Matrix-wise ℓ_0-constrained Sparse Nonnegative Least Squares

and application to hyperspectral unmixing

Nicolas Nadisic[1,(2)], Jeremy E Cohen[3], Arnaud Vandaele[2], Nicolas Gillis[2]

September 2023 — ECMLPKDD, Torino

[1]Ghent University, Belgium
[2]University of Mons, Belgium
[3]CNRS, Univ Lyon, France

Starting point: low-rank nonnegative linear models

Focus of this work: models of the form

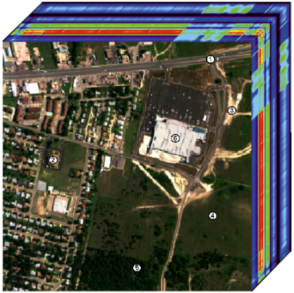
$$B \approx AX,$$

where

- $B \in \mathbb{R}_+^{m \times n}$ is the data/input matrix, representing measures or observations,
- $A \in \mathbb{R}_+^{m \times r}$ is a coeficient matrix, called dictionary, representing features, atoms, or components.
- $X \in \mathbb{R}_+^{r \times n}$ is a signal or information matrix,
- $r \ll \min(m, n)$
- Nonnegativity assumes data is generated from an additive linear combination of features

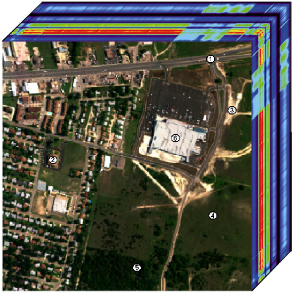
One application — Hyperspectral unmixing

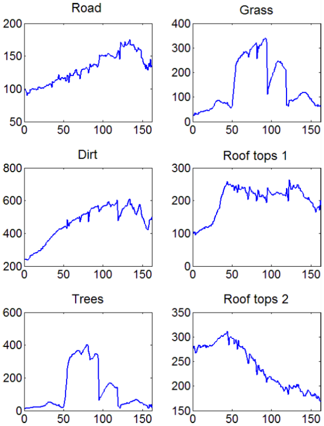
$$\underbrace{B(:,j)}_{\text{spectral signature of j-th pixel}}$$

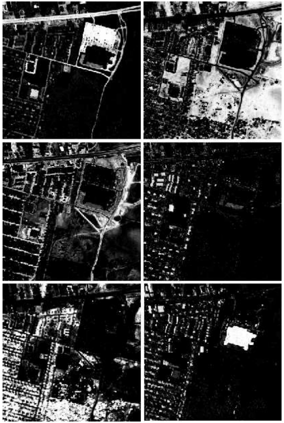


One application — Hyperspectral unmixing

$$\underbrace{B(:,j)}_{\substack{\text{spectral signature of}\\ \text{j-th pixel}}} \approx \sum_p \underbrace{A(:,p)}_{\substack{\text{spectral signature of}\\ \text{p-th material}}} \underbrace{X(p,j)}_{\substack{\text{abundance of p-th material}\\ \text{in j-th pixel}}}$$

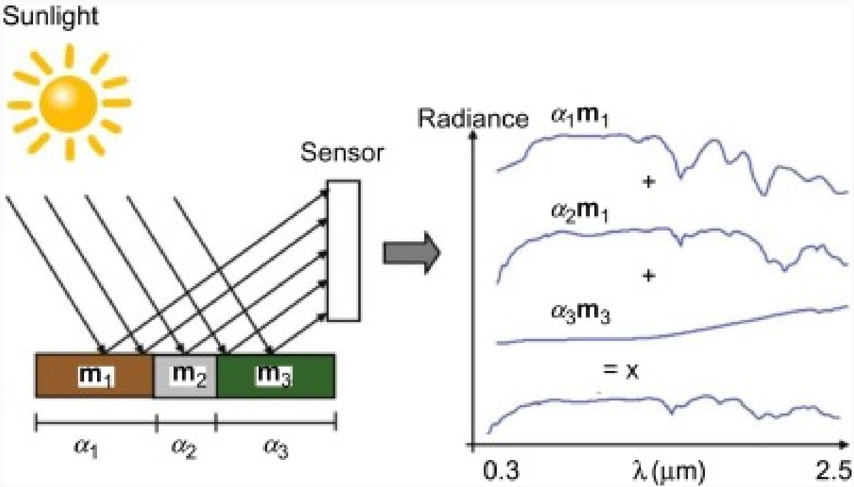






Images from Bioucas Dias and Nicolas Gillis.

Linear mixing model



How to find X given B and A?

