathbf{L}$
  - $[a]_\epsilon=\max\{\epsilon,a\}$ for $\epsilon=0$ or some small constant (e.g., $\epsilon=10^{-9}$)
- Difficult to analyze due to non-linear update steps
- Often slow convergence to a "bad" local minimum (better when regularized)

Example (ANLS)

- $f(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2)$ and set $\epsilon = 10^{-9}$
- $l \leftarrow \left[\frac{2r}{2r^2+0.1}\right]_\epsilon$
- $r \leftarrow \left[\frac{2l}{2l^2+0.1}\right]_\epsilon$

Step	l	r
0	2	0
1	$1 \cdot 10^{-9}$	0
2	$1 \cdot 10^{-9}$	$2 \cdot 10^{-8}$
3	$4 \cdot 10^{-7}$	$2 \cdot 10^{-8}$
4	$4 \cdot 10^{-7}$	$8 \cdot 10^{-6}$
$\vdots$	$\vdots$	$\vdots$
100	0.97	0.97

- Converges to local minimum



Hierarchical alternating least squares (HALS)

- Work locally on a single factor, then proceed to next factor, and so on
- Let $\mathbf{D}^{(k)}$ be the residual matrix (error) when k-th factor is removed:

$$\mathbf{D}^{(k)} = \mathbf{D} - \mathbf{L}\mathbf{R} + \mathbf{L}_{*k}\mathbf{R}_{k*} = \mathbf{D} - \sum_{k' \neq k} \mathbf{L}_{*k'}\mathbf{R}_{k'*}$$

- HALS minimizes $\|\mathbf{D}^{(k)} - \mathbf{L}_{*k}\mathbf{R}_{k*}\|_F^2$ for $k = 1, 2, \ldots, r, 1, \ldots$
  (equivalently: finds best solution for k-th factor, fixing the rest)
- In each iteration, set (once or multiple times):

$$\mathbf{L}_{*k} = \frac{1}{\|\mathbf{R}_{k*}\|_F^2} \left[\mathbf{D}^{(k)}\mathbf{R}_{k*}^T\right]_\epsilon \quad \text{and} \quad \mathbf{R}_{k*}^T = \frac{1}{\|\mathbf{L}_{*k}\|_F^2} \left[(\mathbf{D}^{(k)})^T\mathbf{L}_{*k}\right]_\epsilon$$

- $\mathbf{D}^{(k)}$ can be incrementally maintained $\rightarrow$ fast implementation

$$\mathbf{D}^{(k+1)} = \mathbf{D}^{(k)} + \mathbf{L}_{*k}\mathbf{R}_{k*} - \mathbf{L}_{*(k+1)}\mathbf{R}_{(k+1)*}$$

- Often better performance in practice than ANLS
- Converges to stationary point when initialized with positive matrix and sufficiently small ϵ

Multiplicative updates

- Gradient descent step with step size η_{kj}

$$\mathbf{R}_{kj} \leftarrow \mathbf{R}_{kj} + \eta_{kj}([\mathbf{L}^T\mathbf{D}]_{kj} - [\mathbf{L}^T\mathbf{L}\mathbf{R}]_{kj})$$

- Setting $\eta_{kj} = \frac{\mathbf{R}_{kj}}{(\mathbf{L}^T\mathbf{L}\mathbf{R})_{kj}}$, we obtain the multiplicative update rules

$$\mathbf{L} \leftarrow \mathbf{L} \circ \frac{\mathbf{D}\mathbf{R}^T}{\mathbf{L}\mathbf{R}\mathbf{R}^T} \quad \text{and} \quad \mathbf{R} \leftarrow \mathbf{R} \circ \frac{\mathbf{L}^T\mathbf{D}}{\mathbf{L}^T\mathbf{L}\mathbf{R}},$$

  where multiplication ($\circ$) and division are element-wise
- Does not necessarily find optimum $\mathbf{L}$ (or $\mathbf{R}$), but can be shown to never increase loss
- Faster than ANLS (no computation of pseudo-inverse), easy to implement and parallelize
- Zeros in factors are problematic (divisions become undefined)

