Log-determinant Non-negative Matrix Factorization
via Successive Trace Approximation
Optimizing an optimization algorithm

Andersen Ang

Mathématique et recherche opérationnelle
UMONS, Belgium

Email: manshun.ang@umons.ac.be
Homepage: angms.science

File created: November 30, 2017
Last update: May 18, 2018

Joint work with my supervisor: Nicolas Gillis (UMONS, Belgium)
Overview

1 Background and the research problem
   ▶ Non-negative Matrix Factorization
   ▶ Separable NMF
   ▶ Why Separable NMF is not enough
   ▶ The minimum volume criterion

2 Minimum volume log-determinant (logdet) NMF
   ▶ The logdet regularization
   ▶ A logdet inequality
   ▶ An upper bound of logdet

3 Solving logdet NMF - Successive Trace Approximation
   ▶ STA algorithm
   ▶ The NNQPs
   ▶ Optimizing STA

4 Experiments

5 Discussions, extensions and directions
   ▶ Coordinate acceleration by randomization
   ▶ On robustness and robust formulations
      ★ De-noising norm
      ★ Iterative Reweighted Least Squares
   ▶ Theoretical convergences
   ▶ Comparisons with benchmark algorithms (skipped here)
   ▶ On selecting the regularization parameter $\lambda$
   ▶ On automatic detection of $r$
   ▶ Further acceleration with weighted column fitting norms
What is Non-negative Matrix Factorization (NMF)?

Given $X \in \mathbb{R}^{m \times n}$ and integer $r$, find matrices $W \in \mathbb{R}^{m \times r}, H \in \mathbb{R}^{r \times n}$ s.t.

$$X = WH.$$  

1. This is called: exact NMF and it is NP-hard (Vavasis2007).
What is Non-negative Matrix Factorization (NMF) ?

Given \( X \in \mathbb{R}^{m \times n} \) and integer \( r \), find matrices \( W \in \mathbb{R}^{m \times r}, H \in \mathbb{R}^{r \times n} \) s.t.

\[
X = WH.
\]

1. This is called : exact NMF and it is \textbf{NP-hard} (Vavasis2007).
2. We consider
   - low rank/complexity model \( 1 \leq r \leq \min\{m, n\} \).
   - approximate NMF

\[
[W, H] = \underset{W \geq 0, H \geq 0}{\text{arg min}} \quad \frac{1}{2} \|X - WH\|_F^2.
\]
What is Non-negative Matrix Factorization (NMF) ?

Given $X \in \mathbb{R}^{m \times n}$ and integer $r$, find matrices $W \in \mathbb{R}_+^{m \times r}$, $H \in \mathbb{R}_+^{r \times n}$ s.t.

$$X = WH.$$ 

1. This is called : exact NMF and it is NP-hard (Vavasis2007).

2. We consider
   - low rank/complexity model $1 \leq r \leq \min\{m, n\}$.
   - approximate NMF

$$[W, H] = \arg\min_{W \geq 0, H \geq 0} \frac{1}{2} \|X - WH\|_F^2.$$ 

3. Such minimization problem is
   - also NP-hard
   - a non-convex problem
   - an ill-posed problem
What is Non-negative Matrix Factorization (NMF)?

Given $X \in \mathbb{R}^{m \times n}$ and integer $r$, find matrices $W \in \mathbb{R}^{m \times r}$, $H \in \mathbb{R}^{r \times n}$ s.t.

$$X = WH.$$ 

1. This is called: exact NMF and it is NP-hard (Vavasis2007).

2. We consider
   - low rank/complexity model $1 \leq r \leq \min\{m, n\}$.
   - approximate NMF
      $$[W, H] = \arg \min_{W \geq 0, H \geq 0} \frac{1}{2} \|X - WH\|_F^2.$$ 

3. Such minimization problem is
   - also NP-hard
   - a non-convex problem
   - an ill-posed problem

4. Assumptions (1) $W$, $H$ full rank, (2) $r$ is known.

5. Notation note: we use $WH$ instead of $WH^T$. 

6 / 95
Separable NMF

Given $X \in \mathbb{R}^{m \times n}$ and integer $r$, find matrices $W \in \mathbb{R}^{m \times r}$, $H \in \mathbb{R}^{r \times n}$ s.t.

$$X = WH \quad \text{and} \quad W = X(:, \mathcal{K}) \quad \text{(separability)}$$

1. Separable NMF = NMF + additional condition $W = X(:, \mathcal{K})$
   - Meaning: $W$ is some columns of $X$.
   - $\mathcal{K}$: column index set
   - $|\mathcal{K}| = r$

Not NP-hard anymore (by Donoho, Arora et al., etc.)

Existing algorithms that solve Separable NMF:
- XRAY (Kumar, 2013)
- SPA, SNPA (Gillis, 2013)
Separable NMF

Given \( X \in \mathbb{R}^{m \times n} \) and integer \( r \), find matrices \( W \in \mathbb{R}^{m \times r}, H \in \mathbb{R}^{r \times n} \) s.t.

\[
X = WH \quad \text{and} \quad W = X(:, K)
\]

1. **Separable NMF = NMF + additional condition** \( W = X(:, K) \)
   - Meaning: \( W \) is some columns of \( X \).
   - \( K \): column index set
   - \(|K| = r\)

2. Not NP-hard anymore (by Donohol, Arora et al., etc.)

3. Existing algorithms that solve Separable NMF:
   - XRAY (Kumar, 2013)
   - SPA, SNPA (Gillis, 2013)
Illustrative examples

Figure: NMF, $m = 5, n = 10, r = 3$.

Figure: Separable NMF, $\mathcal{K} = \{8, 1, 3\}$. H has some special columns: only one '1', and other elements are 0.

"Pure pixels" assumption: matrix $H = [I_r H']\Pi$.

Related terms: self-expressive, self-dictionary
Motivation — why study NMF

- In hyperspectral imaging application, $X = \text{image}$.
- $W = \text{absorption behaviour of materials: non-negative spectrum}$.
- $r = \#\text{ fundamental materials (e.g. rock, vegetation, water)}$.
- $H = \text{abundance of materials: non-negative and sum-to-1}$.

**Figure:** Hypersectral images decomposition. Figure copied shamelessly from N. Gillis.

Interpretation in short:

$X = \text{data, } W = \text{basis, } r = \#\text{basis, } H = \text{membership of data w.r.t. basis}$

NMF has many other signal processing applications.

NMF vs SVD: SVD has lower fitting error (in fact SVD achieve the optimal fitting), but basis of SVD are not interpretable. Separable NMF basis comes form data, interpretable!
Geometry of Separable NMF: normalizing columns of $H$.

$H$ tells the membership of data points in $X$ w.r.t basis $W$.

Column form expression of $X = WH$ is

$$X(:, j) = WH(:, j).$$

**Example.** $m = 5, n = 7, r = 2$
**Geometry of Separable NMF**: normalizing columns of $H$

$H$ tells the membership of data points in $X$ w.r.t basis $W$.

Column form expression of $X = WH$ is

$$X(:, j) = WH(:, j).$$

- Consider $j = 3$
\( H \) tells the membership of data points in \( X \) w.r.t basis \( W \).

Column form expression of \( X = WH \) is

\[
X(:, j) = WH(:, j).
\]

- A linear combination!
\( \mathbf{H} \) tells the membership of data points in \( \mathbf{X} \) w.r.t basis \( \mathbf{W} \).

Column form expression of \( \mathbf{X} = \mathbf{WH} \) is

\[
\mathbf{X}(\::, \ j) = \mathbf{WH}(\::, \ j).
\]

- Nonnegativity : \( H_{ij} \geq 0 \) so it is in fact conical combination.
H tells the membership of data points in X w.r.t basis W.

Column form expression of $X = WH$ is

$$X(:, j) = WH(:, j).$$

- If $H(:, j)$ are normalized $\implies 0 \leq H_{ij} \leq 1 \implies$ convex combination
Geometry of Separable NMF: normalizing columns of $H$

$H$ tells the membership of data points in $X$ w.r.t basis $W$.

Column form expression of $X = WH$ is

$$X(:, j) = WH(:, j).$$

- If columns of $H$ are normalized: it means $W$ is forming a *convex hull* encapsulating the data columns of $X$. 

