ℓ_1-NMF for Sparse Data

Arnaud Vandaele, François Moutier, Nicolas Gillis

Faculté Polytechnique - Service MARO

UMONS

October 29, 2022

Summary

1 Known facts about the ℓ_1-norm in optimization

2 What we want to do

3 The ℓ_1-NMF model $\rightarrow$ our problem

4 How to update ?

5 How to initialize ?

6 Some numerical experiments

7 Conclusion and perspectives

Plan

1 Known facts about the ℓ_1-norm in optimization

2 What we want to do

3 The ℓ_1-NMF model $\rightarrow$ our problem

4 How to update ?

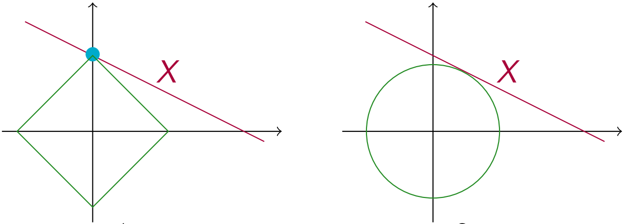
5 How to initialize ?

6 Some numerical experiments

7 Conclusion and perspectives

The ℓ_1-norm induces sparsity

$$\min_x \|x\|_p$$
$$\text{s.t. } x \in X$$

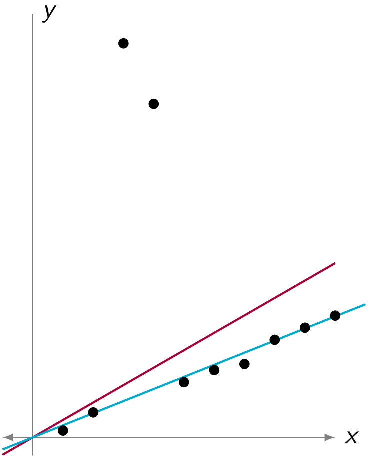


$p = 1$ $p = 2$

The ℓ_1-norm has a special shape: it has spikes at sparse points.

The ℓ_1-norm is more robust to outliers

Given n points (x_i, y_i), we want to fit the model $y = \alpha x$.



For ℓ_2-norm, $\alpha^* = \arg\min_{\alpha} \; (y_1 - \alpha x_1)^2 + ... + (y_n - \alpha x_n)^2 \quad = \arg\min_{\alpha} \|y - \alpha x\|_2$.

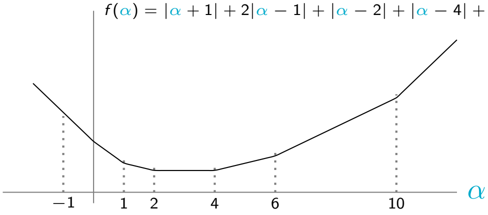
For ℓ_1-norm, $\alpha^* = \arg\min_{\alpha} |y_1 - \alpha x_1| + ... + |y_n - \alpha x_n| \quad = \arg\min_{\alpha} \|y - \alpha x\|_1$.

Focus on these 1-variable problems

Given $x, y \in \mathbb{R}^n$, find $\boldsymbol{\alpha} \in \mathbb{R}$ s.t. $\boldsymbol{\alpha} x$ is *as close as possible* to y.

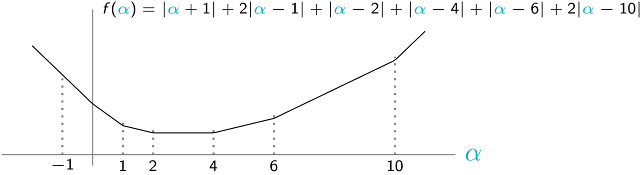
- With the ℓ_2-norm, solving $\min_{\alpha} \|y - \alpha x\|_2^2$ is a quadratic problem:
$$\alpha^* = \frac{x^T y}{\|x\|_2^2}.$$
- With the ℓ_1-norm, $f(\alpha) = \|y - \alpha x\|_1$ is a sum of absolute values:

$$f(\alpha) = |y_1 - \alpha x_1| + \ldots + |y_n - \alpha x_n| = |x_1| \left| \frac{y_1}{x_1} - \alpha \right| + \ldots + |x_n| \left| \frac{y_n}{x_n} - \alpha \right|.$$