$$\mathbf{L} \leftarrow \mathbf{L} \circ \frac{[\mathbf{D}\mathbf{R}^T]_\epsilon}{\mathbf{L}\mathbf{R}\mathbf{R}^T + \epsilon} \quad \text{and} \quad \mathbf{R} \leftarrow \mathbf{R} \circ \frac{[\mathbf{L}^T\mathbf{D}]_\epsilon}{\mathbf{L}^T\mathbf{L}\mathbf{R} + \epsilon}$$

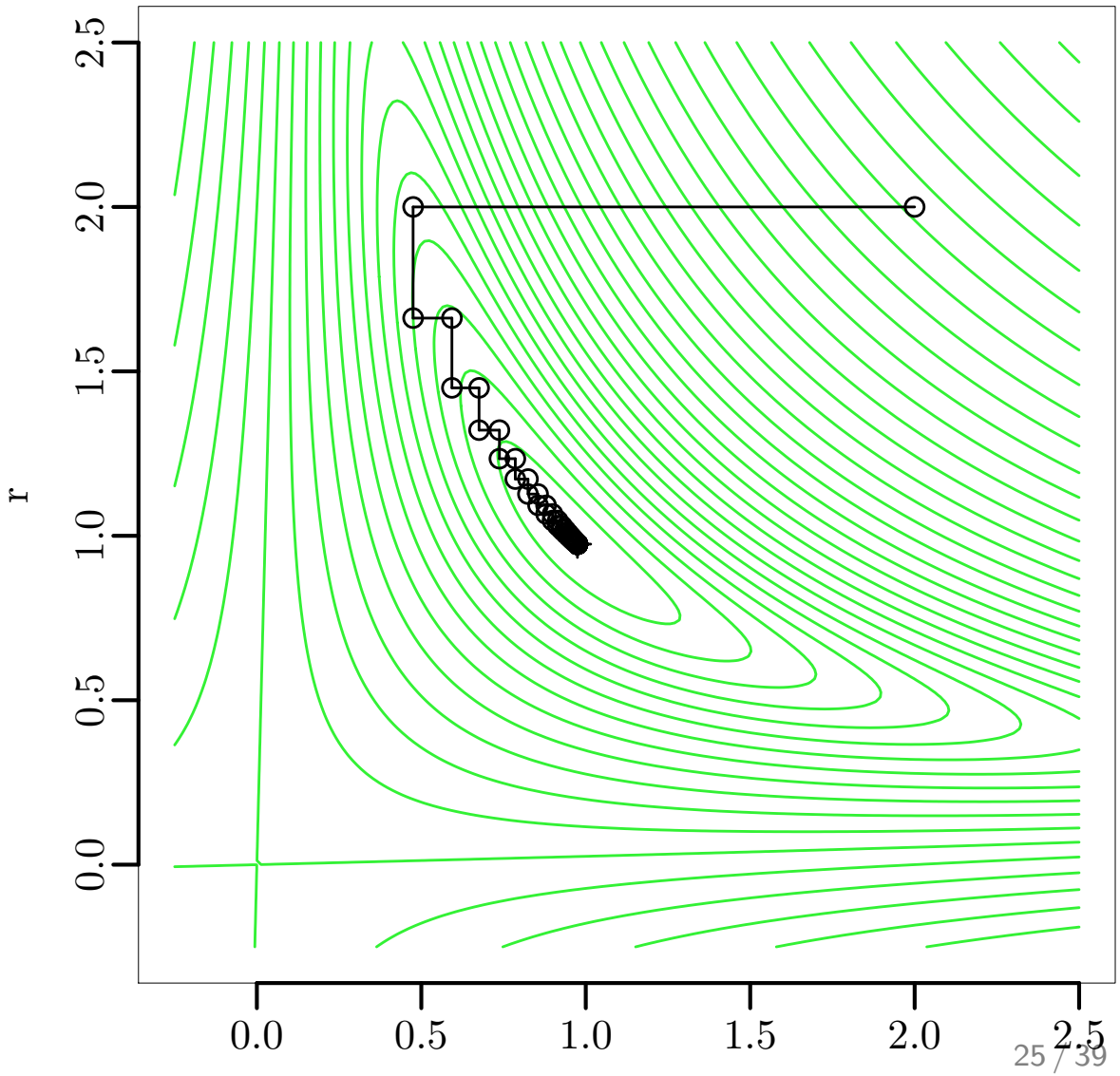
Lee and Seung, 2001. Cichocki et al., 2006.

