



Inverse problems and sparse models (1/2)

Rémi Gribonval

INRIA Rennes - Bretagne Atlantique, France

remi.gribonval@inria.fr

Structure of the tutorial

- **Session 1:**
 - ✓ Introduction to inverse problems & sparse models
 - ✓ Sparse recovery:
 - ◆ Well-posedness
 - ◆ Complexity
 - ✓ Pursuits
 - ◆ Optimization principles
 - ◆ Greedy algorithms
- **Session 2: Recovery guarantees**
 - ✓ Coherence
 - ✓ Restricted Isometry Property
 - ✓ Null Space Properties

Further material on sparsity

- Books

- ✓ Signal Processing perspective

- ◆ S. Mallat, «Wavelet Tour of Signal Processing», 3rd edition, 2008
 - ◆ M. Elad, «Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing», 2009.

- ✓ Mathematical perspective

- ◆ S. Foucart, H. Rauhut, «A Mathematical Introduction to Compressed Sensing», Springer, in preparation.

- Review paper:

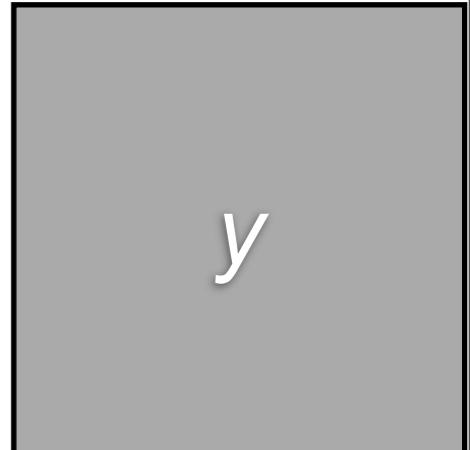
- ◆ Bruckstein, Donoho, Elad, SIAM Reviews, 2009

Inverse problems

Example: Inpainting Problem

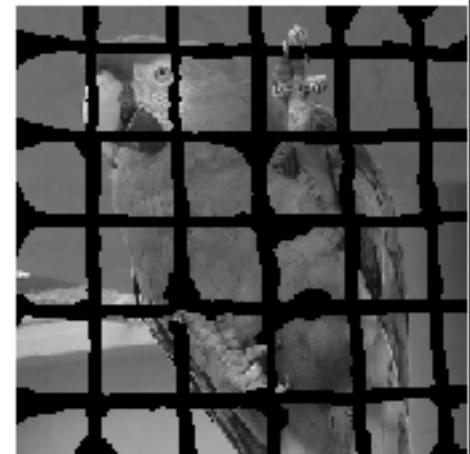
- Unknown image with N pixels

$$\mathbf{y} \in \mathbb{R}^N$$



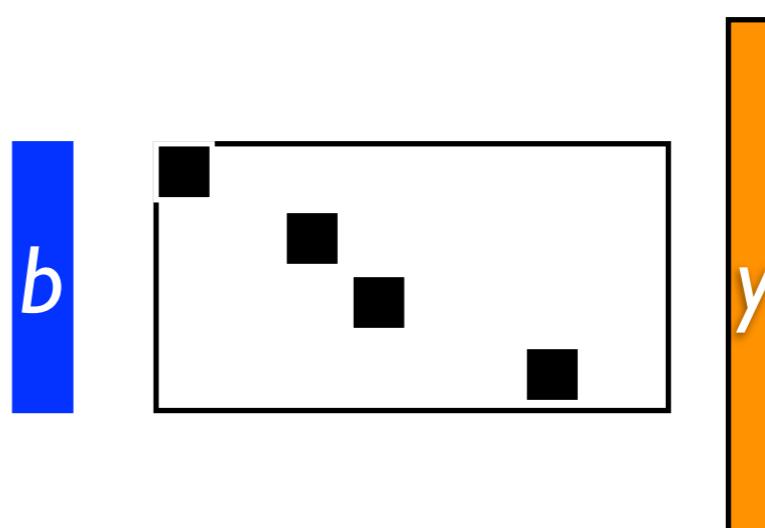
- Partially observed image:
 - ✓ $m < N$ observed pixels

$$b[\vec{p}] = y[\vec{p}], \vec{p} \in \text{Observed}$$



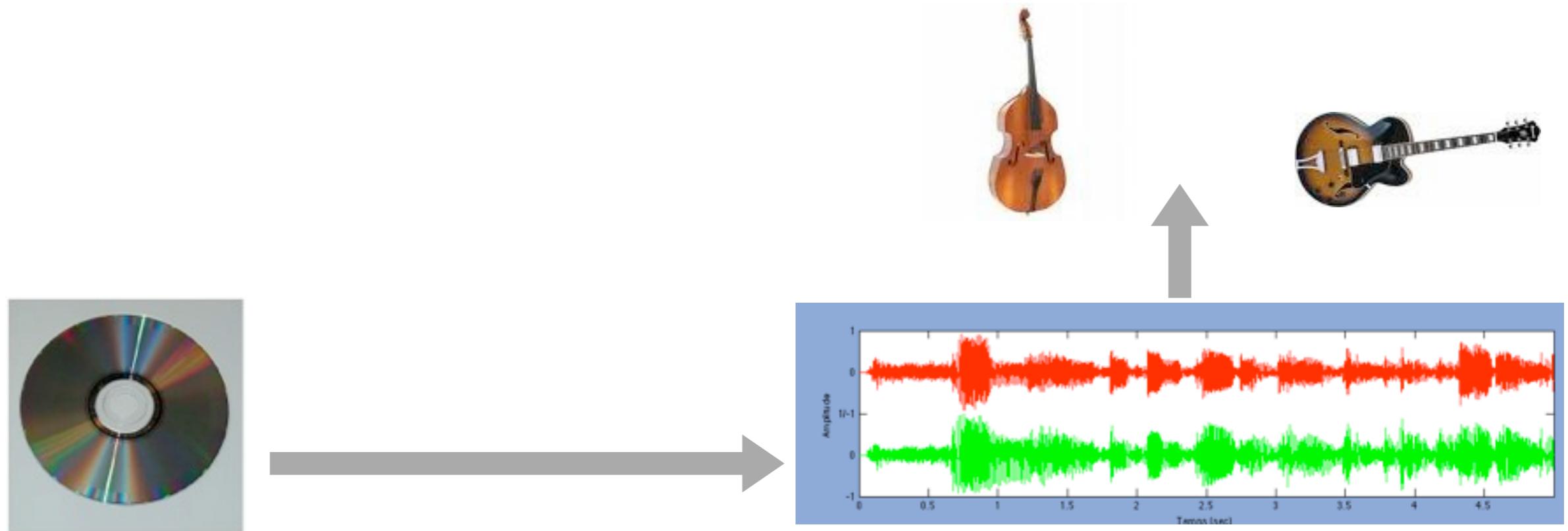
- Measurement matrix

$$\mathbf{b} = \mathbf{M}\mathbf{y}$$



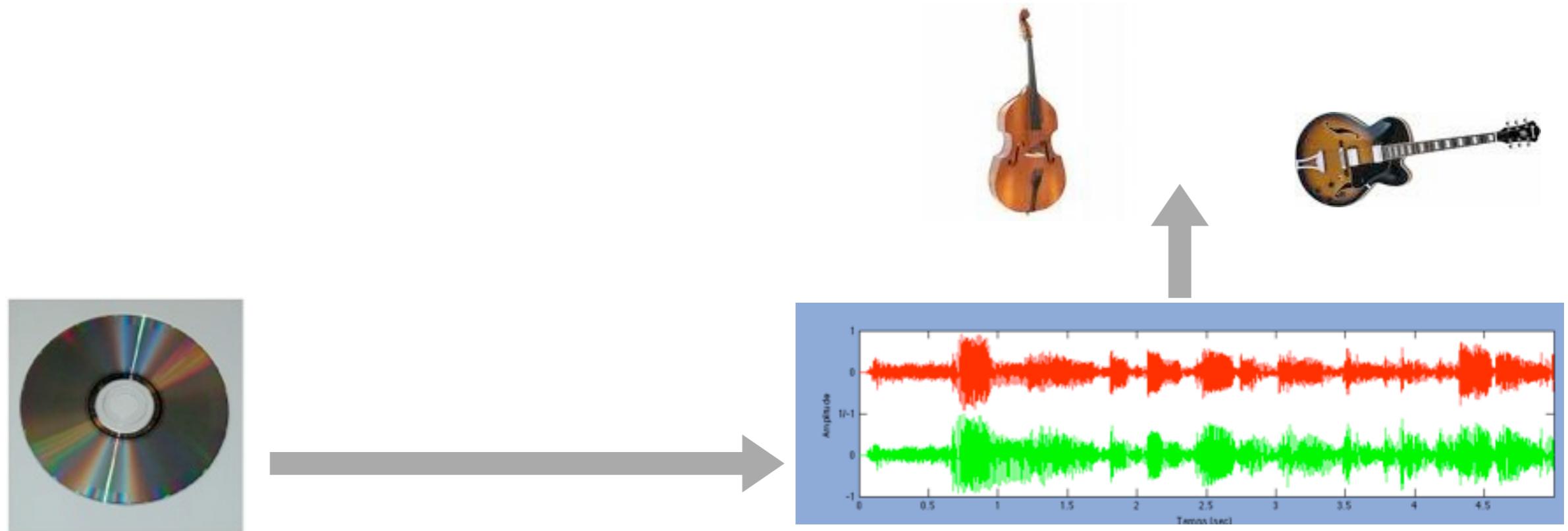
Example : audio source separation

- « Softly as in a morning sunrise »



Example : audio source separation

- « Softly as in a morning sunrise »



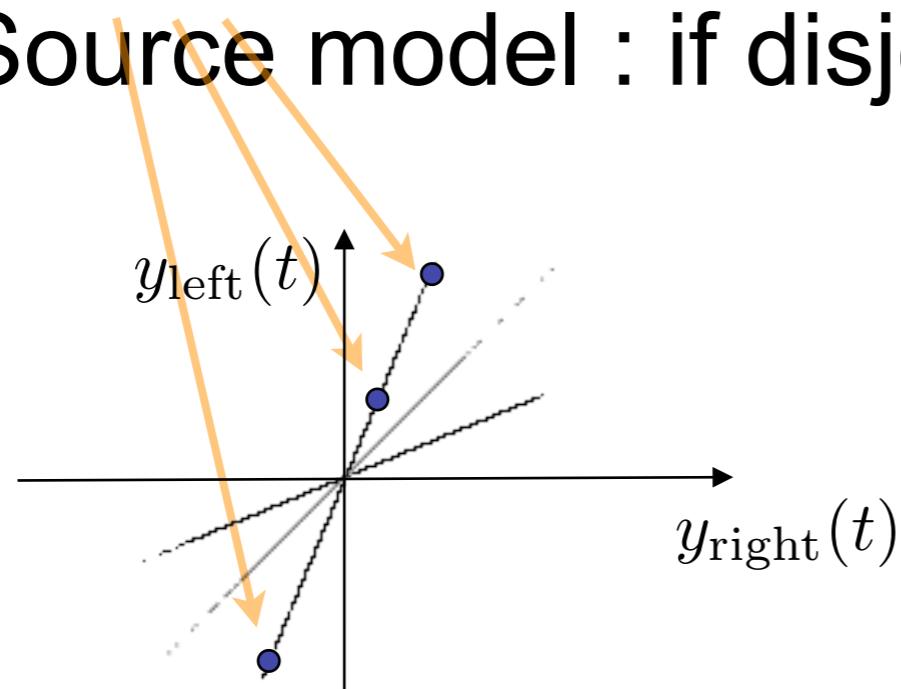
Blind Source Separation

- Mixing model : linear instantaneous mixture

$$\begin{matrix} y_{\text{right}}(t) \\ y_{\text{left}}(t) \end{matrix} = \mathbf{A} \begin{matrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{matrix}$$

The diagram shows two mixed signals, $y_{\text{right}}(t)$ and $y_{\text{left}}(t)$, on the left, represented by blue waveforms. These are multiplied by a mixing matrix \mathbf{A} to produce three source signals, $s_1(t)$, $s_2(t)$, and $s_3(t)$, on the right, also represented by blue waveforms.

- Source model : if disjoint time-supports ...



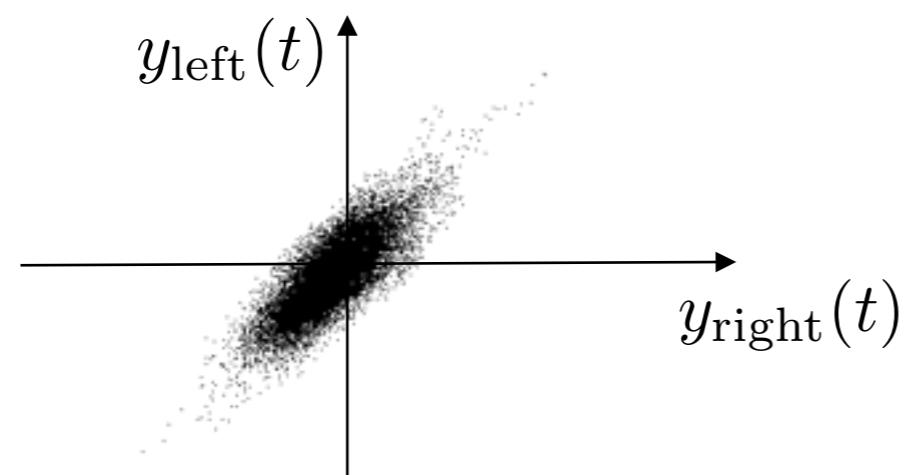
... then clustering to :
1- identify (columns of) the mixing matrix
2- recover sources

Blind Source Separation

- Mixing model : linear instantaneous mixture

$$\begin{matrix} y_{\text{right}}(t) \\ y_{\text{left}}(t) \end{matrix} \left(\begin{array}{c} \text{[sound波形]} \\ \text{[sound波形]} \end{array} \right) = \mathbf{A} \left(\begin{array}{c} \text{[sound波形]} \\ \text{[sound波形]} \\ \text{[sound波形]} \end{array} \right) \begin{matrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{matrix}$$

- In practice ...

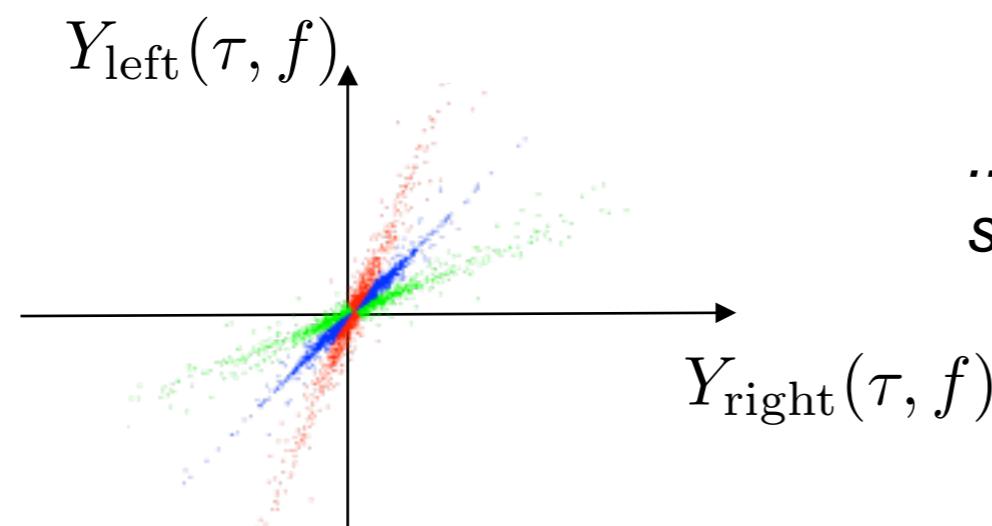


Time-Frequency Masking

- Mixing model in the time-frequency domain

$$\begin{matrix} Y_{\text{right}}(\tau, f) \\ Y_{\text{left}}(\tau, f) \end{matrix} \left(\begin{array}{c} \text{[Heatmap]} \\ \hline \text{[Heatmap]} \end{array} \right) = \mathbf{A} \mathbf{S}(\tau, f)$$

- And “miraculously” ...



*... time-frequency representations of audio signals are (often) **almost disjoint**.*

Inverse problems & Sparsity

Inverse problems

- **Inverse problem** : exploit indirect or incomplete observation to reconstruct some data

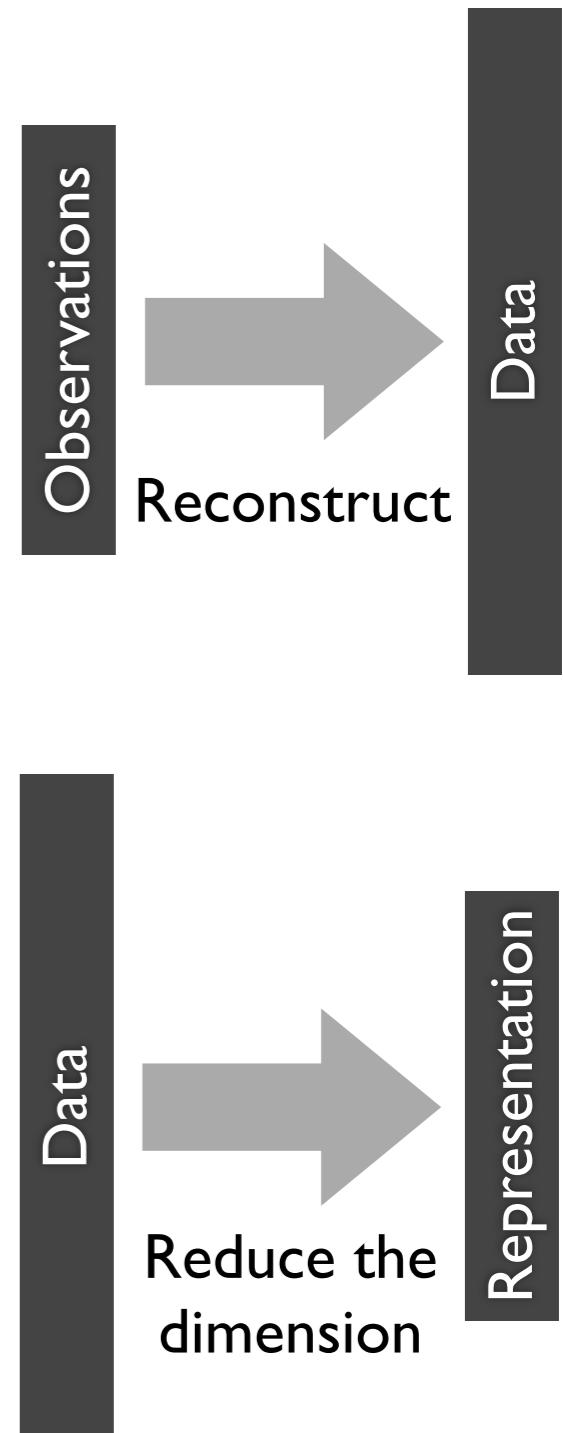
$$z = \mathbf{M}\mathbf{y}$$

↑
fewer equations than unknowns

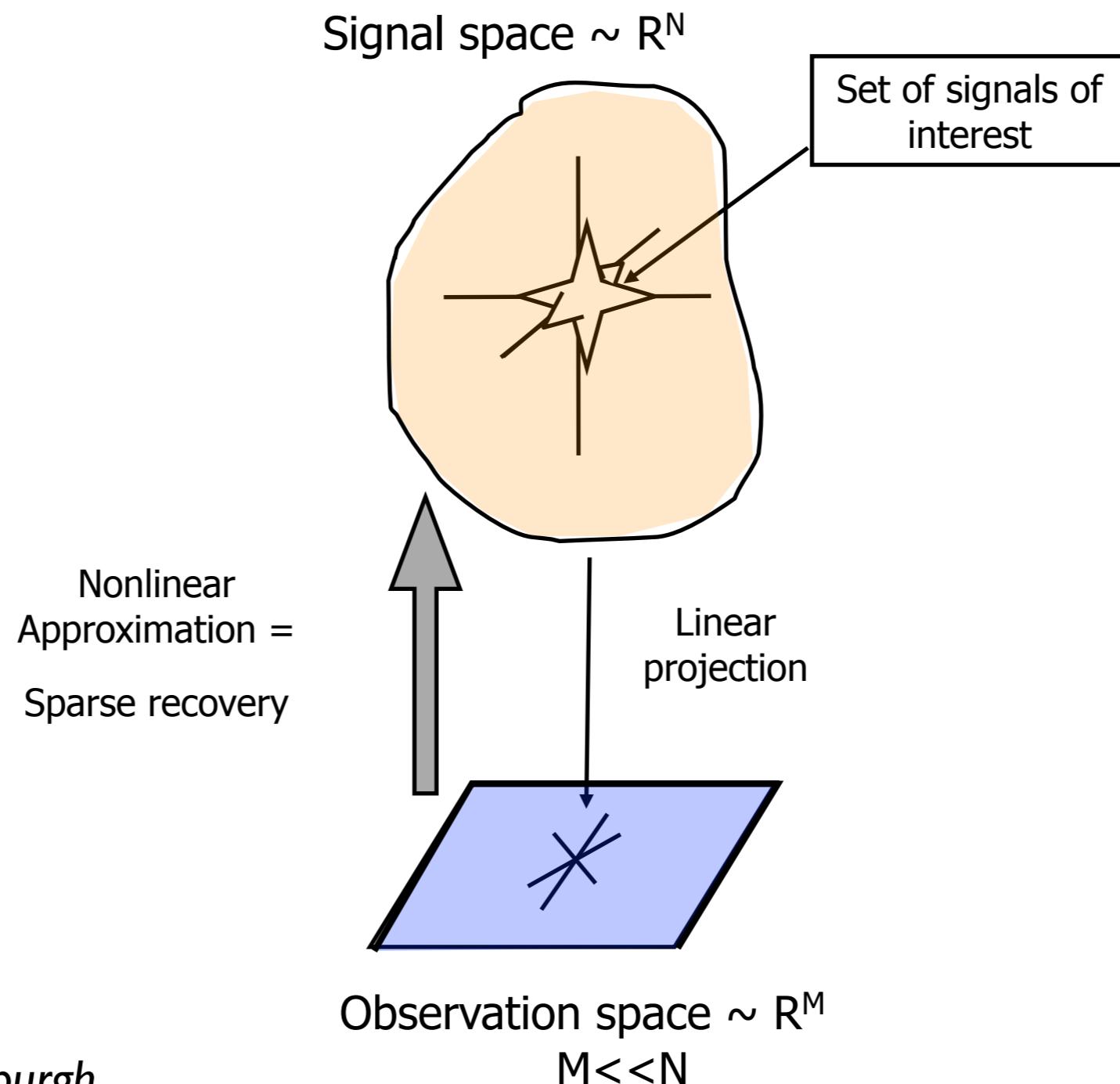
- **Sparsity** : represent / approximate high-dimensional & complex data using few parameters

$$\mathbf{y} \approx \Phi\mathbf{x}$$

↑
few nonzero components



Inverse problems



Courtesy: M. Davies, U. Edinburgh

Sparsity: definition

- A vector is
 - ✓ **sparse** if it has (many) zero coefficients
 - ✓ **k -sparse** if it has k nonzero coefficients