$\rightarrow$ $\boldsymbol{\alpha^*}$ **is at one of the breakpoints** $\boldsymbol{\frac{y_i}{x_i}}$**.**

The weighted median algorithm



How to find α^* ?

- Brute-force: $\mathcal{O}(n^2)$ (n breakpoints to evaluate in $\mathcal{O}(n)$)
- $\mathcal{O}(n)$ algorithm related to the median of medians (but unpractical)
- The weighted median algorithm running in $\mathcal{O}(n \log(n))$:

Data: $x \in \mathbb{R}^n$, $y \in \mathbb{R}^n$
Result: $\arg\min_\alpha \sum_i^n |y_i - \alpha x_i|$ (w.l.o.g., we suppose $x_i > 0$ for all i)
$[S, indices] \leftarrow \text{sort}\left(\left\{\frac{y_i}{x_i}\right\}\right)$;
$x \leftarrow x(indices)$;
$t \leftarrow$ smallest index such that $\sum_{i=1}^t x_i \geq \frac{1}{2} \sum_{i=1}^n x_i$;
return S_t;

Plan

1 Known facts about the ℓ_1-norm in optimization

2 What we want to do

3 The ℓ_1-NMF model $\rightarrow$ our problem

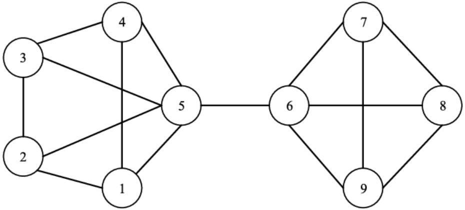
4 How to update ?

5 How to initialize ?

6 Some numerical experiments

7 Conclusion and perspectives

Communities and clusters identification via Matrix Factor.



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

$$\rightarrow \quad W = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

A model/method taking into account the following properties:

- Large-scale data
- Sparse data
- Laplacian noise / Binary noise

Plan

1. Known facts about the ℓ_1-norm in optimization
2. What we want to do
3. The ℓ_1-NMF model $\rightarrow$ our problem
4. How to update ?
5. How to initialize ?
6. Some numerical experiments
7. Conclusion and perspectives

The ℓ_1-NMF model

Given a matrix $X \in \mathbb{R}^{m\times n}$,

$$\min_{W\geq 0, H\geq 0} \|X - WH\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{n} |X - WH|_{ij}.$$

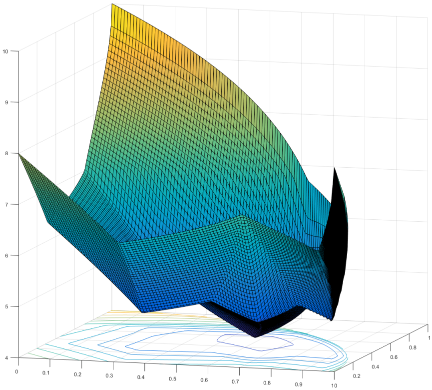
Examples with $r = 1$:

X	ℓ_2-NMF solution	ℓ_1-NMF solution
$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1.03 & 0.92 & 0.92 & 0.92 & 0.44 \\ 1.15 & 1.02 & 1.02 & 1.02 & 0.50 \\ 1.03 & 0.92 & 0.92 & 0.92 & 0.44 \\ 0.40 & 0.36 & 0.36 & 0.36 & 0.17 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$
$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

- This model is much more robust to outliers and non-Gaussian noise
- NP-hard even for $r = 1$

Picture in the rank-1 case

Let us consider the ℓ_1-NMF model with $r = 1$ for $X = \begin{pmatrix} 1 & 3 & 1 \\ 1 & 1 & 1 \\ 3 & 1 & 1 \end{pmatrix}$.



