Inverse problems and sparse representations

Rémi Gribonval, DR INRIA
EPI METISS (Speech and Audio Processing)
INRIA Rennes - Bretagne Atlantique

remi.gribonval@inria.fr
http://www.irisa.fr/metiss/members/remi/talks

Further material on sparsity

- Books with a 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.
- Review paper:
  - Bruckstein, Donoho, Elad, SIAM Reviews, 2009
- Video lectures
  - E. Candès, MLSS'09
  - F. Bach, NIPS 2009
  - Sparsity in Machine Learning and Statistics SMLS'09

Structure of the course

- Session 1: Panorama
  - ✓ sparsity: compression, inverse problems, learning
  - ✓ introduction to compressed (random) sensing
- Session 2: Algorithms
  - ✓ review of main algorithms & complexities
- Session 3: Guarantees for Deterministic *vs* Random dictionaries
  - ✓ compared success guarantees for different algorithms
  - ✓ robust guarantees & Restricted Isometry Property
  - ✓ explicit guarantees for various inverse problems

Overview of Session 1

- Sparsity and compression of large-scale data
- Sparsity for source separation, inverse problems, and learning
- Sparse decomposition algorithms
  - ✓ L1 minimisation
  - ✓ Matching Pursuits
- Provably good algorithms
- Sparsity and compressed sensing

Sparsity & data compression

Large-scale data

- **Fact** : digital data = large volumes

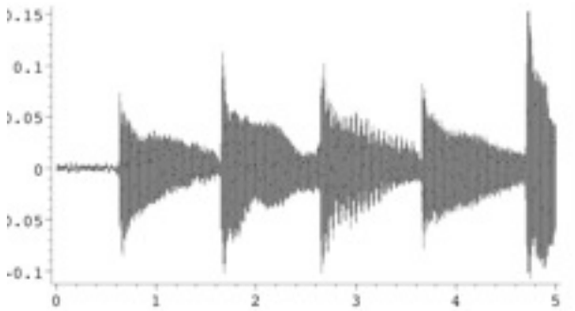
  - ✓ 1 second stereo audio, CD quality = 1,4 Mbit
  - ✓ 1 uncompressed 10 Mpixels picture = 240 Mbit

- **Need** : «concise» data representations

  - ✓ storage & transmission (volume / bandwidth) ...
  - ✓ manipulation & processing (algorithmic complexity)

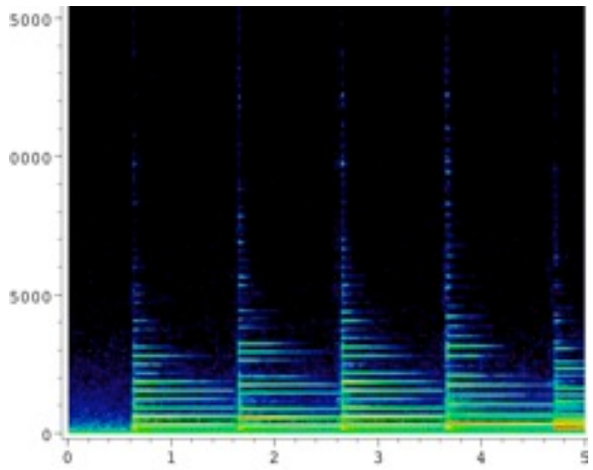
Sparse representations

- Audio : time-frequency representations (MP3)



ANALYSIS

SYNTHESIS



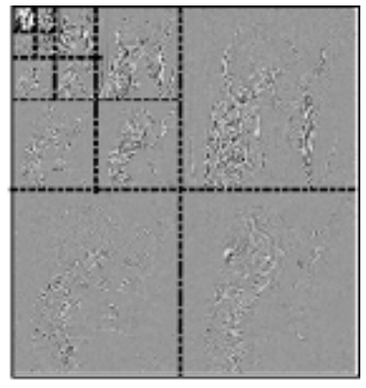
Black = zero

- Images : wavelet transform (JPEG2000)

ORIGINAL
128, 129, 125, 64, 65,

ANALYSIS

SYNTHESIS



TRANSFORM COEFFICIENTS
4123, -12.4, -96.7, 4.5,

Gray = zero

Mathematical expression

- Signal / image = high dimensional vector

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

- **Model** = linear combination of basis vectors (ex: *time-frequency atoms, wavelets*)

$$y \approx \sum_k x_k \varphi_k = \mathbf{\Phi} x$$

atoms
dictionary

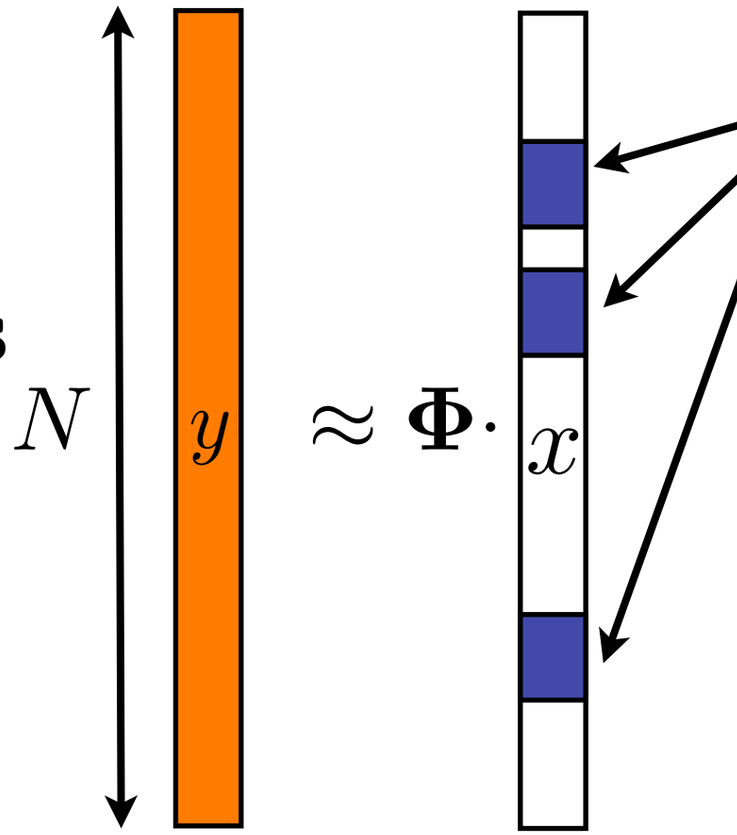
- **Sparsity** = small L0 (quasi)-norm

$$\|x\|_0 = \sum_k |x_k|^0 = \mathrm{card}\{k, x_k \neq 0\}$$

Sparsity & compression

- Full vector

N entries
= N `floats`



- Sparse vector

$k \ll N$ nonzero entries
= k `floats`

+ k positions among N

$$= \log_2 \binom{N}{k} \approx k \log_2 \frac{N}{k} \texttt{ bits}$$

Key practical issues: choose dictionary

Sparsity & inverse problems

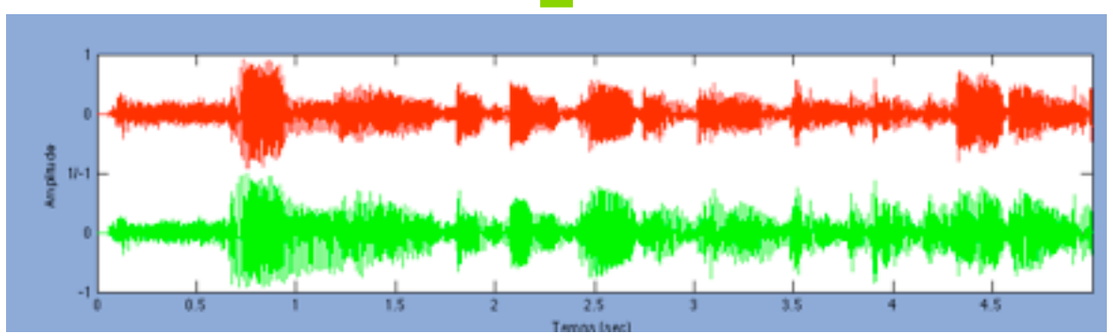
Example: image inpainting

Courtesy of: G. Peyré, Ceremade, Université Paris 9 Dauphine

Inpainting

Example : audio source separation

- « Softly as in a morning sunrise »

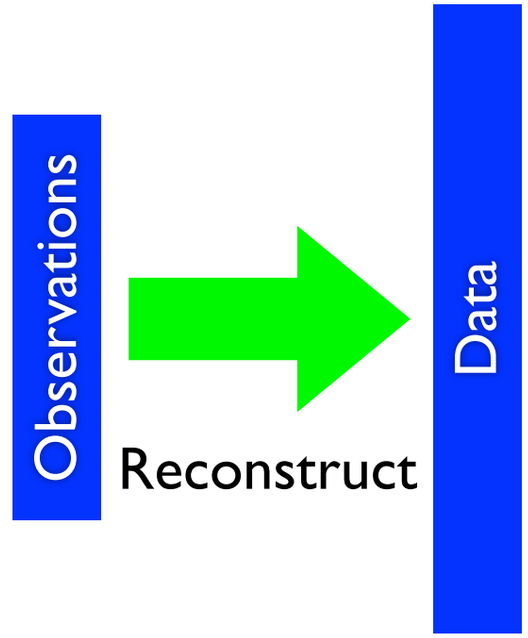


Inverse problems

- **Inverse problem** : exploit indirect or incomplete obervation to recontruct some data

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

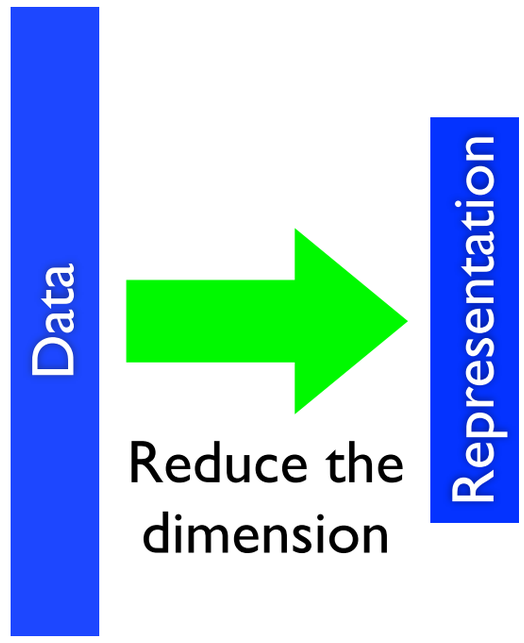
fewer equations than unknowns



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

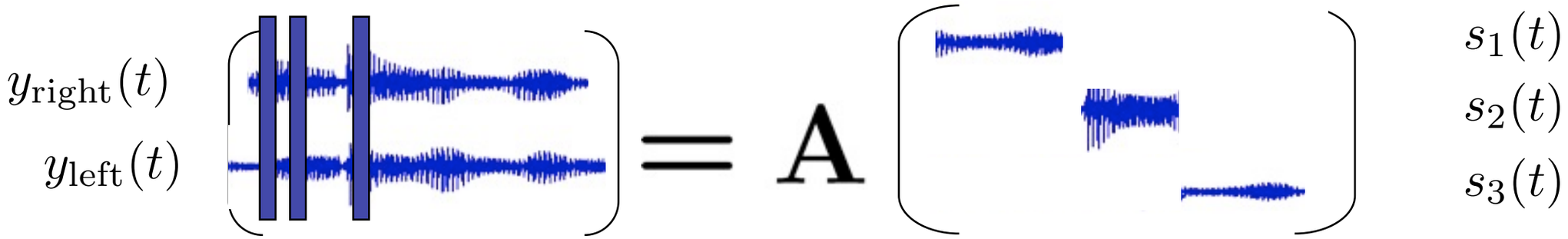
$$\mathbf{y} \approx \mathbf{\Phi} x$$

few nonzero components

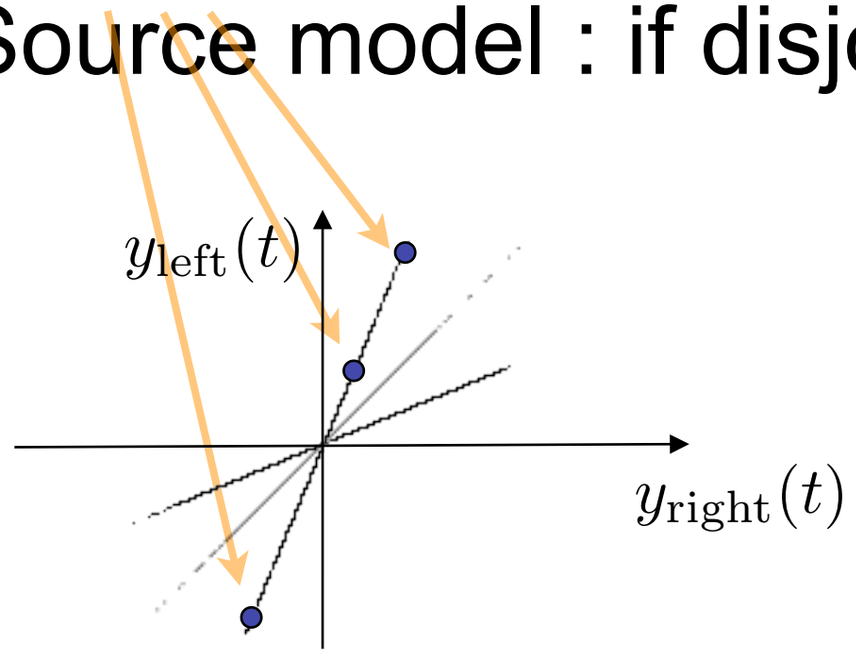


Blind Source Separation

- Mixing model : linear instantaneous mixture



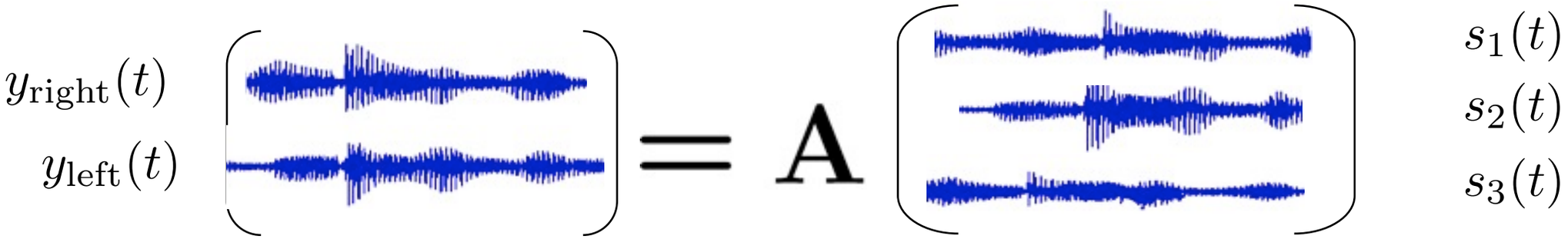
- Source model : if disjoint time-supports …



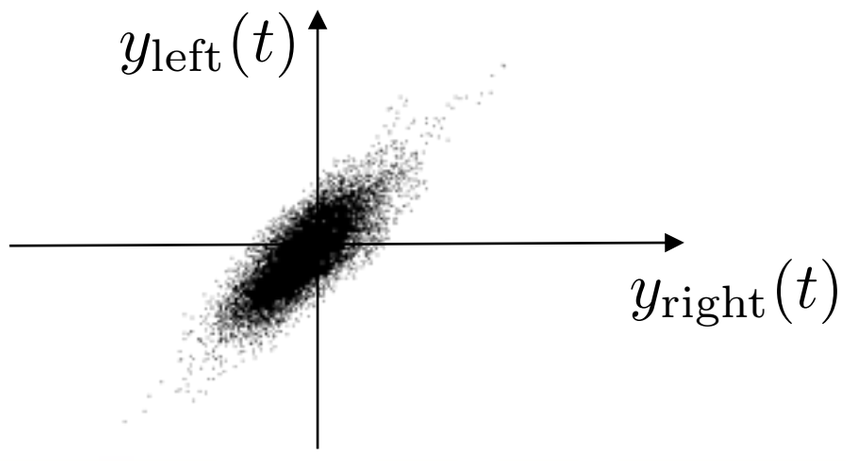
... then clustering to :
1- identify (columns of) the mixing matrix
2- recover sources

Blind Source Separation

- Mixing model : linear instantaneous mixture



- In practice ...

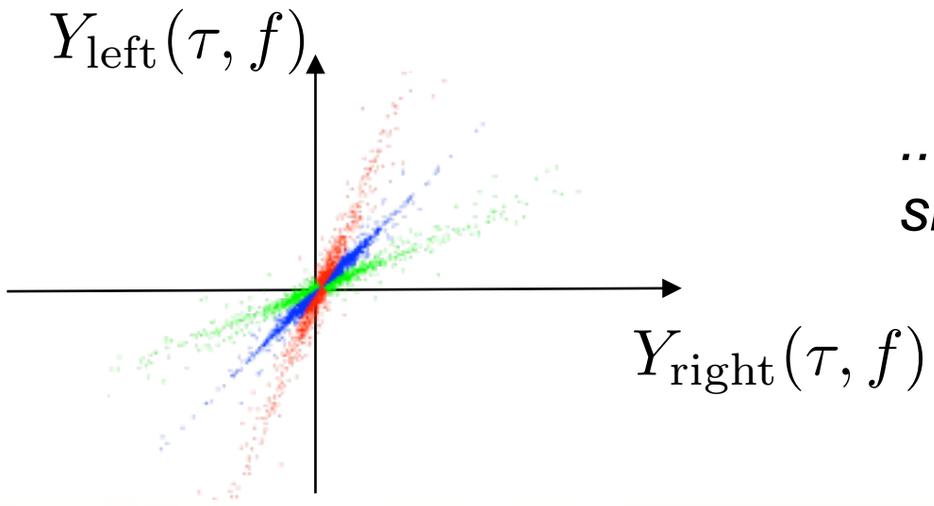


Time-Frequency Masking

- Mixing model in the time-frequency domain

$$\begin{matrix} Y_{\text{right}}(\tau, f) \\ Y_{\text{left}}(\tau, f) \end{matrix} \left(\begin{matrix} \\ \\ \end{matrix} \right) = \mathbf{A}\, \mathbf{S}(\tau, f)$$

- And “miraculously” ...



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

Inverse Problems & Sparsity: Mathematical foundations

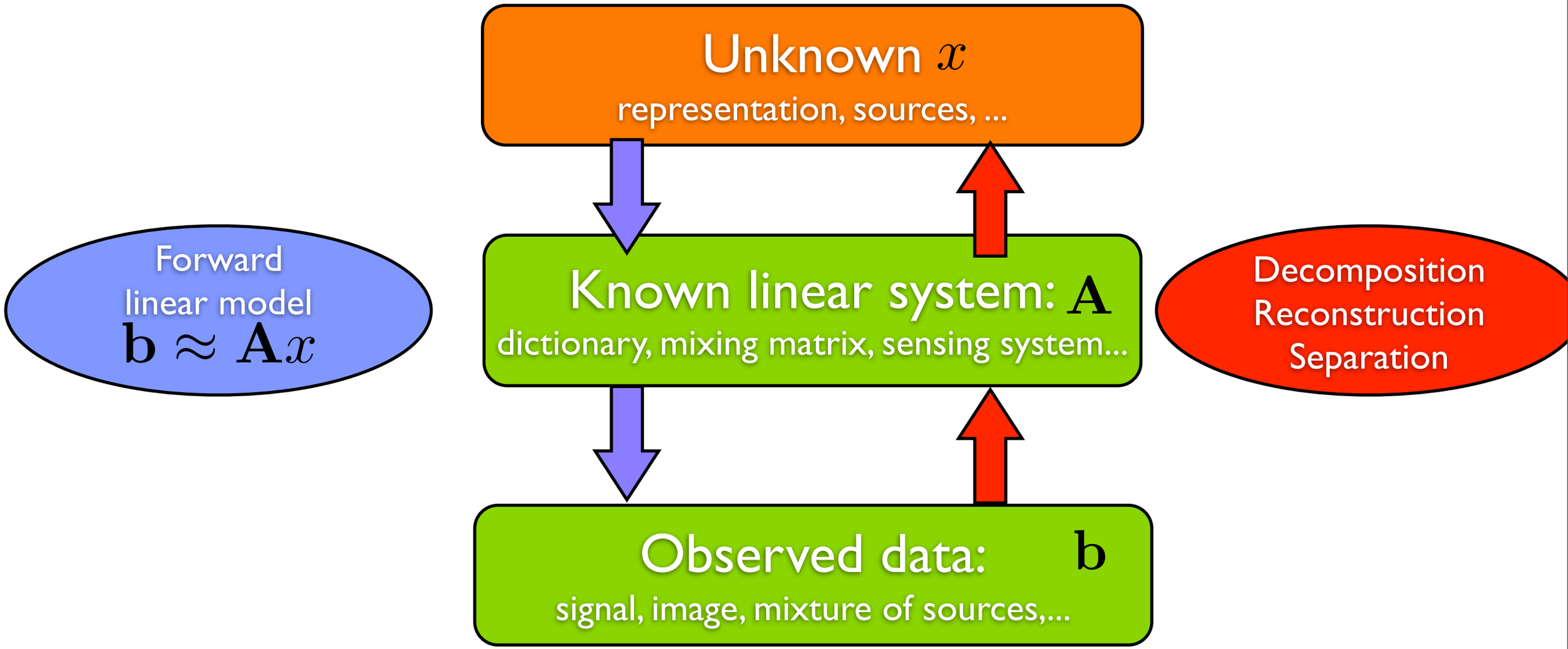
- Bottleneck 1990-2000 : fewer equations than unknowns

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

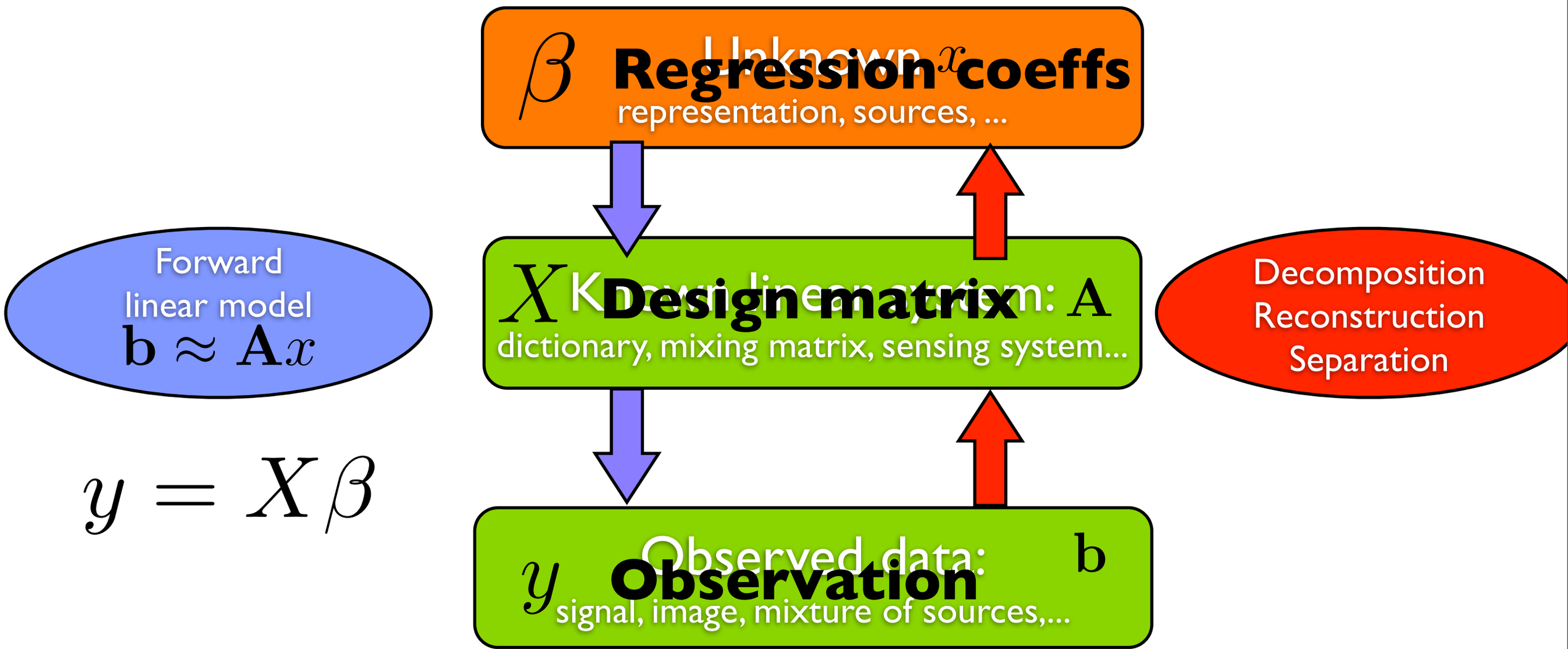
- Novelty 2001-2006 :
  - ✓ Uniqueness of sparse solution:
    - ✦ if x_0, x_1 are "sufficiently sparse",
    - ✦ then $\mathbf{A}x_0 = \mathbf{A}x_1 \Rightarrow x_0 = x_1$
  - ✓ Recovery of x_0 with practical algorithms
    - ✦ Thresholding, Matching Pursuits, Minimisation of Lp norms p<=1,...

Algorithmic principles for sparse approximation

Signal Processing Vocabulary



(My Basic Understanding of) Machine Learning Vocabulary



Ideal sparse approximation

- Input:

  m x N matrix **A**, with $m < N$, m-dimensional vector **b**

- Possible objectives:

  find the sparsest approximation within tolerance

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

  find best approximation with given sparsity

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

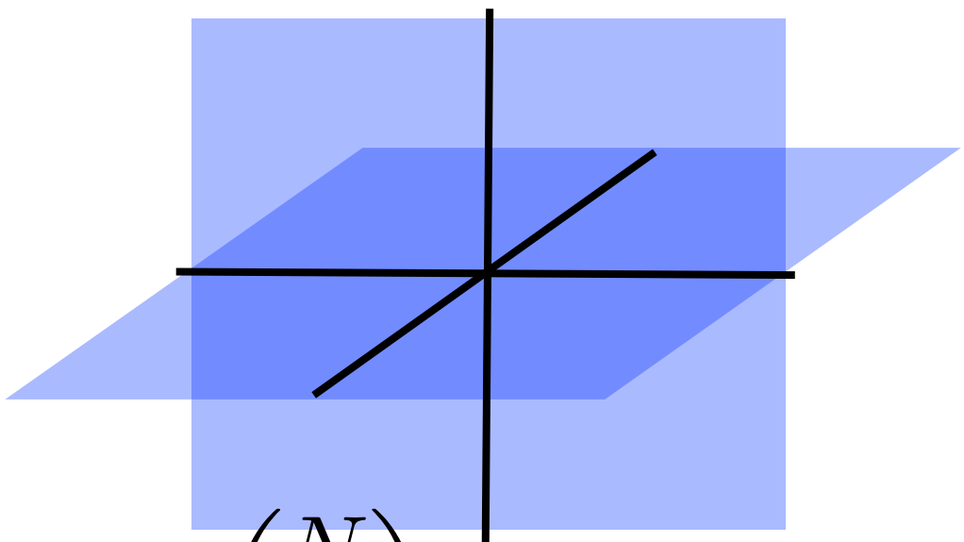
  find a solution x to

$$\|\mathbf{b} - \mathbf{A}x\| \leq \epsilon, \ \text{and} \ \|x\|_0 \leq k$$

Geometric interpretation of sparse approximation

- Coefficient domain $\mathbb{R}^N$:
  - ✓ set Σ_k of sparse vectors

$$\|x\|_0 \leq k$$



$\binom{N}{k}$ subspaces

- Set $\mathbf{A}\Sigma_k$ = $\binom{N}{k}$ subspaces in signal domain

- Ideal sparse approximation = find nearest subspace among $\binom{N}{k}$

Combinatorial search!
Actual complexity ?
NP-complete!

Practical approaches:
Optimization *principles*

Overall compromise

- Approximation quality

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

- Ideal sparsity measure : ℓ^0 “norm”

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

- “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

Quasi-triangle inequality $\|x + y\|_p \leq 2^{1/p} \left(\|x\|_p + \|y\|_p \right), \forall x, y$

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

- "*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

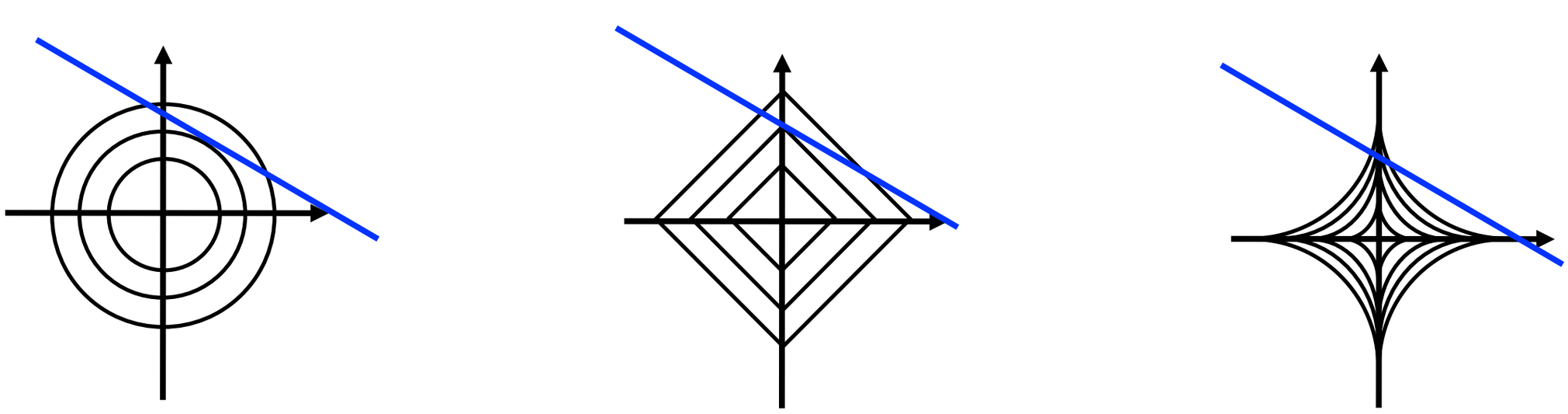
$$\min_x \|x\|_p \text{ s.t. } \|\mathbf{b} - \mathbf{A}x\|_2 \leq \epsilon$$

- 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*

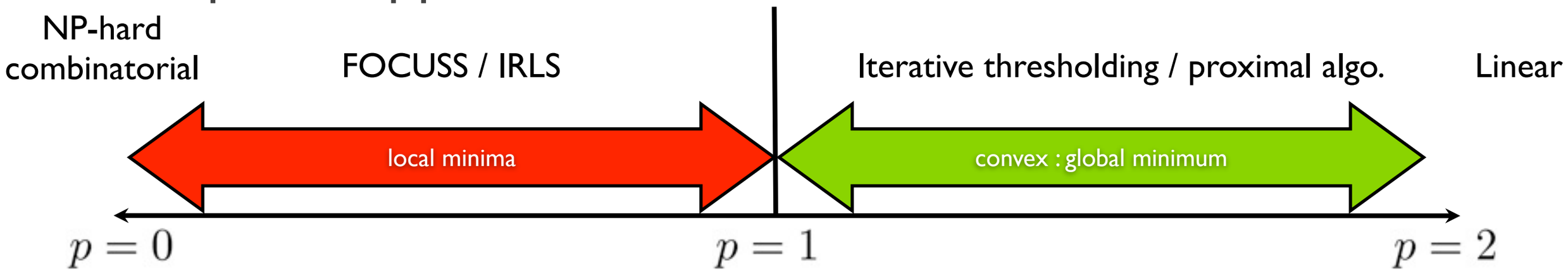
$\{x \text{ s.t.} \mathbf{b} = \mathbf{A}x\}$

Global Optimization : from Principles to Algorithms

- Optimization principle

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

  - ✓ Sparse representation $\lambda \to 0 \quad \mathbf{A}x = \mathbf{b}$
  - ✓ Sparse approximation $\lambda > 0 \quad \mathbf{A}x \approx \mathbf{b}$



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, ...]

Greedy Algorithms

Greedy algorithms

- Observation: when **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

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

Greedy algorithms

- Iterative algorithm (= *Matching Pursuit*)
  - ✓ Initialize a residual to $\mathbf{r}_0 = \mathbf{b} \qquad i = 1$
  - ✓ Compute all inner products

$$\mathbf{A}^T \mathbf{r}_{i-1} = (\mathbf{a}_n^T \mathbf{r}_{i-1})_{n=1}^N$$

  - ✓ Select the largest in magnitude

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

  - ✓ Compute an updated residual

$$\mathbf{r}_i = \mathbf{r}_{i-1} - (\mathbf{a}_{n_i}^T \mathbf{r}_{i-1}) \mathbf{a}_{n_i}$$

  - ✓ If $i \geq k$ then stop, otherwise increment *i* and iterate

Matching Pursuit (MP)

- Matching Pursuit (*aka* Projection Pursuit, CLEAN)
  - ✓ Initialization $\mathbf{r}_0 = \mathbf{b} \qquad i = 1$
  - ✓ Atom selection:

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

  - ✓ Residual update

$$\mathbf{r}_i = \mathbf{r}_{i-1} - (\mathbf{a}_{n_i}^T \mathbf{r}_{i-1})\mathbf{a}_{n_i}$$

- Energy preservation (Pythagoras theorem)

$$\|\mathbf{r}_{i-1}\|_2^2 = |\mathbf{a}_{n_i}^T \mathbf{r}_{i-1}|^2 + \|\mathbf{r}_i\|_2^2$$

Summary

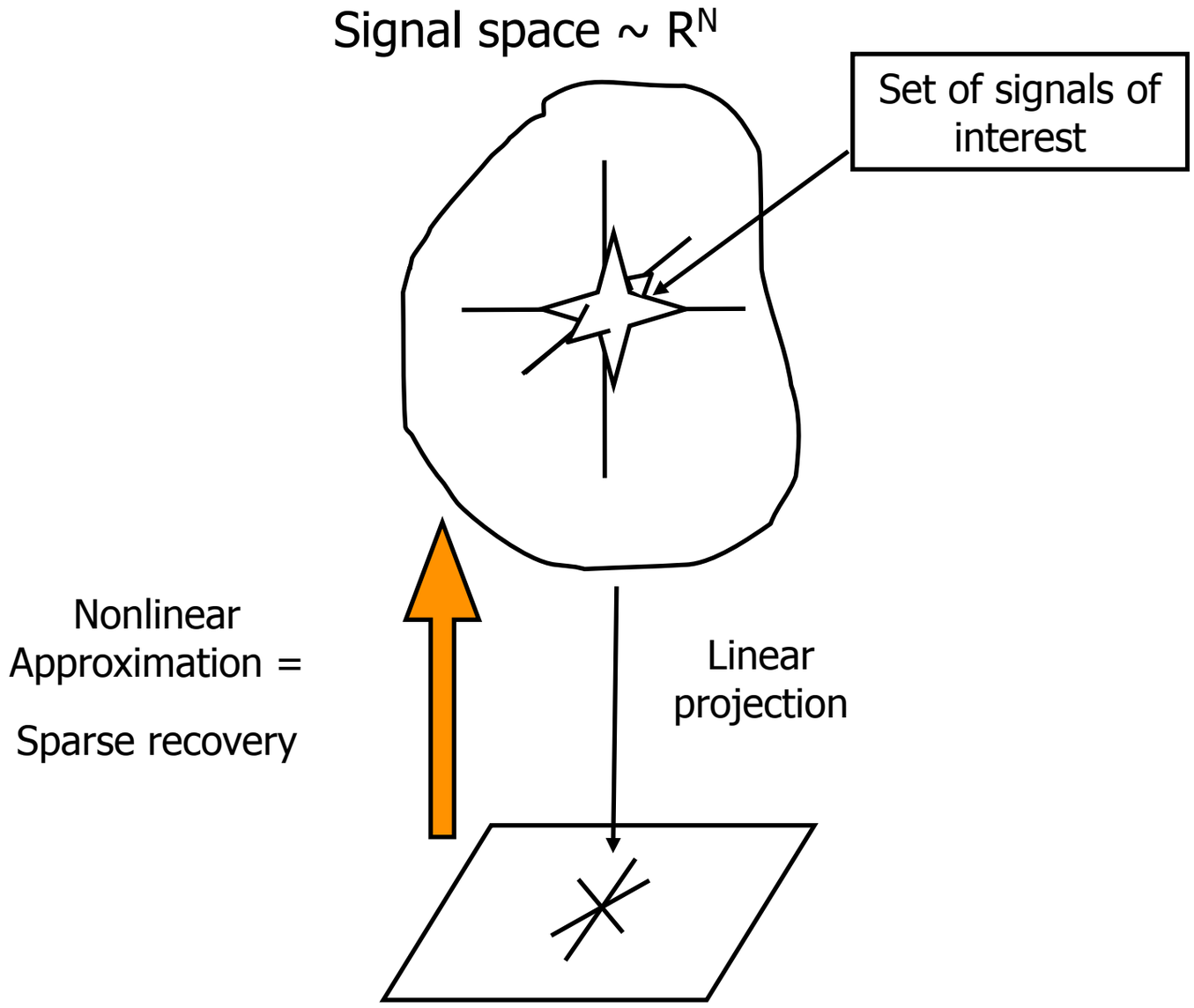
Global optimization

Iterative greedy algorithms

Principle	$\min_x \frac{1}{2}\|\mathbf{A}x - \mathbf{b}\|_2^2 + \lambda\|x\|_p^p$	iterative decomposition $\mathbf{r}_i = \mathbf{b} - \mathbf{A}x_i$ • select new components • update residual
Tuning quality/sparsity	regularization parameter λ	stopping criterion (nb of iterations, error level, ...) $\|x_i\|_0 \geq k$ $\|\mathbf{r}_i\| \leq \epsilon$
Variants	• choice of sparsity measure p • optimization algorithm • initialization	•selection criterion (weak, stagewise ...) •update strategy (orthogonal ...)