Sparsity: definition

- A vector is
 - ✓ **sparse** if it has (many) zero coefficients
 - ✓ **k -sparse** if it has k nonzero coefficients
- Symbolic representation as column vector

Not sparse



Sparsity: definition

- A vector is
 - ✓ **sparse** if it has (many) zero coefficients
 - ✓ **k -sparse** if it has k nonzero coefficients
- Symbolic representation as column vector

Not sparse



3-sparse



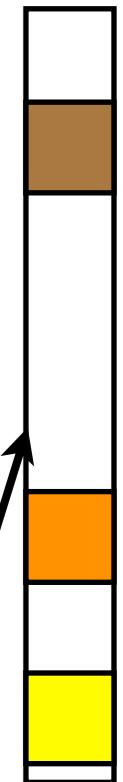
Sparsity: definition

- A vector is
 - ✓ **sparse** if it has (many) zero coefficients
 - ✓ **k -sparse** if it has k nonzero coefficients
- Symbolic representation as column vector
- **Support** = indices of nonzero components

Not sparse



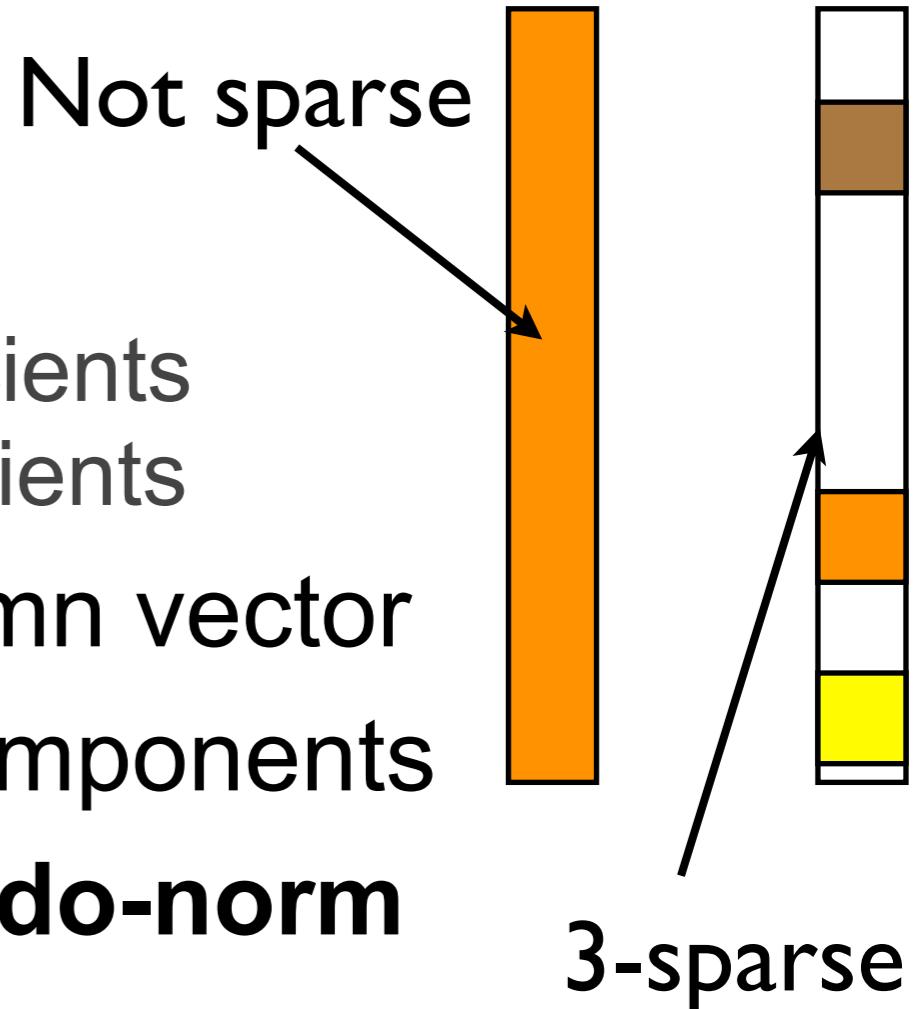
3-sparse



Sparsity: definition

- A vector is
 - ✓ **sparse** if it has (many) zero coefficients
 - ✓ **k -sparse** if it has k nonzero coefficients
- Symbolic representation as column vector
- **Support** = indices of nonzero components
- Sparsity measured with **L0 pseudo-norm**

$$\|x\|_0 := \#\{n, x_n \neq 0\} = \sum_n |x_n|^0$$

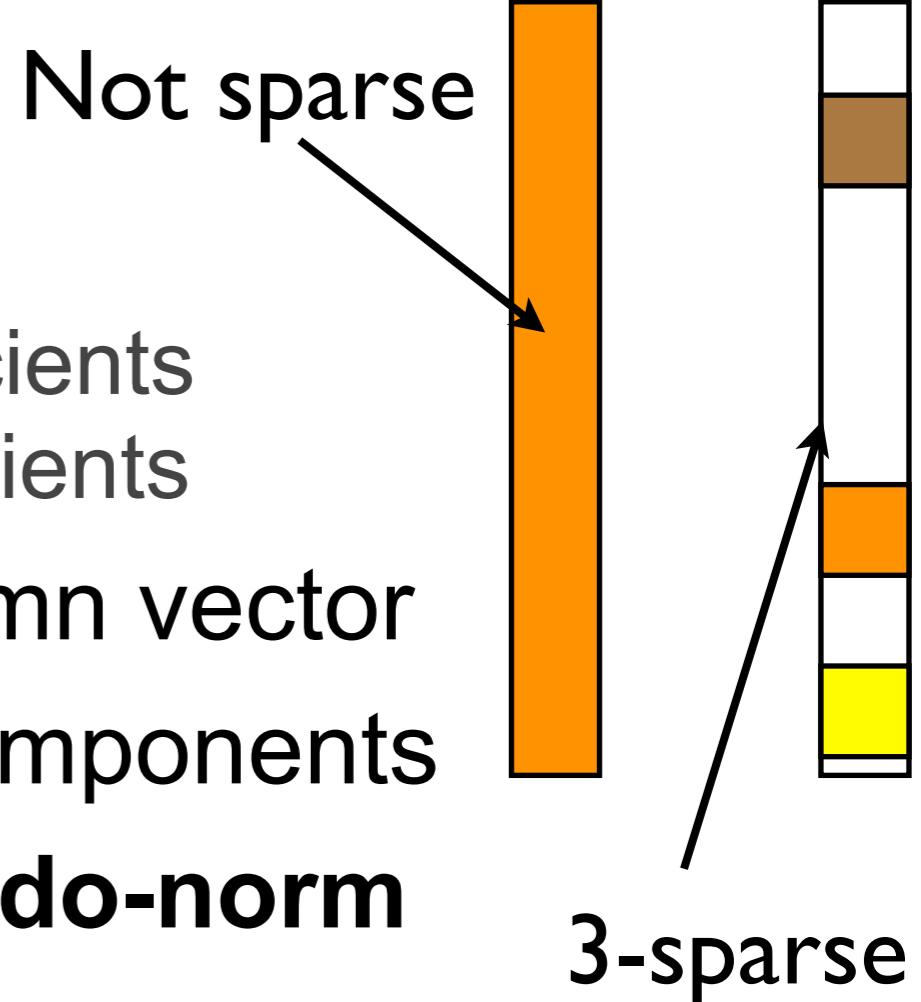


Sparsity: definition

- A vector is
 - ✓ **sparse** if it has (many) zero coefficients
 - ✓ **k -sparse** if it has k nonzero coefficients
- Symbolic representation as column vector
- **Support** = indices of nonzero components
- Sparsity measured with **L0 pseudo-norm**

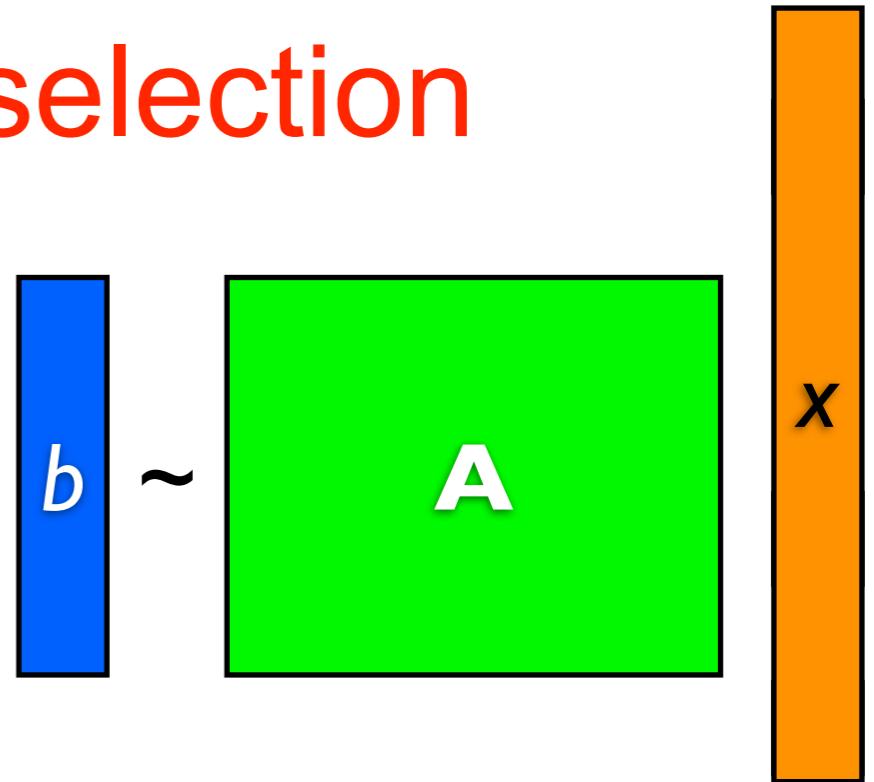
$$\|x\|_0 := \#\{n, x_n \neq 0\} = \sum_n |x_n|^0$$

- *In french:*
 - ◆ sparse → «creux», «parcimonieux»
 - ◆ sparsity, sparseness → «parcimonie», ~~«sparsité»~~



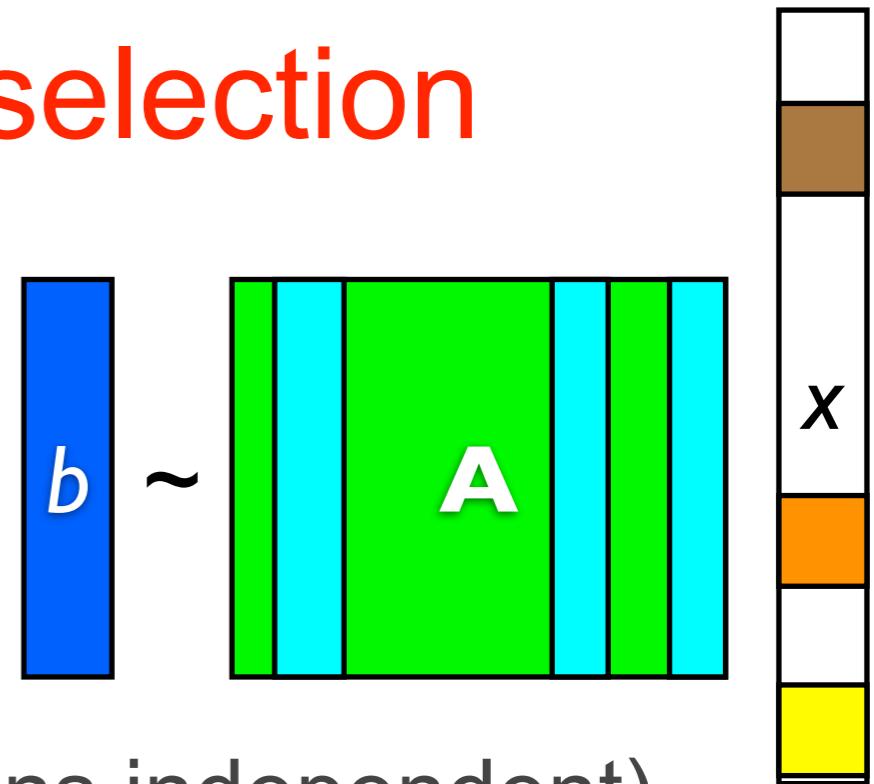
Sparsity and subset selection

- Under-determined system
 - ✓ Infinitely many solutions
- If vector is sparse:



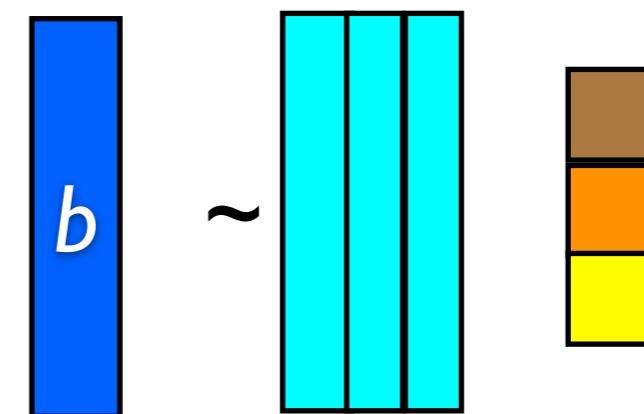
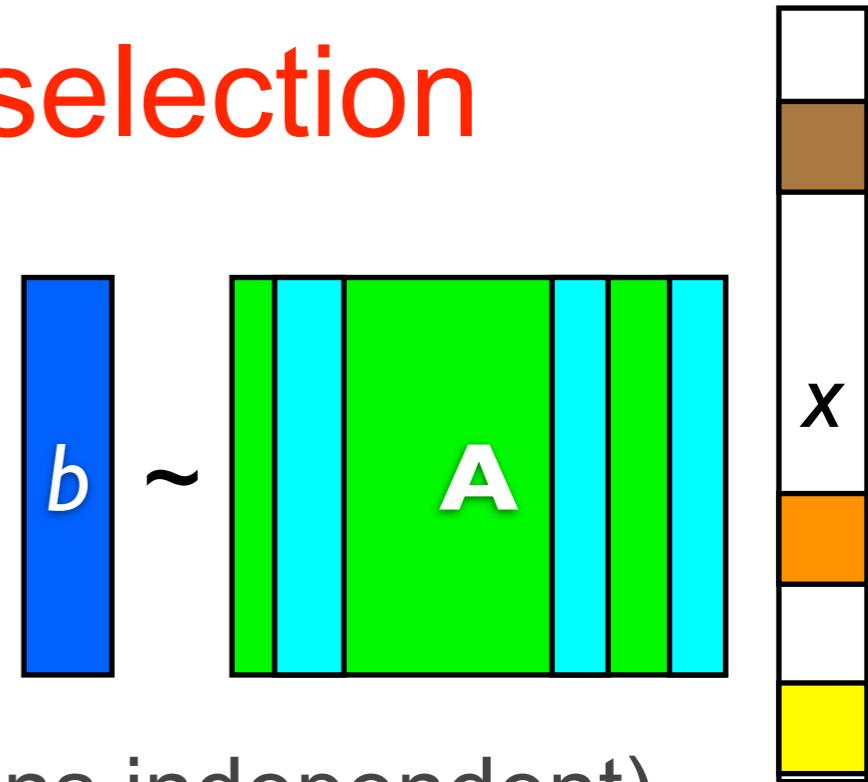
Sparsity and subset selection

- Under-determined system
 - ✓ Infinitely many solutions
- If vector is sparse:
 - ✓ If support is known (and columns independent)



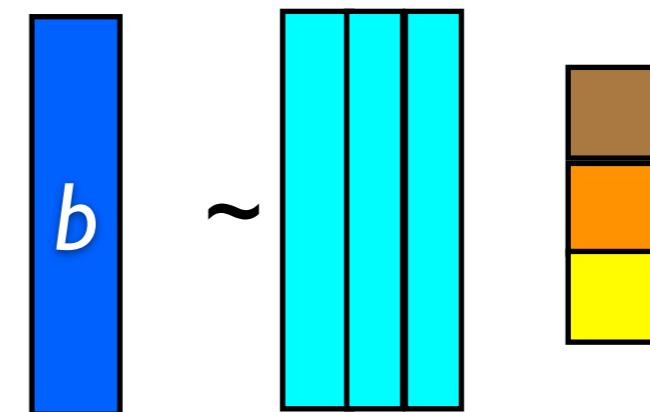
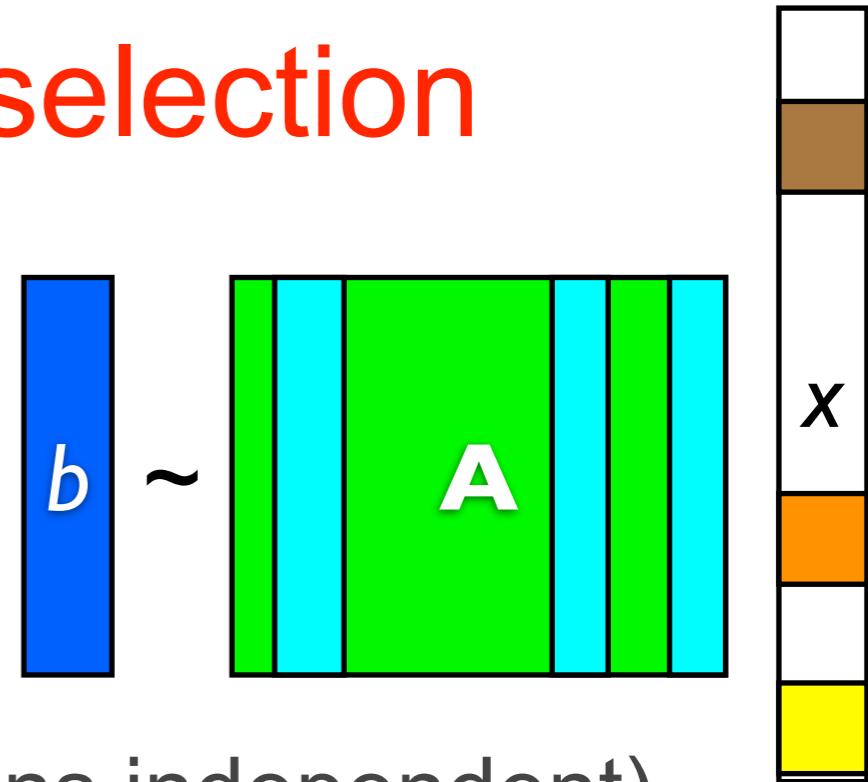
Sparsity and subset selection