Example (multiplicative updates)

- $f(l, r) = (1 - lr)^2 + 0.05(l^2 + r^2)$
- $l \leftarrow l\frac{1r - 0.05l}{lr^2}$
- $r \leftarrow r\frac{l1 - 0.05r}{l^2 r}$

Step	l	r
0	2	2
1	0.48	2
2	0.48	1.66
3	0.59	1.66
4	0.58	1.45
⋮	⋮	⋮
100	0.97	0.97

- Converges to local minimum



Outline

Topic modeling

- Consider a document-word matrix constructed from some corpus

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

- Documents seem to talk about two "topics"
  1. Environment (with words air, water, and pollution)
  2. Congress (with words democrat and republican)

Can we automatically detect topics in documents?

A probabilistic viewpoint

- Let's normalize such that the entries sum to unity

$$\mathbf{D} = \begin{array}{c} \\ \text{doc 1} \\ \text{doc 2} \\ \text{doc 3} \\ \text{doc 4} \\ \text{doc 5} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{air} & \text{water} & \text{pollution} & \text{democrat} & \text{republican} \end{array} \\ \begin{pmatrix} 0.04 & 0.03 & 0.12 & 0.00 & 0.00 \\ 0.01 & 0.06 & 0.17 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.14 & 0.16 \\ 0.00 & 0.00 & 0.00 & 0.12 & 0.07 \\ 0.01 & 0.01 & 0.01 & 0.01 & 0.015 \end{pmatrix} \end{array}$$

- Put all words in an urn and draw. The probability to draw word w from document d is given by

$$P(d, w) = \mathbf{D}_{dw}$$

- Matrix $\mathbf{D}$ can represent *any* probability distribution
- pLSA tries to find a distribution that is "close" to $\mathbf{D}$ but exposes information about topics

Probabilistic latent semantic analysis (pLSA)

Definition (pLSA, NMF formulation)

Given a rank r, find matrices $\mathbf{L}$, $\mathbf{\Sigma}$, and $\mathbf{R}$ such that

$$\mathbf{D} \approx \mathbf{L}\mathbf{\Sigma}\mathbf{R}$$

where

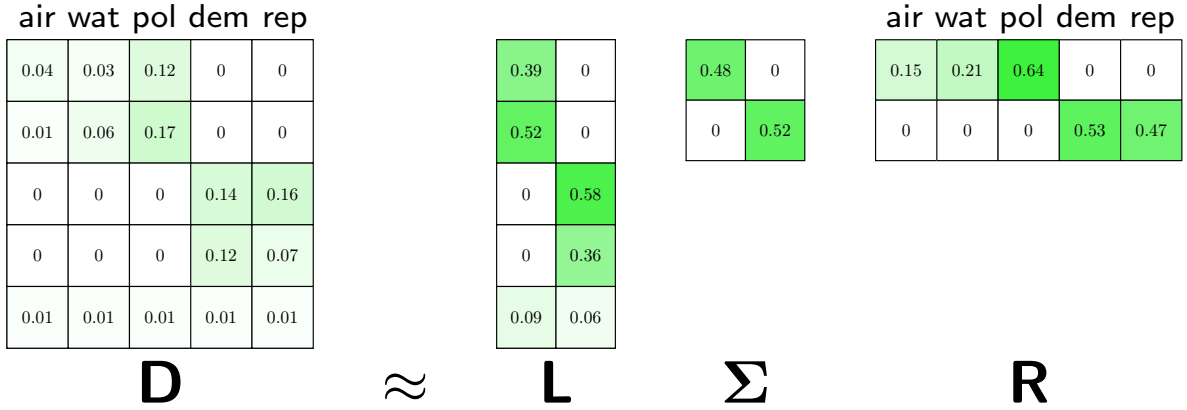
- $\mathbf{L}_{m \times r}$ is a non-negative, column-stochastic matrix (columns sum to unity),
- $\mathbf{\Sigma}_{r \times r}$ is a non-negative, diagonal matrix that sums to unity, and
- $\mathbf{R}_{r \times n}$ is a non-negative, row-stochastic matrix (rows sum to unity).