Three local minima: $(\frac{3}{\sqrt{11}}, \frac{1}{\sqrt{11}})$, $(\frac{1}{\sqrt{11}}, \frac{1}{\sqrt{11}})$ and $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$ which is the only global min.

Main contribution: an algorithm for the ℓ_1-NMF model

As for the ℓ_2-norm, the general problem in both W and H is nonconvex

$$\min_{W \geq 0, H \geq 0} \|X - WH\|_1.$$

but when one of the two factors is given, the subproblems becomes convex :

$$\min_{W \geq 0} \|X - WH\|_1 \qquad \min_{H \geq 0} \|X - WH\|_1.$$

Data: $X \in \mathbb{R}_+^{m \times n}$ and factorization rank k.
Result: $(W, H) \geq 0$: a rank-r ℓ_1-NMF of $X \approx WH$
$(W^{(0)}, H^{(0)}) \geq 0 \leftarrow$ initialization step;
for $k = 1, 2, \ldots$ **do**
    $H^{(k)} \leftarrow \text{update}\left(X, W^{(k-1)}\right)$;
    ${W^{(k)}}^T \leftarrow \text{update}\left(X^T, {H^{(k-1)}}^T\right)$;
end

In the following:

- how to *update* ?
- how to *initialize* ?

Plan

1 Known facts about the ℓ_1-norm in optimization

2 What we want to do

3 The ℓ_1-NMF model $\rightarrow$ our problem

4 How to update ?

5 How to initialize ?

6 Some numerical experiments

7 Conclusion and perspectives

How to update ? Not exactly.

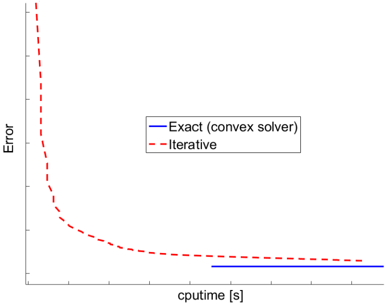
The update of one factor can be separated into n independent subproblems

$$\min_{H \geq 0} \|X - WH\|_1 \quad \rightarrow \quad \min_{H(:,j) \geq 0} \sum_{j=1}^{n} \|X(:,j) - WH(:,j)\|_1,$$

each of them is a convex problem, which can be modelized as an LP

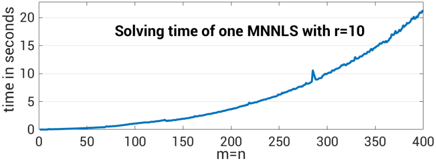
$$\min_{x \geq 0} \|Ax - b\|_1 \qquad \rightarrow \qquad \begin{array}{c} \min_{x \geq 0, t} \; t_1 + \ldots + t_m \\ \begin{pmatrix} -A & I \\ A & I \end{pmatrix} \begin{pmatrix} x \\ t \end{pmatrix} \leq \begin{pmatrix} -b \\ b \end{pmatrix}. \end{array}$$

The end of the story ? No.



No need to update exactly the factors:

- alternate scheme
- cost



How to update ? One coordinate at a time.

The idea is to update iteratively the coordinates x_1, x_2, ... x_r.
For the kth coordinate, the objective function becomes:

$$\min_{x_k \geq 0} \|Ax - b\|_1 \qquad \rightarrow \qquad \min_{\boldsymbol{x_k} \geq 0} \left\| A(:,k)\boldsymbol{x_k} - \left(b - \sum_{i \neq k} A(:,i)x_i \right) \right\|_1,$$

which is exactly the 1-variable problem $\min_{\boldsymbol{\alpha} \geq 0} \|y - \boldsymbol{\alpha} x\|_1$.