![Diagram showing data points, generators/endmembers, convex hull, and outlier/Noisy Points]
Geometry of Separable NMF: normalizing columns of $H$

$H$ tells the membership of data points in $X$ w.r.t basis $W$.

Column form expression of $X = WH$ is

$$X(:, j) = WH(:, j).$$

- Algebraically, column normalization of $H$ removes the scaling ambiguity of factorization, prevents huge $H$ and super small $W$ as

$$W_1H_1 = W_1\Lambda \Pi \Pi^{-1} \Lambda^{-1}H_1$$

No normalization: convex hull $\rightarrow$ conical hull $\implies$ scaling ambiguity!
What has been done in literature

The problem statement: given the data points that has pure pixel (i.e. data points are distributed in the entire data subspace).
Goal: find the vertices = (1) find $r$ (number of vertices), (2) locate them.

Figure: A 2D PCA projection of a high dimensional data, showing the data points (black dots) encapsulated inside convex hull spanned by the generator (vertices).

This problem ⊆ Blind source identification with even distributed data

Existing methods: XRAY, SPA, SNPA ...
This work: what if pure pixel ($H_{ij} = 1$) is hidden in data?

The problem statement: given the data points that are at least $1 - \theta$ away from the vertices (i.e. the pure pixel are hidden, data points are not distributed in the entire data subspace).

Goal – find the vertices: (1) find $r$ (#vertices), (2) locate them.

Figure: A 2D PCA projection of a high dimensional data, showing the data points encapsulated inside convex hull spanned by the generator vertices.
This work: what if pure pixel \( (H_{ij} = 1) \) is hidden in data?

The problem statement: given the data points that are at least \( 1 - \theta \) away from the vertices (i.e. the pure pixel are hidden, data points are not distributed in the entire data subspace).

Goal – find the vertices: (1) find \( r \) (\#vertices), (2) locate them.

Figure: A 2D PCA projection of a high dimensional data, showing the data points encapsulated inside convex hull spanned by the generator vertices.

- \( \theta \): ”purity” of the data, \( \theta \in [0 \ 1] \).
- \( \theta = 1 \): Separable NMF.
- Problem (1) is not considered here: \( r = \# \) vertices (red dots) is assume known.
Existing algorithms aimed at solving the Separable NMF ($\theta = 1$) work poorly on the problems with $\theta < 1$.

Figure: Results (2d PCA projection) of SNPA with decreasing $\theta$. The dimensions are $(m, n, r) = (8, 1000, 3)$. 
Why separable NMF is not enough ... (2/2)

Due to the nature of high dimensional geometry, when \((m, r)\) increase, the data points are getting more and more concentrated around the annulus of the origin and thus they are **not distributed in entire data subspace**. Making approaches that use \(L_2\) norm of data points (such as SNPA) less workable.

---

**Figure:** Results (2d PCA proj.) of SNPA with increasing \((m, r)\). Red dots: ground truth vertices. Blue dots: estimated vertices. The dimensions are \((n, \theta) = (1000, 0.999)\).
The research problem: find the vertices from the data without the pure pixel \(H_{ij} = 1\)

- An idea from 1994\(^1\): fit a low rank convex hull with minimum volume.


The research problem: find the vertices from the data without the pure pixel \( (\mathbf{H}_{ij} = 1) \)

- An idea from 1994\(^1\): fit a low rank convex hull with minimum volume.
- Theoretical result in 2015\(^2\): such hull is identifiable if the data points are *well spreaded* (the underlying \( \theta \) is not too small)
  \[ \implies \text{volume} \text{ regularized NMF}. \]

---


The research problem: find the vertices from the data without the pure pixel \((H_{ij} = 1)\)

- An idea from 1994\(^1\): fit a low rank convex hull with minimum volume.
- Theoretical result in 2015\(^2\): such hull is identifiable if the data points are well spreaded (the underlying \(\theta\) is not too small) \(\Rightarrow\) volume regularized NMF.
- Volume is related to determinant \(\Rightarrow\) \(\det W\) regularization

---


The research problem: find the vertices from the data without the pure pixel \( (H_{ij} = 1) \)

- An idea from 1994\(^1\): fit a low rank convex hull with minimum volume.
- Theoretical result in 2015\(^2\): such hull is identifiable if the data points are *well spreaded* (the underlying \( \theta \) is not too small)
  \( \implies \text{volume regularized NMF} \).
- Volume is related to determinant \( \implies \det W \) regularization
- \( \det W \) only works for square \( W \) \( \implies \) consider \( \det W^T W \) or \( \log \det W^T W \)

---


The research problem: find the vertices from the data without the pure pixel \((H_{ij} = 1)\)

- An idea from 1994\(^1\): fit a low rank convex hull with minimum volume.
- Theoretical result in 2015\(^2\): such hull is identifiable if the data points are well spreaded (the underlying \(\theta\) is not too small)

\[
\Rightarrow \text{volume regularized NMF.}
\]

- Volume is related to determinant \(\Rightarrow \det W\) regularization
- \(\det W\) only works for square \(W\) \(\Rightarrow\) consider \(\det W^T W\) or \(\log \det W^T W\)

\[
\begin{align*}
\text{detNMF:} & \quad \min_{W \geq 0, \ H \geq 0} \frac{1}{2} \|X - WH\|_F^2 + \frac{\lambda}{2} \det(W^T W), \\
\text{data fitting term } & \quad \mathcal{F} \\
\text{volume regularizer } & \quad \mathcal{G}
\end{align*}
\]

\[
\begin{align*}
\text{logdetNMF:} & \quad \min_{W \geq 0, \ H \geq 0} \frac{1}{2} \|X - WH\|_F^2 + \frac{\lambda}{2} \log \det(W^T W + \delta I), \\
\text{data fitting term } & \quad \mathcal{F} \\
\text{volume regularizer } & \quad \mathcal{G}
\end{align*}
\]


Given \( X \in \mathbb{R}^{m \times n} \), find matrices \( W \in \mathbb{R}^{m \times r}_+ \), \( H \in \mathbb{R}^{r \times n}_+ \) by solving

\[
\min_{\begin{subarray}{l}
W \geq 0 \\
H \geq 0 \\
1_r^\top H \leq 1_n
\end{subarray}} \left\{ \frac{1}{2} \| X - WH \|_F^2 + \frac{\lambda}{2} \log \det(W^\top W + \delta I_r) \right\}.
\]

- \( \lambda > 0 \) : tuning parameters (regularization parameter).

- \( r \in \mathbb{N}_+ \) assumed known, \( W, H \) assumed full rank.)
Given $X \in \mathbb{R}^{m \times n}$, find matrices $W \in \mathbb{R}_{+}^{m \times r}, H \in \mathbb{R}_{+}^{r \times n}$ by solving

$$\min_{\substack{W \geq 0 \\ H \geq 0 \\ 1_{r}^{\top} H \leq 1_{n}}} \frac{1}{2} \|X - WH\|_{F}^{2} + \frac{\lambda}{2} \log \det(W^{\top}W + \delta I_{r}) + \text{volume regularizer}.$$ 

- $\lambda > 0$: tuning parameters (regularization parameter).
- $\delta$: fix small positive constant (e.g. 1).
- Why $\delta I_{r}$: to bound $\log \det$ (otherwise $\lim_{\|W\| \to 0} \log \det W^{\top}W \to -\infty$).
- $(r \in \mathbb{N}_{+}$ assumed known, $W, H$ assumed full rank).
Two Block Coordinate Descent solution framework

The optimization problem: given \( X \in \mathbb{R}^{m \times n} \) and \( r \geq 1 \)

\[
\min_{W \geq 0, \ H \geq 0, \ 1_r^\top H \leq 1_n} \|X - WH\|_F^2 + \lambda \log \det(W^\top W + \delta I_r),
\]

1: INPUT: \( X \in \mathbb{R}^{m \times n}, \ r \in \mathbb{N}_+ \) and \( \lambda \geq 0 \)
2: OUTPUT: \( W \in \mathbb{R}^{m \times r}_+ \) and \( H \in \mathbb{R}^{r \times n}_+ \)
3: INITIALIZATION: \( W \in \mathbb{R}^{m \times r}_+ \) and \( H \in \mathbb{R}^{r \times n}_+ \)
4: for \( k = 1 \) to itermax do
5: \( W \leftarrow \arg \min_{W \geq 0} \|X - WH\|_F^2 + \lambda \log \det(W^\top W + \delta I_r). \)
6: \( H \leftarrow \arg \min_{H \geq 0, 1_r^\top H \leq 1_n} \|X - WH\|_F^2 + \lambda \log \det(W^\top W + \delta I_r). \)
7: end for