- $\approx$ is usually taken to be the (generalized) KL divergence
- Additional regularization or tempering necessary to avoid overfitting

Hofmann, 2001.

Example

- pLSA factorization of example matrix



- Rank r corresponds to number of topics
- $\mathbf{\Sigma}_{kk}$ corresponds to overall frequency of topic k
- $\mathbf{L}_{dk}$ corresponds to contribution of document d to topic k
- $\mathbf{R}_{kw}$ corresponds to frequency of word w in topic k
- pLSA constraints allow for probabilistic interpretation
  $P(d, w) \approx [\mathbf{L\Sigma R}]_{dw} = \sum_k \mathbf{\Sigma}_{kk} \mathbf{L}_{dk} \mathbf{R}_{kw} = \sum_k P(k) P(d \mid k) P(w \mid k)$
- pLSA model imposes conditional independence constraints
  $\rightarrow$ restricted space of distributions

Another 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

Hofmann, 2004.

pLSA geometry

- Rewrite probabilistic formulation

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

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

- Generative process of creating a word
  1. Pick a document according to $P(d)$
  2. Select a topic acc. to $P(z \mid d)$
  3. Select a word acc. to $P(w \mid z)$

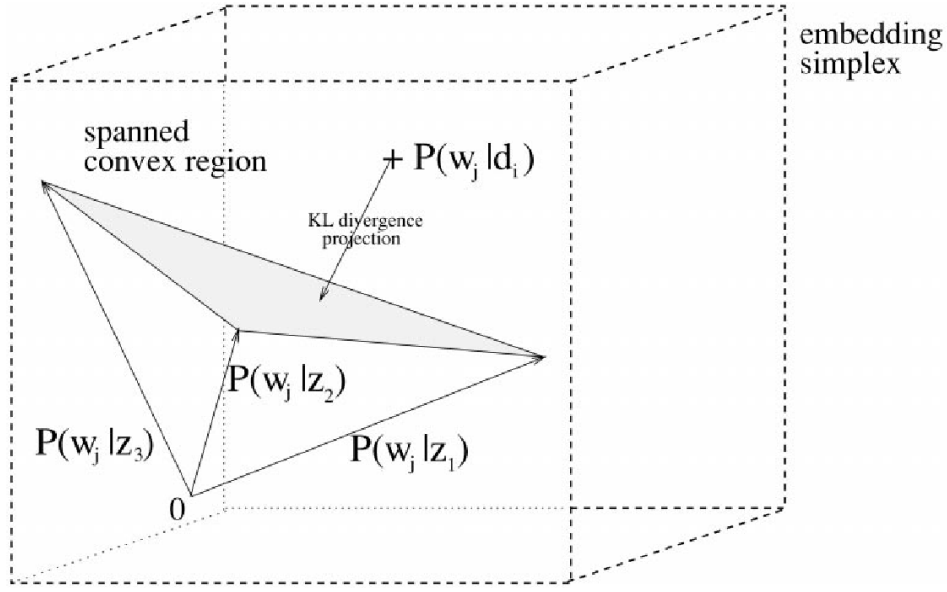


Figure 2. Sketch of the probability simplex and a convex region spanned by class-conditional probabilities in the aspect model.