Considering the update of the kth entry of the jth column of H, we have:

$$\min_{\boldsymbol{H_{kj}} \geq 0} \left\| W(:,k)\boldsymbol{H_{kj}} - \left(X(:,j) - \sum_{i \neq k} W(:,i)H_{ij} \right) \right\|_1.$$

- If the residual $X(:,j) - \sum_{i \neq k} W(:,i)H_{ij}$ is available in $\mathcal{O}(m)$, there is an $\mathcal{O}(m)$ algorithm to find the optimal value of H_{kj}
- Overall, updating once every entry of H takes $O(mnr)$ operations

A first algorithm running in $\mathcal{O}(mnr)$ operations

Data: $X \in \mathbb{R}^{m\times n}$, $W \in \mathbb{R}_+^{m\times r}$, $H_0 \in \mathbb{R}_+^{r\times n}$
Result: $H \in \mathbb{R}_+^{r\times n}$
$H \leftarrow H_0$;
for $j = 1 : n$ **do**
    $v = WH(:,j) \quad (\mathcal{O}(mr)$ oper.);
    for $k = 1 : r$ **do**
        $x = W(:,k)$;
        $y = X(:,j) - v + W(:,k)H_{k,j} \quad (\mathcal{O}(m)$ oper.);
        $H^+_{k,j} \leftarrow \texttt{weighted\_median}(x,y) \quad (\mathcal{O}(m)$ oper.) ;
        $v \leftarrow v + W(:,k)(H^+_{k,j} - H^{\text{old}}_{k,j}) \quad (\mathcal{O}(m)$ oper.);
    end
end

- The residual is available in $\mathcal{O}(m)$, it allows to update one factor in $\mathcal{O}(mnr)$
- But, unlike HALS for the ℓ_2-norm,
  it is not obvious to use the fact that X can be sparse

First advantage: modif. to scale with sparse matrices

Let $\mathcal{K}^+ = \{i \mid X_{i,j} > 0\}$ and $\mathcal{K}^0 = \{i \mid X_{i,j} = 0\}$ such that $\mathcal{K}^+ \cup \mathcal{K}^0 = \{1, ..., m\}$.

The objective function becomes:

$$\left\| W(:,k)\boldsymbol{H_{kj}} - \left(X(:,j) - \textstyle\sum_{i\neq k} W(:,i)H_{ij} \right) \right\|_1$$
$$= \|W(\mathcal{K}^+,k)\boldsymbol{H_{kj}} - (X(\mathcal{K}^+,j) - \sum_{i\neq k} W(\mathcal{K}^+,i)H_{ij})\|_1 + \|W(\mathcal{K}^0,k)\boldsymbol{H_{kj}} + \sum_{i\neq k} W(\mathcal{K}^0,i)H_{ij}\|_1.$$

And **since $\boldsymbol{W} \geq \boldsymbol{0}$ and $\boldsymbol{H} \geq \boldsymbol{0}$**, the 1-variable optimization problem becomes

$$\min_{\boldsymbol{H_{kj}} \geq 0} \left\| W(\mathcal{K}^+,k)\boldsymbol{H_{kj}} - \left(X(\mathcal{K}^+,j) - \sum_{i\neq k} W(\mathcal{K}^+,i)H_{ij} \right) \right\|_1 + \left\| W(\mathcal{K}^0,k) \right\|_1 \boldsymbol{H_{kj}}.$$

- The size of this problem is $|\mathcal{K}^+| + 1$.
- When X is sparse, $|\mathcal{K}^+| \ll m$.