Provably good algorithms

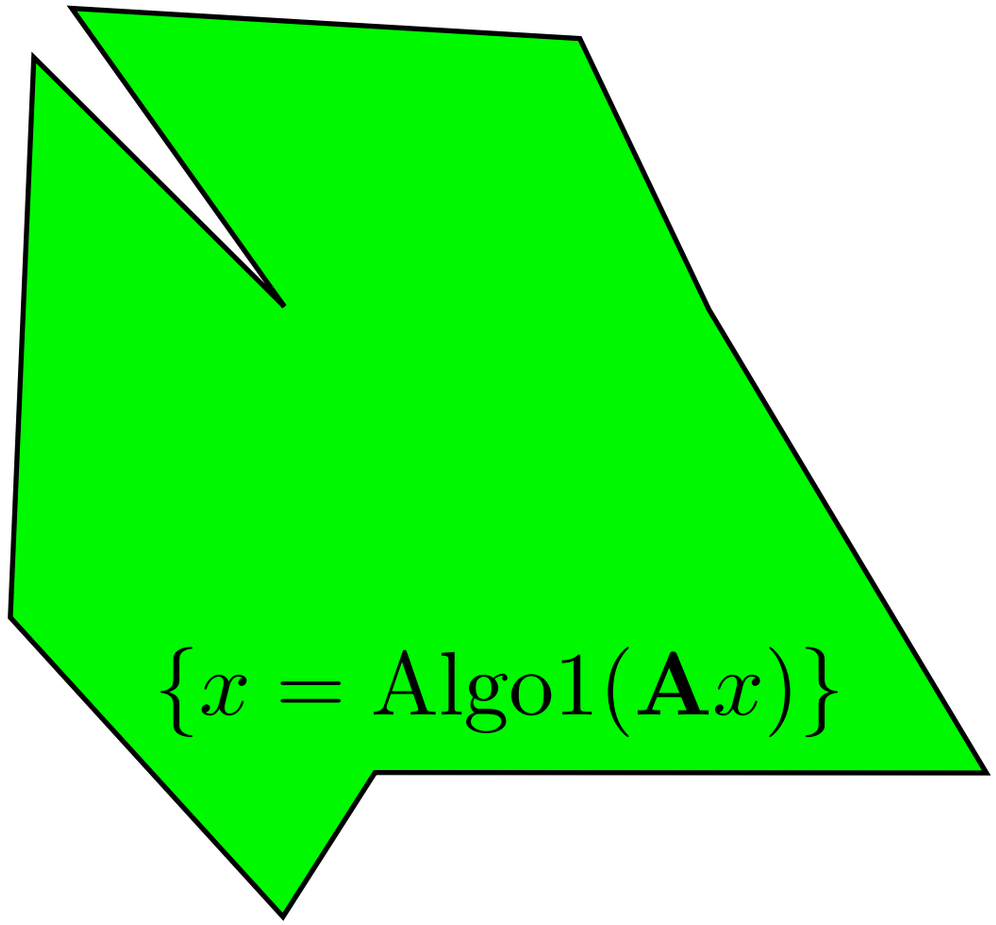
Inverse problems



Courtesy: M. Davies, U. Edinburgh

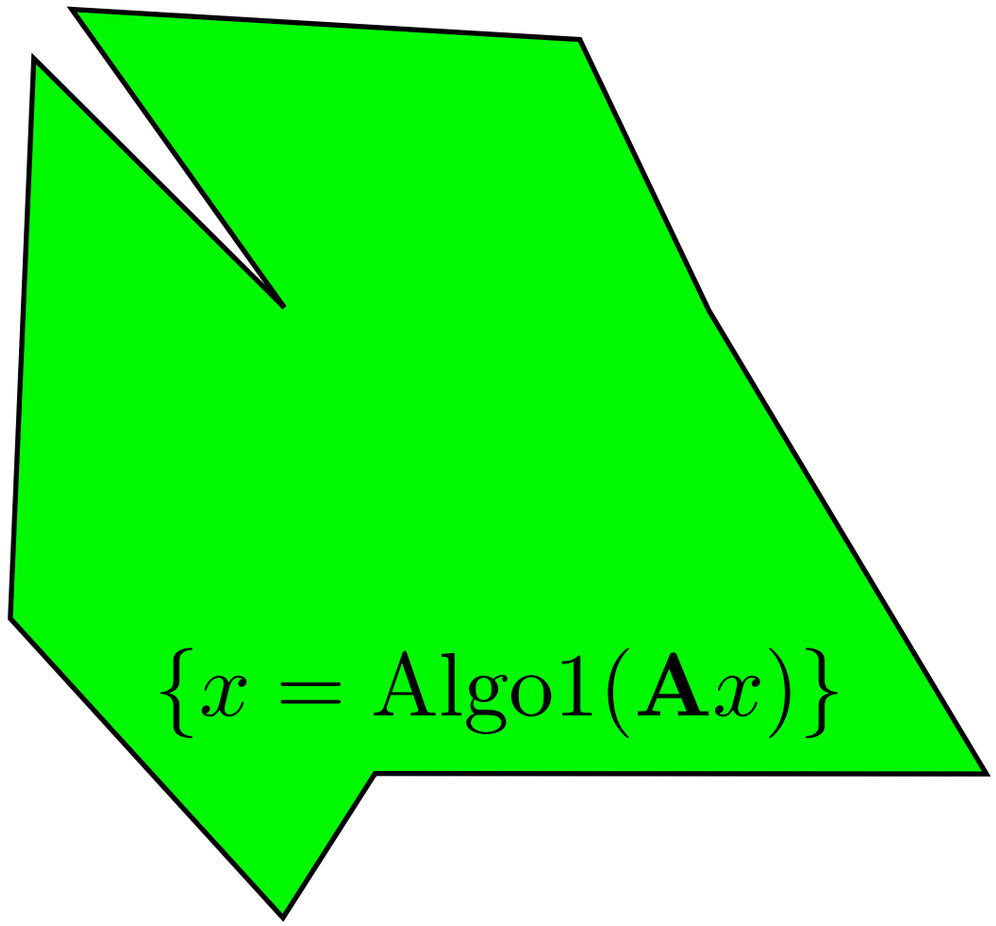
Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

- Recoverable set for a given “inversion” algorithm



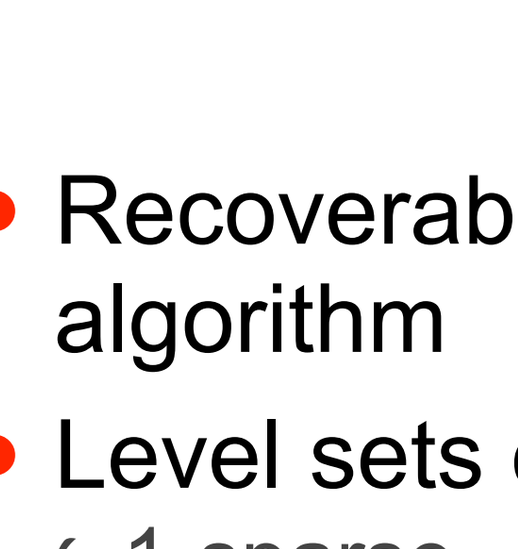
Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

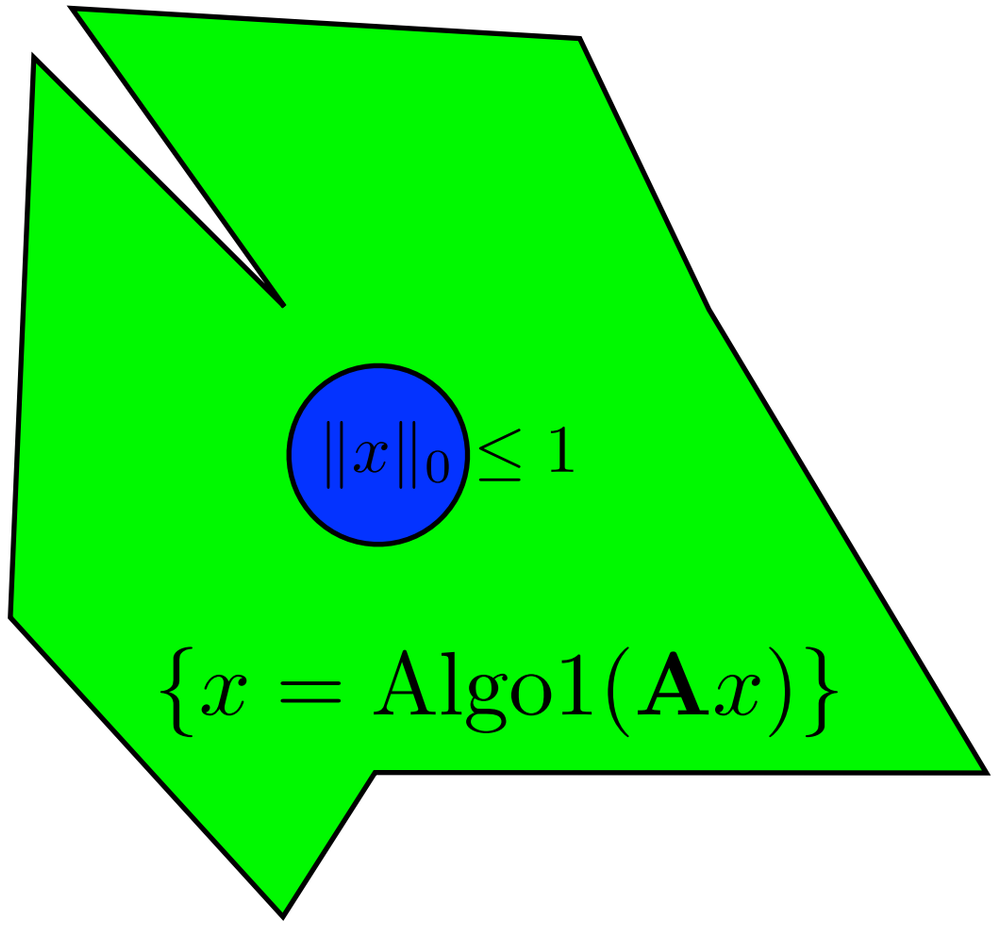
- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm



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

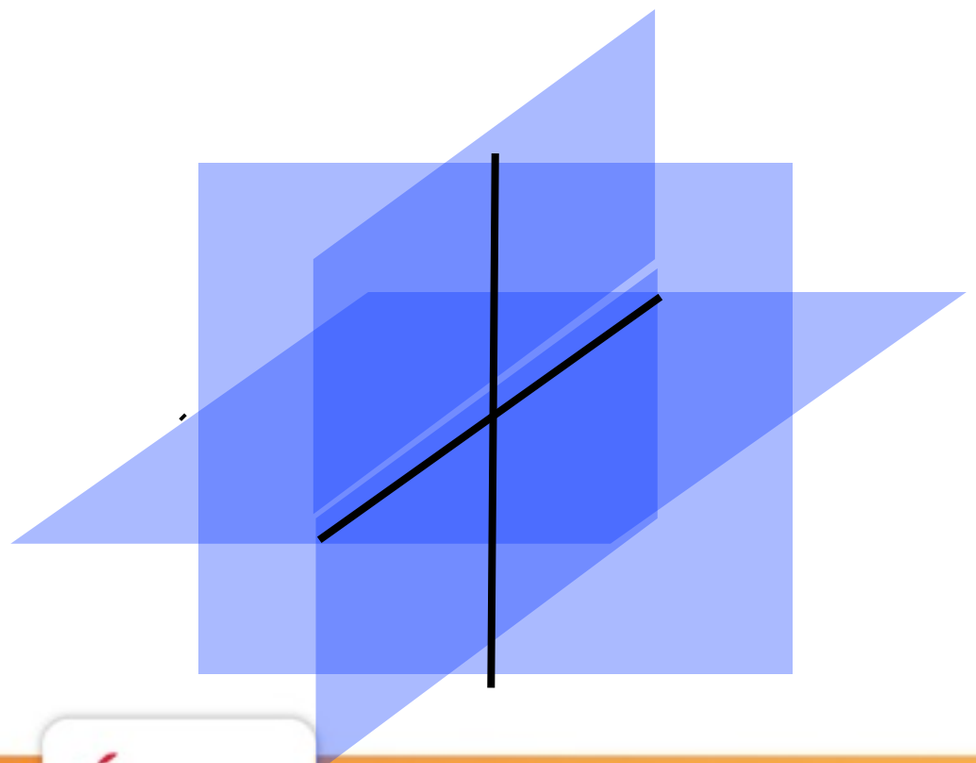
- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
  - ✓ 1-sparse

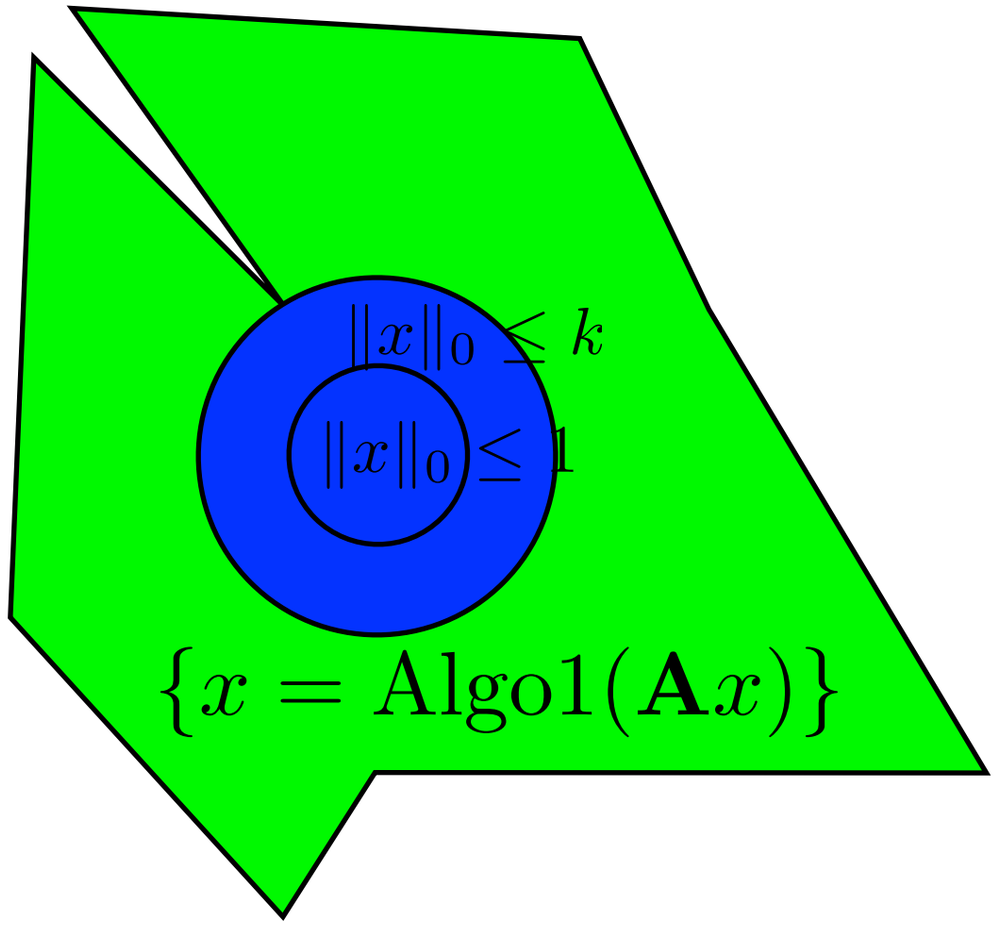




Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

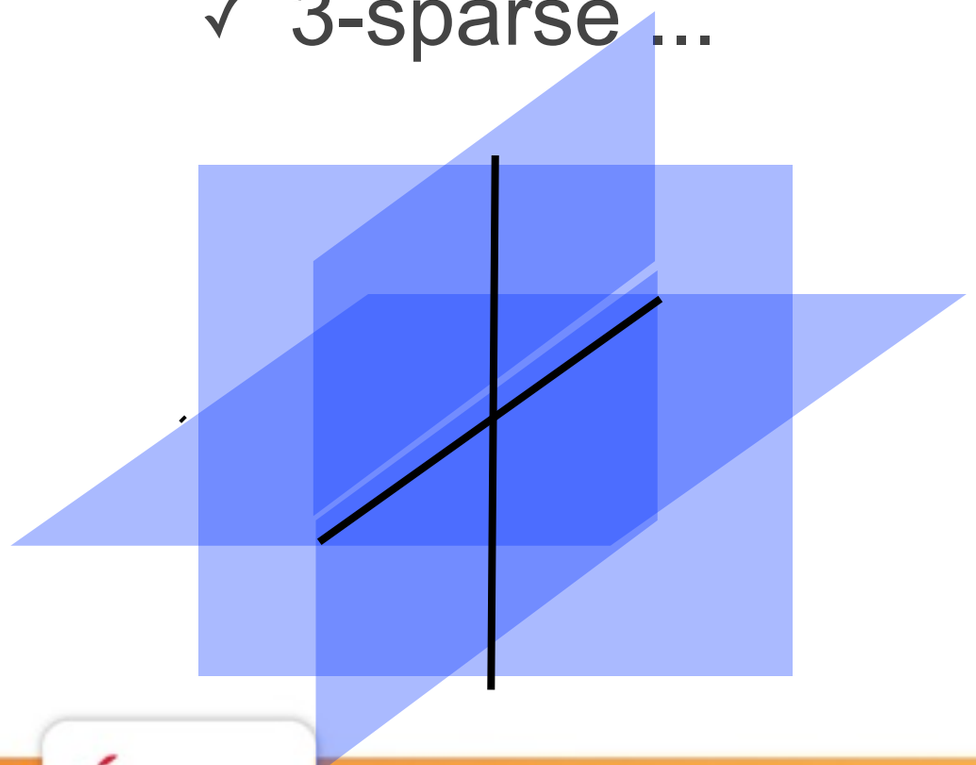
- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
  - ✓ 1-sparse
  - ✓ 2-sparse

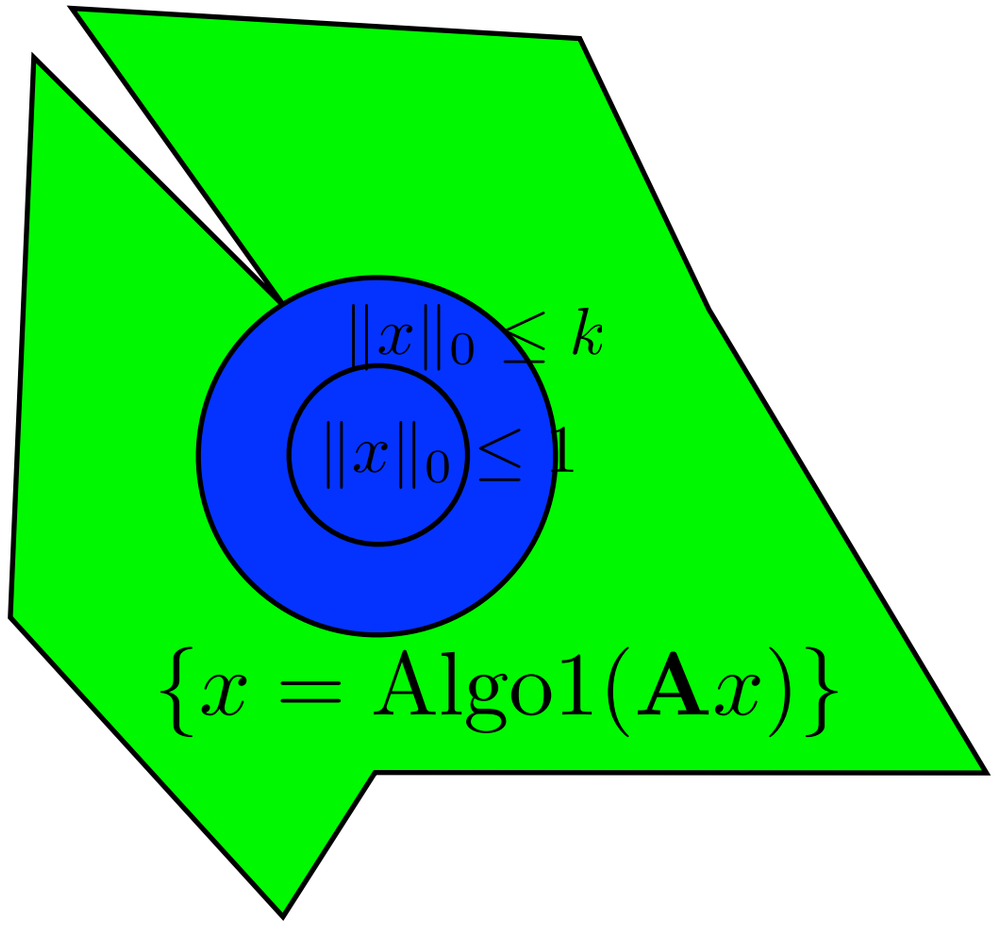




Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

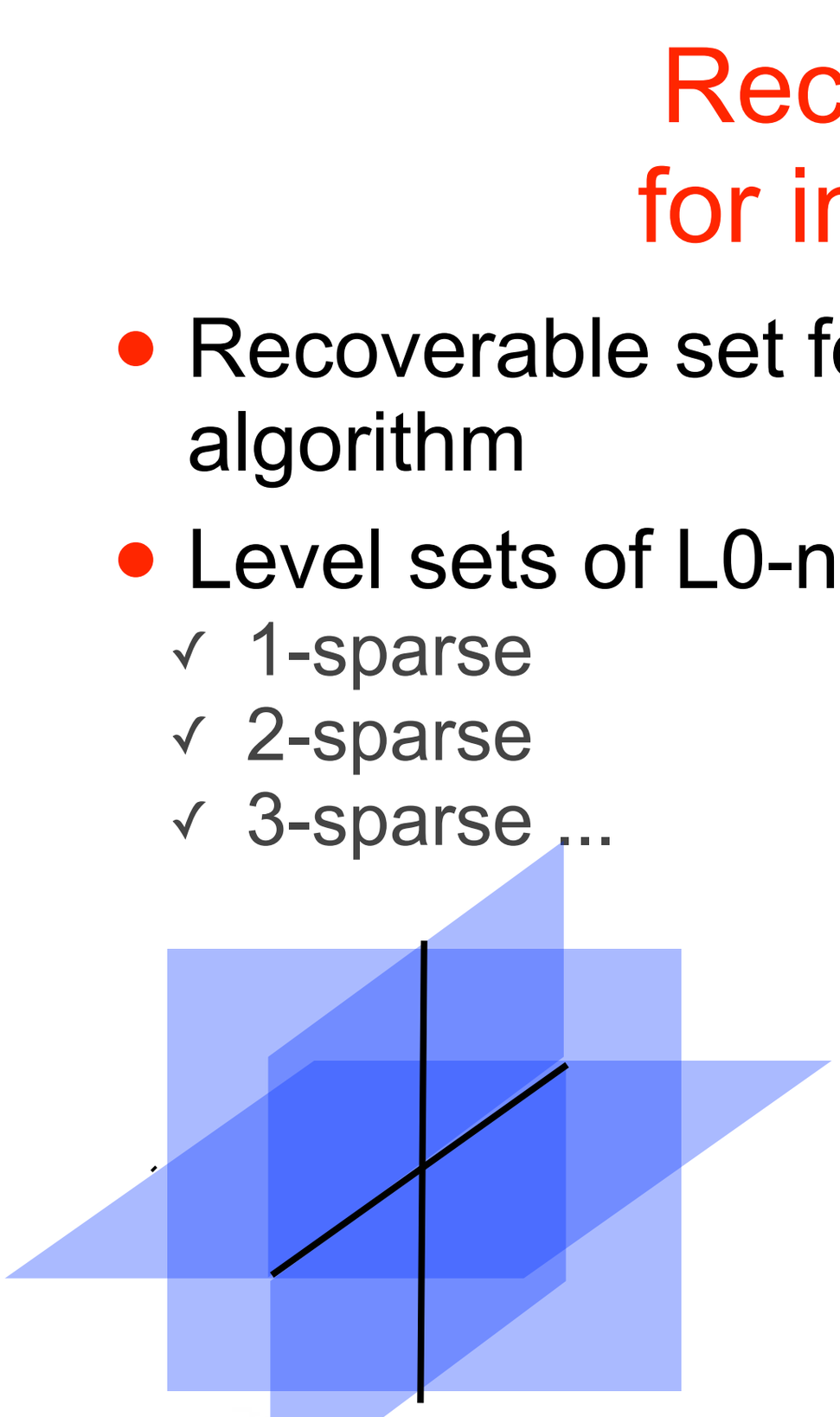
- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
  - ✓ 1-sparse
  - ✓ 2-sparse
  - ✓ 3-sparse ...

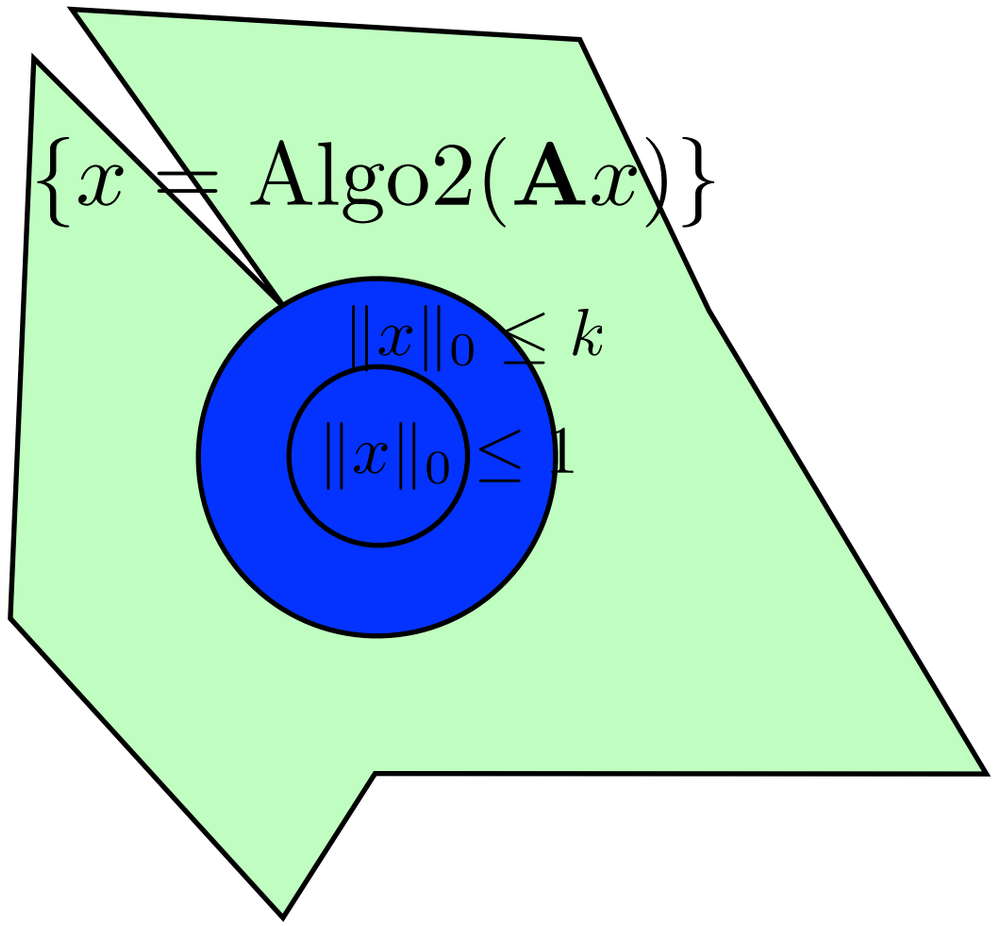




Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

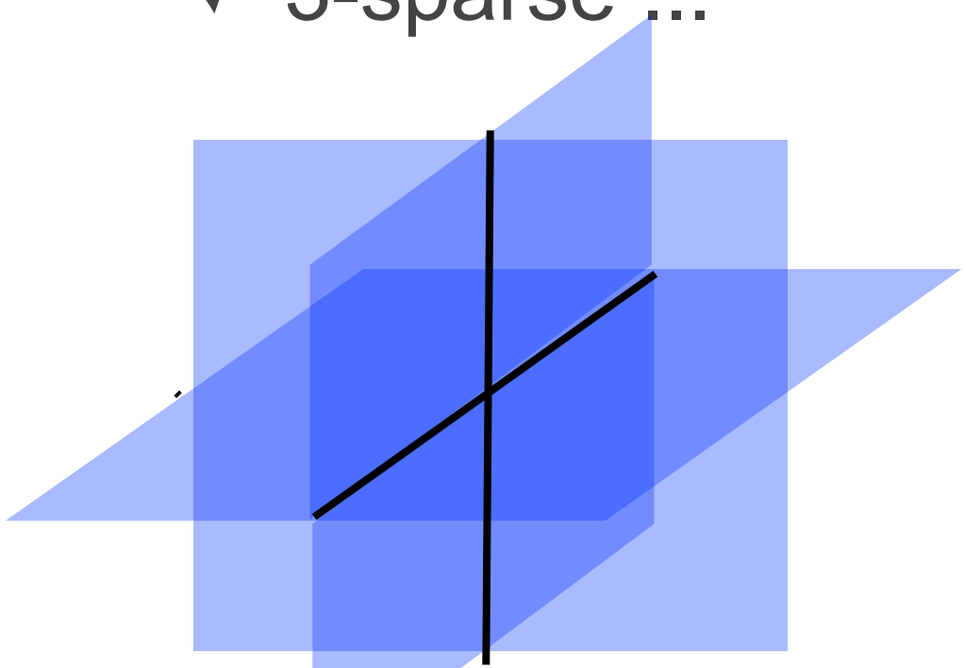
- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
  - ✓ 1-sparse
  - ✓ 2-sparse
  - ✓ 3-sparse ...

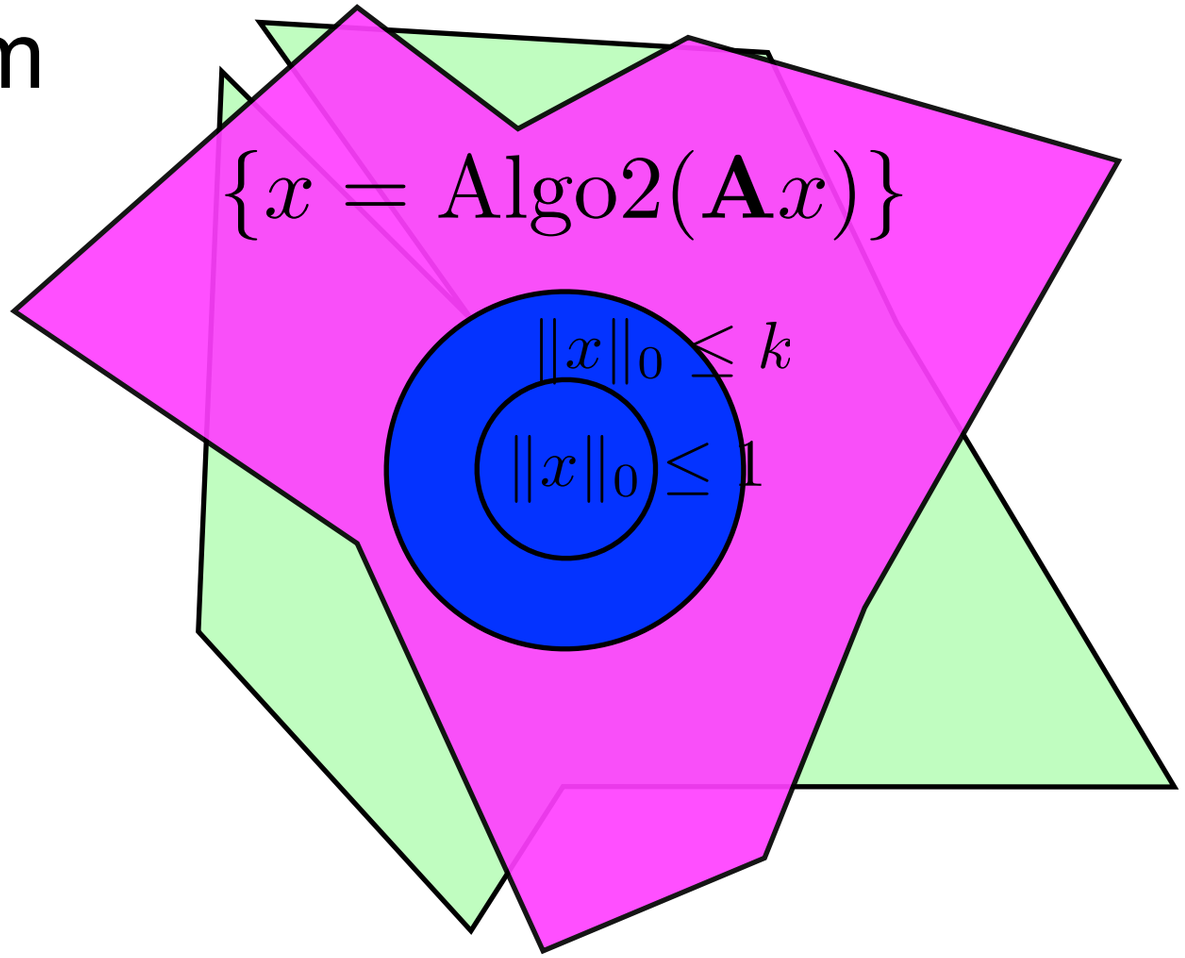




Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

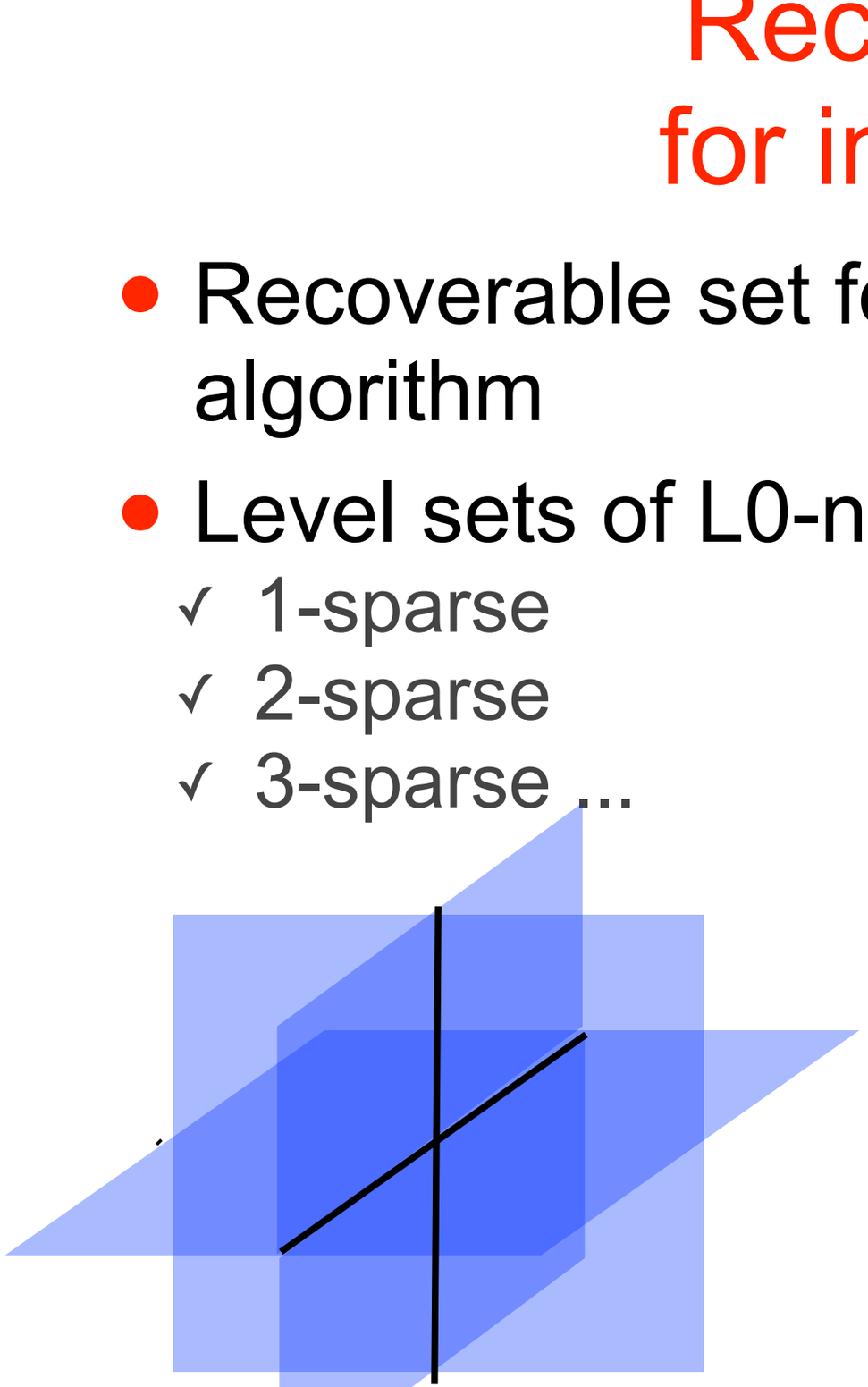
- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
  - ✓ 1-sparse
  - ✓ 2-sparse
  - ✓ 3-sparse ...

