Chapter 3

# Non-Negative Matrix Factorization

Part 1: Introduction & computation

max planck institut
informatik

# Motivating NMF

Skillicorn chapter 8; Berry et al. (2007)

# Reminder

$A$

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |

=

$U_1\Sigma_{1,1}V_1^T$

| | | | | |
|---|---|---|---|---|
| 0.6 | 1.3 | 0.6 | 1.3 | 0.6 |
| 0.3 | 0.8 | 0.3 | 0.8 | 0.3 |
| 0.3 | 0.8 | 0.3 | 0.8 | 0.3 |

+

$U_2\Sigma_{2,2}V_2^T$

| | | | | |
|---|---|---|---|---|
| 0.4 | –0.3 | 0.4 | –0.3 | 0.4 |
| –0.3 | 0.2 | –0.3 | 0.2 | –0.3 |
| –0.3 | 0.2 | –0.3 | 0.2 | –0.3 |

The components of the SVD are not very interpretable

# Non-negative factors

$$
\underset{\boldsymbol{A}}{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}}
=
\underset{\boldsymbol{W}_1\boldsymbol{H}_1}{\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}}
+
\underset{\boldsymbol{W}_2\boldsymbol{H}_2}{\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}}
$$

$\boldsymbol{W}$ $\boldsymbol{H}$

Forcing the factors to be non-negative can, and often will, improve the interpretability of the factorization

# The definition

# Definition of NMF

Given a non-negative matrix $\boldsymbol{A} \in \mathbb{R}_+^{n \times m}$ and an integer $k$, find non-negative matrices $\boldsymbol{W} \in \mathbb{R}_+^{n \times k}$ and $\boldsymbol{H} \in \mathbb{R}_+^{k \times m}$ such that

$$\frac{1}{2} \|\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}\|_F^2$$

is minimized.

# Non-negative rank

- The **non-negative rank** of matrix $\boldsymbol{A}$, $\text{rank}_+(\boldsymbol{A})$, is the size of the smallest exact non-negative factorization $\boldsymbol{A} = \boldsymbol{W}\boldsymbol{H}$
  - $\text{rank}(\boldsymbol{A}) \leq \text{rank}_+(\boldsymbol{A}) \leq \min\{n, m\}$

# Some comments

- NMF is **not** unique
  - If $\boldsymbol{X}$ is nonnegative and with nonnegative inverse, then $\boldsymbol{W}\boldsymbol{X}\boldsymbol{X}^{-1}\boldsymbol{H}$ is equivalent valid decomposition
- Computing NMF (and non-negative rank) is NP-hard
  - This was open until 2008

# Example of non-uniqueness

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0.5 & 1 & 0.5 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

# NMF has no order

- The factors in NMF have no inherent order
  - The first component is no more important than the second is no more important…
- NMF is not **hierarchical**
  - The factors of rank-($k$+1) decomposition can be completely different to those of rank-$k$ decomposition

# Example

$$\begin{pmatrix}1&1&1&1&1\\0&1&0&1&0\\0&1&0&1&0\end{pmatrix} \approx \begin{pmatrix}0.8\\0.5\\0.5\end{pmatrix}\begin{pmatrix}0.7&1.5&0.7&1.5&0.7\end{pmatrix}$$

