Non-negative Matrix Factorization(s)

Applications, Theories and Computations

Andersen Ang

Mathématique et recherche opérationnelle
UMONS, Belgium

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

August 20, 2019
Dept. Applied Maths, U.Waterloo, Canada

Outline

Non-negative Matrix Factorization (NMF)

- **Given** matrix $\mathbf{M} \in \mathbb{R}_+^{m \times n}$, positive integer r.
- **Find** matrices $\mathbf{U} \in \mathbb{R}_+^{m \times r}, \mathbf{V} \in \mathbb{R}_+^{n \times r}$ s.t. $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$.

Non-negative Matrix Factorization (NMF)

- **Given** matrix $\mathbf{M} \in \mathbb{R}_+^{m \times n}$, positive integer r.
- **Find** matrices $\mathbf{U} \in \mathbb{R}_+^{m \times r}, \mathbf{V} \in \mathbb{R}_+^{n \times r}$ s.t. $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$.
- Everything is non-negative.

Related problems

- NMF

$$\mathbf{M} = \mathbf{U}\mathbf{V}^\top$$

- N. Tensor Factorization

$$\mathcal{T} = \mathbf{U} * \mathbf{V} * \mathbf{W}$$

- N. Least Squares

$$\mathbf{b} = \mathbf{A}\mathbf{x}$$

- "N" $\implies$ all these problems no analytic sol.
  $\implies$ seek for numerical sol.

Related problems

- NMF

$$\underset{\{\mathbf{U},\mathbf{V}\}\geq\mathbf{0}}{\operatorname{argmin}} \|\mathbf{M}-\mathbf{U}\mathbf{V}^{\top}\|_F$$

- NTF

$$\underset{\{\mathbf{U},\mathbf{V},\mathbf{W}\}\geq\mathbf{0}}{\operatorname{argmin}} \|\mathcal{T}-\mathbf{U}*\mathbf{V}*\mathbf{W}\|_F$$

- NNLS

$$\underset{\mathbf{x}\geq\mathbf{0}}{\operatorname{argmin}} \|\mathbf{b}-\mathbf{A}\mathbf{x}\|_2$$

- all are constrained non-linear programming problem (also non-smooth : between boundary of $\mathbb{R}_+$, $\mathbb{R}_-$)

Outline

Why study these problems?

- **Model interpretability**

  NMF gives better decomposition than PCA, SVD, ICA due to the interpretability on non-negative data.

- **Model correctness**

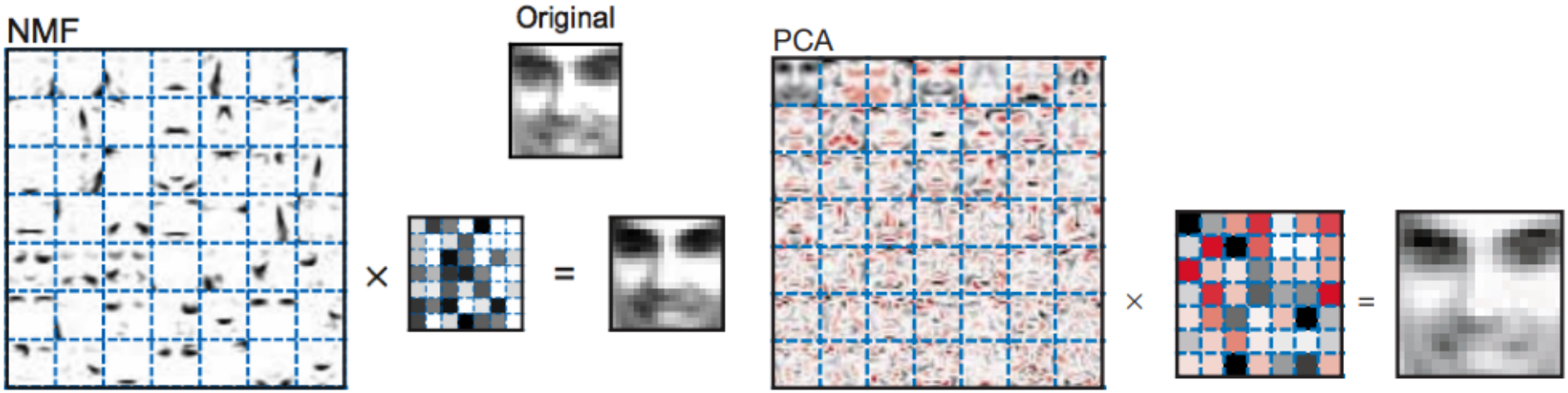
  NMF can find ground truth (under certain conditions).

- **Mathematical curiosity**

  NMF is related to some serious problems in mathematics.

- ~~My boss tell me to do it.~~

Application 1 - Representation Learning

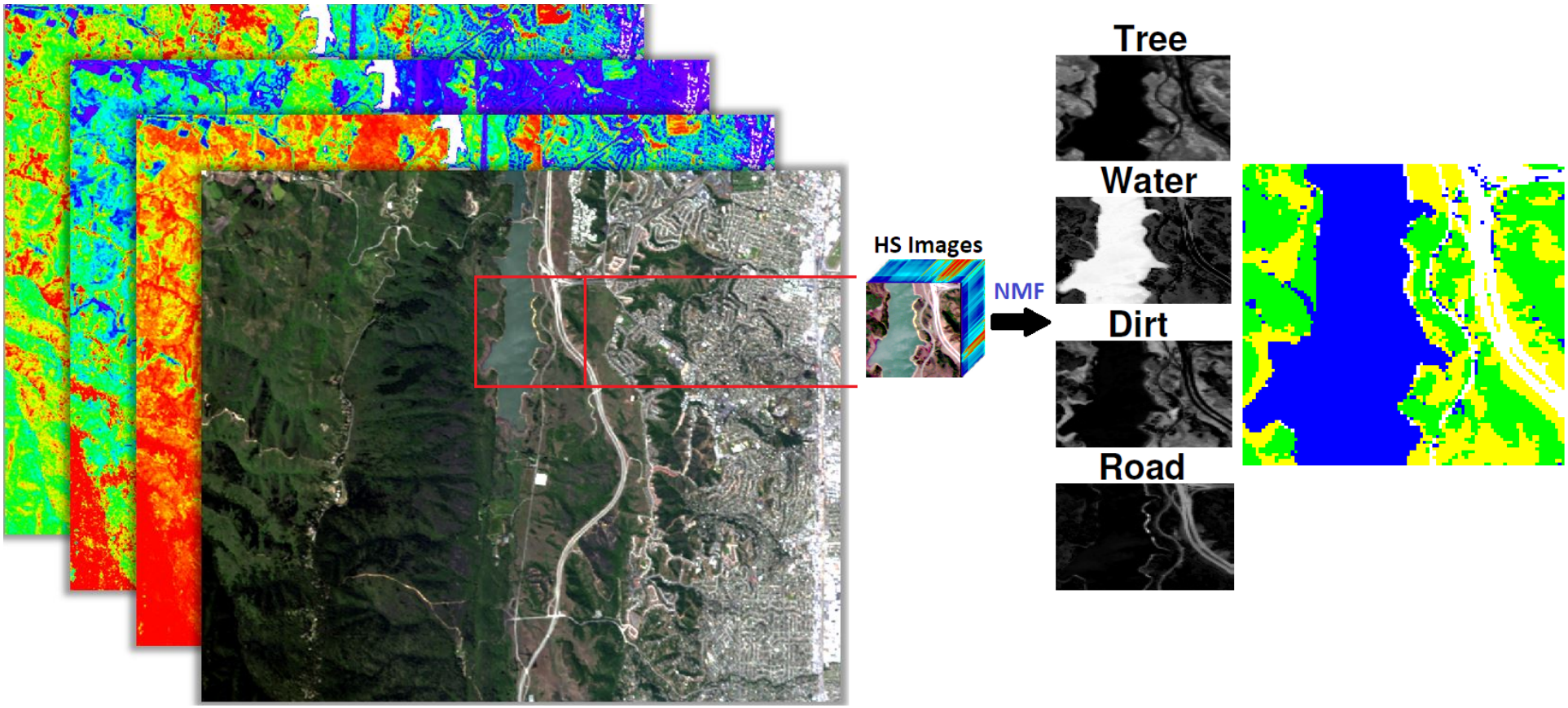


(Lee & Seung, 1999)

The work that "popularized" NMF.

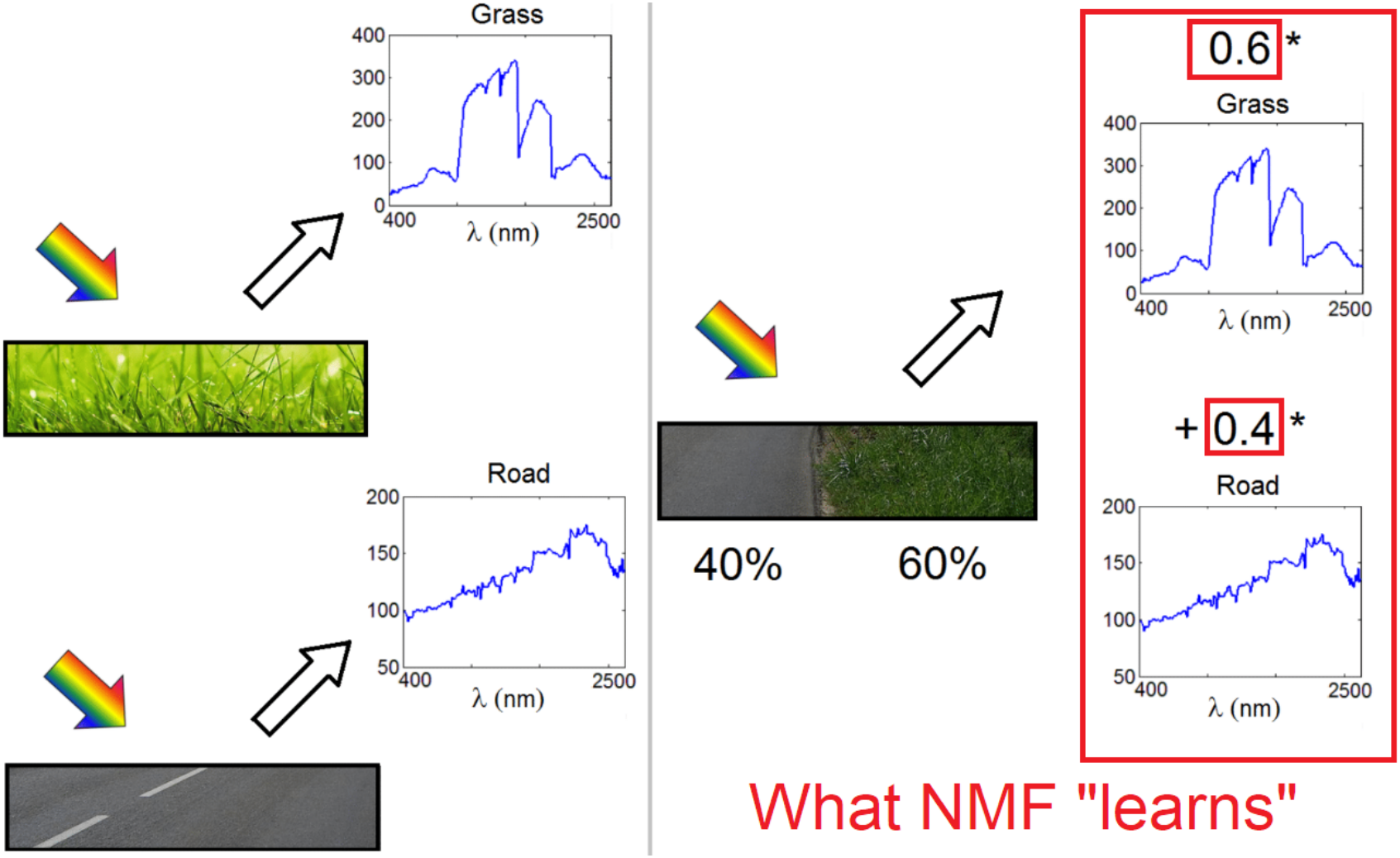
Application 2 - Hyper-Spectral Imaging

NMF gives good *unsupervised* image segmentation.



Decomposition of hyper-spectral image of Jasper Ridege, California.
Left : From HySpeed Computing. Right : (A. & Gillis, 2019-HSI).
Note : the left and right are not taken in the same period.

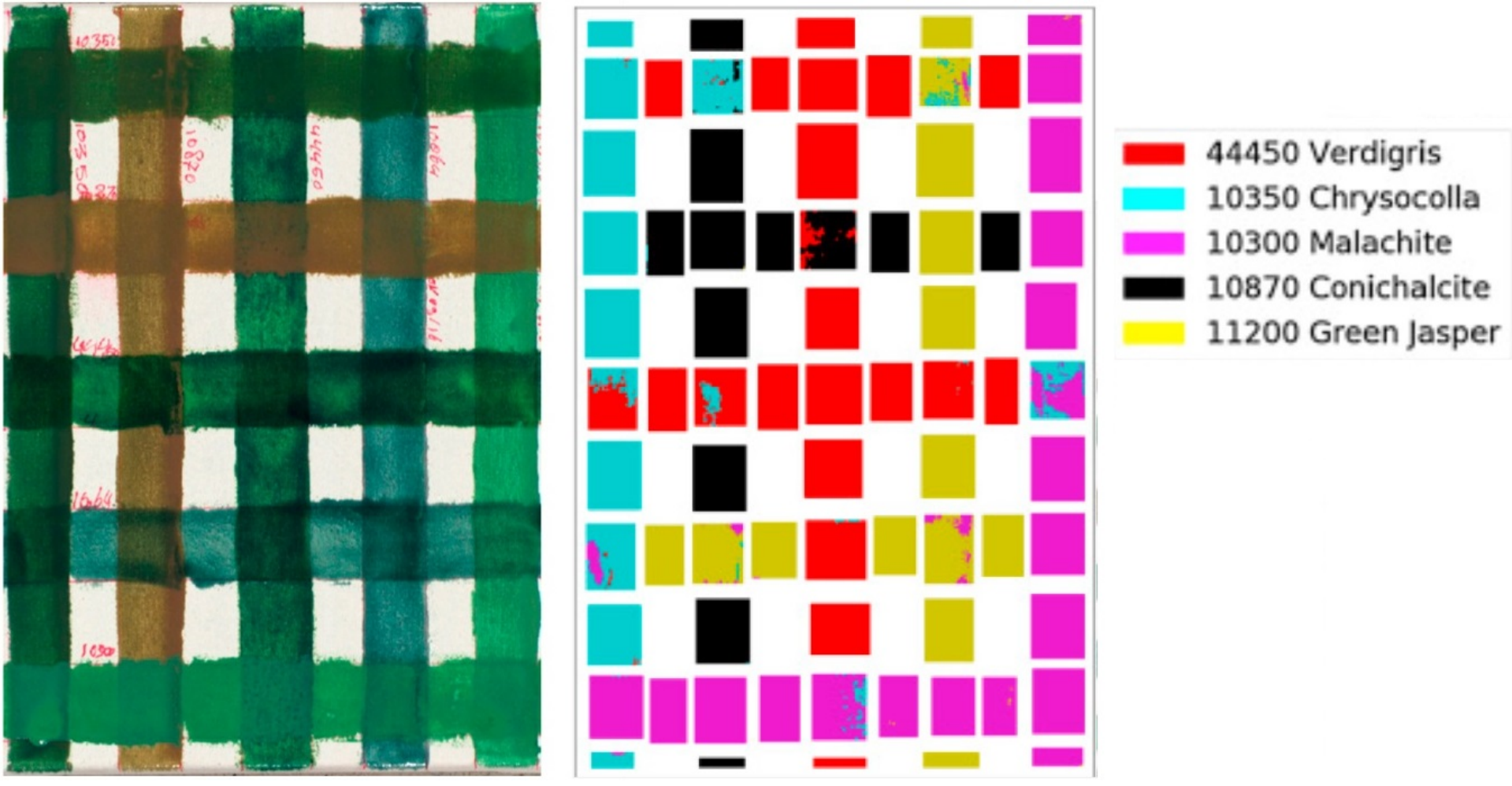
Application 2 - Hyper-Spectral Imaging



HSI decomposition. Figure modified from the slide of Nicolas Gillis.

Related models : NMF with sparsity / volume regularizations.