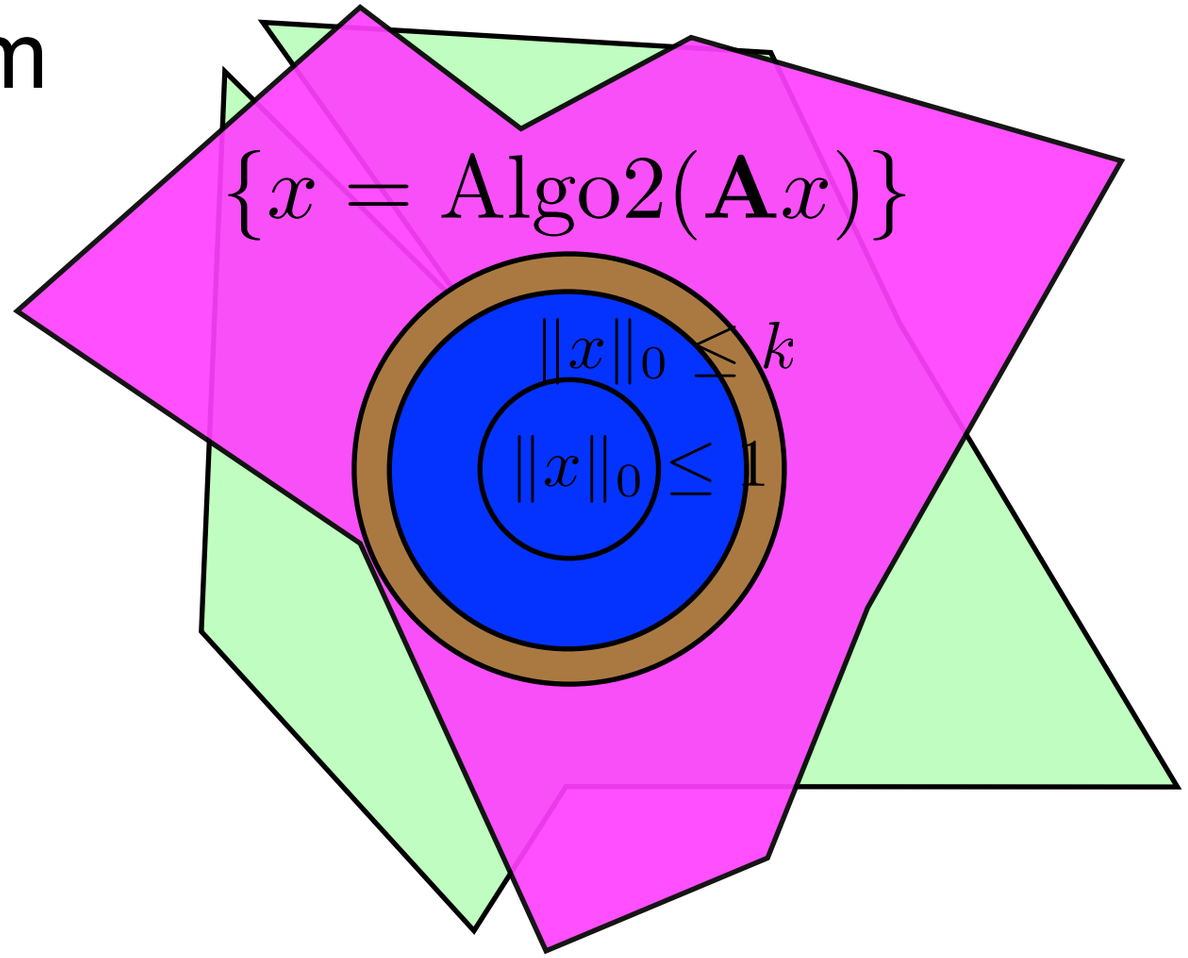




Recovery analysis for inverse problem $\mathbf{b} = \mathbf{A}x$

- 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 $x_0 = x_0^\star$
  - ✓ 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*

Compressed sensing

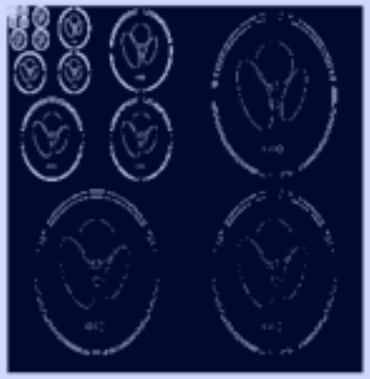
Example: tomography

- MRI from incomplete data

[Candès, Romberg & Tao]

Model / knowledge

The (unknown) wavelet transform is sparse



x

$y = \Phi x$

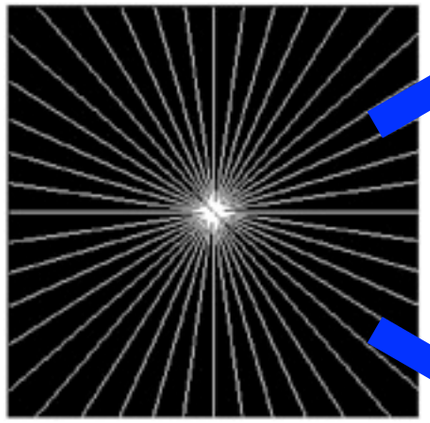
Data to be captured

y

$z = \mathbf{M}y$

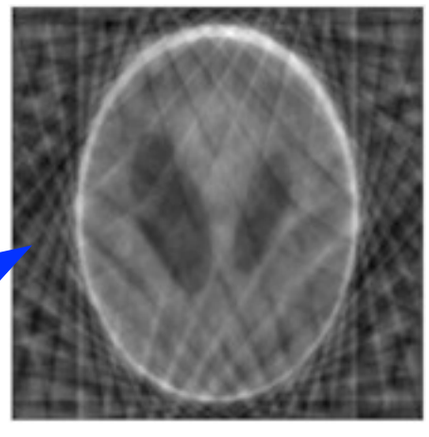
Tomography = incomplete projection

Measured observations (incomplete FFT)



z

FFT^{-1}



Reconstruction

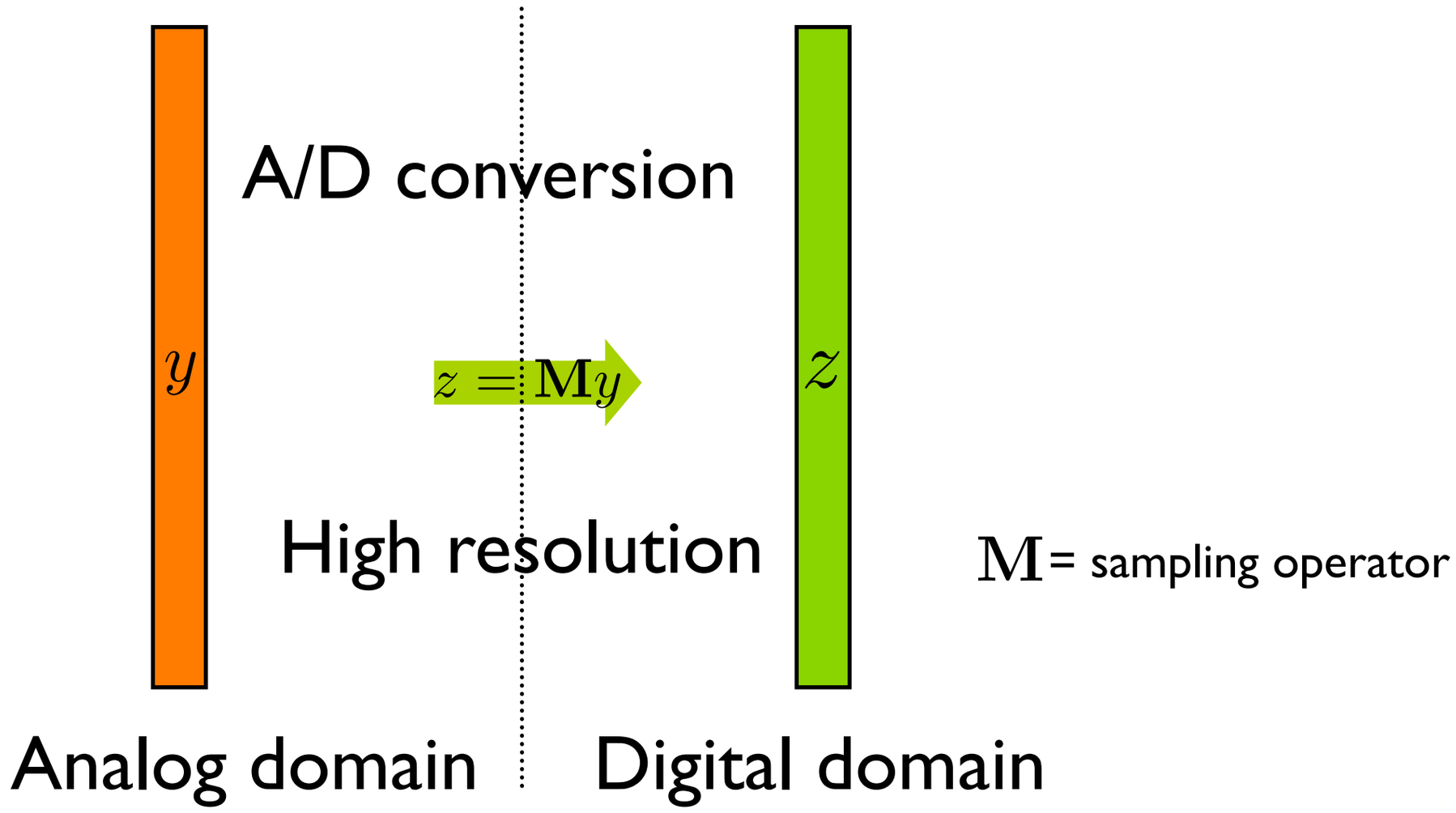
Sparse reconstruction (Candes et al 2004)

Analog domain *Digital domain*

$$\min \|x\|_1 \text{ s.t. } z = \mathbf{M}\Phi x$$

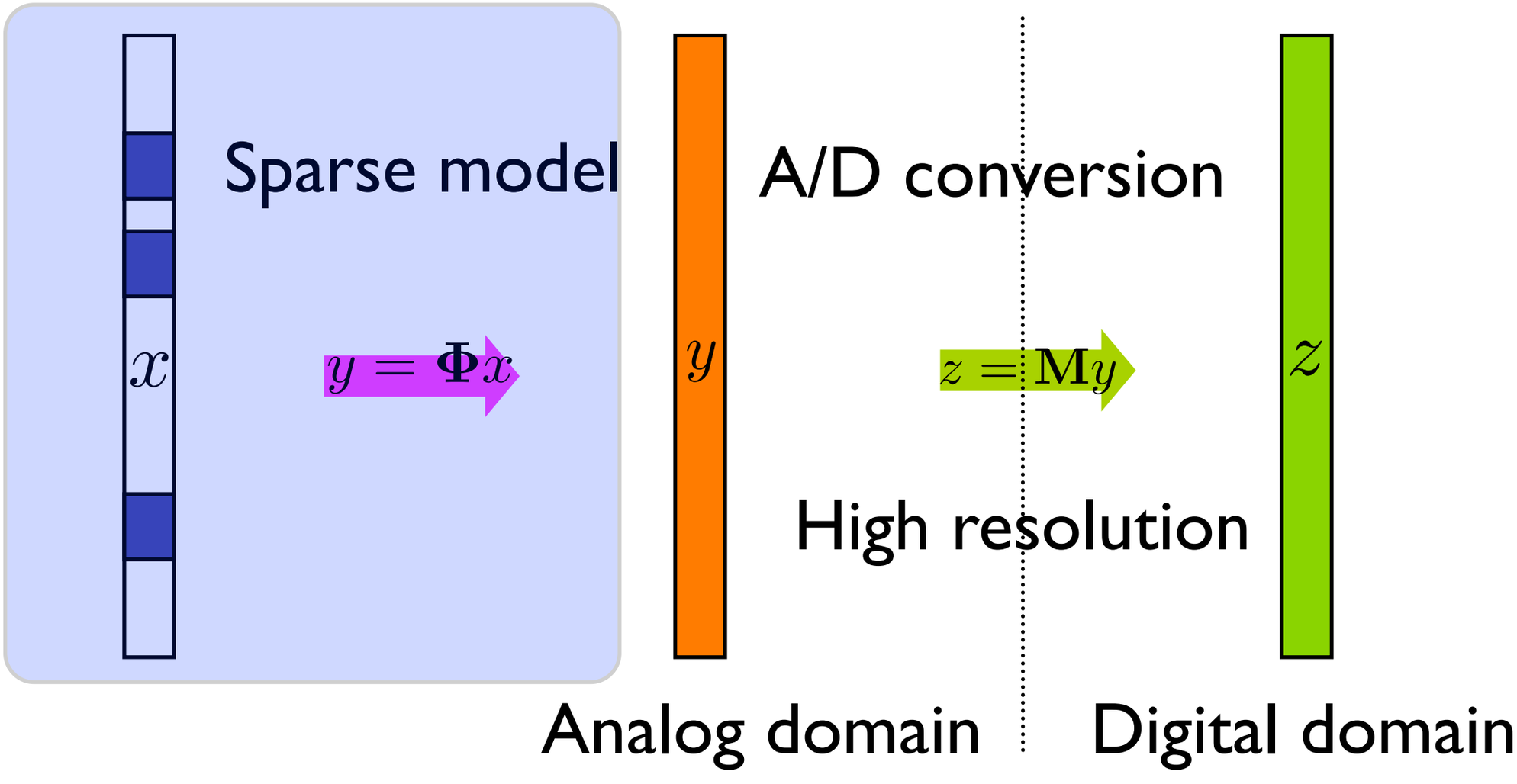
Classical Shannon Sampling

- « Sample first, think and compress afterwards »



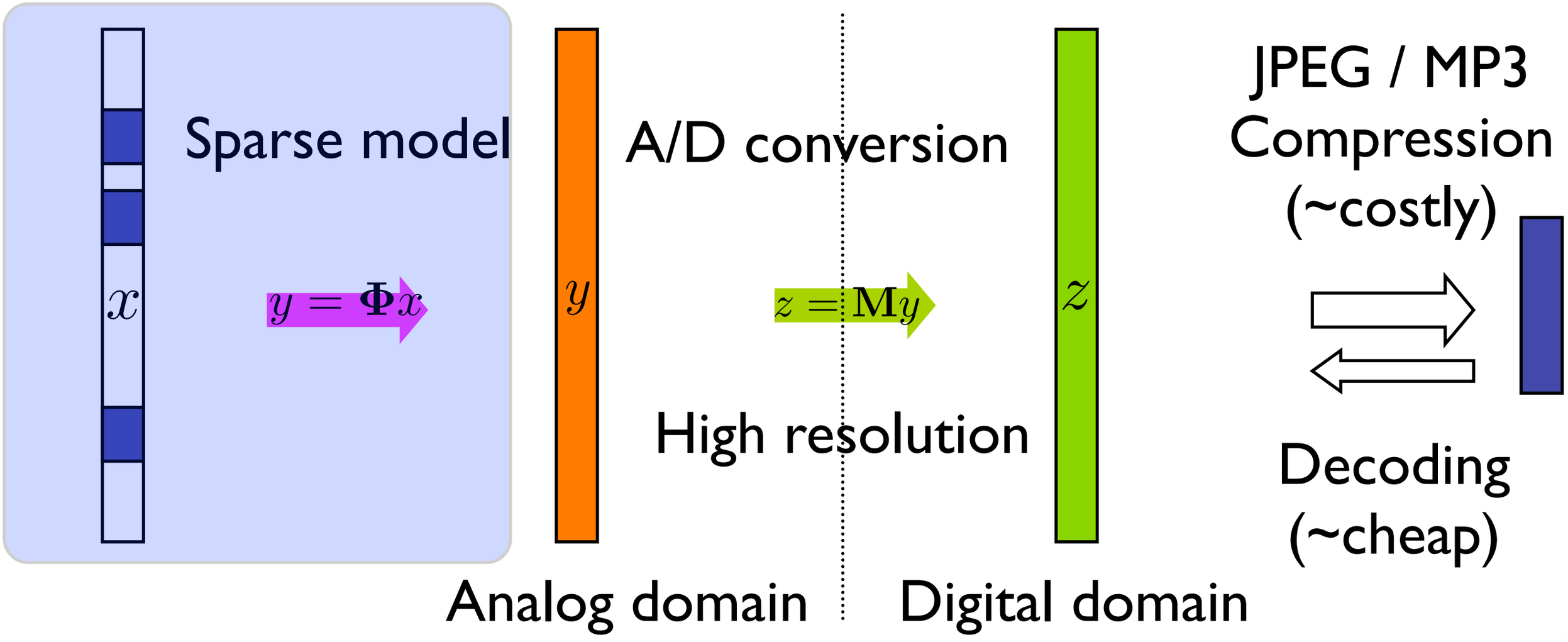
Classical Shannon Sampling

- « Sample first, think and compress afterwards »



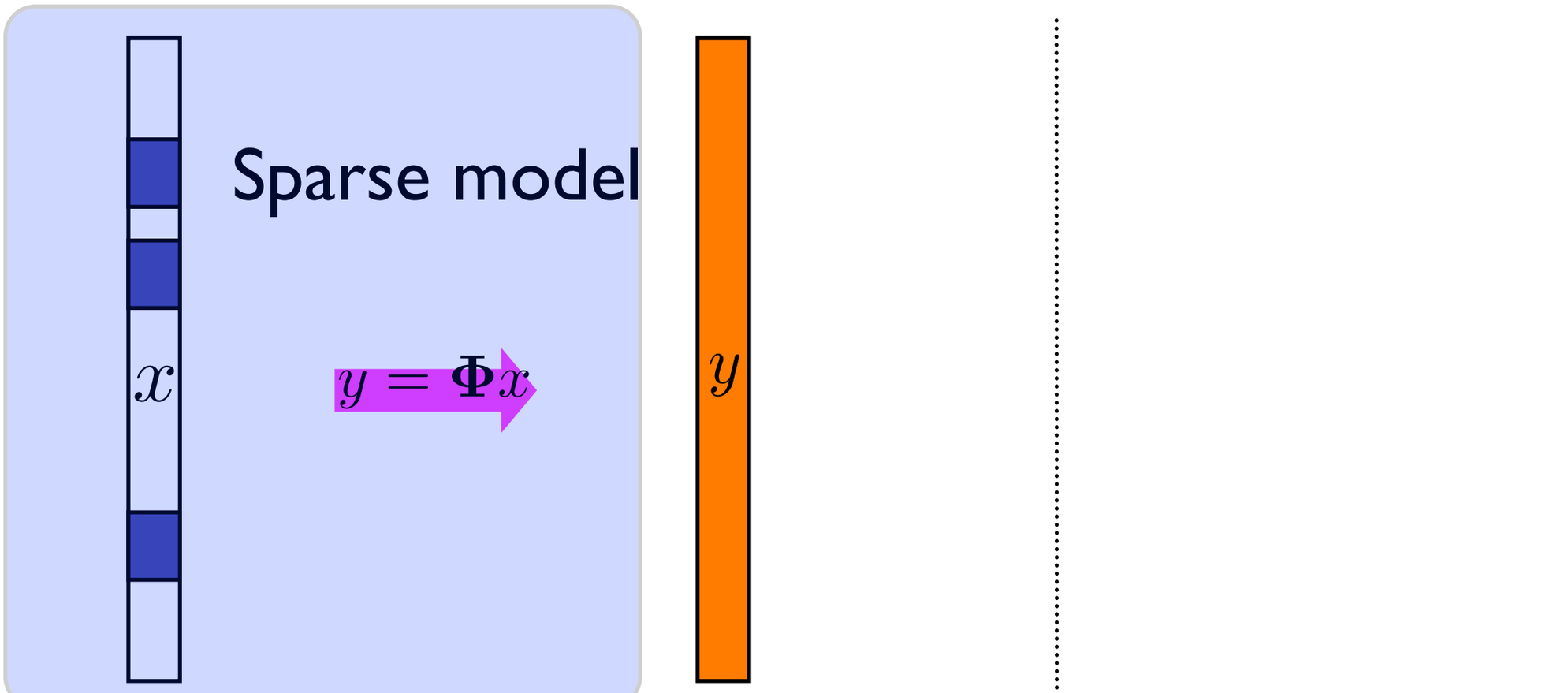
Classical Shannon Sampling

- « Sample first, think and compress afterwards »



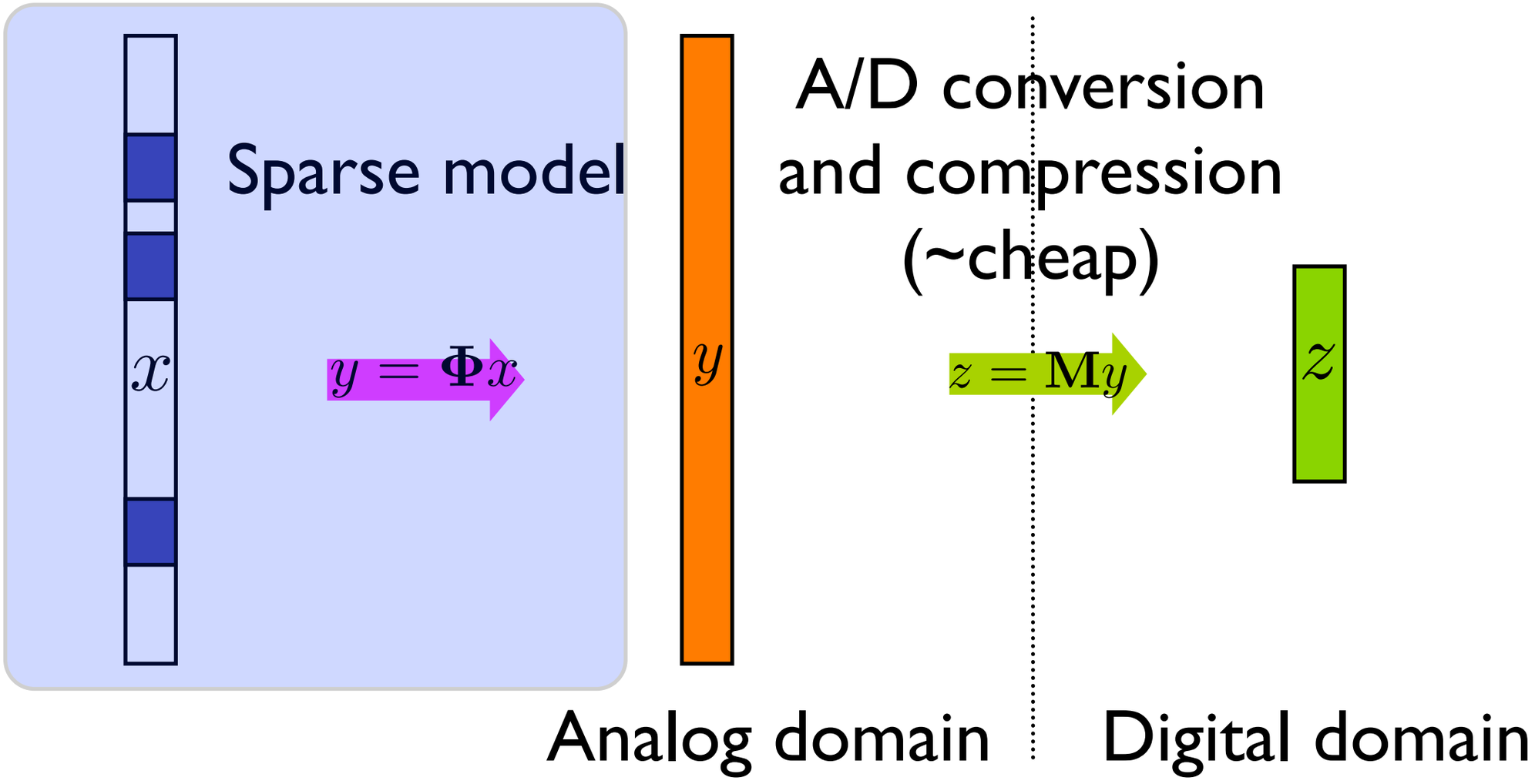
Compressed Sensing

- First model the data, then sample & compress



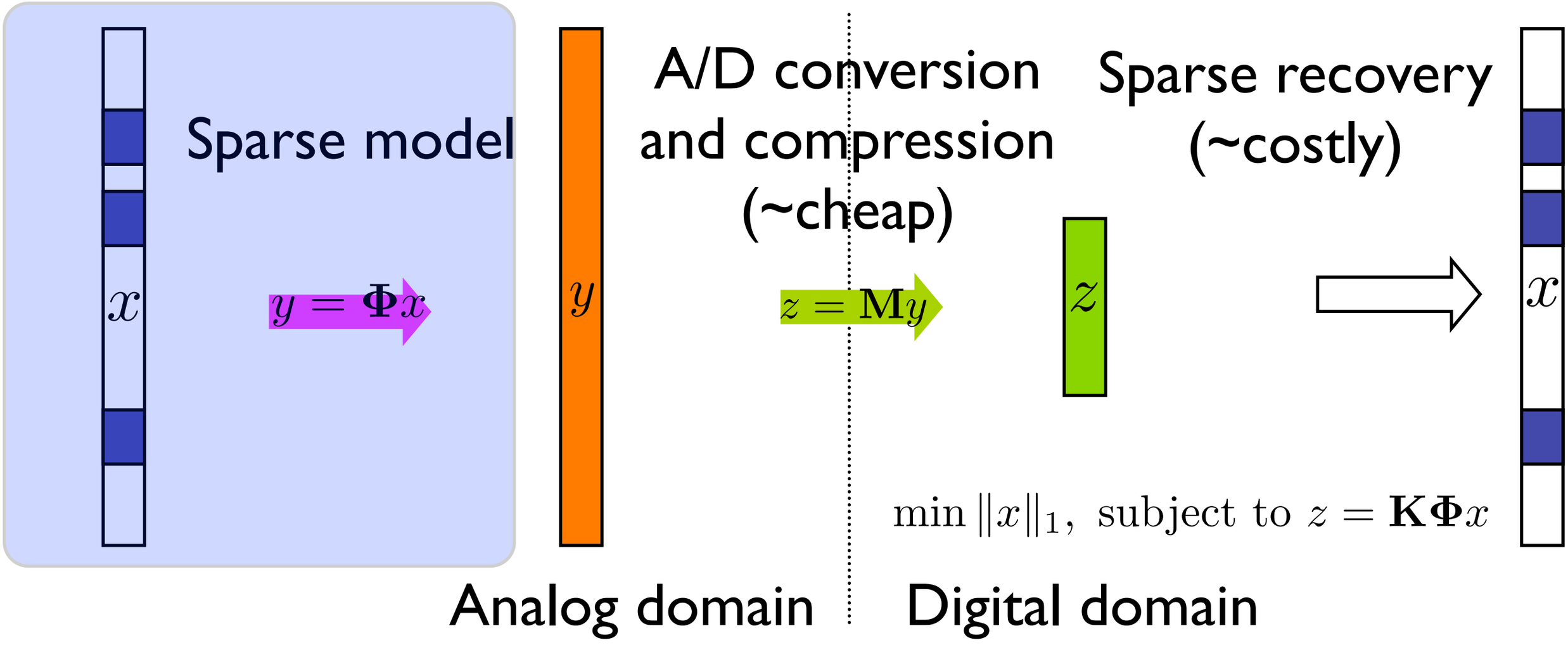
Compressed Sensing

- First model the data, then sample & compress



Compressed Sensing

- First model the data, then sample & compress



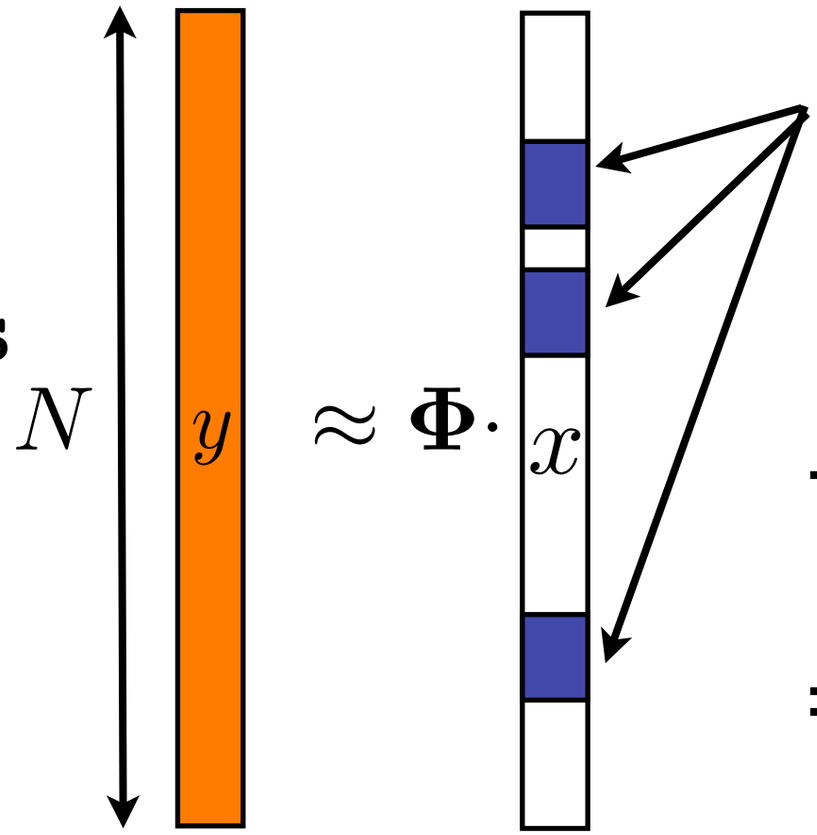
Conditions of success of Compressed Sensing

- Knowledge: transform domain where data is sparse
- «Incoherence» between measurement domain and sparse domain (uncertainty principle *à la* Heisenberg)
  - ✓ time domain / frequency domain
  - ✓ spatial domain / frequency domain
  - ✓ **random measurements!**
- Sufficiently many measures $m \geq Ck \log_2 \frac{N}{k}$
  - ✓ necessary
  - ✓ sufficient (with random Gaussian measures)

Why the log factor ?

- Full vector

N entries
= N `floats`



- Sparse vector

$k \ll N$ nonzero entries
= k `floats`

+ k positions among N

$$= \log_2 \binom{N}{k} \approx k \log_2 \frac{N}{k} \texttt{ bits}$$

Key practical issues: choose dictionary

Summary

Notion of sparsity
(Fourier, wavelets, ...)

Sparsity

Compression
Representation
Description
Classification

Denoising
Blind source separation
Compressed sensing
...

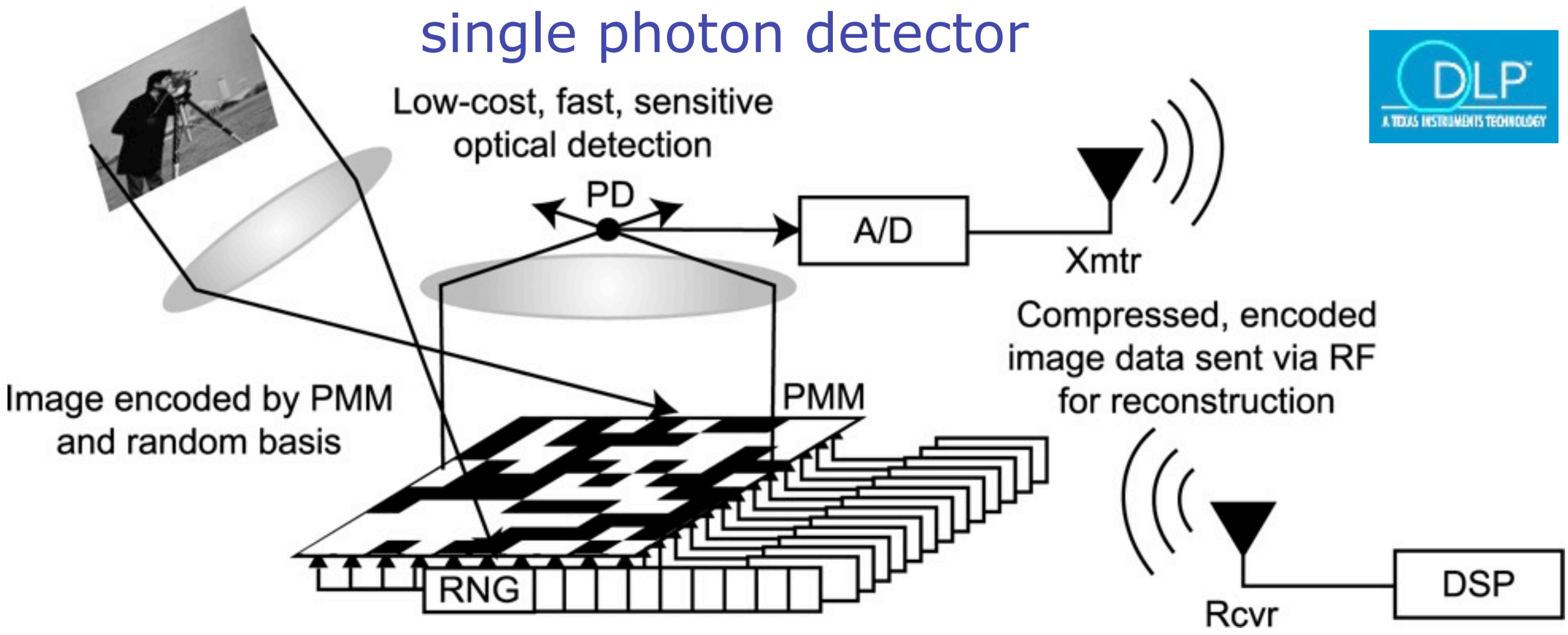
Natural / traditional role

Sparsity = low cost (bits, computations, ...)
Direct objective

Novel indirect role

Sparisty = prior knowledge, regularization
Tool for inverse problems

Example : single-pixel camera, Rice University



Random pattern on DMD array

image reconstruction

INVENTEURS DU MONDE NUMÉRIQUE

Pursuit Algorithms for Sparse Representations

Rémi Gribonval, DR INRIA
EPI METISS (Speech and Audio Processing)
INRIA Rennes - Bretagne Atlantique

remi.gribonval@inria.fr
http://www.irisa.fr/metiss/members/remi/talks

Structure of the course

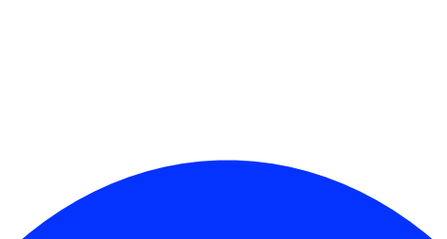
- Session 1: Panorama
  - ✓ sparsity: compression, inverse problems, learning
  - ✓ introduction to compressed (random) sensing
- Session 2: Algorithms
  - ✓ review of main algorithms & complexities
- Session 3: Guarantees for Deterministic *vs* Random dictionaries
  - ✓ compared success guarantees for different algorithms
  - ✓ robust guarantees & Restricted Isometry Property
  - ✓ explicit guarantees for various inverse problems

Summary

Notion of sparsity
(Fourier, wavelets, ...)

Sparsity

Compression
Representation
Description
Classification



Denoising
Blind source separation
Compressed sensing
...

Natural / traditional role

Sparsity = low cost (bits, computations, ...)
Direct objective

Novel indirect role

Sparsity = prior knowledge, regularization
Tool for inverse problems

Overview of Session 2

Convex & nonconvex optimization principles

Convex & nonconvex optimization algorithms

Greedy algorithms

Comparison of complexities

Overall compromise

- Approximation quality

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

- Ideal sparsity measure : ℓ^0 “norm”

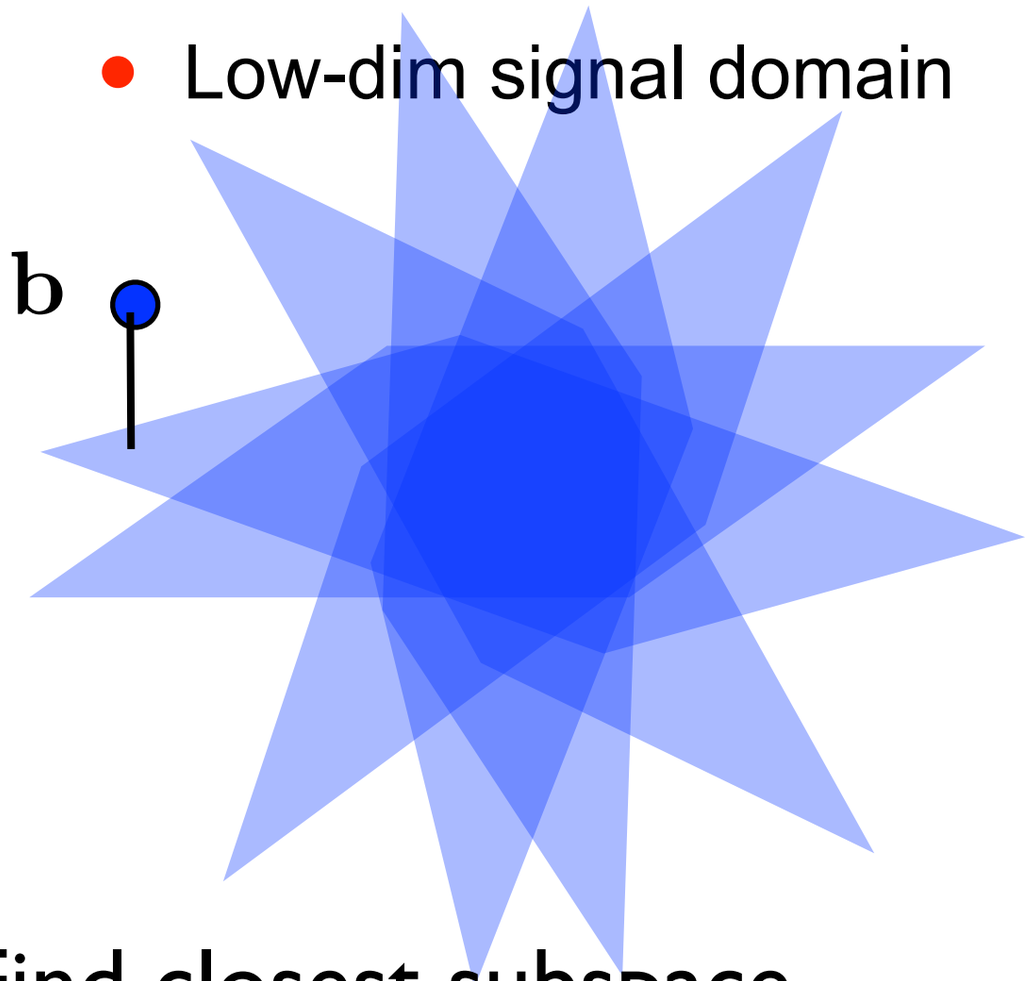
$$\|x\|_0 := \sharp\{n,\ x_n \neq 0\} = \sum_n |x_n|^0$$

- “Relaxed” sparsity measures

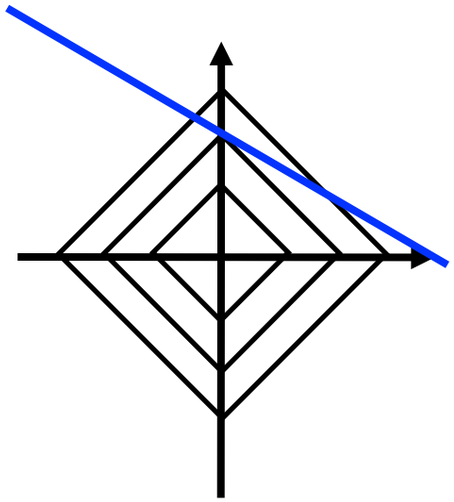
$$0 < p < \infty,\ \|x\|_p := \left(\sum_n |x_n|^p\right)^{1/p}$$

Two geometric viewpoints

- Low-dim signal domain



- High-dim coeff domain



$\{x \text{ s.t.} \mathbf{b} = \mathbf{A}x\}$

Find closest subspace through correlations $\mathbf{A}^T\mathbf{b}$

Find sparsest representation through (convex) optimization

Algorithms for L1: Linear Programming

- L1 minimization problem of size *m* x *N*

Basis Pursuit (BP)
LASSO

$$\min_x \|x\|_1, \ \text{s.t.} \ \mathbf{A}x = \mathbf{b}$$

- Equivalent linear program of size *m* x 2*N*

$$\min_{z \geq 0} \mathbf{c}^T z, \ \text{s.t.} \ [\mathbf{A}, -\mathbf{A}]z = \mathbf{b}$$

$$\mathbf{c} = (c_i), \ c_i = 1, \forall i$$

L1 regularization: Quadratic Programming

- L1 minimization problem of size *m* x *N*

Basis Pursuit Denoising (BPDN)

$$\min_x \frac{1}{2}\|\mathbf{b} - \mathbf{A}x\|_2^2 + \lambda\|x\|_1$$

- Equivalent quadratic program of size *m* x 2*N*

$$\min_{z \geq 0} \frac{1}{2}\|\mathbf{b} - [\mathbf{A}, -\mathbf{A}]z\|_2^2 + \mathbf{c}^T z$$

$$\mathbf{c} = (c_i),\ c_i = 1, \forall i$$

Generic approaches *vs* specific algorithms

- Many algorithms for linear / quadratic programming
- Matlab Optimization Toolbox: `linprog` /`qp`
- But ...
  - ✓ The problem size is “doubled”
  - ✓ Specific structures of the matrix **A** can help solve BP and BPDN more efficiently
  - ✓ More efficient toolboxes have been developed
- CVX package (Michael Grant & Stephen Boyd):
  - ✓ `http://www.stanford.edu/~boyd/cvx/`

Overview

Convex & nonconvex optimization principles

Convex & nonconvex optimization algorithms

Greedy algorithms

Comparison of complexities

What if **A** is orthonormal ?