A new version running in $\mathcal{O}(\text{nnz}(X)r)$ operations

Data: $X \in \mathbb{R}^{m \times n}$, $W \in \mathbb{R}_+^{m \times r}$, $H_0 \in \mathbb{R}_+^{r \times n}$
Result: $H \in \mathbb{R}_+^{r \times n}$
$H \leftarrow H_0$;
$nW(k) = \|W(:,k)\|_1$ for all $k = 1, ..., r$ $\quad (\mathcal{O}(mr)$ oper.);
for $j = 1 : n$ **do**
  $\mathcal{K}^+ = \{i \mid X_{i,j} > 0\}$;
  $v = W(\mathcal{K}^+, :)H(:,j)$ $\quad (\mathcal{O}(|\mathcal{K}^+|r)$ oper.);
  for $k = 1 : r$ **do**
    $x = W(\mathcal{K}^+, k)$;
    $y = X(\mathcal{K}^+, j) - v + W(\mathcal{K}^+, k)H_{k,j}$ $\quad (\mathcal{O}(|\mathcal{K}^+|)$ oper.);
    $c = nW(k) - \|W(\mathcal{K}^+, k)\|_1$ $\quad (\mathcal{O}(|\mathcal{K}^+|)$ oper.);
    $H^+_{k,j} \leftarrow \texttt{weighted\_median}([x\ c], [y\ 0])$ $\quad (\mathcal{O}(|\mathcal{K}^+|)$ oper.);
    $v \leftarrow v + W(\mathcal{K}^+, k)(H^+_{k,j} - H^{\text{old}}_{k,j})$ $\quad (\mathcal{O}(|\mathcal{K}^+|)$ oper.);
  end
end

Updating once every entry of one factor can be done in $\mathcal{O}(\text{nnz}(X)r)$ operations.

Second advantage: when $X, W^{(0)}, H^{(0)}$ are binary

Let's take the update in its elementary form:

$$x = W(:,k)$$
$$y = X(:,j) - \sum_{i \neq k} W(:,i) H_{i,j}$$
$$H^+_{k,j} \leftarrow \texttt{weighted\_median}(x, y)$$

and the weighted-median algorithm:

Data: $x \in \mathbb{R}^m$, $y \in \mathbb{R}^m$
$[S, indices] \leftarrow \text{sort}\left(\left\{\frac{y_i}{x_i}\right\}\right)$;
$x \leftarrow x(indices)$;
$t \leftarrow$ smallest index such that $\sum_{i=1}^{t} x_i \geq \frac{1}{2}\sum_{i=1}^{n} x_i$;
return $\max(0, S_t)$;

- the entries of x belong to $\{0, 1\}$
- the entries of y belong to $\{-r+1, -r+2, ..., 0, 1\}$
- the entries of the ratios S belong to $\{-r+1, -r+2, ..., 0, 1\}$
- because of $\max(0, S_t)$, $H^+_{k,j}$ belongs to $\{0, 1\}$

The factors W, H remain binary along the iterations.

Observation: the initialization is important

Let's take the update in its "sparse" form:

$$
\begin{aligned}
x &= W(:, k) \\
y &= X(:, j) - \sum_{i \neq k} W(:, i) H_{i,j} \\
c &= \|W(:, k)\|_1 - \|W(\mathcal{K}^+, k)\|_1 \\
H^+_{k,j} &\leftarrow \texttt{weighted\_median}([x\ c], [y\ 0])
\end{aligned}
$$

and the weighted-median algorithm:

Data: $x \in \mathbb{R}^m$, $y \in \mathbb{R}^m$
$[S, indices] \leftarrow \text{sort}\left(\left\{\frac{y_i}{x_i}\right\}\right)$;
$x \leftarrow x(indices)$;
$t \leftarrow$ smallest index such that $\sum_{i=1}^{t} x_i \geq \frac{1}{2} \sum_{i=1}^{n} x_i$;
return $\max(0, S_t)$;

- The ratio $\frac{0}{c} = 0$ is always sorted in first position, that is, $x_1 = c$ and $S_1 = 0$
- Observe that the r.h.s of the inequality is $\frac{1}{2}\|W(:, k)\|_1$
- If we have $c = \|W(:, k)\|_1 - \|W(\mathcal{K}^+, k)\|_1 \geq \frac{1}{2}\|W(:, k)\|_1$, that is,

  $$\|W(\mathcal{K}^+, k)\|_1 \leq \frac{1}{2}\|W(:, k)\|_1,$$

  then $t = 1$ and $H^+_{k,j} = 0$.

