The Why and How of Nonnegative Matrix Factorization

Johannes Friedrich

Nov 1, 2019

Flatiron-wide Algorithms and Mathematics (FWAM)

Outline

The Why – NMF Generates Sparse and Meaningful Features

The How – Some Algorithms

Take away messages

Nonnegative Matrix Factorization

Given a matrix $X \in \mathbb{R}_+^{p \times n}$ and a factorization rank $r \ll \min(p, n)$, find $W \in \mathbb{R}_+^{p \times r}$ and $H \in \mathbb{R}_+^{r \times n}$ such that

$$\min_{W \geq 0, H \geq 0} f(W, H) \quad \text{with}^1 \quad f = \|X - WH\|_F^2 = \sum_{i,j} (X - WH)_{ij}^2. \quad \text{(NMF)}$$

[1] or another β-divergence $D_\beta(X|Y) = \sum_{ij} \frac{1}{\beta(\beta-1)} \left(X_{ij}^\beta + (\beta - 1) Y_{ij}^\beta - \beta X_{ij} Y_{ij}^{\beta-1} \right)$

FLATIRON INSTITUTE

Nonnegative Matrix Factorization

Given a matrix $X \in \mathbb{R}_+^{p \times n}$ and a factorization rank $r \ll \min(p, n)$, find $W \in \mathbb{R}_+^{p \times r}$ and $H \in \mathbb{R}_+^{r \times n}$ such that

$$\min_{W \geq 0, H \geq 0} f(W, H) \quad \text{with}^1 \quad f = \|X - WH\|_F^2 = \sum_{i,j} (X - WH)_{ij}^2. \quad \text{(NMF)}$$

NMF is a linear dimensionality reduction technique for nonnegative data:

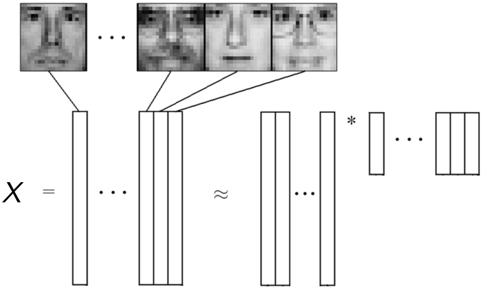
$$\underbrace{X(:, i)}_{\geq 0} \approx \sum_{k=1}^{r} \underbrace{W(:, k)}_{\geq 0} \underbrace{H(k, i)}_{\geq 0} \qquad \forall i.$$

Why nonnegativity?

---

[1] or another β-divergence $D_\beta(X|Y) = \sum_{ij} \frac{1}{\beta(\beta-1)} \left(X_{ij}^\beta + (\beta - 1) Y_{ij}^\beta - \beta X_{ij} Y_{ij}^{\beta-1} \right)$

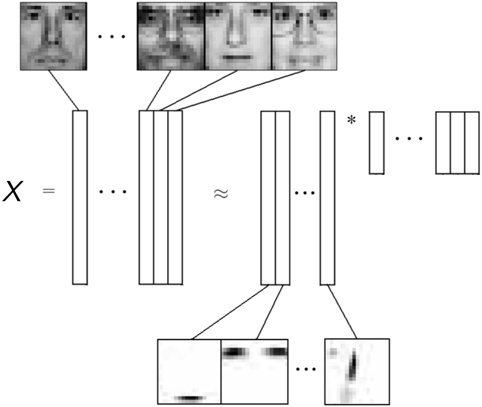
Application 1: Image Processing



Each column of X represents an image, i.e. a vector of intensities.
Each entry X_{ij} is the intensity of pixel i in image j.

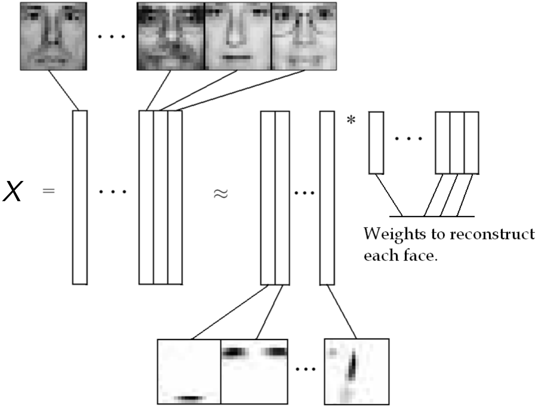
Application 1: Image Processing

$W \geq 0$ constraints the basis elements to be **nonnegative**.



