# Exact Sparse Nonnegative Least Squares

## A branch-and-bound algorithm for faster and robust exact solving.

## Nonnegative Least Squares

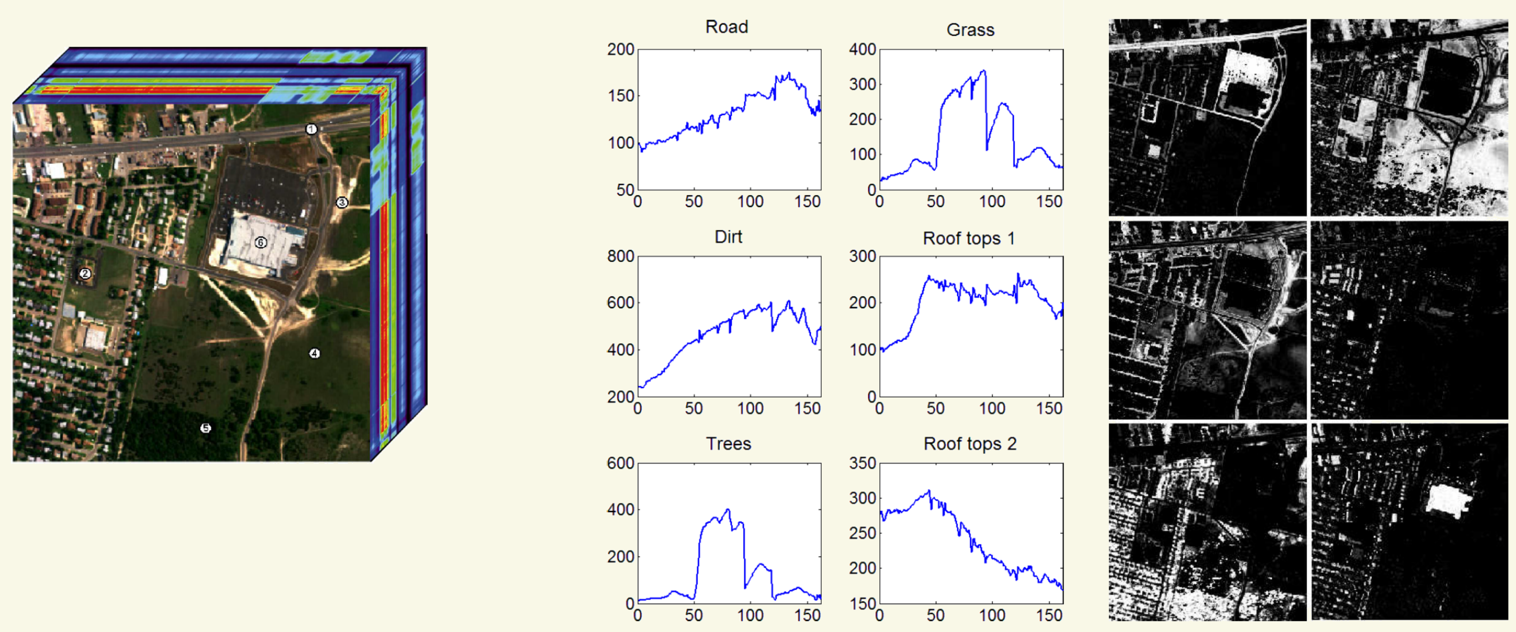
Given $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, find $x^* \in \mathbb{R}^n$,

$$x^* = \arg\min_x ||Ax - b||_2^2 \text{ such that } x \geq 0. \quad (1)$$

- Solution constrained to be entry-wise nonnegative.
- Useful when data point = additive linear combinations of atoms.

## Example: hyperspectral imaging

$$\underbrace{M(:,j)}_{\substack{\text{spectral signature}\\ \text{of } j\text{th pixel}}} \approx \sum_{k=1} \underbrace{U(:,k)}_{\substack{\text{spectral signature}\\ \text{of } k\text{th endmember}}} \underbrace{V(k,j)}_{\substack{\text{abundance of } k\text{th endmember}\\ \text{in } j\text{th pixel}}}.$$




$b = M(:,j)$ $\quad A = U$ $\quad x = V(:,j)$

## Sparsity

**What?**

- Sparse solution = has only a few nonzero elements.
- Data point $b$ = combination of a few atoms $A(:,i)$.