- 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



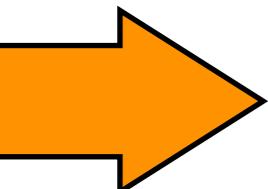
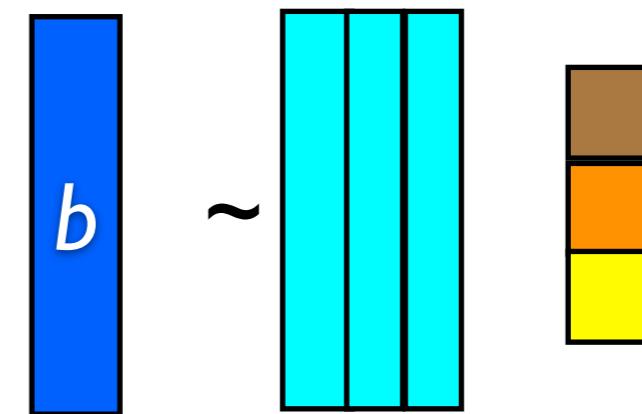
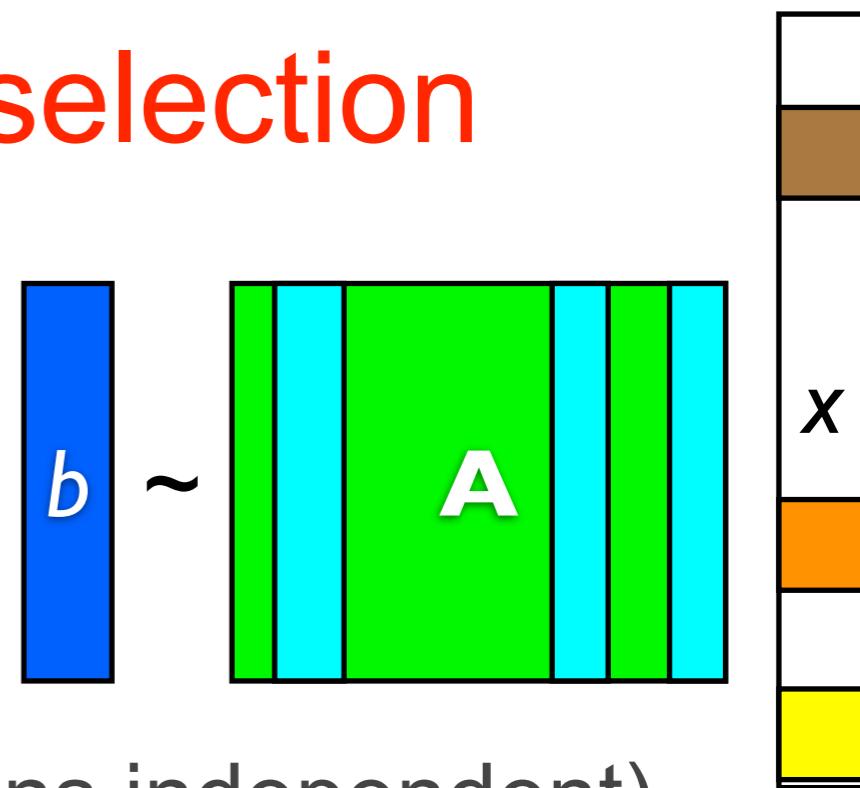
Sparsity and subset selection

- 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**



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 tutorial
 - ◆ **Well-posedness** of subset selection
 - ◆ Efficient subset selection algorithms = **pursuit algorithms**
 - ◆ **Stability guarantees** of pursuits



Inverse Problems & Sparsity: Mathematical foundations

- **Bottleneck 1990-2000 :**

- ✓ *Ill-posedness* when fewer equations than unknowns

$$\mathbf{A}x_0 = \mathbf{A}x_1 \not\Rightarrow x_0 = x_1$$

- **Novelty 2001-2006 :**

- ✓ *Well-posedness* = uniqueness of sparse solution:

- ◆ if x_0, x_1 are “sufficiently sparse”,

- ◆ then $\mathbf{A}x_0 = \mathbf{A}x_1 \Rightarrow x_0 = x_1$

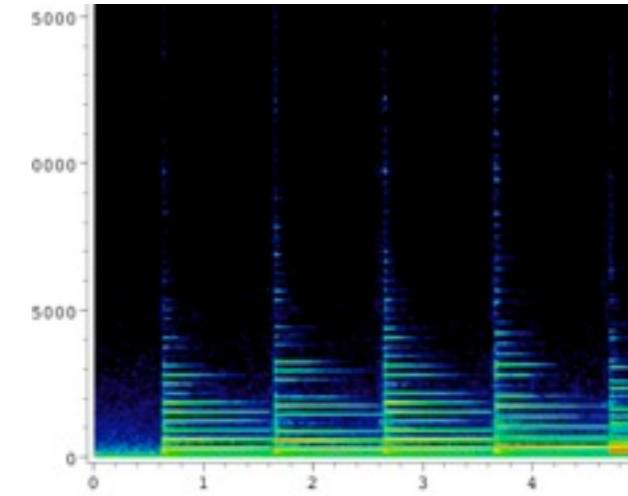
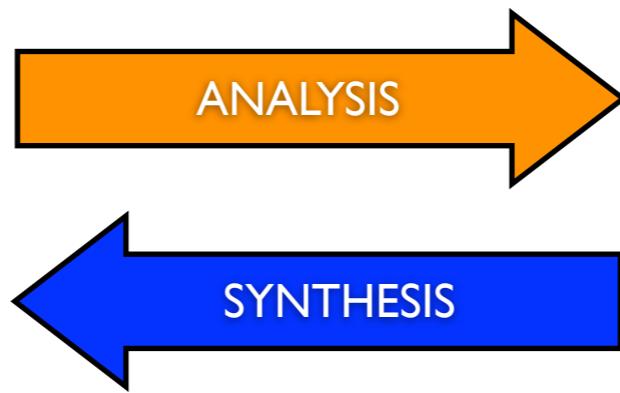
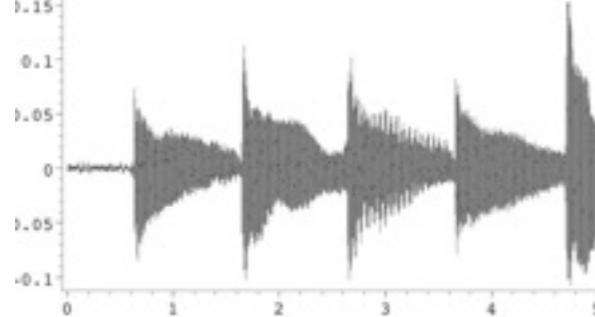
- ✓ *Recovery of x_0 with practical pursuit algorithms*

- ◆ Thresholding, Matching Pursuits, Minimisation of L_p norms $p \leq 1, \dots$

Sparse models & data compression

Relevance of the sparsity assumption

- Audio : time-frequency representations (MP3)

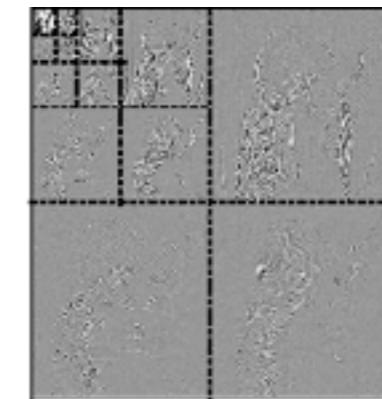
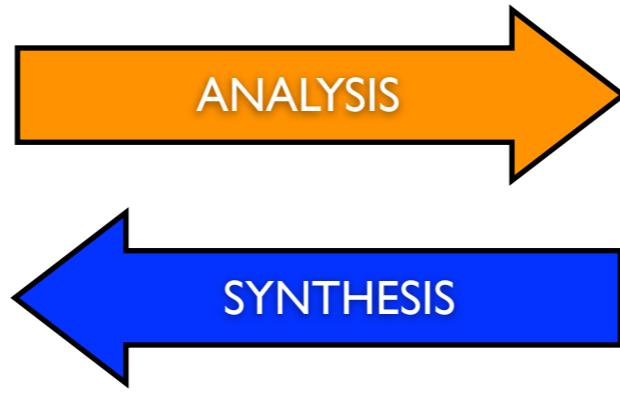


Black
= zero

- Images : wavelet transform (JPEG2000)



ORIGINAL
128, 129, 125, 64, 65,



TRANSFORM COEFFICIENTS
4123, -12.4, -96.7, 45,

Gray
= zero

Mathematical expression of the sparsity assumption

- Signal / image = high dimensional vector

$$y \in \mathbb{R}^N$$

- Definition:

- ✓ **Atoms:** basis vectors $\varphi_k \in \mathbb{R}^N$
 - ◆ ex: time-frequency atoms, wavelets

- ✓ **Dictionary:**
 - ◆ collection of atoms $\{\varphi_k\}_{1 \leq k \leq K}$

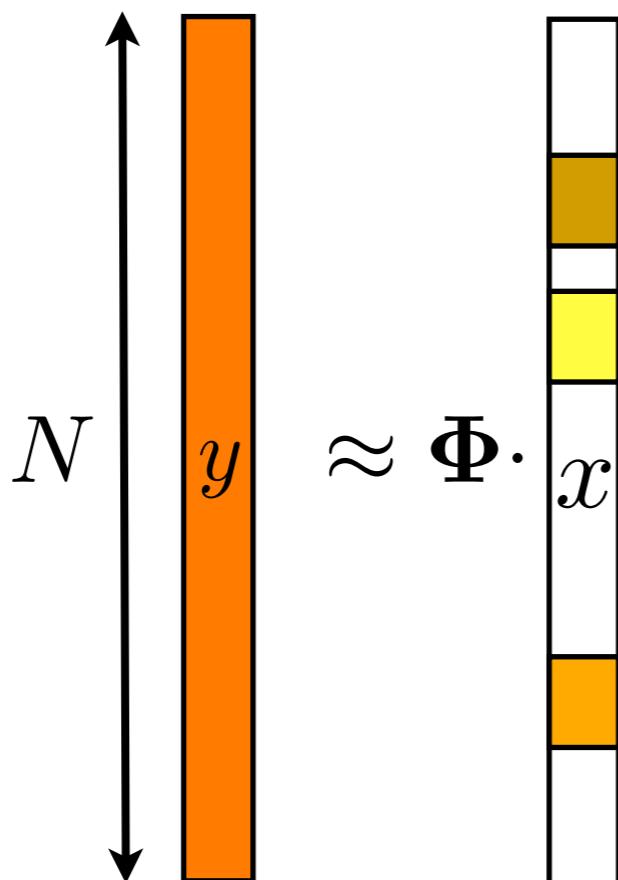
- ◆ matrix $\Phi = [\varphi_k]_{1 \leq k \leq K}$ which columns are the atoms

- Sparse signal model = combination of *few* atoms

$$y \approx \sum_k x_k \varphi_k = \Phi x$$

Sparsity & compression

- Full vector

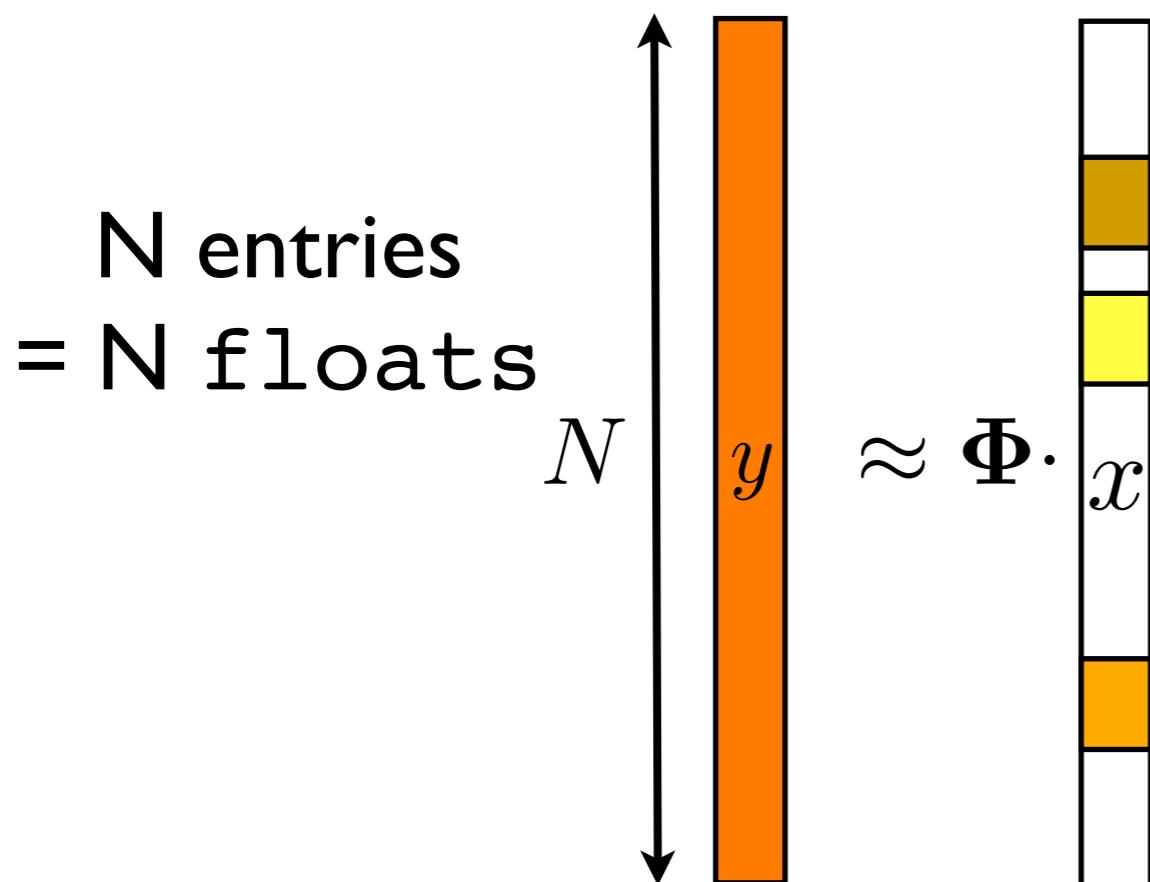


- Sparse vector

Key practical issues: choose dictionary

Sparsity & compression

- Full vector

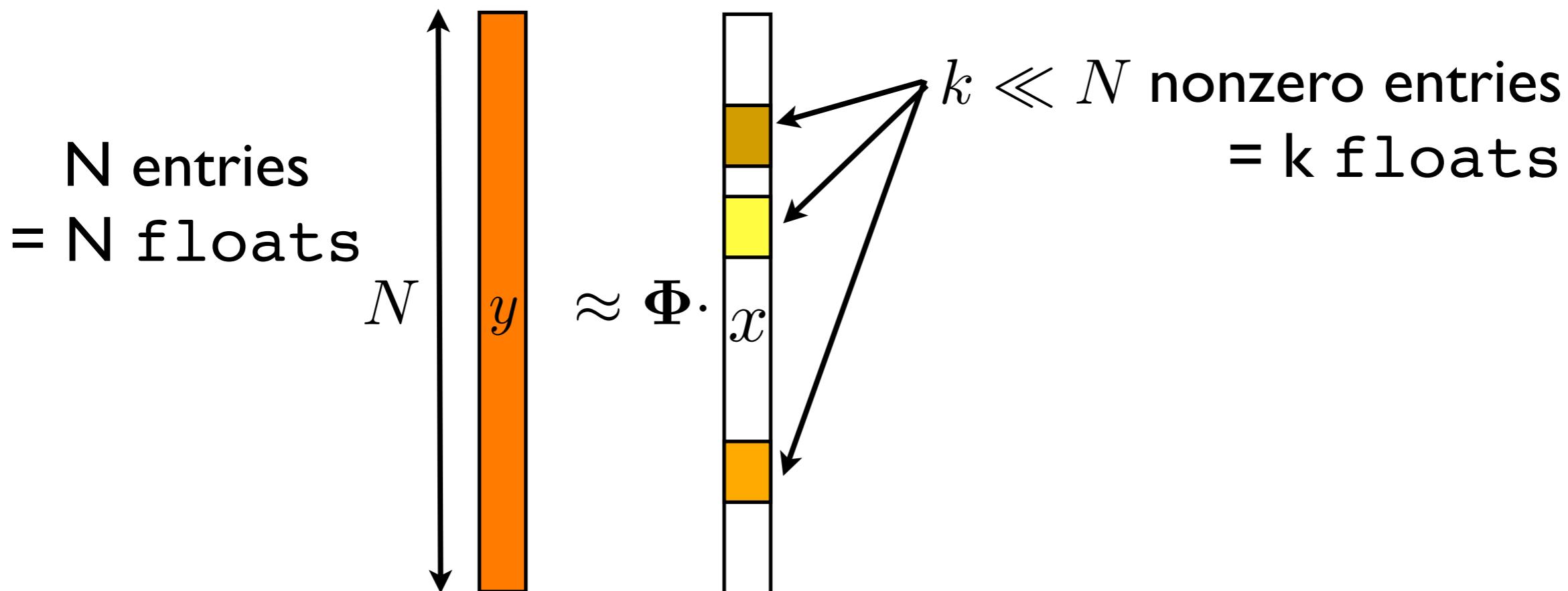


- Sparse vector

Key practical issues: choose dictionary

Sparsity & compression

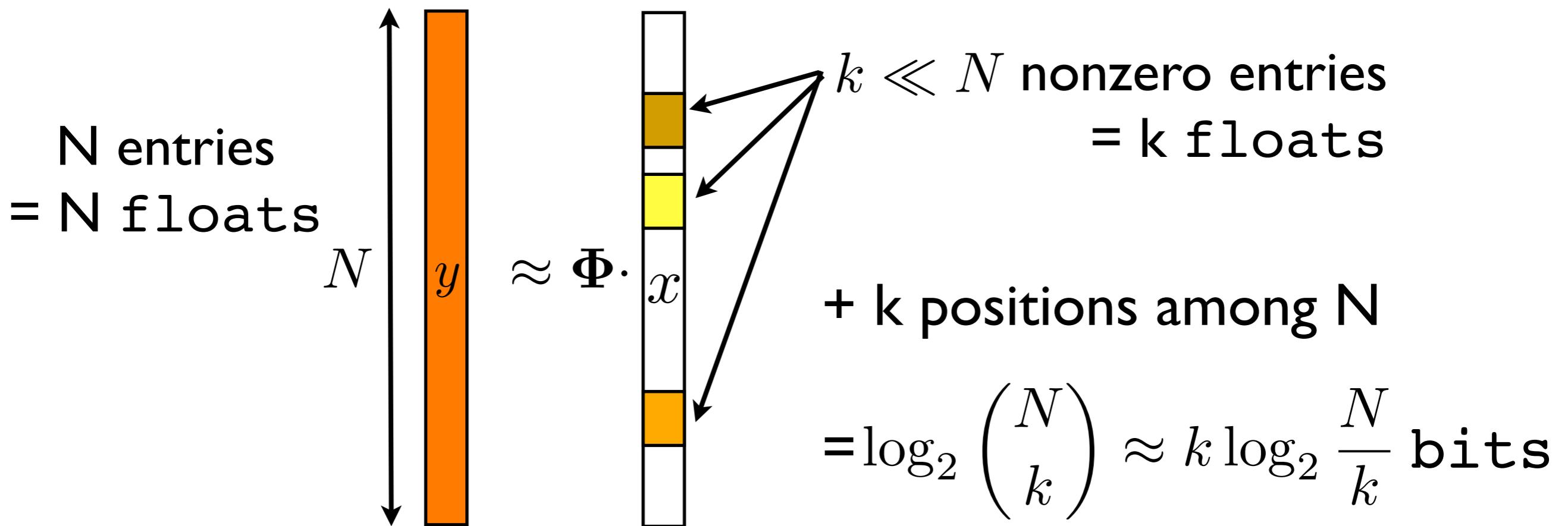
- Full vector



Key practical issues: choose dictionary

Sparsity & compression

- Full vector



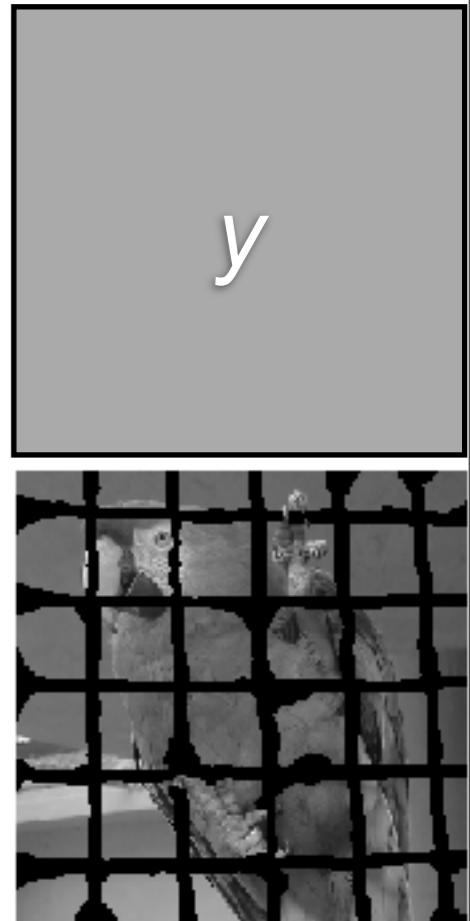
Key practical issues: choose dictionary

Sparse recovery: well-posedness

Example: Inpainting Problem

- Unknown image with N pixels

$$\mathbf{y} \in \mathbb{R}^N$$



- Sparse Model in wavelet domain

◆ wavelets coefficients are sparse

$$\mathbf{x} \approx \Phi^T \mathbf{y}$$

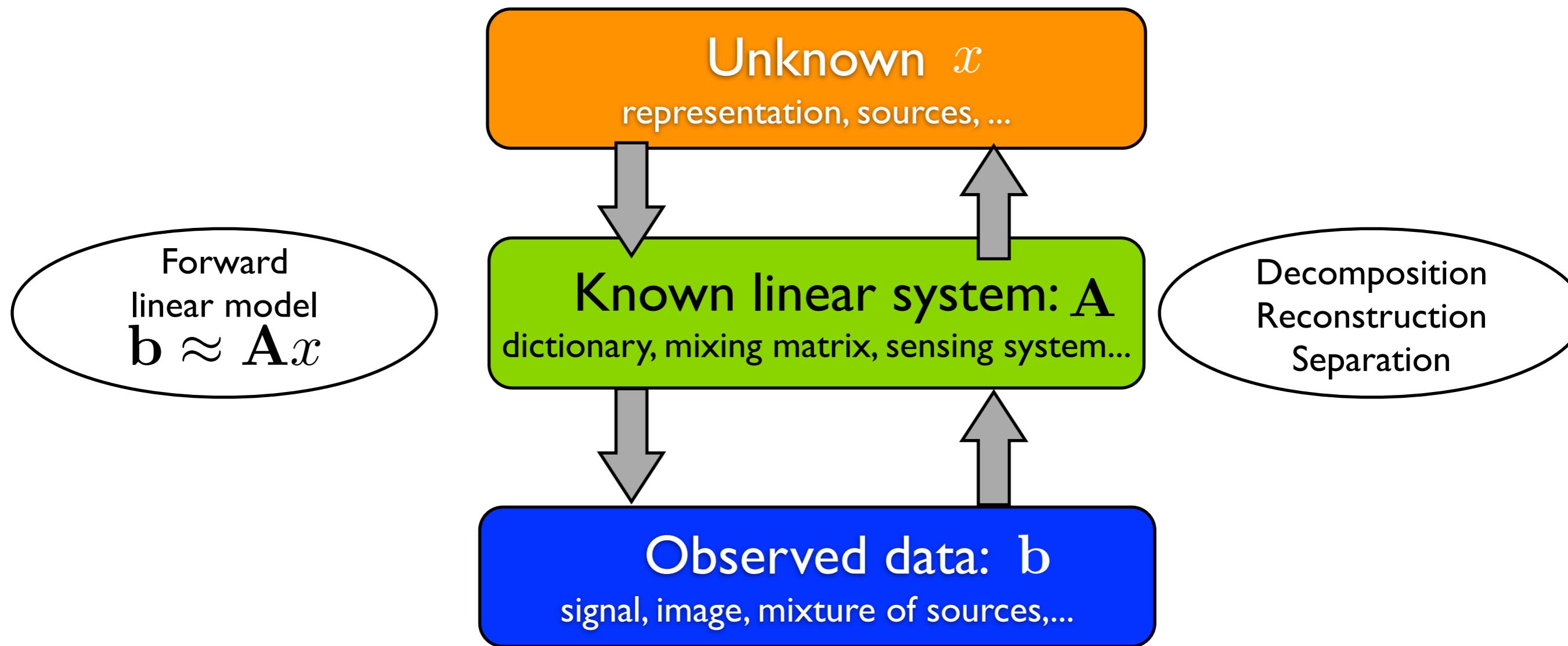
◆ sparse representation of unknown image $\mathbf{y} \approx \Phi \mathbf{x}$