- Assumption : *m*=*N* and **A** is *orthonormal*

$$\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{Id}_N$$

$$\|\mathbf{b} - \mathbf{A}x\|_2^2 = \|\mathbf{A}^T\mathbf{b} - x\|_2^2$$

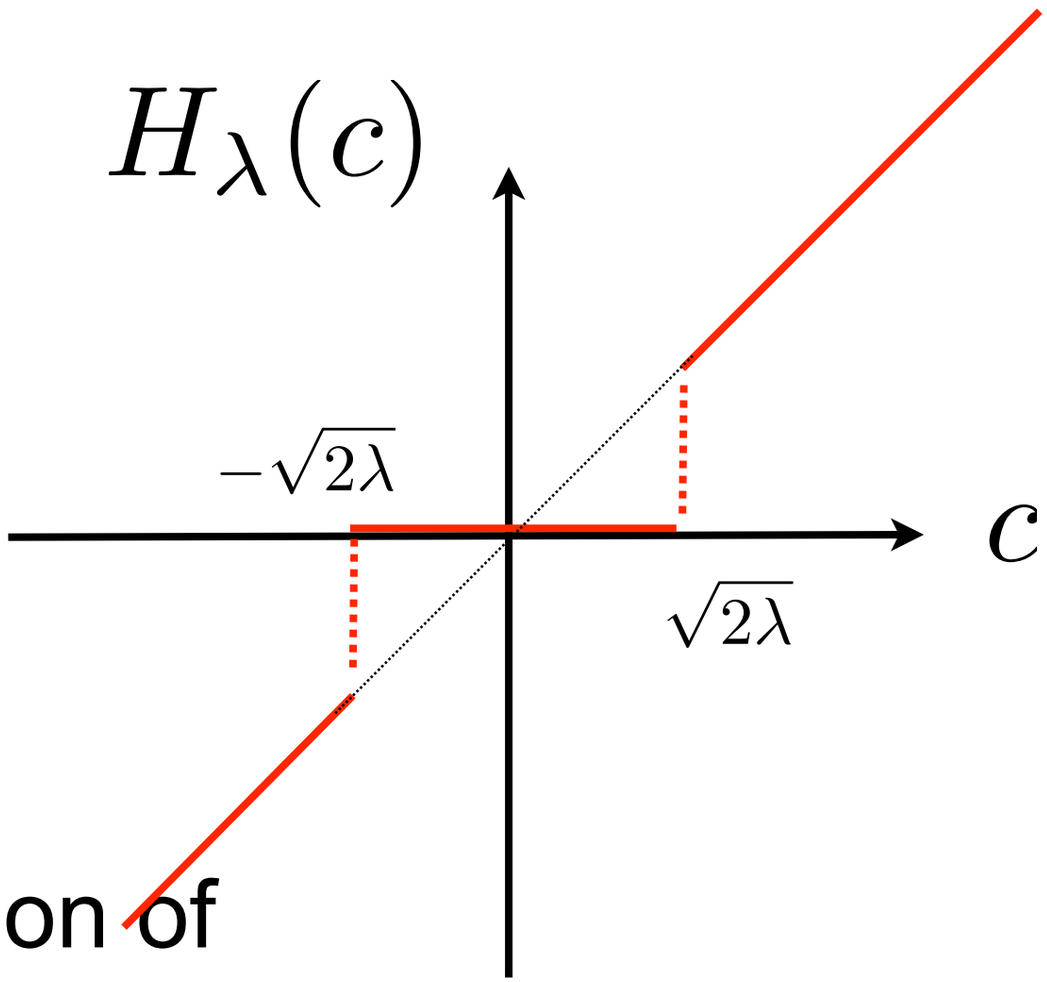
- Expression of BPDN criterion to be minimized

$$\sum_n \frac{1}{2}\left((\mathbf{A}^T\mathbf{b})_n - x_n\right)^2 + \lambda|x_n|^p$$

- Minimization can be done coordinate-wise

$$\min_{x_n} \frac{1}{2}\left(c_n - x_n\right)^2 + \lambda|x_n|^p$$

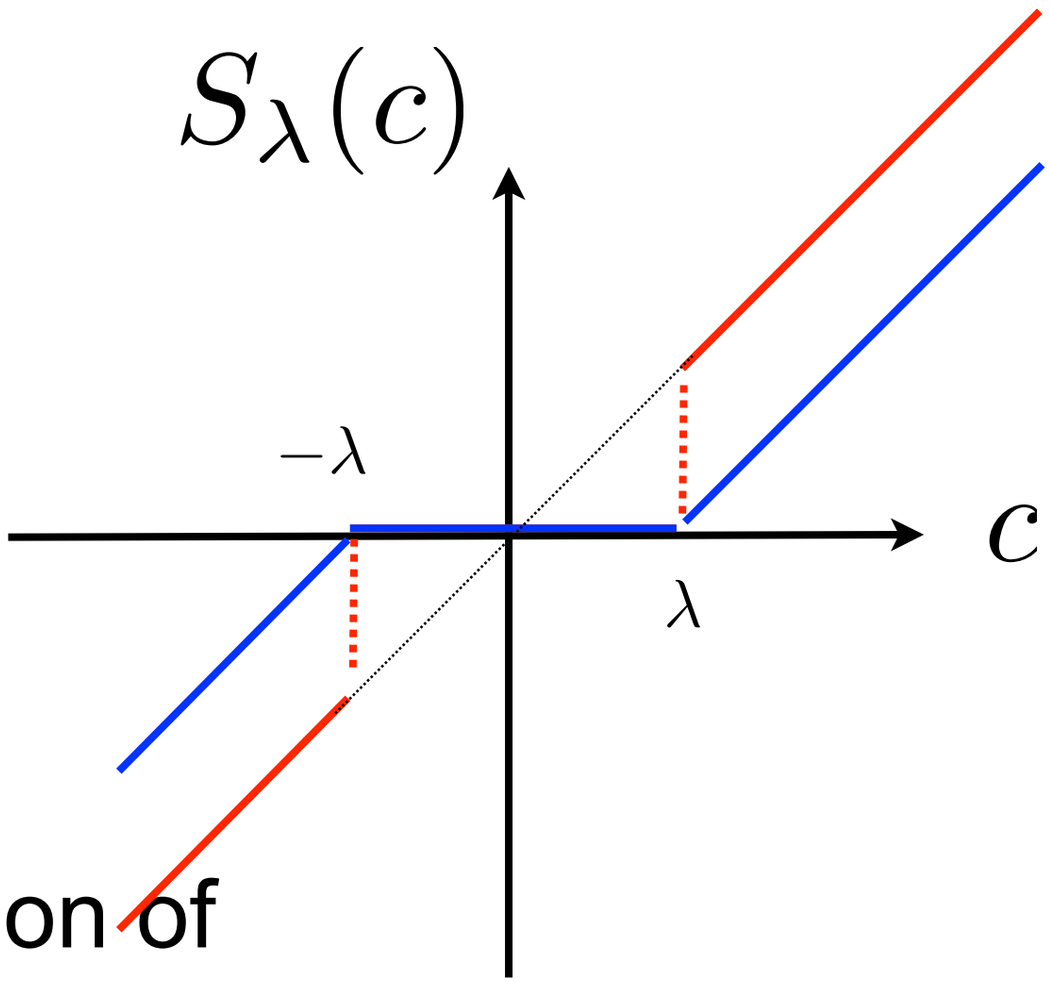
Hard-thresholding (p=0)



- Solution of

$$\min_x \frac{1}{2}(c-x)^2 + \lambda \cdot |x|^0$$

Soft-thresholding (p=1)



- Solution of

$$\min_x \frac{1}{2}(c-x)^2 + \lambda \cdot |x|$$

Iterative thresholding

- Proximity operator

$$\Theta_\lambda^p(c) = \arg\min_x \frac{1}{2}(x-c)^2 + \lambda|x|^p$$

- Goal = compute

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

- Approach = iterative alternation between
  - ✓ gradient descent on fidelity term

$$x^{(i+1/2)} := x^{(i)} + \alpha^{(i)}\mathbf{A}^T(\mathbf{b} - \mathbf{A}x^{(i)})$$

  - ✓ thresholding

$$x^{(i+1)} := \Theta^p_{\lambda^{(i)}}(x^{(i+1/2)})$$

Iterative Thresholding

- **Theorem** : [Daubechies, de Mol, Defrise 2004, Combettes & Pesquet 2008]
  - ✓ consider the iterates $x^{(i+1)} = f(x^{(i)})$ defined by the thresholding function, with $p \geq 1$

$$f(x) = \Theta^p_{\alpha\lambda}(x + \alpha \mathbf{A}^T(\mathbf{b} - \mathbf{A}x))$$

  - ✓ assume that $\forall x,\ \|\mathbf{A}x\|_2^2 \leq c\|x\|_2^2$ and $\alpha < 2/c$
  - ✓ then, the iterates converge strongly to a limit $x^\star$

$$\|x^{(i)} - x^\star\|_2 \rightarrow_{i\rightarrow\infty} 0$$

  - ✓ the limit $x^\star$ is a global minimum of $\frac{1}{2}\|\mathbf{A}x - \mathbf{b}\|_2^2 + \lambda\|x\|_p^p$
  - ✓ if *p*>1, or if **A** is invertible, $x^\star$ is the *unique* minimum

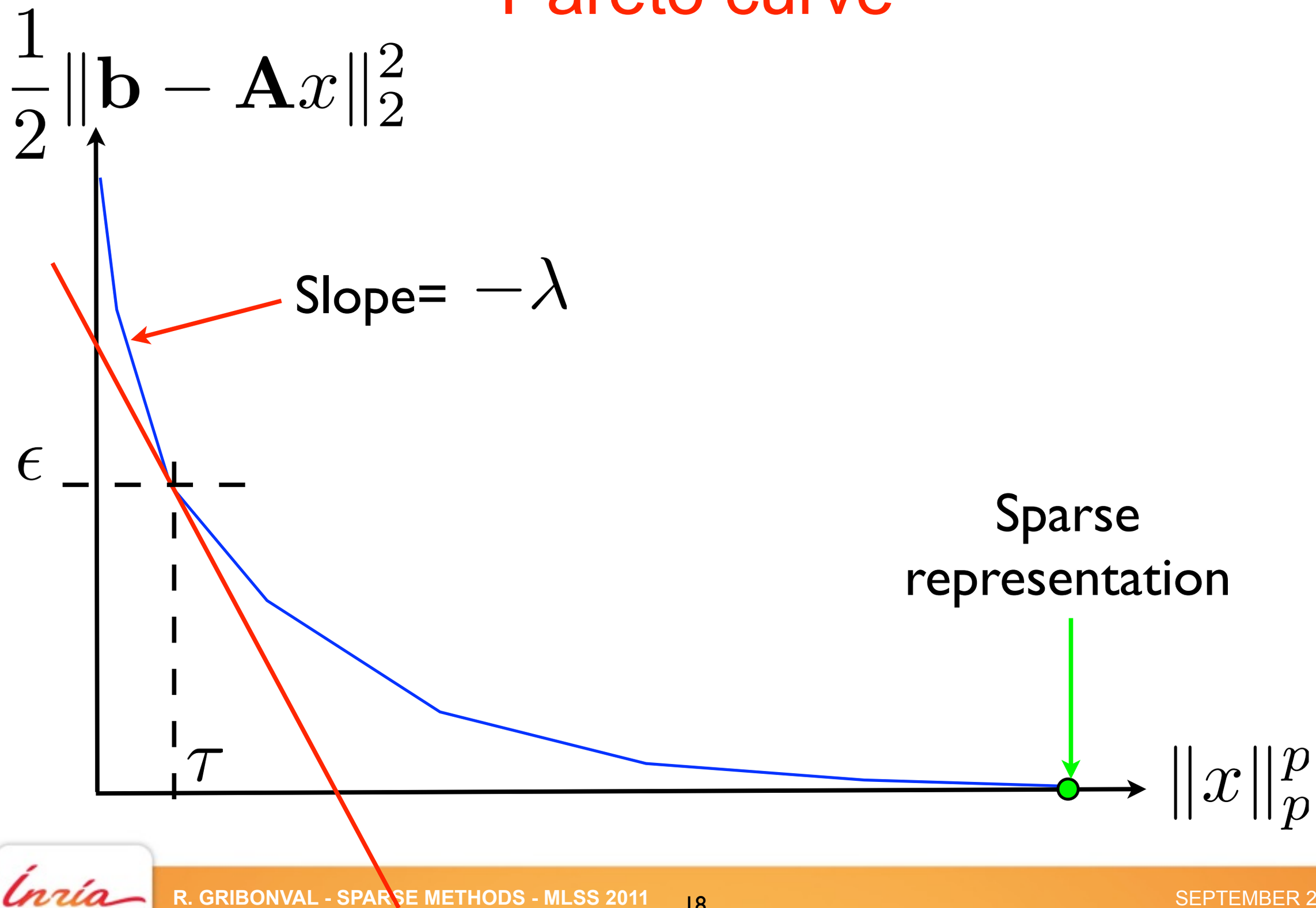
Iterative Thresholding: convex penalties

- Strong convergence to global minimum
- Accelerated convergence: Nesterov schemes
  - ✓ see e.g. Beck & Teboulle 2009;
- Many variants of iterative thresholding
  - ✓ depends on properties of penalty terms
    - ✦ smoothness
    - ✦ strong convexity
    - ✦ etc.
  - ✓ see course by L. Vandenberghe

Iterative Thresholding: nonconvex penalties

- Example: Iterative Hard Thresholding for L0
  - ✓ keep components above threshold
  - ✓ *or rather* **keep *k* largest components**
    - ✦ [IHT: Blumensath & Davies 2009]
- More generally, with *nonconvex* cost functions
  - ✓ Possible ‘spurious’ local minima
  - ✓ Convergence: fixed point, under certain assumptions
  - ✓ Limit = global min: under certain assumptions (RIP)
- Pruning strategies:
  - ✓ ex: keep 2k components, project, keep k components
    - ✦ ex: CoSAMP [Needell &Tropp 2008], ALPS [Cevher 2011], ...

Pareto curve



Path of the solution

- **Lemma**: let $x^\star$ be a local minimum of BPDN

[Fuchs 97]

$$\arg\min_x \frac{1}{2}\|\mathbf{A}x - \mathbf{b}\|_2^2 + \lambda\|x\|_1$$

- let *I* be its support
- Then $$\mathbf{A}_I^T(\mathbf{A}x^\star - \mathbf{b}) + \lambda \cdot \mathrm{sign}(x_I^\star) = 0$$
$$\|\mathbf{A}_{I^c}^T(\mathbf{A}x^\star - \mathbf{b})\|_\infty < \lambda$$
- In particular
$$x_I = (\mathbf{A}_I^T\mathbf{A}_I)^{-1}\left(\mathbf{A}_I^T\mathbf{b} - \lambda \cdot \mathrm{sign}(x_I)\right)$$

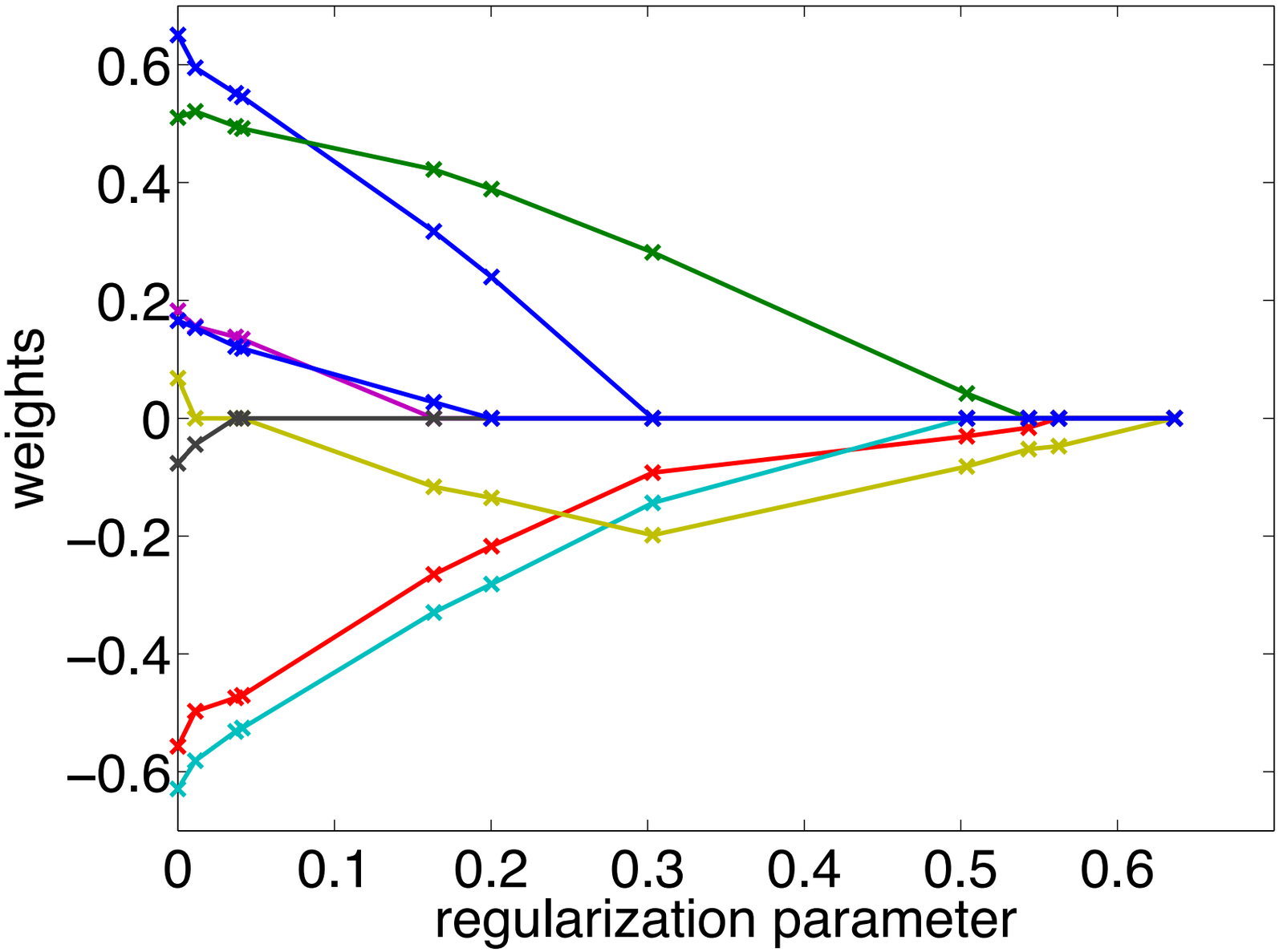
Homotopy method

- **Principle**: track the solution $x^\star(\lambda)$ of BPDN along the Pareto curve
- **Property**: [Fuchs 97, 05; Osborne 2000]
  - ✓ solution is characterized by its sign pattern through

$$x_I = (\mathbf{A}_I^T \mathbf{A}_I)^{-1} \left(\mathbf{A}_I^T \mathbf{b} - \lambda \cdot \mathrm{sign}(x_I) \right)$$

  - ✓ for given sign pattern, dependence on λ is affine
  - ✓ sign patterns are piecewise constant functions of λ
  - ✓ overall, the solution is piecewise affine
- **Method** = iteratively find *breakpoints*
  - ✓ [Osborne 2000; Efron & al 2004]

Piecewise Linear Path



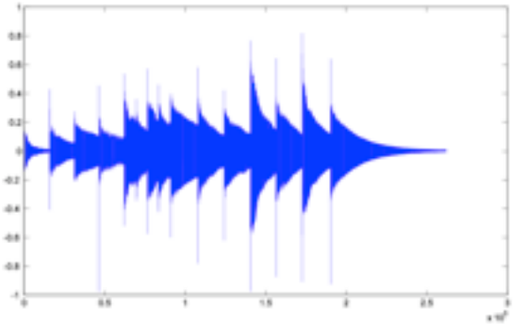
Courtesy: F. Bach

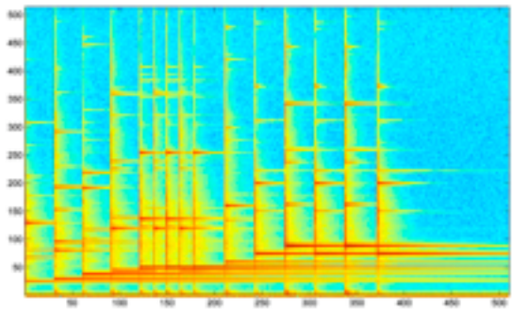
Overview

Convex & nonconvex optimization principles
Convex & nonconvex optimization algorithms
Greedy algorithms
Comparison of complexities

Matching Pursuit with Time-Frequency Atoms

- Audio = superimposition of structures
- Example : glockenspiel



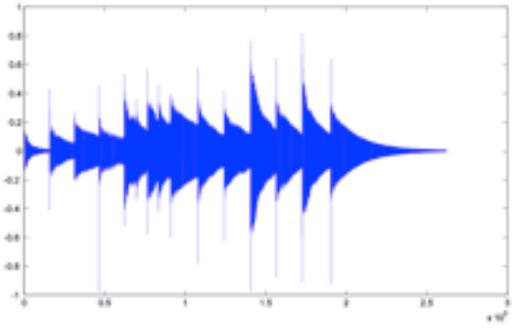


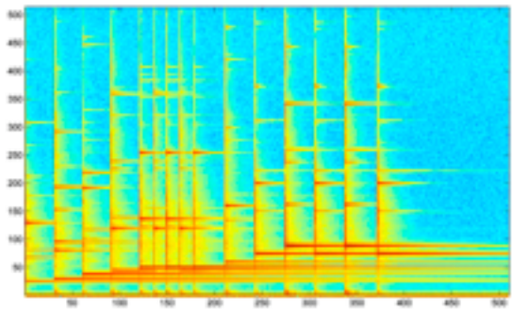
- ✓ transients = short, small scale
- ✓ harmonic part = long, large scale

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

Matching Pursuit with Time-Frequency Atoms

- Audio = superimposition of structures
- Example : glockenspiel



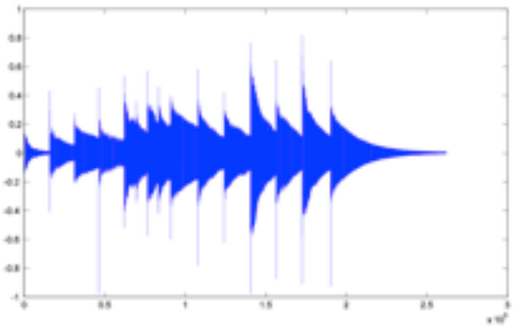


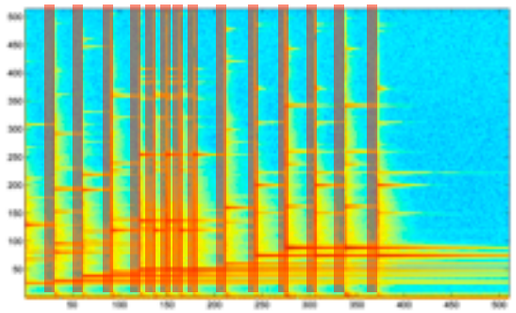
- ✓ transients = short, small scale
- ✓ harmonic part = long, large scale

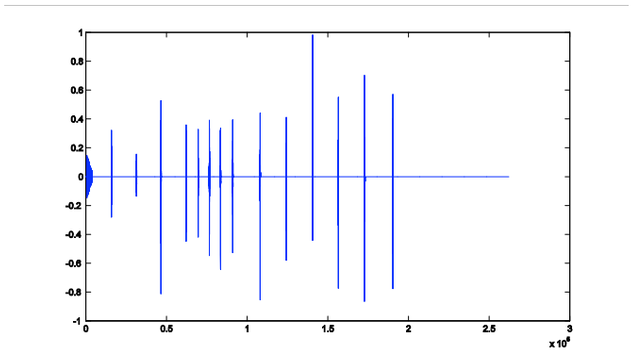
- Gabor atoms $\left\{ g_{s,\tau,f}(t) = \frac{1}{\sqrt{s}} w\left(\frac{t-\tau}{s}\right) \; e^{2i\pi f t} \right\}_{s,\tau,f}$

Matching Pursuit with Time-Frequency Atoms

- Audio = superimposition of structures
- Example : glockenspiel





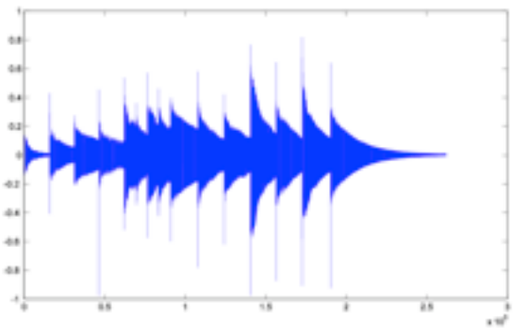


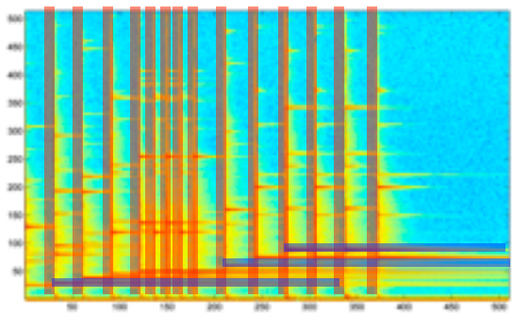
- ✓ transients = short, small scale
- ✓ harmonic part = long, large scale

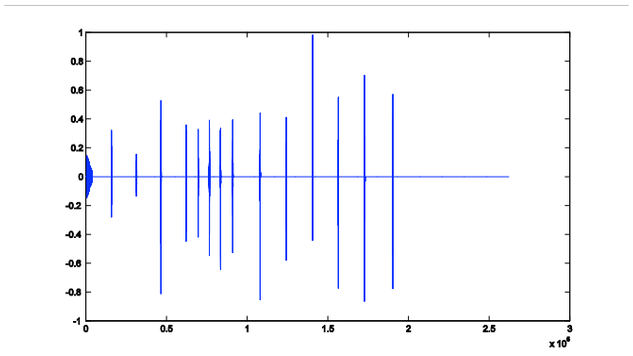
- Gabor atoms $\left\{ g_{s,\tau,f}(t) = \frac{1}{\sqrt{s}} w\left(\frac{t-\tau}{s}\right) \; e^{2i\pi f t} \right\}_{s,\tau,f}$

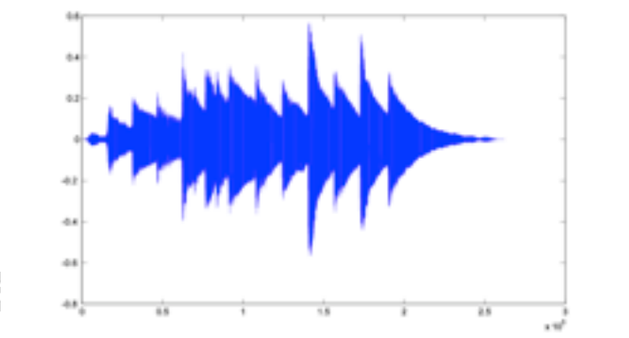
Matching Pursuit with Time-Frequency Atoms

- Audio = superimposition of structures
- Example : glockenspiel









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

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

Matching Pursuit (MP)

[Friedman & Stuetzle 81; Mallat & Zhang 93]

- Matching Pursuit (*aka* Projection Pursuit, CLEAN)
  - ✓ Initialization $\mathbf{r}_0 = \mathbf{b} \qquad i = 1$
  - ✓ Atom selection: (assuming normed atoms: $\|\mathbf{A}_n\|_2 = 1$)

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

  - ✓ Residual update

$$\mathbf{r}_i = \mathbf{r}_{i-1} - (\mathbf{A}_{n_i}^T \mathbf{r}_{i-1}) \mathbf{A}_{n_i}$$

- Energy preservation (Pythagoras theorem)

$$\|\mathbf{r}_{i-1}\|_2^2 = |\mathbf{A}_{n_i}^T \mathbf{r}_{i-1}|^2 + \|\mathbf{r}_i\|_2^2$$

Main properties

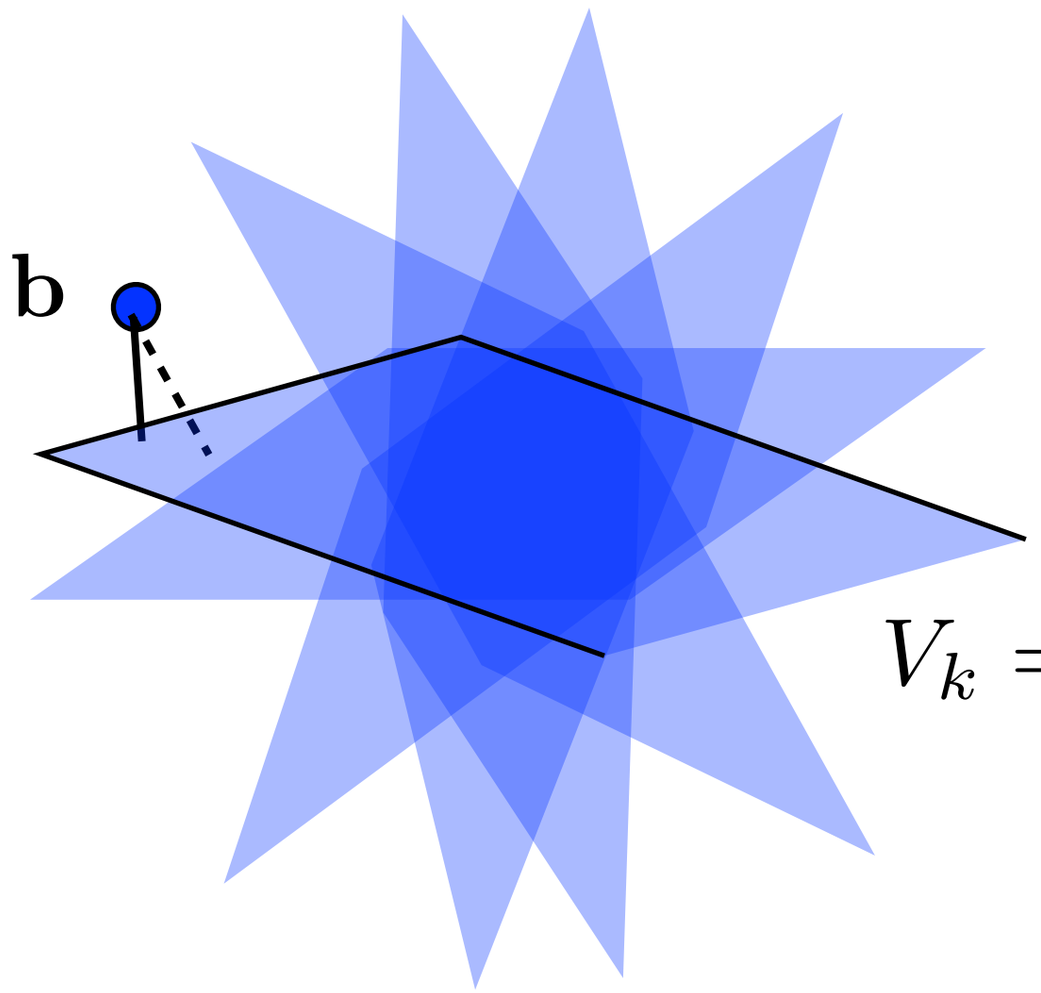
- Global energy preservation

$$\|\mathbf{b}\|_2^2 = \|\mathbf{r}_0\|_2^2 = \sum_{i=1}^{k} |\mathbf{A}_{n_i}^T \mathbf{r}_{i-1}|^2 + \|\mathbf{r}_k\|_2^2$$

- Global reconstruction

$$\mathbf{b} = \mathbf{r}_0 = \sum_{i=1}^{k} (\mathbf{A}_{n_i}^T \mathbf{r}_{i-1}) \mathbf{A}_{n_i} + \mathbf{r}_k$$

- Strong convergence (assuming full-rank dictionary) $\lim_{i \to \infty} \|\mathbf{r}_i\|_2 = 0$



$$V_k = \mathrm{span}(\mathbf{A}_n, n \in \Lambda_k)$$

Orthonormal MP (OMP)

[Mallat & Zhang 93, Pati & al 94]

- Observation: after k iterations $\mathbf{r}_k = \mathbf{b} - \sum_{i=1}^{k} \alpha_k \mathbf{A}_{n_i}$
- Approximant belongs to

$$V_k = \mathrm{span}(\mathbf{A}_n, n \in \Lambda_k)$$
$$\Lambda_k = \{n_i, 1 \leq i \leq k\}$$

- Best approximation from V_k = orthoprojection

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

- **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*

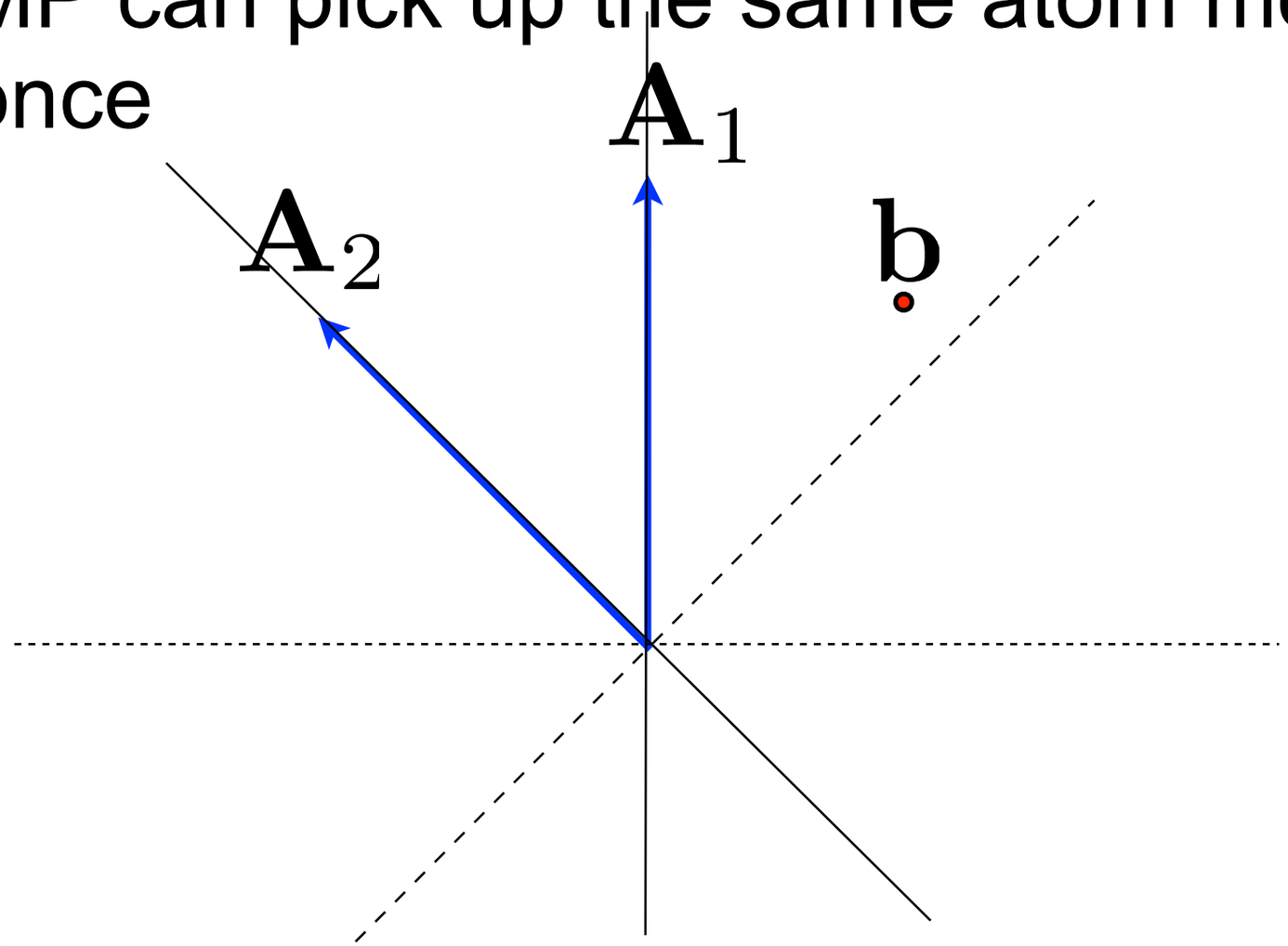
$$V_i = \mathrm{span}(\mathbf{A}_n, n \in \Lambda_i)$$

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

- Property : strong convergence $\lim_{i \to \infty} \|\mathbf{r}_i\|_2 = 0$

Caveats (1)

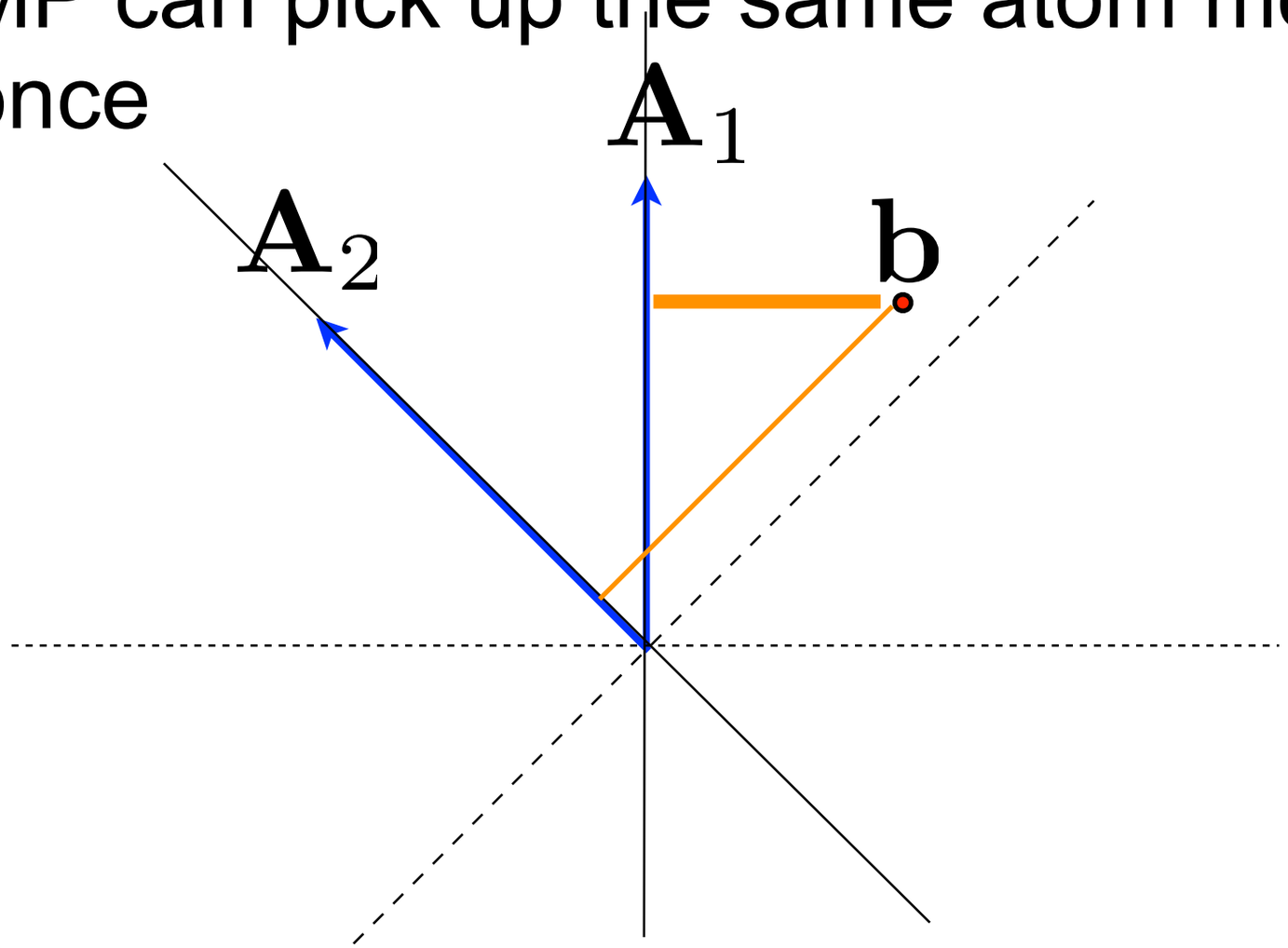
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

