Inverse problems and sparse models (5/6)

Rémi Gribonval
INRIA Rennes - Bretagne Atlantique, France
remi.gribonval@inria.fr

Overview of the course

- **Introduction**
  - ✓ sparsity & data compression
  - ✓ inverse problems in signal and image processing
    - image deblurring, image inpainting,
    - channel equalization, signal separation,
    - tomography, MRI
  - ✓ sparsity & under-determined inverse problems
    - relation to subset selection problem
- **Pursuit Algorithms**
  - ✓ Greedy algorithms: Matching Pursuit & al
  - ✓ L1 minimization principles
  - ✓ L1 minimization algorithms
  - ✓ (Complexity of Pursuit Algorithms)

Overview of the course

- **Recovery guarantees for Pursuit Algorithms**
  - ✓ Well-posedness
  - ✓ Coherence *vs* Restricted Isometry Constant
  - ✓ Worked examples
  - ✓ Summary

Exercice at home

- Write Matlab code for MP
- Idem for OMP
- Idem for L1 minimization with CVX
- Idem for Iterative Hard Thresholding

Exercice: Matlab code for (O)MP

- Full clean code would include some checking (column normalization, dimension checking, etc.)

```
function [x res] = mp(b,A,k)
% explain here what the function should do
....
end

function [x res] = omp(b,A,k)
% explain here what the function should do
....
end
```