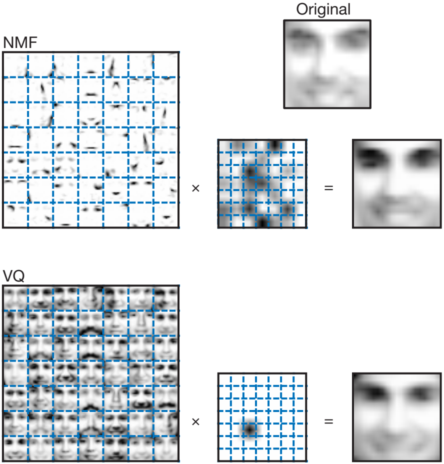
Application 1: Image Processing

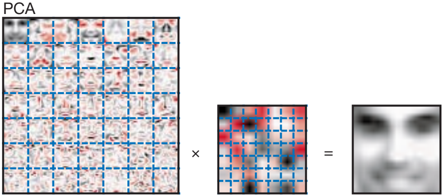
$W \geq 0$ constraints the basis elements to be **nonnegative**.
Moreover $H \geq 0$ imposes an **additive reconstruction**.



The basis elements **extract facial features** such as eyes, nose and lips.

Application 1: Image Processing





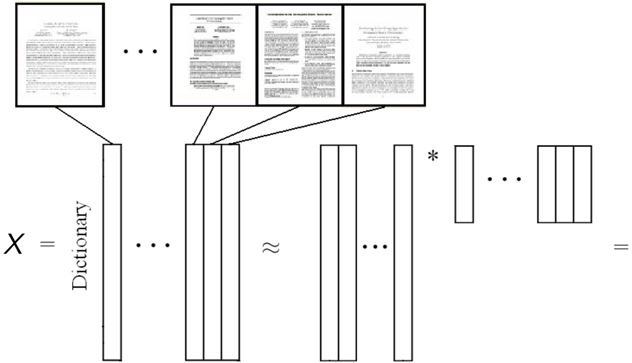
NMF learns parts-based representation

PCA and vector quantization (VQ) learn holistic representations

- PCA: ‘eigenfaces’
- VQ: prototypes

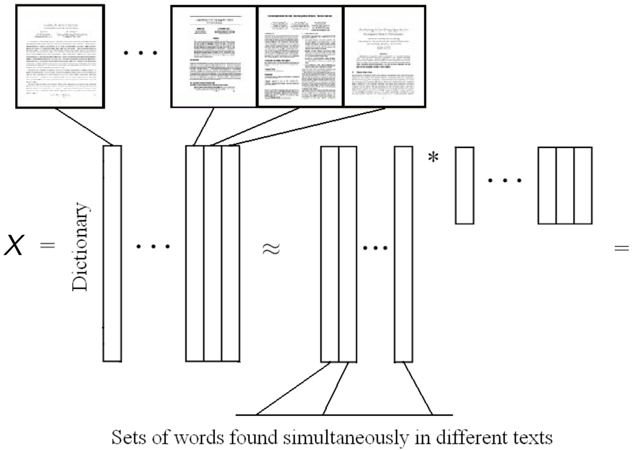
[Lee and Seung, 1999]

Application 2: Text Mining



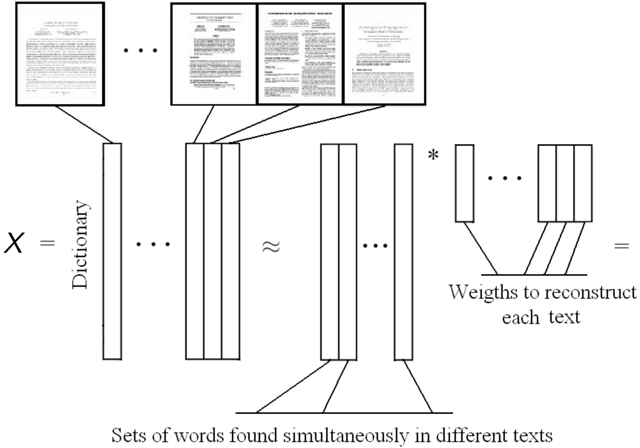
Each column of X represents a document, i.e. a vector of word counts.
Each entry X_{ij} is the number of times word i appears in document j.