Caveats (1)

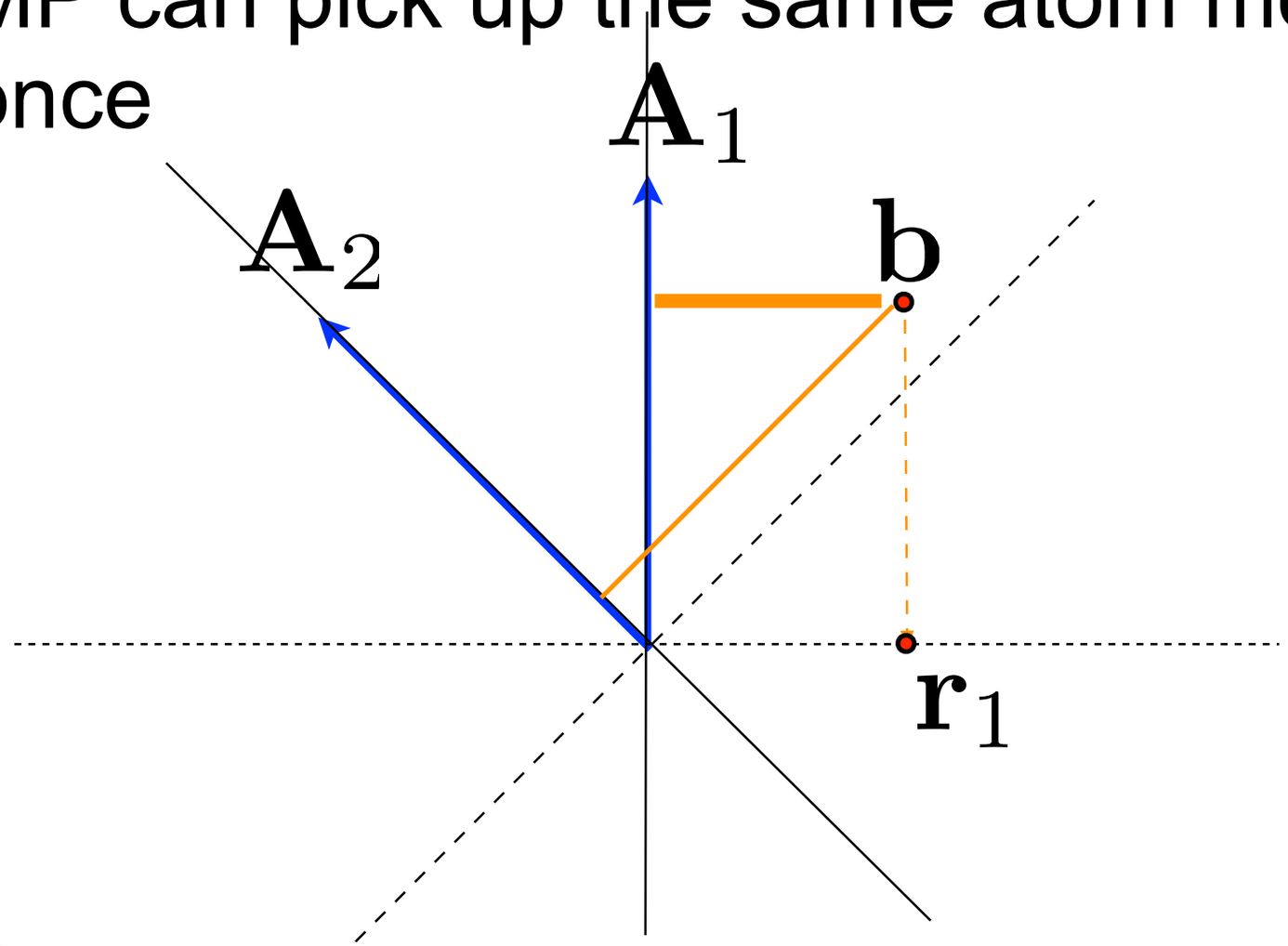
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

Caveats (1)

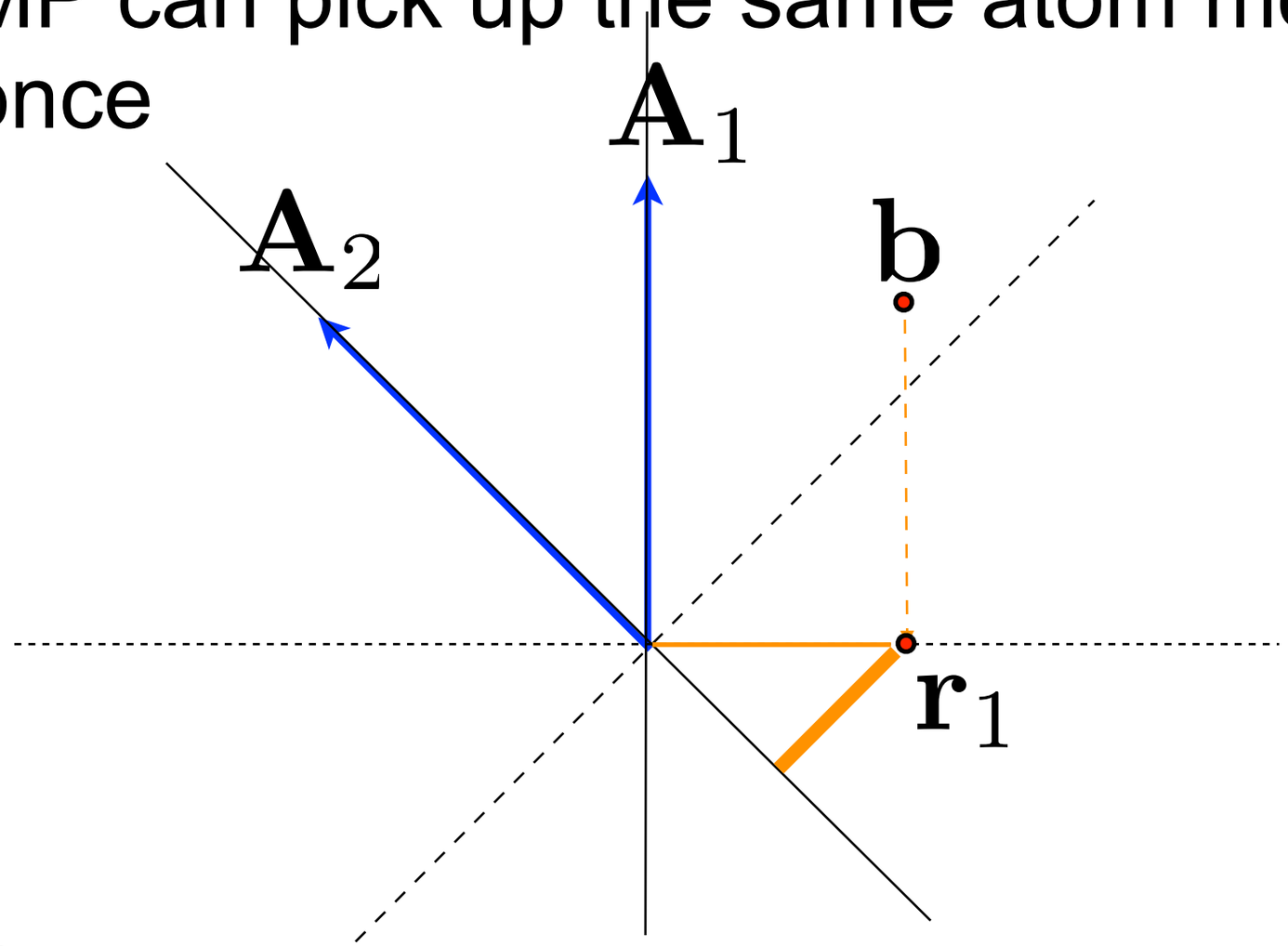
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

Caveats (1)

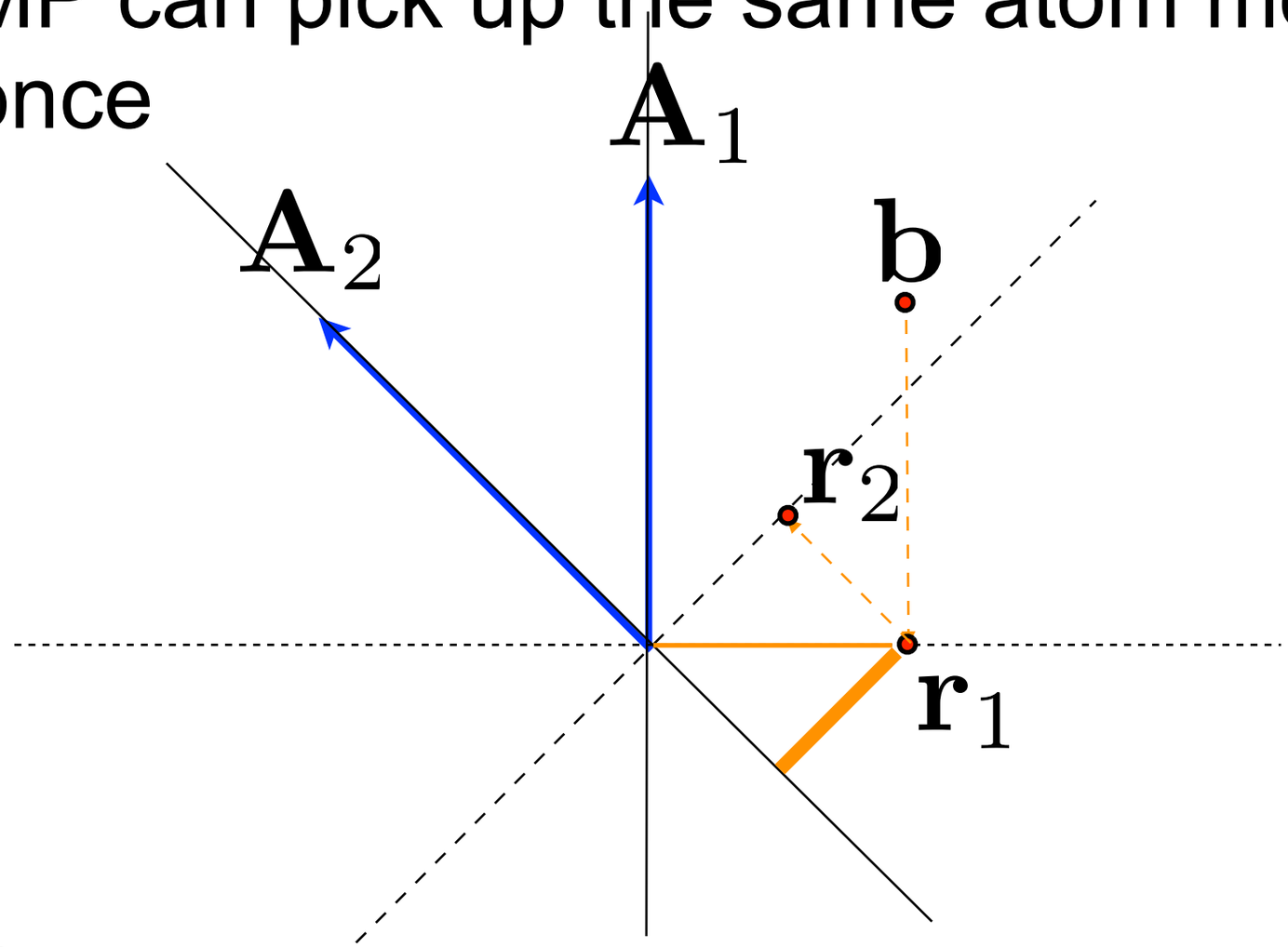
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

Caveats (1)

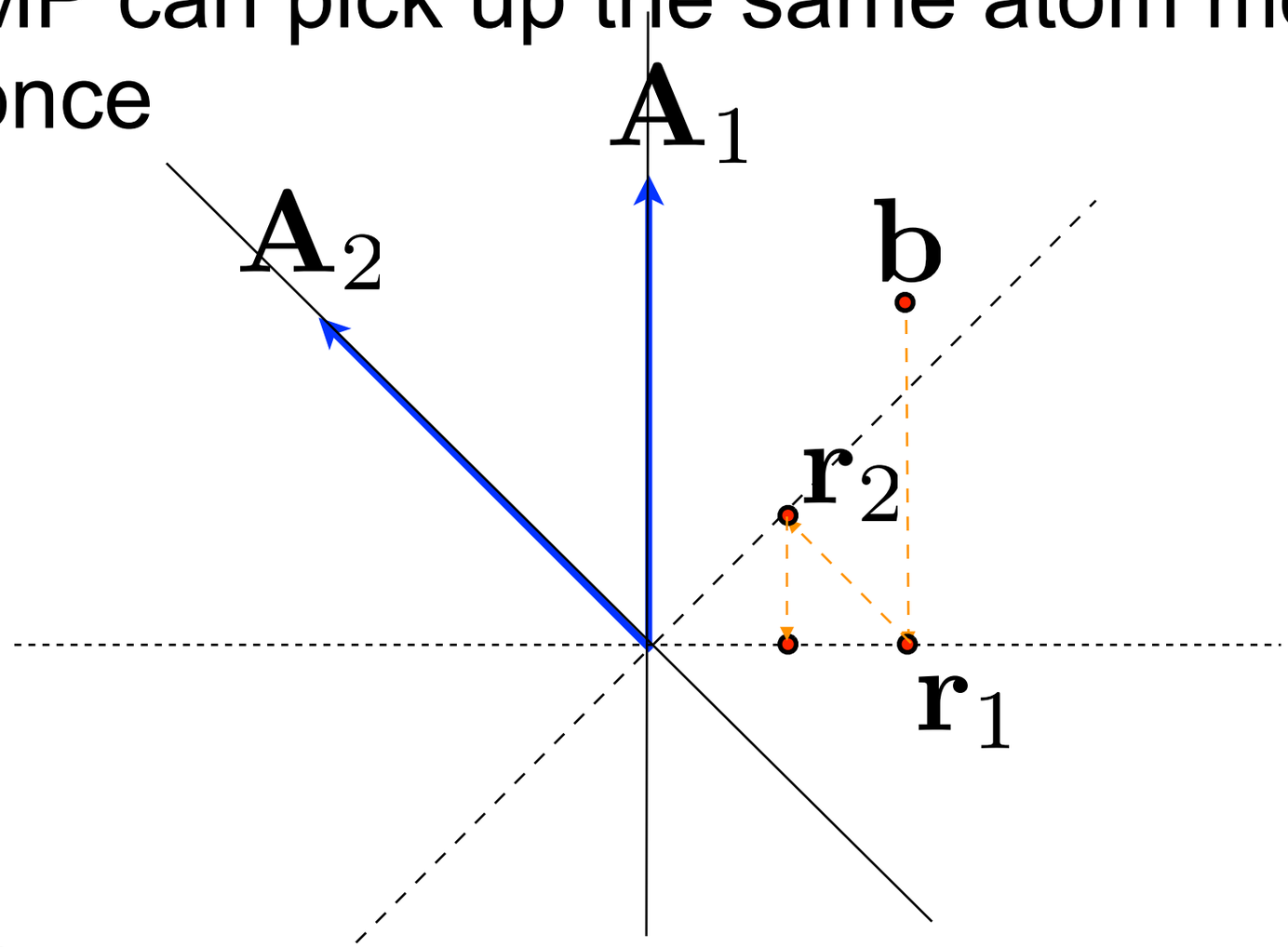
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

Caveats (1)

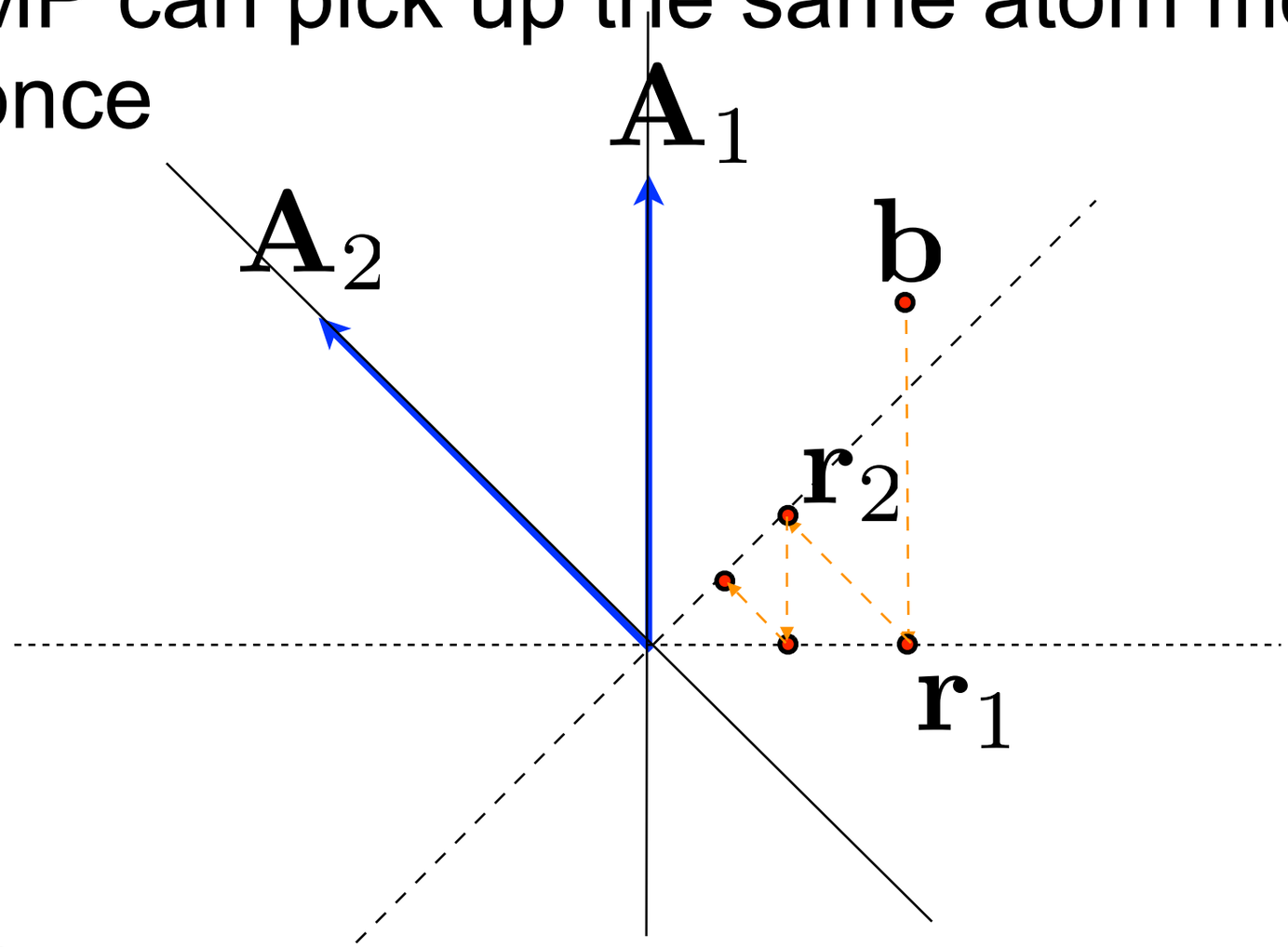
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

Caveats (1)

- MP can pick up the same atom more than once



- OMP will never select twice the same atom

Caveats (2)

- "Improved" atom selection does not necessarily improve convergence
- There exists two dictionaries **A** and **B**
  - ✓ Best atom from **B** at step i:

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

  - ✓ Better atom from **A**

$$|\mathbf{A}_{\ell_i}^T \mathbf{r}_{i-1}| \geq |\mathbf{B}_n^T \mathbf{r}_{i-1}|$$

  - ✓ Residual update

$$\mathbf{r}_i = \mathbf{r}_{i-1} - (\mathbf{A}_{\ell_i}^T \mathbf{r}_{i-1}) \mathbf{A}_{\ell_i}$$

- Divergence! $\exists c > 0, \forall i, \|\mathbf{r}_i\|_2 \geq c$

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

$$\mathbf{b} = \mathbf{A}x_i + \mathbf{r}_i \qquad \mathbf{A} = [\mathbf{A}_1, \dots \mathbf{A}_N]$$

	Matching Pursuit	OMP	Stagewise
Selection	$\Gamma_i := \arg\max_n \lvert\mathbf{A}_n^T \mathbf{r}_{i-1}\rvert$		$\Gamma_i := \{n \mid \lvert\mathbf{A}_n^T \mathbf{r}_{i-1}\rvert > \theta_i\}$
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 $\mathbf{r}_i = \mathbf{b} - \mathbf{A}x_i$ • select new components • update residual
Tuning quality/sparsity	regularization parameter λ	stopping criterion (nb of iterations, error level, ...) $\|x_i\|_0 \geq k$ $\|\mathbf{r}_i\| \leq \epsilon$
Variants	• choice of sparsity measure p • optimization algorithm • initialization	•selection criterion (weak, stagewise ...) •update strategy (orthogonal ...)

Overview

Convex & nonconvex optimization principles
Convex & nonconvex optimization algorithms
Greedy algorithms
Comparison of complexities

Complexity of IST

- Notation: $O(\mathbf{A})$ cost of applying $\mathbf{A}$ or $\mathbf{A}^T$
- Iterative Thresholding $f(x) = \Theta^p_{\alpha\lambda}(x + \alpha\mathbf{A}^T(\mathbf{b} - \mathbf{A}x))$
  - ✓ cost per iteration = $O(\mathbf{A})$
  - ✓ when **A** invertible, linear convergence at rate

$$\|x^{(i)} - x^\star\|_2 \lesssim C\beta^i \|x^\star\|_2 \qquad \beta \leq 1 - \frac{\sigma^2_{\min}}{\sigma^2_{\max}}$$

  - ✓ number of iterations guaranteed to approach limit within relative precision ϵ

$$O(\log 1/\epsilon)$$

- Limit depends on choice of penalty factor λ, added complexity to adjust it

Complexity of MP

- Number of iterations depends on stopping criterion

$$\|\mathbf{r}_i\|_2 \leq \epsilon, \|x_i\|_0 \geq k$$

- Cost of first iteration = atom selection $O(\mathbf{A})$ (computation of all inner products)
- Naive cost of subsequent iterations = $O(\mathbf{A})$
- If "local" structure of dictionary *[Krstulovic & al, MPTK]*
  - ✓ subsequent iterations only cost $O(\log N)$

	Generic **A**	Local **A**
k iterations	$O(k\mathbf{A}) \geq O(km)$	$O(\mathbf{A} + k \log N)$
$k \propto m$	$O(m^2)$	$O(m \log N)$

Complexity of OMP

- Number of iterations depends on stopping criterion

$$\|\mathbf{r}_i\|_2 \leq \epsilon, \|x_i\|_0 \geq k$$

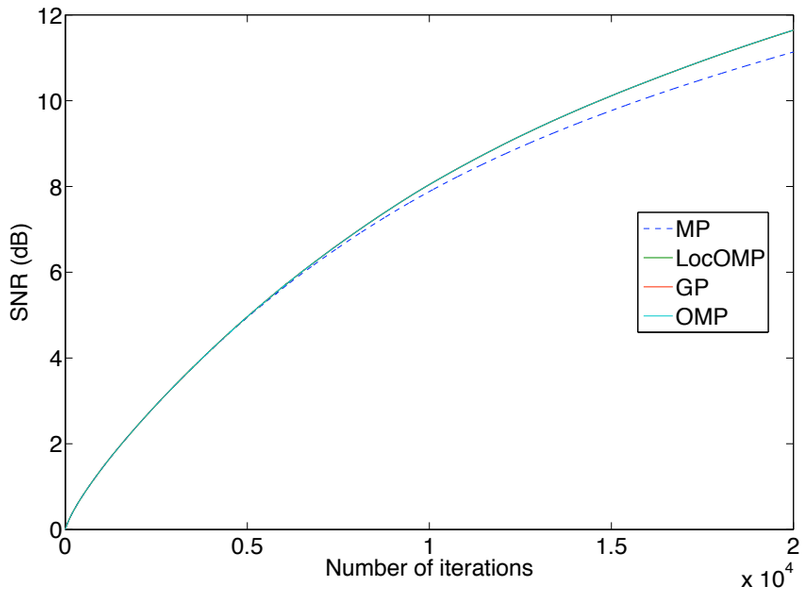
- Naive cost of iteration *i*
  - ✦ atom selection $O(\mathbf{A})$ + orthoprojection $O(i^3)$
- With iterative matrix inversion lemma
  - ✦ atom selection $O(\mathbf{A})$ + coefficient update $O(i^2)$
- If "local" structure of dictionary *[Mailhé & al, LocOMP]*
  - ✓ subsequent approximate iterations only cost $O(\log N)$

	Generic **A**	Local **A**
k iterations	$O(k\mathbf{A} + k^3)$	$O(\mathbf{A} + k \log N)$
$k \propto m$	$O(m^3)$	$O(m \log N)$

LoCOMP

- A variant of OMP for shift invariant dictionaries
  (Ph.D. thesis of Boris Mailhé, ICASSP09)

Fig. 1. SNR depending on the number of iterations



$N = 5.10^5$ samples, k= 20 000 iterations

Table 3. CPU time per iteration (s)

Iteration	MP	LocOMP	GP	OMP
First ($i = 0$)	3.4	3.4	3.4	3.5
Begin ($i \approx 1$)	0.028	0.033	3.4	3.4
End ($i \approx I$)	0.028	0.050	40.5	41
Total time	571	854	$4.50 \cdot 10^5$	$4.52 \cdot 10^5$

- Implementation in MPTK in progress for larger scale experiments

Software ?

- Matlab (simple to adapt, medium scale problems):
  - ✦ **Thousands** of unknowns, few seconds of computations
  - ✦ L1 minimization with an available toolbox
    - ➡ http://www.l1-magic.org/ (Candès, Romberg et al.), CVX, ...
  - ✦ Iterative thresholding
    - ➡ http://www.morphologicaldiversity.org/ (Starck et al.), FISTA, NESTA, ...
  - ✦ Matching Pursuits
    - ➡ sparsify (Blumensath), GPSR, ...
- SMALLbox (): unified API for Matlab toolboxes
  - ➡ http://small-project.eu/software-data/smallbox/
- MPTK : C++, large scale problems
  - ✦ **Millions** of unknowns, few minutes of computation
  - ✦ specialized for local + shift-invariant dictionaries
  - ✦ built-in multichannel
    - ➡ http://mptk.irisa.fr

Sparse Models, Algorithms and Learning for Large-scale data SMALL

Performance of Sparse Decomposition Algorithms with Deterministic versus Random Dictionaries

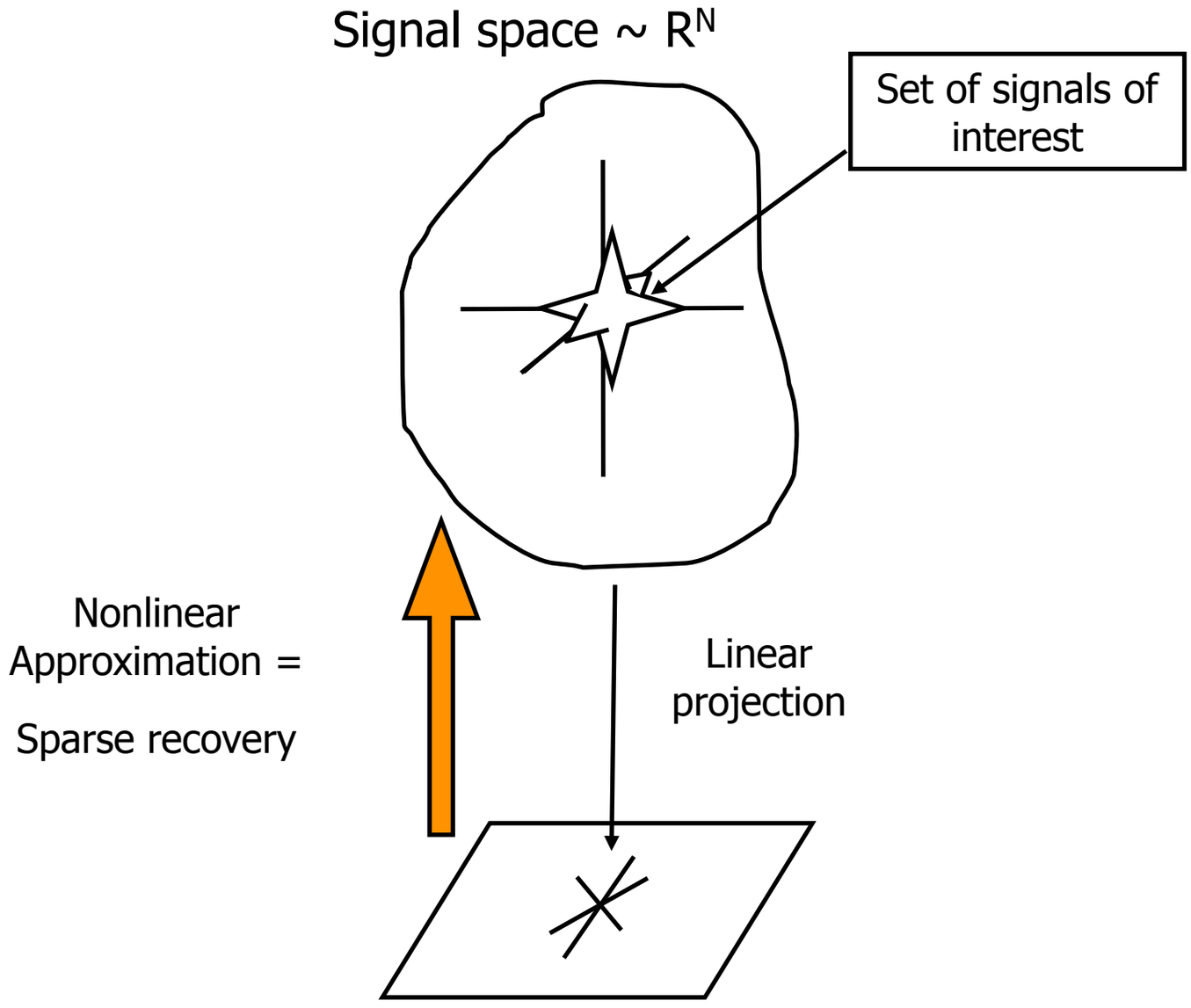
Rémi Gribonval, DR INRIA
EPI METISS (Speech and Audio Processing)
INRIA Rennes - Bretagne Atlantique

remi.gribonval@inria.fr
http://www.irisa.fr/metiss/members/remi/talks

Structure of the course

- Session 1: Panorama
  - ✓ role of sparsity for compression,inverse problems, and learning
  - ✓ introduction to compressed (random) sensing
- Session 2: Algorithms
  - ✓ Review of main algorithms & complexities
- Session 3: Guarantees for Deterministic & Random dictionaries
  - ✓ compared success guarantees for different algorithms
  - ✓ robust guarantees & Restricted Isometry Property
  - ✓ explicit guarantees for various inverse problems

Inverse problems



Courtesy: M. Davies, U. Edinburgh

Exact recovery conditions for Lp

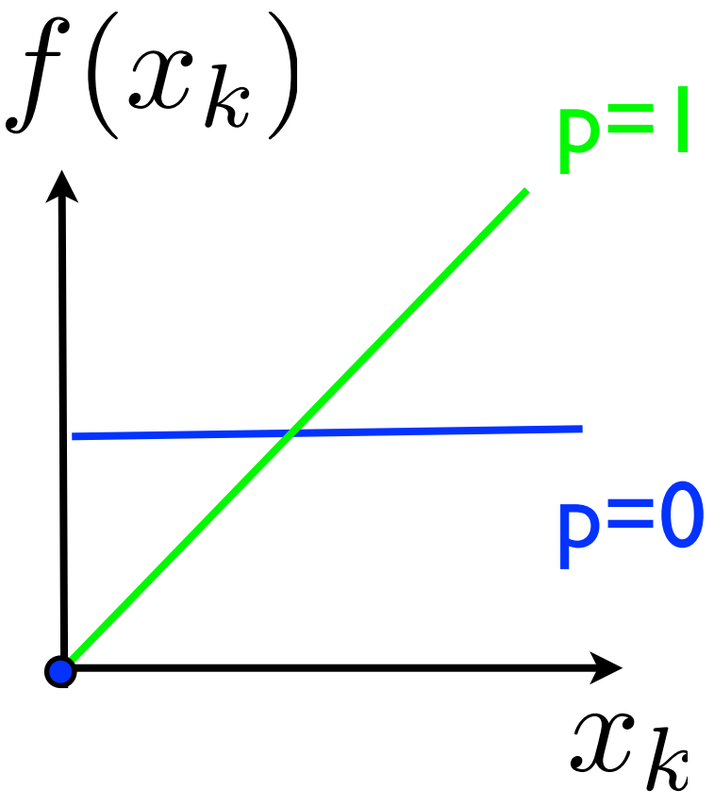
Proved Equivalence between L0 and L1

- "Empty" 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 $\|x_0\|_0 \leq k_1(\mathbf{A})$ $x_0 = x_1^\star$

- Content = estimation of $k_0(\mathbf{A})$ and $k_1(\mathbf{A})$
  - ✓ *Donoho & Huo 2001 : pair of bases, coherence*
  - ✓ *Donoho & Elad 2003, Gribonval & Nielsen 2003 : dictionary, coherence*
  - ✓ *Candes, Romberg, Tao 2004 : random dictionaries, restricted isometry constants*
  - ✓ *Tropp 2004 : idem for Orthonormal Matching Pursuit, cumulative coherence*
- What about $x_p^\star, 0 \leq p \leq 1$?

General sparsity measures

- Lp-norms $\|x\|_p^p := \sum_k |x_k|^p, 0 \leq p \leq 1$
- f-norms! $\|x\|_f := \sum_k f(|x_k|)$
- Constrained minimization



$$x_f^\star = x_f^\star(\mathbf{b}, \mathbf{A}) \in \arg\min_x \|x\|_f \quad \text{subject to} \quad \mathbf{b} = \mathbf{A}x$$

When do we have $x_f^\star(\mathbf{A}x_0, \mathbf{A}) = x_0$?