Multiple Nonnegative Least Squares (MNNLS) problem

$$\min_{X \geq 0} \|B - AX\|_F^2$$

How to find X given B and A?

Multiple Nonnegative Least Squares (MNNLS) problem

$$\min_{X \geq 0} \|B - AX\|_F^2$$

Can be divided in n independent NNLS subproblems,

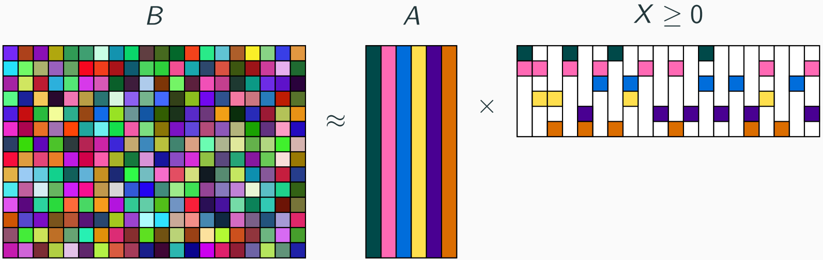
$$\min_{X(:,j) \geq 0} \|B(:,j) - AX(:,j)\|_2^2$$

$$\Leftrightarrow \min_{x \geq 0} \|b - Ax\|_2^2$$

Sparsity — Why?

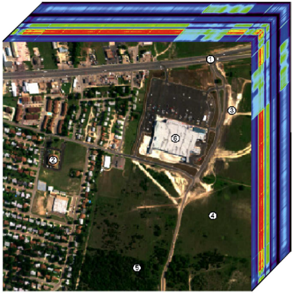
Sparsity of X $\Rightarrow$ Each data point is a combination of only a few features

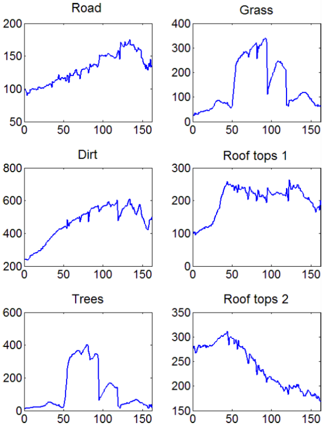
- Regularize the problem
- Better interpretability
- Natural in many applications $\Rightarrow$ leverage a-priori knowledge to improve the model

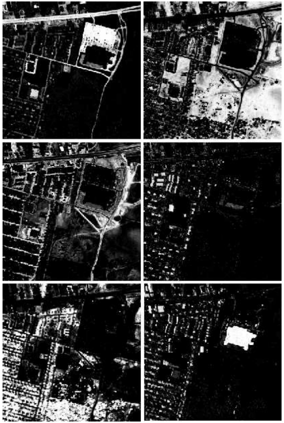


Sparsity in hyperspectral unmixing

$$\underbrace{B(:,j)}_{\substack{\text{spectral signature of}\\ \text{j-th pixel}}} \approx \sum_{p} \underbrace{A(:,p)}_{\substack{\text{spectral signature of}\\ \text{p-th material}}} \underbrace{X(p,j)}_{\substack{\text{abundance of p-th material}\\ \text{in j-th pixel}}}$$







Sparsity — How?

The classical way: ℓ_1 penalty

$$\min_{X \geq 0} \|B - AX\|_F^2 + \lambda \|X\|_1$$

Advantages:

- Convex, easy to optimize

Issues:

- Restrictive condititions for support recovery
- Parameter λ is hard to tune, no physical meaning

Sparsity — How?

More intuitive formulation: column-wise k-sparsity constraint, using the ℓ_0-"norm", $\|x\|_0 = |\{i : x_i \neq 0\}|$

$$\min_{X \geq 0} \|B - AX\|_2^2 \text{ s.t. } \|X(:,j)\|_0 \leq k \text{ for all } j$$

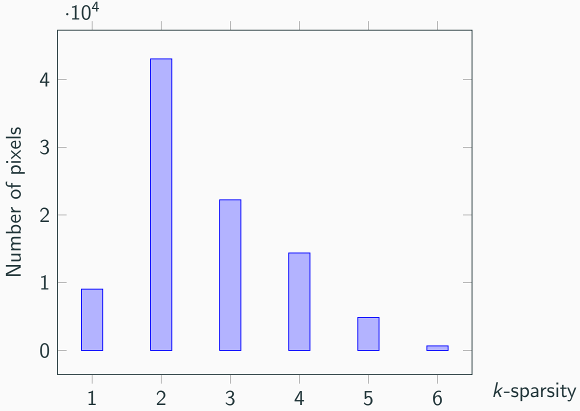
Advantage:

- Interpretable: each data point is a combination of at most k features

Limits of column-wise sparse NNLS

Issue of the column-wise constraint:

- What if the relevant k varies between columns?
- For instance, the number of materials varies between pixels



Sparsity ℓ_0 of the columns of the ground truth X of the HSI Urban, $n = 94249$, $r = 6$

Our solution: A matrix-wise ℓ_0 constraint

Matrix-wise q-sparse MNNLS

$$\min_{X \geq 0} \|B - AX\|_F^2 \quad \text{s.t.} \quad \|X\|_0 \leq q$$

- Can be seen as a global sparsity budget
- If $q = k \times n$, this enforces an average k-sparsity on the columns of X

Our solution: A matrix-wise ℓ_0 constraint

Matrix-wise q-sparse MNNLS

$$\min_{X \geq 0} \|B - AX\|_F^2 \quad \text{s.t.} \quad \|X\|_0 \leq q$$

- Can be seen as a global sparsity budget
- If $q = k \times n$, this enforces an average k-sparsity on the columns of X

How to solve it?

- With a k-sparse NNLS methods, by vectorizing the problem
  $\Rightarrow$ leads to a huge NNLS problem, too expensive to solve
- Our contribution: dedicated algorithm, divide and conquer

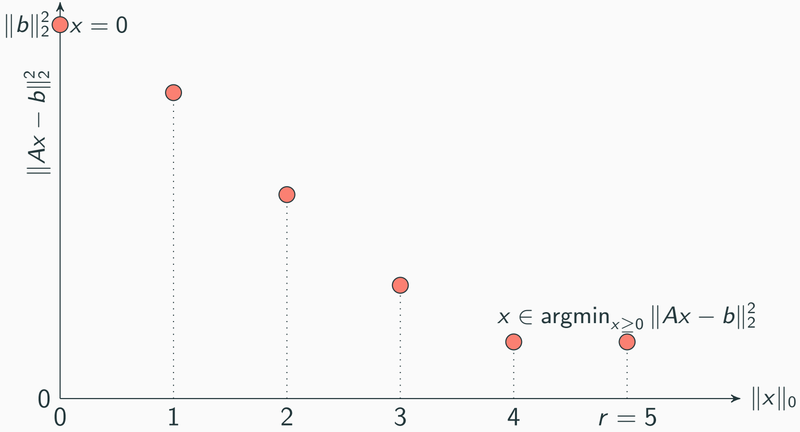
Bi-objective sparse NNLS

$$\min_{x \geq 0} \begin{cases} \|Ax - b\|_2^2 \\ \|x\|_0 \end{cases}$$

Equivalent to $\min_{x \geq 0} \|Ax - b\|_2^2$ s.t. $\|x\|_0 \leq k$ for all $k \in \{0, \ldots, r\}$

Bi-objective sparse NNLS — Pareto front

Example for $r = 5$



Our contribution: a two-step algorithm

Algorithm Salmon[1]:

1. Generate a set of solutions for every column of X, with different tradeoffs between reconstruction error and sparsity
   - Divide the sparse MNNLS problem into n biobjective sparse NNLS subproblems

$$\min_{X(:,j)\geq 0} \left\{ \quad \|B(:,j) - AX(:,j)\|_2^2 \quad , \quad \|X(:,j)\|_0 \quad \right\}$$

   - Solve with branch-and-bound, or heuristic (homotopy, greedy algo)
   - Build a cost matrix C

[1] Salmon Applies ℓ_0-constraints Matrix-wise On NNLS problems

Salmon — Step 1: Build the cost matrix C

- Each row = one sparsity level
- Each column = one column of the MNNLS problem

$$\begin{pmatrix} C_{0,1} & C_{0,2} & \cdots & C_{0,n} \\ C_{1,1} & C_{1,2} & \cdots & C_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{r,1} & C_{r,2} & \cdots & C_{r,n} \end{pmatrix}$$

$$C(i,j) \quad \approx \quad \min_{x \geq 0} \|B(:,j) - Ax\|_2^2 \text{ s.t. } \|x\|_0 \leq i$$

Our contribution: a two-step algorithm

Algorithm Salmon[2]:

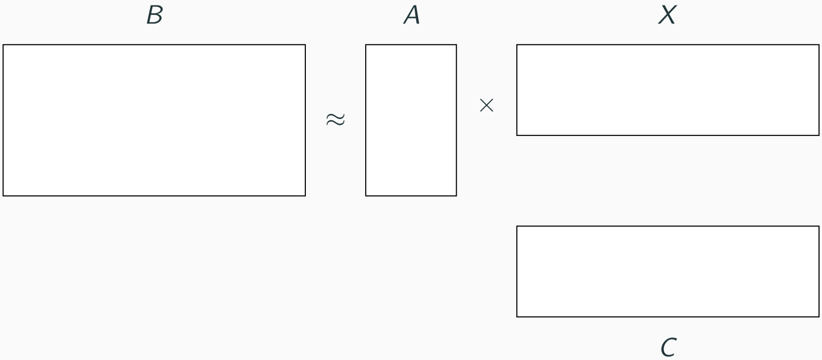
1. Generate a set of solutions for every column of X, with different tradeoffs between reconstruction error and sparsity
   - Divide the sparse MNNLS problem into n biobjective sparse NNLS subproblems

$$\min_{X(:,j)\geq 0} \left\{ \quad \|B(:,j) - AX(:,j)\|_2^2 \quad , \quad \|X(:,j)\|_0 \quad \right\}$$

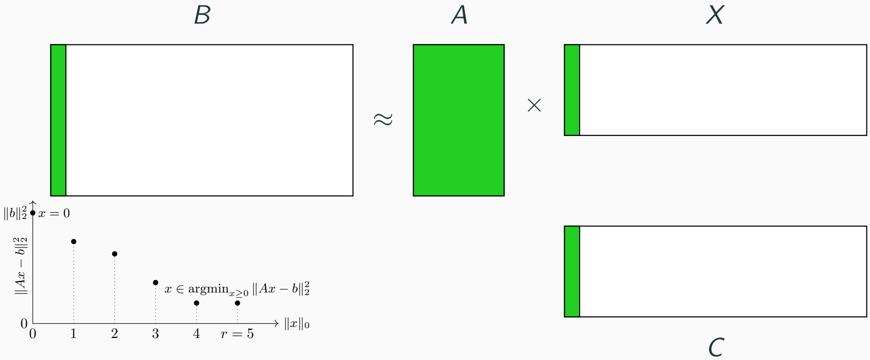
   - Solve with branch-and-bound, or heuristic (homotopy, greedy algo)
   - Build a cost matrix C
2. Select one solution per column such that in total X has q nonzero entries and the error is minimized $\Rightarrow$ assignment-like problem
   - Dedicated greedy algorithm proved near-optimal