Rank-1

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

Rank-2

# Interpreting NMF

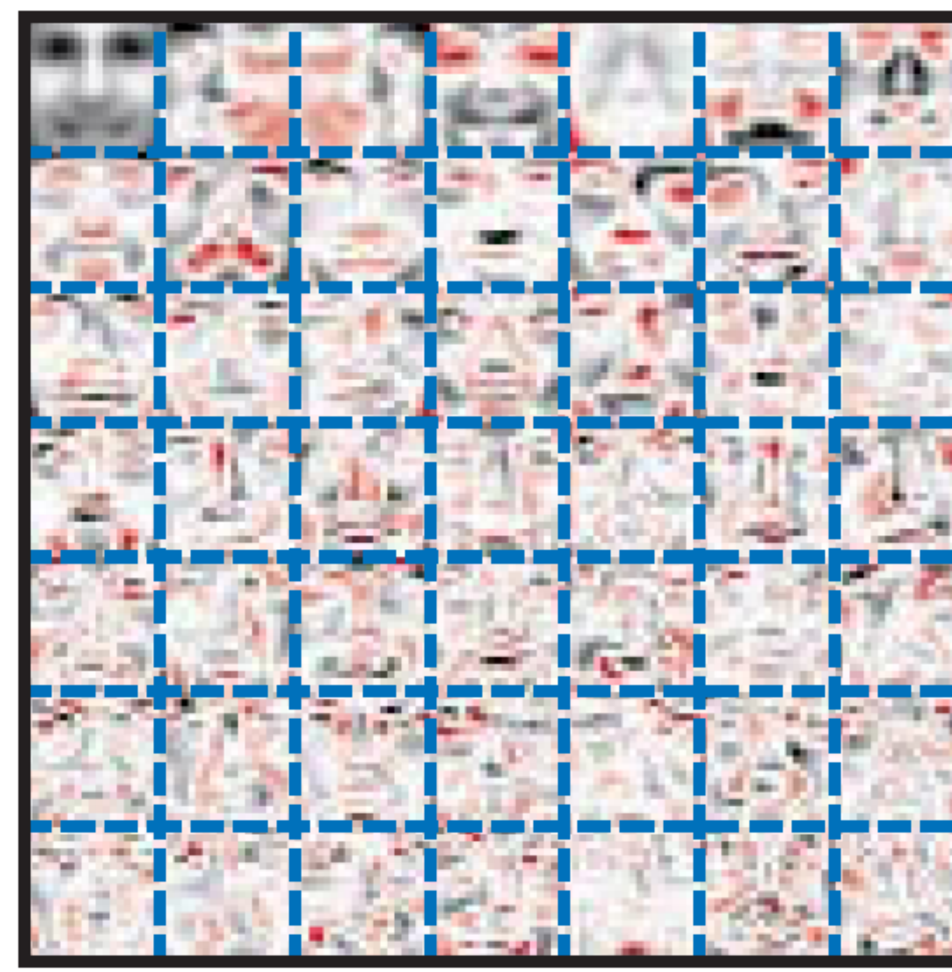
# Parts-of-whole

- NMF works over **anti-negative semiring**
  - There is no subtraction
- Each rank-1 component $\boldsymbol{w}_i\boldsymbol{h}_i$ explains a part of the whole
  - This can yield to sparse factors