**Why?**

- Improves interpretability.
- Used as a constraint, can represent an a priori knowledge and help reducing noise.

**How?**

- Nonnegativity naturally produces sparsity, but with no guarantee.
- Need to constrain the objective function.
- A natural measure is the $\ell_0$-"norm".
- $\|x\|_0 = Card\{i : x_i \neq 0\}$ = number of nonzero entries of x.

## Problem statement

**Sparse Nonnegative Least Squares** or **Nonnegative Sparse Coding**.
Given $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $k \in \mathbb{N}$, find $x^* \in \mathbb{R}^n$,

$$x^* = \arg\min_x ||Ax - b||_2^2 \text{ such that } x \geq 0 \text{ and } \|x\|_0 \leq k. \quad (2)$$

To solve (2) is equivalent to find the best $k$-support of $x$.

## Issues

- $\ell_0$-"norm" is nonconvex and nonsmooth, thus hard to enforce.
- The problem is NP-hard and combinatorial.
- $\binom{n}{k}$ possible supports.

Example: for $n = 4$ and $k = 2$, possible k-sparse supports are
$[\ 0\ 0\ x_3\ x_4\ ]$ $[\ 0\ x_2\ 0\ x_4\ ]$ $[\ 0\ x_2\ x_3\ 0\ ]$ $[\ x_1\ 0\ 0\ x_4\ ]$ $[\ x_1\ 0\ x_3\ 0\ ]$ $[\ x_1\ x_2\ 0\ 0\ ]$.

## Existing methods

**Relaxation to LASSO**

$\ell_1$-norm penalty $\longrightarrow \min_x ||Ax - b||_2^2 + \lambda\|x\|_1$.

☺ convex, can be solved with coordinate/gradient descent.

☹ $\lambda$ hard to tune, bias introduced, no guarantee.

**Greedy heuristics**

OMP, OLS …Select atoms one by one, to maximize decrease of error.

☺ fast, gives required sparsity.

☹ no guarantee.

**Bruteforce**

Enumerate all possible supports, solve the NNLS subproblem for each one, keep the one with lowest error.

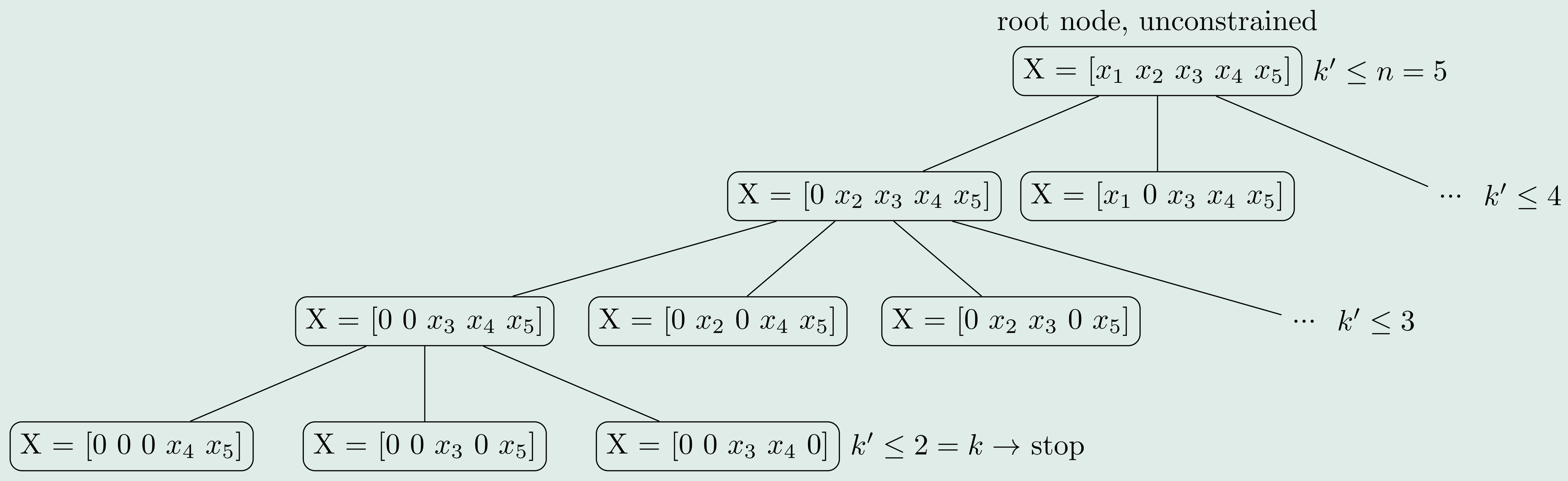
☺ exact resolution.