[2] Salmon Applies ℓ_0-constraints Matrix-wise On NNLS problems

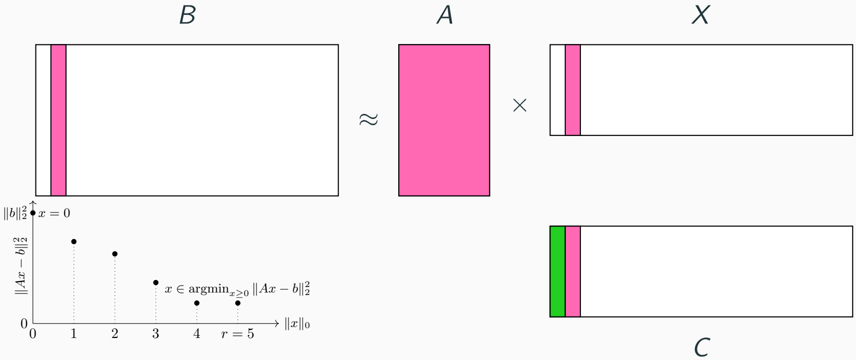
Salmon — Step 1: Generate Pareto fronts



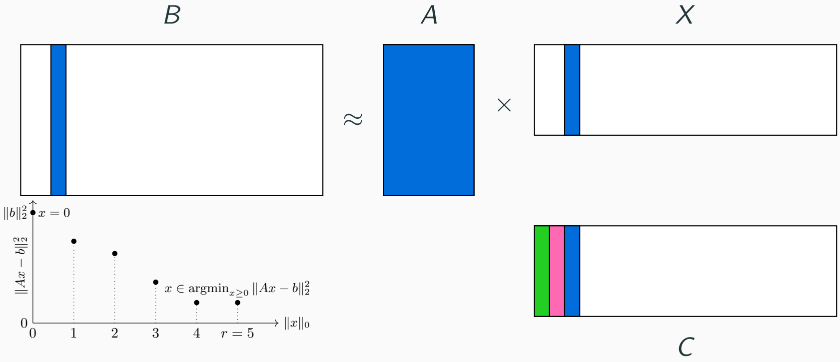
Salmon — Step 1: Generate Pareto fronts



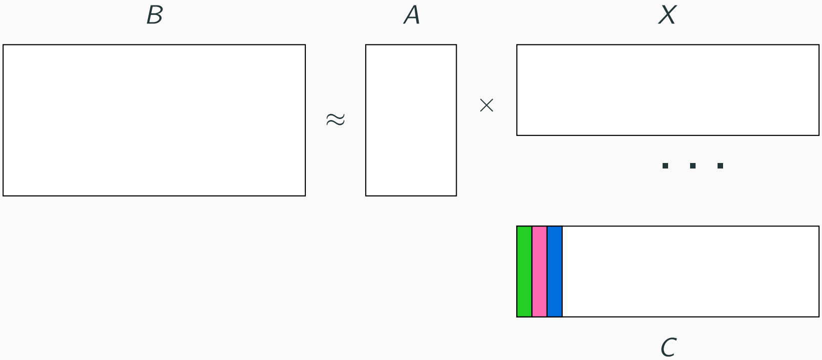
Salmon — Step 1: Generate Pareto fronts



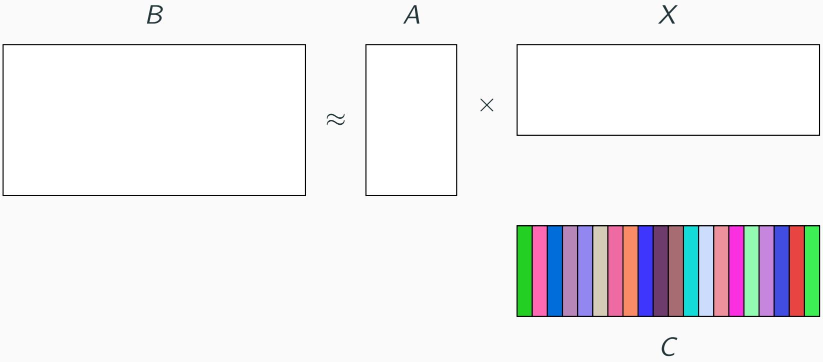
Salmon — Step 1: Generate Pareto fronts



Salmon — Step 1: Generate Pareto fronts



Salmon — Step 1: Generate Pareto fronts



Salmon — Step 2: Select one solution per column

$$\begin{pmatrix} C_{0,1} & C_{0,2} & \cdots & C_{0,n} \\ C_{1,1} & C_{1,2} & \cdots & C_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{r,1} & C_{r,2} & \cdots & C_{r,n} \end{pmatrix} \qquad \text{with } C(i,j) \approx \min_{x \geq 0} \|B(:,j) - Ax\|_2^2 \text{ s.t. } \|x\|_0 \leq i$$

Salmon — Step 2: Select one solution per column

$$\begin{pmatrix} C_{0,1} & C_{0,2} & \cdots & C_{0,n} \\ C_{1,1} & C_{1,2} & \cdots & C_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{r,1} & C_{r,2} & \cdots & C_{r,n} \end{pmatrix} \qquad \text{with } C(i,j) \approx \min_{x\geq 0} \|B(:,j) - Ax\|_2^2 \text{ s.t. } \|x\|_0 \leq i$$

Similar to an assignment problem

Let $z_{i,j} \in \{0,1\}$ such that $z_{i,j} = 1$ if and only if the jth column of X is i-sparse,

$$\min_{z \in \{0,1\}^{r \times n}} \sum_{i,j} z_{i,j} C(i,j)$$

$$\text{such that } \sum_i z_{i,j} = 1 \text{ for all } j, \quad \text{and} \quad \sum_{i,j} i\, z_{i,j} \leq q.$$

Solved with a dedicated greedy algorithm, fast but proved near-optimal

Near-optimality of the selection step (step 2)

In short:

- The worst case is not too bad (wrong support in at most one column)
- In practice, often optimal (19 out of 22 cases in our exp)

Near-optimality of the selection step (step 2)

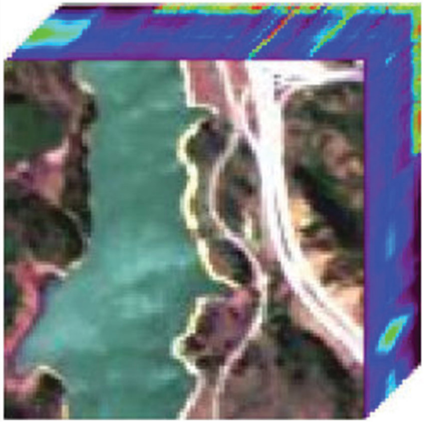
In short:

- The worst case is not too bad (wrong support in at most one column)
- In practice, often optimal (19 out of 22 cases in our exp)

Intuition of the proof:

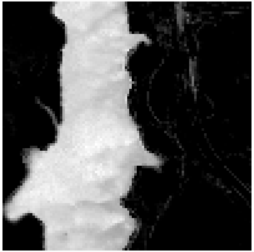
- The objective function is separable by columns
- At each iteration, we maximize the global decrease in error

Exp: Unmixing of the hyperspectral image Jasper Ridge

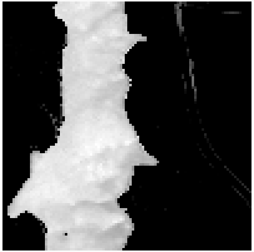


Exp: Unmixing of the hyperspectral image Jasper Ridge ($r = 4$)

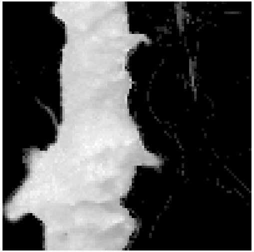
Using homotopy in step 1. We show only one material (row of X reshaped) corresp. to water.



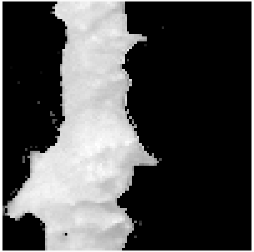
NNLS (no sparse)
Error $= 5.71\%$
Actual sparsity $= 2.27$



Col-wise, $k = 2$
Error $= 6.99\%$
Actual sparsity $= 1.78$



Salmon, $q/n = 2$
Error $= 5.72\%$



Salmon, $q/n = 1.8$
Error $= 5.95\%$

Conclusion

- We introduced a sparse MNNLS model with matrix-wise ℓ_0-sparsity constraint
- We developed a two-step algorithm to tackle it
- Makes tractable some problems that are too big for standard NNLS solvers
- Improves results, allows a finer parameter tuning
- Interesting where sparsity varies between columns

Matrix-wise ℓ_0-constrained Sparse Nonnegative Least Squares

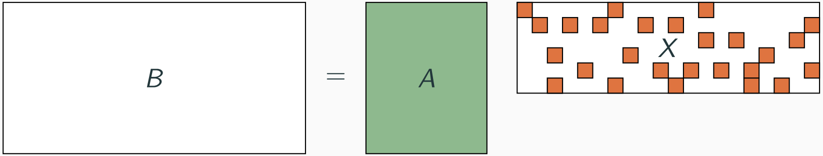
Nicolas Nadisic, Jeremy E Cohen, Arnaud Vandaele, Nicolas Gillis

Contact:

nicolas.nadisic@ugent.be

Paper and code:

http://nicolasnadisic.xyz



Back-up slides

Vectorizing the MNNLS problem is expensive

$$\min_{H \geq 0} \|M - WH\|_2^2 \text{ s.t. } \|H\|_0 \leq q$$

$\Rightarrow$ vectorize

$$\min_{h \geq 0} \|m - \Omega h\|_2^2 \text{ s.t. } \|h\|_0 \leq q$$

where $\Omega = W \otimes I \in \mathbb{R}^{(m.n)\times(r.n)}$ and $m = \begin{bmatrix} M(:,1) \\ M(:,2) \\ \vdots \\ M(:,n) \end{bmatrix} \in \mathbb{R}^{(m.n)}$

Experiments — Computing time, error

		AS	ℓ_1-CD	Hcw	**H+S**	OGcw	OGg	**OG+S**	ARBOcw	**ARBO+S**
$k = 3$	Sparsity	3.45	3	2.86	2.99	2.7	3	3	2.76	3
Jasper	Time	0.34	0.22	0.38	0.48	0.39	6.08	1.12	1.21	1.93
$r = 4$	Error	5.71	**5.72**	6.99	**5.72***	7.49	5.76	5.73	6.18	**5.71***
$k = 2$	Sparsity	2.27	2	1.78	1.99	1.72	2	2	1.78	2
Jasper	Time	-	0.18	-	0.44	-	5.26	1.15	-	1.7
$r = 4$	Error	-	7.87	-	5.95*	-	6.06	**5.77***	-	**5.74***
$q/n = 1.8$	Sparsity	-	1.8	-	1.79	-	1.8	1.8	-	1.8
Samson	Time	0.22	0.24	0.2	0.26	0.31	3.67	0.57	0.52	0.8
$r = 3$	Error	3.3	**3.3**	3.34	**3.3***	6.76	3.32	**3.3***	3.4	**3.3***
$k = 2$	Sparsity	2.2	2	1.85	2	1.6	1.99	1.99	1.83	2
Urban	Time	5.08	4.31	4.86	7.79	3.38	958	16.4	33.5	73.1
$r = 6$	Error	7.67	8.13	8.62	7.83*	8.97	8.07	**7.76***	8.27	**7.71***
$k = 2$	Sparsity	2.63	2	1.9	2	1.7	2	2	1.83	2
Cuprite	Time	5.19	3.32	7.86	10.1	5.06	620	31.5	784	4829
$r = 12$	Error	1.74	3.17	2.37	2.01	2.32	1.97	**1.89***	1.93	**1.83***
$k = 4$	Sparsity	6.61	4	3.92	4	3.53	4	4	3.81	4