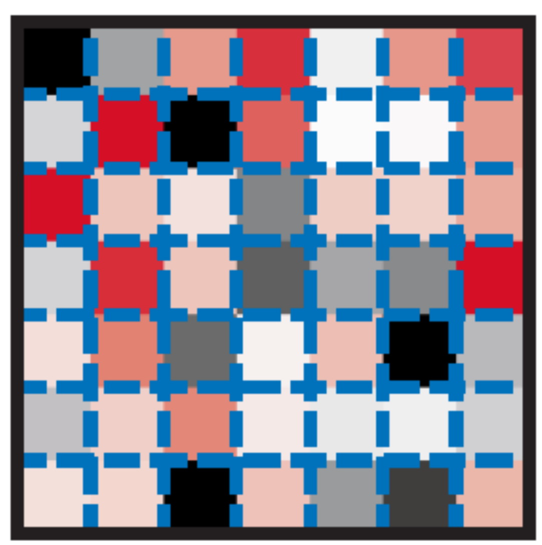
# NMF example: faces

Row of original

PCA/SVD



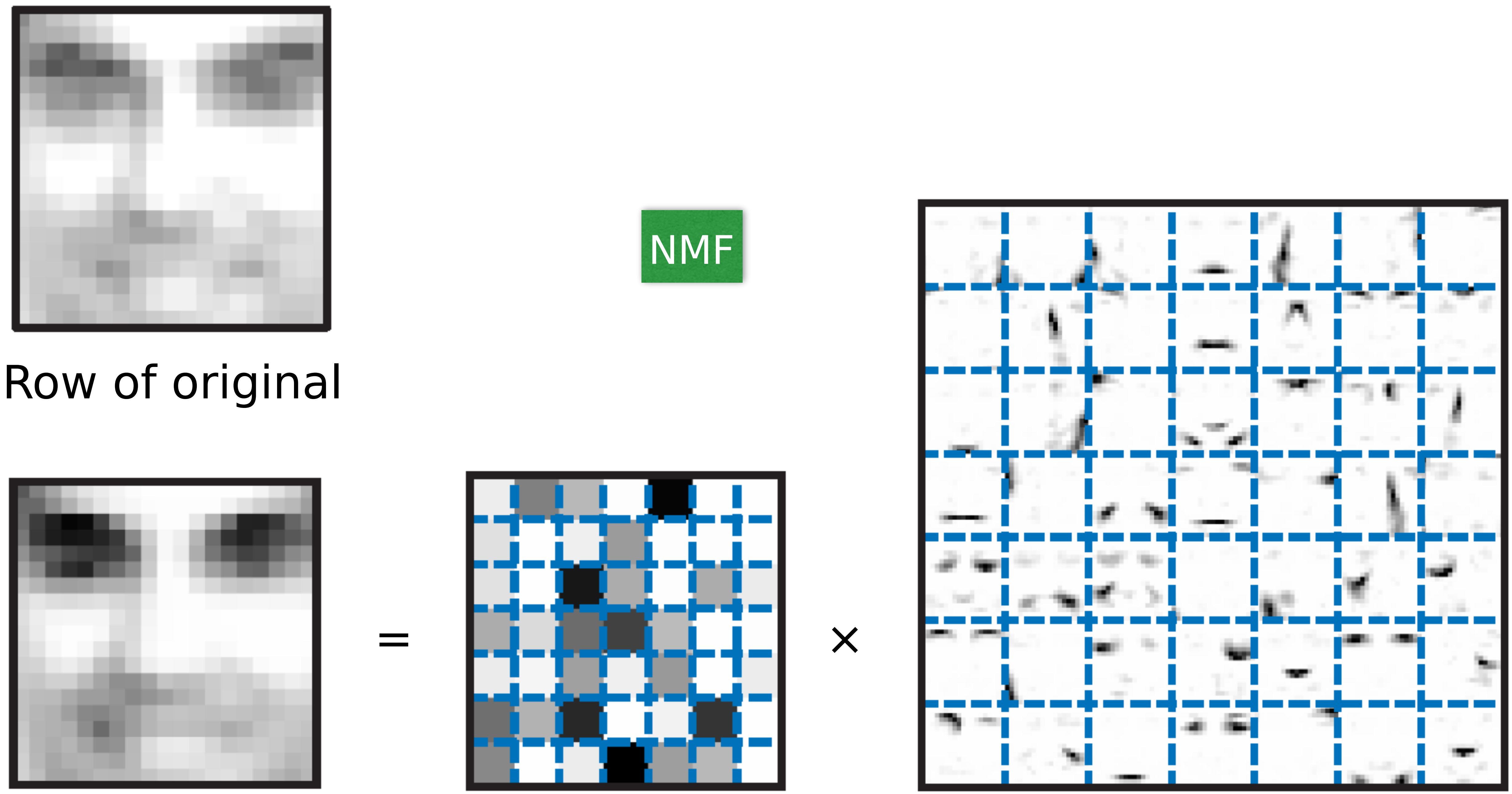
=



×

Row of reconstruction

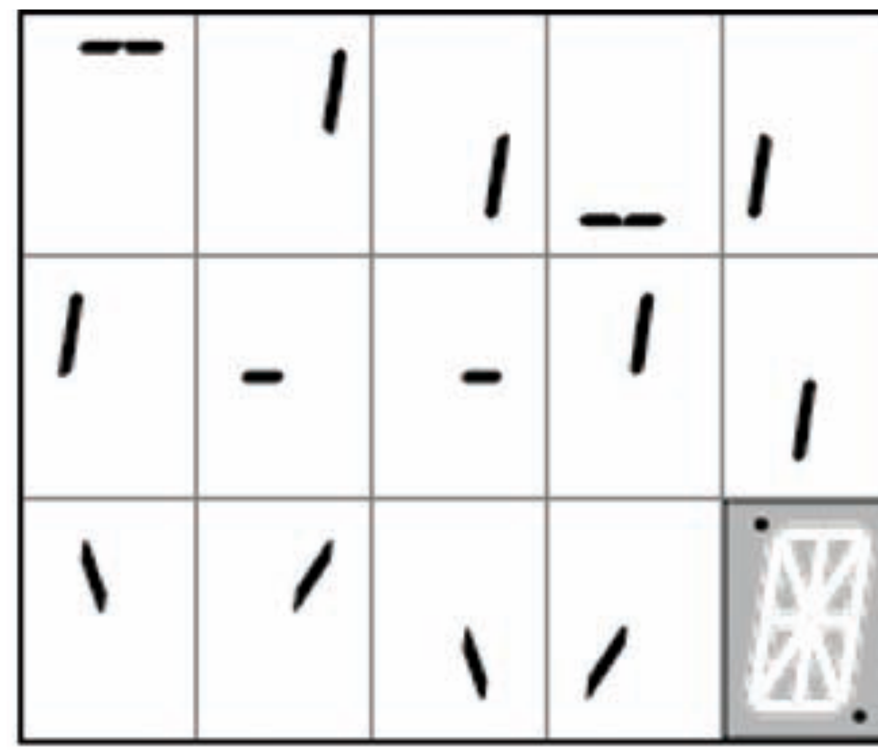
# NMF example: faces



# NMF example: digits

NMF factors correspond to patterns and background


***A***



***H***

# Some NMF applications

- Text mining (more later)
- Bioinformatics
- Microarray analysis
- Mineral exploration
- Neuroscience
- Image understanding
- Air pollution research
- Weather forecasting
- ...