- Measurement matrix

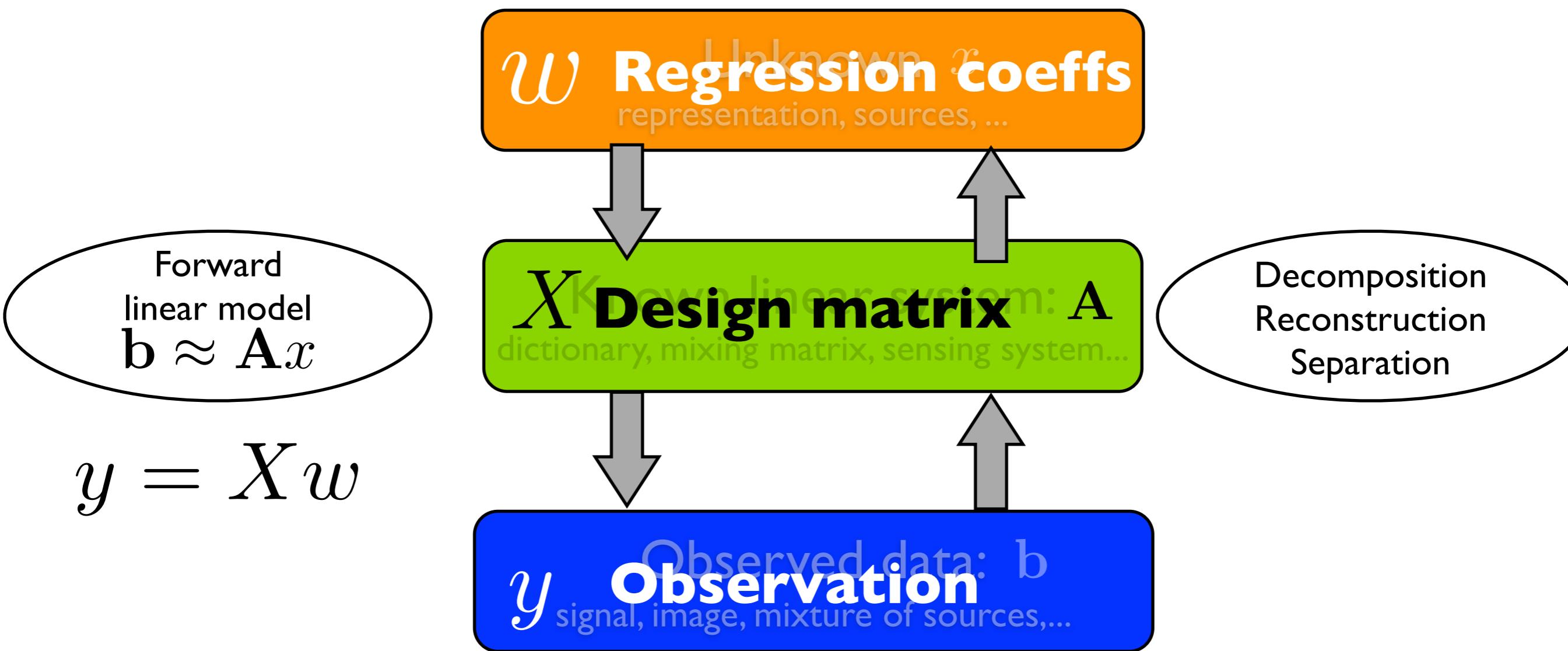
$$\mathbf{b} = \mathbf{M} \mathbf{y} \quad \mathbf{b} \approx \mathbf{M} \Phi \mathbf{x}$$

$$\mathbf{A} := \mathbf{M} \Phi$$

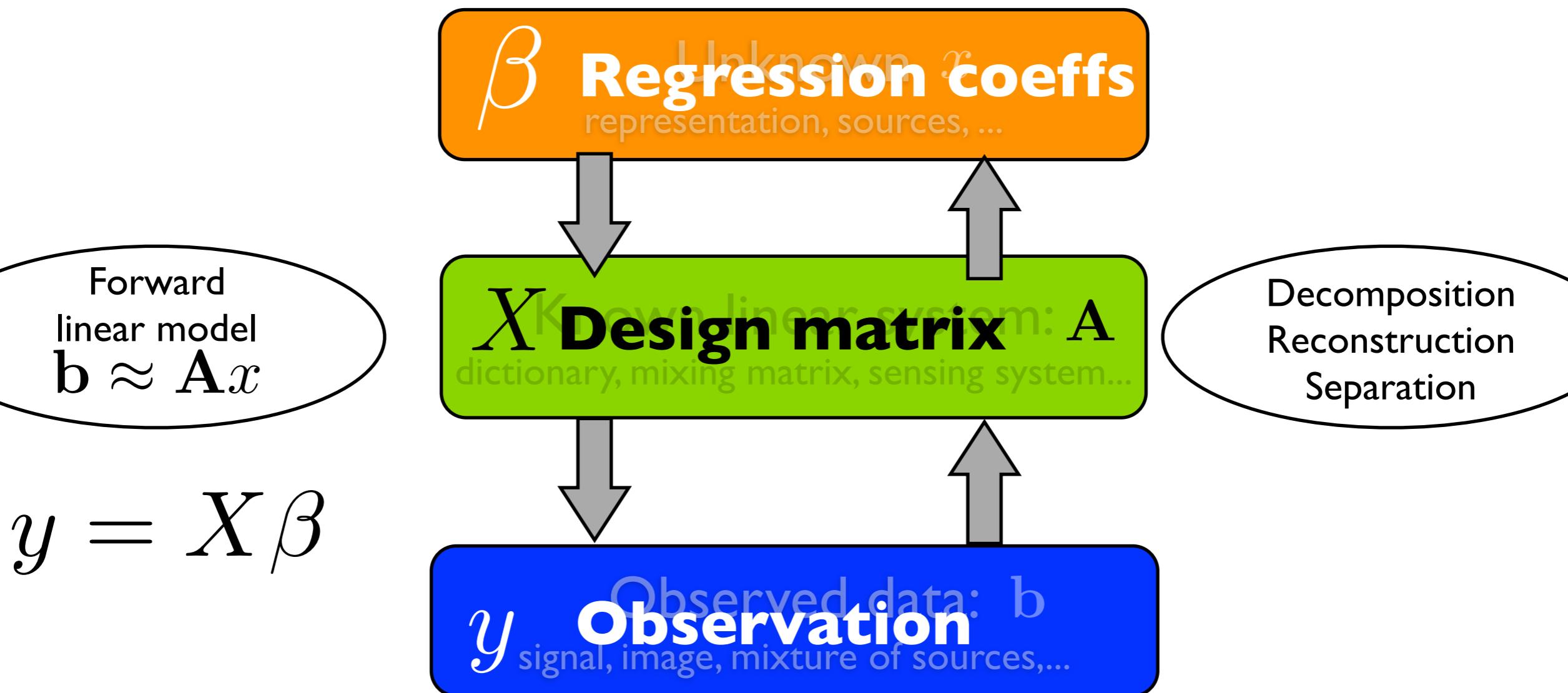
Signal Processing Vocabulary



(My Basic Understanding of) Machine Learning Vocabulary



(My Basic Understanding of) Statistics Vocabulary



Well-posedness ?

- What property should \mathbf{A} satisfy such that, for any pair of k -sparse vectors x_0, x_1

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

Well-posedness = identifiability of k -sparse vectors

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

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

Well-posedness = identifiability of k -sparse vectors

- **Theorem:** if every $2k$ columns of \mathbf{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 $2k$
$$\#I = \|z\|_0 \leq \|x_0\|_0 + \|x_1\|_0 \leq 2k$$
 - ✓ It is in the **null space** of \mathbf{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:** $\text{spark}(\mathbf{A})$
 - ✓ size of minimal set of linearly dependent columns
- **Definition:** Kruskal rank $\text{K-rank}(\mathbf{A})$:
 - ✓ 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**

Examples / Exercices

- **Definition:** Kruskal rank K-rank(\mathbf{A}):
 - ✓ maximal L such that every L columns linearly indep.
- Small spark / Kruskal-rank
 - ✓ if \mathbf{A} contains two copies of the same column
- Largest spark:
 - ✓ $m \times N$ «Vandermonde» matrix with $\omega_i \neq \omega_j, \forall i \neq j$
 $m < N$
$$\begin{pmatrix} \omega_1^1 & \dots & \omega_N^1 \\ \vdots & \dots & \vdots \\ \omega_1^m & \dots & \omega_N^m \end{pmatrix}$$
 - ✓ Random Gaussian matrix: $\mathbf{A} = (a_{ij})$ $a_{ij} \sim \mathcal{N}(0, 1)$
 - ◆ with probability one:
$$\text{K-rank}(\mathbf{A}) = ??$$

Sparse recovery: complexity

Sparse recovery ?

- Assuming well-posedness, how to compute the unique k -sparse solution to

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

Ideal sparse approximation

- Input:
 $m \times N$ matrix \mathbf{A} , with $m < N$, m -dimensional vector \mathbf{b}
- Objective:
find a sparsest approximation within given tolerance

$$\arg \min_x \|x\|_0, \text{ s.t. } \|\mathbf{b} - \mathbf{A}x\| \leq \epsilon$$

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

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

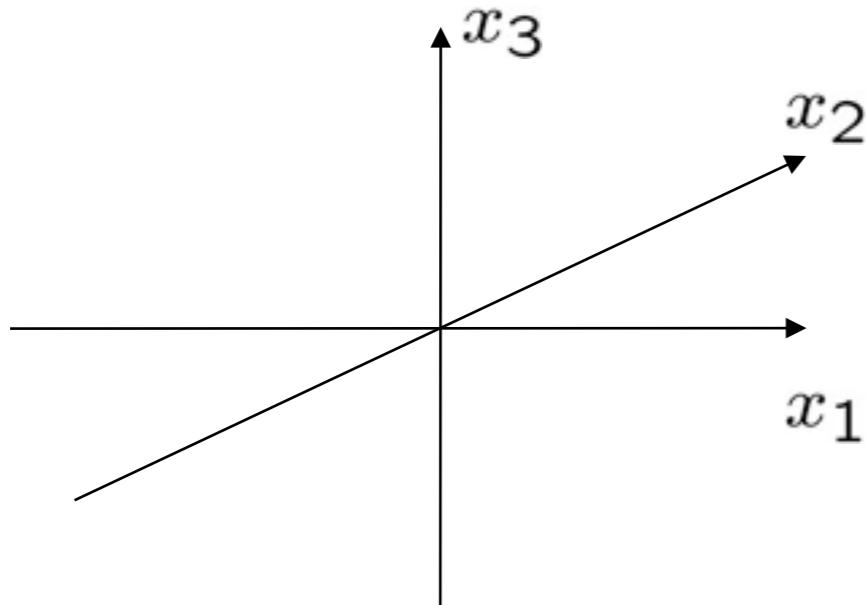
- ✓ Recovery by L0 minimization

- ◆ for every k -sparse vector x_0 we have

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

Complexity of Ideal Sparse Approximation

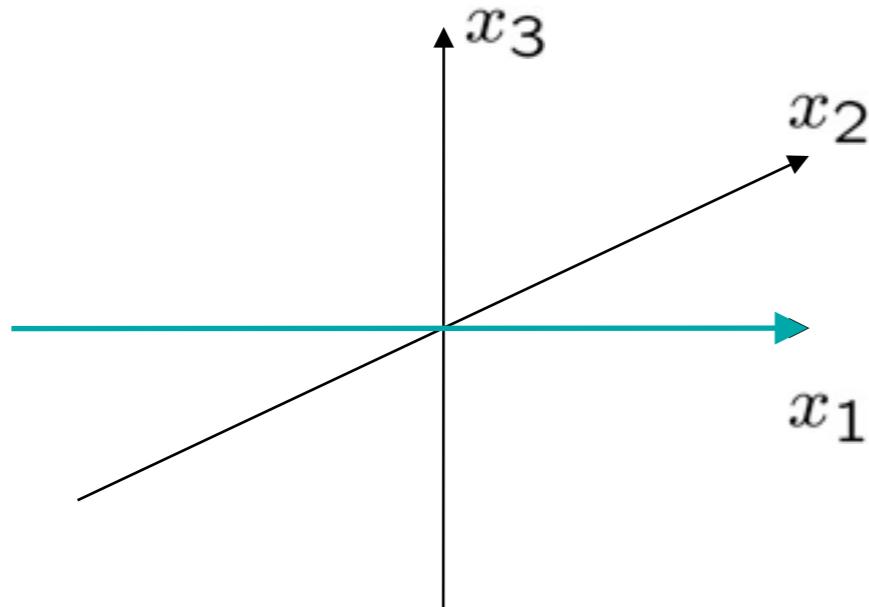
- Naive: Brute force search



- **Theorem** (*Davies et al, Natarajan*)
Solving the L0 optimization problem is NP-complete
- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

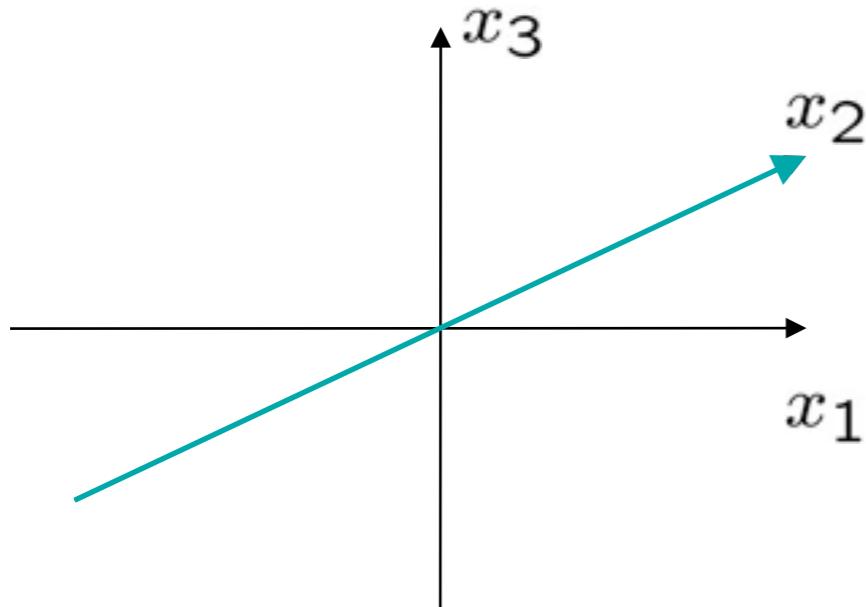
- Naive: Brute force search



- **Theorem** (*Davies et al, Natarajan*)
Solving the L0 optimization problem is NP-complete
- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

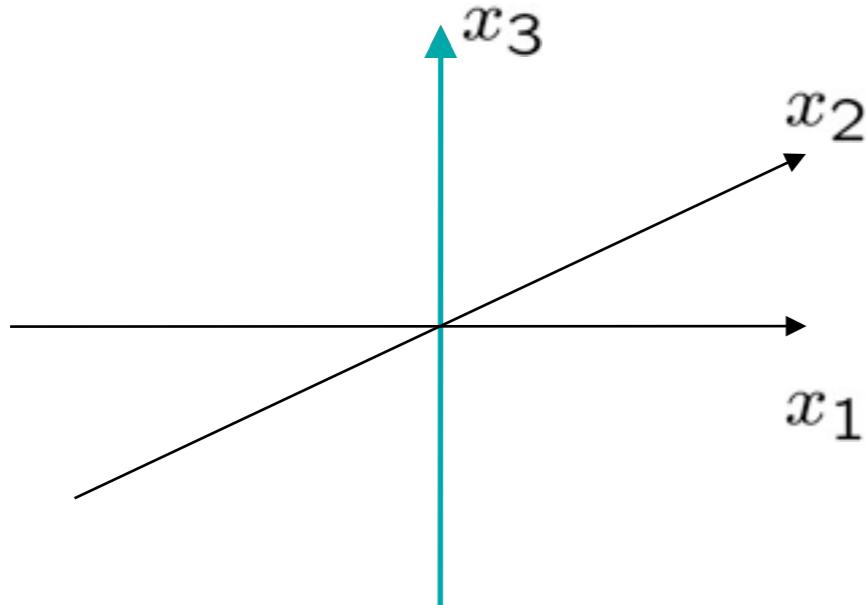
- Naive: Brute force search



- **Theorem** (*Davies et al, Natarajan*)
Solving the L0 optimization problem is NP-complete
- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

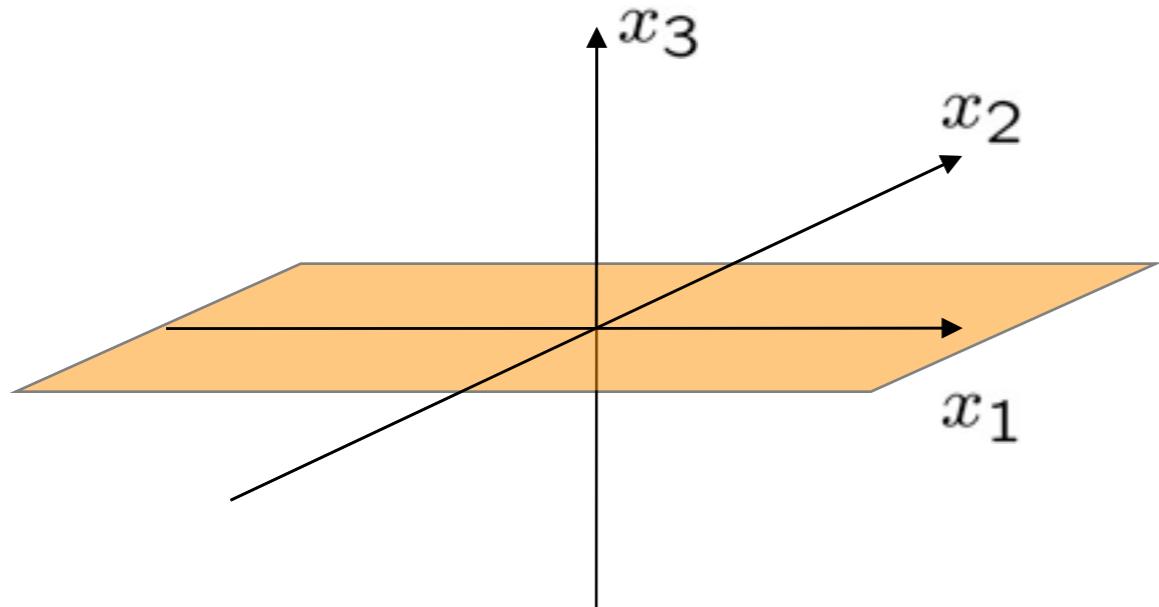
- Naive: Brute force search



- **Theorem** (*Davies et al, Natarajan*)
Solving the L0 optimization problem is NP-complete
- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