Hofmann, 2001.

Kullback-Leibler divergence (1)

- Let $\tilde{\mathbf{D}}$ be the unnormalized word-count data and denote by N total number of words
- Likelihood of seeing $\tilde{\mathbf{D}}$ when drawing N words with replacement is proportional to

$$\prod_{d=1}^{m} \prod_{w=1}^{n} P(d, w)^{\tilde{\mathbf{D}}_{dw}}$$

- pLSA maximizes the log-likelihood of seeing the data given the model

$$\begin{aligned}
\log P(\tilde{\mathbf{D}} \mid \mathbf{L}, \boldsymbol{\Sigma}, \mathbf{R}) &\propto \sum_{d=1}^{m} \sum_{w=1}^{n} \tilde{\mathbf{D}}_{dw} \log P(d, w \mid \mathbf{L}, \boldsymbol{\Sigma}, \mathbf{R}) \\
&\propto -\sum_{d=1}^{m} \sum_{w=1}^{n} \mathbf{D} \log \frac{1}{[\mathbf{L}\boldsymbol{\Sigma}\mathbf{R}]_{dw}} \\
&= \underbrace{-\sum_{d=1}^{m} \sum_{w=1}^{n} \mathbf{D}_{dw} \log \frac{\mathbf{D}_{dw}}{[\mathbf{L}\boldsymbol{\Sigma}\mathbf{R}]_{dw}}}_{\text{Kullback-Leibler divergence}} + c_{\mathbf{D}}
\end{aligned}$$

Gaussier and Goutte, 2005.

Kullback-Leibler divergence (2)

- KL divergence

$$D_{KL}(\mathbf{P}\|\mathbf{Q}) = \sum_{d=1}^{m}\sum_{w=1}^{n}\mathbf{P}_{dw}\log\frac{\mathbf{P}_{dw}}{\mathbf{Q}_{dw}}$$

- Interpretation: expected number of extra bits for encoding a value drawn from $\mathbf{P}$ using an optimum code for distribution $\mathbf{Q}$
- $D_{KL}(\mathbf{P}\|\mathbf{Q}) \geq 0$
- $D_{KL}(\mathbf{P}\|\mathbf{P}) = 0$
- $D_{KL}(\mathbf{P}\|\mathbf{Q}) \neq D_{KL}(\mathbf{Q}\|\mathbf{P})$
- NMF-based pLSA algorithms minimize the generalized KL divergence

$$D_{GKL}(\tilde{\mathbf{P}}\|\tilde{\mathbf{Q}}) = \sum_{d=1}^{m}\sum_{w=1}^{n}(\tilde{\mathbf{P}}_{dw}\log\frac{\tilde{\mathbf{P}}_{dw}}{\tilde{\mathbf{Q}}_{dw}} - \tilde{\mathbf{P}}_{dw} + \tilde{\mathbf{Q}}_{dw}),$$

where $\tilde{\mathbf{P}} = \tilde{\mathbf{D}}$ and $\tilde{\mathbf{Q}} = \mathbf{L}\tilde{\mathbf{\Sigma}}\mathbf{R}$

Multiplicative updates for GKL (w/o tempering)

- We first find a decomposition $\tilde{\mathbf{D}} \approx \tilde{\mathbf{L}}\tilde{\mathbf{R}}$, where $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{R}}$ are non-negative matrices
- Update rules

$$\tilde{\mathbf{L}} \leftarrow \tilde{\mathbf{L}} \circ \frac{\tilde{\mathbf{D}}}{\tilde{\mathbf{L}}\tilde{\mathbf{R}}} \tilde{\mathbf{R}}^T \operatorname{diag}(1/\operatorname{rowSums}(\tilde{\mathbf{R}}))$$

$$\tilde{\mathbf{R}} \leftarrow \tilde{\mathbf{R}} \circ \operatorname{diag}(1/\operatorname{colSums}(\tilde{\mathbf{L}}))\tilde{\mathbf{L}}^T \frac{\tilde{\mathbf{D}}}{\tilde{\mathbf{L}}\tilde{\mathbf{R}}}$$