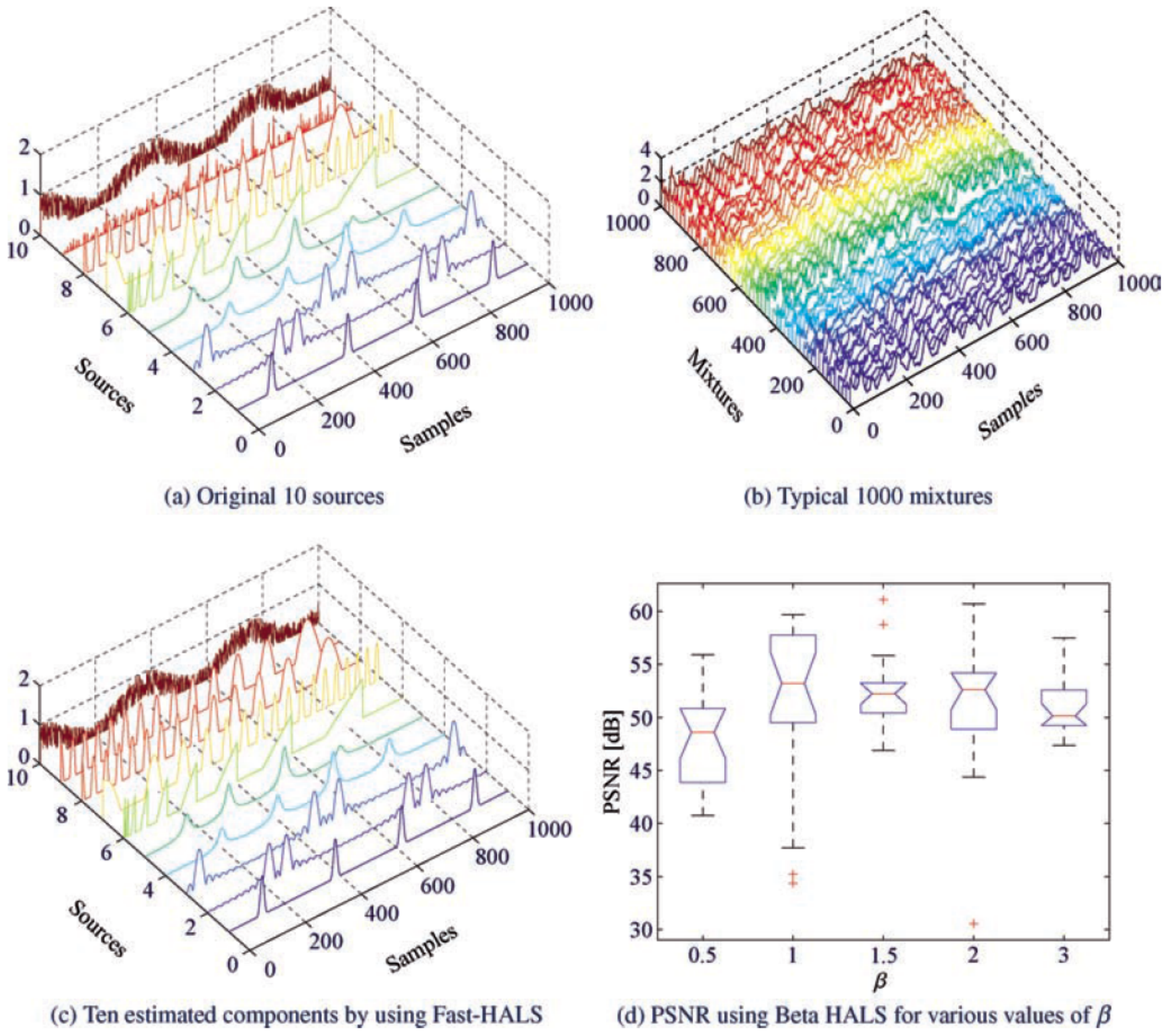



**Figure 4.8** Illustration for (a) benchmark used in large-scale experiments with 10 nonnegative sources; (b) Typical 1000 mixtures; (c) Ten estimated components by using FAST HALS NMF from the observations matrix **Y** of dimension $1000 \times 1000$. (d) Performance expressed via the PSNR using the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

# Computing NMF

# General idea

- NMF is not convex, but it is **biconvex**
  - If $\boldsymbol{W}$ is fixed, $\frac{1}{2}\|\boldsymbol{A}-\boldsymbol{W}\boldsymbol{H}\|_F^2$ is convex
- Start from random $\boldsymbol{W}$ and **repeat**
  - Fix $\boldsymbol{W}$ and update $\boldsymbol{H}$
  - Fix $\boldsymbol{H}$ and update $\boldsymbol{W}$
- **until** the error doesn't decrease anymore

# Notes on the general idea

- How to create a good random starting point?
    - Is the algorithm robust to initial solutions?
- How to update $\boldsymbol{W}$ and $\boldsymbol{H}$?
- When (and how quickly) has the process converged?
    - Fixed number of iterations? Minimum change in error?