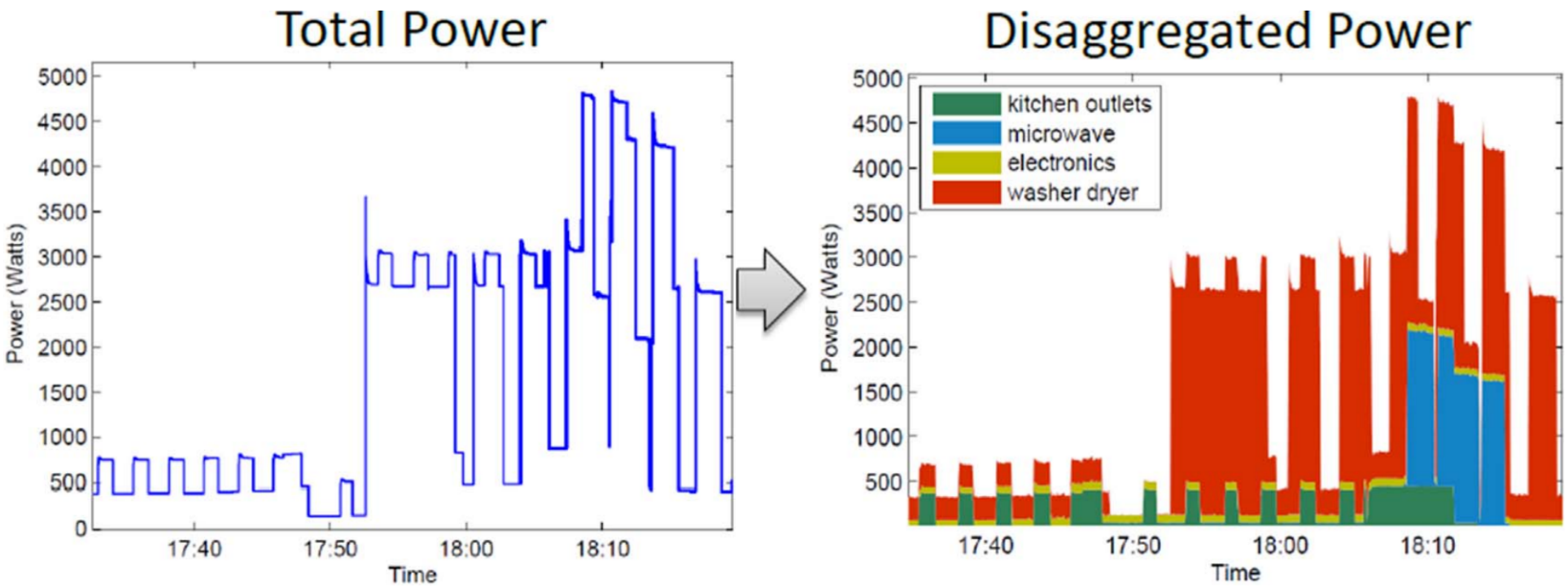
Application 3 - Art preservation and archaeology



Pigment identification (Grabowski, et. al, 2018).

Related model : NMF dictionary learning (here dictionary = colour book).

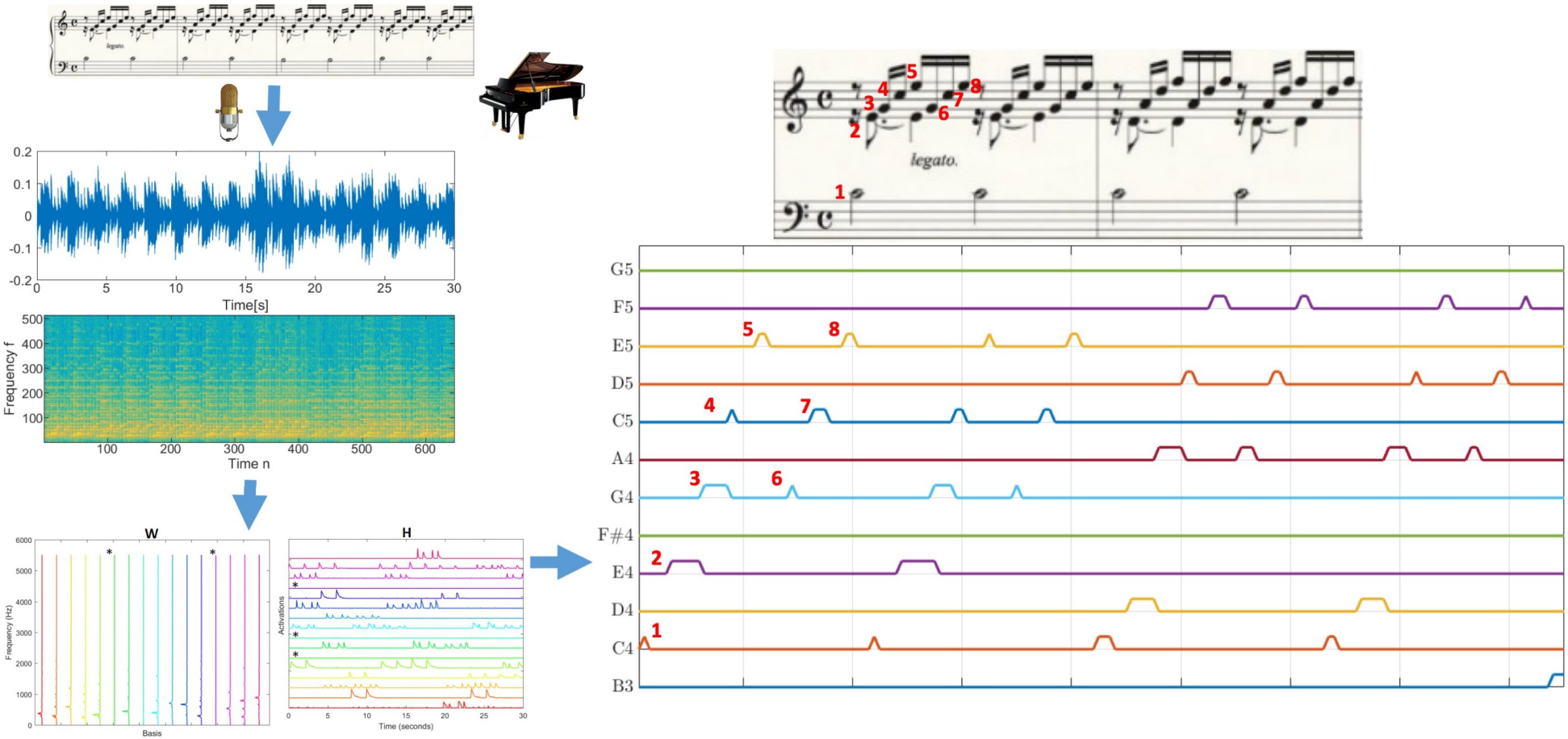
Application 4 - Smart Home, Electricity Disaggregation



Electricity Disaggregation (Kolter, 2011).

Related model : NMF with l_0/sparsity constraints/regularizers.

Application 5 - Audio Blind Source Separation



(Leplat, Gillis, Siebert, A., 2019) and (Leplat, Gillis, A., 2019)

Related model : Beta-divergence NMF with volume regularizer.

Application side

- Demixing : analytical chemistry (earliest work), HSI image, Audio
- Representation learning on human face (the work that popularizes NMF)
- Topic modeling in text mining
- Probability distribution application on identification of Hidden Markov Model
- Bioinformatics : gene expression
- (Non-negative) Data compression for tensor completion, video foreground-background separation
- Speech denoising
- Recommender system
- Video summarization
- Radio
- Forensics
- Art work conservation (identify true color used in painting)
- Medical imaging – image processing on small object
- Mid-infrared astronomy – image processing on large object
- Telling whether a banana or a fish is healthy

Numerical side

- A test-box for generic optimization programs : NMF is a constrained non-convex (but biconvex) problem
- Robustness analysis of algorithm
- Tensor
- Sparsity

Theoretical side

- Non-negative rank $\text{rank}_+ :=$ smallest r s.t.

$$\mathbf{X} = \sum_{i=1}^{r} \mathbf{X}_i, \quad : \ \mathbf{X}_i \text{ non-negative rank-1.}$$

  How to find / estimate / bound rank_+, e.g. $\text{rank}_{\text{psd}}(\mathbf{X}) \leq \text{rank}_+(\mathbf{X})$, CP

- Extended formulations and combinatorics
- Log-rank Conjecture of communication system
- 3-SAT, Exponential time hypothesis, $\mathbf{P} \neq \mathbf{NP}$

Various models for various applications

- Basic NMF model

$$\underset{\{\mathbf{U},\mathbf{V}\}\geq\mathbf{0}}{\operatorname{argmin}} f(\mathbf{U},\mathbf{V};\mathbf{M}) = \|\mathbf{M}-\mathbf{U}\mathbf{V}^\top\|_F$$

- NMF with l_0-norm/sparsity constraint

$$\underset{\{\mathbf{U},\mathbf{V}\}\geq\mathbf{0}}{\operatorname{argmin}} f(\mathbf{U},\mathbf{V};\mathbf{M}) + \lambda_U\|\mathbf{U}\|_0 + \lambda_V\|\mathbf{V}\|_0$$

- NMF under other objective functions

$$\underset{\{\mathbf{U},\mathbf{V}\}\geq\mathbf{0}}{\operatorname{argmin}} \mathcal{D}(\mathbf{M},\mathbf{U}\mathbf{V})$$

- NMF under separability constraints

$$\underset{\{\mathbf{U},\mathbf{V}\}\geq\mathbf{0}}{\operatorname{argmin}} f(\mathbf{U},\mathbf{V};\mathbf{M}) \text{ s.t. } \mathbf{U}=\mathbf{M}(,:\mathcal{J}), \mathbf{V}=\begin{bmatrix}\mathbf{I}_r\\ \mathbf{V}'\end{bmatrix}, \mathbf{V}\mathbf{1}_r\leq\mathbf{1}_n$$

- NMF with volume regularizer

$$\underset{\{\mathbf{U},\mathbf{V}\}\geq\mathbf{0}}{\operatorname{argmin}} f(\mathbf{U},\mathbf{V};\mathbf{M}) + \lambda_U\mathcal{V}(\mathbf{U})$$

- NMF under general separability constraint (Pan & Gillis, 2019)
- NMF in polynomial basis (Otto, Barel, Lathauwer, 2017), (Hautecoeur & Glineur, 2019)

Outline

NMF is just very hard

- NMF is NP-Hard (Jiang & Ravikumar, 1993), (Vavasis, 2007)
- NF ($\mathbf{M}$ is any matrix) is NP-Hard (Gillis & Glineur, 2008)
- There is an algorithm for the (exact) NMF that runs in time $\mathcal{O}((nm)^{r^2 2^r})$ (Arora, et al., 2012)
- NMF is NP-Hard for Boolean matrices (Shitov, 2016)
- NMF with matrix in $\mathbb{Q}$ requires irrationality (Chistikov, et al., 2017)
- Non-negative rank of a matrix is NP-hard to compute (Shitov, 2017)
- Heavy use of Graph-theoretic arguments, simplex geometry, and more
- Open problems :
  - Exact complexity of the nested polytope problem
  - On (bounded) estimation of the non-negative rank
  - and much more

NMF is just very hard

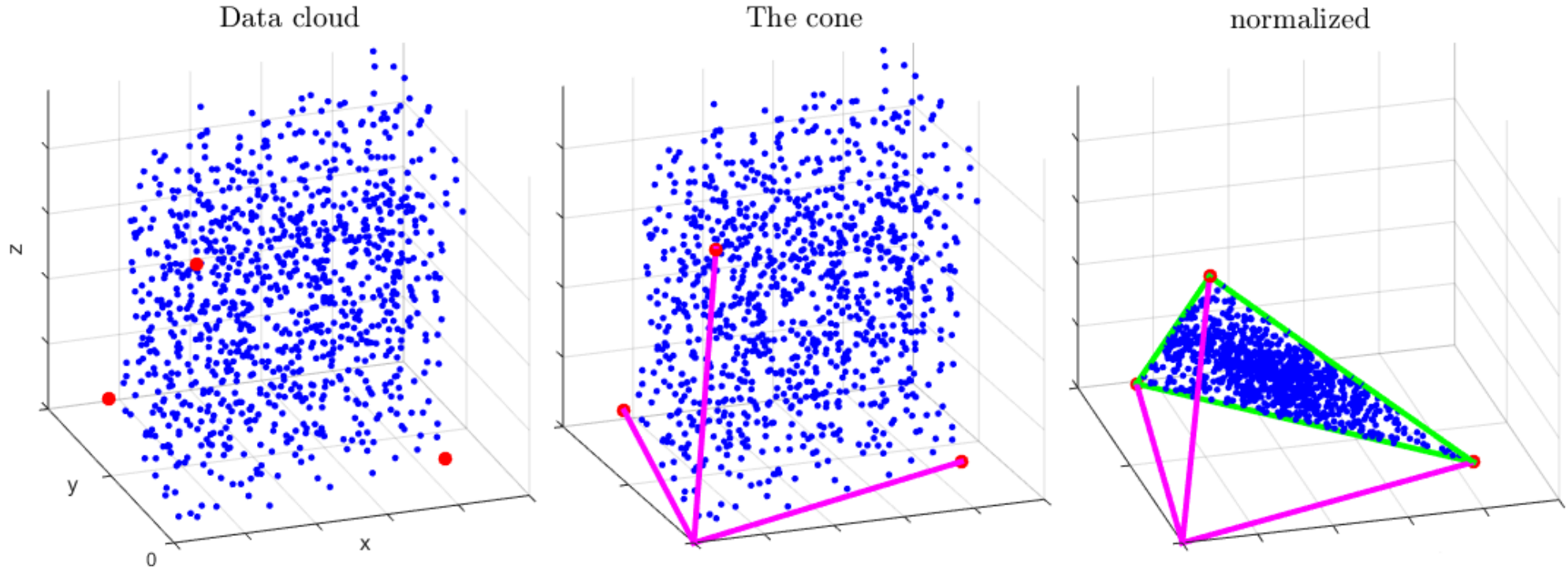
- NMF is NP-Hard (Jiang & Ravikumar, 1993), (Vavasis, 2007)
- NF ($\mathbf{M}$ is any matrix) is NP-Hard (Gillis & Glineur, 2008)
- There is an algorithm for the (exact) NMF that runs in time $\mathcal{O}((nm)^{r^2 2^r})$ (Arora, et al., 2012)
- NMF is NP-Hard for Boolean matrices (Shitov, 2016)
- NMF with matrix in $\mathbb{Q}$ requires irrationality (Chistikov, et al., 2017)
- Non-negative rank of a matrix is NP-hard to compute (Shitov, 2017)
- Heavy use of Graph-theoretic arguments, simplex geometry, and more
- Open problems :
  - Exact complexity of the nested polytope problem
  - On (bounded) estimation of the non-negative rank
  - and much more

NMF NP-Hard.

$\Longrightarrow$ add conditions on the NMF model to make it not NP-hard !

NMF tells a picture of a cone/hull

Given $\mathbf{M}$, the NMF $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$ tells a picture of a non-negative simplicial† convex cone.

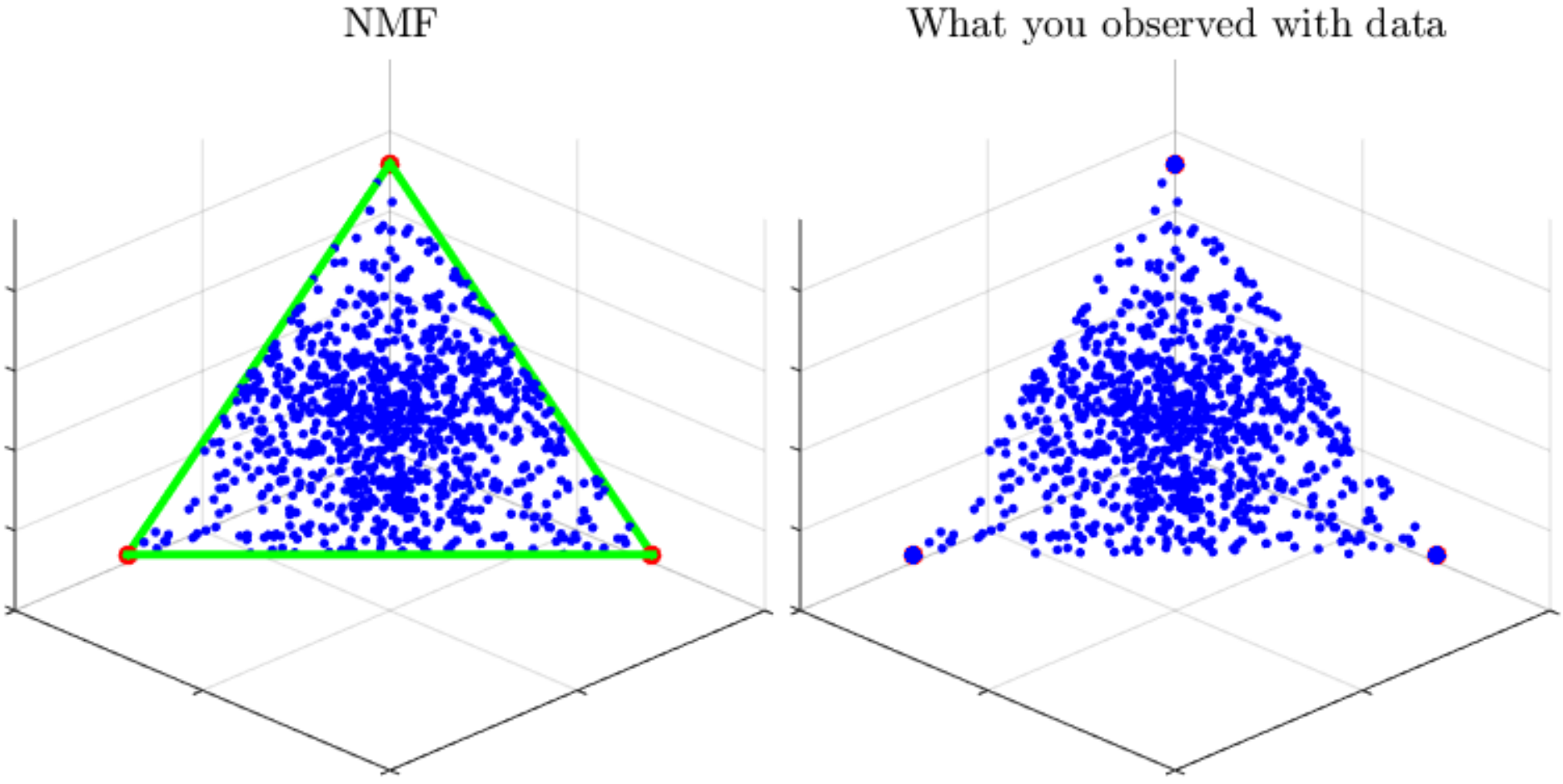


If the rows of $\mathbf{V}$ (columns of $\mathbf{V}^\top$) are normalized as sum-to-1, the cone compressed into a convex hull.