Application 2: Text Mining



The basis elements allow to **recover the different topics**.

Application 2: Text Mining

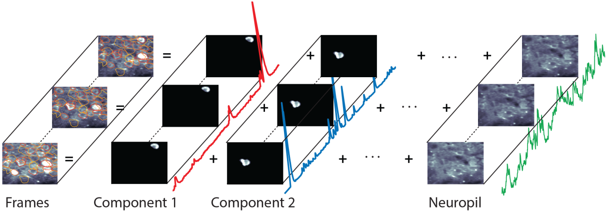


The basis elements allow to **recover the different topics**.
Weights allow to **assign each text to its corresponding topics**.

FLATIRON
INSTITUTE

Application 3: Calcium Imaging of Neuronal Populations

$X =$

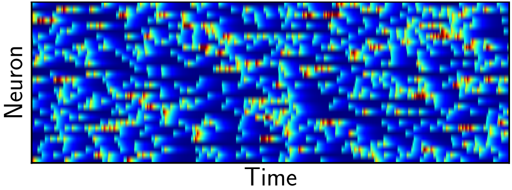


$+ E = WH + E$

$W \in \mathbb{R}_{+}^{p \times r}$ spatial matrix

$H \in \mathbb{R}_{+}^{r \times n}$ temporal matrix

$E \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}(\boldsymbol{\sigma}))$ noise



Basis elements extract **neural footprints**.

Weights extract (convolved) **neural activity**.

Outline

The Why – NMF Generates Sparse and Meaningful Features

The How – Some Algorithms

Take away messages

Standard NMF Framework: Block Coordinate Descent

$$\min_{W \geq 0, H \geq 0} \|X - WH\|_F^2. \tag{NMF}$$

- NMF is NP-hard [Vavasis, 2009].
- NMF is non-convex. However, it is bi-convex.

Standard NMF Framework: Block Coordinate Descent

$$\min_{W \geq 0, H \geq 0} \|X - WH\|_F^2. \qquad \text{(NMF)}$$

- NMF is NP-hard [Vavasis, 2009].
- NMF is non-convex. However, it is bi-convex.

Two-Block Coordinate Descent – Framework of most NMF Algs

Initialize (W, H). Then, alternatively update W and H:

$$\text{Update } W \approx \arg\min_{W \geq 0} \|X - WH\|_F^2. \qquad \text{(NNLS)}$$

$$\text{Update } H \approx \arg\min_{H \geq 0} \|X - WH\|_F^2. \qquad \text{(NNLS)}$$

Standard NMF Framework: Block Coordinate Descent

$$\min_{W \geq 0, H \geq 0} \|X - WH\|_F^2. \qquad \text{(NMF)}$$

- NMF is NP-hard [Vavasis, 2009].
- NMF is non-convex. However, it is bi-convex.

Two-Block Coordinate Descent – Framework of most NMF Algs

Initialize (W, H). Then, alternatively update W and H:

$$\text{Update } W \approx \arg\min_{W \geq 0} \|X - WH\|_F^2. \qquad \text{(NNLS)}$$

$$\text{Update } H \approx \arg\min_{H \geq 0} \|X - WH\|_F^2. \qquad \text{(NNLS)}$$

First-order optimality conditions for stationary points

$$\begin{aligned} W \geq 0, \quad \nabla_W f &= WHH^\top - XH^\top \geq 0, \quad W \circ \nabla_W f = 0, \\ H \geq 0, \quad \nabla_H f &= W^\top WH - W^\top X \geq 0, \quad H \circ \nabla_H f = 0, \end{aligned} \qquad \text{(KKT)}$$

Multiplicative Updates

Multiplicative Updates

$$W \leftarrow W \circ \frac{XH^\top}{WHH^\top}, \qquad H \leftarrow H \circ \frac{W^\top X}{W^\top WH} \tag{MU}$$

[Lee and Seung, 1999, Lee and Seung, 2001]

Multiplicative Updates

Multiplicative Updates

$$W \leftarrow W \circ \frac{XH^\top}{WHH^\top}, \qquad H \leftarrow H \circ \frac{W^\top X}{W^\top WH} \tag{MU}$$

- The objective $\|X - WH\|$ is non increasing under the update rules.
- Converge relatively slowly.