From now on, line 1-3 will be skipped (for space)
Subproblems lines 5-6 can be solve by projected gradient.
On solving $H$ and $W$

To solve

$$H \leftarrow \arg \min_{H \geq 0, 1_r^T H \leq 1_n} \|X - WH\|_F^2 + \lambda \log \det(W^T W + \delta I_r),$$

the FGM (Fast gradient method on constrained least square on unit simplex) from N. Gillis\(^\dagger\) will be used:

\[\begin{align*}
1: & \text{ for } k = 1 \text{ to itermax do} \\
2: & \quad W \leftarrow \arg \min_{W \geq 0} \|X - WH\|_F^2 + \lambda \log \det(W^T W + \delta I_r). \\
3: & \quad \text{Update } H \text{ using FGM}^{\dagger} \text{ with } \{X, W, H\}. \\
4: & \text{ end for}
\]\n
On solving $H$ and $W$

So the key problem is to solve for $W$.

- Data fitting part $\|X - WH\|_F^2$ is easy to handle.
- The regularizer $\log \det(\mathbf{W}^\top \mathbf{W} + \delta \mathbf{I}_r)$ is problematic: non-convex, column-coupled, non-proximable.
- *Don't forget we can always solve this problem just by vanilla projected gradient, but such approach does not utilize the structure of the problem — not good!*

---

1: \textbf{for} $k = 1$ to $\text{itermax}$ \textbf{do}
2: \hspace{1em} $\mathbf{W} \leftarrow \arg \min_{\mathbf{W} \geq 0} \|X - \mathbf{WH}\|_F^2 + \lambda \log \det(\mathbf{W}^\top \mathbf{W} + \delta \mathbf{I}_r)$.
3: \hspace{1em} Update $H$ using FGM$^\dagger$ with $\{X, W, H\}$.
4: \hspace{1em} \textbf{end for}

Previous lines of attack on $\log \det$

Previous lines of attack (fails or no result):

- Proximal operator on $+ \log \det \mathbf{W}^\top \mathbf{W}$
- Hadamard’s inequality: $|\det(\mathbf{A})| < \prod_i \|a_i\|_2^2$
  (Upper bound of volume spanned by $\mathbf{A} = [a_1 \ a_2 \ ...]$)
- exp-trace-log equation: $\det(\mathbf{I}_r + \mathbf{A}) = \exp \tr \log(\mathbf{I}_r + \mathbf{A})$
- Approximating the determinant 1. Diagonal Approximations
- Approximating the determinant 2. Eigenspectrum approximations

$$
\det(\mathbf{I}_r + \delta \mathbf{A}) = \det(\mathbf{I}_r + \delta \mathbf{P}^{-1} \mathbf{J} \mathbf{P}) \\
= \det(\mathbf{P}^{-1} (\mathbf{I} + \delta \mathbf{J}) \mathbf{P}) \\
= \det(\mathbf{P}^{-1} \mathbf{P} (\mathbf{I} + \delta \mathbf{J})) \\
= \prod_i (1 + \delta J_{ii}) \\
= 1 + \delta \sum_i J_{ii} + \delta^2 \sum_{i,j,j \neq i} J_{ii} J_{jj} + ...
$$

- A bound from telecom research:
  $$\log \det(\mathbf{W}^\top \mathbf{W} + \delta \mathbf{I}_r) \leq \tr(\mathbf{D}^\text{Taylor} \mathbf{A}^\top \mathbf{A}) - \log \det(\mathbf{D}^\text{Taylor}) - r$$
  where $\mathbf{D}^\text{Taylor} = (\mathbf{W}^{-1}_- \mathbf{W}_-^\top + \delta \mathbf{I}_r)^{-1}$, this is in fact the first order Taylor convex upper bound of the function $\log \det(\mathbf{X})$. Reference includes: S. Christensen et al., ”Weighted Sum-Rate Maximization using Weighted MMSE for MIMO-BC Beamforming Design”, IEEE Trans. Wireless Com., pp. 4792-4799, vol. 7, issue 12, 2008
The key inequality: logdet-trace inequality

Given a positive definite matrix $A \in \mathbb{R}^{r \times r}$, we have

$$\text{tr}(I_r - A^{-1}) \leq \log \det A \leq \text{tr}(A - I_r)$$

Sowe now have an upper bound: put $A = W^\top W + \delta I_r$

$$\log \det(W^t W + \delta I_r) \leq \text{tr}(W^\top W + (\delta - 1)I_r)$$

$$\|X - WH\|_F^2 + \lambda \log \det(W^\top W + \delta I_r) \leq \|X - WH\|_F^2 + \lambda \text{tr}(W^\top W + (\delta - 1)I_r)$$

- $\log \det(W^\top W + \delta I_r)$ is not convex w.r.t. $W$ but the trace is.
The key inequality: logdet-trace inequality

Given a positive definite matrix $A \in \mathbb{R}^{r \times r}$, we have

$$\text{tr}(I_r - A^{-1}) \leq \log \det A \leq \text{tr}(A - I_r)$$

So we now have an upper bound: put $A = W^\top W + \delta I_r$

$$\log \det(W^t W + \delta I_r) \leq \text{tr}(W^\top W + (\delta - 1)I_r)$$
$$\|X - WH\|_F^2 + \lambda \log \det(W^\top W + \delta I_r) \leq \|X - WH\|_F^2 + \lambda \text{tr}(W^\top W + (\delta - 1)I_r)$$

- $\log \det(W^\top W + \delta I_r)$ is not convex w.r.t. $W$ but the trace is.
- Algorithm that minimizes this upper bound:

```plaintext
1: for $k = 1$ to itermax do
2:   $W \leftarrow \arg \min_{W \geq 0} \|X - WH\|_F^2 + \lambda \text{tr}(W^\top W + (\delta - 1)I_r)$.
4: end for
```
The key inequality: logdet-trace inequality

Given a positive definite matrix $A \in \mathbb{R}^{r \times r}$, we have

$$\text{tr}(I_r - A^{-1}) \leq \log \det A \leq \text{tr}(A - I_r)$$

Sowe now have an upper bound: put $A = W^\top W + \delta I_r$

$$\log \det(W^t W + \delta I_r) \leq \text{tr}(W^\top W + (\delta - 1)I_r)$$
$$\|X - WH\|_F^2 + \lambda \log \det(W^\top W + \delta I_r) \leq \|X - WH\|_F^2 + \lambda \text{tr}(W^\top W + (\delta - 1)I_r)$$

- $\log \det(W^\top W + \delta I_r)$ is not convex w.r.t. $W$ but the trace is.
- Algorithm that minimizes this upper bound:

1. **for** $k = 1$ to $\text{itermax}$ **do**
2. $W \leftarrow \arg \min_{W \geq 0} \|X - WH\|_F^2 + \lambda \text{tr}(W^\top W + (\delta - 1)I_r)$.
4. **end for**

- Don’t stop here, it can be better !!
Given a positive definite matrix $A \in \mathbb{R}^{r \times r}$, we have

$$\log \det A \leq \text{tr}(A - I_r).$$
A closer look on the logdet-trace inequality

- Given a positive definite matrix $A \in \mathbb{R}^{r \times r}$, we have
  \[ \log \det A \leq \text{tr}(A - I_r). \]

- Let $\mu$ denotes eigenvalues, we have
  \[ \det A = \prod_{i} \mu_i \quad \text{and} \quad \text{tr} A = \sum_{i} \mu_i. \]

- \[ \log \det A \leq \text{tr}(A - I_r) \iff \sum_{i} \log \mu_i \leq \sum_{i} (\mu_i - 1). \]
A closer look on the logdet-trace inequality

- Given a positive definite matrix $A \in \mathbb{R}^{r \times r}$, we have

  $$\log \det A \leq \text{tr}(A - I_r).$$

- Let $\mu$ denotes eigenvalues, we have

  $$\det A = \prod_i \mu_i \quad \text{and} \quad \text{tr} A = \sum_i \mu_i.$$

- $\Rightarrow$

  $$\log \det A \leq \text{tr}(A - I_r) \iff \sum_i \log \mu_i \leq \sum_i (\mu_i - 1).$$

- $\log \mu_i$ means matrix $A$ has to be positive definite ($\mu_i > 0 \forall i$), which is satisfied for $A = W^T W + \delta I_r$. 

\[39 / 95\]
A closer look on the logdet-trace inequality

- Given a positive definite matrix $A \in \mathbb{R}^{r \times r}$, we have

\[ \log \det A \leq \text{tr}(A - I_r). \]

- Let $\mu$ denotes eigenvalues, we have

\[ \det A = \prod_{i} \mu_i \quad \text{and} \quad \text{tr} A = \sum_{i} \mu_i. \]

\[ \implies \log \det A \leq \text{tr}(A - I_r) \iff \sum_{i} \log \mu_i \leq \sum_{i} (\mu_i - 1). \]

- $\log \mu_i$ means matrix $A$ has to be positive definite ($\mu_i > 0 \ \forall \ i$), which is satisfied for $A = W^T W + \delta I_r$.