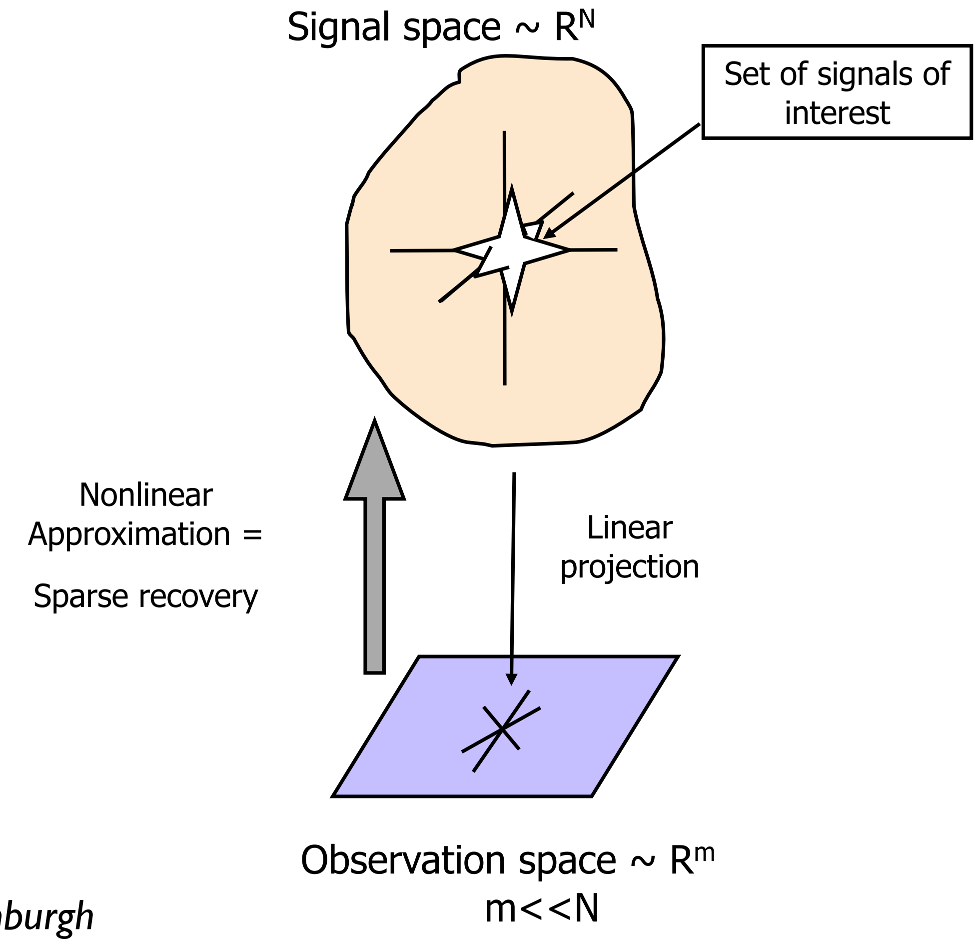
Exercice: Matlab code for (O)MP

- Full clean code would include some checking (column normalization, dimension checking, etc.)

```
function [x res] = mp(b,A,k)
% compute k-sparse approximation to b with matrix A using Matching pursuit
[m,N] = size(A);
x = zeros(N,1);
res = b;
for i=1:k
    % compute correlation between residual and columns of A
    corr = A'*res;
    % find position n (and value c) of the maximally correlated column
    [c n] = max(abs(corr)); % NB: modern Matlab notation allows [~ n] = max(abs(corr))
    % update the representation
    x(n) = x(n) + corr(n);
    % update the residual
    res  = res - corr(n)*A(:,n)
end
```

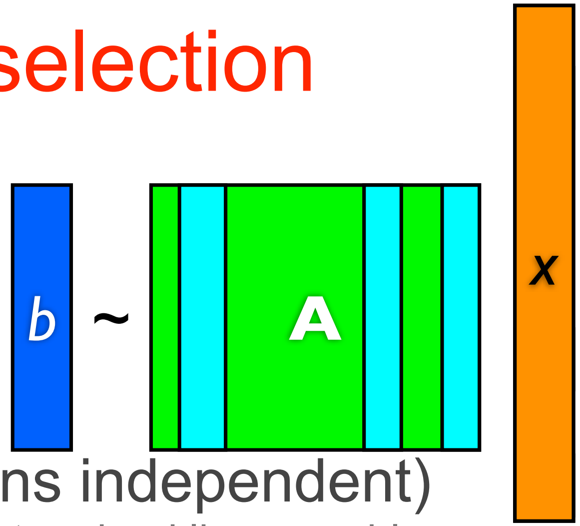
Sparse recovery: well-posedness

Inverse problems

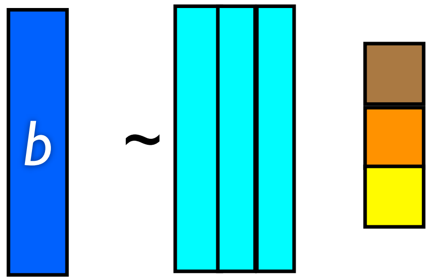


Courtesy: M. Davies, U. Edinburgh

Sparsity and subset selection



- Under-determined system
  - ✓ Infinitely many solutions
- If vector is sparse:
  - ✓ If support is known (and columns independent)
    - ✦ nonzero values characterized by (over)determined linear problem
  - ✓ **If support is unknown**
    - ✦ Main issue = finding the support!
    - ✦ This is the **subset selection problem**



- Objectives of the course
    - ✦ Efficient subset selection algorithms = **pursuit algorithms**
    - ✦ **Well-posedness** of subset selection
    - ✦ **Stability guarantees** of pursuits

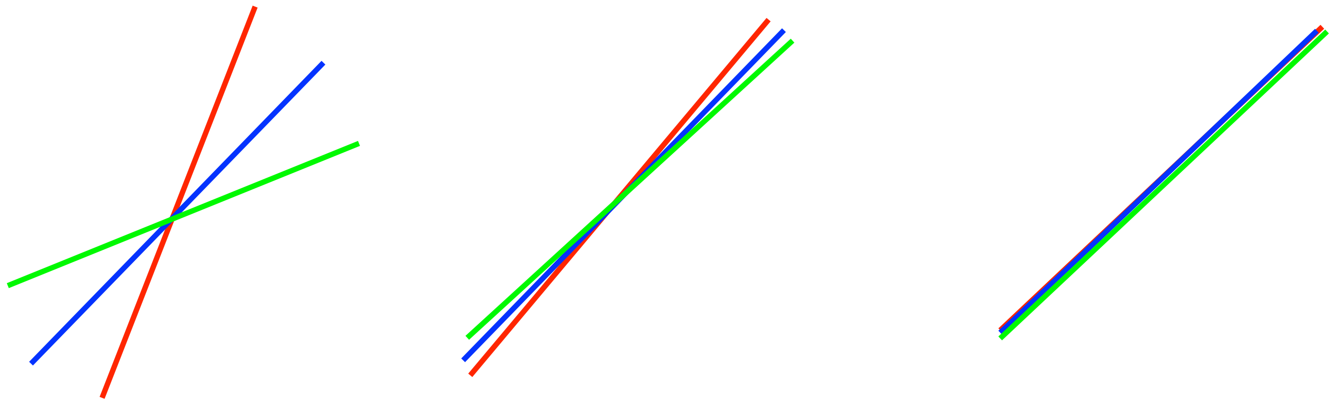
Well-posedness ?