Plan

1 Known facts about the ℓ_1-norm in optimization

2 What we want to do

3 The ℓ_1-NMF model $\rightarrow$ our problem

4 How to update ?

5 How to initialize ?

6 Some numerical experiments

7 Conclusion and perspectives

How to initialize ? Not randomly.

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

$$W^{(0)} = \begin{pmatrix} 0.28 & 0.35 & 0.73 \\ 0.45 & 0.80 & 0.54 \\ 0.98 & 0.98 & 0.89 \\ 0.82 & 0.49 & 0.25 \\ 0.91 & 0.55 & 0.52 \\ 0.56 & 0.33 & 0.49 \\ 0.86 & 0.94 & 0.93 \\ 0.57 & 0.82 & 0.87 \\ 0.52 & 0.62 & 0.20 \\ 0.62 & 0.83 & 0.68 \end{pmatrix} \quad H^{(0)} = \begin{pmatrix} 0.59 & 0.29 & 0.22 & 0.52 & 0.66 & 0.68 & 0.99 & 0.53 & 0.83 & 0.58 \\ 0.29 & 0.42 & 0.14 & 0.12 & 0.62 & 0.40 & 0.46 & 0.31 & 0.92 & 0.32 \\ 0.22 & 0.20 & 0.53 & 0.17 & 0.56 & 0.44 & 0.82 & 0.54 & 0.79 & 0.84 \end{pmatrix}$$

If we update H, we obtain:

$$H^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

How to update ? Another idea: with the ℓ_2-norm

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

$$W^{(0)} = \begin{pmatrix} 1.05 & 0.08 & 0 \\ 1.11 & 0 & 0 \\ 0.10 & 1.03 & 0 \\ 0 & 0.93 & 0 \\ 0.10 & 1.03 & 0 \\ 0 & 0.96 & 0.30 \\ 0 & 0 & 0.74 \\ 0 & 0 & 1.11 \\ 0 & 0 & 1.11 \\ 0.45 & 0 & 0.79 \end{pmatrix} H^{(0)} = \begin{pmatrix} 1 & 0.84 & 0 & 0 & 0.45 & 0 & 0.37 & 0 & 0.03 & 0.03 \\ 0 & 0 & 1 & 1 & 0.48 & 1 & 0.99 & 0 & 0 & 0.17 \\ 0.11 & 0 & 0.002 & 0.002 & 0 & 0.002 & 0 & 0.79 & 1 & 0.86 \end{pmatrix}$$

If we update H, we obtain:

$$H^{(1)} = \begin{pmatrix} 0.95 & 0.89 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.96 & 0.96 & 0.96 & 0.96 & 0.96 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.89 & 0.89 & 0.89 \end{pmatrix}.$$