- $\sum_{i} \log \mu_i \leq \sum_{i} (\mu_i - 1) \iff \log \mu_i \leq \mu_i - 1 \ \forall \ i$. We can focus on the inequality $\log \mu_i \leq \mu_i - 1$ with $\mu_i \geq 0$.
On $\log x \leq x - 1$, $x \geq 0$

- $\log x$ is concave.
- $x - 1$ is the first order Taylor approximation of $\log x$ at $x = 1$.
- $x - 1$ is the only **convex-tight** upper bound of $\log x$.†
- **Tight**: $x - 1$ touch $\log x$ at the point $x = 1$.

† Higher order Taylor approximation of $\log x$ is tight, more accurate but not convex.
On $\log x \leq x - 1$, $x \geq 0$

- $\log x$ is concave.
- $x - 1$ is the first order Taylor approximation of $\log x$ at $x = 1$.
- $x - 1$ is the only **convex-tight** upper bound of $\log x$.†
- **Tight**: $x - 1$ touch $\log x$ at the point $x = 1$.
- Generalize to point $x_0$: $\log x \leq g(x|x_0) = a_1(x_0)x + a_0(x_0)$ is

$$\log x \leq \frac{1}{x_0}x + \log x_0 - 1.$$ 

† Higher order Taylor approximation of $\log x$ is tight, more accurate but not convex.
A parametric trace upper bound for $\log \det A$

\[
\log \det A = \sum \log \mu_i \\
\leq \sum \frac{1}{\mu_i} \mu_i + \log \mu_i^{-} - 1 \\
\leq \sum \frac{1}{\mu_{\min}} \mu_i + \log \mu_i^{-} - 1 \\
= \text{tr}(D^1 A + D^0)
\]

$D^1 = \frac{1}{\mu_{\min}} I_r$, $D^0 = \text{Diag}(\log \mu_i^{-} - 1)$, $\mu_i^{-}$ is $\mu_i$ of the previous step

Put $A = W^\top W + \delta I_r$, we have

\[
\log \det(W^\top W + \delta I_r) \leq \text{tr}(D^1 W^\top W + \delta D^1 + D^0) \\
(\text{ignore constants}) = \text{tr} D^1 W^\top W
\]
A note on weighted sum of eigenvalues and trace

Consider the matrix $A$ has eigen-decomposition as $A = V \Lambda V^T$. Let the weighting $\frac{1}{\mu_i}$ be $a_i$, then $\sum_i \frac{1}{\mu_i} \mu_i = \sum_i a_i \mu_i$ and

$$\sum_i a_i \mu_i = \text{tr} \begin{bmatrix} a_1 \mu_1 & a_2 \mu_2 & \cdots \end{bmatrix} = \text{tr} \begin{bmatrix} a_1 & a_2 & \cdots \\ \mu_1 & \mu_2 & \cdots \end{bmatrix}$$

$$= \text{tr} D^a V^T AV$$

$\neq \text{tr} D^a A$
The original function \( F(W) = \log \det(W^\top W + \delta I_r) \) is upper bounded by :

- **Eigen bound** \((B_1)\) : \( \text{tr} \, D^1 W^\top W + \text{constants} \).
- **Taylor bound** \((B_2)\) : \( \text{tr} \, D^{Taylor} W^\top W + \text{constants} \).

- Constants \( D^1, D^0 \) are defined as before, and constant \( D^{Taylor} = (W_{-1}^\top W_{-1} + \delta I_r)^{-1} \).

- Both bounds are trace functional with an **relaxation gap** :
  - \((B_1)\) has eigen gap \( \mu_i \geq \mu_{\text{min}} \)
  - \((B_2)\) has convexification gap
  - \( D^1 \) is **diagonal** but \( D^{Taylor} \) is not (it is dense) \( \Rightarrow \) column-wise decomposition is possible
Algorithm 1 Successive Trace Approximation

1: INPUT: $X \in \mathbb{R}^{m \times n}$, $r \in \mathbb{N}_+$, $\lambda > 0$, $\delta > 0$.
2: OUTPUT: $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$.
3: INITIALIZATION: $W \in \mathbb{R}_+^{m \times r}$, $H \in \mathbb{R}_+^{r \times n}$, $D^1 = I_r$
4: for $K = 1$ to itermax do
5:    for $k = 1$ to itermax do
6:        $W \leftarrow \arg \min_{W \geq 0} \|X - WH\|_F^2 + \lambda \text{tr } D^1 W^\top W$.
8: end for
9: $\mu_i \leftarrow \text{svd}(W^\top W + \delta I_r)$, $D^1 = \text{Diag}(\mu_{\text{min}}^{-1})$
10: end for
Small summary: a model relaxation

\[ \|X - WH\|_F^2 \rightarrow \|X - WH\|_F^2 + \lambda \log \det(W^\top W + \delta I_r) \]

\[ \|X - WH\|_F^2 + \lambda \text{tr}(W^\top W + (\delta - 1)I_r) \]

\[ \|X - WH\|_F^2 + \lambda \text{tr}(D^1W^\top W + D^0) \]

In one sentence: Eigenval-wise convex relaxation of a non-convex problem using logdet – trace inequality.
$D^1$ is diagonal: column decoupling and column-wise BCD

Consider one vector $w_i$ while fixing all other things:

$$\|X - WH\|_F^2 + \lambda \text{tr}(D^1W^TW + D^0)$$

$$= \|X - \sum_i w_i h_i\|_F^2 + \lambda \sum_i (D^1_{ii}\|w_i\|_2^2 + D^0_{ii})$$

Ignoring constants, we have a constrained regularized QP

$$\min w_i \geq 0 \|X - w_i h_i\|_F^2 + \lambda D^1_{ii}\|w_i\|_2^2 + \gamma^2 \|w_i - w_{-i}\|_2^2$$

where $w_{-i}$ is the previous iterate of $w_i$, $\gamma > 0$ is a (small) constant. The proximal term $\|w_i - w_{-i}\|_2^2$ penalizes $w_i$ for leaving $w_{-i}$ too far.
Consider one vector \( w_i \) while fixing all other things:

\[
\|X - WH\|_F^2 + \lambda \text{tr}(D^1W^TW + D^0)
\]

\[
= \|X - \sum_i w_i h_i\|_F^2 + \lambda \sum_i (D^1_{ii} \|w_i\|_2^2 + D^0_{ii})
\]

\[
= \| (X - \sum_{j \neq i} w_j h_j) - w_i h_i \|_F^2 + \lambda \left( D^1_{ii} \|w_i\|_2^2 + \sum_{j \neq i} (D^1_{jj} \|w_j\|_2^2 + D^0_{ii}) \right)
\]

\[
\leq \|X_i - w_i h_i\|_F^2 + \lambda \left( D^1_{ii} \|w_i\|_2^2 + \sum_{j \neq i} (D^1_{jj} \|w_j\|_2^2 + D^0_{ii}) \right)
\]
Consider one vector $w_i$ while fixing all other things:

\[
\|X - WH\|_F^2 + \lambda \text{tr}(D^1W^T W + D^0)
\]