# Alternating least squares

- Without the non-negativity constraint, this is the standard least-squares:
  - $\boldsymbol{w}_i \leftarrow \text{argmin}_{\boldsymbol{w}} \, ||\boldsymbol{wH} - \boldsymbol{a}_i||_F$
  - we can update $\boldsymbol{W} \leftarrow \boldsymbol{AH}^+$ and $\boldsymbol{H} \leftarrow \boldsymbol{W}^+\boldsymbol{A}$
  - $\boldsymbol{X}^+$ is the pseudo-inverse of $\boldsymbol{X}$ which is LS-optimal
- The method is called **alternating least-squares** (ALS)
- This can introduce negative values

# Enforcing non-negativity in ALS

- Least-squares optimal update of $\boldsymbol{W}$ (or $\boldsymbol{H}$) with non-negativity constraints is convex optimization problem
  - In theory in P, in practice slow, but subject to much research
- Simple approach: truncate all negative values to 0
  - Update $\boldsymbol{W} \leftarrow [\boldsymbol{A}\boldsymbol{H}^{+}]_{+}$

# The NMF-ALS algorithm

1. $\boldsymbol{W} \leftarrow \mathrm{random}(n, k)$
2. **repeat**
    2.1. $\boldsymbol{H} \leftarrow [\boldsymbol{W}^{+}\boldsymbol{A}]_{+}$
    2.2. $\boldsymbol{W} \leftarrow [\boldsymbol{A}\boldsymbol{H}^{+}]_{+}$
3. **until** convergence

# When has there been enough convergence?

- When the error doesn't change too much
  - $||\boldsymbol{A} - \boldsymbol{W}^{(k)}\boldsymbol{H}^{(k)}||_F - ||\boldsymbol{A} - \boldsymbol{W}^{(k+1)}\boldsymbol{H}^{(k+1)}||_F \leq \epsilon$
- After some number of maximum iterations has been achieved
- Usually, whichever of these two happens first

# Gradient descent

- We can compute the gradient of the error function (with one factor matrix fixed)
  - $f(\boldsymbol{H}) = \frac{1}{2}\|\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}\|_F^2 = \frac{1}{2}\sum_i \|\boldsymbol{a}_i - \boldsymbol{W}\boldsymbol{h}_i\|_F^2$
  - $\nabla_{\boldsymbol{H}_{ij}} f(\boldsymbol{H}) = (\boldsymbol{W}^T\boldsymbol{A})_{ij} - (\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H})_{ij}$
- We can move slightly towards the negative gradient
  - How much is the step size and deciding it is a big problem

# The NMF gradient descent algorithm

1. $\boldsymbol{W} \leftarrow \text{random}(n, k)$
2. $\boldsymbol{H} \leftarrow \text{random}(k, m)$
3. **repeat**
    3.1. $\boldsymbol{H} \leftarrow \boldsymbol{H} - \varepsilon_{\boldsymbol{H}} \frac{\partial f}{\partial \boldsymbol{H}}$
    3.2. $\boldsymbol{W} \leftarrow \boldsymbol{W} - \varepsilon_{\boldsymbol{W}} \frac{\partial f}{\partial \boldsymbol{W}}$
4. **until** convergence

# Oblique Projected Landweber (OPL) for NMF

- OPL provides one way to select the step size
- With $\boldsymbol{H} \leftarrow \boldsymbol{H} - \varepsilon_{\boldsymbol{H}} \frac{\partial f}{\partial \boldsymbol{H}}$ updates, the convergence radius is $2/\lambda_{\max}(\boldsymbol{W}^T\boldsymbol{W})$, where $\lambda_{\max}$ is the largest eigenvalue
  - $\lambda_{\max} \leq \max(\texttt{rowSums}(\boldsymbol{W}^T\boldsymbol{W}))$
- We can set the learning rates to $1/\texttt{rowSums}(\boldsymbol{W}^T\boldsymbol{W})$ for a good convergence