- What property should **A** satisfy such that, for any pair of k-sparse vectors x_0, x_1

$$\mathbf{A}x_0 = \mathbf{A}x_1 \Rightarrow x_0 = x_1$$

Identifiability

- Identifiability of *1-sparse* vectors, with **A** 2x3 matrix



Identifiable ? Identifiable ? Identifiable?

- Here (k=1): identifiable iff every pair of columns is linearly independent

Well-posedness = identifiability of *k*-sparse vectors

- **Theorem**: if every 2*k* columns of **A** are linearly independent, then for every *k*-sparse vectors x_0, x_1

$$\mathbf{A}x_0 = \mathbf{A}x_1 \Rightarrow x_0 = x_1$$

- **Proof**: define the vector $z = x_0 - x_1$
  - ✓ Its **support** $I := \{i : z_i \neq 0\}$ is of size at most 2*k*

$$\sharp I = \|z\|_0 \leq \|x_0\|_0 + \|x_1\|_0 \leq 2k$$

  - ✓ It is in the **null space** of **A** hence $\sum_{i \in I} z_i \mathbf{a}_i = \mathbf{A}z = 0$
  - ✓ The columns indexed by *I* are linearly independent hence $z = 0$

Notions of spark / Kruskal rank

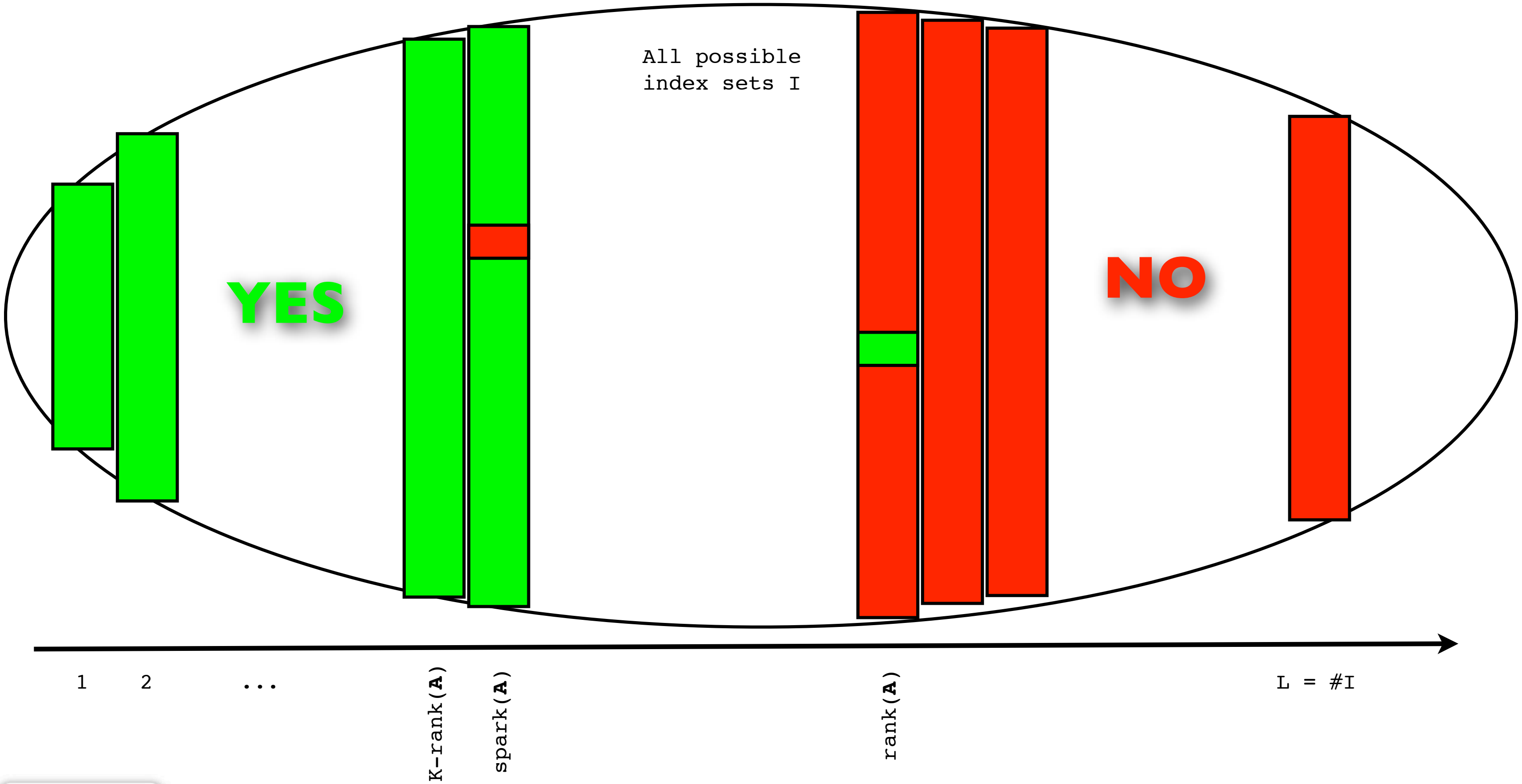
- **Definition**: spark(**A**)
  - ✓ size of minimal set of linearly dependent columns
- **Definition:** Kruskal rank K-rank(**A**):
  - ✓ maximal L such that every L columns linearly indep.
- **Property** $\text{K-rank}(\mathbf{A}) = \text{spark}(\mathbf{A}) - 1 \leq \text{rank}(\mathbf{A})$
- Well-posedness for *k*-sparse vectors iff