\[
= \|X - \sum_i w_i h_i\|_F^2 + \lambda \sum_i \left( D^1_{ii} \|w_i\|_2^2 + D^0_{ii} \right)
\]

\[
= \| (X - \sum_{j \neq i} w_j h_j) - w_i h_i \|_F^2 + \lambda \left( D^1_{ii} \|w_i\|_2^2 + \sum_{j \neq i} \left( D^1_{jj} \|w_j\|_2^2 + D^0_{ii} \right) \right)
\]

\[
= \|X_i - w_i h_i\|_F^2 + \lambda D^1_{ii} \|w_i\|_2^2 + c
\]
Consider one vector $w_i$ while fixing all other things:

$$
\|X - WH\|_F^2 + \lambda \text{tr}(D^1 W^\top W + D^0) \\
= \|X - \sum_i w_i h_i\|_F^2 + \lambda \sum_i (D^1_{ii} \|w_i\|_2^2 + D^0_{ii}) \\
= \| (X - \sum_{j \neq i} w_j h_j) - w_i h_i\|_F^2 + \lambda \left( D^1_{ii} \|w_i\|_2^2 + \sum_{j \neq i} (D^1_{jj} \|w_j\|_2^2 + D^0_{ii}) \right) \\
= \|X_i - w_i h_i\|_F^2 + \lambda D^1_{ii} \|w_i\|_2^2 + c \\
\leq \|X_i - w_i h_i\|_F^2 + \lambda D^1_{ii} \|w_i\|_2^2 + \frac{\gamma}{2} \|w_i - w_i^-\|_2^2 + c.
$$

Ignoring constants, we have a constrained regularized QP

$$
\min_{w_i \geq 0} \|X_i - w_i h_i\|_F^2 + \lambda D^1_{ii} \|w_i\|_2^2 + \frac{\gamma}{2} \|w_i - w_i^-\|_2^2.
$$

where $w_i^-$ is the previous iterate of $w_i$, $\gamma > 0$ is a (small) constant. The proximal term $\|w_i - w_i^-\|_2^2$ penalizes $w$ for leaving $w^-$ too far.
A non-negative quadratic program (NNQP)

\[
\min_{w_i \geq 0} \| X_i - w_i h_i \|_F^2 + \lambda D_{ii}^1 \| w_i \|_2^2 + \frac{\gamma}{2} \| w_i - w_i^- \|_2^2.
\]

Introducing the proximal term:

- turns the QP problem strongly convex.
- guarantees \( \| h \|_2^2 + \lambda D_{ii}^1 + \gamma > 0 \):

**Lemma** Given \( X, h, z, \alpha, \beta \), the optimal solution of

\[
\min_{w \geq 0} \| X - w h \|_F^2 + \alpha \| w \|_2^2 + \beta \| w - z \|_2^2
\]

is

\[
w = \frac{[Xh^T + \beta z]_+}{\| h \|_2^2 + \alpha + \beta}.
\]

**Proof**: skipped.

Note. This is also related to solving the problem using Newton iteration.
The STA algorithm with decomposed $f(w_i)$

1: **INPUT**: $X \in \mathbb{R}^{m \times n}$, $r \in \mathbb{N}_+$, $\lambda > 0$ and $\delta > 0$

2: **OUTPUT**: $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$

3: **INITIALIZATION**: $W \in \mathbb{R}^{m \times r}$, $H \in \mathbb{R}^{r \times n}$ and $D^1 = I_r$, $\gamma = 10^{-6}$

4: **for** $K = 1$ to itermax **do**

5:  **for** $k = 1$ to itermax **do**

6:   **for** $i = 1$ to $r$ **do**

7:     $w_i = \arg\min_{w_i \geq 0} f(w_i) = \|X_i - w_i h_i\|_F^2 + \lambda D_{ii}^1 \|w_i\|_2^2 + \frac{\gamma}{2} \|w_i - w_i^-\|_2^2$

8:   **end for**

9: **end for**

10: **end for**

11: $\mu_i \leftarrow \text{svd}(W^\top W + \delta I)$ and $D^1 = \text{Diag}(\mu_{\min}^{-1})$

12: **end for**
Applies a close form solution to $f(w_i)$

**Input:** $X \in \mathbb{R}^{m \times n}$, $r \in \mathbb{N}_+$, $\lambda > 0$ and $\delta > 0$

**Output:** $W \in \mathbb{R}_{+}^{m \times r}$ and $H \in \mathbb{R}_{+}^{r \times n}$

**Initialization:** $W \in \mathbb{R}_{+}^{m \times r}$, $H \in \mathbb{R}_{+}^{r \times n}$ and $D^1 = I_r$, $\gamma = 10^{-6}$

**Algorithm:**

1. for $K = 1$ to itermax do
2. for $k = 1$ to itermax do
3. for $i = 1$ to $r$ do
4. $w_i = \frac{[X_i h_i^T + \gamma w_i^-]}{\|h_i\|^2_2 + \lambda D_{ii}^1 + \frac{\gamma}{2}}$ where $X_i = X - \sum_{j \neq i} w_j h_j$
5. Update $H$ by FGM with $X$, $W$, $H$
6. end for
7. end for
8. end for
9. $\mu_i \leftarrow \text{svd}(W^T W + \delta I_r)$ and $D^1 = \text{Diag}(\mu_{\text{min}}^{-1})$
10. end for
Move the update of $H$ outside the loop to reduce computation burden.

1: INPUT: $X \in \mathbb{R}_{+}^{m \times n}$, $r \in \mathbb{N}_{+}$, $\lambda > 0$ and $\delta > 0$
2: OUTPUT: $W \in \mathbb{R}_{+}^{m \times r}$ and $H \in \mathbb{R}_{+}^{r \times n}$
3: INITIALIZATION: $W \in \mathbb{R}_{+}^{m \times r}$, $H \in \mathbb{R}_{+}^{r \times n}$ and $D^1 = I_r$, $\gamma = 10^{-6}$
4: for $K = 1$ to itermax do
5: \hspace{1em} for $k = 1$ to itermax do
6: \hspace{2em} for $i = 1$ to $r$ do
7: \hspace{3em} $w_i = \frac{[X_i h_i^T + \gamma w_i^-]_+}{\|h_i\|_2^2 + \lambda D^1_{ii} + \frac{\gamma}{2}}$ \hspace{0.5em} where \hspace{0.5em} $X_i = X - \sum_{j \neq i} w_j h_j$
8: \hspace{2em} end for
9: \hspace{1em} Update $H$ by FGM with $X$, $W$, $H$
10: end for
11: $\mu_i \leftarrow \text{svd}(W^T W + \delta I_r)$ and $D^1 = \text{Diag}(\mu_{\text{min}}^{-1})$
12: end for
Optimizing STA ... 4

Move the update of $D^1$ inside the loop ($W^TW$ is $r$-by-$r$, small !)

1: INPUT: $X \in \mathbb{R}^{m \times n}$, $r \in \mathbb{N}_+$, $\lambda > 0$ and $\delta > 0$
2: OUTPUT: $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$
3: INITIALIZATION : $W \in \mathbb{R}_+^{m \times r}$, $H \in \mathbb{R}_+^{r \times n}$ and $D^1 = I_r$, $\gamma = 10^{-6}$
4: for $K = 1$ to $\text{itermax}$ do
  5: for $k = 1$ to $\text{itermax}$ do
    6: for $i = 1$ to $r$ do
      7: 
         $w_i = \frac{\begin{bmatrix} X_ih_i^T + \gamma w_i^- \end{bmatrix}^+}{\|h_i\|_2^2 + \lambda D^1_{ii} + \frac{\gamma}{2}}$
         where $X_i = X - \sum_{j \neq i} w_jh_j$
    8: $\mu_i \leftarrow \text{svd}(W^TW + \delta I_r)$ and $D^1 = \text{Diag}(\mu_{\text{min}}^{-1})$
  9: end for
10: Update $H$ by FGM with $X$, $W$, $H$
11: end for
12: end for
Now consider line 7, it can be show that it can be further optimized\(^3\).