†Assumes $\mathbf{U}$ full rank.

NMF tells a picture of a cone/hull

For $r = 3$, facing the hull we see a triangle.



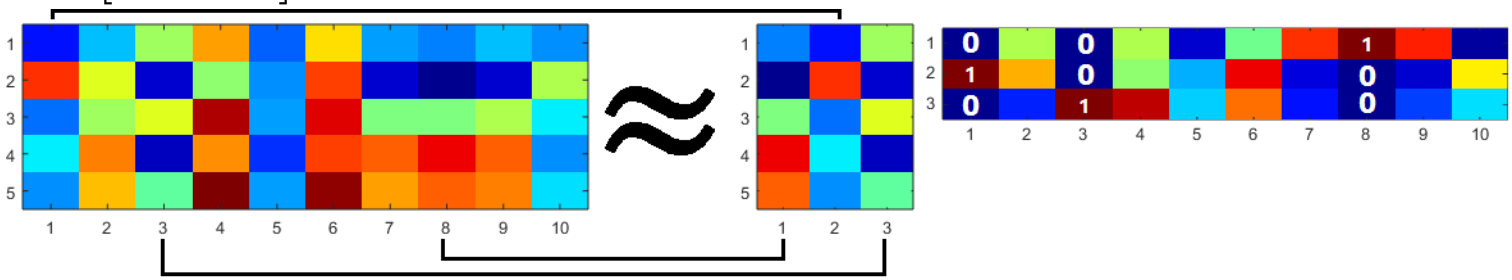
NMF$_{\mathbf{V}\text{ sum-to-1}}$ problem geometrically means "find the vertices".

Separable NMF

- Algebra : $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$
  - $\mathbf{U} = \mathbf{M}(:, \mathcal{J})$, $\mathcal{J}$ index set

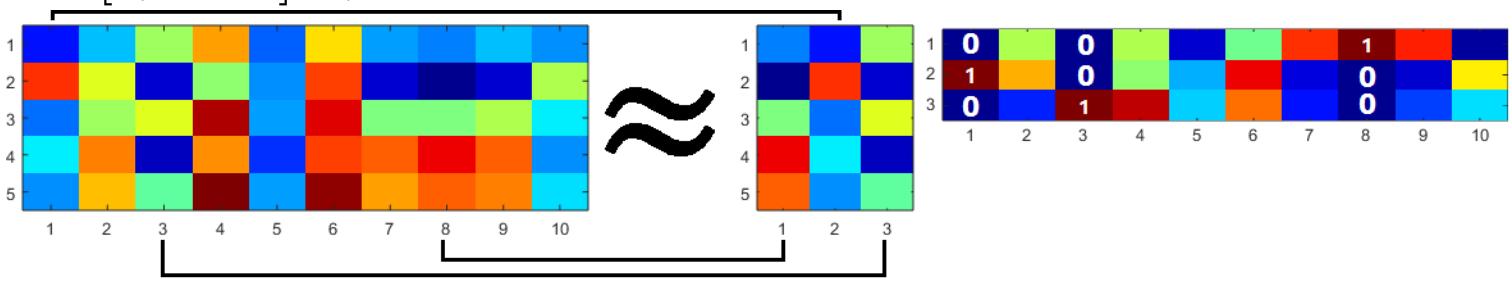
Separable NMF

- Algebra : $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$
  - $\mathbf{U} = \mathbf{M}(:, \mathcal{J})$, $\mathcal{J}$ index set
  - $\mathbf{V}^\top = [\mathbf{I}_r \ \mathbf{V}'^\top]\mathbf{\Pi}_r$, cols of $\mathbf{V}'^\top$ sum-to-1.



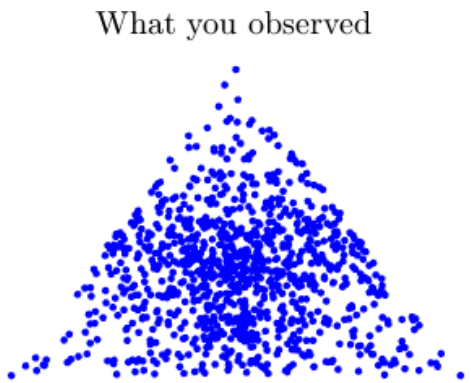
Separable NMF

- Algebra : $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$
  - $\mathbf{U} = \mathbf{M}(:, \mathcal{J})$, $\mathcal{J}$ index set
  - $\mathbf{V}^\top = [\mathbf{I}_r \ \mathbf{V}'^\top]\mathbf{\Pi}_r$, cols of $\mathbf{V}'^\top$ sum-to-1.



- Geometry : Find $\mathbf{U}$ $\iff$ find vertices from data cloud

$\mathbf{M}$ (pts) are cvx combination (described by $\mathbf{V}$) of vertices ($\mathbf{U}$).



- Not NP-hard anymore, solvable. Algorithm : LP, SPA, X-ray, ...
- Separability condition (Donoho-Stodden, 2004) – already proposed in 90s in HSI community, known as "pure pixel"
  Other names : anchord words, extreme ray enumeration

Successive Projection Algorithm (SPA)

One of the "best" method for (near-) Separable NMF :

Successive Projection Algorithm (SPA)

One of the "best" method for (near-) Separable NMF :

- **Robust**
  - It can find the vertices under bounded additive noise.
  - **Theorem** (Gillis-Vavasis, 2014)

    If $\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\mathbf{U}}^{\min}}{\sqrt{r}\kappa_{\mathbf{U}}^2}\right)$, SPA satisfies

    $$\max_k \left\|\mathbf{U}(:,k) - \mathbf{M}(:,\mathcal{J}(k))\right\| \leq \mathcal{O}(\epsilon\kappa_{\mathbf{U}}^2).$$

    In English : if noise is bounded, then the worse case fitting error is bounded.

Successive Projection Algorithm (SPA)

One of the "best" method for (near-) Separable NMF :

- **Robust**
  - It can find the vertices under bounded additive noise.
  - **Theorem** (Gillis-Vavasis, 2014)

    If $\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\mathbf{U}}^{\min}}{\sqrt{r}\kappa_{\mathbf{U}}^2}\right)$, SPA satisfies

    $$\max_k \left\|\mathbf{U}(:,k) - \mathbf{M}(:,\mathcal{J}(k))\right\| \leq \mathcal{O}(\epsilon\kappa_{\mathbf{U}}^2).$$

    In English : if noise is bounded, then the worse case fitting error is bounded.
- **Fast**
  - Computing $\mathbf{U}$: a modified Gram-Schmidt with column pivoting
  - Computing $\mathbf{V}$: a 1st-order method with Nesterov's acceleration
- Not many methods† achieve **both** robustness and speed

However,

Successive Projection Algorithm (SPA)

One of the "best" method for (near-) Separable NMF :

- **Robust**
  - It can find the vertices under bounded additive noise.
  - **Theorem** (Gillis-Vavasis, 2014)

    If $\epsilon \leq \mathcal{O}\left(\frac{\sigma_{\mathbf{U}}^{\min}}{\sqrt{r}\kappa_{\mathbf{U}}^2}\right)$, SPA satisfies

    $$\max_k \left\|\mathbf{U}(:,k) - \mathbf{M}(:,\mathcal{J}(k))\right\| \leq \mathcal{O}(\epsilon\kappa_{\mathbf{U}}^2).$$

    In English : if noise is bounded, then the worse case fitting error is bounded.
- **Fast**
  - Computing $\mathbf{U}$: a modified Gram-Schmidt with column pivoting
  - Computing $\mathbf{V}$: a 1st-order method with Nesterov's acceleration
- Not many methods† achieve **both** robustness and speed

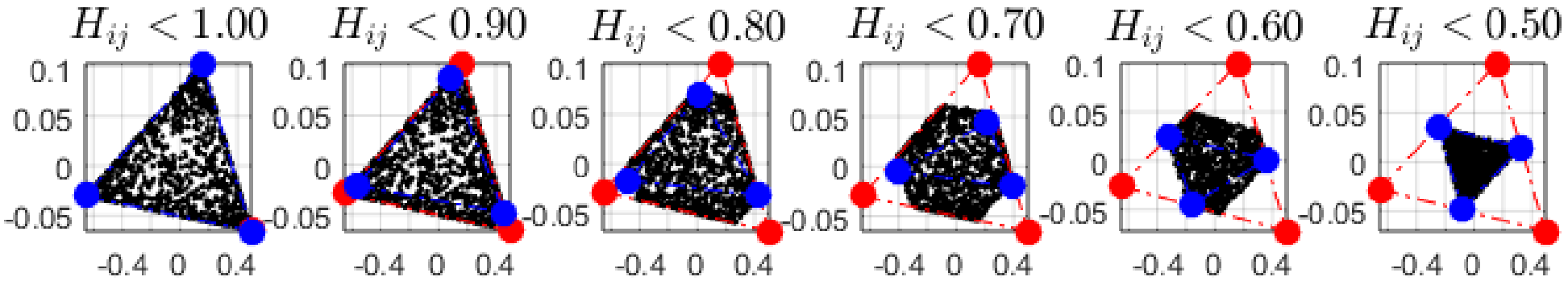
However, SPA assumes separability :

$$\mathbf{U} = \mathbf{M}(:,\mathcal{J}) : \text{Vertices } \mathbf{U} \text{ are } \textit{presented} \text{ in observed data } \mathbf{M}$$

What if this is false?

†Two examples : SNPA and preconditioned SPA by Gillis et al.

SPA fails when no separability



Projected onto 2d. From my old slide : $\mathbf{H}$ here is the matrix $\mathbf{V}^\top$.

Why fail : SPA takes the col. of $\mathbf{M}$ with k-th largest norm (after projection) as the k-th col. of $\mathbf{U}$.

How to solve it : minimum volume hull fitting [gif]

Volume regularized NMF

- Idea : fit NMF with minimum volume

$$\underset{\{\mathbf{U},\mathbf{V}\}\geq\mathbf{0}}{\operatorname{argmin}} f(\mathbf{U},\mathbf{V};\mathbf{M}) + \lambda_U \mathcal{V}(\mathbf{U}),$$

where $\mathcal{V}(\cdot)$ is a prox function measures vol(cvx hull of $\mathbf{U}$):

Volume regularized NMF

- Idea : fit NMF with minimum volume

$$\underset{\{\mathbf{U},\mathbf{V}\}\geq\mathbf{0}}{\operatorname{argmin}} f(\mathbf{U},\mathbf{V};\mathbf{M}) + \lambda_U \mathcal{V}(\mathbf{U}),$$

  where $\mathcal{V}(\cdot)$ is a prox function measures vol(cvx hull of $\mathbf{U}$):
  - $\det(\mathbf{U}^\top\mathbf{U})$ — det of Gramian
  - $\log\det(\mathbf{U}^\top\mathbf{U}+\delta\mathbf{I}_r)$ — log-det of Gramian
  - $\prod_{i=1}^r / \sum_{i=1}^r \|\mathbf{u}_i\|_2^2$ — rectangular box
  - $\|\mathbf{U}\|_*$ — nuclear norm
- Many works in this directions :
  - (Lin, et al., 2015) – Sufficient Scatter Condition
  - (A. & Gillis, 2018-NL), (A. & Gillis. 2019-HSI) – comparison of $\mathcal{V}$
  - (Leplat, A., Gillis, 2019-UK) – logdet works for rank deficient case
  - (Leplat, Gillis, Siebert, A., 2019) – on audio blind source separation
  - (Leplat, Gillis, A., 2019) – identifiability on minimum volume

Minvol. Identifiability (Leplat, Gillis, A., 2019)

- Model

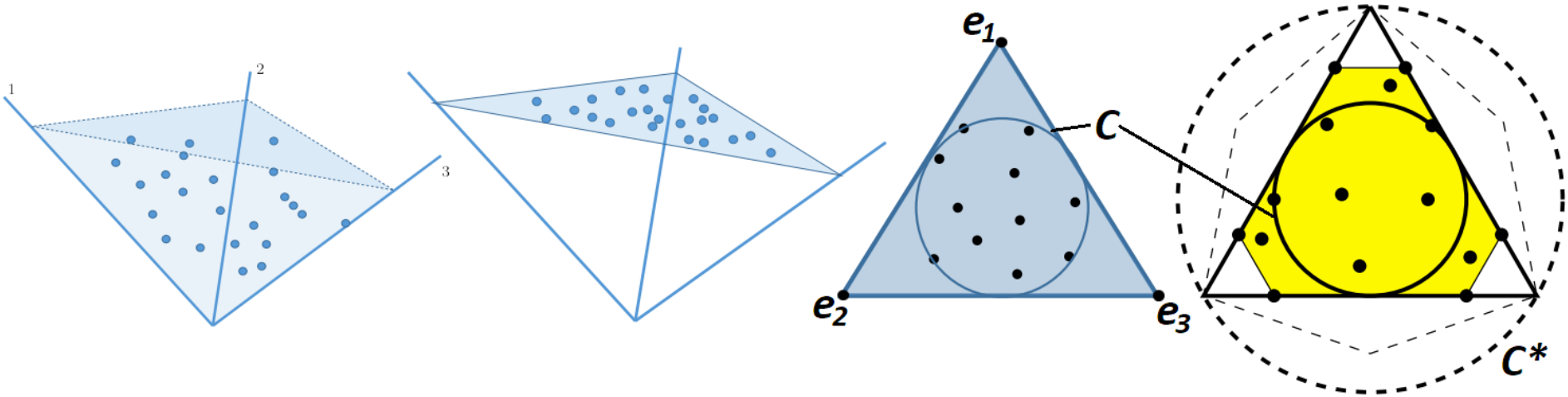
$$\begin{aligned} \underset{\mathbf{U},\mathbf{V}}{\operatorname{argmin}}\ \mathcal{V}(\mathbf{U}) \quad &:= \quad \log\det\left(\mathbf{U}^\top\mathbf{U}\right) \\ \text{subject to} \quad & \mathbf{U} \in \mathbb{R}_+^{m\times r}, \mathbf{V} \in \mathbb{R}_+^{n\times r} \\ & \mathbf{M} = \mathbf{U}\mathbf{V}^\top \\ & \mathbf{U}^\top\mathbf{1} = \mathbf{1} \end{aligned} \tag{1}$$

- Theorem : if $\mathbf{M} = \mathbf{U}_\#\mathbf{V}_\#^\top$, $\text{rank}(\mathbf{M}) = r$, $\mathbf{U}_\# \geq \mathbf{0}$ and $\mathbf{V}_\#^\top$ satisfies the sufficiently scattered condition :
  - $\mathcal{C} \subseteq \text{cone}(\mathbf{V}^\top)$, and
  - $\text{cone}(\mathbf{V}^{\top *}) \cap \text{bd}\mathbf{C}^* = \{\lambda\mathbf{e}_k | \lambda \geq 0, \forall k \in [1, 2, \ldots, r]\}$
  - where
    - $\mathcal{C} = \{\mathbf{x} | \mathbf{x}^\top\mathbf{1} \geq \sqrt{r-1}\|\mathbf{x}\|_2\}$
    - $\mathcal{C}^* = \{\mathbf{x} | \mathbf{x}^\top\mathbf{1} \geq \|x\|_2\}$
    - $\text{cone}(\mathbf{V}^\top) = \{\mathbf{x} | \mathbf{x} = \mathbf{V}^\top\theta\}$

  then the optimal solution of (1) recovers $(\mathbf{U}_\#, \mathbf{V}_\#)$ up to permutation and scaling.