[Lee and Seung, 1999, Lee and Seung, 2001]

Multiplicative Updates

Multiplicative Updates

$$W \leftarrow W \circ \frac{XH^\top}{WHH^\top}, \qquad H \leftarrow H \circ \frac{W^\top X}{W^\top WH} \tag{MU}$$

- The objective $\|X - WH\|$ is non increasing under the update rules.
- Converge relatively slowly.
- Not guaranteed to converge to a stationary point, may get "stuck" at zero.
  Fix: reinitialize zero entries to a small positive constant when their partial derivatives become negative.

[Lee and Seung, 1999, Lee and Seung, 2001]

Hierarchical Alternating Least Squares

Exact coordinate descent method, updating one column of W at a time.

$W_{:,k} \leftarrow \arg\min_{W_{:,k}\geq 0} \|X - \sum_{j\neq k} W_{:,j} H_{j,:} - W_{:,k} H_{k,:}\|_F^2$

Hierarchical Alternating Least Squares

$$\begin{aligned} W_{:,k} &\leftarrow \left[W_{:,k} + \frac{(XH^\top)_{:,k} - (WHH^\top)_{:,k}}{(HH^\top)_{kk}} \right]_+ \\ H_{k,:} &\leftarrow \left[H_{k,:} + \frac{(W^\top X)_{k,:} - (W^\top WH)_{k,:}}{(W^\top W)_{kk}} \right]_+ \end{aligned} \tag{HALS}$$

[Cichocki and Phan, 2009]

Hierarchical Alternating Least Squares

Exact coordinate descent method, updating one column of W at a time.

$W_{:,k} \leftarrow \arg\min_{W_{:,k}\geq 0} \|X - \sum_{j\neq k} W_{:,j}H_{j,:} - W_{:,k}H_{k,:}\|_F^2$

Hierarchical Alternating Least Squares

$$\begin{aligned} W_{:,k} &\leftarrow \left[W_{:,k} + \frac{(XH^\top)_{:,k} - (WHH^\top)_{:,k}}{(HH^\top)_{kk}}\right]_+ \\ H_{k,:} &\leftarrow \left[H_{k,:} + \frac{(W^\top X)_{k,:} - (W^\top WH)_{k,:}}{(W^\top W)_{kk}}\right]_+ \end{aligned} \qquad \text{(HALS)}$$

- Be smart with the order of matrix-products: $WHH^\top = W(HH^\top)$.

[Cichocki and Phan, 2009]

Hierarchical Alternating Least Squares

Exact coordinate descent method, updating one column of W at a time.

$W_{:,k} \leftarrow \arg\min_{W_{:,k}\geq 0} \|X - \sum_{j\neq k} W_{:,j}H_{j,:} - W_{:,k}H_{k,:}\|_F^2$

Hierarchical Alternating Least Squares

$$\begin{aligned} W_{:,k} &\leftarrow \left[W_{:,k} + \frac{(XH^\top)_{:,k} - (WHH^\top)_{:,k}}{(HH^\top)_{kk}} \right]_+ \\ H_{k,:} &\leftarrow \left[H_{k,:} + \frac{(W^\top X)_{k,:} - (W^\top WH)_{k,:}}{(W^\top W)_{kk}} \right]_+ \end{aligned} \tag{HALS}$$

- Be smart with the order of matrix-products: $WHH^\top = W(HH^\top)$.
- Converges much faster than the MU.
- Update order $W_{:,1}, \ldots, W_{:,r}, H_{1,:}, \ldots, H_{r,:}$ is more efficient than $W_{:,1}, H_{1,:}, \ldots, W_{:,r}, H_{r,:}$ (sped up further by updating each factor several times).

[Cichocki and Phan, 2009]

Hierarchical Alternating Least Squares

Exact coordinate descent method, updating one column of W at a time.