1: INPUT: \(X \in \mathbb{R}^{m \times n}, r \in \mathbb{N}_+, \lambda > 0 \) and \(\delta > 0\)
2: OUTPUT: \(W \in \mathbb{R}^{m \times r}_+\) and \(H \in \mathbb{R}^{r \times n}_+\)
3: INITIALIZATION : \(W \in \mathbb{R}^{m \times r}_+, H \in \mathbb{R}^{r \times n}_+\) and \(D^1 = I_r, \gamma = 10^{-6}\)
4: for \(K = 1\) to \(\text{itermax}\) do
5: \hspace{1em} for \(k = 1\) to \(\text{itermax}\) do
6: \hspace{2em} for \(i = 1\) to \(r\) do
7: \hspace{3em} \(w_i = \frac{[X_i h^T + \gamma w_i^-]}{\|h\|^2 + \lambda D^1_{ii} + \frac{\gamma}{2}}\) where \(X_i = X - \sum_{j \neq i} w_j h_j\)
8: \hspace{3em} \(\mu_i \leftarrow \text{svd}(W^T W + \delta I_r)\) and \(D^1 = \text{Diag}(\mu_{\text{min}}^{-1})\)
9: \hspace{2em} end for
10: Update \(H\) by FGM with \(X, W, H\)
11: end for
12: end for

\(^3\)N. Gillis and F. Glineur, ”Accelerated Multiplicative Updates and Hierarchical ALS Algorithms for Nonnegative Matrix Factorization”, Neural Computation 24 (4), 2012
On update of $w_i$

Line 7

$$w_i = \frac{[X_i h_i^T + \gamma w_i^-]_+}{\|h_i\|^2_2 + \lambda D_{ii} + \frac{\gamma}{2}} \quad \text{where} \quad X_i = X - \sum_{j \neq i} w_j h_j.$$
On update of $w_i$

Line 7

$$w_i = \frac{[X_i h_i^T + \gamma w_i^-]_+}{\|h_i\|_2^2 + \lambda D_{ii}^1 + \frac{\gamma}{2}}$$

where

$$X_i = X - \sum_{j \neq i} w_j h_j.$$  

- Consider $w_2$. 
On update of $w_i$

Line 7

$$w_i = \frac{[X_i h_i^T + \gamma w_i^-]}{\|h_i\|^2 + \lambda D_{ii} + \gamma} + \|h_i\|^2 + \lambda D_{ii} + \gamma$$

where $X_i = X - \sum_{j \neq i} w_j h_j$.

$\bullet \quad X - WH = X_i - w_i h_i$. 

Diagram: 

\[\text{Grid Diagram}\]
On update of $w_i$

Line 7

$$w_i = \frac{[X_i h_i^T + \gamma w_i^-]_+}{\| h_i \|^2_2 + \lambda D_{ii}^1 + \frac{\gamma}{2}} \quad \text{where} \quad X_i = X - \sum_{j \neq i} w_j h_j.$$ 

- The term $X_i h_i^T$
On update of $w_i$

Line 7

$$w_i = \frac{[X_i h_i^T + \gamma w_i^-]}{\|h_i\|^2 + \lambda D_{ii}^1 + \frac{\gamma}{2}}$$

where

$$X_i = X - \sum_{j \neq i} w_j h_j.$$

\[\text{\checkmark} \quad X_i h_i^T = X h_i^T - \sum_{j \neq i} w_j h_j h_i^T\]
On update of $w_i$

Line 7

\[ w_i = \frac{[X_i h_i^T + \gamma w_i^-]}{\|h_i\|^2_2 + \lambda D_{ii}^1 + \frac{\gamma}{2}} \]

where \( X_i = X - \sum_{j \neq i} w_j h_j \).

\[ Xh_i^T - \sum_{j \neq i} w_j h_j h_i^T = [XH^\top](; , i) - W(:, 1, ..., i - 1, i + 1, ..., r)[HH^\top](j, i) \]

without this
On update of $w_i$

Line 7

$$w_i = \frac{[X_i h_i^T + \gamma w_i^-]}{\|h_i\|^2_2 + \lambda D_{i i}^1 + \frac{\gamma}{2}}$$

where $X_i = X - \sum_{j \neq i} w_j h_j$.

is equivalent to

$$w_i = \frac{[P_i - \sum_{j=1}^{i-1} w_j Q_{ji} - \sum_{j=i+1}^{r} w_j Q_{ji} + \gamma w_i^-]}{Q_{ii} + \lambda D_{i i}^1 + \frac{\gamma}{2}}$$

where $P = XH^\top$, $P_i = P(:, i)$, $Q = HH^\top$ and $Q_{ji} = Q(j, i)$. $P$ and $Q$ can be pre-computed.
Finally we have Algorithm 2 STA

1: INPUT: $X \in \mathbb{R}^{m \times n}$, $r \in \mathbb{N}_+$, $\lambda > 0$ and $\delta > 0$
2: OUTPUT: $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$
3: INITIALIZATION : $W \in \mathbb{R}^{m \times r}$, $H \in \mathbb{R}^{r \times n}$ and $D^1 = I_r$, $\gamma = 10^{-6}$
4: for $K = 1$ to itermax do
5:    for $k = 1$ to itermax do
6:        $P = XH^\top$ and $Q = HH^\top$.
7:        for $i = 1$ to $r$ do
8:            $w_i = \left[ P_i - \sum_{j=1}^{i-1} w_j Q_{ji} - \sum_{j=i+1}^r w_j^- Q_{ji} + \gamma w_i^- \right]^+$
9:            $Q_{ii} + \lambda D^1_{ii} + \frac{\gamma}{2}$
10:       $\mu_i \leftarrow \text{svd}(W^\top W + \delta I_r)$ and $D^1 = \text{Diag}(\mu_{\text{min}}^{-1})$
11:    Update $H$ by FGM with $X$, $W$, $H$
12: end for
13: end for

In fact, line 7 – 10 can be run multiple times for ”better” convergence.
\[ \|X - WH\|_F^2 \rightarrow \|X - WH\|_F^2 + \lambda \log \det(W^TW + \delta I_r) \]

\[ \|X - WH\|_F^2 + \lambda \text{tr}(W^TW + (\delta - 1)I_r) \]

\[ \|X - WH\|_F^2 + \lambda \text{tr}(D^1W^TW + D^0) \]

\[ \sum_i \|X_i - w_i h_i\|_F^2 + \lambda D^1_{ii} \|w_i\|_2^2 + \frac{\gamma}{2} \|w_i - w_i^-\|_2^2 \]
Synthetic data

- Ground truth: $W_0 \in \mathbb{R}^{m \times r}$ and $H_0 \in \mathbb{R}^{r \times n}$ (with $H^T 1 = \alpha$)
- $m$, $n$ are sizes, $r$ is rank, $\alpha \in (0, 1]$ tells how "well spread" the data are ($\alpha = 1$ means pure pixel)
- Form $X_0$ as $X_0 = W_0 H_0 \in \mathbb{R}^{m \times n}$
- Add noise $N \in \mathbb{R}^{m \times n}$ and $N \sim N(0, R)$ as $X = X_0 + N$.
- As $N \in \mathbb{R}$ not $\mathbb{R}^+$, corrupted data points may lie outside the convex hull.
Real data (An example: Copperas Cove Texas Walmart)

Figure: RGB image of Copperas Cove Texas Walmart

Figure: Three spectral images of Copperas Cove Texas Walmart

\[ \mathbf{X} \in \mathbb{R}_+^{94249 \times 162}, \text{ or } \mathbf{X} \in \mathbb{R}_+^{307 \times 307 \times 162}, \text{ pick } r = 5, 6, 7. \]
Performance measurements. Algorithm produces $\hat{W}$ and $\hat{H}$ and $\hat{X} = \hat{W}\hat{H}$

- For simulation with known $W_0$, $H_0$, $X_0$:
  - Data fitting error: $\frac{\|X_0 - \hat{X}\|_F}{\|X_0\|_F}$
  - Endmember fitting error: $\frac{\|W_0 - \hat{W}\|_F}{\|W_0\|_F}$
  - Computational time

- For real data without knowing $W_0$, $H_0$, $X_0$:
  - Data fitting error: $\frac{\|X - \hat{X}\|_F}{\|X\|_F}$
  - Volume of convex hull: $\log \det(\hat{W}^\top \hat{W} + \delta I_r)$
  - Computational time
Results and fancy figures

- What iterations of the algorithm looks like: the gif file
- Effect of fix lambda: EXAMPLE Rotate
- Effect of very big lambda: EXAMPLE Big lambda
Comparing the two logdet inequality - error

\[ \frac{\|W_0 - W\|_F}{\|W_0\|_F} \text{ vs. } \frac{\|X - WH\|_F}{\|X\|_F} \]

- **Taylor**
- **Eigen**
- inner loop
- outer loop
Comparing the two logdet inequality - fitting
Theoretical stuff — convergence

We have problems:

\[ (P_0) \] minimizes \( \| X - WH \|^2_F + \lambda \log \det(W^T W + \delta I_r) \),

\[ (P_1) \] minimizes \( \| X - WH \|^2_F + \lambda \text{tr}(D^1 W^T W + D^0) \),

under the constraints: \( W \geq 0, H \geq 0, H^T 1 \leq 1 \).