Solving column-wise k-sparse NNLS

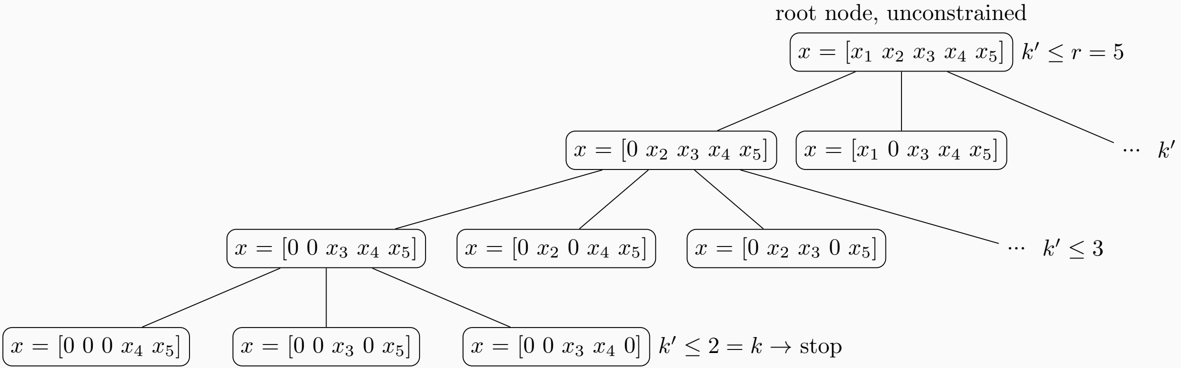
Let us focus on the one-column problem for now,

$$\min_{x \geq 0} \|Ax - b\|_2^2 \text{ s.t. } \|x\|_0 \leq k$$

- Reduces to finding the support of x (set of non-zero entries)
- Combinatorial problem, $\binom{r}{k}$ possible supports
- Can be solved approximately by greedy algorithms
- Or optimally with branch-and-bound algorithms

A branch-and-bound algorithm for k-sparse NNLS

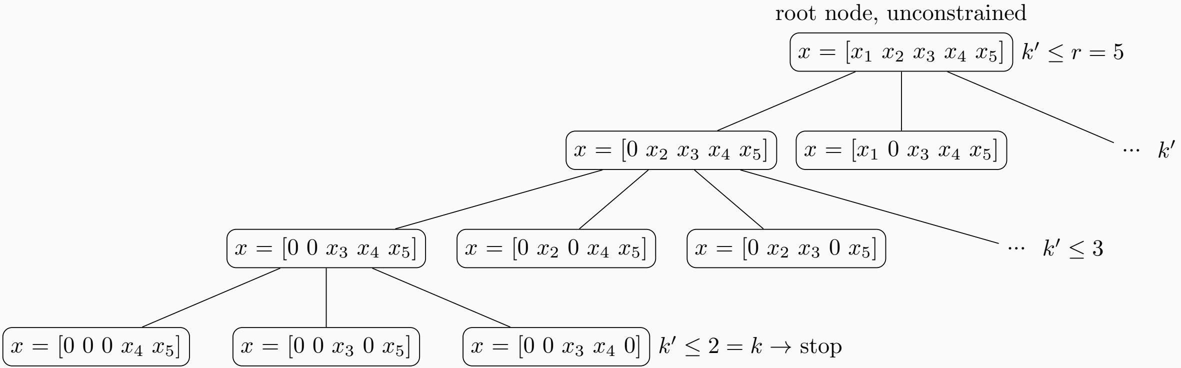
Example for $r = 5$ and $k = 2$



Able to prune large parts of the search space.