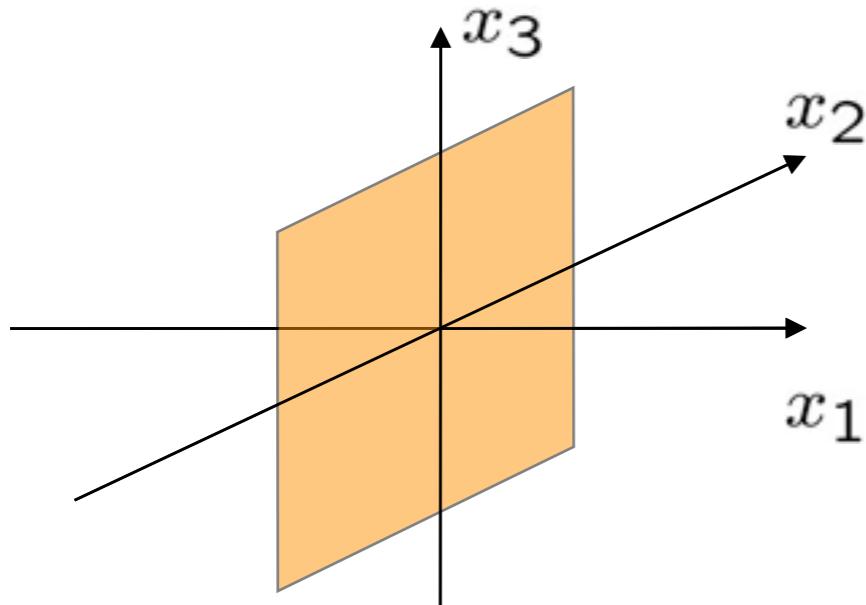
- Naive: Brute force search



- **Theorem** (*Davies et al, Natarajan*)
Solving the L0 optimization problem is NP-complete
- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

- Naive: Brute force search



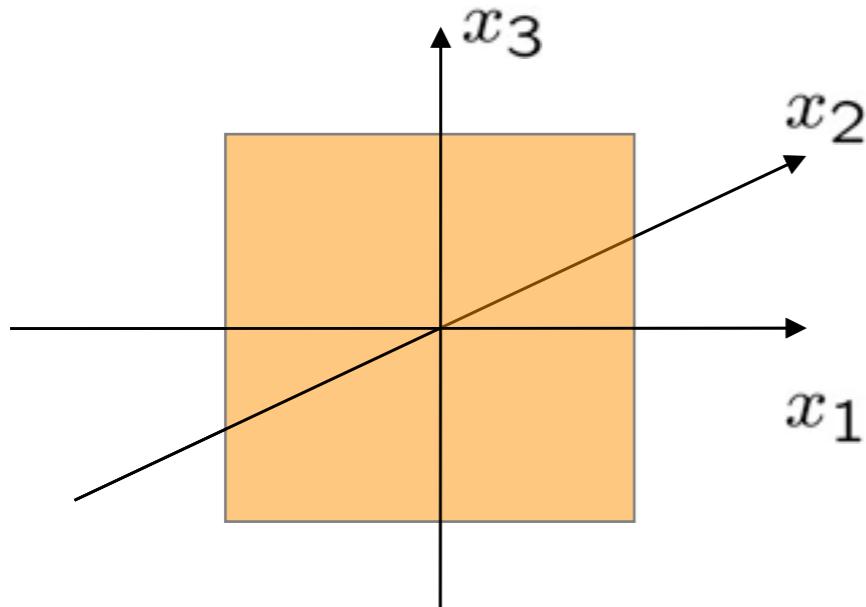
- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is NP-complete

- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

- Naive: Brute force search



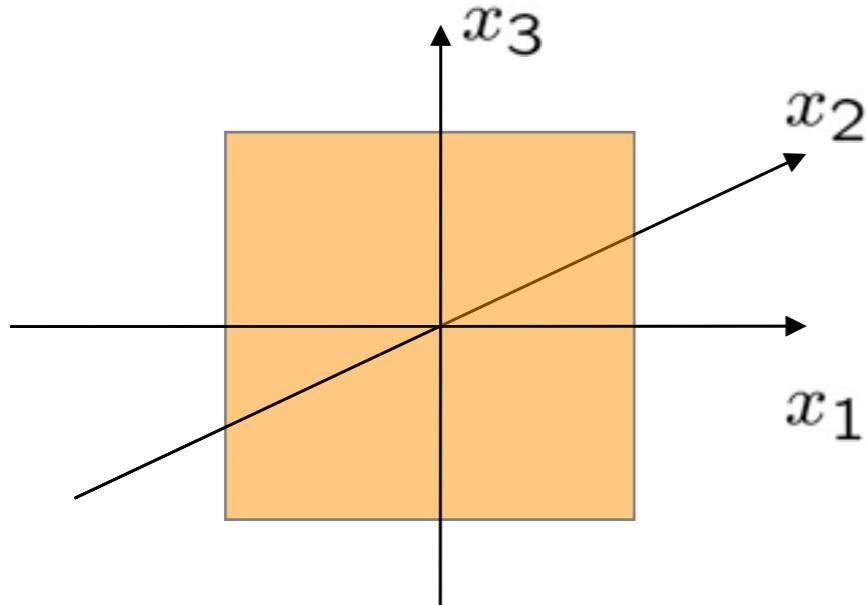
- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is NP-complete

- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

- Naive: Brute force search



$$\min_x \|\mathbf{b} - \mathbf{Ax}\|_2 \text{ s.t. } \text{support}(x) = I$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{b}$$

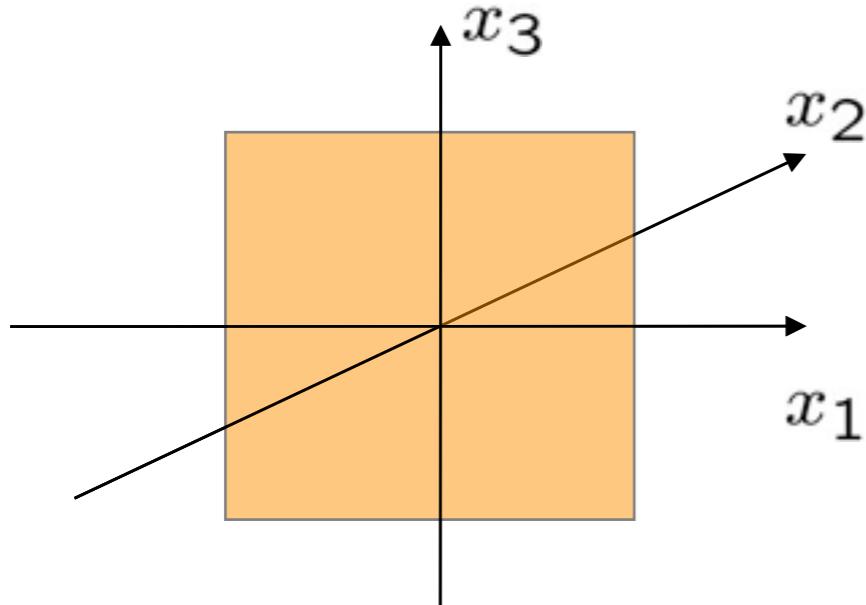
- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is NP-complete

- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

- Naive: Brute force search



$$\min_x \|\mathbf{b} - \mathbf{Ax}\|_2 \text{ s.t. } \text{support}(x) = I$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ 0 \\ x_3 \end{pmatrix} = \mathbf{b}$$

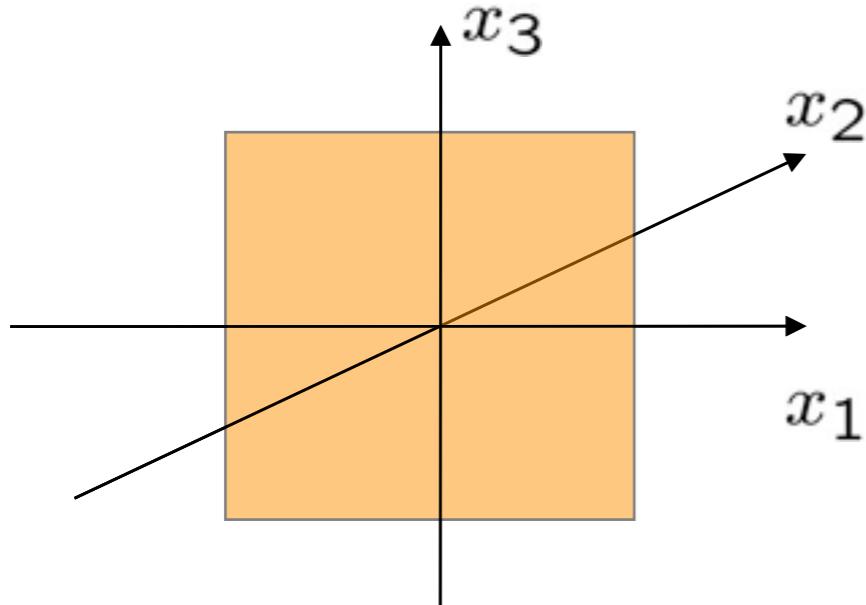
- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is NP-complete

- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

- Naive: Brute force search



$$\min_x \|\mathbf{b} - \mathbf{Ax}\|_2 \text{ s.t. } \text{support}(x) = I$$

$$\begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \mathbf{b}$$

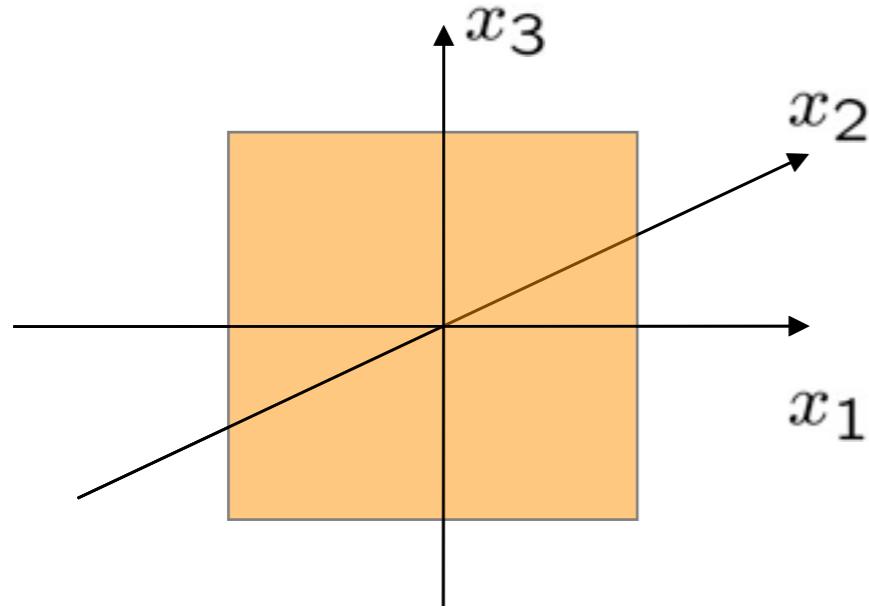
- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is NP-complete

- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

- Naive: Brute force search



$$\min_x \|\mathbf{b} - \mathbf{Ax}\|_2 \text{ s.t. } \text{support}(x) = I$$

$$\begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \mathbf{b}$$

$$\begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{bmatrix}^{-1} \cdot \mathbf{b}$$

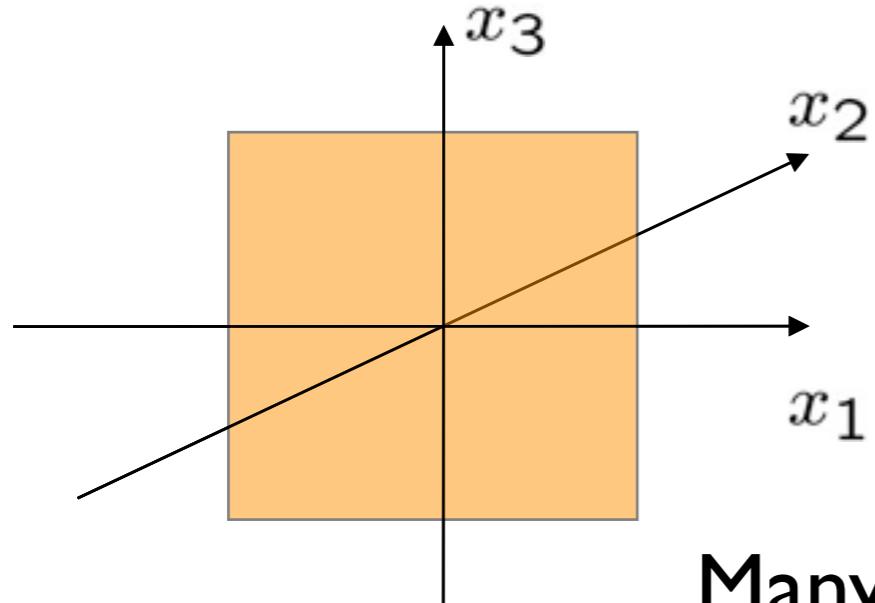
- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is NP-complete

- Are there other more efficient alternatives ?

Complexity of Ideal Sparse Approximation

- Naive: Brute force search



$$\min_x \|\mathbf{b} - \mathbf{Ax}\|_2 \text{ s.t. } \text{support}(x) = I$$

$$\begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \mathbf{b}$$

$$\begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{bmatrix}^{-1} \cdot \mathbf{b}$$

Many k -tuples to try!

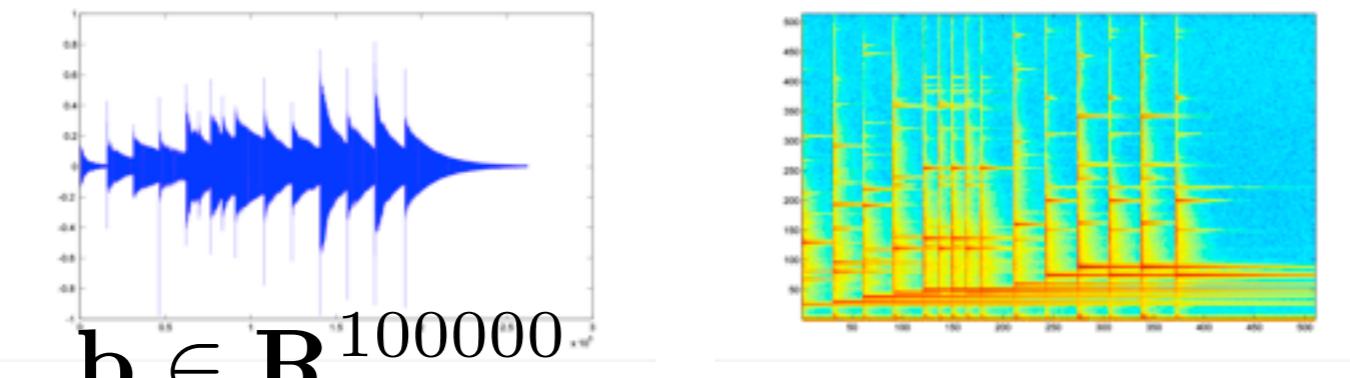
- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is NP-complete

- Are there other more efficient alternatives ?

Efficient Sparse Approximation With Time-Frequency Atoms

- Audio = superimposition of structures
- Example : glockenspiel



$$\mathbf{b} \in \mathbb{R}^{100000}$$

- ◆ transients = short, small scale
- ◆ harmonic part = long, large scale

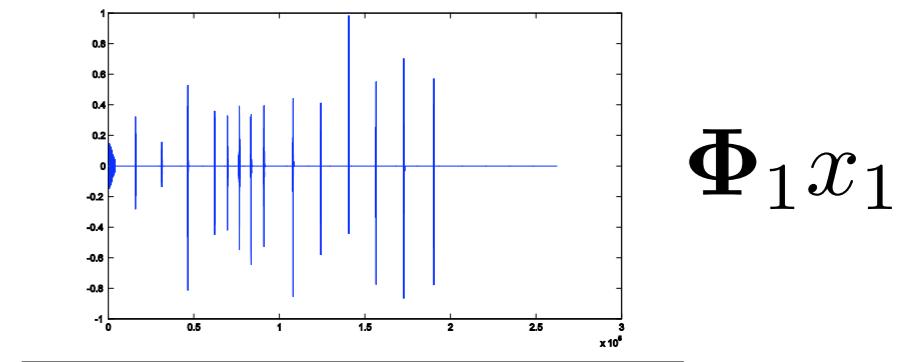
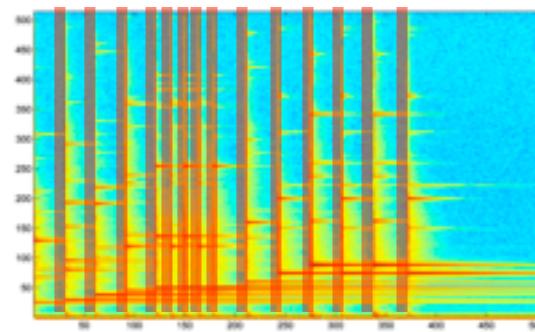
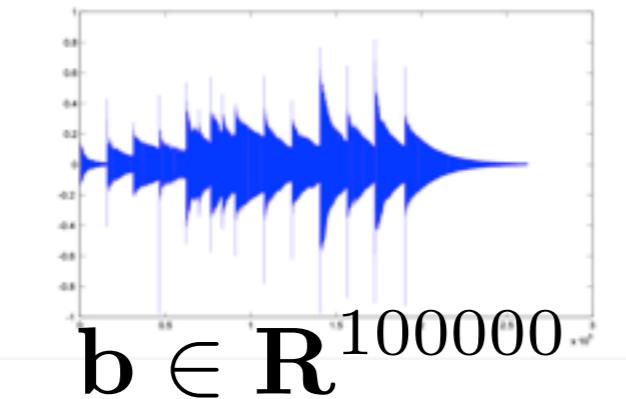
- Two-layer sparse model with Gabor atoms

$$\left\{ \varphi_{s,\tau,f}(t) = \frac{1}{\sqrt{s}} w\left(\frac{t-\tau}{s}\right) e^{2i\pi f t} \right\}_{s,\tau,f} \quad \mathbf{b} \approx \Phi_1 x_1 + \Phi_2 x_2$$

Efficient Sparse Approximation With Time-Frequency Atoms

- Audio = superimposition of structures

- Example : glockenspiel



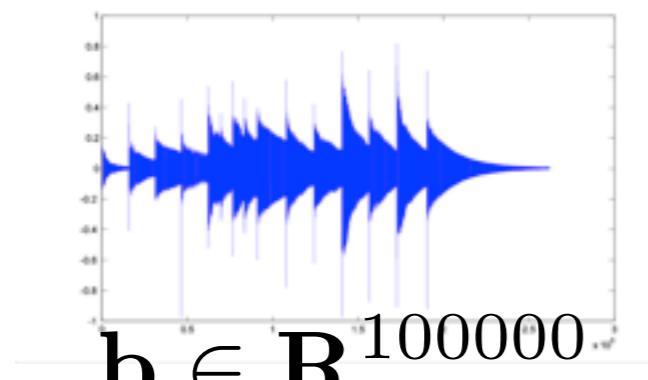
- ◆ transients = short, small scale
- ◆ harmonic part = long, large scale

- Two-layer sparse model with Gabor atoms

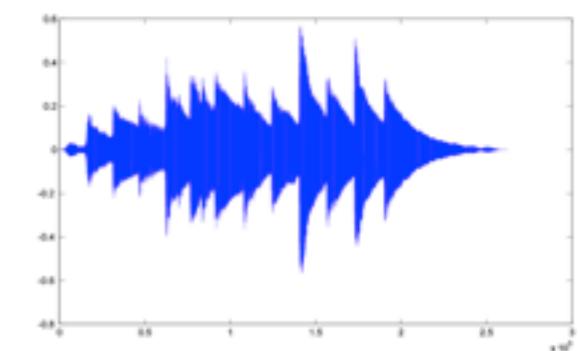
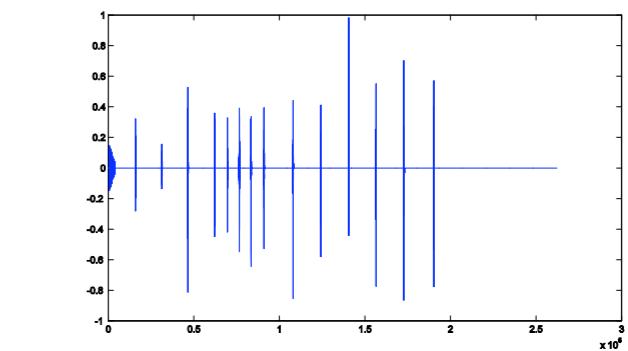
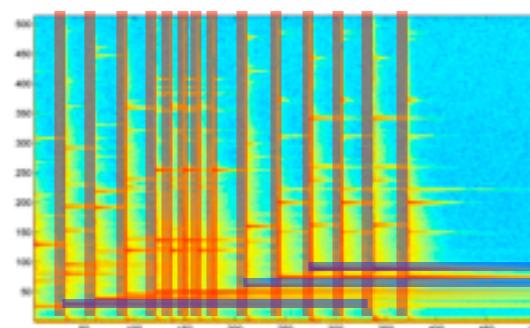
$$\left\{ \varphi_{s,\tau,f}(t) = \frac{1}{\sqrt{s}} w\left(\frac{t-\tau}{s}\right) e^{2i\pi f t} \right\}_{s,\tau,f} \quad b \approx \Phi_1 x_1 + \Phi_2 x_2$$