How to update ? In a greedy way: one rank-1 factor at a time

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

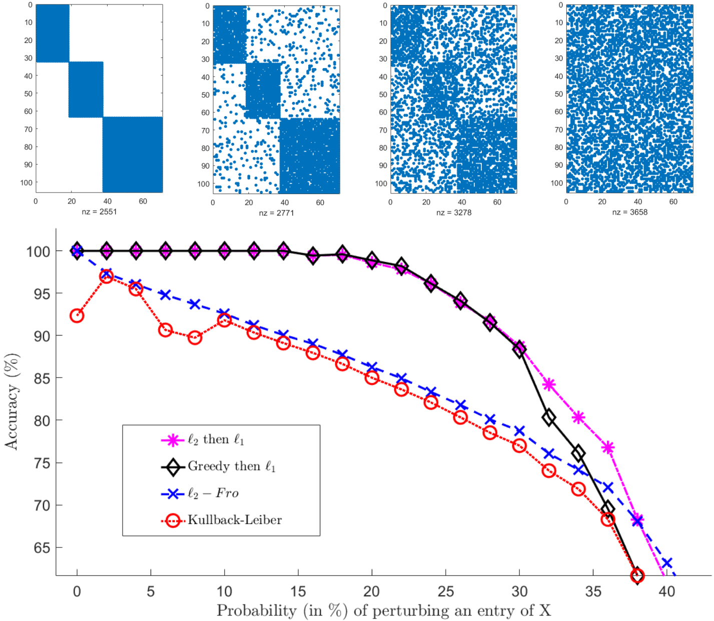
Data: $X \in \mathbb{R}_+^{m \times n}$ **Result:** $w \in \mathbb{R}_+^m$, $h \in \mathbb{R}_+^n$
$w \leftarrow 0^m$;
$i^* \leftarrow \arg\max_{i=1,\ldots,m} \|X_{i,:}\|_1$;
$w(i^*) \leftarrow 1$;
$h \leftarrow X(i^*,:)$;
$[w, h] \leftarrow \arg\min ||X - wh||_1$ via alternate optimization ;

$$W^{(0)} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad H^{(0)} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

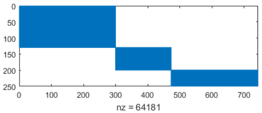
Plan

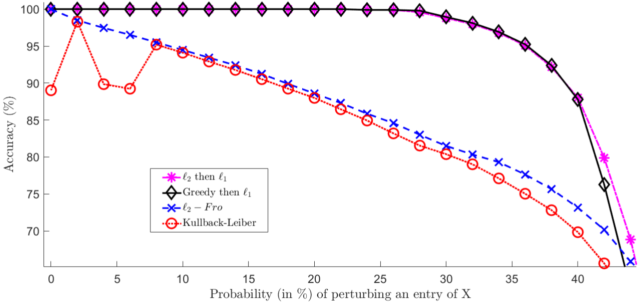
1. Known facts about the ℓ_1-norm in optimization
2. What we want to do
3. The ℓ_1-NMF model $\rightarrow$ our problem
4. How to update ?
5. How to initialize ?
6. Some numerical experiments
7. Conclusion and perspectives

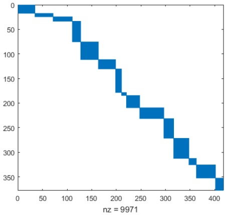
Synthetic datasets

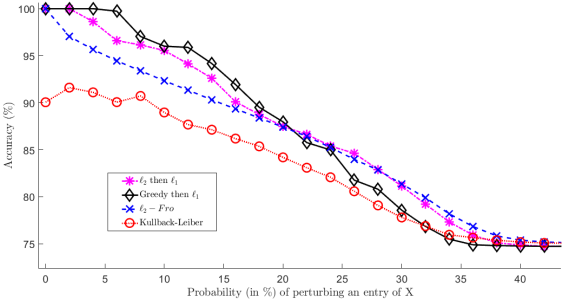


Synthetic datasets









Plan

1 Known facts about the ℓ_1-norm in optimization

2 What we want to do

3 The ℓ_1-NMF model $\rightarrow$ our problem

4 How to update ?

5 How to initialize ?

6 Some numerical experiments

7 Conclusion and perspectives

Conclusion and perspectives

- An algorithm for the ℓ_1-NMF model
- which scales and can handle sparse data
- some pros and cons
- using the ℓ_1-NMF model is *like doing sparse-NMF*

In the near future:

- not wait 6 months before the next iteration on this work
- convince myself about the greedy initialization (and dig a little deeper)
- more compelling *Numerical Experiments* section

Thank you! Questions ?