# The OPL algorithm for updating $\boldsymbol{H}$

1. $\boldsymbol{\eta} \leftarrow \texttt{diag}(1 / \texttt{rowSums}(\boldsymbol{W}^T\boldsymbol{W}))$
2. **repeat**
   2.1. $\boldsymbol{G} \leftarrow \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H} - \boldsymbol{W}^T\boldsymbol{A}$
   2.2. $\boldsymbol{H} \leftarrow [\boldsymbol{H} - \boldsymbol{\eta}\boldsymbol{G}]_+$
3. **until** a stopping criterion is met

(small) number of iterations
OR
***H*** doesn't change much

# Interior Point Gradient (IPG) for NMF

- In OPL, we might (temporarily) have negative values in $\boldsymbol{W}$ or $\boldsymbol{H}$
- In **Interior Point Gradient** (IPG) algorithm, we set the step sizes so that we never update to negative

# The IPG algorithm for updating H

1. **repeat until** a stopping criterion is met
   1.1. $\boldsymbol{G} \leftarrow \boldsymbol{W}^T(\boldsymbol{WH} - \boldsymbol{A})$ Gradient
   1.2. $\boldsymbol{D} \leftarrow \boldsymbol{H} \,/\, (\boldsymbol{W}^T\boldsymbol{WH})$ Scaling
   1.3. $\boldsymbol{P} \leftarrow -\boldsymbol{D} * \boldsymbol{G}$ Update direction
   1.4. $\eta^* \leftarrow -\langle \mathrm{vec}(\boldsymbol{P}), \mathrm{vec}(\boldsymbol{G})\rangle / \langle \mathrm{vec}(\boldsymbol{WP}), \mathrm{vec}(\boldsymbol{WP})\rangle$ Best step size
   1.5. $\eta' \leftarrow \max\{\eta : \boldsymbol{H} + \eta\boldsymbol{P} \geq 0\}$ Positive step size
   1.6. $\eta \leftarrow \min\{\tau\eta', \eta^*\}$
   1.7. $\boldsymbol{H} \leftarrow \boldsymbol{H} + \eta\boldsymbol{P}$ Update

/ and * are element-wise

# Multiplicative updates

- The KKT conditions for $\boldsymbol{H}$ in NMF are
  - $\boldsymbol{H} \geq 0$; $\nabla_{\boldsymbol{H}} \|\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}\|^2/2 \geq 0$
  - $\boldsymbol{H} * \nabla_{\boldsymbol{H}} \|\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}\|^2/2 = 0$

* is element-wise product

- Substituting $\nabla_{\boldsymbol{H}} \|\boldsymbol{A} - \boldsymbol{W}\boldsymbol{H}\|^2/2 = \boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H} - \boldsymbol{W}^T\boldsymbol{A}$ one gets $\boldsymbol{H} * (\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}) = \boldsymbol{H} * (\boldsymbol{W}^T\boldsymbol{A})$
  - This gives us an update rule for $\boldsymbol{H}$

# The NMF multiplicative updates algorithm

1. $\boldsymbol{W} \leftarrow$ random$(n, k)$
2. $\boldsymbol{H} \leftarrow$ random$(k, m)$
3. **repeat**
    1. $h_{ij} \leftarrow h_{ij} \frac{(\boldsymbol{W}^T\boldsymbol{A})_{ij}}{(\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H})_{ij}+\varepsilon}$
    2. $w_{ij} \leftarrow w_{ij} \frac{(\boldsymbol{A}\boldsymbol{H}^T)_{ij}}{(\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T)_{ij}+\varepsilon}$
4. **until** convergence

# Notes on multiplicative updates

- Proposed by Lee & Seung (Nature, 1999)
- Equivalent to gradient descent with dynamic step size
- Zeros in initial solutions will never turn into non-zeros; non-zeros will never turn into zeros
  - Problems if the correct solution contains zeros

# Summary

- NMF can provide factorizations that are more interpretable than those given by SVD
- Harder to compute than SVD, but many different approaches
  - Or are they so different…
- In two weeks: Applications & alternations of NMF… *Stay tuned!*