Efficient Sparse Approximation With Time-Frequency Atoms

- Audio = superimposition of structures
- Example : glockenspiel



- ◆ transients = short, small scale
- ◆ harmonic part = long, large scale



- Two-layer sparse model with Gabor atoms

$$\left\{ \varphi_{s,\tau,f}(t) = \frac{1}{\sqrt{s}} w\left(\frac{t-\tau}{s}\right) e^{2i\pi f t} \right\}_{s,\tau,f}$$

$$b \approx \Phi_1 x_1 + \Phi_2 x_2$$

Towards Pursuit Algorithms Optimization *principles*

Overall compromise

- Approximation quality

$$\|\mathbf{A}x - \mathbf{b}\|_2$$

- Ideal sparsity measure : ℓ^0 “norm”

$$\|x\|_0 := \#\{n, x_n \neq 0\} = \sum_n |x_n|^0$$

- “Relaxed” sparsity measures

$$0 < p < \infty, \|x\|_p := \left(\sum_n |x_n|^p \right)^{1/p}$$

L_p norms / quasi-norms

- **Norms when $1 \leq p < \infty$ = convex**

$$\|x\|_p = 0 \Leftrightarrow x = 0$$

$$\|\lambda x\|_p = |\lambda| \|x\|_p, \forall \lambda, x$$

Triangle inequality $\|x + y\|_p \leq \|x\|_p + \|y\|_p, \forall x, y$

- **Quasi-norms when $0 < p < 1$ = nonconvex**

$$\|x + y\|_p \leq 2^{1/p} (\|x\|_p + \|y\|_p), \forall x, y$$

Quasi-triangle inequality

$$\|x + y\|_p^p \leq \|x\|_p^p + \|y\|_p^p, \forall x, y$$

- “*Pseudo*”-norm for $p=0$

$$\|x + y\|_0 \leq \|x\|_0 + \|y\|_0, \forall x, y$$

Optimization problems

- Approximation

$$\min_x \|\mathbf{b} - \mathbf{A}x\|_2 \text{ s.t. } \|x\|_p \leq \tau$$

- Sparsification

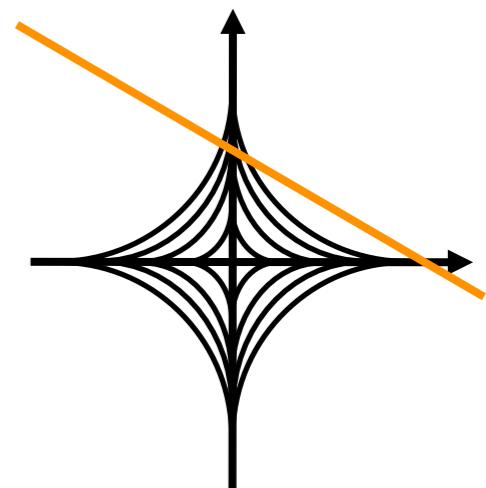
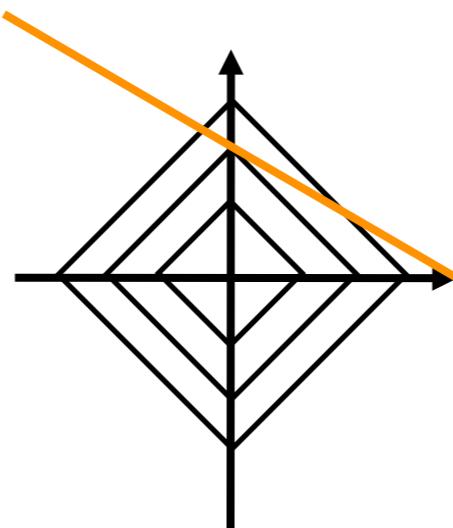
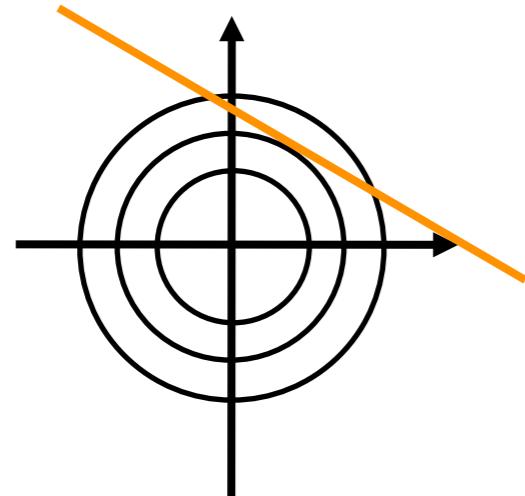
$$\min_x \|x\|_p \text{ s.t. } \|\mathbf{b} - \mathbf{A}x\|_2 \leq \epsilon$$

- Regularization

$$\min_x \frac{1}{2} \|\mathbf{b} - \mathbf{A}x\|_2 + \lambda \|x\|_p$$

L_p “norms” level sets

- Strictly convex when $p>1$
- Convex $p=1$
- Nonconvex $p<1$



Observation: the minimizer is sparse when $p \leq 1$

— $\{x \text{ s.t. } b = Ax\}$

Global Optimization : from Principles to Algorithms

- Optimization principle

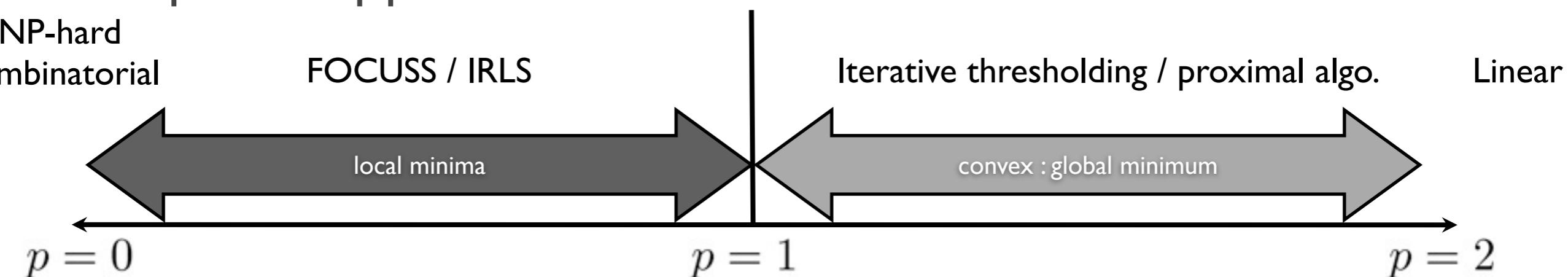
- ✓ Sparse representation
- ✓ Sparse approximation

NP-hard
combinatorial

FOCUSS / IRLS

$$\min_x \frac{1}{2} \|\mathbf{A}x - \mathbf{b}\|_2^2 + \lambda \|x\|_p^p$$

$$\begin{aligned} \lambda \rightarrow 0 & \quad \mathbf{A}x = \mathbf{b} \\ \lambda > 0 & \quad \mathbf{A}x \approx \mathbf{b} \end{aligned}$$



Lasso [Tibshirani 1996], Basis Pursuit (Denoising) [Chen, Donoho & Saunders, 1999]

Linear/Quadratic programming (interior point, etc.)

Homotopy method [Osborne 2000] / Least Angle Regression [Efron & al 2002]

Iterative / proximal algorithms [Daubechies, de Frise, de Mol 2004, Combettes & Pesquet 2008, Beck & Teboulle 2009 ...]

Greedy Pursuit Algorithms

The orthonormal case: thresholding

- Observation: when \mathbf{A} is orthonormal,
 - ✓ the problem

$$\min_x \|\mathbf{b} - \mathbf{A}x\|_2^2 \text{ s.t. } \|x\|_0 \leq k$$

- ✓ is equivalent to

$$\min_x \sum_n (\mathbf{a}_n^T \mathbf{b} - x_n)^2 \text{ s.t. } \|x\|_0 \leq k$$

- Let Λ_k index the k largest inner products

$$\min_{n \in \Lambda_k} |\mathbf{a}_n^T \mathbf{b}| \geq \max_{n \notin \Lambda_k} |\mathbf{a}_n^T \mathbf{b}|$$

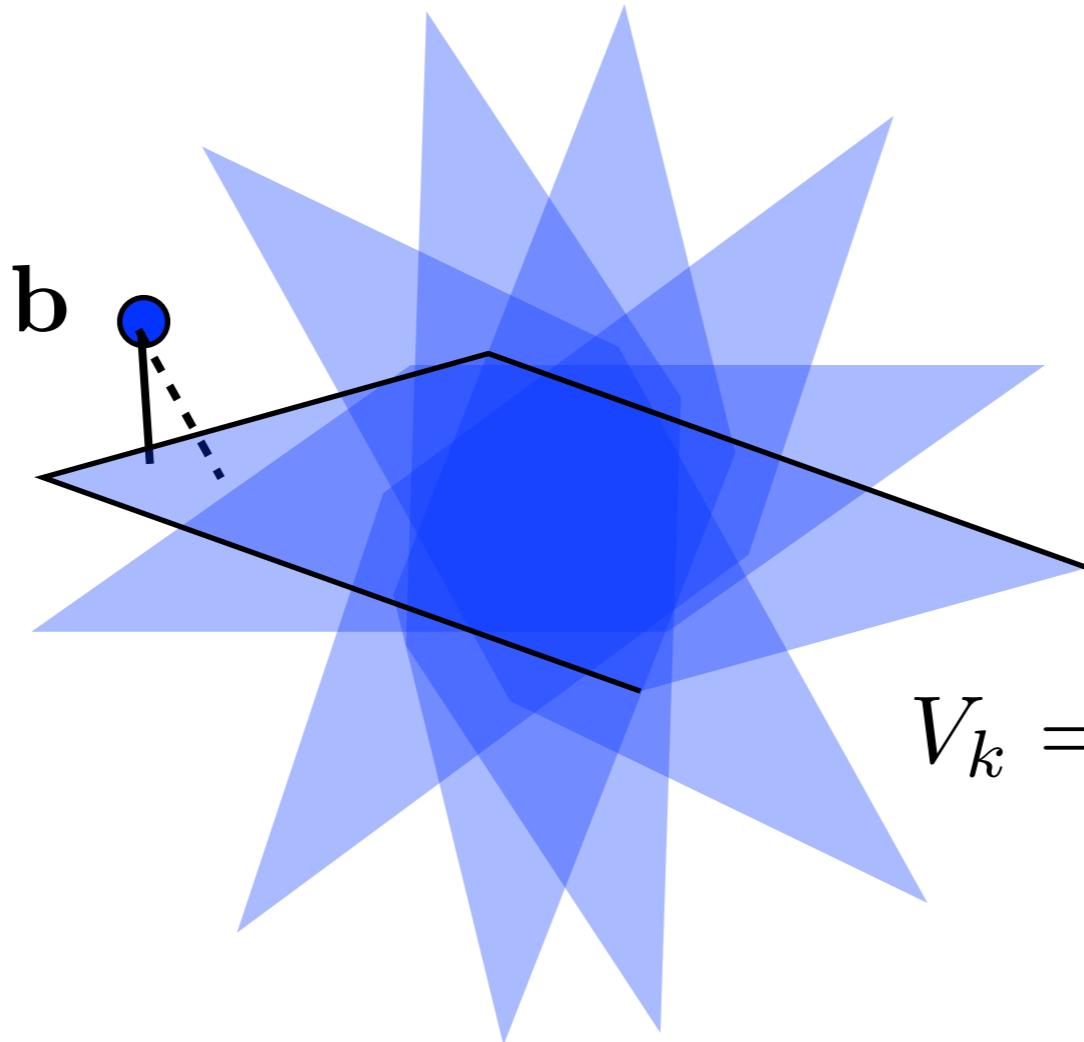
- ✓ an optimum solution is

$$x_n = \mathbf{a}_n^T \mathbf{b}, n \in \Lambda_k; \quad x_n = 0, n \notin \Lambda_k$$

Matching Pursuit (MP)

- Iterative algorithm (aka Projection Pursuit, CLEAN)
 - ✓ Initialization $\mathbf{r}_0 = \mathbf{b}$ $i = 1$
 - ✓ Atom selection:
$$n_i = \arg \max_n |\mathbf{a}_n^T \mathbf{r}_{i-1}|$$
 - ✓ Residual update
$$\mathbf{r}_i = \mathbf{r}_{i-1} - (\mathbf{a}_{n_i}^T \mathbf{r}_{i-1}) \mathbf{a}_{n_i}$$
- Sparse approximation after k steps

$$\mathbf{b} = \sum_{i=1}^k \mathbf{a}_{n_i}^T \mathbf{r}_{i-1} \mathbf{a}_{n_i} + \mathbf{r}_k$$



$$V_k = \text{span}(\mathbf{a}_{n_i}, 1 \leq i \leq k)$$

Orthonormal MP (OMP)

[Mallat & Zhang 93, Pati & al 94]

- Observation: after k iterations
- Approximant belongs to

$$V_k = \text{span}(\mathbf{a}_{n_i}, 1 \leq i \leq k)$$

- Best approximation from V_k = orthoprojection

$$P_{V_k} \mathbf{b} = \mathbf{A}_{\Lambda_k} \mathbf{A}_{\Lambda_k}^+ \mathbf{b}$$

$$\Lambda_k = \{n_i, 1 \leq i \leq k\}$$

- OMP residual update rule

$$\mathbf{r}_k = \mathbf{b} - P_{V_k} \mathbf{b}$$

OMP

- Same as MP, except *residual update rule*
 - ✓ Atom selection:

$$n_i = \arg \max_n |\mathbf{a}_n^T \mathbf{r}_{i-1}|$$

- ✓ Index update $\Lambda_i = \Lambda_{i-1} \cup \{n_i\}$
- ✓ *Residual update*

$$V_i = \text{span}(\mathbf{a}_n, n \in \Lambda_i)$$

$$\mathbf{r}_i = \mathbf{b} - P_{V_i} \mathbf{b}$$

Stagewise greedy algorithms

- Principle
 - ✓ select *multiple* atoms at a time to accelerate the process
 - ✓ possibly *prune out* some atoms at each stage
- Example of such algorithms
 - ◆ Morphological Component Analysis [MCA, Bobin et al]
 - ◆ Stagewise OMP [Donoho & al]
 - ◆ CoSAMP [Needell & Tropp]
 - ◆ ROMP [Needell & Vershynin]
 - ◆ Iterative Hard Thresholding [Blumensath & Davies 2008]

Overview of greedy algorithms

$$\mathbf{b} = \mathbf{A}x_i + \mathbf{r}_i$$

$$\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_N]$$

	Matching Pursuit	OMP	Stagewise OMP
Selection	$\Gamma_i := \arg \max_n \mathbf{A}_n^T \mathbf{r}_{i-1} $	$\Gamma_i := \{n \mid \mathbf{A}_n^T \mathbf{r}_{i-1} > \theta_i\}$	
Update	$\Lambda_i = \Lambda_{i-1} \cup \Gamma_i$ $x_i = x_{i-1} + \mathbf{A}_{\Gamma_i}^+ \mathbf{r}_{i-1}$ $\mathbf{r}_i = \mathbf{r}_{i-1} - \mathbf{A}_{\Gamma_i} \mathbf{A}_{\Gamma_i}^+ \mathbf{r}_{i-1}$	$\Lambda_i = \Lambda_{i-1} \cup \Gamma_i$ $x_i = \mathbf{A}_{\Lambda_i}^+ \mathbf{b}$ $\mathbf{r}_i = \mathbf{b} - \mathbf{A}_{\Lambda_i} x_i$	

MP & OMP: *Mallat & Zhang 1993*
 StOMP: *Donoho & al 2006* (similar to MCA, *Bobin & al 2006*)

Summary

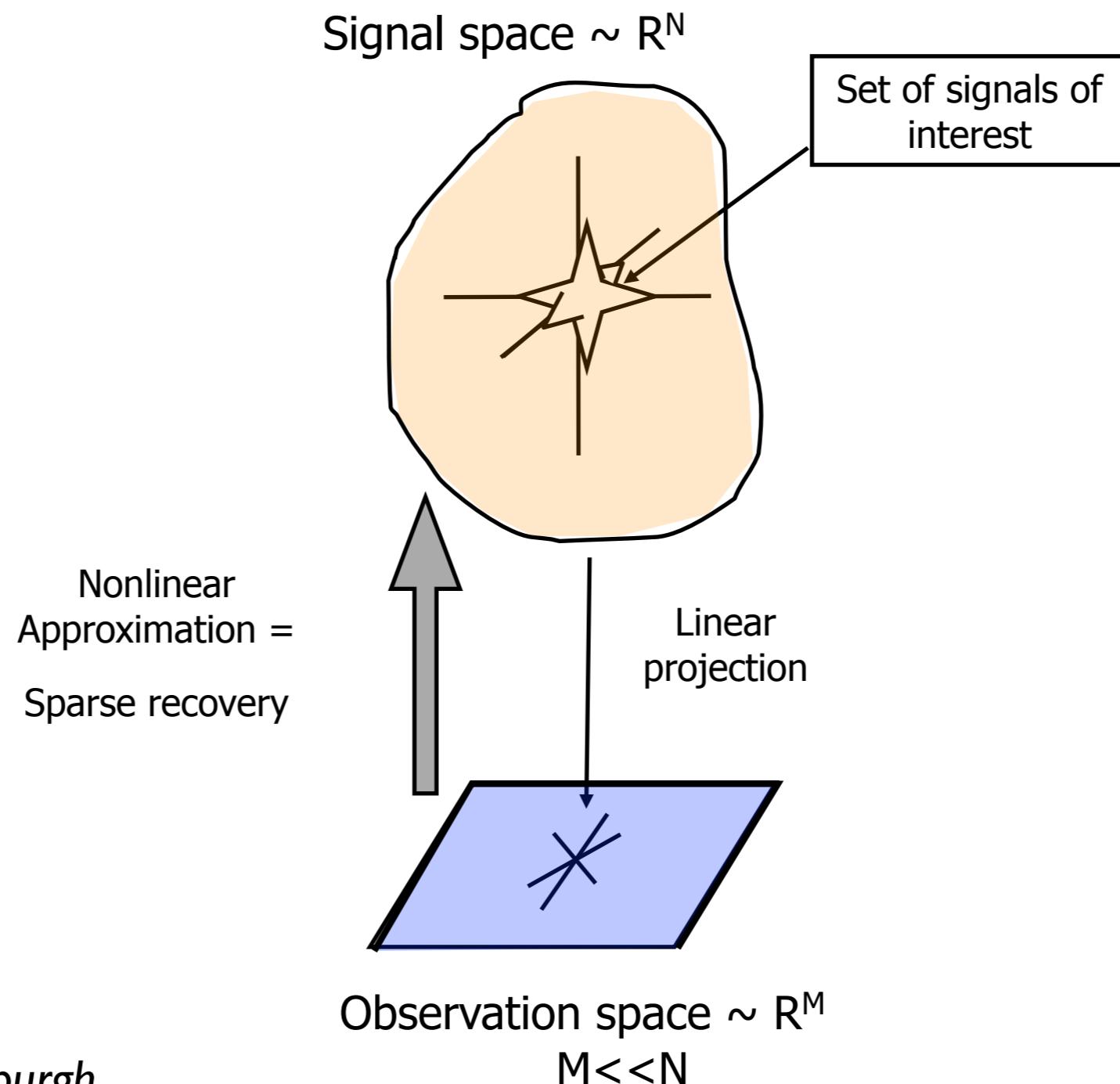
Global optimization

Iterative greedy algorithms

Principle	$\min_x \frac{1}{2} \ \mathbf{A}x - \mathbf{b}\ _2^2 + \lambda \ x\ _p^p$	iterative decomposition $\mathbf{r}_i = \mathbf{b} - \mathbf{A}x_i$ <ul style="list-style-type: none"> • select new components • update residual
Tuning quality/sparsity	regularization parameter λ	stopping criterion (nb of iterations, error level, ...) $\ x_i\ _0 \geq k \quad \ \mathbf{r}_i\ \leq \epsilon$
Variants	<ul style="list-style-type: none"> • choice of sparsity measure p • optimization algorithm • initialization 	<ul style="list-style-type: none"> • selection criterion (weak, stagewise ...) • update strategy (orthogonal ...)