Null space

- Null space = kernel

$$z \in \mathcal{N}(\mathbf{A}) \Leftrightarrow \mathbf{A}z = 0$$

- Particular solution *vs* general solution
  - ✓ particular solution

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

  - ✓ general solution

$$\mathbf{A}x' = \mathbf{b} \Leftrightarrow x' - x \in \mathcal{N}(\mathbf{A})$$

Recoverable supports : the “Null Space Property” (1)

- **Theorem 1** [*Donoho & Huo 2001 for L1, G. & Nielsen 2003 for Lp & more*]
  - ✓ Assumption 1: sub-additivity (for quasi-triangle inequality)

$$f(a+b) \leq f(a) + f(b), \forall a, b$$

  - ✓ Assumption 2: «Null Space Property»

NSP

$$\|z_I\|_f < \|z_{I^c}\|_f \text{ when } z \in \mathcal{N}(\mathbf{A}), z \neq 0$$

  - ✓ Conclusion: $x_f^\star$ recovers every x **supported in** I
  - ✓ The result is sharp: if NSP fails on support *I* there is **at least one failing vector** x supported in *I*

NSP is necessary

- Notations
  - ✓ index set *I*
  - ✓ vector *z*
  - ✓ restriction

$$z_I = (z_i)_{i \in I}$$

- Assume there exists $z \in \mathcal{N}(\mathbf{A})$ with

$$\|z_I\|_f > \|z_{I^c}\|_f$$

- Define $\mathbf{b} := Az_I = A(-z_{I^c})$
- The vector z_I is supported in *I* but is ***not*** the minimum norm representation of $\mathbf{b}$

NSP is sufficient

- Assume quasi-triangle inequality

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

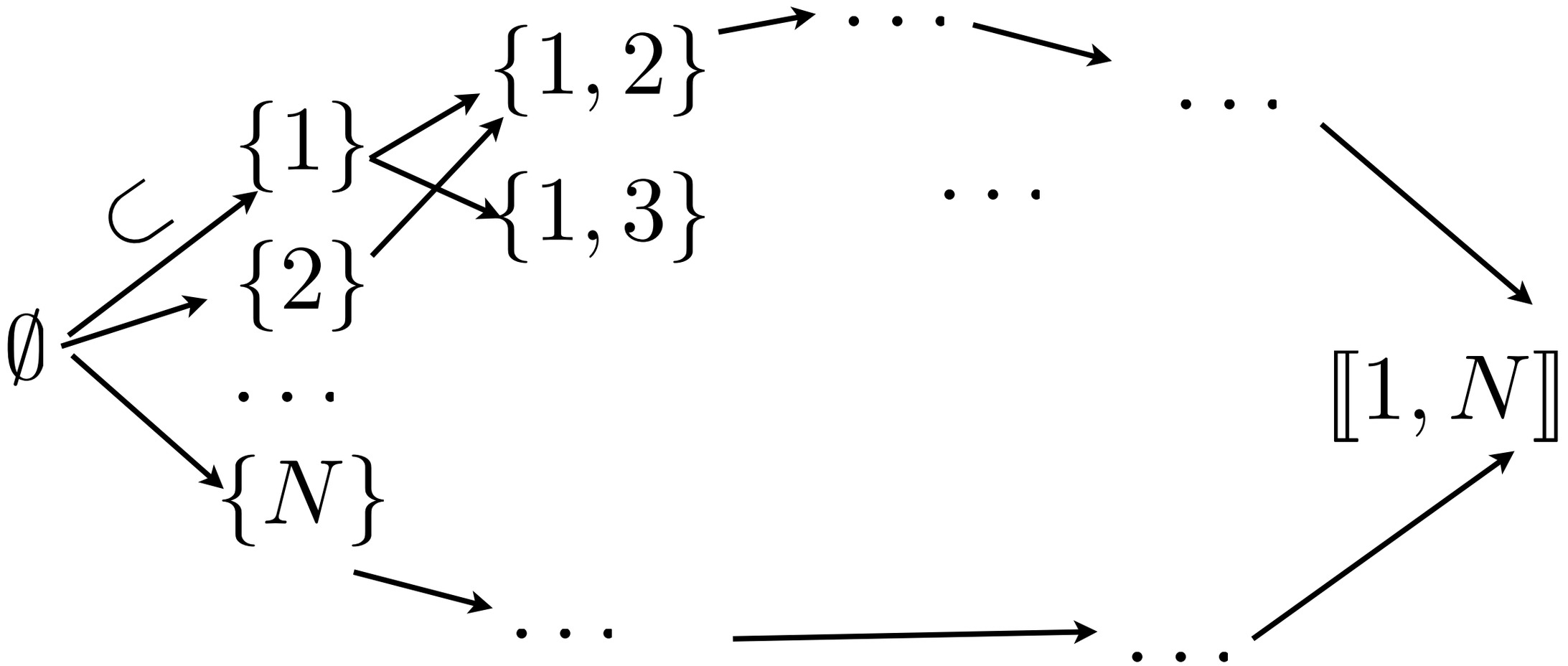
- Consider *x* with support set *I* and *x'* with $\mathbf{A}x' = \mathbf{A}x$
- Denote $z := x' - x \in \mathcal{N}(\mathbf{A})$ and observe

$$\begin{aligned}\|x'\|_f = \|x+z\|_f &= \|(x+z)_I\|_f + \|(x+z)_{I^c}\|_f \\ &= \|x + z_I\|_f + \|z_{I^c}\|_f \\ &\geq \|x\|_f - \|z_I\|_f + \|z_{I^c}\|_f\end{aligned}$$

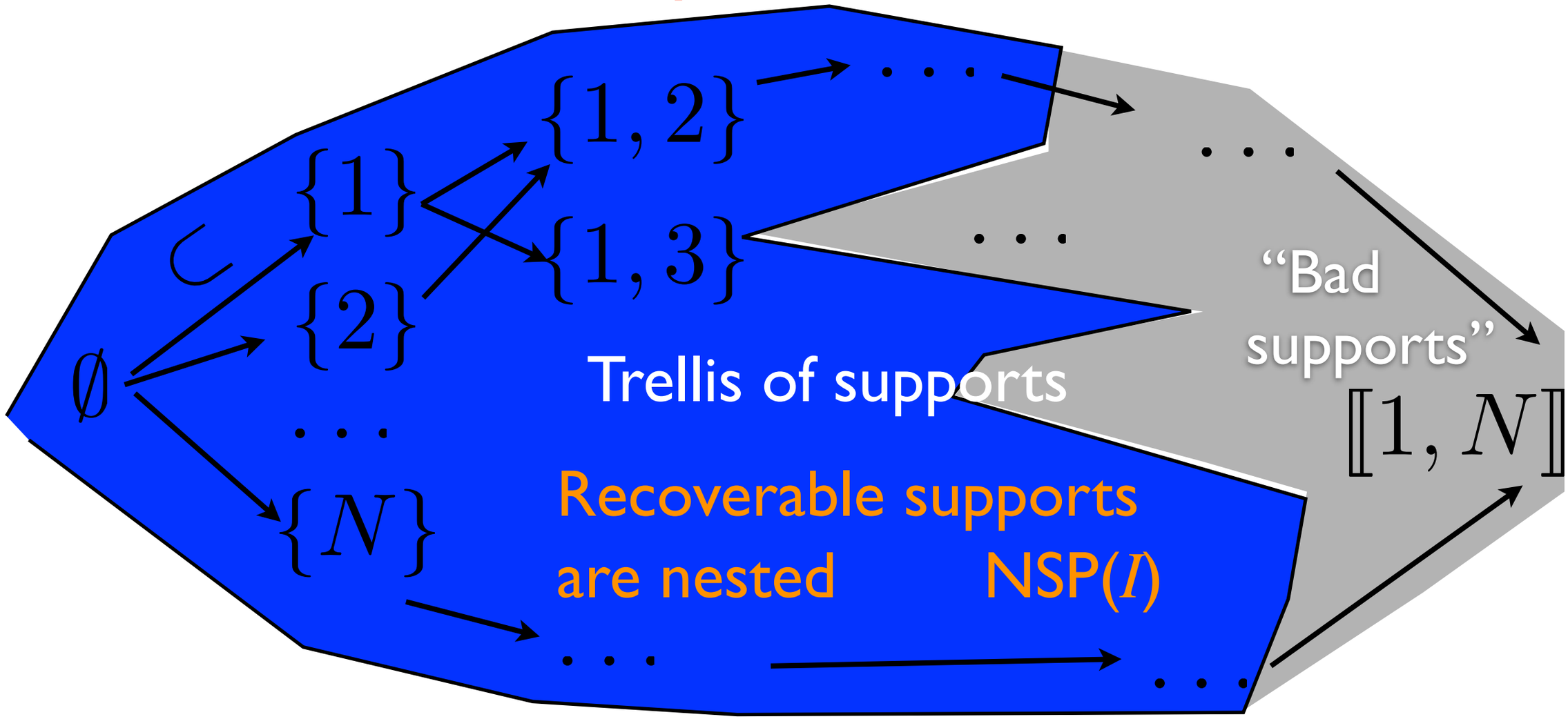
- Conclude:

If $\|z_{I^c}\|_f > \|z_I\|_f$ when $z \in \mathcal{N}(\mathbf{A})$ then I is recoverable

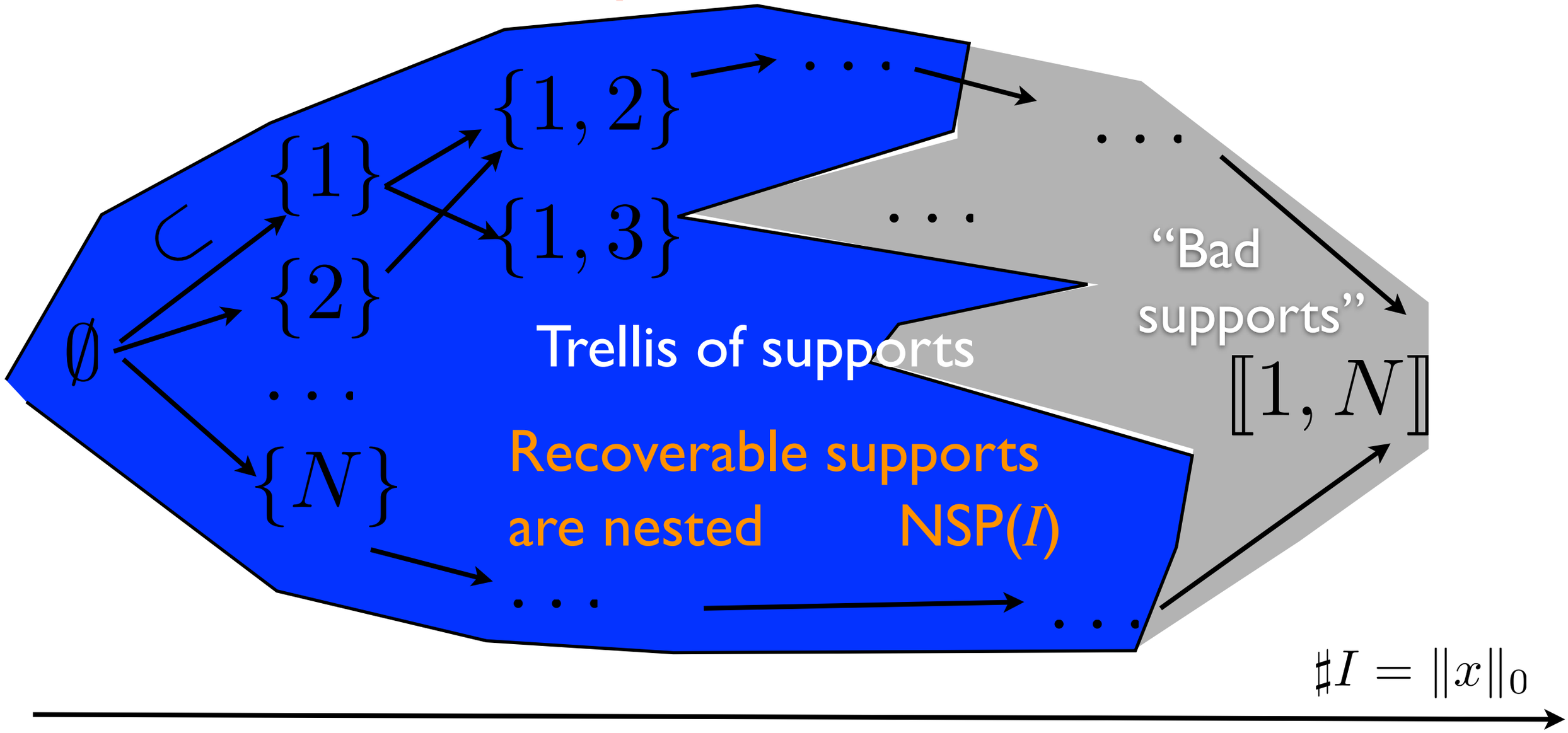
From “recoverable” supports to “sparse” vectors



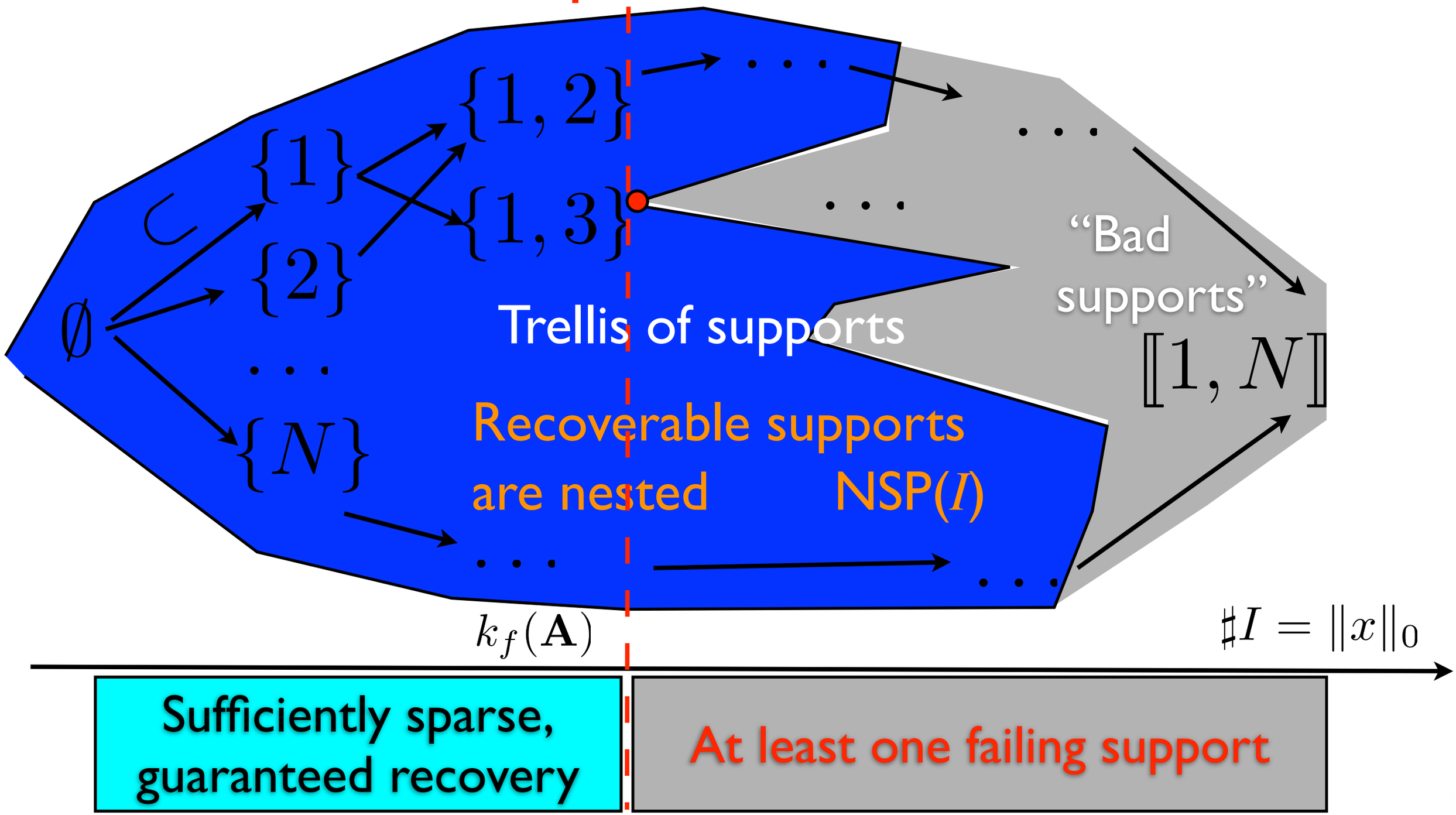
From “recoverable” supports to “sparse” vectors



From “recoverable” supports to “sparse” vectors



From “recoverable” supports to “sparse” vectors



Recoverable sparsity levels: the "Null Space Property" (2)

- **Corollary 1** [*Donoho & Huo 2001 for L1, G. Nielsen 2003 for Lp*]
  - ✓ Definition :

$$I_k = \quad \text{index of } k \text{ largest components of z}$$

  - ✓ Assumption :

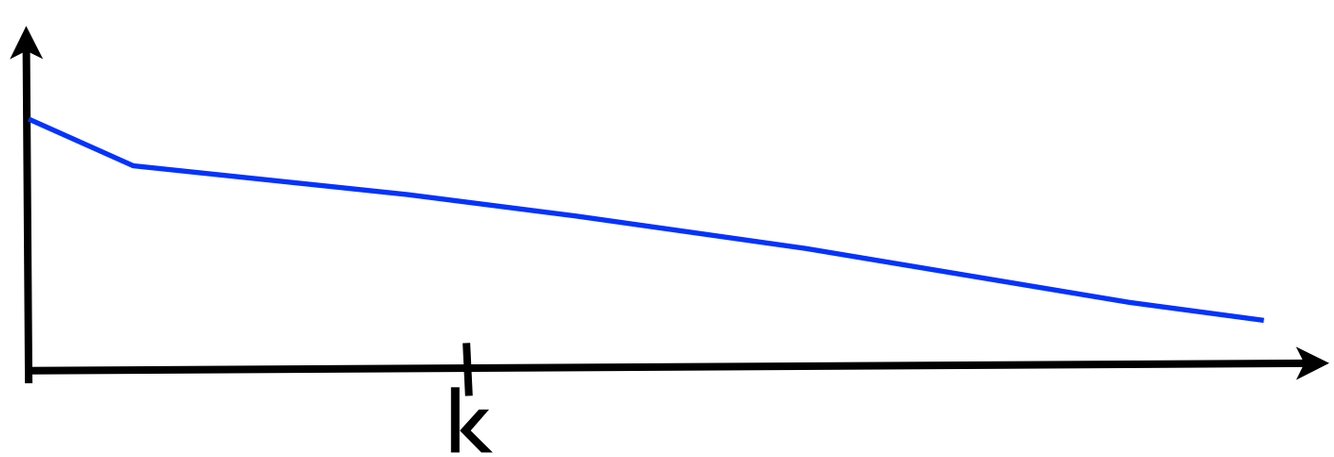
NSP

$$\|z_{I_k}\|_f < \|z_{I_k^c}\|_f \quad \text{when } z \in \mathcal{N}(\mathbf{A}), z \neq 0$$

- ✓ Conclusion: $x_f^\star$ recovers every x with $\|x\|_0 \leq k$
- ✓ The result is sharp: if NSP fails there is **at least one failing vector** x with $\|x\|_0 = k$

Interpretation of NSP

- Geometry in coefficient space:
  - ✓ consider an element *z* of the Null Space of A
  - ✓ order its entries in decreasing order

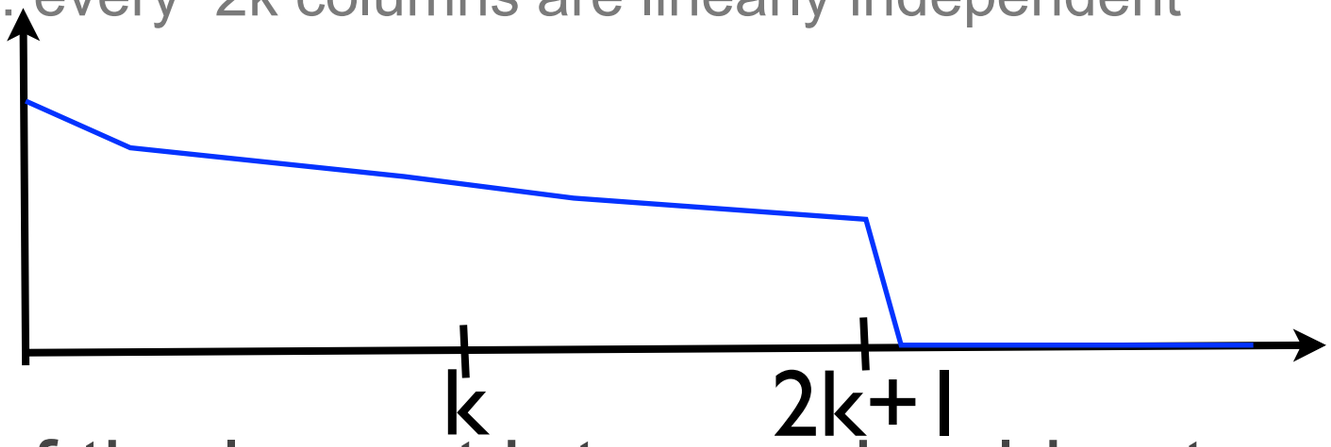


  - ✓ the mass of the largest k-terms should not exceed that of the tail $\|z_{I_k}\|_f < \|z_{I_k^c}\|_f$

Null space vectors must be “flat”, not sparse

NSP for L0:
Identifiability of sparse representations

- Case of L0: **identifiability**
  - ✓ L0 min = **guaranteed unique sparsest solution** if
    - ✦ elements in null space have at least 2k+1 nonzeros
    - ✦ equivalently: every 2k columns are linearly independent

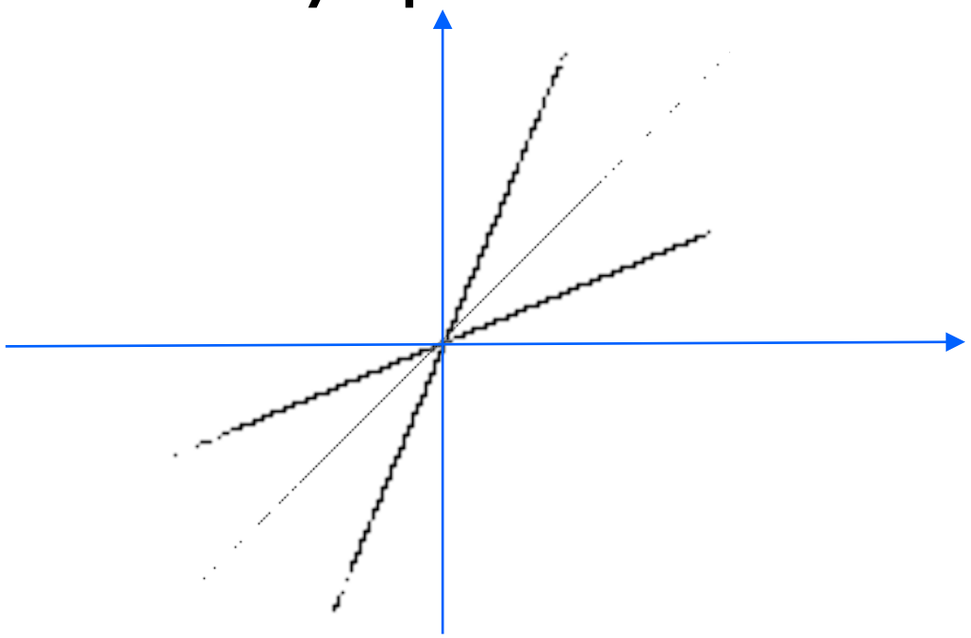


  - ✓ the mass of the largest k-terms should not exceed that of the tail $k = \|z_{I_k}\|_0 < \|z_{I_k^c}\|_0$

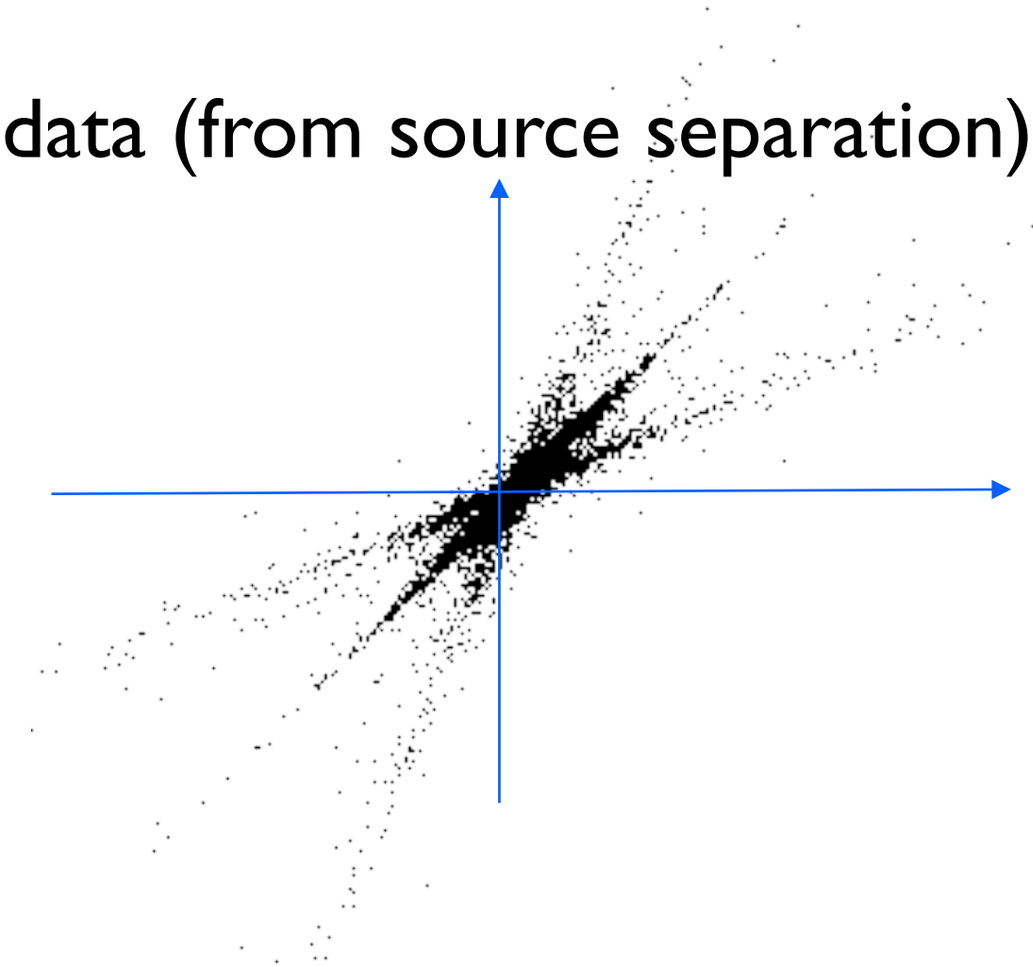
Stability and robustness

Need for stable recovery

Exactly sparse data



Real data (from source separation)



Formalization of stability

- Toy problem: exact recovery from $\mathbf{b} = \mathbf{A}x$
  - ✓ Assume sufficient sparsity $\|x\|_0 \leq k_p(\mathbf{A}) < m$
  - ✓ Wish to obtain $x_p^\star(\mathbf{b}) = x$

- Need to relax sparsity assumption
  - ✓ New benchmark = best k-term approximation

$$\sigma_k(x) = \inf_{\|y\|_0 \leq k} \|x - y\|$$

  - ✓ Goal = stable recovery = *instance optimality*

$$\|x_p^\star(\mathbf{b}) - x\| \leq C \cdot \sigma_k(x)$$

[Cohen, Dahmen & De Vore 2006]

Stability for Lp minimization

- Assumption: «stable Null Space Property»

NSP(k, ℓ_p^p, t)

$$\|z_{I_k}\|_p^p \leq t \cdot \|z_{I_k^c}\|_p^p \qquad \text{when} \quad z \in \mathcal{N}(\mathbf{A}), z \neq 0$$

- Conclusion: *instance optimality* for all *x*

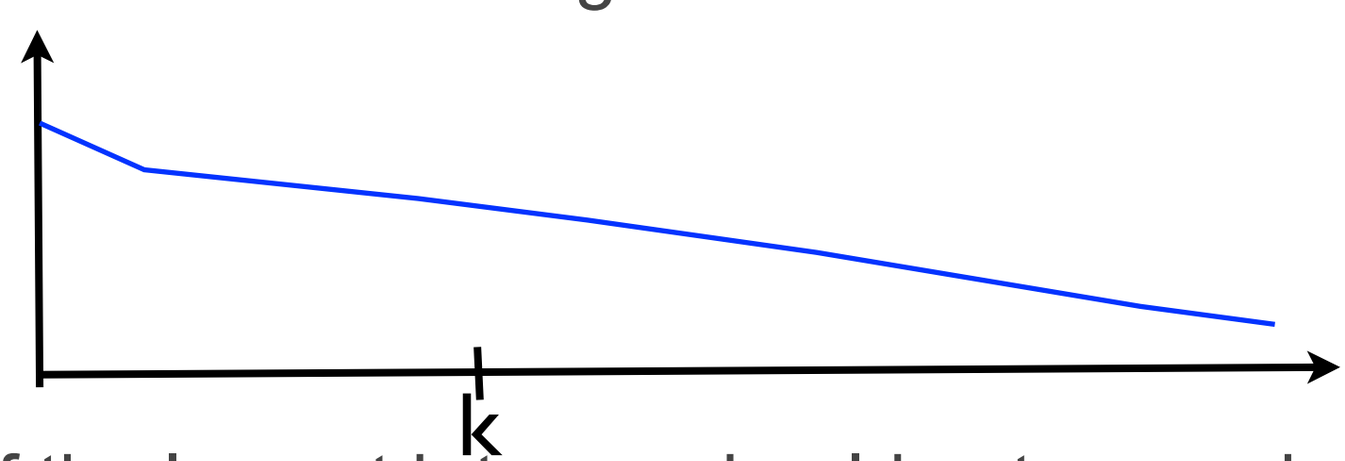
$$\|x_p^\star(\mathbf{b}) - x\|_p^p \leq C(t) \cdot \sigma_k(x)_p^p$$

$$C(t) := 2\frac{1+t}{1-t}$$

[Davies & Gribonval, SAMPTA 2009]

Reminder on NSP

- Geometry in coefficient space:
  - ✓ consider an element *z* of the Null Space of A
  - ✓ order its entries in decreasing order



  - ✓ the mass of the largest k-terms should not exceed a fraction of that of the tail

$$\|z_{I_k}\|_p^p \leq t \cdot \|z_{I_k^c}\|_p^p$$

Null space vectors must be “flat”, not sparse

Robustness

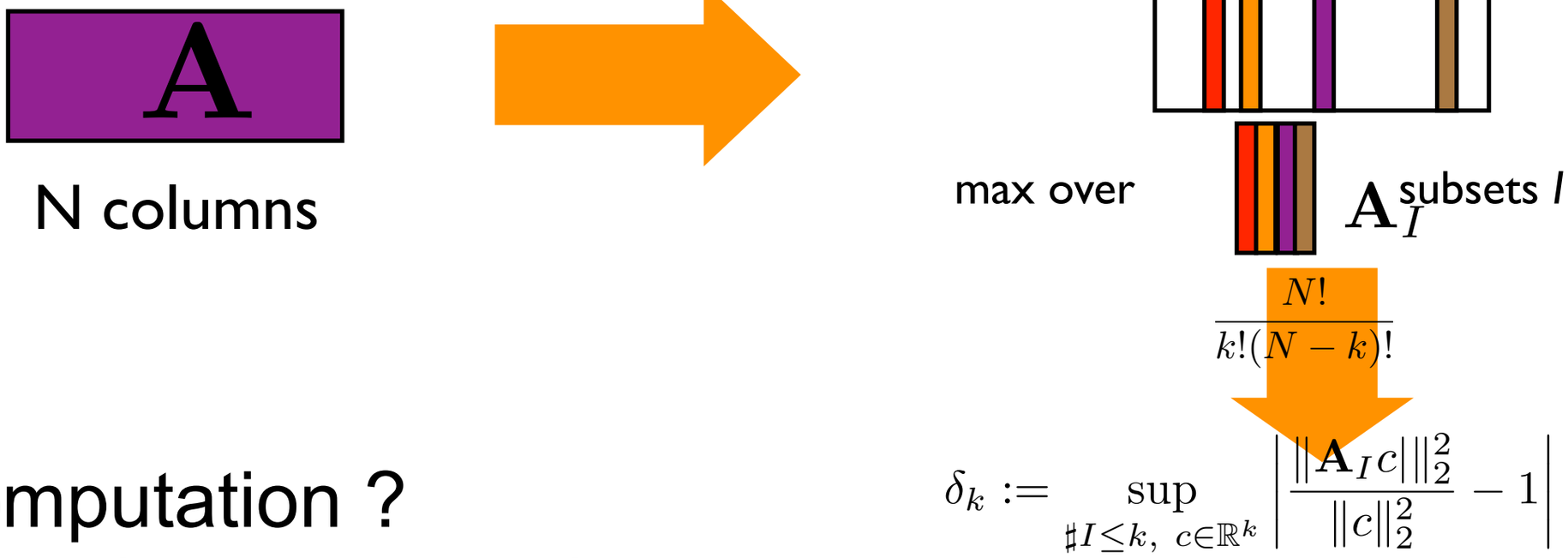
- Toy model = noiseless $\mathbf{b} = \mathbf{A}x$
- Need to account for noise $\mathbf{b} = \mathbf{A}x + \mathbf{e}$
  - ✓ measurement noise
  - ✓ modeling error
  - ✓ numerical inaccuracies ...
- Goal: predict robust estimation

$$\|x_p^\star(\mathbf{b}) - x\| \leq C\|e\| + C'\sigma_k(x)$$

- Tool: restricted isometry property

Restricted Isometry Property

- Definition



- Computation ?
  - ✓ naively: combinatorial
  - ✓ **open question**: NP ? NP-complete ?

Stability & robustness from RIP

RIP(k, δ)

$$\delta_{2k}(\mathbf{A}) \leq \delta$$

[Candès 2008]

$$t := \sqrt{2}\delta/(1-\delta)$$

NSP(k, ℓ^1,t)

$$\|z_{I_k}\|_1 \leq t \cdot \|z_{I_k^c}\|_1 \quad \textbf{when} \quad z \in \mathcal{N}(\mathbf{A}), z \neq 0$$

- Result: **stable + robust** L1-recovery under assumption that

$$\delta_{2k}(\mathbf{A}) < \sqrt{2} - 1 \approx 0.414$$

- ✓ *Foucart-Lai 2008*: Lp with p<1, and $\delta_{2k}(\mathbf{A}) < 0.4531$
- ✓ *Chartrand 2007, Saab & Yilmaz 2008*: other RIP condition for p<1
- ✓ *G., Figueras & Vandergheynst 2006*: robustness with *f*-norms
- ✓ *Needell & Tropp 2009, Blumensath & Davies 2009*: RIP for greedy algorithms

Is the RIP a sharp condition ?

- The Null Space Property
  - ✓ “algebraic” + sharp property for Lp, only depends on $\mathcal{N}(\mathbf{A})$

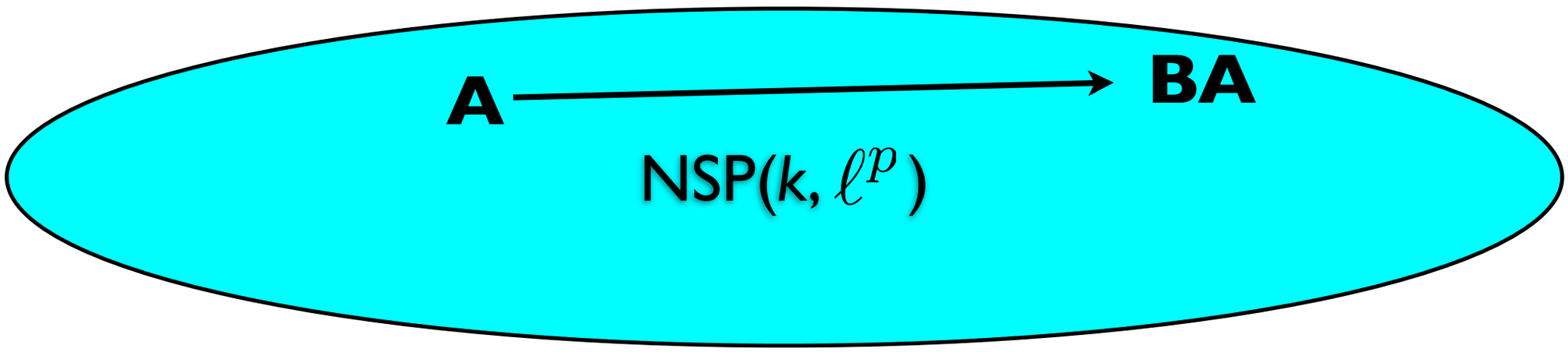
A

NSP(k, ℓ^p)

[Davies & Gribonval, IEEE Inf.Th. 2009]

Is the RIP a sharp condition ?

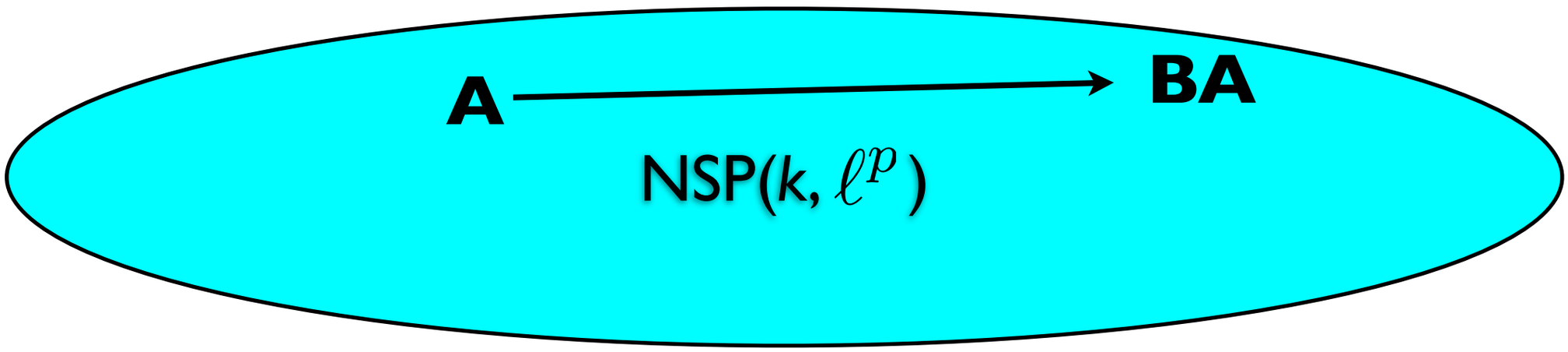
- The Null Space Property
  - ✓ “algebraic” + sharp property for Lp, only depends on $\mathcal{N}(\mathbf{A})$
  - ✓ invariant by linear transforms $\mathbf{A} \rightarrow \mathbf{BA}$



[Davies & Gribonval, IEEE Inf.Th. 2009]

Is the RIP a sharp condition ?

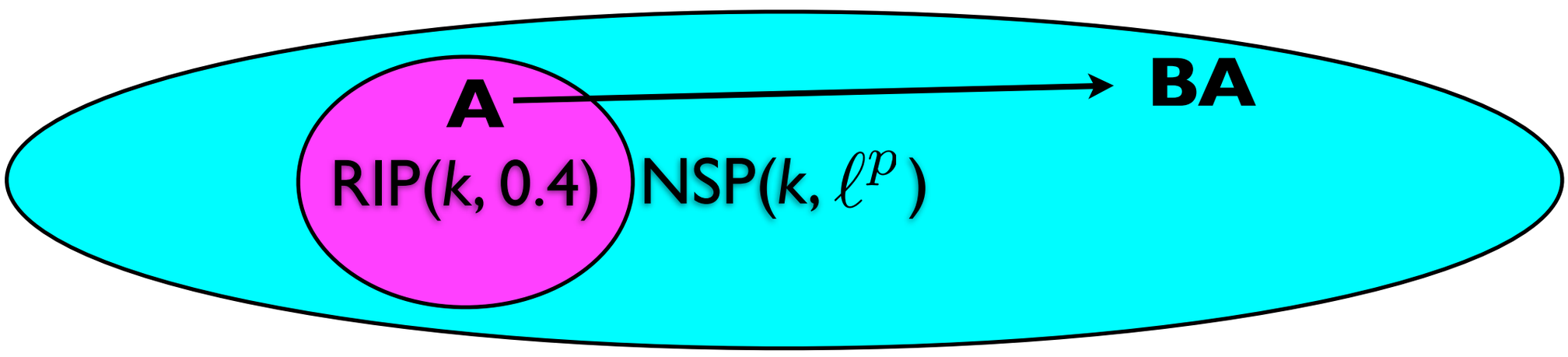
- The Null Space Property
  - ✓ "algebraic" + sharp property for Lp, only depends on $\mathcal{N}(\mathbf{A})$
  - ✓ invariant by linear transforms $\mathbf{A} \rightarrow \mathbf{BA}$
- The RIP(k, δ) condition



[Davies & Gribonval, IEEE Inf.Th. 2009]

Is the RIP a sharp condition ?

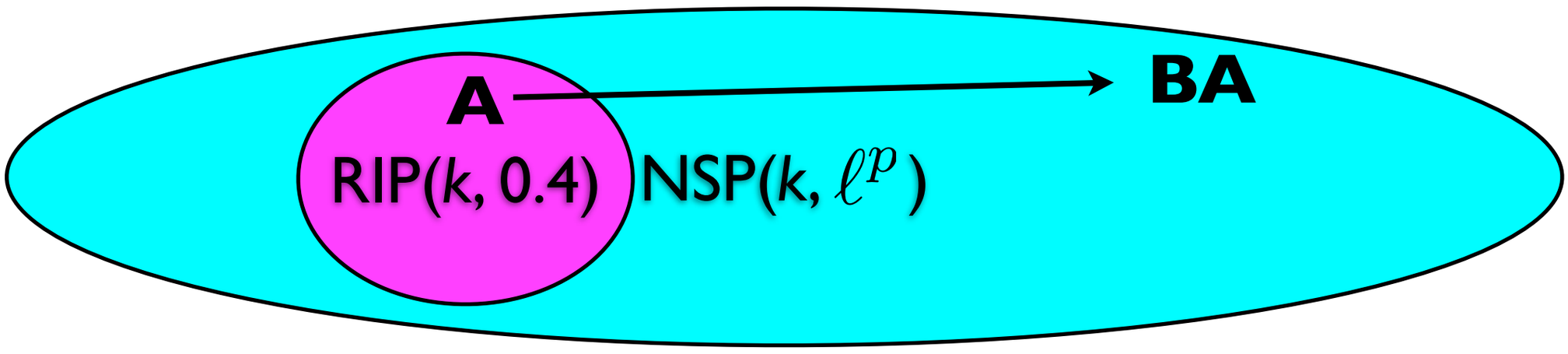
- The Null Space Property
  - ✓ "algebraic" + sharp property for Lp, only depends on $\mathcal{N}(\mathbf{A})$
  - ✓ invariant by linear transforms $\mathbf{A} \rightarrow \mathbf{BA}$
- The RIP(k, δ) condition
  - ✓ "metric" ... and not invariant by linear transforms



[Davies & Gribonval, IEEE Inf.Th. 2009]

Is the RIP a sharp condition ?