$W_{:,k} \leftarrow \arg\min_{W_{:,k} \geq 0} \|X - \sum_{j \neq k} W_{:,j} H_{j,:} - W_{:,k} H_{k,:}\|_F^2$

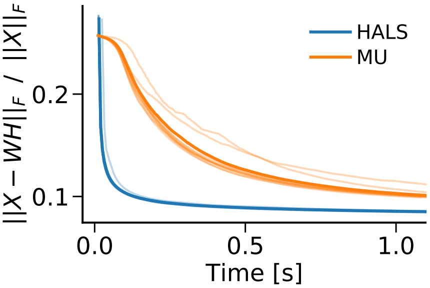
Hierarchical Alternating Least Squares

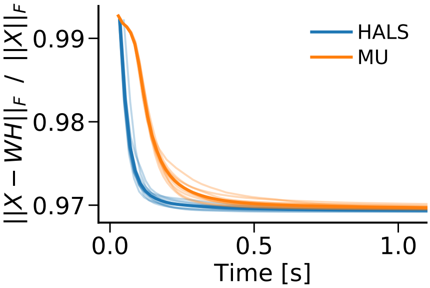
$$\begin{aligned} W_{:,k} &\leftarrow \left[W_{:,k} + \frac{(XH^\top)_{:,k} - (WHH^\top)_{:,k}}{(HH^\top)_{kk}} \right]_+ \\ H_{k,:} &\leftarrow \left[H_{k,:} + \frac{(W^\top X)_{k,:} - (W^\top WH)_{k,:}}{(W^\top W)_{kk}} \right]_+ \end{aligned} \qquad \text{(HALS)}$$

- Be smart with the order of matrix-products: $WHH^\top = W(HH^\top)$.
- Converges much faster than the MU.
- Update order $W_{:,1}, \ldots, W_{:,r}, H_{1,:}, \ldots, H_{r,:}$ is more efficient than $W_{:,1}, H_{1,:}, \ldots, W_{:,r}, H_{r,:}$ (sped up further by updating each factor several times).
- Guaranteed to converge to a stationary point (under mild assumptions).
- Be careful when initializing otherwise the algorithm could set some columns of W to zero initially.

[Cichocki and Phan, 2009]

Comparison





Left: CBCL Faces: $n = 2429, p = 361, r = 49$, dense
Right: 20 newsgroups: $n = 6019, p = 11314, r = 20$, sparse

Regularized NMF

$$\min_{W\geq 0, H\geq 0} \frac{1}{2}\|X - WH\|_F^2 + \frac{\alpha_W}{2}\|W\|_F^2 + \frac{\alpha_H}{2}\|H\|_F^2 + \beta_W \sum_{ij}|W_{ij}| + \beta_H \sum_{ij}|H_{ij}|$$

Hierarchical Alternating Least Squares

$$\begin{aligned} W_{:,k} &\leftarrow \left[\frac{W_{:,k}(HH^\top)_{kk} + (XH^\top)_{:,k} - (WHH^\top)_{:,k} - \beta_W \mathbf{1}}{(HH^\top)_{kk} + \alpha_W}\right]_+ \\ H_{k,:} &\leftarrow \left[\frac{H_{k,:}(W^\top W)_{kk} + (W^\top X)_{k,:} - (W^\top WH)_{k,:} - \beta_H \mathbf{1}}{(W^\top W)_{kk} + \alpha_H}\right]_+ \end{aligned} \qquad \text{(HALS)}$$

- L1 regularization corresponds to decrease of the numerator,
  $\Leftrightarrow$ decrease of each element of $XH^\top$ and $W^\top X$.
- L2 regularization corresponds to increase of the denominator,
  $\Leftrightarrow$ increase of the diagonal of $HH^\top$ and $W^\top W$.

FLATIRON INSTITUTE

Implementation issues

Initialization

- random; scale by $\alpha^* = \arg\min_\alpha \|X - \alpha WH\|_F = \frac{\langle XH^\top, W\rangle}{\langle W^\top W, HH^\top\rangle}$
- SVD: replace each rank-one factor in $\sum_{k=1}^{r} \mathbf{u}_k \mathbf{v}_k^\top$ with either $[\mathbf{u}_k]_+[\mathbf{v}_k^\top]_+$ or $[-\mathbf{u}_k]_+[-\mathbf{v}_k^\top]_+$, selecting the one with larger norm.
- use domain knowledge