Sparse recovery: Provably good algorithms?

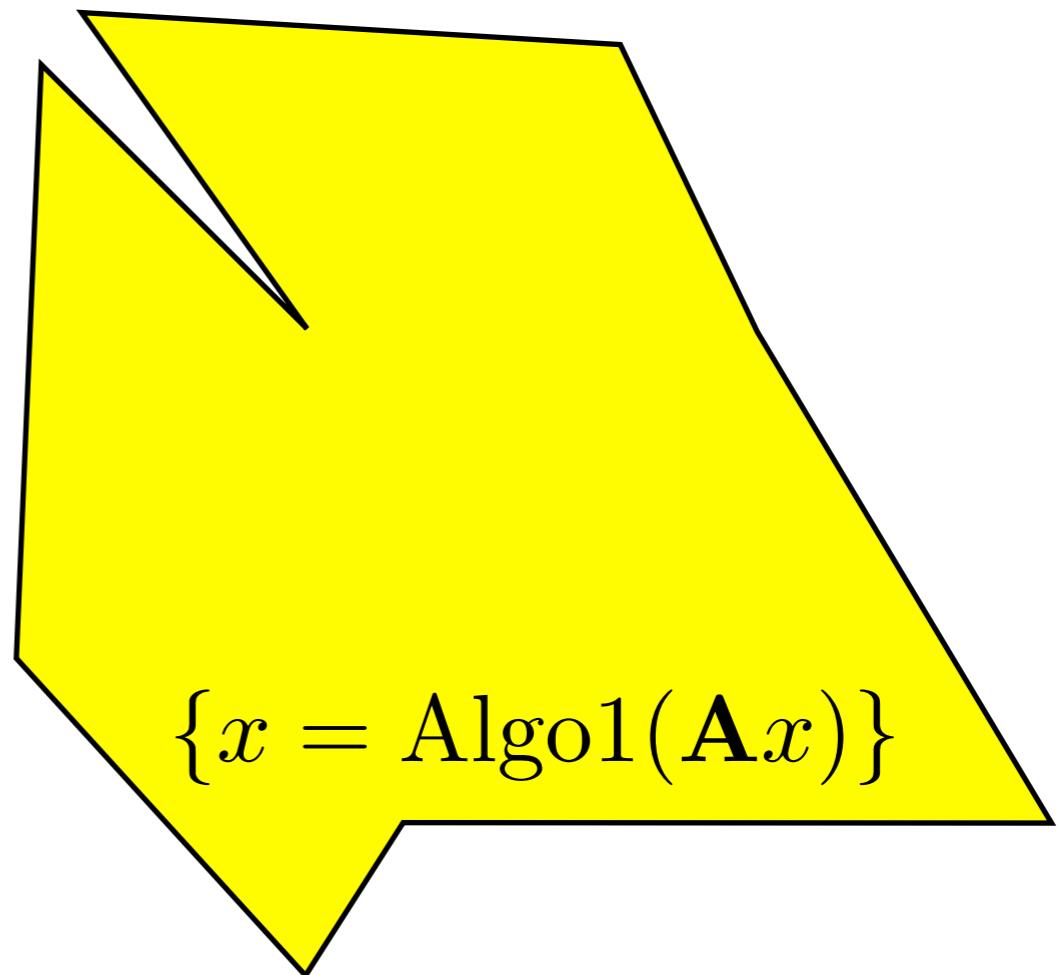
Inverse problems



Courtesy: M. Davies, U. Edinburgh

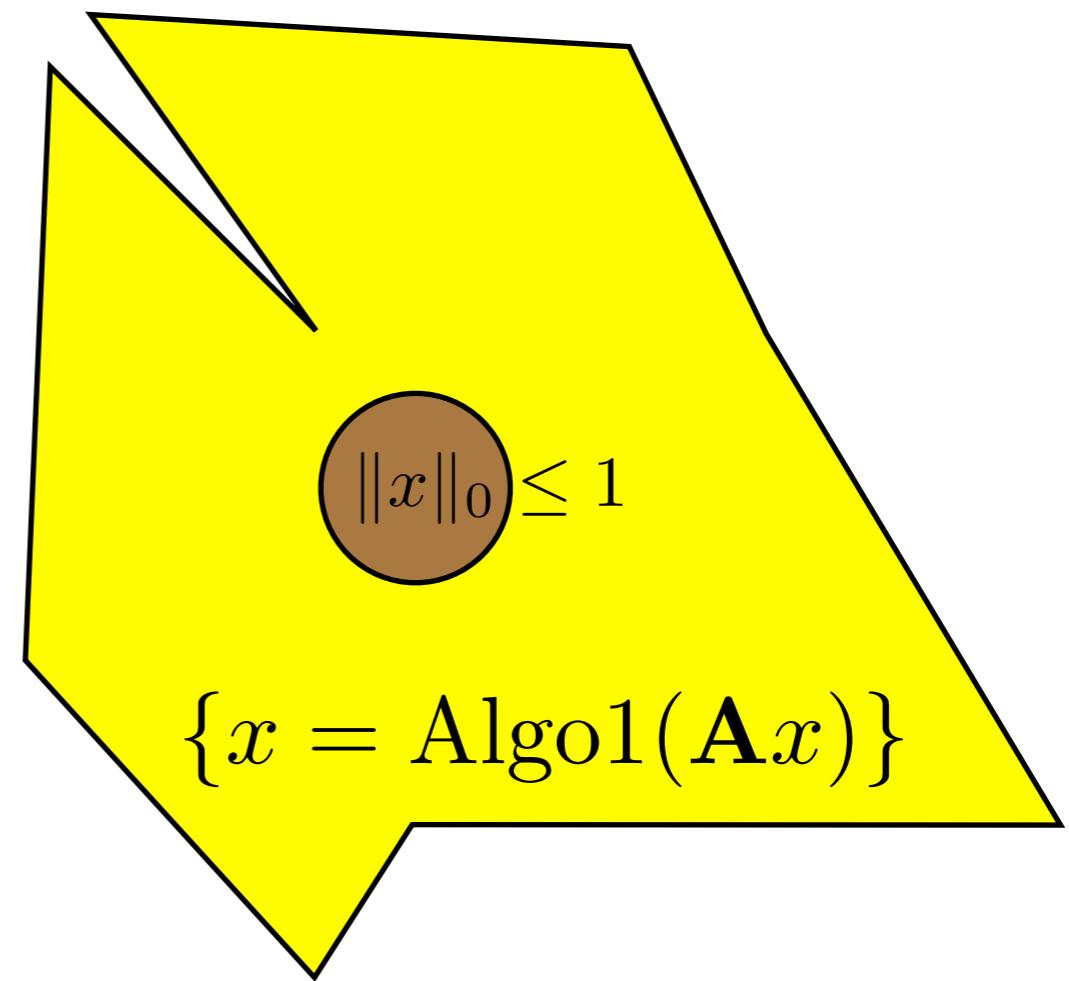
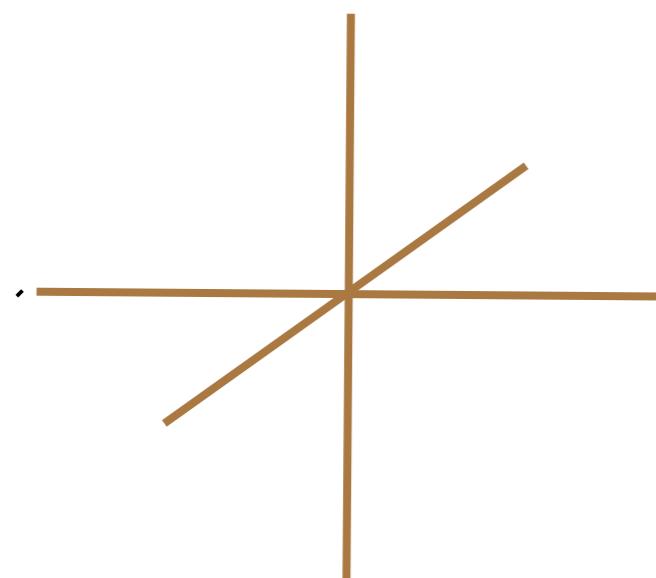
Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

- Recoverable set for a given “inversion” algorithm



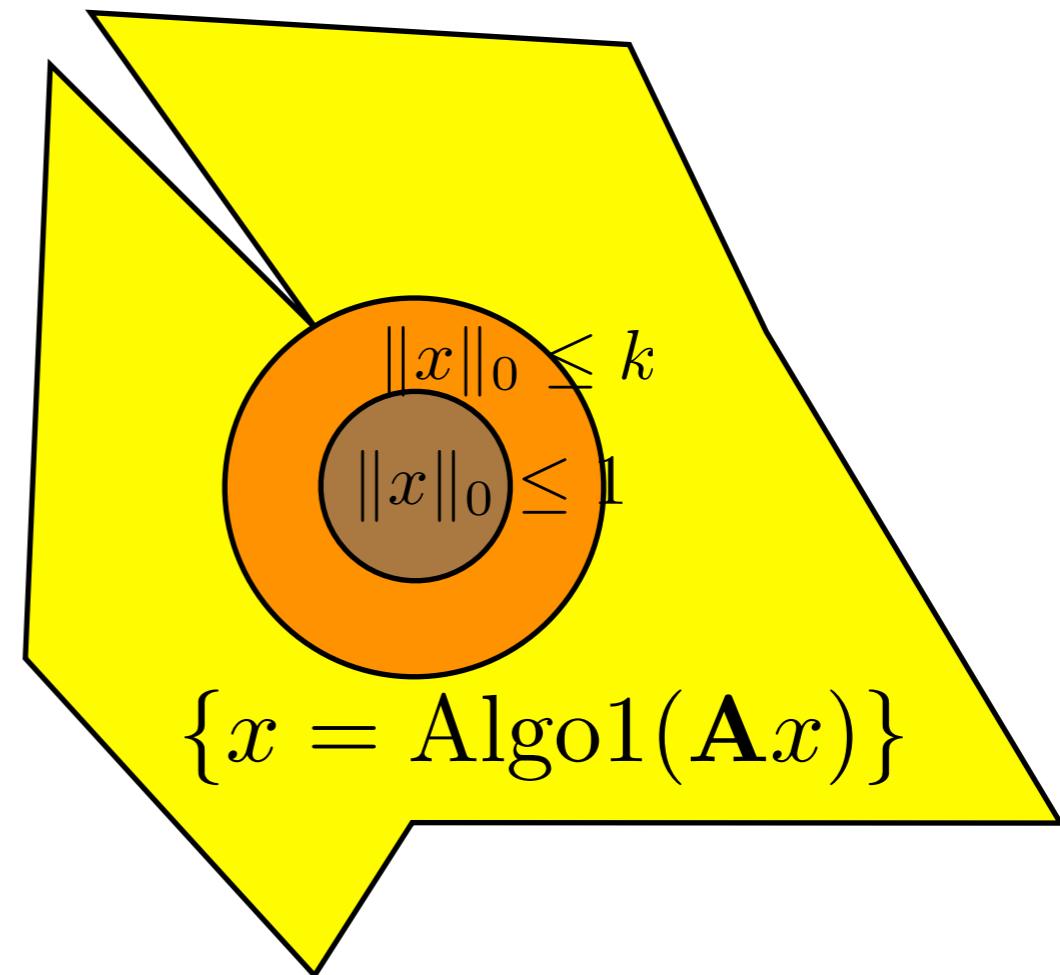
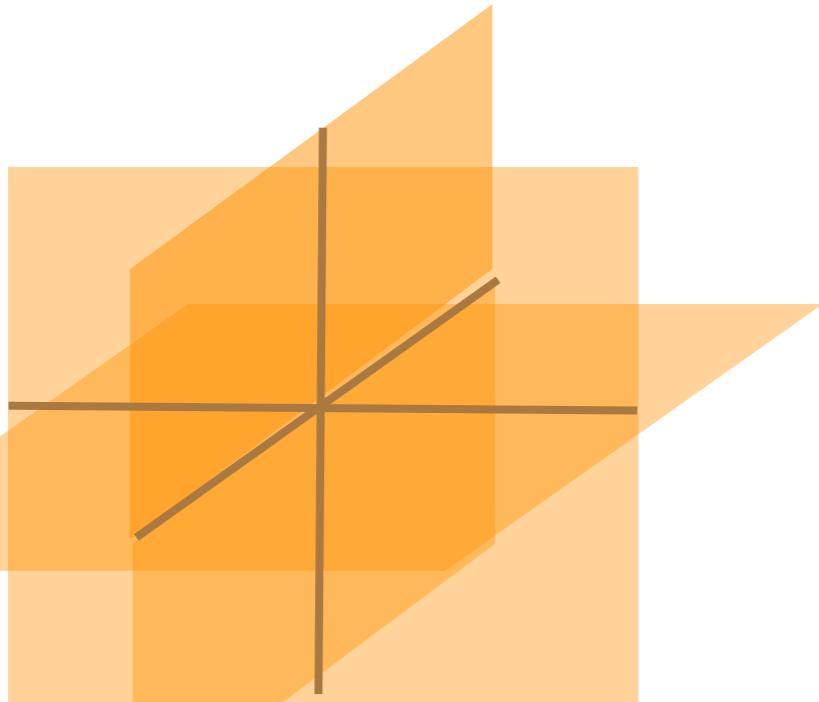
Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
 - ✓ 1-sparse



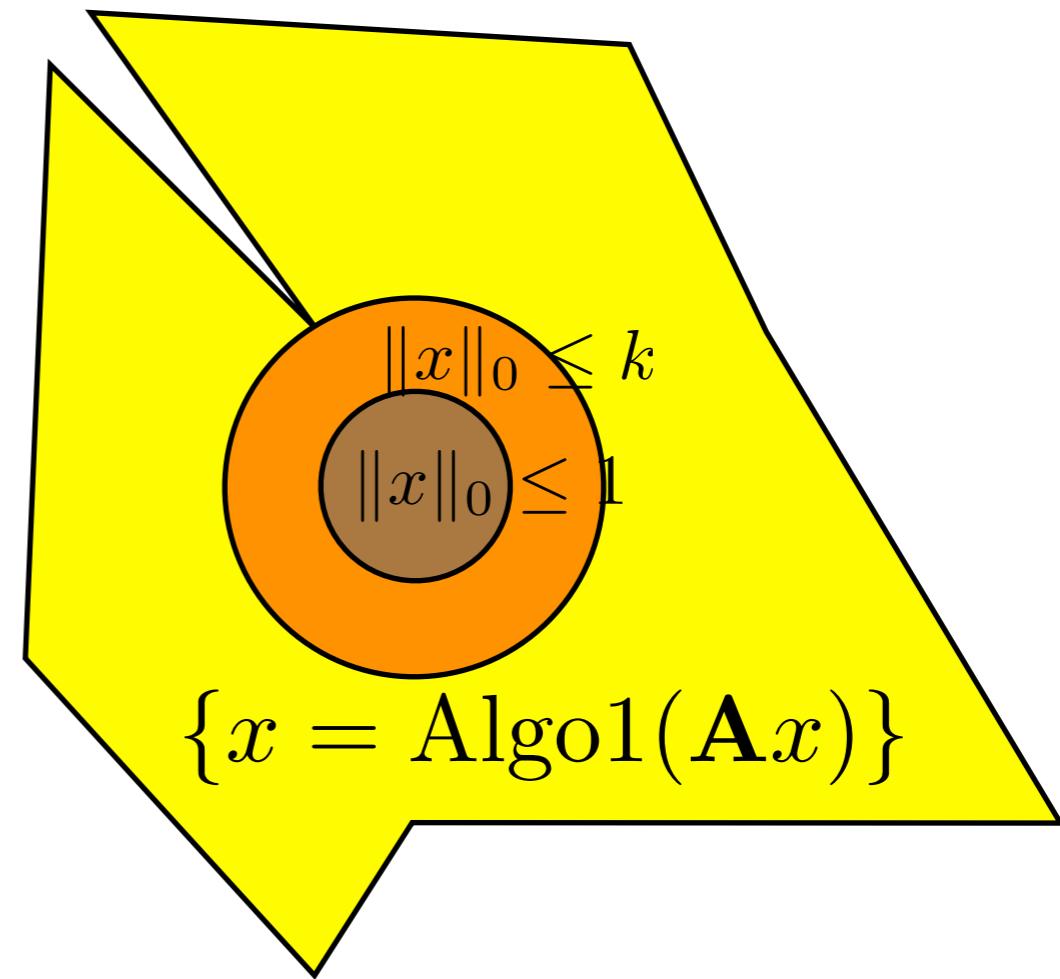
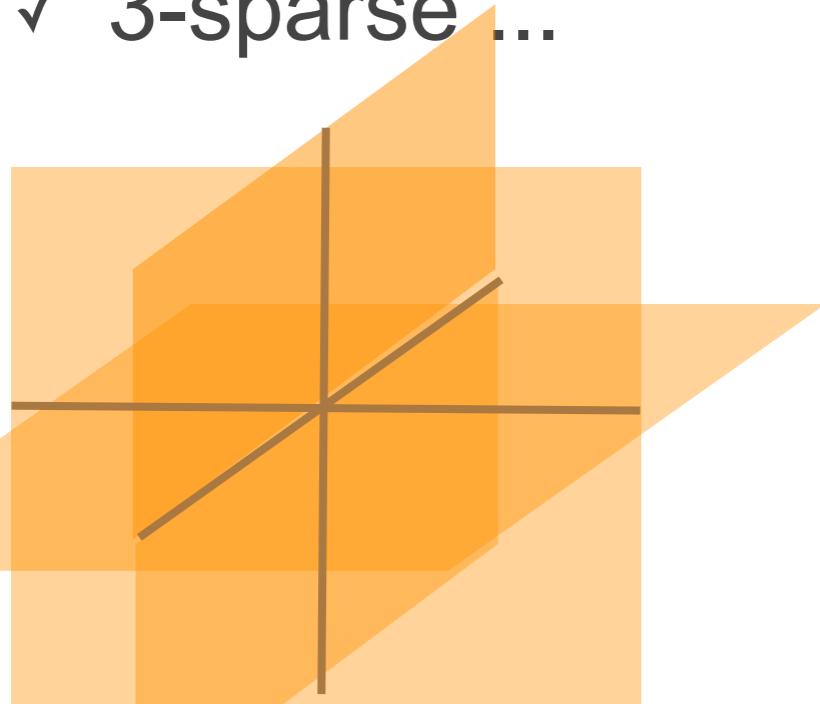
Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
 - ✓ 1-sparse
 - ✓ 2-sparse



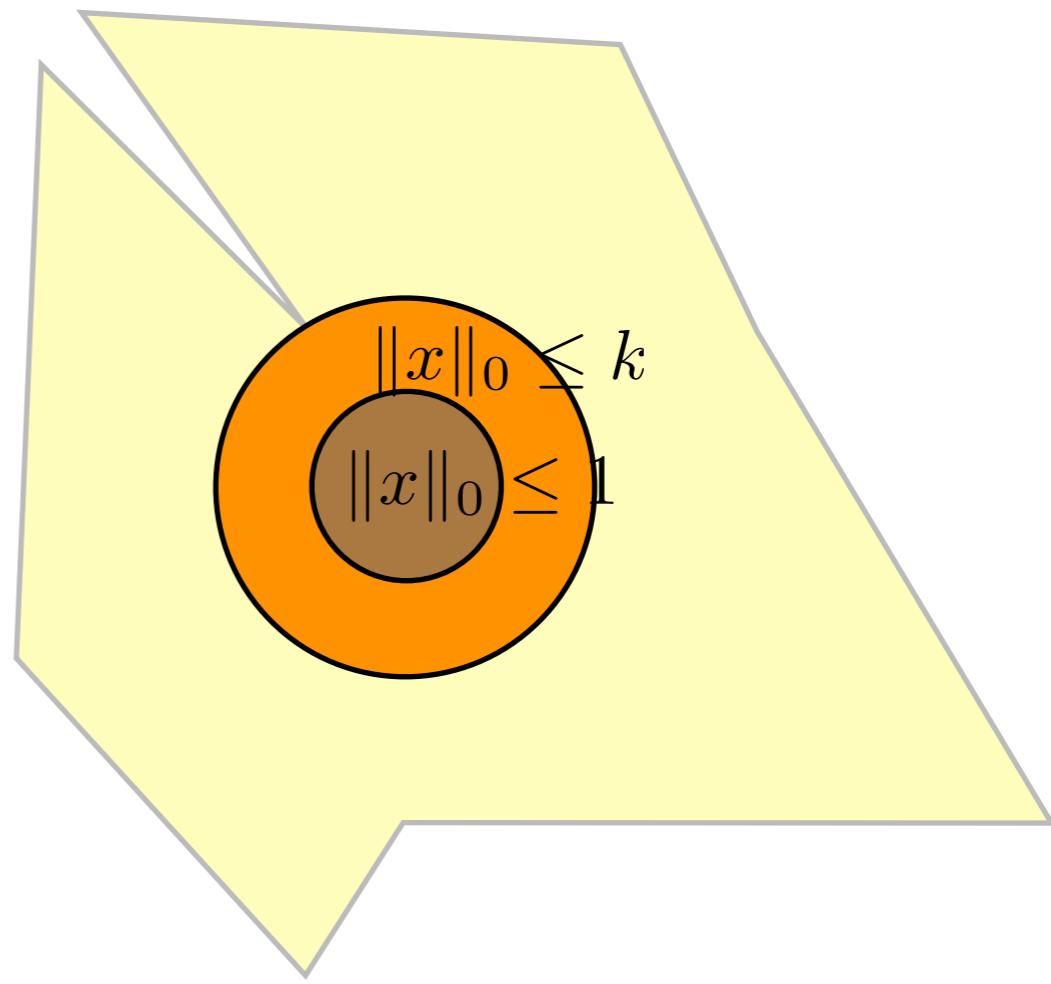
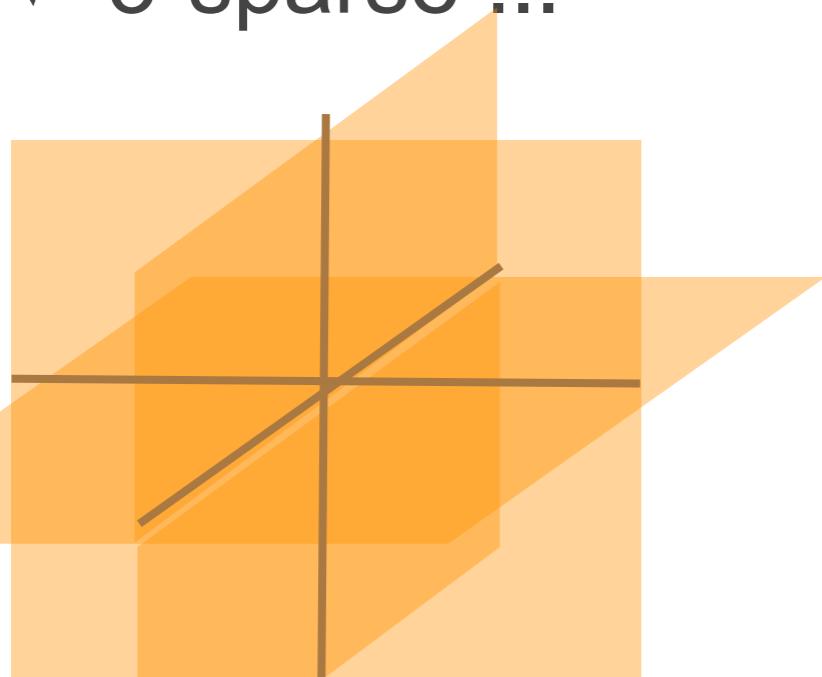
Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
 - ✓ 1-sparse
 - ✓ 2-sparse
 - ✓ 3-sparse ...