Convergence properties:

1. STA algorithm produces a stationary point for problem \( P_1 \).
2. The solution of \( P_1 \) obtained by STA converges to the solution of \( P_0 \).
3. Convergence rate of STA algorithm.

Idea: consider \( W, H \) and \( D^1 \), (and \( D^0 \)) as variables and treat STA as an Inexact Block Coordinate Descent (BCD) algorithm / Block Successive Upper bound Minimization (BSUM): at each iteration on variable \( W \), we are not considering the original problem but an upper bound function (with the proximal term) \( \| X_i - w_i h_i \|^2_F + \lambda D^1_{ii} + \frac{\gamma}{2} \| w_i - w_i^- \|^2 \), notice that the inexactness is also contributed by the eigen-gap introduced by \( \mu_i \geq \mu_{\min} \).
The "simplified" STA algorithm (the updates of $D$ and $H$ are hidden here)

Algorithm 3 STA with cyclic indexing

1: for $k = 1$ to itermax do
2:   for $i = 1$ to $r$ do
3:     Update $w_i$ by doing something
4:   end for
5: end for

STA is a Inexact BCD with cycling indexing. That is, $w_i$ is selected according to $i = 1, 2, ..., r, 1, 2, ..., r, ....$

In fact, cyclic indexing is not optimal. Acceleration can be made on using random indexing and extrapolation. The next page will discuss accelerated STA that, in expectation, converges faster.
Algorithm 4 Accelerated STA with random indexing

1: Set $v_i^k = w_i$ for $i = 1, 2, ..., r$.

2: for $k = 1$ to itermax do

3: Pick $i = i_k$ by random (with uniform probability)

4: $y_i^k = a_k v_i^k + (1 - a_k) w_i^k$

5: $w_i^{k+1} = y_i^k - \frac{1}{L_{i_k}} \nabla_{i_k} f(y_i^k)$

6: $v_i^{k+1} = b_k v_i^k + (1 - b_k) y_i^k - \frac{c_k}{L_{i_k}} \nabla_{i_k} f(y_i^k)$

7: end for

where $\sigma$ is strong convexity parameter and $L_{i_k}$ is smoothness parameter of function $f(w_i)$. And parameters $a, b, c$ are obtained by solving the following non-linear equations

$$c_k^2 - \frac{c_k}{r} = (1 - \frac{c_k \sigma}{r}) c_{k-1}^2, \quad a_k = \frac{r - c_k \sigma}{c_k (r^2 - \sigma)}, \quad b_k = 1 - \frac{c_k \sigma}{r}$$

Idea is similar to the "extrapolation induced acceleration" of the Nesterov's accelerated (full-)gradient.
The "simplified" accelerated STA algorithm:

**Algorithm 5** Accelerated STA with random indexing

1: for $k = 1$ to $\text{itermax}$ do
2: Pick $i = i_k$ by random
3: update $w_i$ by doing something, together with two additional series $v_i^k$ and $y_i^k$
4: end for

The trade off of faster convergence is the "randomness": picking up repeated index is possible $i = 1, 3, 3, 3, 4, 2, 5, ...$

Solution is to use random shuffle instead of totally random. For example, $r = 3$, and

\[ i = [1, 3, 2], [3, 2, 1], [3, 1, 2], [1, 2, 3], [2, 1, 3], ... \]

But the analysis of the convergence property becomes complicated.
Extension: noise, outlier and robustness (1/3)

Formulation $(P_0)$ and $(P_1)$ are sensitive to **outlier** and **noise**.

- **Outlier**: a single outlier **can kill** the algorithm.

![Graphs showing original data, data with outlier, changes, and final result.](image)
Formulation \((P_0)\) and \((P_1)\) are sensitive to outlier and noise.

- **Outlier**: a single outlier can kill the algorithm.

Solution: robust norm on data fitting:

\[
\|X - WH\|_F^2 \xrightarrow{\text{changed to}} \|X - WH\|_\phi^2
\]

Examples of \(\phi\)

- \[
\|X - WH\|_{2-1}^2 + \lambda \text{tr}(D^1W^TW + D^0) \quad (L_{2-1} \text{ norm, column robust})
\]
- \[
\|X - WH\|_1^2 + \lambda \text{tr}(D^1W^TW + D^0) \quad (L_1 \text{ norm, matrix robust})
\]
- \[
\|X - WH\|_p^2 + \lambda \text{tr}(D^1W^TW + D^0) \quad (L_p \text{ norm, tunable})
\]
- \[
\|X - WH\|_B^2 = \sum_{ij} B_{ij}(X - WH)_{ij}^2 = \|B \odot (X - WH)\|_F^2
\]
Extension: noise, outlier and robustness (2/3)

Formulation \((P_0)\) and \((P_1)\) are sensitive to **outlier** and **noise**.

- **Noise**: \(F\)-norm is for additive Gaussian noise.
Formulation \((P_0)\) and \((P_1)\) are sensitive to **outlier** and **noise**.

- **Noise**: \(F\)-norm is for additive Gaussian noise.
- Other noises and corresponding denoising norms:
  - Kullback Leibler divergence for Poisson noise
  - Itakura Saito divergence for Gamma Exponential noise
  - Laplacian / Double Exponential noise, Uniform noise, Lorentz noise

*Figure: Noises. Source: https://gimper.net/resources/noise-generator.*
Formulations \((P_0)\) and \((P_1)\) are sensitive to \textbf{outlier} and \textbf{noise}.

**Solution - 1 :** solve \(\|X - WH\|_p^2 + \lambda \text{tr}(D^1W^TW + D^0)\) by \textbf{Iterative Reweighted Least Squares (IRLS)}

Idea:

\[
\|x\|_p = \left( \sum_i |x_i|^p \right)^{1/p} = \left( \sum_i w_i^2 |x_i|^2 \right)^{1/2}
\]

where weight \(w_i = \frac{p-2}{p} x_i^2\). For application \(w_i\) can be set as \(x_i^{-}\).

i.e. Approximate \(\|X - WH\|_p^2\) as a weighted \(L_2\) norm problem.
Extension: noise, outlier and robustness (3/3)

Formulations \((P_0)\) and \((P_1)\) are sensitive to **outlier** and **noise**.

**Solution - 1**: solve \(\|X - WH\|_p^2 + \lambda \text{tr}(D^1W^TW + D^0)\) by **Iterative Reweighted Least Squares (IRLS)**

Idea:

\[
\|x\|_p = \left( \sum_i |x_i|^p \right)^{1/p} = \left( \sum_i w_i^2 |x_i|^2 \right)^{1/2}
\]

where weight \(w_i = \frac{x_i^{p-2}}{2}\). For application \(w_i\) can be set as \(x_i^{-}\).

i.e. Approximate \(\|X - WH\|_p^2\) as a weighted \(L_2\) norm problem.

Or, approximate matrix \(L_1\) norm by IRLS:

\[
\|X - WH\|_B^2 = \sum_{ij} B_{ij} (X - WH)_{ij}^2
\]

\[
\approx \|X - WH\|_1^2
\]

for \(B_{ij} = \frac{1}{\|X - WH\|_{ij} + \epsilon}\)
Formulations ($P_0$) and ($P_1$) are sensitive to **outlier** and **noise**. 

**Solution - 2** : use $L_{2-1}$ norm, or more specific :

$$
\sum_{j=1}^{n} \frac{1}{2} \left( \|X(:,j) - WH(:,j)\|_2^2 + \epsilon \right)^{\frac{p}{2}} + \lambda \text{tr}(D^1W^\top W + D^0)
$$

where $\epsilon > 0$ is smoothness constant and $p \in (0, 2]$ is robustness parameter.