FLATIRON INSTITUTE

Implementation issues

Initialization

- random; scale by $\alpha^* = \arg\min_\alpha \|X - \alpha WH\|_F = \frac{\langle XH^\top, W\rangle}{\langle W^\top W, HH^\top\rangle}$
- SVD: replace each rank-one factor in $\sum_{k=1}^{r} \mathbf{u}_k \mathbf{v}_k^\top$ with either $[\mathbf{u}_k]_+[\mathbf{v}_k^\top]_+$ or $[-\mathbf{u}_k]_+[-\mathbf{v}_k^\top]_+$, selecting the one with larger norm.
- use domain knowledge

Stopping Criterion

- $f(W^{(i-1)}, H^{(i-1)}) - f(W^{(i)}, H^{(i)}) < \epsilon$

  Difference may become small before a local minimum is achieved.
- Def. proj. grad.: $(\nabla^p_W f)_{ij} := \begin{cases} \min(0, (\nabla_W f)_{ij}) & \textit{if} \quad W_{ij} = 0 \\ (\nabla_W f)_{ij} & \text{otherwise} \end{cases}$

  KKT conditions $\Leftrightarrow \nabla^p_W f = 0$ and $\nabla^p_H f = 0$

  $\frac{\Delta(i)}{\Delta(0)} < \epsilon$ with $\Delta(i) := \sqrt{\|\nabla^p_W f^{(i)}\|_F^2 + \|\nabla^p_H f^{(i)}\|_F^2}$

FLATIRON INSTITUTE

Online NMF

$$W_{:,k} \leftarrow \left[W_{:,k} + \frac{(XH^\top)_{:,k} - (WHH^\top)_{:,k}}{(HH^\top)_{kk}} \right]_+, \quad H_{k,:} \leftarrow \left[H_{k,:} + \frac{(W^\top X)_{k,:} - (W^\top WH)_{k,:}}{(W^\top W)_{kk}} \right]_+$$

Keep track of sufficient statistics $A = XH^\top$, $B = HH^\top$

- Observe next data point $\mathbf{x} := X_{:,n+1}$

[Mairal et al., 2010]

Online NMF

$$W_{:,k} \leftarrow \left[W_{:,k} + \frac{(XH^\top)_{:,k} - (WHH^\top)_{:,k}}{(HH^\top)_{kk}} \right]_+, \quad H_{k,:} \leftarrow \left[H_{k,:} + \frac{(W^\top X)_{k,:} - (W^\top WH)_{k,:}}{(W^\top W)_{kk}} \right]_+$$

Keep track of sufficient statistics $A = XH^\top$, $B = HH^\top$

- Observe next data point $\mathbf{x} := X_{:,n+1}$
- Obtain $\mathbf{h} := H_{:,n+1}$ using current value of W (and warm starts)

  repeat until convergence:

  for $k = 1$ to r do: $\quad \mathbf{h}_k \leftarrow \left[\mathbf{h}_k + \frac{(W^\top \mathbf{x})_k - (W^\top W\mathbf{h})_k}{(W^\top W)_{kk}} \right]_+$

[Mairal et al., 2010]

Online NMF

$$W_{:,k} \leftarrow \left[W_{:,k} + \frac{(XH^\top)_{:,k} - (WHH^\top)_{:,k}}{(HH^\top)_{kk}} \right]_+, \quad H_{k,:} \leftarrow \left[H_{k,:} + \frac{(W^\top X)_{k,:} - (W^\top WH)_{k,:}}{(W^\top W)_{kk}} \right]_+$$

Keep track of sufficient statistics $A = XH^\top$, $B = HH^\top$

- Observe next data point $\mathbf{x} := X_{:,n+1}$
- Obtain $\mathbf{h} := H_{:,n+1}$ using current value of W (and warm starts)

  repeat until convergence:

  for $k = 1$ to r do: $\quad \mathbf{h}_k \leftarrow \left[\mathbf{h}_k + \frac{(W^\top \mathbf{x})_k - (W^\top W \mathbf{h})_k}{(W^\top W)_{kk}} \right]_+$
- Update sufficient statistics: $A \leftarrow A + \mathbf{x}\mathbf{h}^\top$, $B = B + \mathbf{h}\mathbf{h}^\top$

[Mairal et al., 2010]