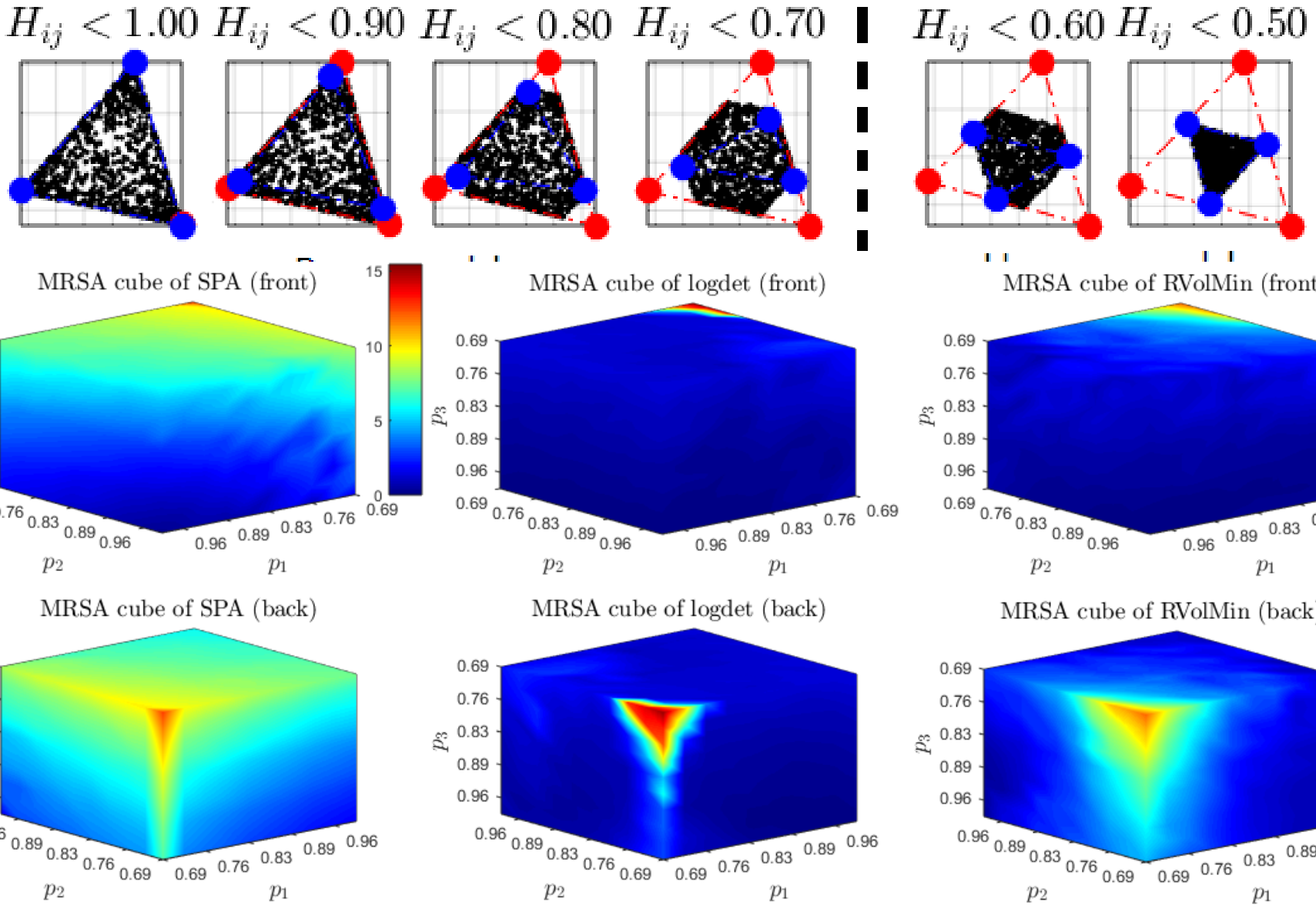
Sufficiently scattered condition

- From (Lin, et al., 2015), and a series of works from that group
- SSC is a generalization of the separability condition
- $\mathbf{V}^\top$ is SSC if
  - $\mathcal{C} \subseteq \text{cone}(\mathbf{V}^\top)$, and
  - $\text{cone}(\mathbf{V}^{\top *}) \cap \text{bd}\mathbf{C}^* = \{\lambda \mathbf{e}_k | \lambda \geq 0, \forall k \in [1, 2, \ldots, r]\}$
    - $\mathcal{C} = \{\mathbf{x} | \mathbf{x}^\top \mathbf{1} \geq \sqrt{r-1}\|\mathbf{x}\|_2\}$
    - $\mathcal{C}^* = \{\mathbf{x} | \mathbf{x}^\top \mathbf{1} \geq \|x\|_2\}$
    - $\text{cone}(\mathbf{V}^\top) = \{\mathbf{x} | \mathbf{x} = \mathbf{V}^\top \theta\}$



Modified from (Fu, et al., 2018).

Recoverable / Non-recoverable cases



From (A. & Gillis. 2019-HSI)

Outline

Block Coordinate Descent

$$(\mathcal{P}) \quad : \min_{\mathbf{U},\mathbf{V}} f(\mathbf{U},\mathbf{V}) = \|\mathbf{M} - \mathbf{U}\mathbf{V}^\top\|^2$$

Algorithm Block Coordinate Descent[1]

Input: $\mathbf{M} \in \mathbb{R}_+^{m\times n}$, $r \in \mathbb{N}$, initialization $\mathbf{U} \in \mathbb{R}_+^{m\times r}$, $\mathbf{V} \in \mathbb{R}_+^{r\times n}$

Output: $\mathbf{U}, \mathbf{V}$

1: **for** $k = 1, 2, \ldots$ **do**

2: $\quad \mathbf{U}^{k+1} = \underset{\mathbf{U}\geq 0}{\operatorname{argmin}} f(\mathbf{U}, \mathbf{V}^k)$, initialized at $\mathbf{U}^k$

3: $\quad \mathbf{V}^{k+1} = \underset{\mathbf{V}\geq 0}{\operatorname{argmin}} f(\mathbf{U}^{k+1}, \mathbf{V})$, initialized at $\mathbf{V}^k$

4: **end for**

We have non-increasing sequence

$$f(\mathbf{U}^{k+1}, \mathbf{V}^{k+1}) \leq f(\mathbf{U}^{k+1}, \mathbf{V}^k) \leq f(\mathbf{U}^k, \mathbf{V}^k).$$

(Actually not enough, need sufficient decrease condition)

[1] Other names : Gauss-Seidel iteration, alternating minimization (for 2 blocks)

Exact vs inexact BCD

- Exact BCD
  - $\mathbf{U}^{k+1} = \underset{\mathbf{U} \geq 0}{\operatorname{argmin}} f(\mathbf{U}, \mathbf{V}^k)$, initialized at $\mathbf{U}^k$
  - $\mathbf{V}^{k+1} = \underset{\mathbf{V} \geq 0}{\operatorname{argmin}} f(\mathbf{U}^{k+1}, \mathbf{V})$, initialized at $\mathbf{V}^k$
- Inexact BCD : e.g. alternating gradient update
  - $\mathbf{U}^{k+1} = \mathbf{U}^k - \gamma \nabla f(\mathbf{U}^k; \mathbf{V}^k, \mathbf{M})$
  - $\mathbf{V}^{k+1} = \mathbf{V}^k - \gamma \nabla f(\mathbf{V}^k; \mathbf{U}^{k+1}, \mathbf{M})$
- In terms of computation :
  (more-) exact BCD is better than (more-)inexact BCD

Exact BCD is computationally better

Suppose we use gradient descent on

$$(\mathcal{P}) \ : \min_{\mathbf{U},\mathbf{V}} f(\mathbf{U},\mathbf{V}) = \frac{1}{2}\|\mathbf{M} - \mathbf{U}\mathbf{V}^\top\|_F^2$$

Gradient updates (projection step hidden)

$$\begin{aligned} \mathbf{U}^{k+1} &= \mathbf{U} - \gamma_{\mathbf{U}}^k(\mathbf{M}\mathbf{V}^k - \mathbf{U}^k{\mathbf{V}^k}^\top\mathbf{V}^k) \\ \mathbf{V}^{k+1} &= \mathbf{V} - \gamma_{\mathbf{V}}^k(\mathbf{M}^\top\mathbf{U}^{k+1} - \mathbf{V}^k{\mathbf{U}^{k+1}}^\top\mathbf{U}^{k+1}) \end{aligned}$$

where γ is set to be the inverse Lipschitz constant of ∇f

$$\gamma_{\mathbf{U}}^k = \frac{1}{\|{\mathbf{V}^k}^\top\mathbf{V}^k\|_2}, \qquad \gamma_{\mathbf{V}}^k = \frac{1}{\|{\mathbf{U}^k}^\top\mathbf{U}^k\|_2}$$

Exact BCD is computationally better

Suppose we use gradient descent on

$$(\mathcal{P}) \quad : \min_{\mathbf{U},\mathbf{V}} f(\mathbf{U},\mathbf{V}) = \frac{1}{2}\|\mathbf{M} - \mathbf{U}\mathbf{V}^\top\|_F^2$$

Algorithm Inexact BCD (Alternating gradient update)

1: **for** $k = 1, 2, \ldots$ **do**
2: $\quad \mathbf{U}^{k+1} = \mathbf{U}^k - \gamma(\mathbf{M}\mathbf{V}^k - \mathbf{U}^k {\mathbf{V}^k}^\top \mathbf{V}^k)$
3: $\quad \mathbf{V}^{k+1} = \mathbf{V}^k - \gamma(\mathbf{M}^\top \mathbf{U}^{k+1} - \mathbf{V}^k {\mathbf{U}^{k+1}}^\top \mathbf{U}^{k+1})$
4: **end for**

Exact BCD is computationally better

Suppose we use gradient descent on

$$(\mathcal{P}) \quad : \min_{\mathbf{U},\mathbf{V}} f(\mathbf{U},\mathbf{V}) = \frac{1}{2}\|\mathbf{M} - \mathbf{U}\mathbf{V}^\top\|_F^2$$

Algorithm Inexact BCD (Alternating gradient update)

1: **for** $k = 1, 2, \ldots$ **do**
2: $\quad \mathbf{U}^{k+1} = \mathbf{U}^k - \gamma(\mathbf{M}\mathbf{V}^k - \mathbf{U}^k {\mathbf{V}^k}^\top \mathbf{V}^k)$
3: $\quad \mathbf{V}^{k+1} = \mathbf{V}^k - \gamma(\mathbf{M}^\top \mathbf{U}^{k+1} - \mathbf{V}^k {\mathbf{U}^{k+1}}^\top \mathbf{U}^{k+1})$
4: **end for**

Algorithm Exact BCD (Alternating repeated gradient update)

1: **for** $k = 1, 2, \ldots$ **do**
2: $\quad$ **for** $j = 1, 2, \ldots$ until converge
3: $\quad\quad \mathbf{U}^{k+1} = \mathbf{U}^k - \gamma(\mathbf{M}\mathbf{V}^k - \mathbf{U}^k {\mathbf{V}^k}^\top \mathbf{V}^k)$
4: $\quad$ **endfor**
5: $\quad$ **for** $j = 1, 2, \ldots$ until converge
6: $\quad\quad \mathbf{V}^{k+1} = \mathbf{V}^k - \gamma(\mathbf{M}^\top \mathbf{U}^{k+1} - \mathbf{V}^k {\mathbf{U}^{k+1}}^\top \mathbf{U}^{k+1})$
7: $\quad$ **endfor**
8: **end for**

Exact BCD is computationally better

Algorithm 5 Inexact BCD

1: $\mathbf{U}^1 = \mathbf{U}^0 - \frac{1}{\|\mathbf{V}^{0\top}\mathbf{V}^0\|_2}(\mathbf{M}\mathbf{V}^0 - \mathbf{U}^0\mathbf{V}^{0\top}\mathbf{V}^0)$

2: $\mathbf{V}^1 = \mathbf{V}^0 - \frac{1}{\|\mathbf{U}^{1\top}\mathbf{U}^1\|_2}(\mathbf{M}^\top\mathbf{U}^1 - \mathbf{V}^0\mathbf{U}^{1\top}\mathbf{U}^1)$

3: $\mathbf{U}^2 = \mathbf{U}^1 - \frac{1}{\|\mathbf{V}^{1\top}\mathbf{V}^1\|_2}(\mathbf{M}\mathbf{V}^1 - \mathbf{U}^1\mathbf{V}^{1\top}\mathbf{V}^1)$

4: $\mathbf{V}^2 = \mathbf{V}^1 - \frac{1}{\|\mathbf{U}^{2\top}\mathbf{U}^2\|_2}(\mathbf{M}^\top\mathbf{U}^2 - \mathbf{V}^1\mathbf{U}^{2\top}\mathbf{U}^2)$

5: $\mathbf{U}^3 = \mathbf{U}^2 - \frac{1}{\|\mathbf{V}^{2\top}\mathbf{V}^2\|_2}(\mathbf{M}\mathbf{V}^2 - \mathbf{U}^2\mathbf{V}^{2\top}\mathbf{V}^2)$

6: $\mathbf{V}^3 = \mathbf{V}^2 - \frac{1}{\|\mathbf{U}^{3\top}\mathbf{U}^3\|_2}(\mathbf{M}^\top\mathbf{U}^3 - \mathbf{V}^2\mathbf{U}^{3\top}\mathbf{U}^3)$

7: $\vdots$

Algorithm 6 Exact BCD

1: $\mathbf{U}^1 = \mathbf{U}^0 - \frac{1}{\|\mathbf{V}^{0\top}\mathbf{V}^0\|_2}(\mathbf{M}\mathbf{V}^0 - \mathbf{U}^0\mathbf{V}^{0\top}\mathbf{V}^0)$

2: $\mathbf{U}^2 = \mathbf{U}^1 - \frac{1}{\|\mathbf{V}^{0\top}\mathbf{V}^0\|_2}(\mathbf{M}\mathbf{V}^1 - \mathbf{U}^1\mathbf{V}^{0\top}\mathbf{V}^0)$

3: $\mathbf{U}^3 = \mathbf{U}^2 - \frac{1}{\|\mathbf{V}^{0\top}\mathbf{V}^0\|_2}(\mathbf{M}\mathbf{V}^2 - \mathbf{U}^2\mathbf{V}^{0\top}\mathbf{V}^0)$

4: $\mathbf{V}^1 = \mathbf{V}^0 - \frac{1}{\|\mathbf{U}^{3\top}\mathbf{U}^3\|_2}(\mathbf{M}^\top\mathbf{U}^1 - \mathbf{V}^0\mathbf{U}^{3\top}\mathbf{U}^3)$

5: $\mathbf{V}^2 = \mathbf{V}^1 - \frac{1}{\|\mathbf{U}^{3\top}\mathbf{U}^3\|_2}(\mathbf{M}^\top\mathbf{U}^2 - \mathbf{V}^1\mathbf{U}^{3\top}\mathbf{U}^3)$

6: $\mathbf{V}^3 = \mathbf{V}^2 - \frac{1}{\|\mathbf{U}^{3\top}\mathbf{U}^3\|_2}(\mathbf{M}^\top\mathbf{U}^3 - \mathbf{V}^2\mathbf{U}^{3\top}\mathbf{U}^3)$

7: $\vdots$

Computational costs :

$$\mathbf{V}^\top\mathbf{V}_{(2n-1)m^2},\ \mathbf{M}\mathbf{V}_{(2n-1)mr},\ \mathbf{U}^\top\mathbf{U}_{(2r-1)m^2},\ \mathbf{M}^\top\mathbf{U}_{(2m-1)rn}$$

pre-compute these terms and re-use several times in Exact BCD has big efficiency improvement

Extrapolated BCD : Heuristic Extrapolation with Restarts

$$(\mathcal{P}) \ : \min_{\mathbf{U},\mathbf{V}} f(\mathbf{U},\mathbf{V}) = \|\mathbf{M} - \mathbf{U}\mathbf{V}\|^2$$

Algorithm HER (A. & Gillis. 2019-acc)

Input: $\mathbf{M} \in \mathbb{R}_+^{m\times n}$, $r, \mathbf{U}, \mathbf{V}$, $\hat{\mathbf{U}} = \mathbf{U}, \hat{\mathbf{V}} = \mathbf{V}$
Output: $\mathbf{U}, \mathbf{V}$

1: **for** $k = 1, 2, \ldots$ **do**
2: $\quad \mathbf{U}^{k+1} = \underset{\mathbf{U}\geq 0}{\operatorname{argmin}} f(\mathbf{U}, \hat{\mathbf{V}}^k)$, initialized at $\mathbf{U}^k$
3: $\quad$ Extrapolate[$\mathbf{U}$] : $\hat{\mathbf{U}}^{k+1} = \mathbf{U}^{k+1} + \beta_k(\mathbf{U}^{k+1} - \mathbf{U}^k)$.
4: $\quad \mathbf{V}^{k+1} = \underset{\mathbf{V}\geq 0}{\operatorname{argmin}} f(\hat{\mathbf{U}}^{k+1}, \mathbf{V})$, initialized at $\mathbf{V}^k$.
5: $\quad$ Extrapolate[$\mathbf{V}$] : $\hat{\mathbf{V}}^{k+1} = \mathbf{V}^{k+1} + \beta_k(\mathbf{V}^{k+1} - \mathbf{V}^k)$.
6: $\quad$ Restarts (safe guard mechanism) if needed.
7: **end for**

- Extrapolation may destroy the non-increasing sequence property
- Instead of 2-3-4-5, can do 2-4-3-5
- How to do 6

Algorithm HER (A. & Gillis. 2019-acc)

Input: $\mathbf{M} \in \mathbb{R}_+^{m \times n}$, $r, \mathbf{U}, \mathbf{V}, \hat{\mathbf{U}}, \hat{\mathbf{V}}$,
$\beta_0 \in [0, 1]$, $\gamma \geq 1$, $\bar{\gamma} \geq 1, \eta \geq 1, \bar{\beta}_0 = 1$

1: **for** $k = 1, 2, \ldots$ **do**
2: $\mathbf{U}^{k+1} = \underset{\mathbf{U} \geq 0}{\operatorname{argmin}} f(\mathbf{U}, \hat{\mathbf{V}}^k)$, initialized at $\mathbf{U}^k$
3: $\hat{\mathbf{U}}^{k+1} = \mathbf{U}^{k+1} + \beta_k(\mathbf{U}^{k+1} - \mathbf{U}^k)$
4: $\mathbf{V}^{k+1} = \underset{\mathbf{V} \geq 0}{\operatorname{argmin}} f(\hat{\mathbf{U}}^{k+1}, \mathbf{V})$, initialized at $\mathbf{V}^k$.
5: $\hat{\mathbf{V}}^{k+1} = \mathbf{V}^{k+1} + \beta_k(\mathbf{V}^{k+1} - \mathbf{V}^k)$
6: $\hat{e}^{k+1} = f(\hat{\mathbf{U}}^{k+1}, \mathbf{V}^{k+1})$
7: **if** $\hat{e}^{k+1} > \hat{e}^k$
Restarts
Decay β_k, update $\bar{\beta}_k$
8: **else** $(\hat{e}^{k+1} \leq \hat{e}^k)$
Grow β_k, update $\bar{\beta}_k$
9: **end if**
10: **end for**

Important: the argmin is not really necessary. i.e. It can be inexact BCD.