☹ too expensive.

## Our branch-and-bound algorithm: arborescent

arborescent Realizes a Branch-and-bound Optimization to Require Explicit Sparsity Constraints to be Enforced in NNLS Tasks.

Instead of enumerating all solutions, prune the search space.
Ex, $n = 5$ $k = 2$.




- One node = parent solution + over-support.
- NNLS subproblems solved with active-set (warm start).
- At root node, sort the entries by ascending order of the entries of the unconstrained solution.
- Then, explore depth-first and "left-first".

## Experiments on synthetic datasets

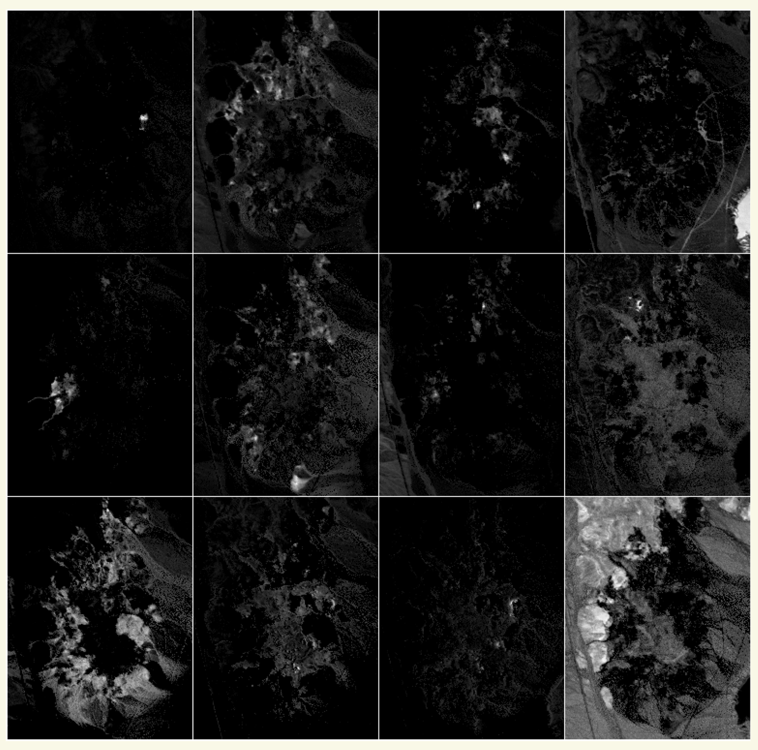
| | Well-cond A, No-noise b | | | Well-cond A, Noisy b | | | Ill-cond A, No-noise b | | | Ill-cond A, Noisy b | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Rel. Err. | Time | Succ. | Rel. Err. | Time | Succ. | Rel. Err. | Time | Succ. | Rel. Err. | Time | Succ. |
| L1-CD | 3.41 | 87.99 | 40 | 4.80 | 85 | 48 | 4.83 | 84.73 | 4 | 5.63 | 79.79 | 4 |
| CVX | 0.23 | 510.69 | 95 | 3.01 | 509.73 | 90 | 2.86 | 481.34 | 14 | 4.11 | 496.19 | 12 |
| NNOMP | 0.48 | 2.64 | 89 | 3.12 | 2.74 | 83 | 3.09 | 2.62 | 3 | 4.02 | 2.88 | 2 |
| NNOLS | 0.20 | 2.36 | 95 | 2.93 | 2.51 | 92 | 2.15 | 2.45 | 14 | 3.62 | 2.59 | 17 |
| arbo. | 0 | 46.85 | 100 | 2.73 | 1211.1 | 100 | 0 | 30.19 | 100 | 2.84 | 1374.1 | 80 |

Results for 100 runs, random data, $m = 100$, $n = 20$, $k = 10$.