- The Null Space Property
  - ✓ "algebraic" + sharp property for Lp, only depends on $\mathcal{N}(\mathbf{A})$
  - ✓ invariant by linear transforms $\mathbf{A} \rightarrow \mathbf{BA}$
- The RIP(k, δ) condition
  - ✓ "metric" ... and not invariant by linear transforms
  - ✓ predicts performance + **robustness of *several* algorithms**



[Davies & Gribonval, IEEE Inf.Th. 2009]

Comparison between algorithms

- Recovery conditions based on number of nonzero components $\|x\|_0$ for $0 \leq q \leq p \leq 1$

$$k*_{\mathrm{MP}}(\mathbf{A}) \leq k_1(\mathbf{A}) \leq k_p(\mathbf{A}) \leq k_q(\mathbf{A}) \leq k_0(\mathbf{A}), \forall \mathbf{A}$$

Proof

- **Warning** :
  - ✓ there often exists vectors beyond these critical sparsity levels, which are recovered
  - ✓ there often exists vectors beyond these critical sparsity levels, where the successful algorithm is not the one we would expect

[Gribonval & Nielsen, ACHA 2007]

Remaining agenda

- Recovery conditions based on number of nonzero components $\|x\|_0$ for $0 \leq q \leq p \leq 1$

$$k*_{\mathrm{MP}}(\mathbf{A}) \leq k_1(\mathbf{A}) \leq k_p(\mathbf{A}) \leq k_q(\mathbf{A}) \leq k_0(\mathbf{A}), \forall \mathbf{A}$$

- **Question**
  - ✓ what is the order of magnitude of these numbers ?
  - ✓ how do we estimate them in practice ?
- A first element:
  - ✓ if **A** is *m* x *N*, then $k_0(\mathbf{A}) \leq \lfloor m/2 \rfloor$
  - ✓ for almost all matrices (in the sense of Lebesgue measure in $\mathbb{R}^{mN}$) this is an equality

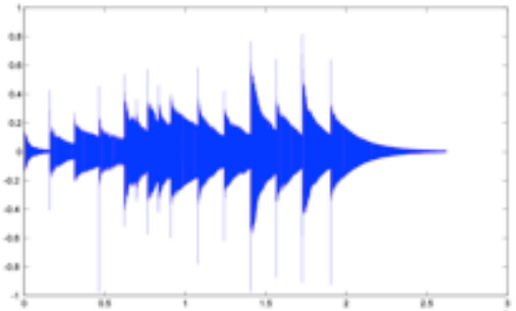
Explicit guarantees in various inverse problems

Scenarios

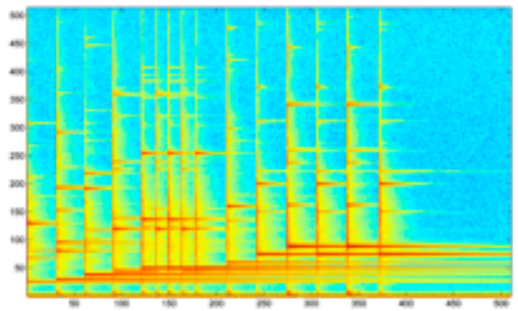
- Range of "choices" for the matrix **A**
  - ✓ Dictionary modeling structures of signals
    - ✦ **Constrained** choice = to fit the data.
    - ✦ *Ex: union of wavelets + curvelets + spikes*
  - ✓ «Transfer function» from physics of inverse problem
    - ✦ **Constrained** choice = to fit the direct problem.
    - ✦ *Ex: convolution operator / transmission channel*
  - ✓ Designed / chosen «Compressed Sensing» matrix
    - ✦ «**Free**» design = to maximize recovery performance *vs* cost of measures
    - ✦ *Ex: random Gaussian matrix... or coded aperture, etc.*
- Estimation of the recovery regimes
  - ✓ coherence for deterministic matrices
  - ✓ typical results for random matrices

Multiscale Time-Frequency Structures

- Audio = superimposition of structures



$$\mathbf{b} = \{b(t)\}_t$$



$$x = \{x(s, \tau, f)\}_{s,\tau,f}$$

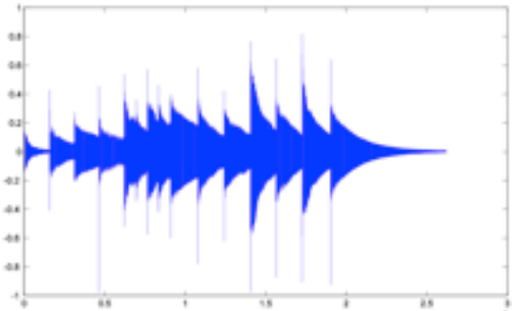
- ✓ transients = short, small scale
- ✓ harmonic part = long, large scale

- Gabor atoms $g_{s,\tau,f}(t) := \frac{1}{\sqrt{s}} w\left(\frac{t-\tau}{s}\right) e^{2i\pi f t}$
- Dictionary matrix: $\mathbf{A}_n = \{g_{s_n,\tau_n,f_n}(t)\}_t$

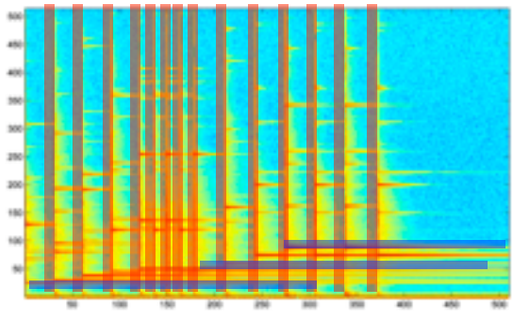
$$\mathbf{A} = [\mathbf{A}_1 \dots \mathbf{A}_N]$$

Multiscale Time-Frequency Structures

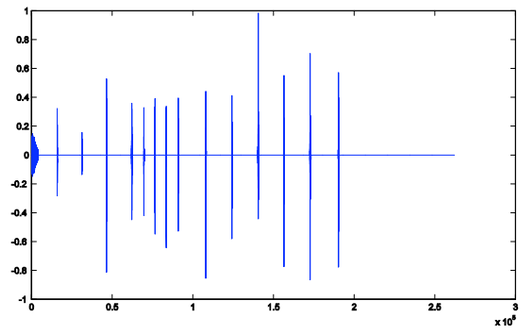
- Audio = superimposition of structures

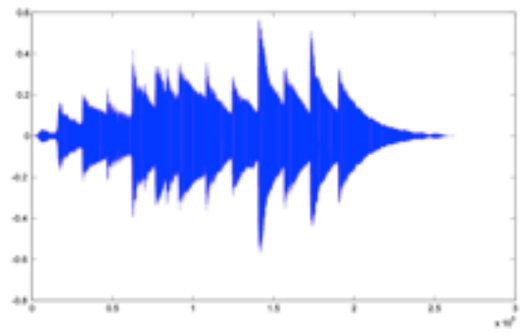


$$\mathbf{b} = \{b(t)\}_t$$



$$x = \{x(s, \tau, f)\}_{s,\tau,f}$$





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

- Gabor atoms $g_{s,\tau,f}(t) := \frac{1}{\sqrt{s}} w\left(\frac{t-\tau}{s}\right) e^{2i\pi f t}$
- Dictionary matrix: $\mathbf{A}_n = \{g_{s_n,\tau_n,f_n}(t)\}_t$

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

Deterministic matrices and coherence

- **Lemma**
  - ✓ Assume normalized columns $\|\mathbf{A}_i\|_2 = 1$
  - ✓ Define **coherence** $\mu = \max_{i \neq j} |\mathbf{A}_i^T \mathbf{A}_j|$
  - ✓ Consider index set I of size $\sharp I \leq k$
  - ✓ Then for any coefficient vector $c \in \mathbb{R}^I$

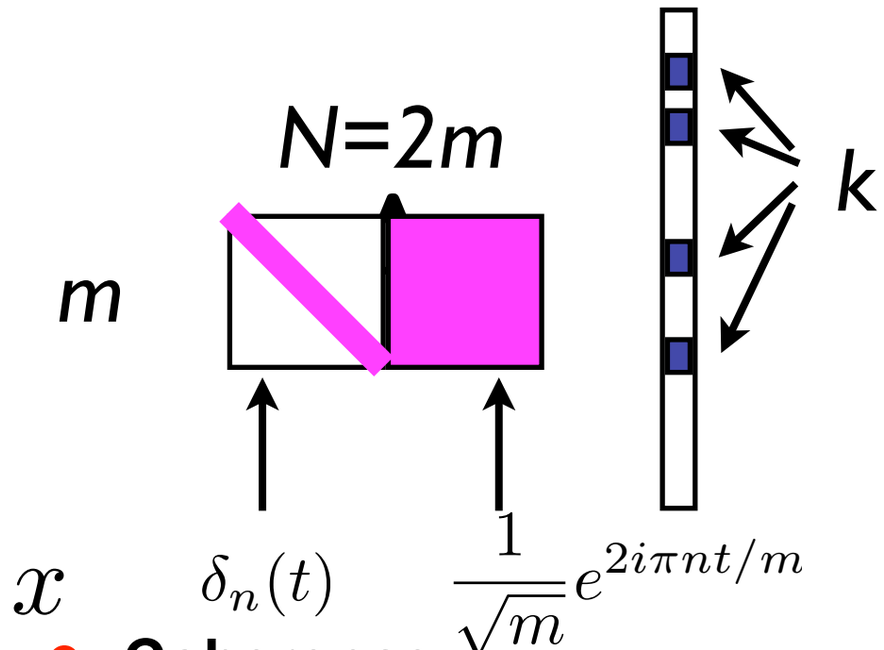
$$1 - (k-1)\mu \leq \frac{\|\mathbf{A}_I c\|_2^2}{\|c\|_2^2} \leq 1 + (k-1)\mu$$

  - ✓ In other words

$$\delta_{2k} \leq (2k-1)\mu$$

Coherence vs RIP

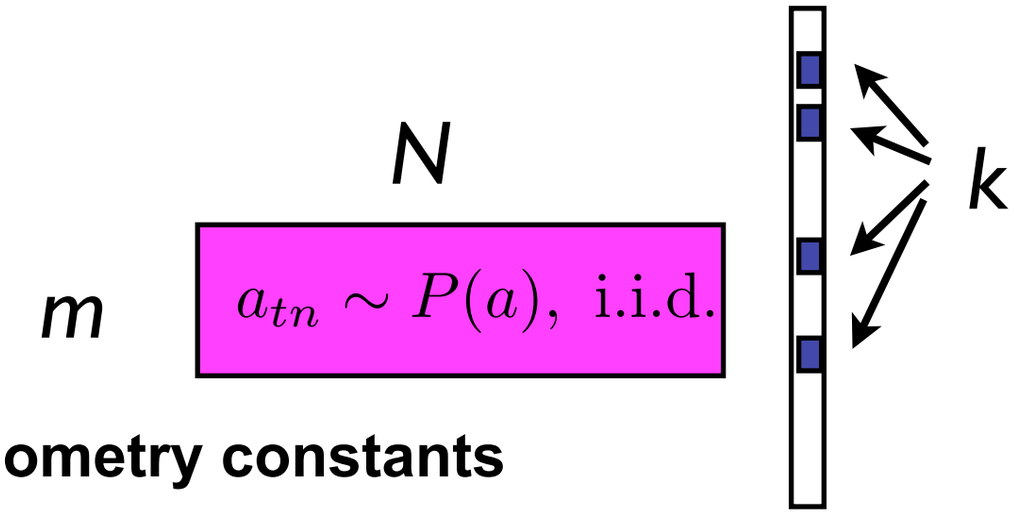
- **Deterministic** matrix, such as Dirac-Fourier dictionary



- **Coherence**

$$\mu = 1/\sqrt{m}$$

- "**Generic**" (random) dictionary
[Candès & al 2004, Vershynin 2006, ...]



- **Isometry constants**

if $\quad m \geq Ck \log N/k$

then $\quad P(\delta_{2k} < \sqrt{2} - 1) \approx 1$

Recovery regimes

$$k_1(\mathbf{A}) \approx 0.914\sqrt{m}$$
$$k^*_{\mathrm{MP}}(\mathbf{A}) \geq 0.5\sqrt{m}$$

[Elad & Bruckstein 2002]

$$k_1(\mathbf{A}) \approx \frac{m}{2e \log N/m}$$

[Donoho & Tanner 2009]

with high probability for *Gaussian* **A**

Example: convolution operator

- Deconvolution problem with spikes

$$b = h \star x + e$$

  - ✓ Matrix-vector form $\mathbf{b} = \mathbf{A}x + \mathbf{e}$ with **A** = Toeplitz or circulant matrix $[\mathbf{A}_1, \ldots, \mathbf{A}_N]$

$$\mathbf{A}_n(i) = h(i-n)$$

by convention $\|\mathbf{A}_n\|_2^2 = \sum_i h(i)^2 = 1$

  - ✓ Coherence = autocorrelation, can be large

$$\mu = \max_{n \neq n'} \mathbf{A}_n^T \mathbf{A}_{n'} = \max_{\ell \neq 0} h \star \tilde{h}(\ell)$$

  - ✓ Recovery guarantees
    - ✦ Worst case = close spikes, usually difficult and not robust
    - ✦ Stronger guarantees assuming distance between spikes *[Dossal 2005]*
  - ✓ Algorithms: exploit fast convolution to apply **A** and adjoint.

Example: image inpainting

Courtesy of: G. Peyré, Ceremade, Université Paris 9 Dauphine

Inpainting

Wavelets

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

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

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

Compressed sensing

- Approach = acquire some data *y* with a limited number *m* of (linear) measures, modeled by a **measurement matrix** $\mathbf{M}$**:** $\mathbf{b} \approx \mathbf{M}y$
- Key hypotheses
  - ✓ Sparse model: the data can be sparsely represented in a known dictionary $\mathbf{\Phi}$: $y \approx \mathbf{\Phi} x$, with $\sigma_k(x) \ll \|x\|$
  - ✓ Overall matrix $\mathbf{A} = \mathbf{M}\mathbf{\Phi}$ is «incoherent»: $\delta_{2k}(\mathbf{A}) \ll 1$ leading to robust + stable sparse recovery
- Reconstruction = sparse recovery algorithm

Compressed Sensing: key requirements

- Sparse model= dictionary Φ
  - ✓ need to «fit» the **data**
    - ✦ does not always exist: e.g. white Gaussian noise cannot be sparsified!
  - ✓ dictionary design:
    - ✦ expert knowledge
    - ✦ **dictionary selection** from a library (wavelets, curvelets, Gabor, ...)
    - ✦ **dictionary learning**
- Measurement matrix $\mathbf{M}$
  - ✓ **physically feasible:** hardware implementation!
  - ✓ **recovery guarantees:** incoherence of $\mathbf{M\Phi}$
- Efficiency:
  - ✓ **fast computation** of $\mathbf{M\Phi}y, (\mathbf{M\Phi})^T\mathbf{b}$

Compressed Sensing: when is it worth it ?

- **Worthless** if high-res. sensing+storage = cheap

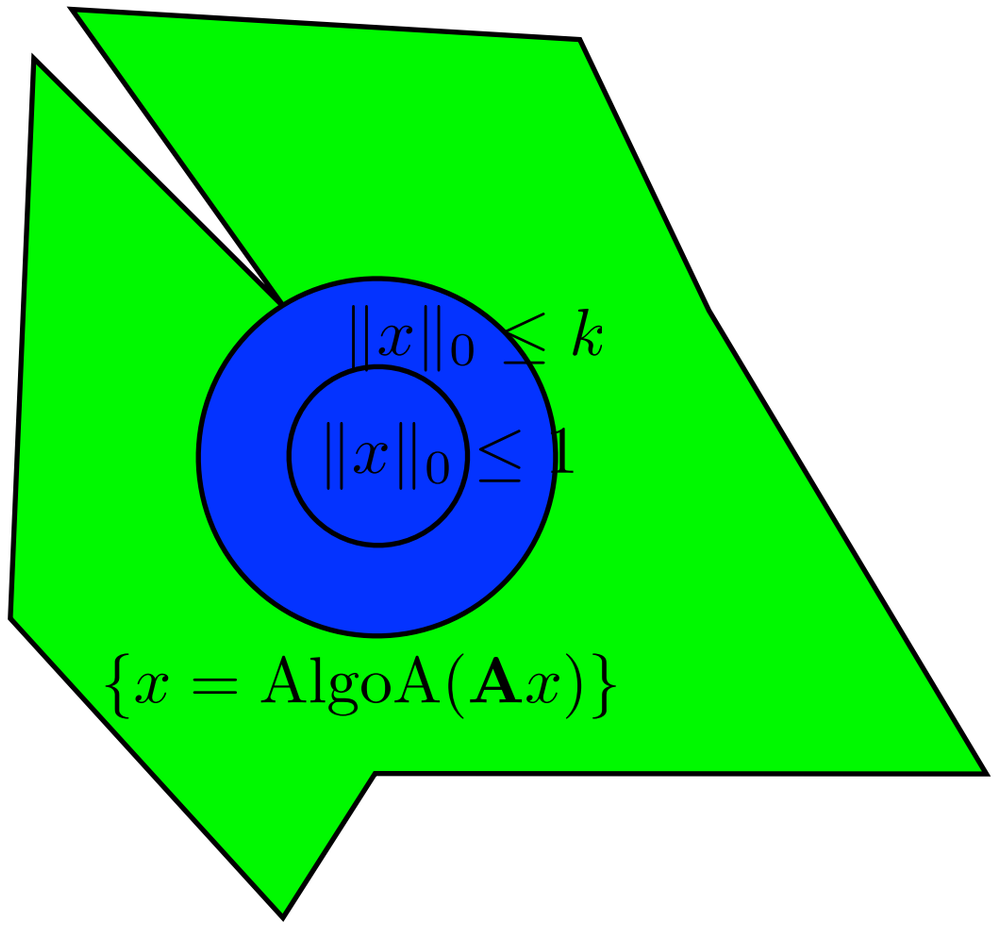
  i.e., not for your personal digital camera!
- **Worth it** whenever
  - ✓ High-res. = impossible
    - ✦ no miniature sensor, e.g, certain wavelength
  - ✓ Cost of each measure is high
    - ✦ Time constraints [fMRI]
    - ✦ Economic constraints [well drilling]
    - ✦ Intelligence constraints [furtive measures]?
    - ✦ Constraints on data flow
  - ✓ Transmission is lossy
    (CS=robust to loss of a few measures)

Excessive pessimism ?

Recovery analysis

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

- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
- Worst case
  = too pessimistic!



Recovery analysis

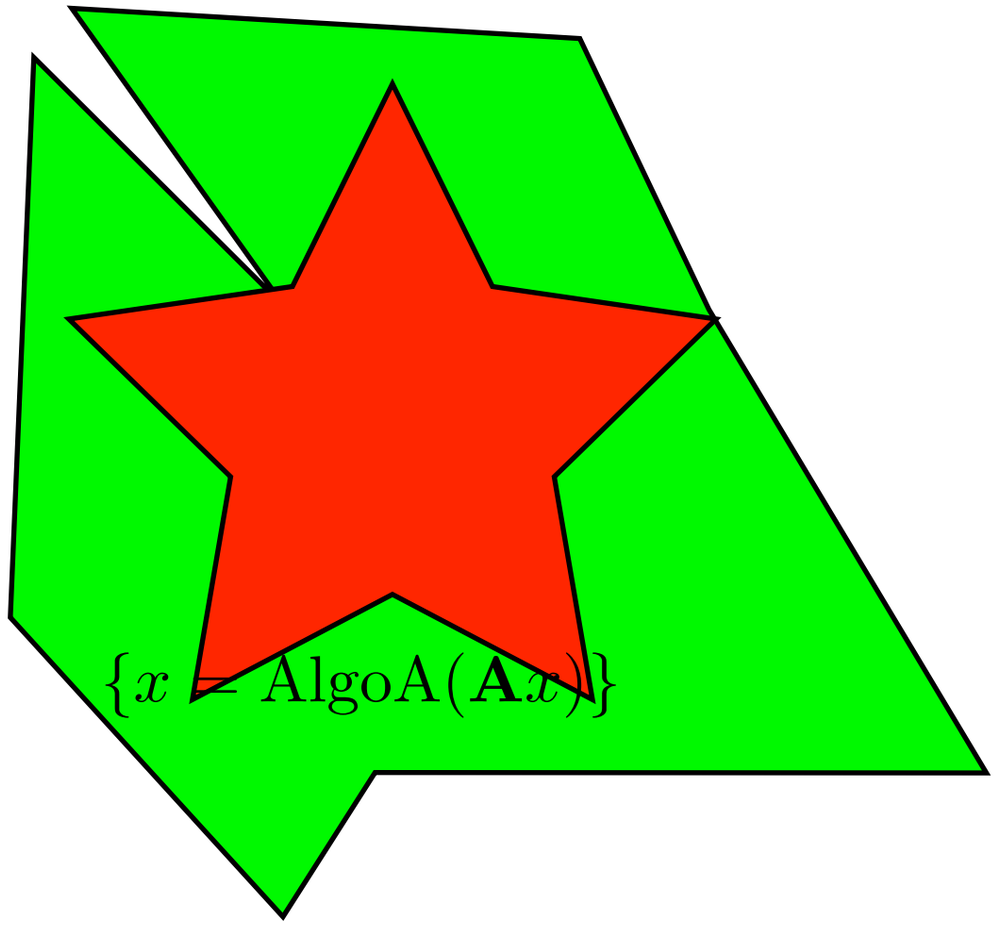
$$\mathbf{b} = \mathbf{A}x$$

- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
- Worst case
  = too pessimistic!
- Finer “structures” of x

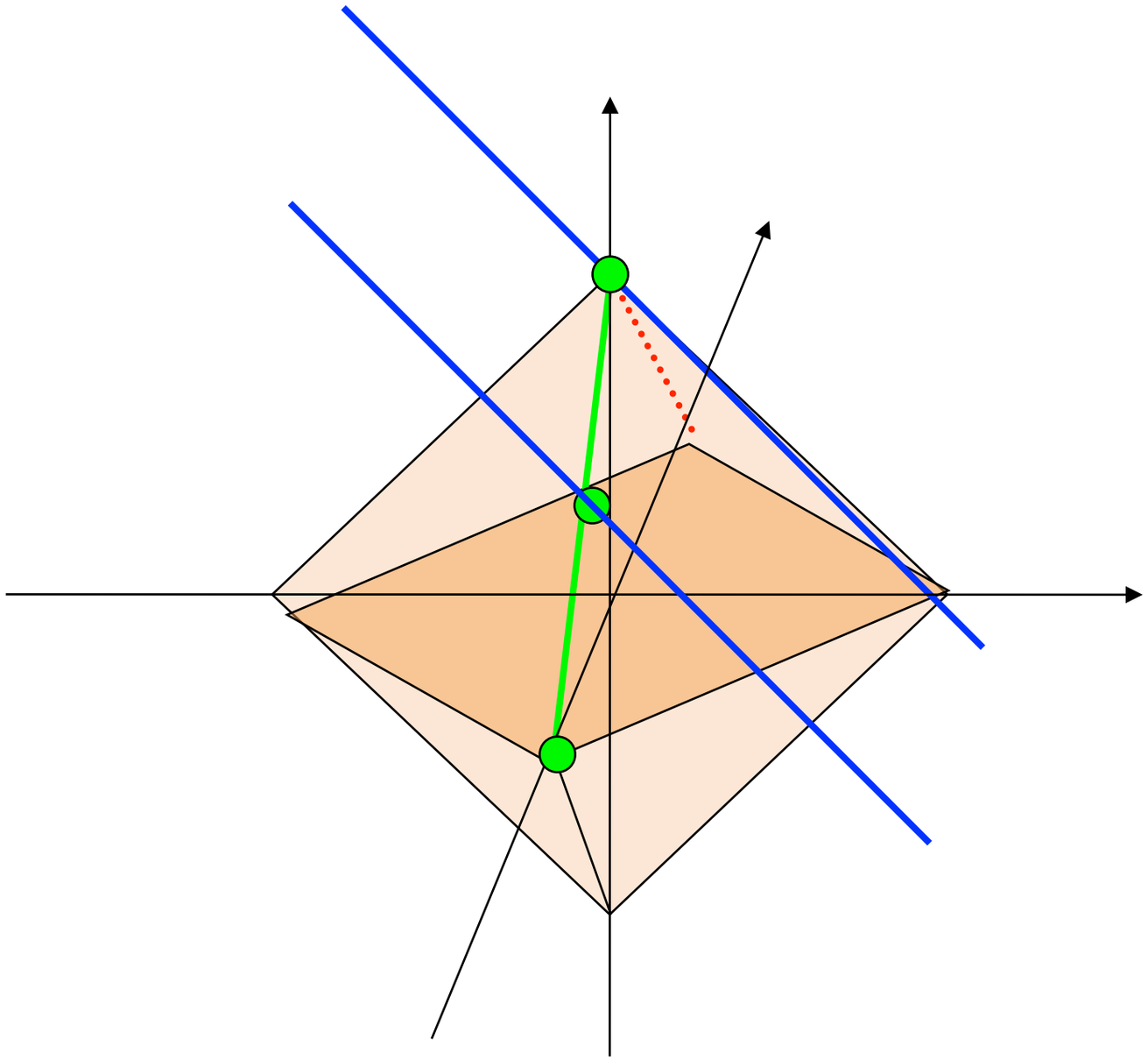
*Borup, G. & Nielsen ACHA 2008, **A** = Wavelets U Gabor, recovery of infinite supports for analog signals*

$$\mathrm{support}(x), \mathrm{sign}(x)$$

Fuchs 2005; Zhao & Yu 2006; Zou 2006; Yuan & Lin 2007; Wainwright 2009;



L1 recovery beyond $k_1(\mathbf{A})$



Recovery analysis

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

- Recoverable set for a given “inversion” algorithm
- Level sets of L0-norm
- Worst case
  = too pessimistic!
- Finer “structures” of x

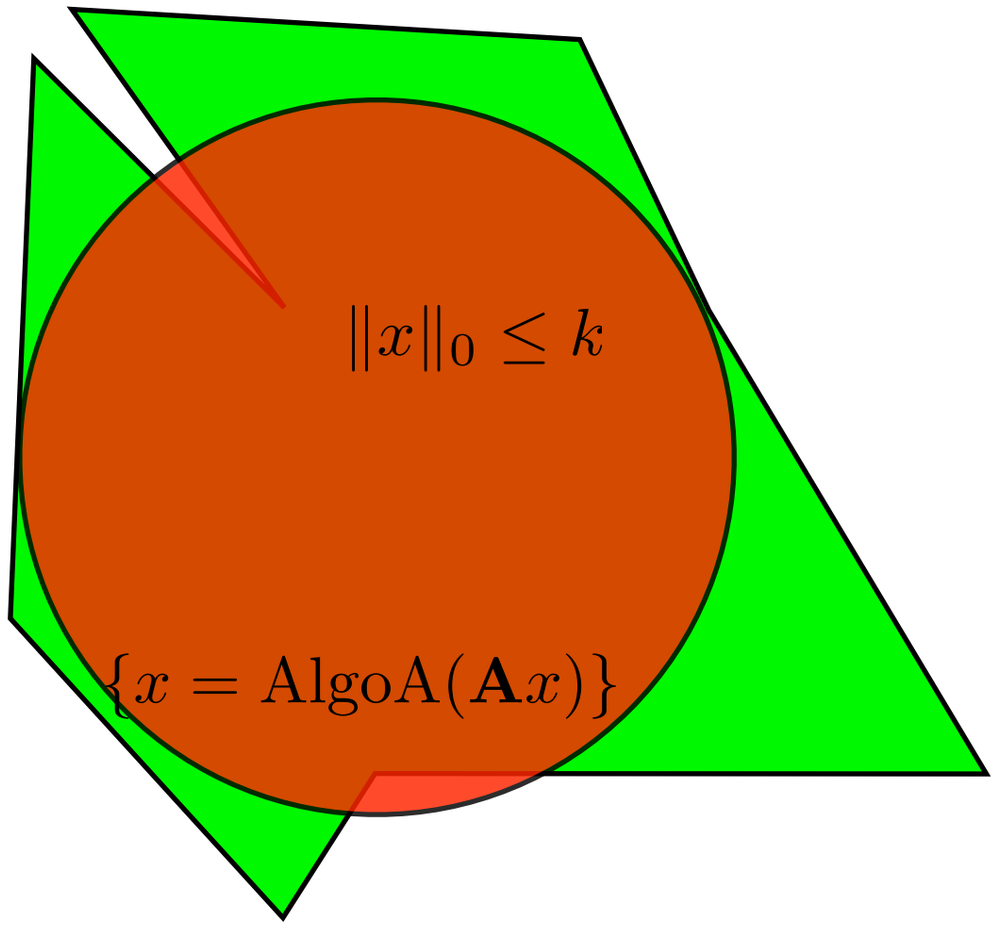
*Borup, G. & Nielsen ACHA 2008, **A** = Wavelets U Gabor, recovery of infinite supports for analog signals*

$$\mathrm{support}(x), \mathrm{sign}(x)$$

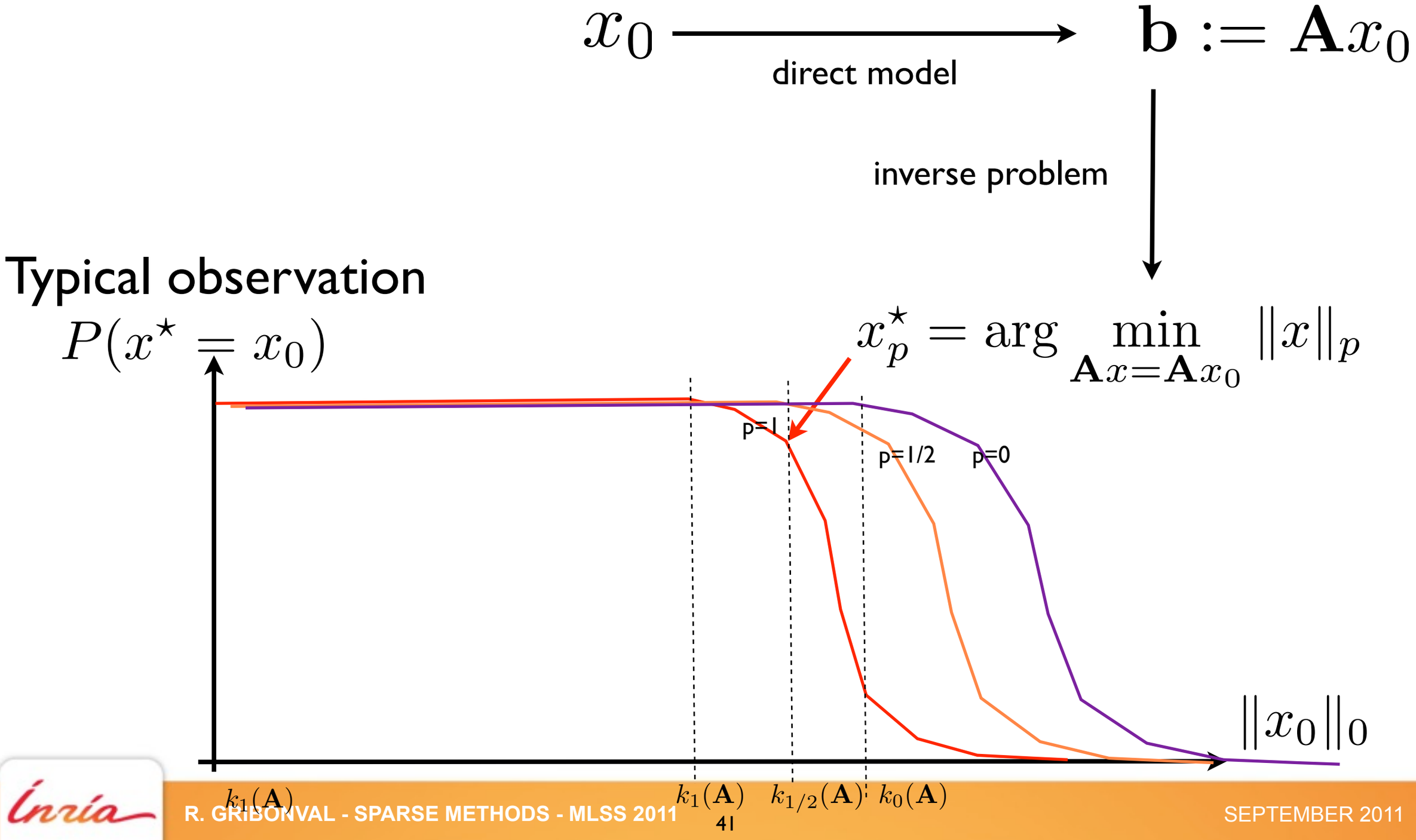
Fuchs 2005; Zhao & Yu 2006; Zou 2006; Yuan & Lin 2007; Wainwright 2009;

- Average/typical case

G., Rauhut,, Schnass & Vandergheynst, JFAA 2008, “Atoms of all channels, unite! Average case analysis of multichannel sparse recovery using greedy algorithms”.



Average case analysis ?



Average case analysis ?

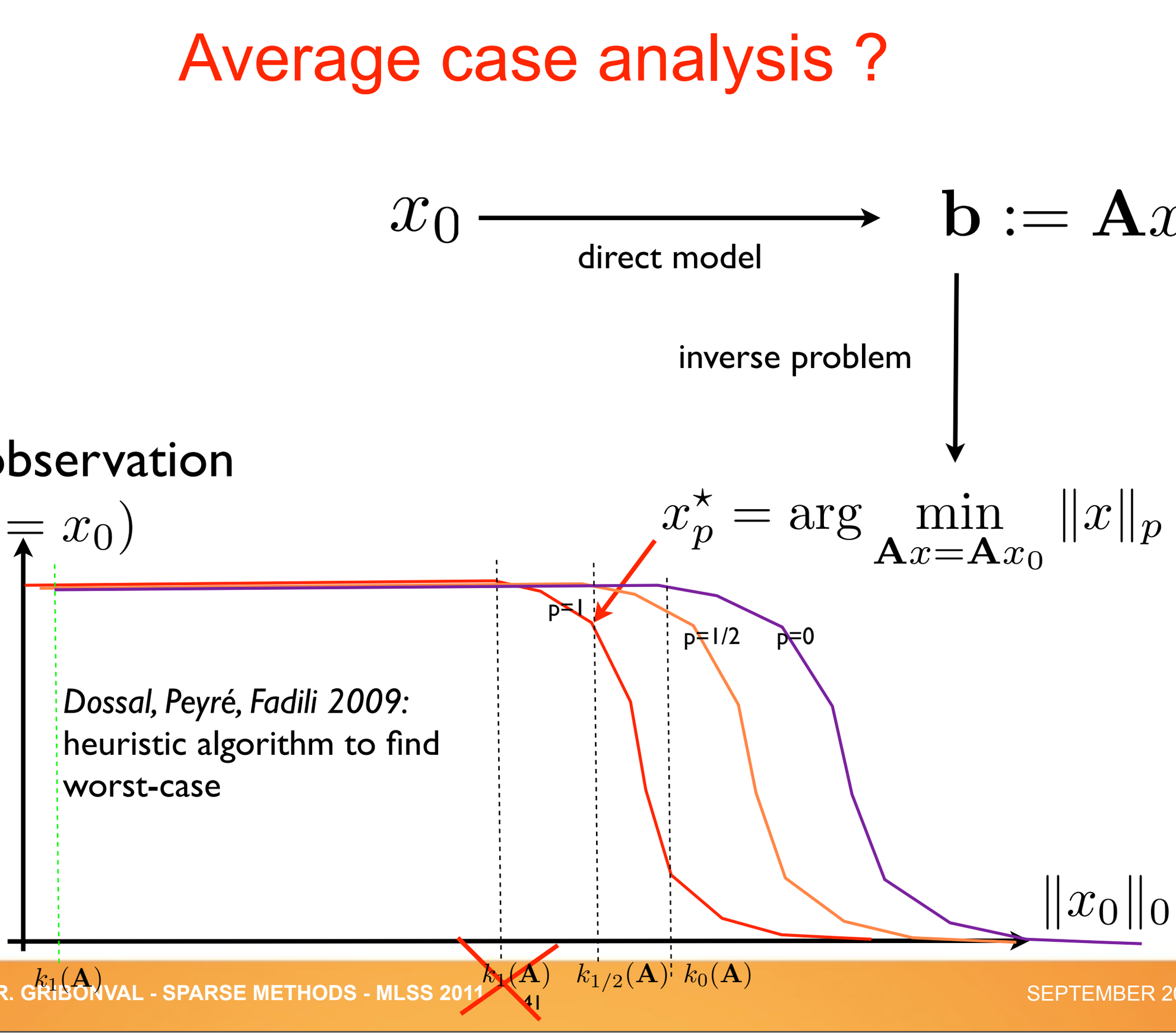
$$x_0 \xrightarrow[\text{direct model}]{} \mathbf{b} := \mathbf{A}x_0$$

inverse problem

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

Typical observation

$P(x^\star = x_0)$



Dossal, Peyré, Fadili 2009:
heuristic algorithm to find worst-case

Average case analysis ?

$x_0 \longrightarrow \mathbf{b} := \mathbf{A}x_0$

direct model

inverse problem

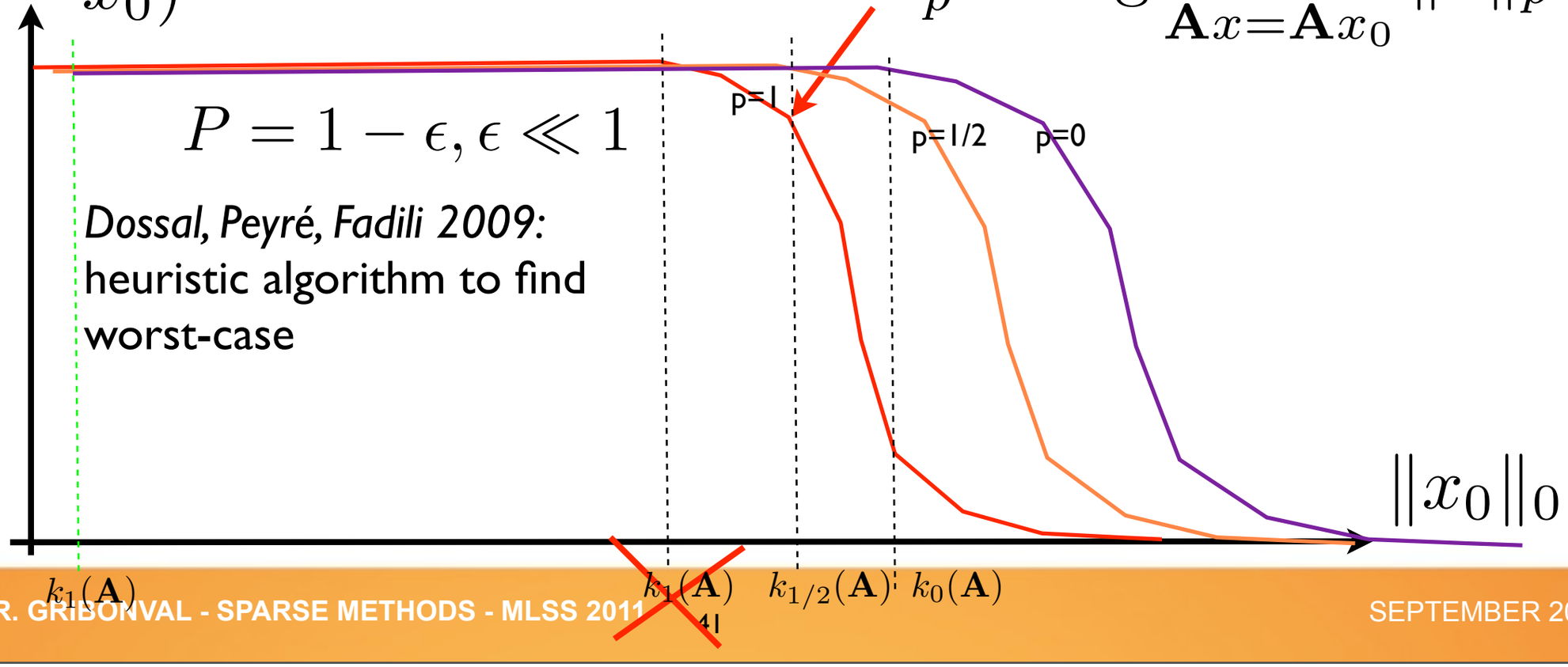
$x_p^\star = \arg \min_{\mathbf{A}x = \mathbf{A}x_0} \|x\|_p$

Typical observation

$P(x^\star = x_0)$

$P = 1 - \epsilon, \epsilon \ll 1$

Dossal, Peyré, Fadili 2009:
heuristic algorithm to find worst-case



Average case analysis ?