Facts

- NMF is non-cvx problem
- Direct application of Nesterov's β gives erratic convergence behaviour Mitchell, et al. "Nesterov Acceleration of Alternating Least Squares for Canonical Tensor Decomposition." arXiv:1810.05846 (2018)

Why heuristics?

- Non-cvx problem is hard :0)
- No better idea :0)
- Currently no convergence analysis (even for NNLS)

What's good ?

- Just a parameter tuning
- Easy implementation
- extension to other models : exact / inexact BCD
- (Empirical) Improvement on convergence speed

Computing β by restarts

With $\bar{\beta}_1 = 1$ and $\beta_0 \in [0, 1]$, update β as

$$\beta_{k+1} = \begin{cases} \min\{\gamma\beta_k, \bar{\beta}\} & \text{if } \hat{e}^k \leq \hat{e}^{k-1} \\ \dfrac{\beta_k}{\eta} & \text{if } \hat{e}^k > \hat{e}^{k-1} \end{cases}$$

Also update $\bar{\beta}_k$

$$\bar{\beta}_{k+1} = \begin{cases} \min\{\bar{\gamma}\bar{\beta}_k, 1\} & \text{if } \hat{e}^k \leq \hat{e}^{k-1} \text{ and } \bar{\beta}_k < 1 \\ \beta_k & \text{if } \hat{e}^k > \hat{e}^{k-1} \end{cases}.$$

The idea of updating β_k

Case 1. "error" decreases : $\hat{e}^k \leq \hat{e}^{k-1}$

- Means the current β is "good"

The idea of updating β_k

Case 1. "error" decreases : $\hat{e}^k \leq \hat{e}^{k-1}$

- Means the current β is "good"
- Be more ambitious on next extrapolation
  - i.e., make β larger
  - How : multiplying it with a growth factor $\gamma > 1$

$$\beta_{k+1} = \beta_k \gamma$$

The idea of updating β_k

Case 1. "error" decreases : $\hat{e}^k \leq \hat{e}^{k-1}$

- Means the current β is "good"
- Be more ambitious on next extrapolation
  - i.e., make β larger
  - How : multiplying it with a growth factor $\gamma > 1$

$$\beta_{k+1} = \beta_k \gamma$$

- Growth of β cannot be indefinite : put a ceiling
  - How :

$$\beta_{k+1} = \min\{\beta_k \gamma, \bar{\beta}_k\}$$

  - $\bar{\beta}$ is also updated with a growth factor $\bar{\gamma}$ with ceiling 1

The idea of updating β_k

Case 2. "error" increases : $\hat{e}^k > \hat{e}^{k-1}$

- Means the current β value is "bad" (too large)

The idea of updating β_k

Case 2. "error" increases : $\hat{e}^k > \hat{e}^{k-1}$

- Means the current β value is "bad" (too large)
- Be less ambitious on the next extrapolation
  - i.e., make β smaller
  - How : divide it with a decay factor $\eta > 1$

$$\beta_{k+1} = \frac{\beta_k}{\eta}$$

The idea of updating β_k

Case 2. "error" increases : $\hat{e}^k > \hat{e}^{k-1}$

- Means the current β value is "bad" (too large)
- Be less ambitious on the next extrapolation
  - i.e., make β smaller
  - How : divide it with a decay factor $\eta > 1$

$$\beta_{k+1} = \frac{\beta_k}{\eta}$$

- As f is continuous and smooth, for β_k being too large, it "should also be" too large in the near future
  - i.e., have to avoid β_{k+1} to grow back to the "bad" β_k too soon
  - How : we set the ceiling parameter

$$\bar{\beta}_{k+1} = \beta_k$$

Variation of the HER algorithm

Together with

$$\beta_{k+1} = \begin{cases} \min\{\gamma\beta_k, \bar{\beta}\} & \text{if } \hat{e}^k \leq \hat{e}^{k-1} \\ \dfrac{\beta_k}{\eta} & \text{if } \hat{e}^k > \hat{e}^{k-1} \end{cases}$$

$$\bar{\beta}_{k+1} = \begin{cases} \min\{\bar{\gamma}\bar{\beta}_k, 1\} & \text{if } \hat{e}^k \leq \hat{e}^{k-1} \text{ and } \bar{\beta}_k < 1 \\ \beta_k & \text{if } \hat{e}^k > \hat{e}^{k-1} \end{cases}.$$

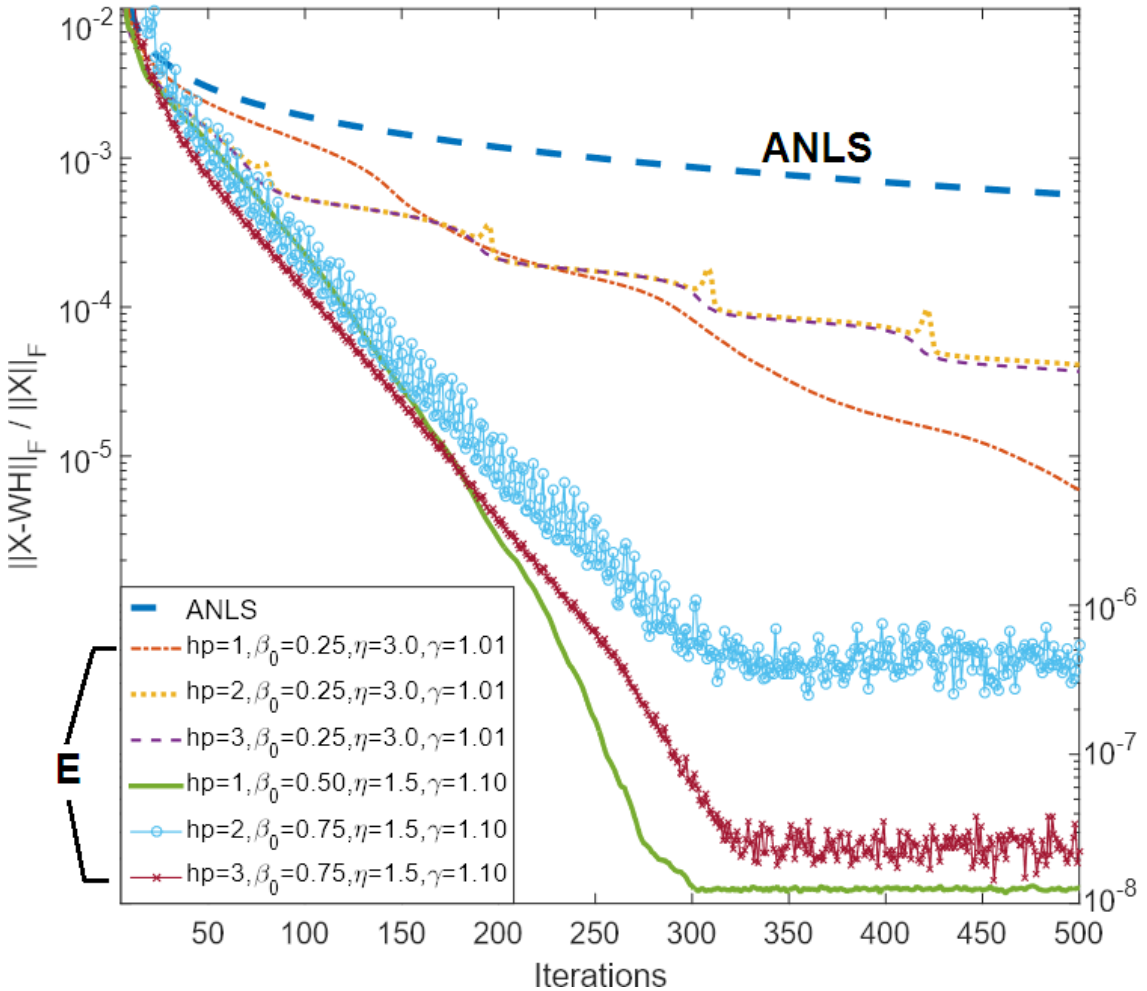
There are variations on the update-extrapolate chain :

- Update $\mathbf{U} \to$ extrapolate $\mathbf{U} \to$ update $\mathbf{V} \to$ extrapolate $\mathbf{V}$
- Update $\mathbf{U} \to$ extrapolate $\mathbf{U} \to$ project $\mathbf{U} \to$ update $\mathbf{V} \to$ extrapolate $\mathbf{V} \to$ project $\mathbf{V}$
- Update $\mathbf{U} \to$ update $\mathbf{V} \to$ extrapolate $\mathbf{U} \to$ extrapolate $\mathbf{V}$
- Update $\mathbf{U} \to$ update $\mathbf{V} \to$ extrapolate $\mathbf{U} \to$ extrapolate $\mathbf{V} \to$ project $\mathbf{U} \to$ project $\mathbf{V}$

Fancy graphs showing numerics

Effectiveness of HER

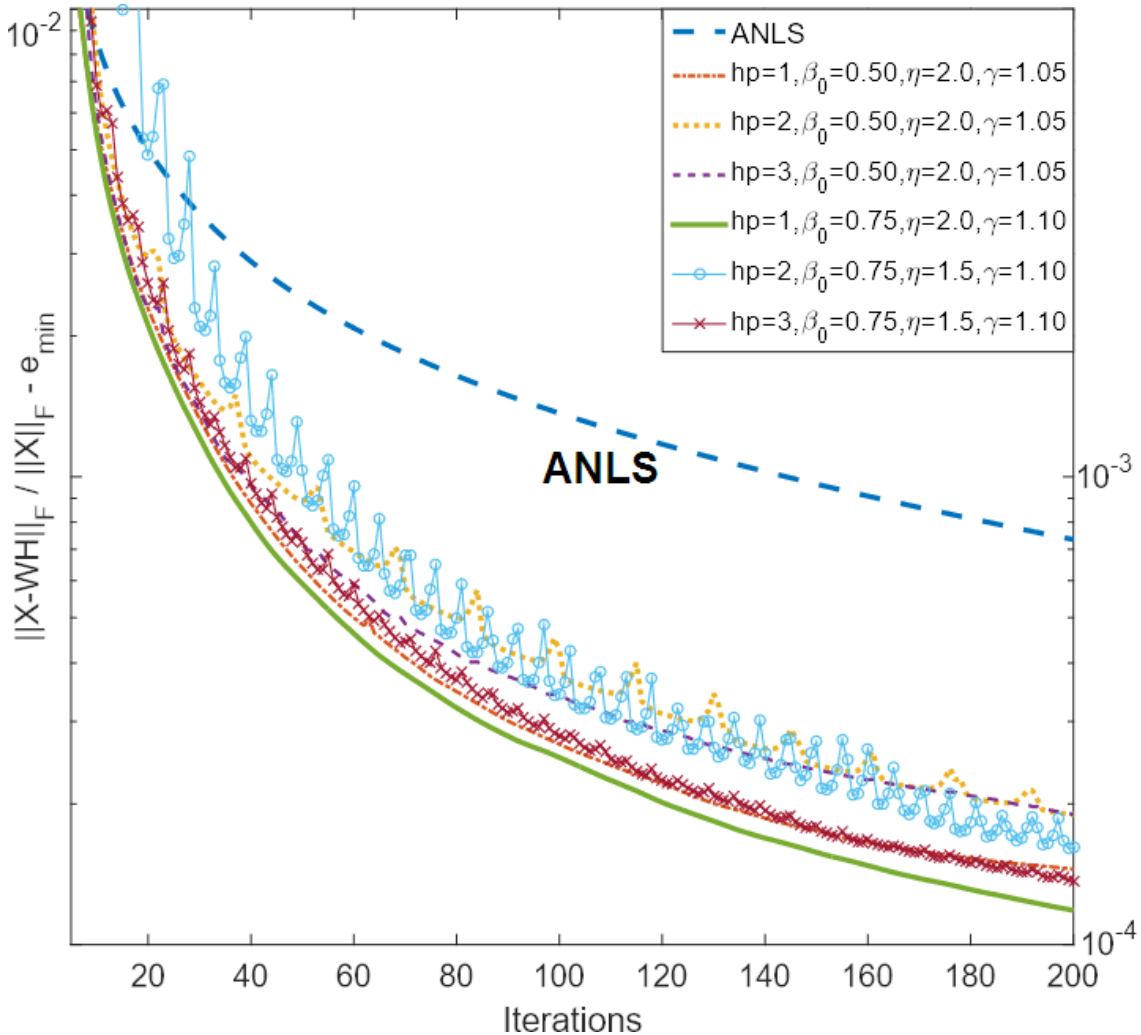
NMF literature use $(\mathbf{X}, \mathbf{W}, \mathbf{H})$ instead of $(\mathbf{M}, \mathbf{U}, \mathbf{V}^\top)$
Here the plots are using e (not $\hat{e}$)



Low-rank synthetic data, figure from (A. & Gillis. 2019-acc)