$$2k \leq \text{K-rank}(\mathbf{A})$$

- ... but the computation of K-rank for an arbitrary A is **NP-complete**

Linearly independent vs linearly dependent

are the columns of $\mathbf{A}_I$ linearly independent ?



Examples / Exercices

- **Definition:** Kruskal rank K-rank(**A**):
  - ✓ maximal L such that every L columns linearly indep.

- Small spark / Kruskal-rank
  - ✓ if **A** contains two copies of the same column

$$\text{K-rank}(\mathbf{A}) = ??$$

- Largest spark:
  - ✓ *m* x *N* «Vandermonde» matrix with $\omega_i \neq \omega_j, \forall i \neq j$

$$m < N$$

$$\mathbf{A} = \begin{pmatrix} \omega_1^0 & \dots & \omega_N^0 \\ \vdots & \dots & \vdots \\ \omega_1^{m-1} & \dots & \omega_N^{m-1} \end{pmatrix}$$

$$\text{K-rank}(\mathbf{A}) = ??$$

NB: by convention here $0^0 = 1$

  - ✓ Random Gaussian matrix: $\mathbf{A} = (a_{ij}) \quad a_{ij} \sim \mathcal{N}(0,1)$
    - ✦ with probability one:

$$\text{K-rank}(\mathbf{A}) = ??$$

Success of Ideal Sparse Approximation

- **Theorem**: if $2k \leq \text{K-rank}(\mathbf{A})$ then
  - ✓ Well-posedness
    - ✦ for every pair of k-sparse vectors x_0, x_1

$$\mathbf{A}x_0 = \mathbf{A}x_1 \Rightarrow x_0 = x_1$$

  - ✓ Recovery by L0 minimization
    - ✦ for every k-sparse vector x_0 we have

$$x_0 = \arg\min_x \|x\|_0 \text{ s.t. } \mathbf{A}x = \mathbf{A}x_0$$

Sparse recovery:
Provably good (efficient) algorithms?