Time in ms. Success = recovery of original support of x.

- arborescent finds the optimal solution for most cases, even the dificult ones, at the cost of a higher computation time.

## Hyperspectral unmixing

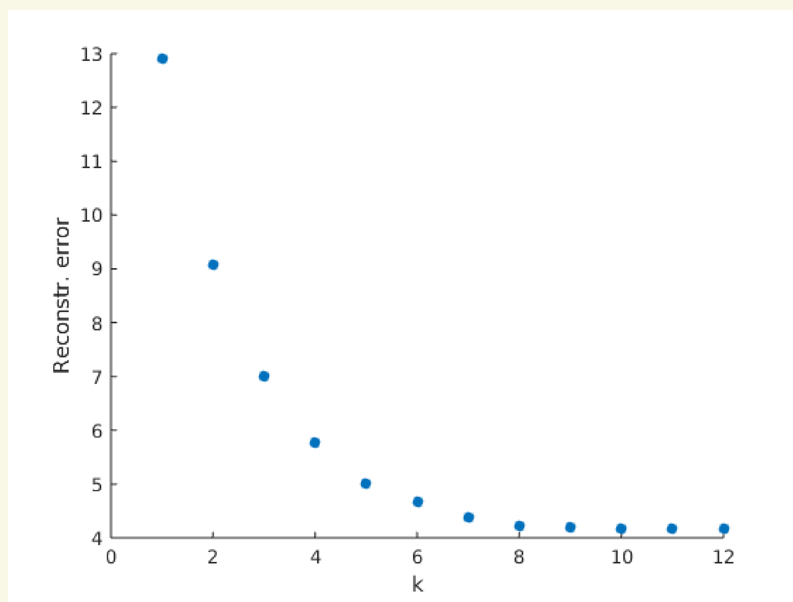


- Cuprite image: $250 \times 191 = 47750$ pixels, $m = 188$ spectral bands, $n = 12$ materials, we look for materials concentrations in pixels.
- Constraint: a pixel is composed of at most $k = 5$ materials.

| | CD | L1-CD | arbo. |
|---|---|---|---|
| Time (s.) | 15.81 | 22.35 | 1053 |
| Rel. Error (%) | 1.74 | 4.21 | 1.78 |
| Sparsity | 6.61 | 4.34 | 4.77 |

## WIP: Pareto front

As a side-effect, arbo. finds the optimal solution for every sparsity level $k' \in \{k, k+1, \ldots, n\}$.
$\longrightarrow$ Pareto set minimizing error and $k$.




Automatic $k$ detection is possible.

## Take-home message

- We developed a branch-and-bound algo. for Exact Sparse Nonnegative Least Squares.
- It finds the optimal solution even with ill-conditioned data, and has a good tolerance to noise.
- Quite slower than non-exact methods.

## WIP: Sparse Nonneg. Matrix Factorization

- Find both the atoms and the coeficients.
- $\min_{U,V} \|M - UV\|_F^2$.
- Solved by alternate optimization of $U$ and $V$.
- Optimize $U$ (resp. $V$) $\Leftrightarrow$ NNLS for each column of $U'$ (resp. $V$).
- Using arbo. we can do Sparse NMF with column-wise sparsity constraints.

## Code and refs


Nicolas Nadisic, Arnaud Vandaele, Jeremy Cohen, Nicolas Gillis
nicolas.nadisic@umons.ac.be - nicolasnadisic.xyz
Université de Mons, Belgium


UMONS Université de Mons