FGM cannot be used directly on this formulation, need some modifications.
\[
\|X - WH\|^2_F + \lambda \log \det(W^TW + \delta I)
\]
\[
\|X - WH\|^2_F + \lambda \text{tr}(W^TW + (\delta - 1)I_r)
\]
\[
\|X - WH\|^2_F + \lambda \text{tr}(D^1W^TW + D^0)
\]
\[
\sum_i \|X_i - w_i h_i\|^2_F + \lambda D^1_{ii}\|w_i\|^2 + \frac{\gamma}{2}\|w_i - w_i^-\|^2
\]
Further discussions

Standard open problems

- How to tune $\lambda$
  - small noise $\lambda \to 0$
  - large noise $\lambda \to \infty$
  - $\lambda(N) = ?$

- How to find $r$ (in real world application you don’t know the $r$ !)

Other directions

- On solving $H$
- On super-big data
- Acceleration by weighted formulation
Further discussions: how to tune $\lambda$

**A very difficult problem.** How to tune $\lambda$ dynamically within the iterations?? That is, find an expression in the form as:

$$\lambda = \lambda(X, W_k, H_k, k)$$

where $k$ is the current number of iteration.

Don’t expect I can give a global solution! A rough idea of approaches:

- Simulated Annealing
- Dynamic approach
- Hybrid approach

**Pros:** easy to implement

**Cons:** very hard to establish theoretical convergence guarantee
Simulated Annealing parameter tuning of $\lambda$

- Annealing (metallurgy) = heat treatment of metal
- At the beginning, start with very high temperature to make coarse adjustment of the metal (hammering)
- Temperature is gradually decrease in the process, and gradually moving from coarse adjustment to fine adjustment
- Finally the metal is cooled down

\[ f(W, H) + \lambda G(W) \]

(high temperature = starting with very large $\lambda$) 

Coarse adjustment of the metal = rotation of the convex hull

Temperature is gradually decrease = gradually decrease the value of $\lambda$

Fine adjustment = growth of convex hull

metal is cooled down = $\lambda$ is very close to zero

( Reminder to myself : refer to the "rotate.gif" )
Simulated Annealing parameter tuning of $\lambda$

- Annealing (metallurgy) = heat treatment of metal
- At the beginning, start with very high temperature to make coarse adjustment of the metal (hammering)
- Temperature is gradually decrease in the process, and gradually moving from coarse adjustment to fine adjustment
- Finally the metal is cooled down

On the problem $f(W, H) + \lambda G(W)$,

- high temperature = starting with very large $\lambda$
- Coarse adjustment of the metal = rotation of the convex hull
- Temperature is gradually decrease = gradually decrease the value of $\lambda$
- Fine adjustment = growth of convex hull
- metal is cooled down = $\lambda_k$ is very close to zero
- (Reminder to myself : refer to the ”rotate.gif”)
Some discussions on the tuning

- Very primitive, not robust, non-determininistic
- Only works with cases that $W_0$ is known: is $\log \det(W^TW + \delta I_r)$ really a good estimator of $\|W_0 - \hat{W}\|_F$? How about $\det W^TW$?

\[ \Rightarrow \text{Compare the models } \log \det(W^TW + \delta I_r) \text{ with } \det W^TW \]

Equivalent problems:

\[
(P_0) \quad \min \quad \|X - WH\|_F^2 + \lambda \log \det(W^TW + \delta I_r),
\]

\[
(P_1) \quad \min. \quad \text{tr}(D^1W^TW + D^0) \quad \text{s.t.} \quad \|X - WH\|_F^2 \leq \epsilon
\]

For every $\epsilon$ in $P_1$, there exists a $\lambda$ in $P_0$ that both of them share the same solution. Solving $P_1$ does not involve parameter tuning.
Recall the STA algorithm (with first 3 lines removed)

1: for \( K = 1 \) to itermax do 
2: \hspace{0.5cm} for \( k = 1 \) to itermax do 
3: \hspace{1cm} \( P = XH^\top \) and \( Q = HH^\top \). 
4: \hspace{1cm} for \( i = 1 \) to \( r \) do 
5: \hspace{1.5cm} \( w_i = \frac{P_i - \sum_{j=1}^{i-1} w_j Q_{ji} - \sum_{j=i+1}^{r} w_j Q_{ji} + \gamma w_i}{Q_{ii} + \lambda D_{ii} + \frac{\gamma}{2}} \) 
6: \hspace{1.5cm} \( \mu_i \leftarrow \text{svd}(W^\top W + \delta I_r) \) and \( D^1 = \text{Diag}(\mu_{\text{min}}^{-1}) \) 
7: \hspace{1cm} end for 
8: \hspace{0.5cm} Update \( H \) by FGM with \( X, W, H \) 
9: \hspace{0.5cm} end for 
10: end for 

This line is to solve

\[
H \leftarrow \arg \min_{H \geq 0, \ 1_r^\top H \leq 1_n} \|X - WH\|_F^2 + \lambda \log \det(W^\top W + \delta I_r) .
\]

\( a \) constant for \( H \)
Further improving STA : on H (2/3)

Is the FGM (Fast gradient method on constrained least square on unit simplex) really the best?

1: for $K = 1$ to itermax do
2:   for $k = 1$ to itermax do
3:     $P = XH^\top$ and $Q = HH^\top$.
4:     for $i = 1$ to $r$ do
5:       $w_i = \left[ P_i - \sum_{j=1}^{i-1} w_j Q_{ji} - \sum_{j=i+1}^{r} w_j Q_{ji} + \gamma w_i \right]_+$
6:       $\mu_i \gets \text{svd}(W^\top W + \delta I_r)$ and $D^1 = \text{Diag}(\mu_{\min}^{-1})$
7:     end for
8:     Update $H$ by FGM with $X$, $W$, $H$
9:   end for
10: end for
Currently, yes.

**Figure:** Four methods on H.

* primal algorithm vs primal-dual algorithm.
We assumed $r$ is known, it is only true for synthetic experiment. In real application, no one know the true $r$.

Idea: if input $r$ is larger than the true $r$, when the minimum volume is 'achieved', then there will be (almost-)colinear columns in $W \implies$ a way to auto-detect $r$!!

Reminder to me: run the large_r.gif in chrome.
Further issues: acceleration by weighted sum formulation

Since
\[ \|X - WH\|_F^2 = \sum_j \|X(:, j) - WH(:, j)\|_2^2. \]

What if the data fitting terms becomes
\[ \sum_j \alpha_i \|X(:, j) - WH(:, j)\|_2^2, \]
where \( \alpha_i \) are weights.

Idea: increase the weight on the points that \( \notin \text{conv}(W) \) to speed up the fitting of the vertices of the next iteration.

An example: The problem is illustrated on a rotated cube (\( W \)).
Introduce / review NMF $\|X - WH\|_F^2$

Minimum volume NMF $\|X - WH\|_F^2 + \lambda \log \det (W^T W + \delta I_r)$

Successive Trace Approximation $\text{tr}(D^1W^T W + D^0)$

The STA algorithm and refinements

BCD acceleration by randomization (on index)

Convergence of STA (just rough idea)

Robust STA (just rough idea) : $\|X - WH\|_\phi$, $\phi \in \{1 \leq p \leq 2, 2 - 1\}$, Iterative reweighted least squares

Experiments : some rough illustrations

Some open / unsolved problems and further refinements

Slides (and code) available at angms.science
Extra - 1. What if some elements of $\mathbf{H}$ is very very small

If some $\mathbf{H}$ of the data are very small, convex hull looks like conical hull
Extra - 1. What if some elements of $\mathbf{H}$ is very very small

**Solution-1** See them as outliers
Their $H_{ij}$ small $\implies$ norm small $\implies$ error small $\implies$ ok if only a few of them

**Solution-2** Augmenting $\mathbf{W} = [\mathbf{W}' \; \mathbf{a}]$ where $\mathbf{a}$ is a small vector.
Note: $r$ changes, and $\mathbf{a}$ cannot be $0$ as matrix $[\mathbf{W} \; 0]$ is not full rank
Extra - 2. What if data are clustered

What if:
Still works, but other method will be better (e.g. some clustering method such as K-means)