Effectiveness of HER

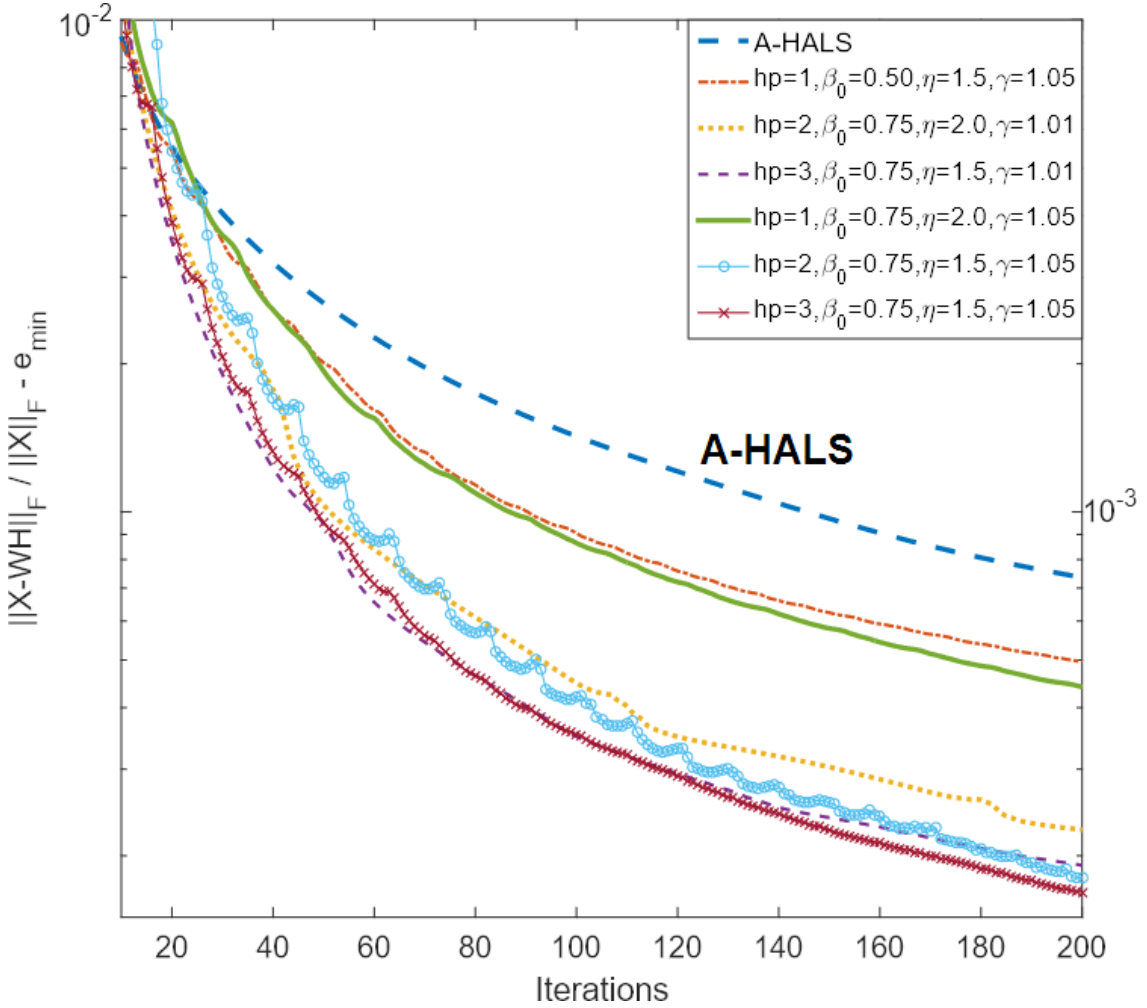
NMF literature use $(\mathbf{X}, \mathbf{W}, \mathbf{H})$ instead of $(\mathbf{M}, \mathbf{U}, \mathbf{V}^\top)$
Here the plots are using e (not $\hat{e}$)



Dense Image data, figure from (A. & Gillis. 2019-acc)

Effectiveness of HER

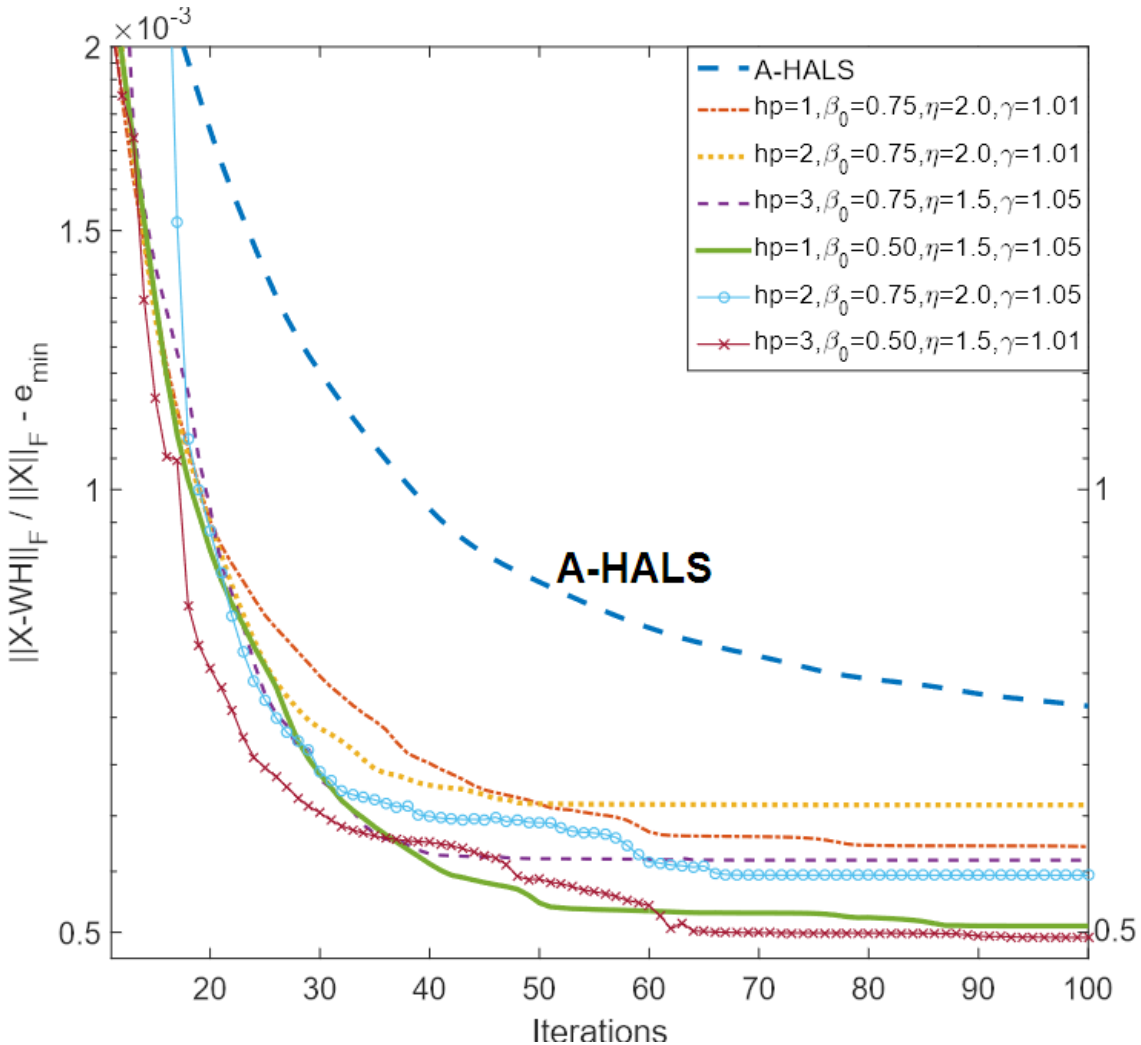
NMF literature use $(\mathbf{X}, \mathbf{W}, \mathbf{H})$ instead of $(\mathbf{M}, \mathbf{U}, \mathbf{V}^\top)$
Here the plots are using e (not $\hat{e}$)



Dense Image data, figure from (A. & Gillis. 2019-acc)

Effectiveness of HER

NMF literature use $(\mathbf{X}, \mathbf{W}, \mathbf{H})$ instead of $(\mathbf{M}, \mathbf{U}, \mathbf{V}^\top)$
Here the plots are using e (not $\hat{e}$)



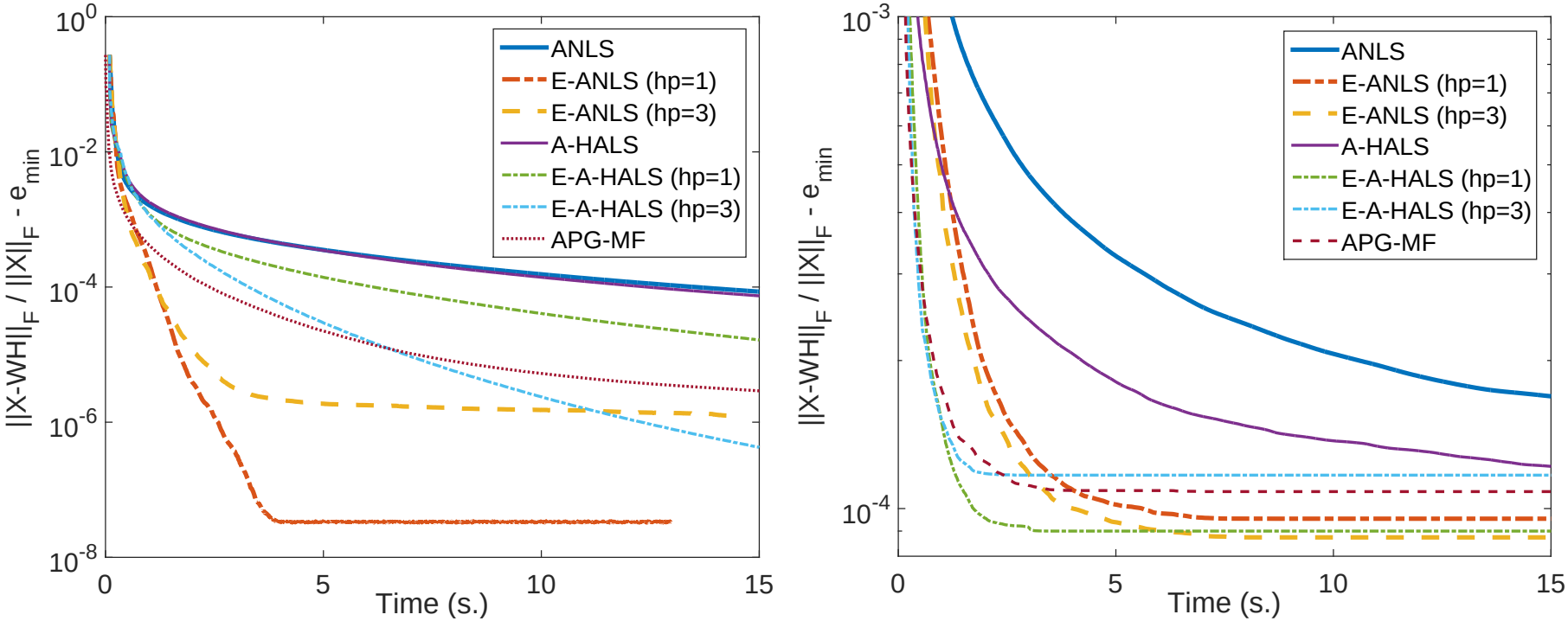
Sparse Text data, figure from (A. & Gillis. 2019-acc)

Effectiveness of HER

NMF literature use $(\mathbf{X}, \mathbf{W}, \mathbf{H})$ instead of $(\mathbf{M}, \mathbf{U}, \mathbf{V}^\top)$

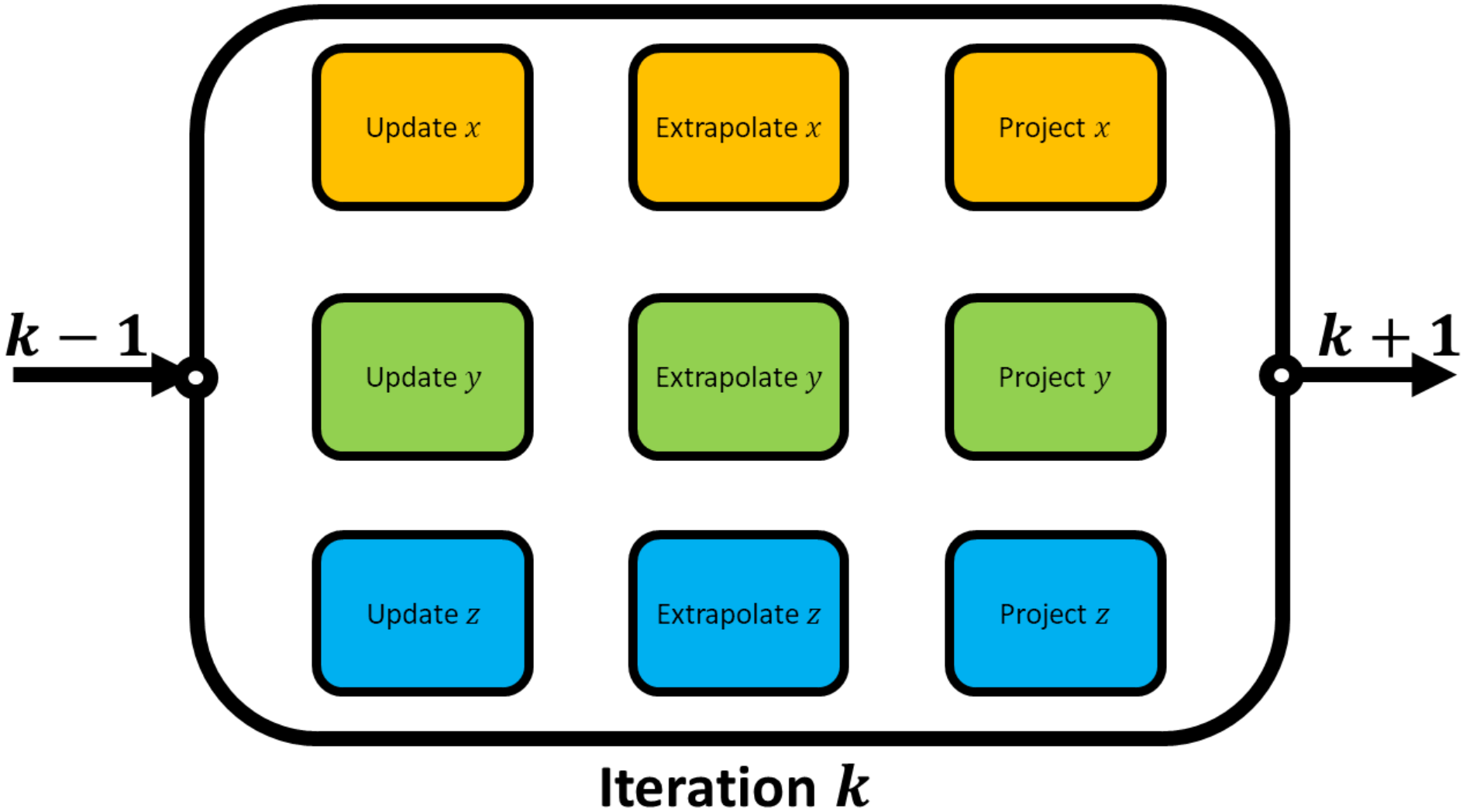
Here the plots are using e (not $\hat{e}$)

Compare to extrapolated alternating gradient (no inner loop) : APG-MF of (Xu-Yin, 2013) A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM J. Img Sci.



For more details see (A., Gillis, 2019-acc)

The $hp = 1, 2, 3$ means different chain structure :



For empirical comparisons of different chain structures, see (A. & Gillis, 2019-acc)

Why $\hat{e}^k = \|\mathbf{M} - \hat{\mathbf{U}}\mathbf{V}\|$ not $e^k = \|\mathbf{M} - \mathbf{U}\mathbf{V}\|$

- $\hat{\mathbf{V}}$ is updated according to $\hat{\mathbf{U}}$
- It gives the algorithm some degrees of freedom to possibly increase the objective function (a vague statement)
- Computationally cheaper (main reason)
  Compute $\|\mathbf{M} - \mathbf{U}\mathbf{V}\|_F$ cost $O(mnr)$ instead of $O(mr^2)$ by re-using previous computed terms :

$$\|\mathbf{M} - \mathbf{U}\mathbf{V}\|_F^2 = \|\mathbf{M}\|_F^2 - 2\left\langle \mathbf{U}, \mathbf{M}\mathbf{V}^\top \right\rangle + \left\langle \mathbf{U}^\top\mathbf{U}, \mathbf{V}\mathbf{V}^\top \right\rangle$$

  Significant if $r \ll n$, which is true in low-rank model.
  Says $r = 5 \sim 50$ with $n = 10^3 \sim 10^6$ or more.
- (If converge) In the long run, $\mathbf{U}$, $\hat{\mathbf{U}}$ is effectively the same :
  $\mathbf{U}^\infty = \hat{\mathbf{U}}^\infty$ after projection

Why $\hat{e}^k = \|\mathbf{M} - \hat{\mathbf{U}}\mathbf{V}\|$ not $e^k = \|\mathbf{M} - \mathbf{U}\mathbf{V}\|$

It gives the algorithm some degrees of freedom to possibly increase the objective function (a vague statement)

By definition of the algorithm, we have

- $\hat{\mathbf{U}}^{k+1} = \mathbf{U}^{k+1} + \beta^{k+1}(\mathbf{U}^{k+1} - \mathbf{U}^k)$
- $\mathbf{V}^{k+1} = \underset{\mathbf{V}\geq 0}{\operatorname{argmin}} f(\hat{\mathbf{U}}^{k+1}, \mathbf{V})$