Extension of the branch-and-bound algorithm

Example for $r = 5$ and $k = 2$

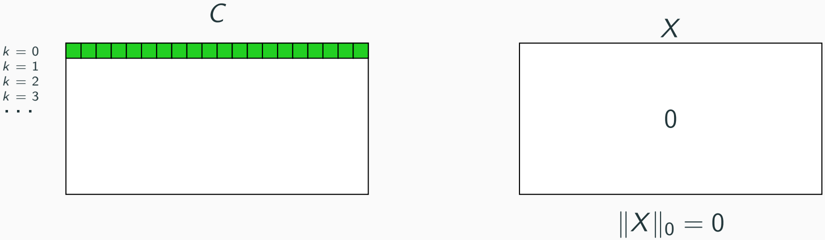


Computes the whole Pareto front!

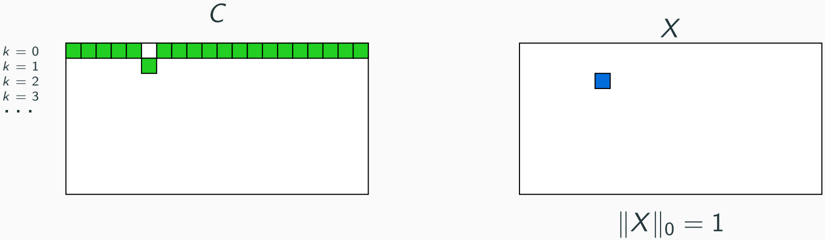
How to leverage this bi-objective formulation on a multicolumn problem?

$$\min_{X \geq 0} \|B - AX\|_F^2$$

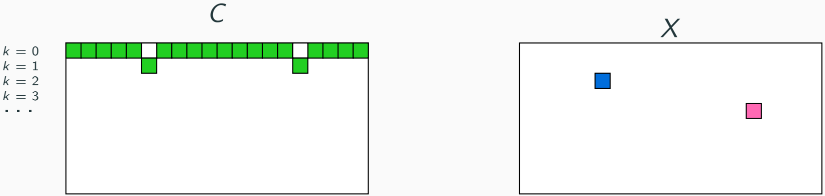
Salmon — Step 2: Greedy selection



Salmon — Step 2: Greedy selection

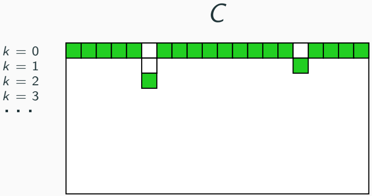


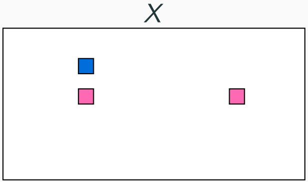
Salmon — Step 2: Greedy selection



$\|X\|_0 = 2$

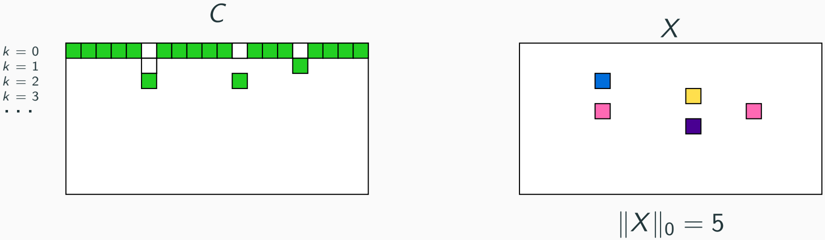
Salmon — Step 2: Greedy selection



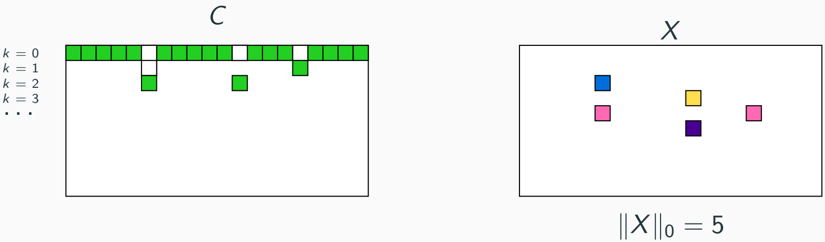


$$\|X\|_0 = 3$$

Salmon — Step 2: Greedy selection

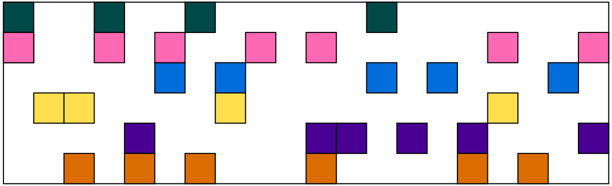


Salmon — Step 2: Greedy selection



Iterate while $\|X\|_0 < q$

Salmon — Step 2: Greedy selection



Final solution X, q-sparse matrix

$$X \approx \arg\min_{X \geq 0} \|B - AX\|_F^2 \quad \text{s.t.} \quad \|X\|_0 \leq q$$