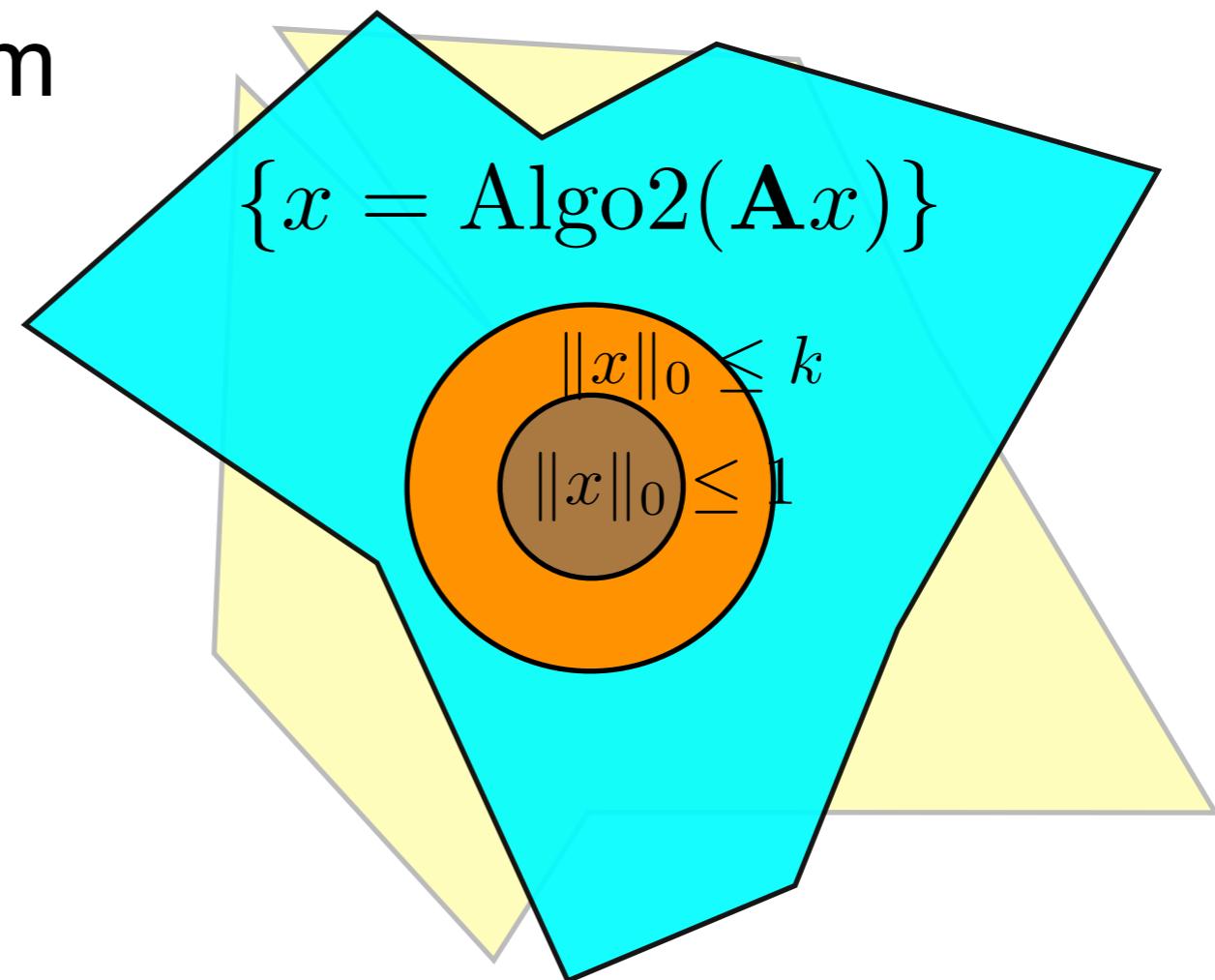
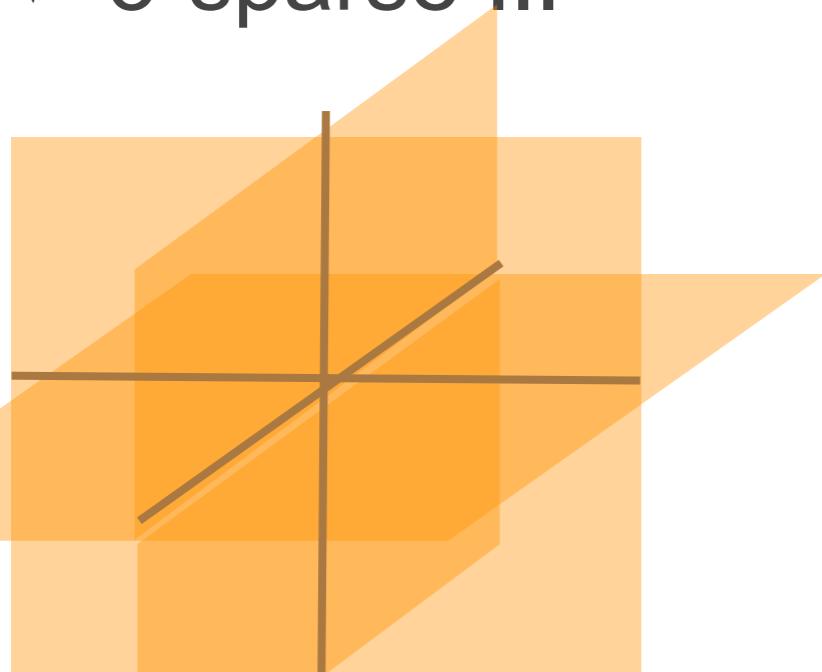
Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
 - ✓ 1-sparse
 - ✓ 2-sparse
 - ✓ 3-sparse ...



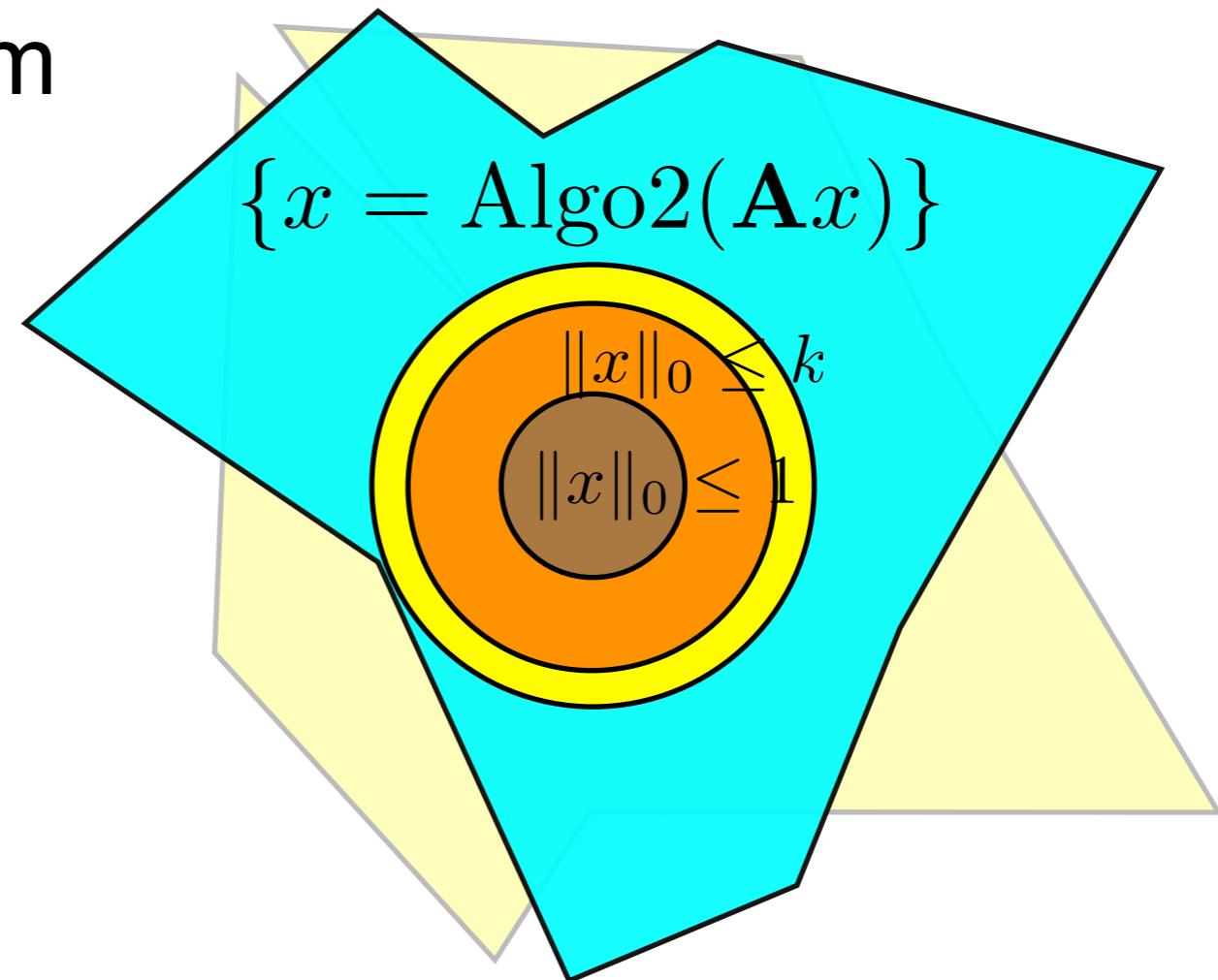
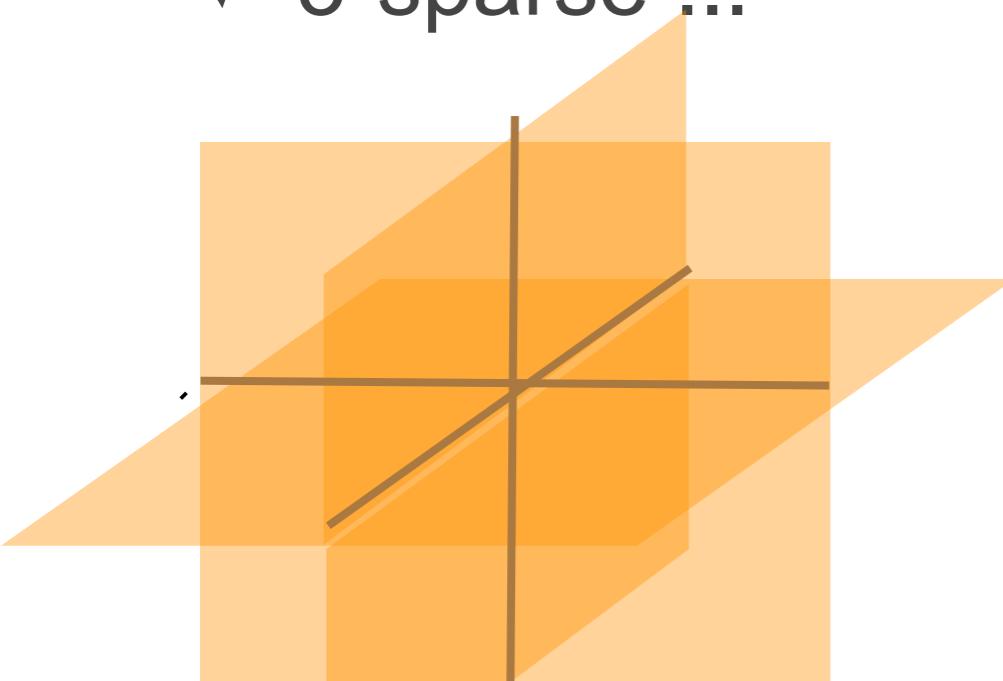
Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
 - ✓ 1-sparse
 - ✓ 2-sparse
 - ✓ 3-sparse ...



Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
 - ✓ 1-sparse
 - ✓ 2-sparse
 - ✓ 3-sparse ...



Equivalence between L0, L1, OMP

- **Theorem** : assume that $\mathbf{b} = \mathbf{A}x_0$

- ✓ if $\|x_0\|_0 \leq k_0(\mathbf{A})$ then $x_0 = x_0^*$
- ✓ if $\|x_0\|_0 \leq k_1(\mathbf{A})$ then $x_0 = x_1^*$

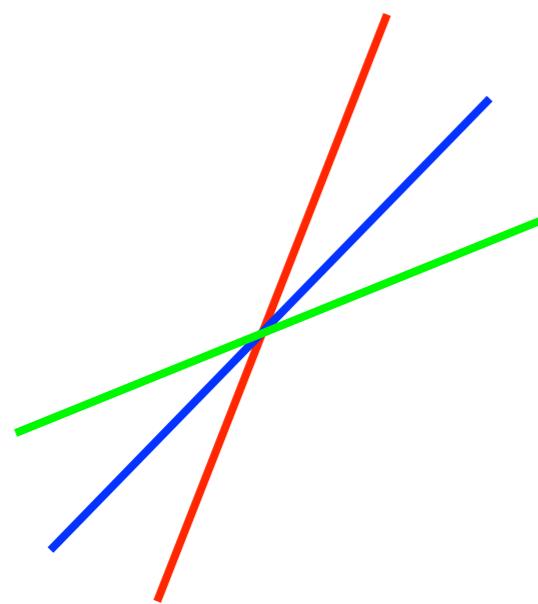
where $x_p^* = \arg \min_{\mathbf{A}x=\mathbf{A}x_0} \|x\|_p$

- *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*

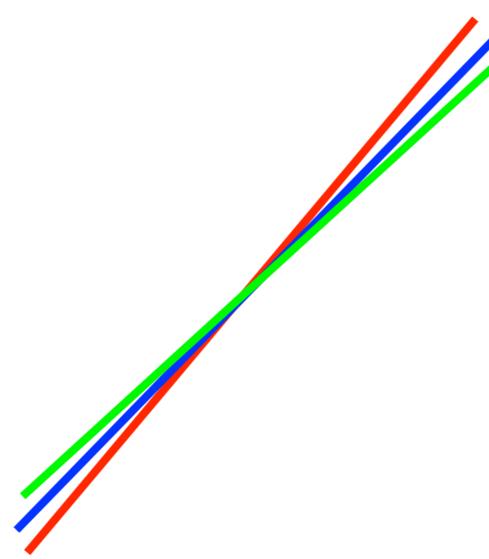
Robust / stable sparse recovery ?

Stable identifiability

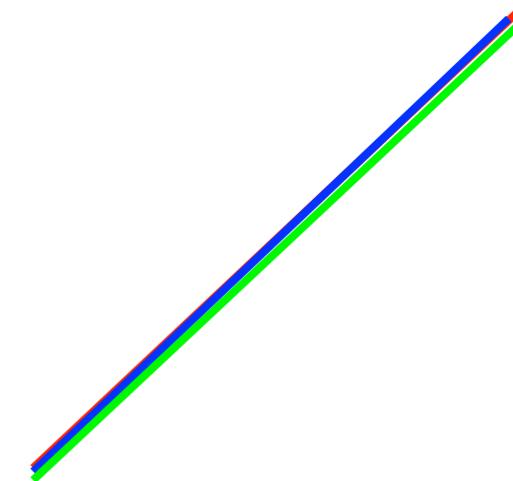
- Identifiability of 1-sparse vectors, with \mathbf{A} 2x3 matrix



Robust



Not robust



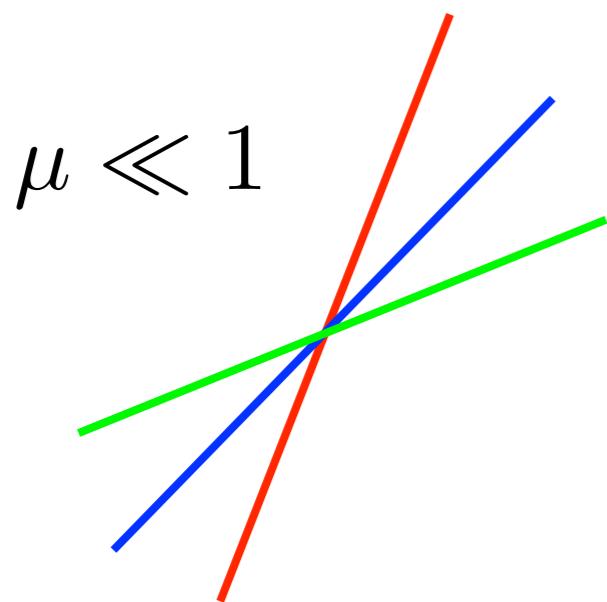
Not identifiable

- Beyond well-posedness: need *good conditioning*

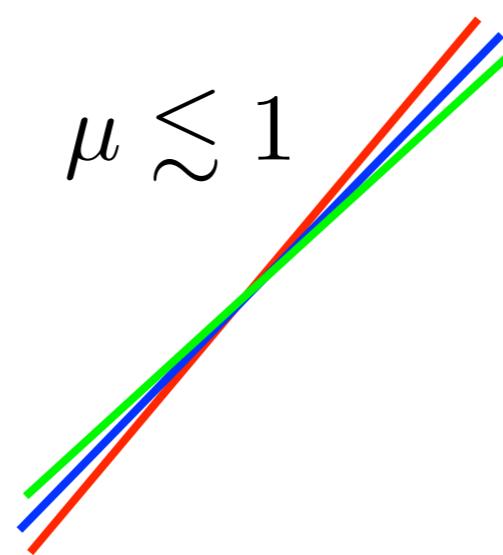
Quantitative well-posedness: Coherence

- Coherence of $\mathbf{A} = [\mathbf{a}_1 \dots \mathbf{a}_N]$

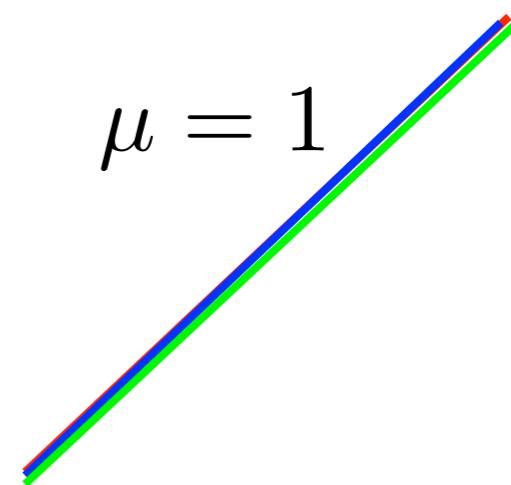
$$\mu(\mathbf{A}) := \max_{i \neq j} \left| \left\langle \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|_2}, \frac{\mathbf{a}_j}{\|\mathbf{a}_j\|_2} \right\rangle \right|$$



Incoherent



Coherent



Perfectly coherent

- Small coherence implies large Kruskal rank