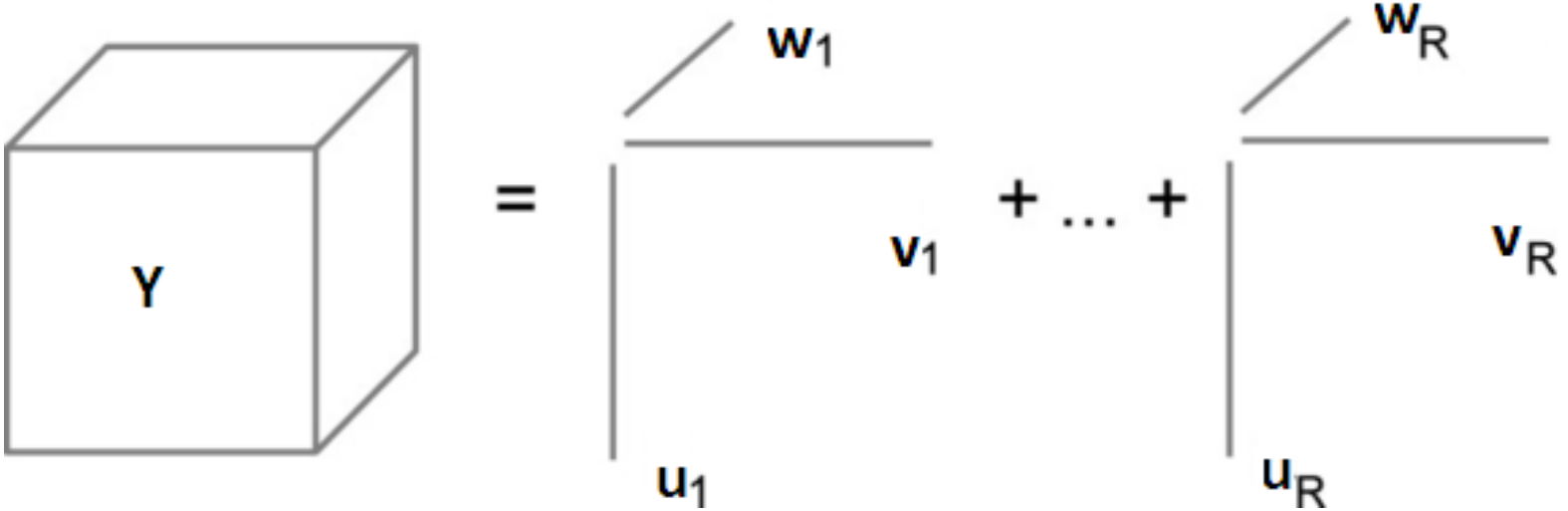
$$
\begin{aligned}
\underbrace{\|\mathbf{M} - \hat{\mathbf{U}}^{k+1}\mathbf{V}^{k+1}\|}_{\hat{e}^{k+1}} &= \|\mathbf{M} - \mathbf{U}^{k+1}\mathbf{V}^{k+1} + \beta^{k+1}(\mathbf{U}^k - \mathbf{U}^{k+1})\mathbf{V}^{k+1}\| \\
&\overset{\Delta IQ.}{\leq} \underbrace{\|\mathbf{M} - \mathbf{U}^{k+1}\mathbf{V}^{k+1}\|}_{e^{k+1}} + \beta^{k+1}\|(\mathbf{U}^k - \mathbf{U}^{k+1})\mathbf{V}^{k+1}\| \\
\hat{e}^{k+1} &\leq e^{k+1} + \beta^{k+1}\underbrace{\|\mathbf{U}^k - \mathbf{U}^{k+1}\|}_{\searrow 0 \text{ if converge}}\|\mathbf{V}^{k+1}\|
\end{aligned}
$$

Non-negative Canonical Polyadic Decomposition

Joint-work with Jeremy Cohen of IRISA, France, (A., Cohen, Gillis, 2019)

For example, order-3 tensor :

$$\begin{aligned} \underset{\mathbf{U}\geq 0,\mathbf{V}\geq 0,\mathbf{W}\geq 0}{\operatorname{argmin}} \quad f(\mathbf{U},\mathbf{V},\mathbf{W}) &= \|\mathcal{Y} - \mathbf{U} * \mathbf{V} * \mathbf{W}\| \\ &= \|\mathcal{Y} - \sum_{i}^{r} \mathbf{u}_i * \mathbf{v}_i * \mathbf{w}_i\| \end{aligned}$$



(Not on Tucker Model in this talk.)

$$(\mathcal{P}) \ : \min_{\{\mathbf{U},\mathbf{V},\mathbf{W}\}\geq 0} f(\mathbf{U},\mathbf{V},\mathbf{W}) = \|\mathcal{Y} - \mathbf{U} * \mathbf{V} * \mathbf{W}\|_F^2$$

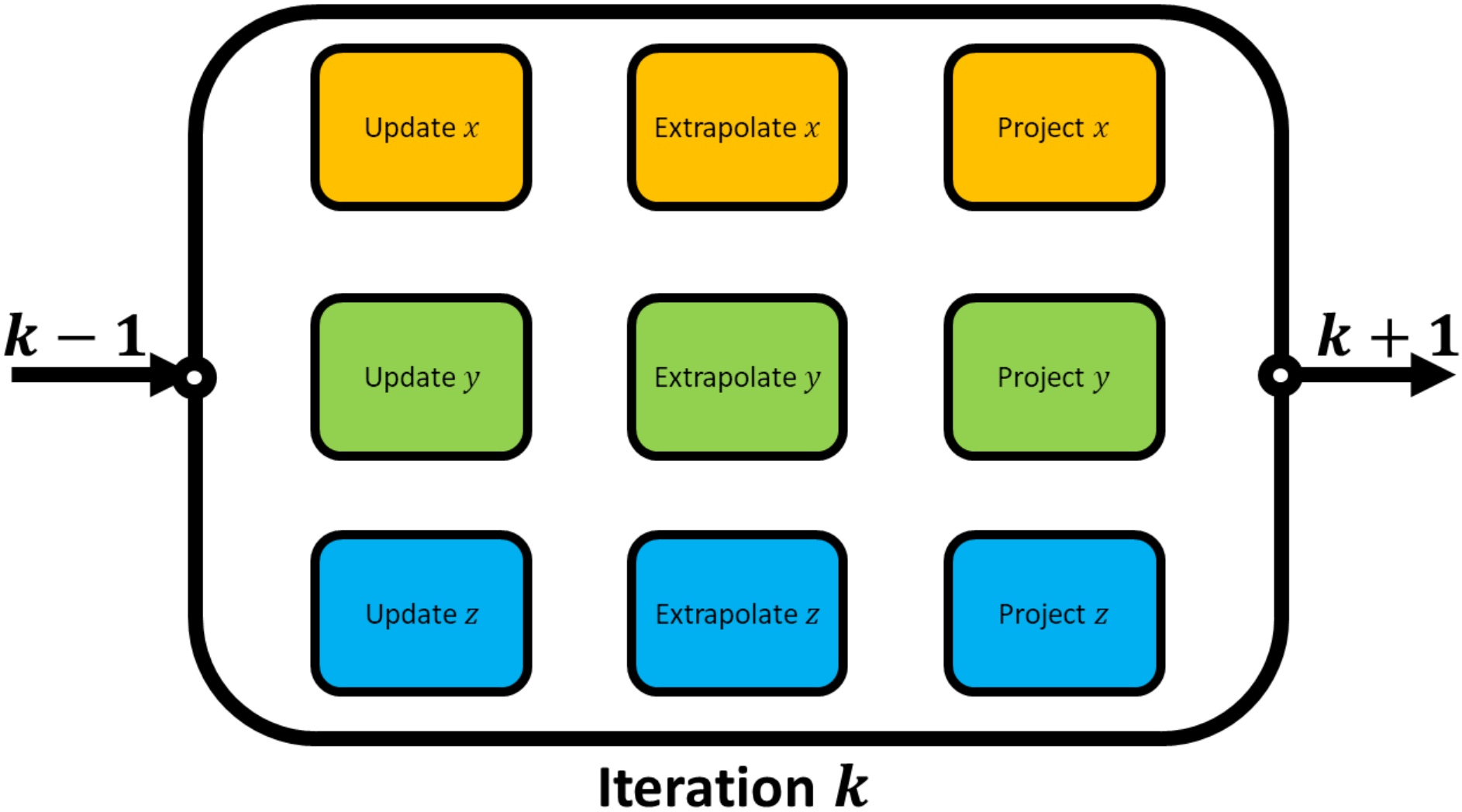
Algorithm HER

Input: $\mathcal{Y} \in \mathbb{R}_+^{I\times J\times K}$, r, $\mathbf{U}, \mathbf{V}, \mathbf{W}$, $\hat{\mathbf{U}} = \mathbf{U}, \hat{\mathbf{V}} = \mathbf{V}, \hat{\mathbf{W}} = \mathbf{W}$
Output: $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$

1: **for** $k = 1, 2, \ldots$ **do**
2: **for** $\mathbf{U}, \mathbf{V}, \mathbf{W}$ **do**
3: $\mathbf{U}^{k+1} = \underset{\mathbf{U}\geq 0}{\operatorname{argmin}} f(\mathbf{U}, \hat{\mathbf{V}}^k, \hat{\mathbf{W}}^k)$
4: $\hat{\mathbf{U}}^{k+1} = \mathbf{U}^{k+1} + \beta_k(\mathbf{U}^{k+1} - \mathbf{U}^k)$.
5: **end for**
6: $\hat{e}^{k+1} = f(\hat{\mathbf{U}}^{k+1}, \hat{\mathbf{V}}^{k+1}, \mathbf{W}^{k+1})$
7: Update $\beta_k, \bar{\beta}_k$ and restarts (if needed)
8: **end for**

- $\hat{e}^k$ is implicitly computed by reusing already compute component : $O(mnr) \to O(mr^2)$ with $m = K$, $n = IJ \ggg r$ (insane!)
- 3 MTTKRP (Matricized tensor times Khatri-Rao product) if using 1st order solver
- Many variations : e.g. project after extrapolation

Open problem : which chain structure to use?



Understanding the relationship between the data structure (rank size, size of each mode) and the chain structure will be crucial.

More fancy graphs showing numerics

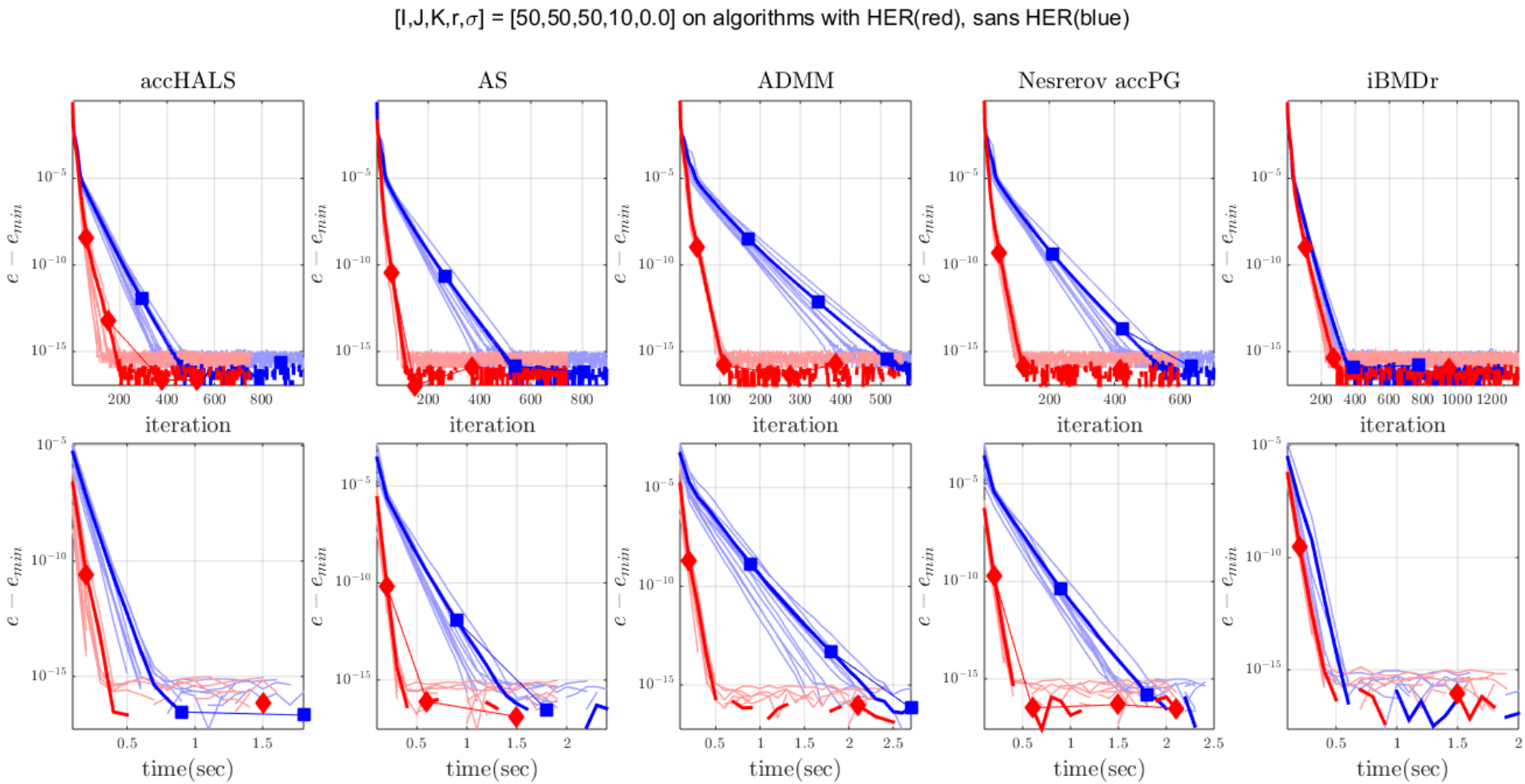


Figure from (A., Cohen, Gillis, Hien, 2019)

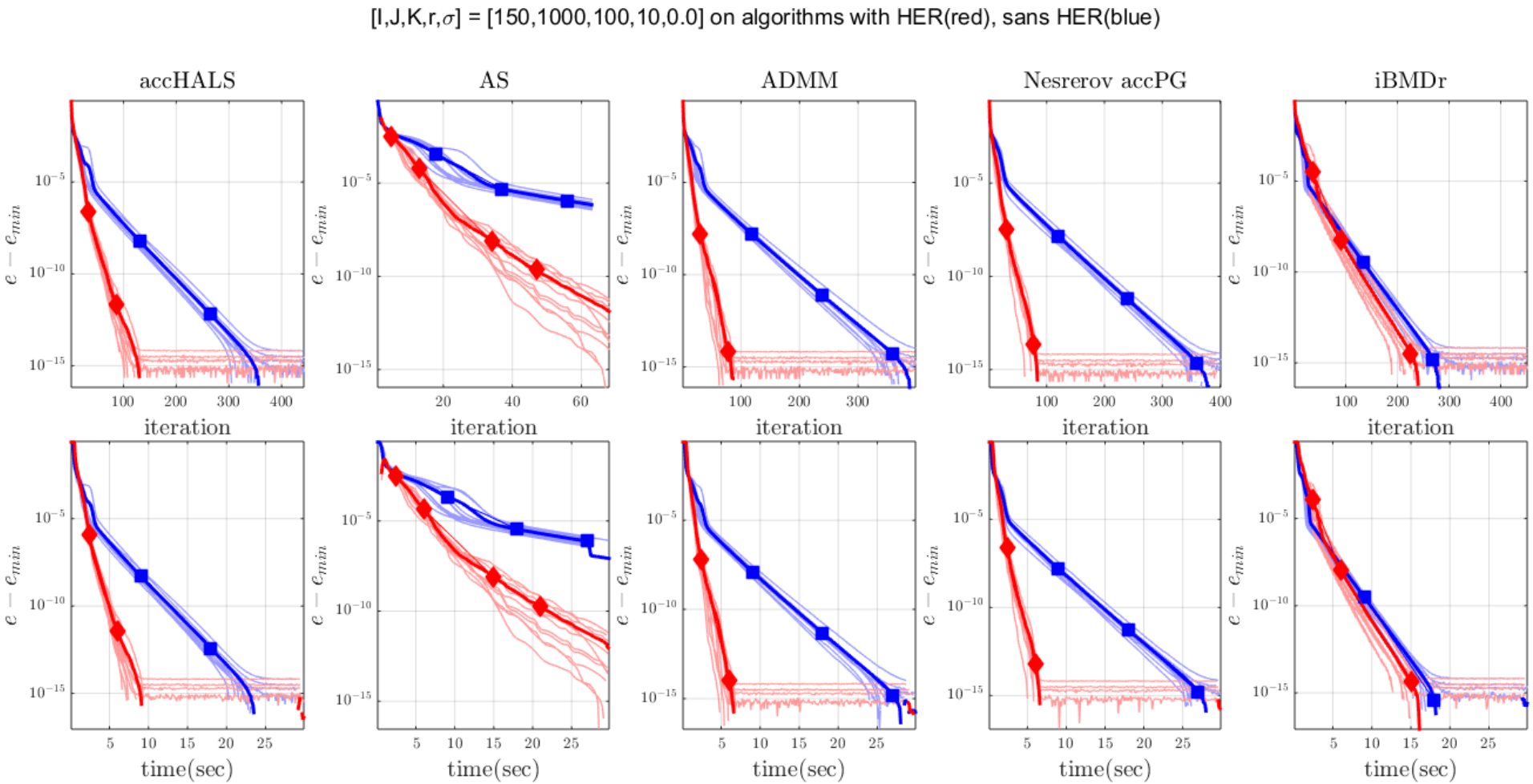


Figure from (A., Cohen, Gillis, Hien, 2019)