- GKL is non-increasing under these update rules
- Normalize by rescaling columns of $\tilde{\mathbf{L}}$ and rows of $\tilde{\mathbf{R}}$ to obtain

$$\mathbf{L} = \tilde{\mathbf{L}} \operatorname{diag}(1/\operatorname{colSums}(\tilde{\mathbf{L}}))$$

$$\mathbf{R} = \operatorname{diag}(1/\operatorname{rowSums}(\tilde{\mathbf{R}}))\tilde{\mathbf{R}}$$

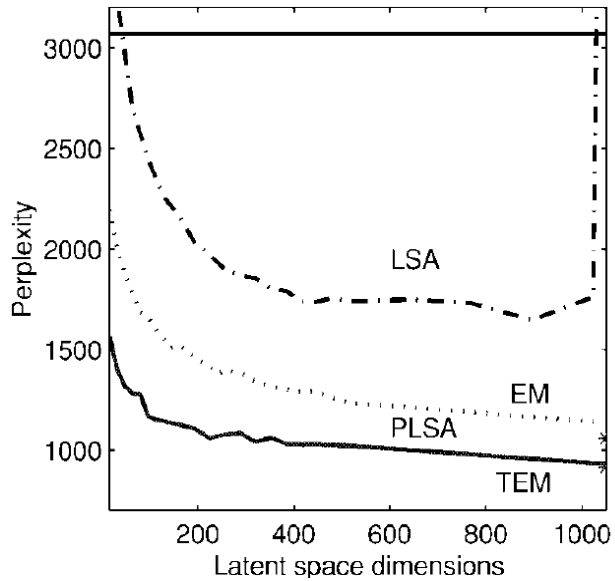
$$\tilde{\mathbf{\Sigma}} = \operatorname{diag}(\operatorname{colSums}(\tilde{\mathbf{L}}) \circ \operatorname{rowSums}(\tilde{\mathbf{R}}))$$

$$\mathbf{\Sigma} = \tilde{\mathbf{\Sigma}} / \textstyle\sum_k \tilde{\mathbf{\Sigma}}_{kk}$$

Lee and Seung, 2001.

Applications of pLSA

- Topic modeling
- Clustering documents
- Clustering terms
- Information retrieval
  - Treat query q as a "new" document (new row in $\tilde{\mathbf{D}}$ and $\mathbf{L}$)
  - Determine $P(k \mid q)$ by keeping $\mathbf{\Sigma}$ and $\mathbf{R}$ fixed ("fold in" the query)
  - Retrieve documents with similar topic mixture as query
  - Can deal with synonymy and polysemy
- Better generalization performance than LSA (=SVD), esp. with tempering
- In practice, outperformed by Latent Dirichlet Allocation (LDA)



Hofmann, 2001.

Outline

Lessons learned

- Non-negative matrix factorization (NMF) appears natural for non-negative data
- NMF encourages parts-based decomposition, interpretability, and (sometimes) sparseness
- Many variants, many applications
- Usually solved via alternating minimization algorithms
  - Alternating non-negative least squares (ANLS)
  - Projected gradient local hierarchical ALS (HALS)
  - Multiplicative updates
- pLSA is an approach to topic modeling that can be seen as an NMF

Literature

- David Skillicorn
  Understanding Complex Datasets: Data Mining with Matrix Decompositions (Chapter 8)
  Chapman and Hall, 2007
- Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari
  Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation
  Wiley, 2009
- Yifeng Li and Alioune Ngom
  The NMF MATLAB Toolbox
  `http://cs.uwindsor.ca/~li11112c/nmf`
- Renaud Gaujoux and Cathal Seoighe
  NMF R package
  `http://cran.r-project.org/web/packages/NMF/index.html`
- References given at bottom of slides