Monte-Carlo simulations

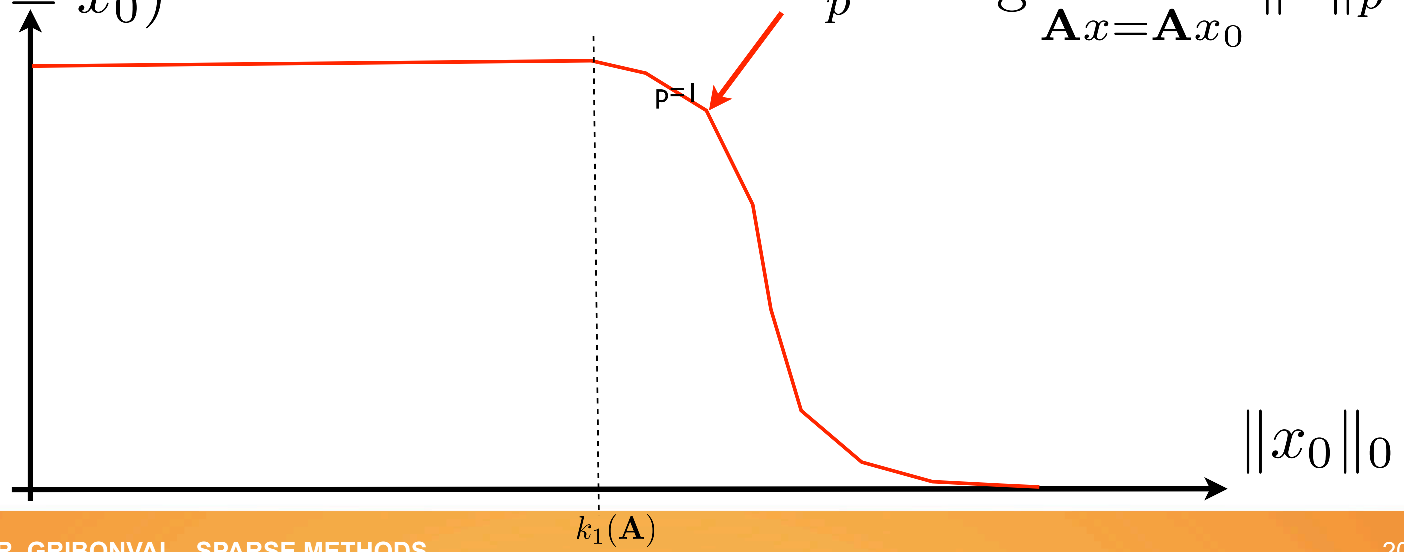
$$P(x_0) \xrightarrow{\text{draw ground truth}} x_0 \xrightarrow{\text{direct model}} \mathbf{b} := \mathbf{A}x_0$$

inverse problem

$$x_p^\star = \arg \min_{\mathbf{A}x = \mathbf{A}x_0} \|x\|_p$$

Typical observation

$P(x^\star = x_0)$



Exercice at home

- Implement in Matlab / Scilab:
  - ✓ Matching Pursuit (MP), Orthonormal MP (OMP)
  - ✓ Basis Pursuit = L1 minimization [with CVX] (BP)
- Generate test problems
  - ✓ Create matrix A (random Gaussian, normalize columns)
  - ✓ Create k-sparse x and b=Ax
- Compute mp(b,A,k) / omp(b,A,k) / bp(b,A)
- Measure quality (SNR on x) & computation time
- Curves of success as function of sparsity k

Equivalence between L0, L1, OMP

- **Theorem** : assume that $\mathbf{b} = \mathbf{A}x_0$
  - ✓ if $\|x_0\|_0 \leq k_0(\mathbf{A})$ **then** $x_0 = x_0^\star$
  - ✓ if $\|x_0\|_0 \leq k_1(\mathbf{A})$ **then** $x_0 = x_1^\star$

where $x_p^\star = \arg \min_{\mathbf{A}x = \mathbf{A}x_0} \|x\|_p$

- *Donoho & Huo 01 : pair of bases, coherence*
- *Donoho & Elad, Gribonval & Nielsen 2003 : dictionary, coherence*
- *Tropp 2004 : Orthonormal Matching Pursuit, cumulative coherence*
- *Candes, Romberg, Tao 2004 : random dictionaries, restricted isometry constants*