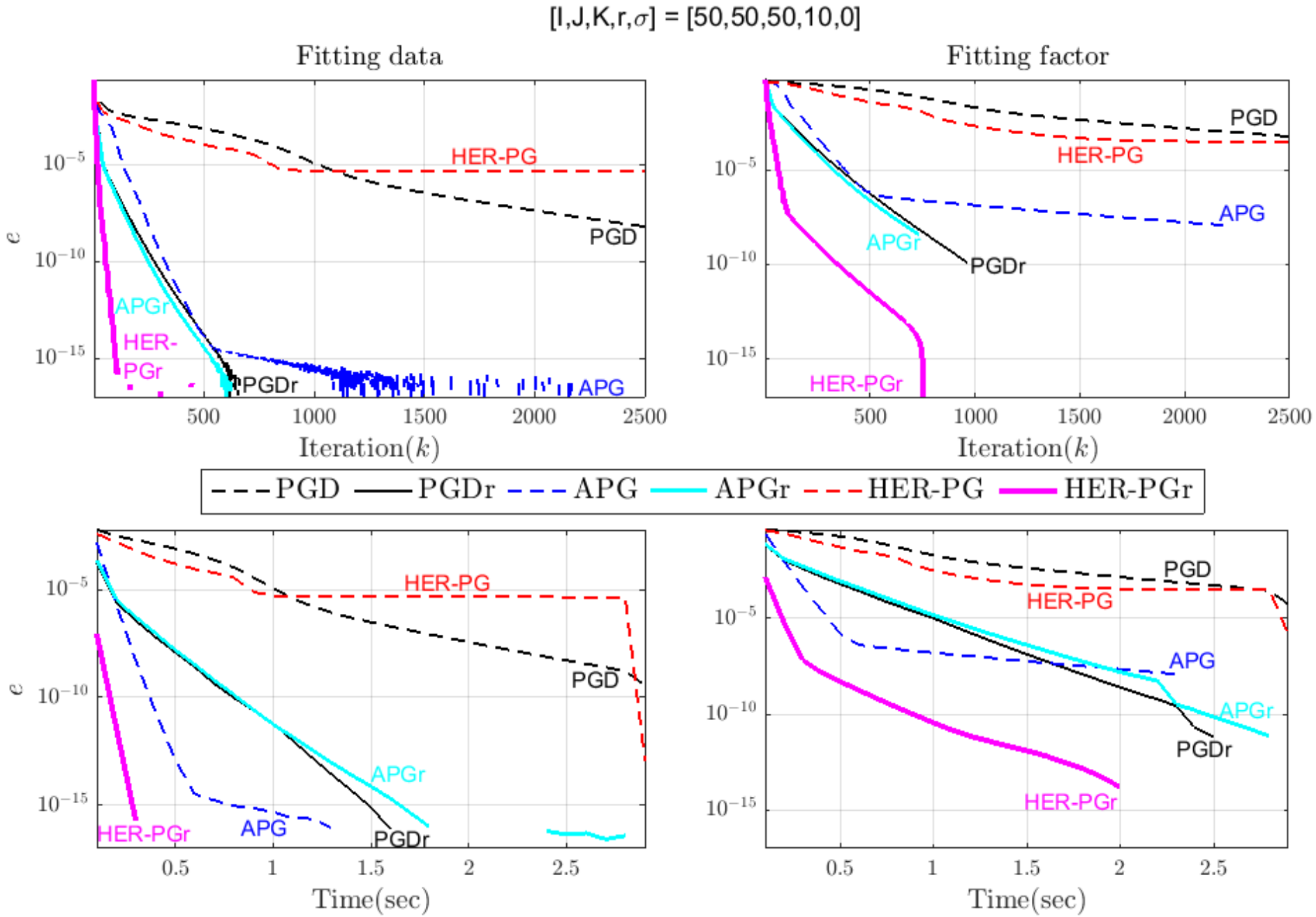
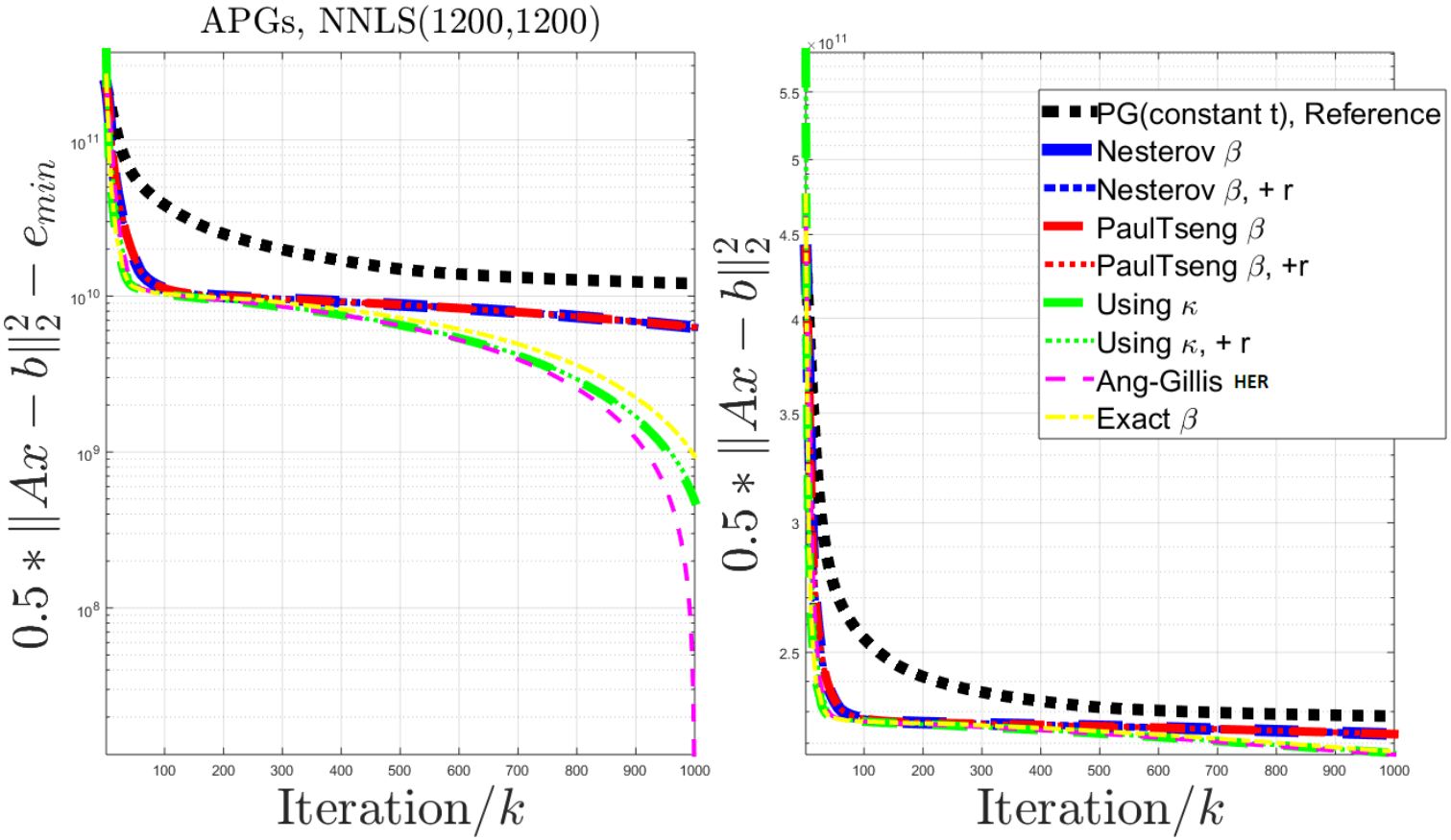


Figure from (A., Cohen, Gillis, Hien, 2019)

Non-negative Least Square

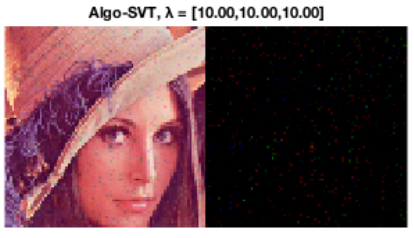
$$\mathbf{x} = \underset{\mathbf{x}\geq\mathbf{0}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

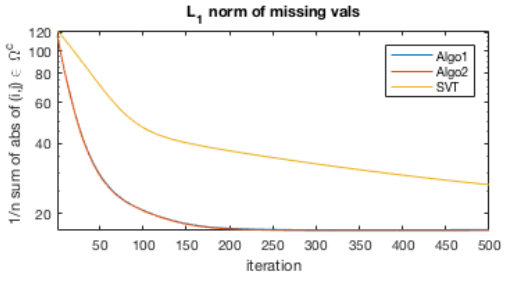


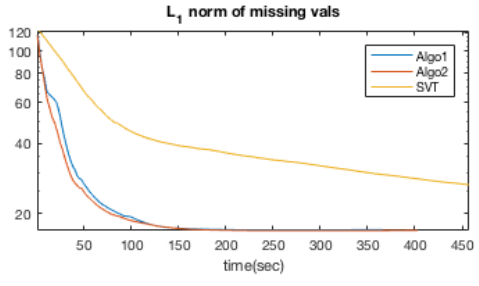
We suspect HER-PG (inexact BCD) just share the same rate as other extrapolated gradients, but again no proof (even NNLS is convex).
For HER-exact BCD, even harder to prove convergence.

Various applications

- HER Empirically also works for other tasks







A toy example on tensor completion. A 9-times speed up (0.11-fraction of time) on the nuclear norm SVT algo.

- Why : HER is highly flexibility – there always exists a suitable parameter for the problem (a hypothesis hard to prove theoretically but easy to verify empirically)

Outline

Summary

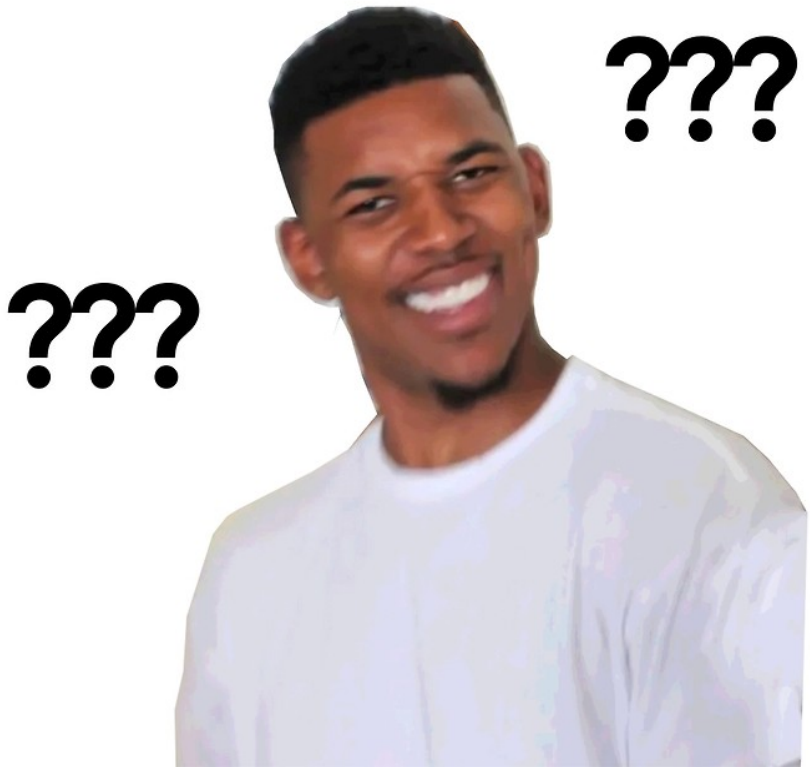
Discussed

- What is NMF
- Applications of NMF
- Theories of NMF : NP-hard, Separability, min-volume, identifiability
- Computations of NMF : HER algorithm framework

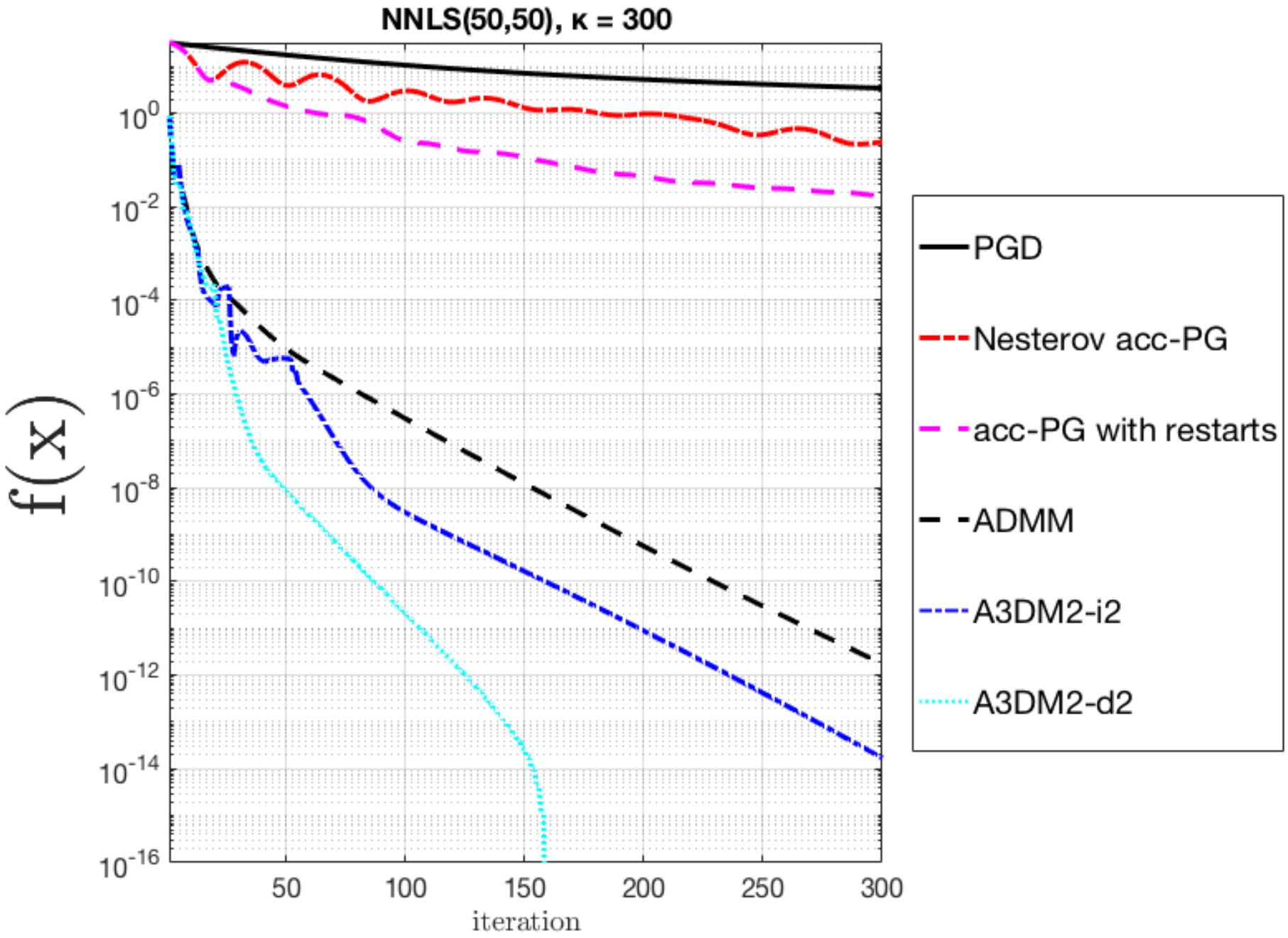
Open problems

- Theoretical : rank_+, NP-completeness
- Robustness of minimum-volume NMF to additive noise
- Convergence analysis of HER : even for the convex case
- and much more !!!

What am I doing here?



What am I doing here?



References

- (A., Cohen, Gillis, Hien, 2019) to appear.
- (Leplat, Gillis, A., 2019)
  Blind Audio Source Separation with Minimum-Volume Beta-Divergence NMF, 2019
- (A., Cohen, Gillis, 2019)
  Accelerating Approximate Nonnegative Canonical Polyadic Decomposition, GRETSI, 2019
- (Leplat, Gillis, Siebert, A., 2019)
  Separation aveugle de sources sonores par factorisation en matrices positives avec penalite sur le volume du dictionnaire, GRETSI, 2019 [**In French**]
- (A. & Gillis. 2019-HSI)
  Algorithms and Comparisons of Non-negative Matrix Factorization with Volume Regularization for Hyperspectral Unmixing, IEEE JSTARS, 2019
- (Leplat, A., Gillis, 2019-UK)
  Minimum-Volume Rank-deficient Non-negative Matrix Factorizations, IEEE ICASSP 2019
- (A. & Gillis, 2019-acc)
  Accelerating Nonnegative Matrix Factorization Algorithms using Extrapolation, Neural Computation, 2019
- (A. & Gillis, 2018-NL)
  Volume regularized non-negative matrix factorizations, IEEE WHISPERS18

Slide, papers, codes all avaliable in *angms.science*

END OF PRESENTATION