Online NMF

$$W_{:,k} \leftarrow \left[W_{:,k} + \frac{(XH^\top)_{:,k} - (WHH^\top)_{:,k}}{(HH^\top)_{kk}} \right]_+, \quad H_{k,:} \leftarrow \left[H_{k,:} + \frac{(W^\top X)_{k,:} - (W^\top WH)_{k,:}}{(W^\top W)_{kk}} \right]_+$$

Keep track of sufficient statistics $A = XH^\top$, $B = HH^\top$

- Observe next data point $\mathbf{x} := X_{:,n+1}$
- Obtain $\mathbf{h} := H_{:,n+1}$ using current value of W (and warm starts)

  repeat until convergence:

  for $k = 1$ to r do: $\quad \mathbf{h}_k \leftarrow \left[\mathbf{h}_k + \frac{(W^\top \mathbf{x})_k - (W^\top W \mathbf{h})_k}{(W^\top W)_{kk}} \right]_+$
- Update sufficient statistics: $A \leftarrow A + \mathbf{x}\mathbf{h}^\top$, $B = B + \mathbf{h}\mathbf{h}^\top$
- Update basis elements W using sufficient statistics and warm starts

  repeat until convergence:

  for $k = 1$ to r do: $\quad W_{:,k} \leftarrow \left[W_{:,k} + \frac{A_{:,k} - WB_{:,k}}{B_{kk}} \right]_+$

[Mairal et al., 2010]

Online NMF

$$W_{:,k} \leftarrow \left[W_{:,k} + \frac{(XH^\top)_{:,k} - (WHH^\top)_{:,k}}{(HH^\top)_{kk}} \right]_+, \quad H_{k,:} \leftarrow \left[H_{k,:} + \frac{(W^\top X)_{k,:} - (W^\top WH)_{k,:}}{(W^\top W)_{kk}} \right]_+$$

Keep track of sufficient statistics $A = XH^\top$, $B = HH^\top$

- Observe next data point $\mathbf{x} := X_{:,n+1}$
- Obtain $\mathbf{h} := H_{:,n+1}$ using current value of W (and warm starts)

  repeat until convergence:

  for $k = 1$ to r do: $\quad \mathbf{h}_k \leftarrow \left[\mathbf{h}_k + \frac{(W^\top \mathbf{x})_k - (W^\top W \mathbf{h})_k}{(W^\top W)_{kk}} \right]_+$
- Update sufficient statistics: $A \leftarrow A + \mathbf{x}\mathbf{h}^\top$, $B = B + \mathbf{h}\mathbf{h}^\top$
- Update basis elements W using sufficient statistics and warm starts

  repeat until convergence:

  for $k = 1$ to r do: $\quad W_{:,k} \leftarrow \left[W_{:,k} + \frac{A_{:,k} - WB_{:,k}}{B_{kk}} \right]_+$
- Converges (almost surely) to a stationary point of the objective function.

[Mairal et al., 2010]

Outline

The Why – NMF Generates Sparse and Meaningful Features

The How – Some Algorithms

Take away messages

Take away messages

- NMF produces **interpretable, sparse, parts-based representations**.
- NMF is **difficult to solve** (NP-hard).
- Use **HALS**, which beats MU. (default in scikit-learn)
- Be aware of non-convexity; **initialize smartly**.
- Use online NMF formulation for large or streaming data.

References

Cichocki, A. and Phan, A.-H. (2009).
Fast local algorithms for large scale nonnegative matrix and tensor factorizations.
IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 92(3):708–721.

Gillis, N. (2014).
The why and how of nonnegative matrix factorization.
Regularization, Optimization, Kernels, and Support Vector Machines, 12(257):257–291.

Kim, J., He, Y., and Park, H. (2014).
Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework.
Journal of Global Optimization, 58(2):285–319.

Lee, D. D. and Seung, H. S. (1999).
Learning the parts of objects by non-negative matrix factorization.
Nature, 401(6755):788.

Lee, D. D. and Seung, H. S. (2001).
Algorithms for non-negative matrix factorization.
In *Advances in Neural Information Processing Systems*, pages 556–562.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010).
Online learning for matrix factorization and sparse coding.
Journal of Machine Learning Research, 11(Jan):19–60.