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

Bayesian! Favorable priors?

inverse problem

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

Typical observation

$P(x^\star = x_0)$

$P = 1 - \epsilon, \epsilon \ll 1$

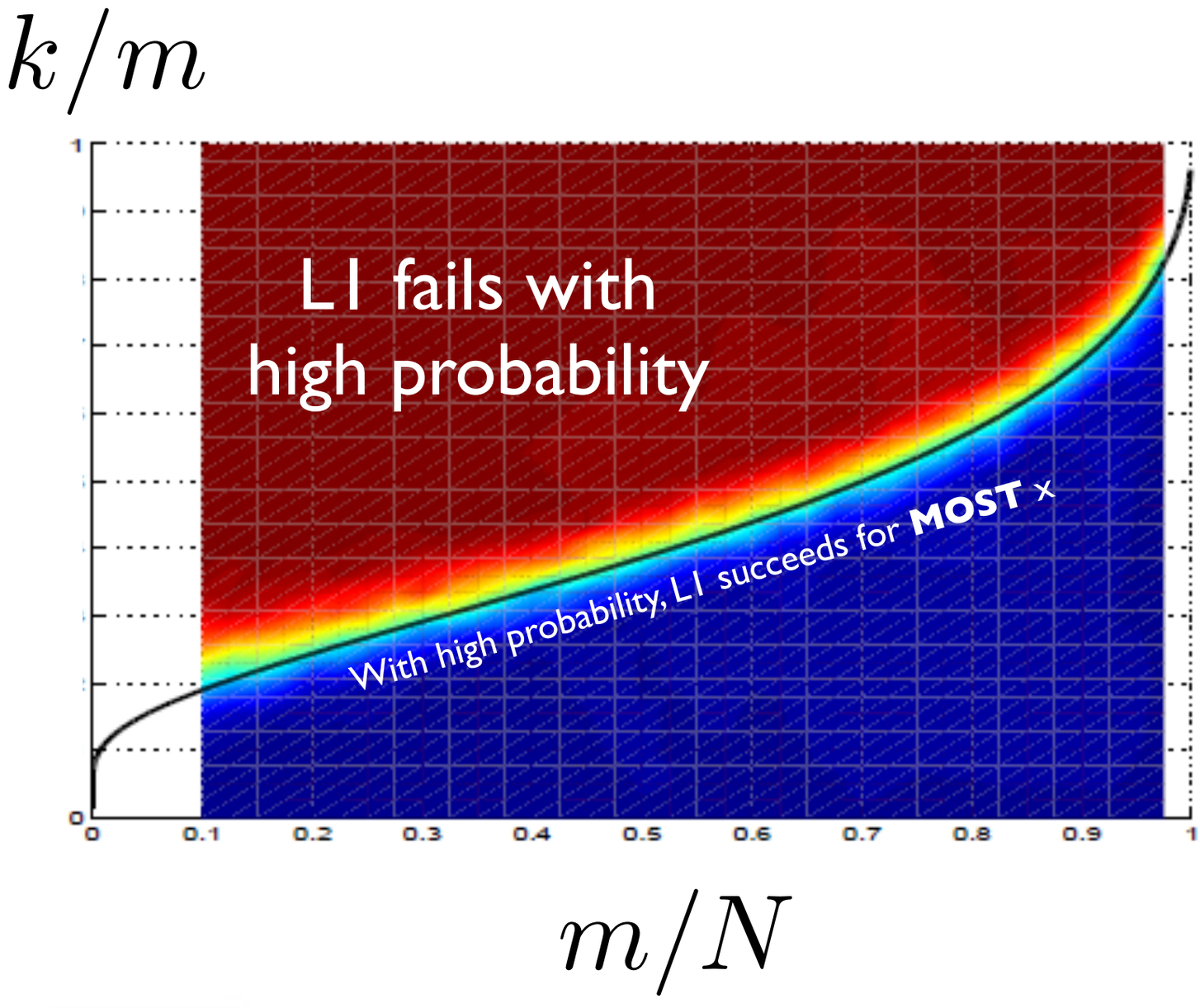
Dossal, Peyré, Fadili 2009:
heuristic algorithm to find worst-case

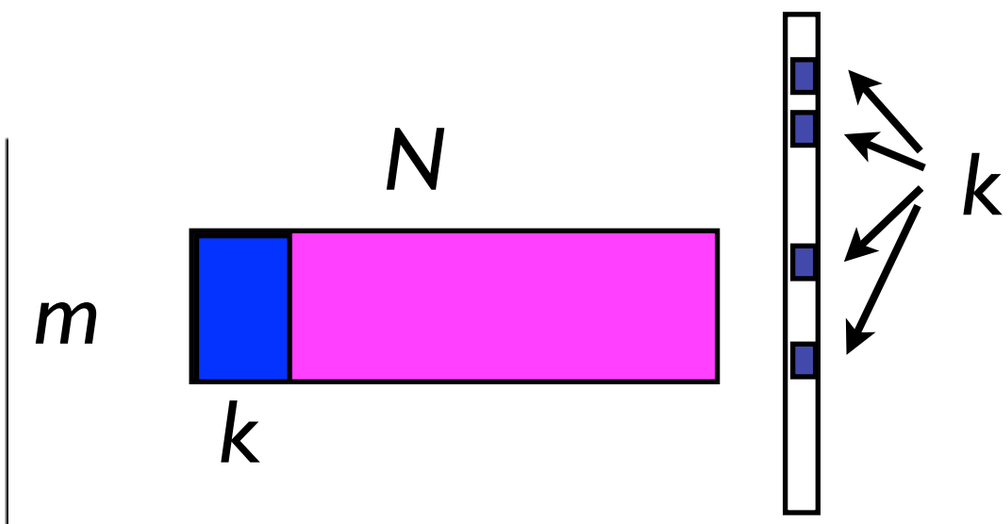
p=1 p=1/2 p=0

$\|x_0\|_0$

$k_1(\mathbf{A})$ $k_1(\mathbf{A})$ $k_{1/2}(\mathbf{A})$ $k_0(\mathbf{A})$

Phase transitions for Gaussian **A**





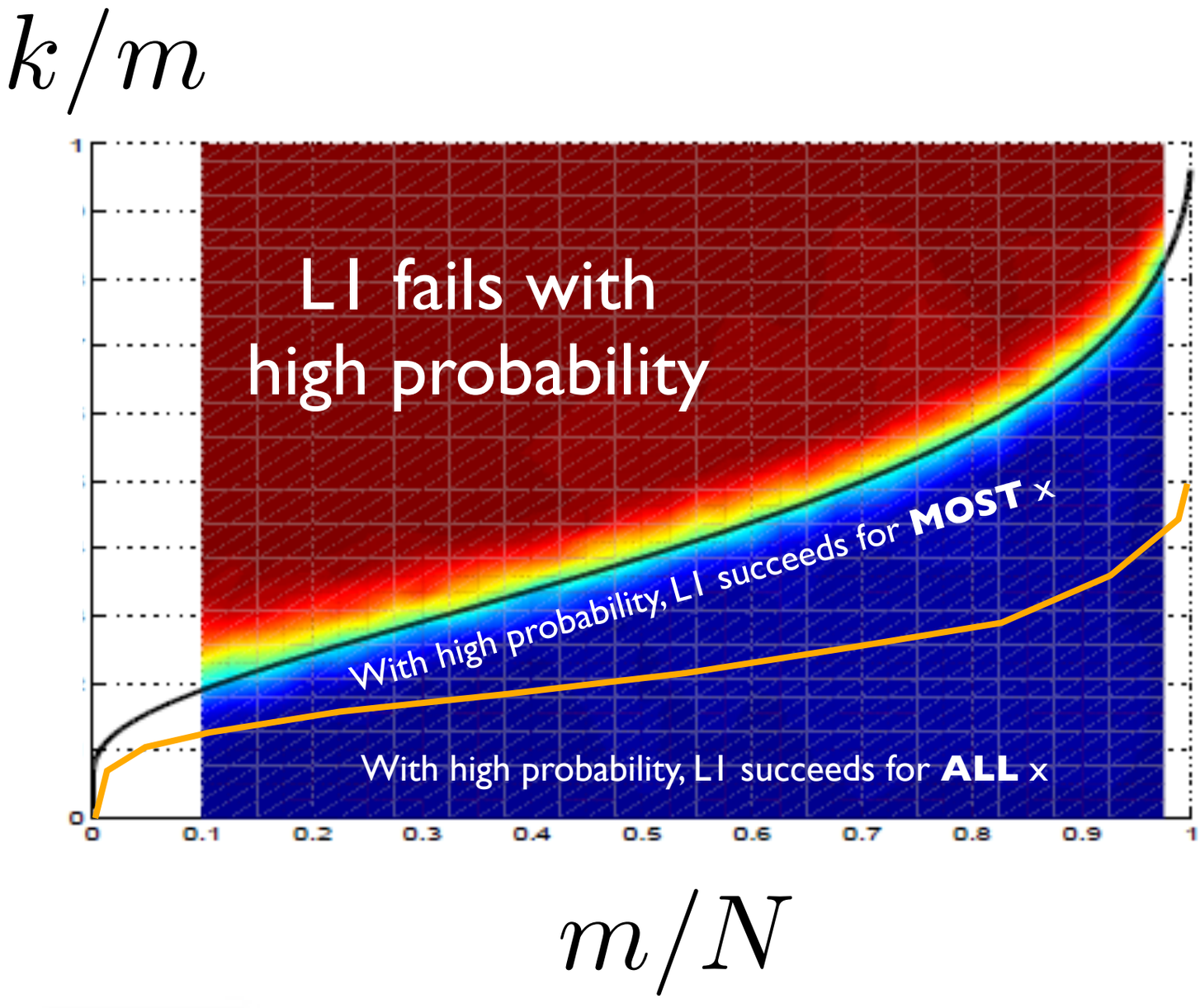
$$m \geq Ck \log N/k$$

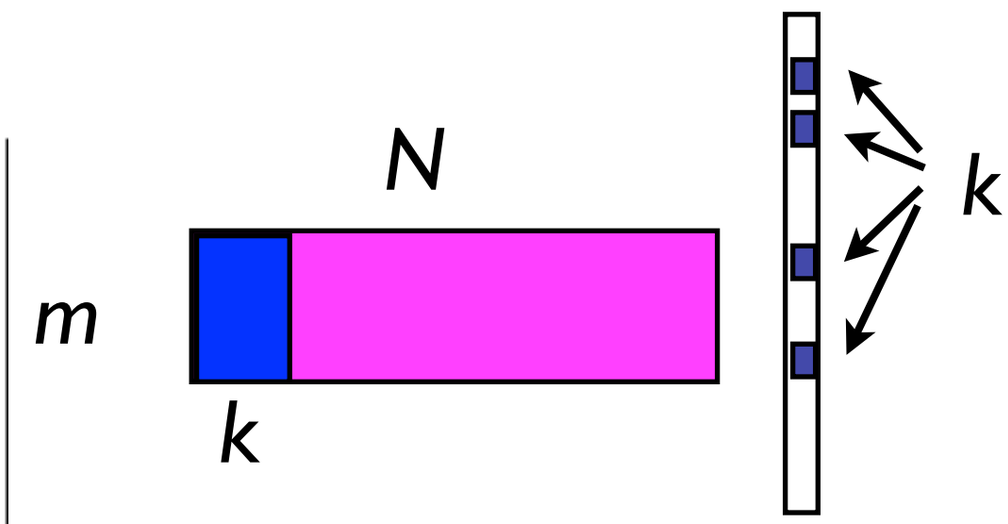
$$P(\delta_{2k} < \sqrt{2} - 1) \approx 1$$

$$k_1(\mathbf{A}) \approx \frac{m}{2e \log N/m}$$

[Donoho & Tanner 2009]

Phase transitions for Gaussian **A**





$$m \geq Ck \log N/k$$

$$P(\delta_{2k} < \sqrt{2} - 1) \approx 1$$

$$k_1(\mathbf{A}) \approx \frac{m}{2e \log N/m}$$

[Donoho & Tanner 2009]

Conclusions

- Sparsity: prior to solve **ill-posed inverse problems**
- If solution sufficiently sparse, **reasonable algorithms are guaranteed to find it** (even one step thresholding!).
- **Computational efficiency still a challenge**
  - ✓ problem sizes up to 1000 x 10000 already efficiently tractable.
- Theoretical guarantees are mostly worst-case
  - ✓ Empirical recovery goes far beyond but is not fully understood.
- Challenging practical issues include:
  - ✓ choosing / learning / designing dictionaries;
  - ✓ exploiting structures beyond sparsity;
  - ✓ designing feasible compressed sensing hardware.

Hot Topics, not covered in this tutorial

- Structured sparsity: group LASSO, etc.
- Combinatorial algorithms: submodular functions, etc.
- Approximate Message Passing algorithms
- Analysis vs synthesis sparsity
- Dictionary learning
- Low-rank matrices & sparsity

Thanks to

- F. Bimbot, S.Krstulovic, A. Ozerov, S. Lesage, B. Mailhé, S. Arberet, P. Sudhakar
- M. Nielsen, L. Borup (Aalborg Univ.)
- P. Vandergheynst, R. Figueras, P. Jost, K. Schnass (EPFL)
- M. Davies (U. Edinburgh), M. Elad (Technion), M. Plumbley (QMUL)
- H. Rauhut (U. Vienna)
- and several other collaborators ...

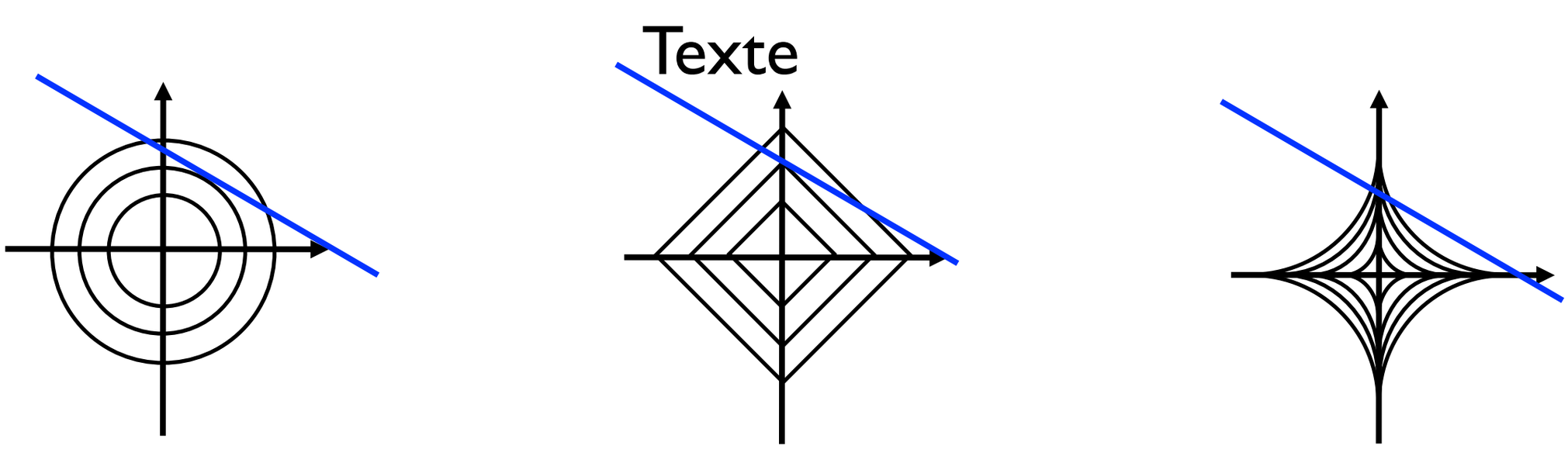
INVENTEURS DU MONDE NUMÉRIQUE

The end

remi.gribonval@inria.fr
www.irisa.fr/metiss/gribonval

L*p* "norms" level sets

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



Observation: *the minimizer is sparse*

$\{x \text{ s.t.} \mathbf{b} = \mathbf{A}x\}$

Sparsity of L*1* minimizers

- Real-valued case
  - ✓ **A** = an *m* x *N* real-valued matrix
  - ✓ **b** = an *m*-dimensional real-valued vector
  - ✓ *X* = set of all minimum L*1* norm solutions to $\mathbf{A}x = \mathbf{b}$

$$\tilde{x} \in X \Leftrightarrow \|\tilde{x}\|_1 = \min \|x\|_1 \text{ s.t. } \mathbf{A}x = \mathbf{b}$$

- **Fact 1**: *X* is convex and contains a “sparse” solution

$$\exists x_0 \in X, \|x_0\|_0 \leq m$$

- Proof : exercice!

Sparsity of L*1* minimizers

- Real-valued case
  - ✓ **A** = an *m* x *N* real-valued matrix
  - ✓ **b** = an *m*-dimensional real-valued vector
  - ✓ *X* = set of al solutions to regularization problem

$$\mathcal{L}(x) := \frac{1}{2}\|\mathbf{A}x - \mathbf{b}\|_2^2 + \lambda\|x\|_1$$

$$\tilde{x} \in X \Leftrightarrow \mathcal{L}(\tilde{x}) = \min_x \mathcal{L}(x)$$

- **Fact 2**: *X* is a convex set and contains a “sparse” solution

$$\exists x_0 \in X, \|x_0\|_0 \leq m$$

- Proof : exercice, using Fact 1!

Sparsity of L*1* minimizers

- A word of caution: this **does not hold true in the complex-valued case**
- Counter example: there is a construction where
  - ✓ **A** = a *2* x *3* complex-valued matrix
  - ✓ **b** = a *2*-dimensional complex-valued vector
  - ✓ the minimum L1 norm solution is unique and has 3 nonzero components

[E.Vincent, Complex Nonconvex Optimization l_p norm minimization for underdetermined source separation, Proc. ICA 2007.]

L1 vs Lp

Lp better than L1 (1)

- **Theorem** 2 [*G. Nielsen 2003*]
  - ✓ Assumption 1: **sub-additivity** of sparsity measures f, g

$$f(a+b) \leq f(a) + f(b), \forall a, b$$

  - ✓ Assumption 2: the function $t \mapsto \frac{f(t)}{g(t)}$ is **non-increasing** ↘
  - ✓ Conclusion: $k_g(\mathbf{A}) \leq k_f(\mathbf{A}), \forall \mathbf{A}$

Minimizing $\|x\|_f$ can recover vectors which are less sparse than required for guaranteed success when minimizing $\|x\|_g$

Lp better than L1 (2)

- **Example**
  - ✓ sparsity measures $f(t) = t^p,\ g(t) = t^q, 0 \leq p \leq q \leq 1$
  - ✓ sub-additivity

$$|a+b|^p \leq |a|^p + |b|^p, \forall a, b, 0 \leq p \leq 1$$

  - ✓ function $\frac{f(t)}{g(t)} = t^{p-q}$ is non-increasing
  - ✓ therefore

$$k_1(\mathbf{A}) \leq k_q(\mathbf{A}) \leq k_p(\mathbf{A}) \leq k_0(\mathbf{A}), \forall \mathbf{A}$$

Lp better than L1: proof

- 1) Since f/g non-decreasing:

$$z_1 \geq z_2 \geq 0 \quad \Rightarrow \quad \frac{f(z_1)}{g(z_1)} \leq \frac{f(z_2)}{g(z_2)}$$

- 2) Similarly

$$z_1 \geq \ldots \geq z_N \geq 0 \quad \Rightarrow \quad \frac{\|z_{1:k}\|_f}{\|z_{1:k}\|_g} \leq \frac{\|z_{k+1:N}\|_f}{\|z_{k+1:N}\|_g}$$

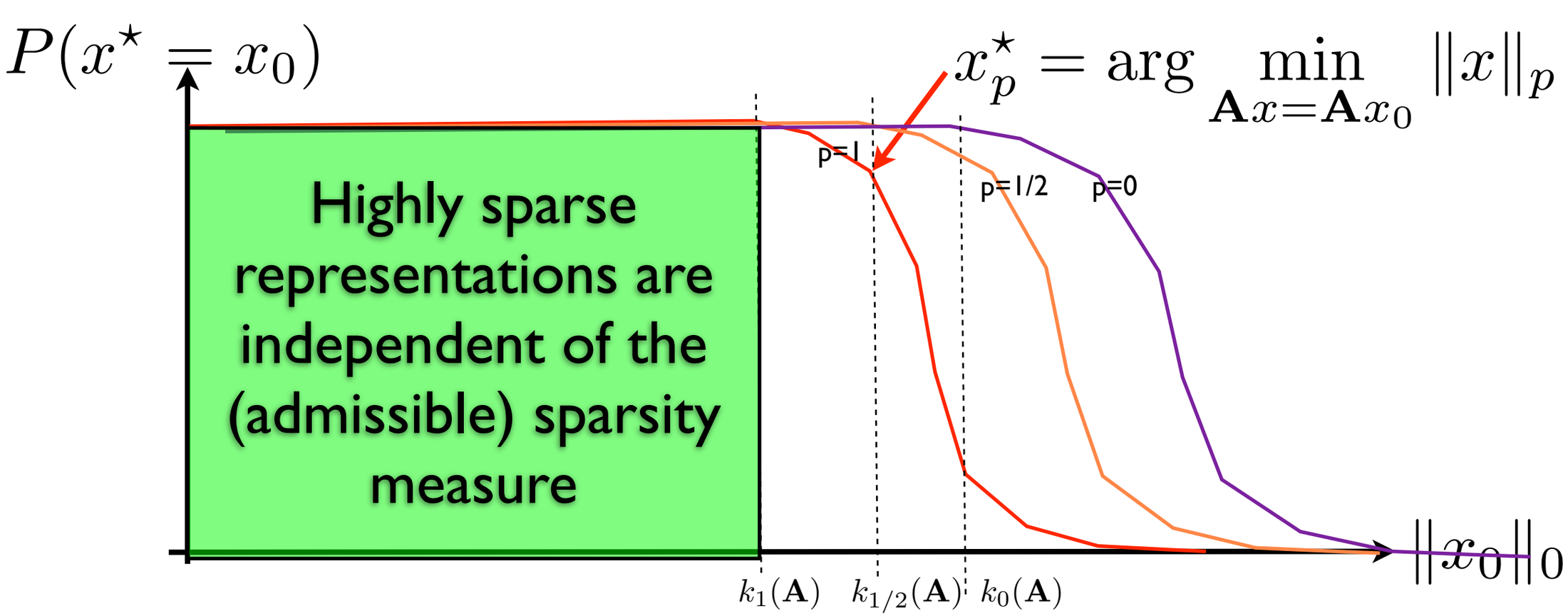
$$I_k = 1:k$$
$$I_k^c = k+1:N$$

$$\frac{\|z_{I_k}\|_f}{\|z_{I_k^c}\|_f} \leq \frac{\|z_{I_k}\|_g}{\|z_{I_k^c}\|_g}$$

- 3) Conclusion : if NSP(g,t,k) then NSP(f,t,k)

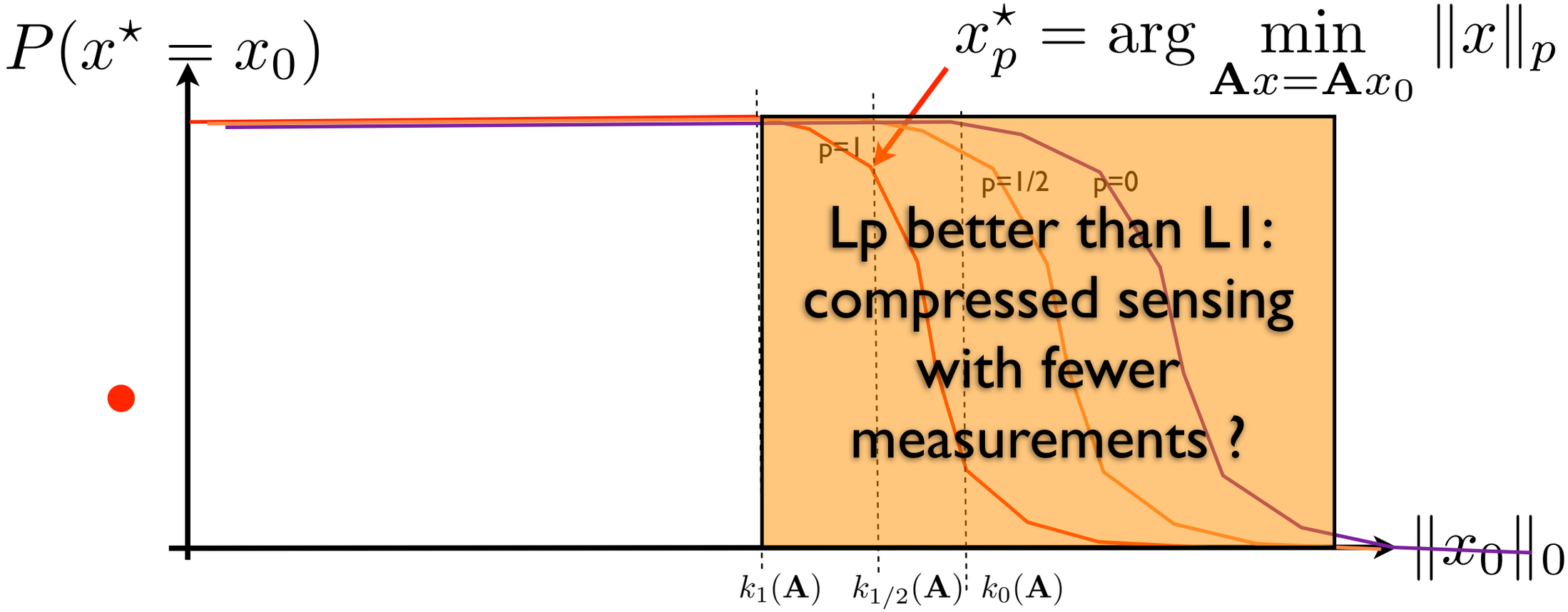
Lp better than L1 (3)

- At sparsity levels where L1 is guaranteed to "succeeds", all Lp p<=1 is also guaranteed to succeed



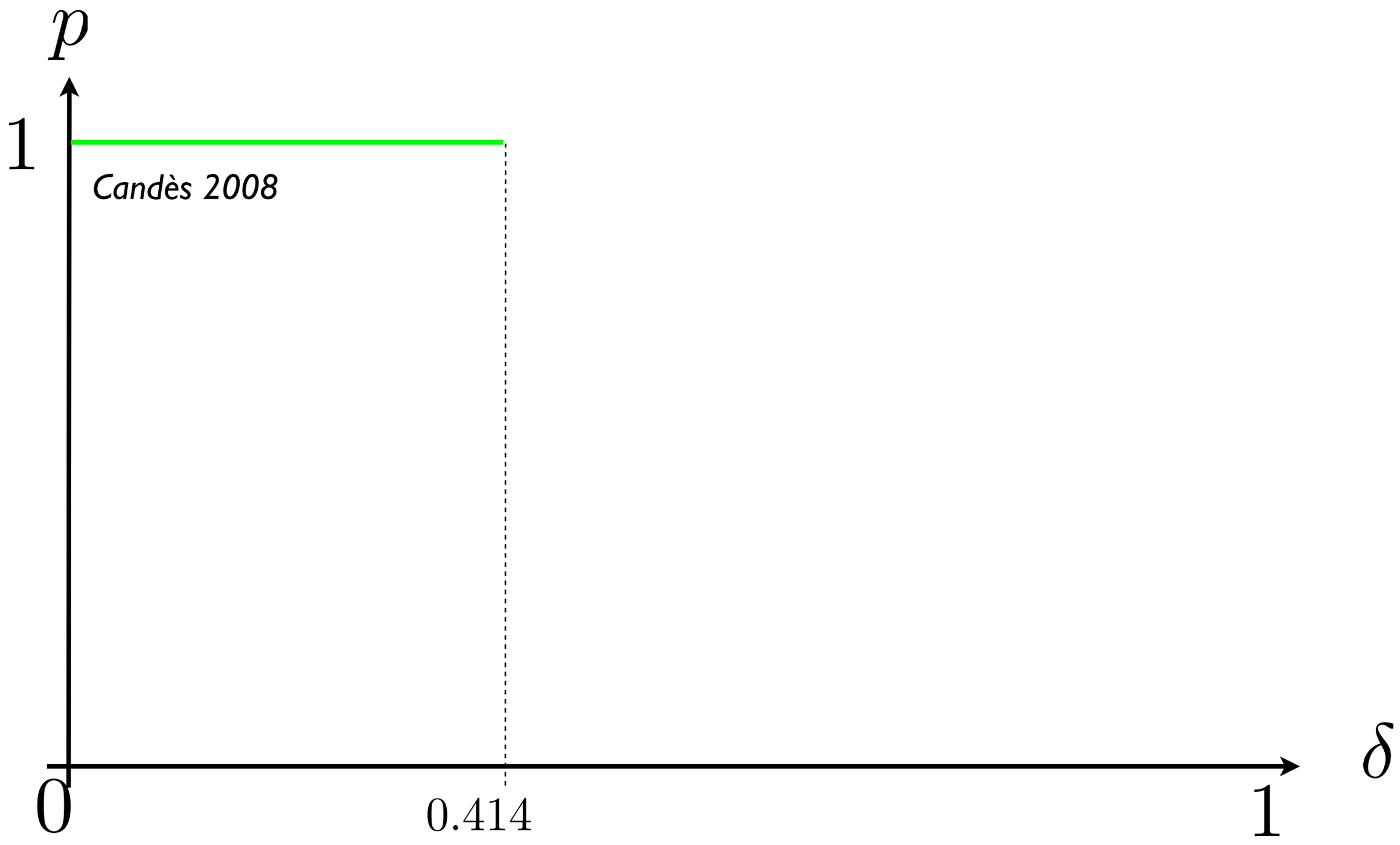
Lp better than L1 (4)

- + Lp p<1 can succeed where L1 fails
  - ✓ How much improvement ? Quantify $k_p(\mathbf{A})$?
- - Lp p<1 : nonconvex, has many local mimima
  - ✓ Better recovery with Lp *principle*, what about *algorithms ?*



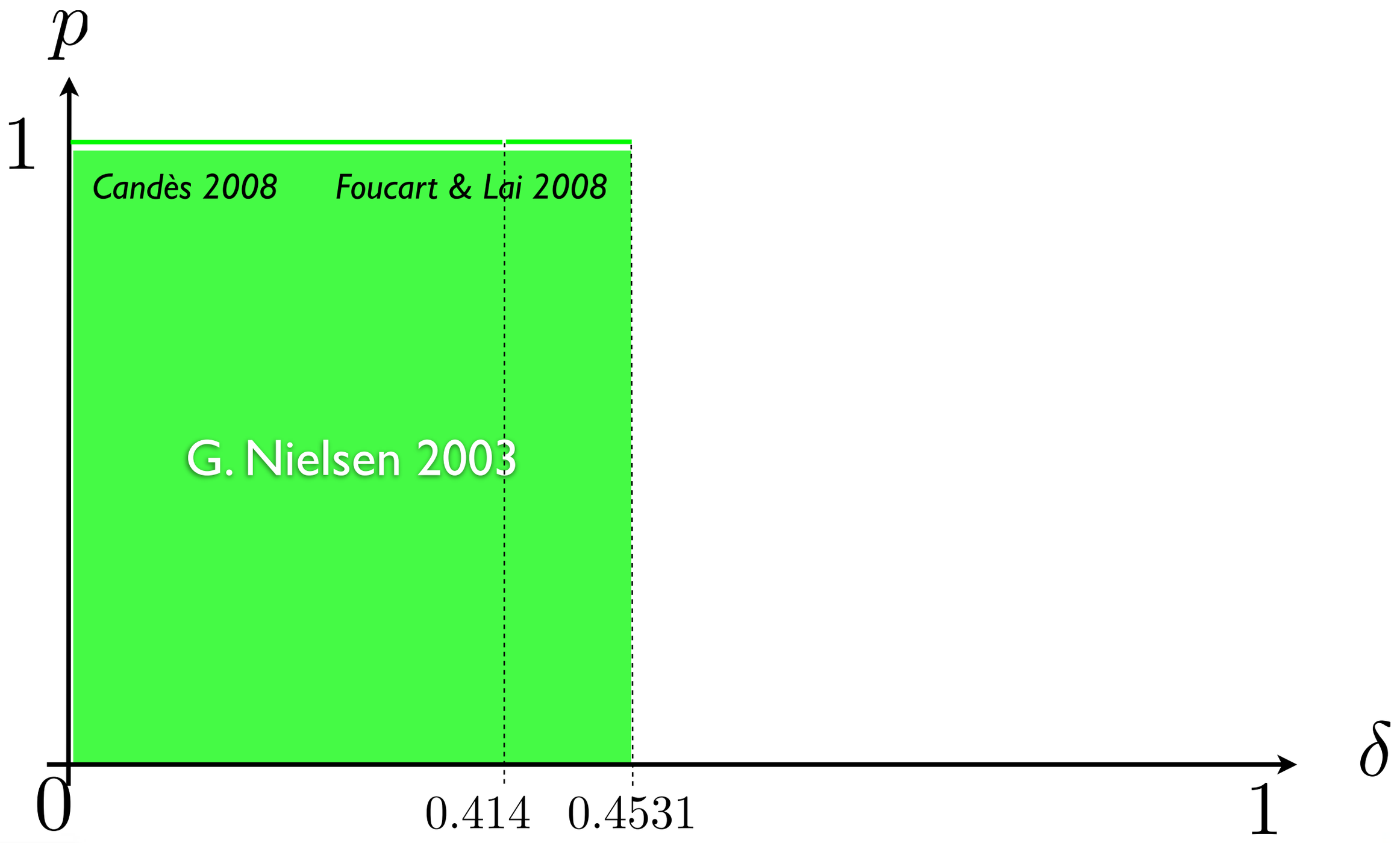
When does $\delta_{2k}(\mathbf{A}) < \delta$ imply $k \leq k_p(\mathbf{A})$?

When does $\delta_{2k}(\mathbf{A}) < \delta$ imply $k \leq k_p(\mathbf{A})$?

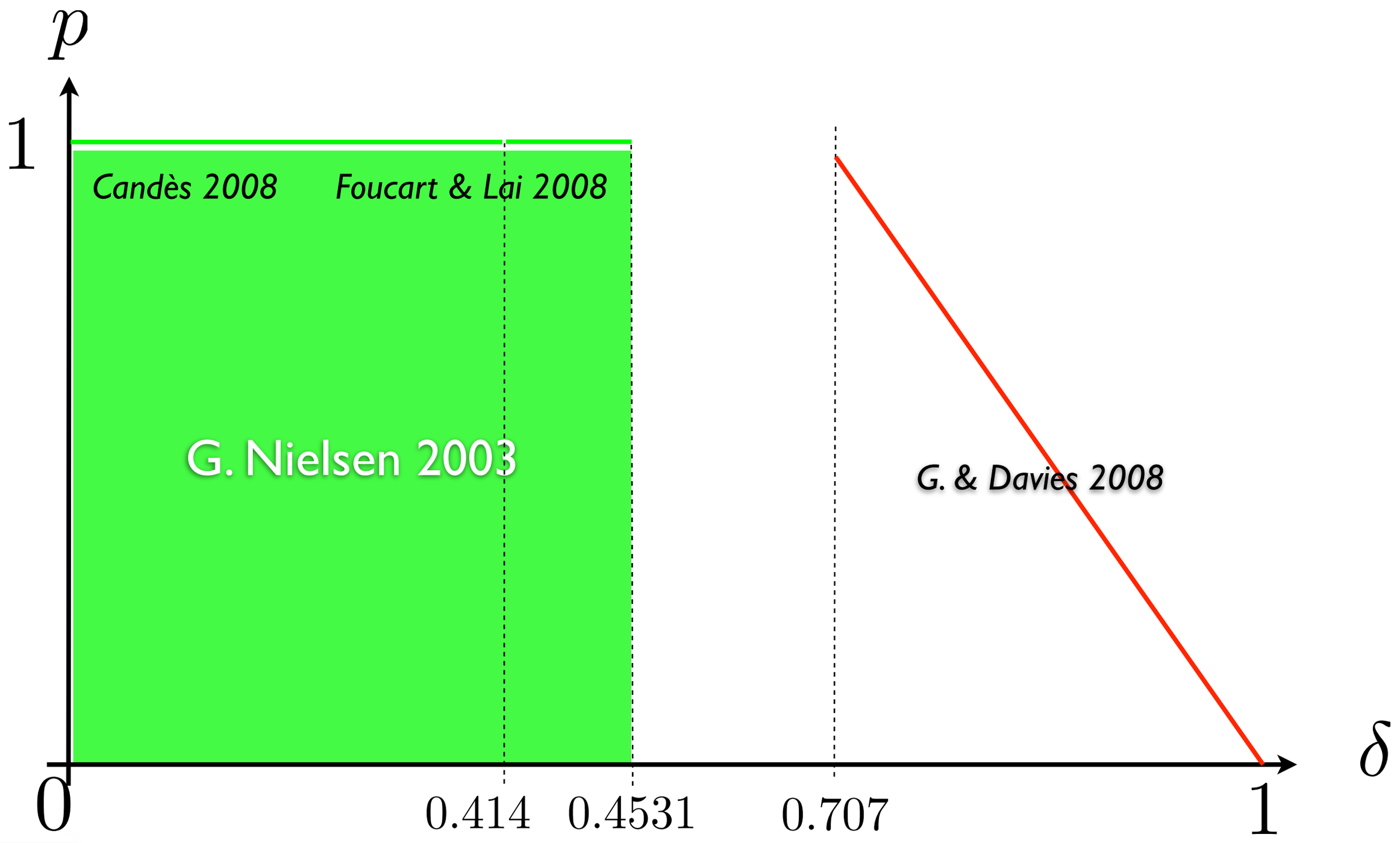


When does $\delta_{2k}(\mathbf{A}) < \delta$ imply $k \leq k_p(\mathbf{A})$?

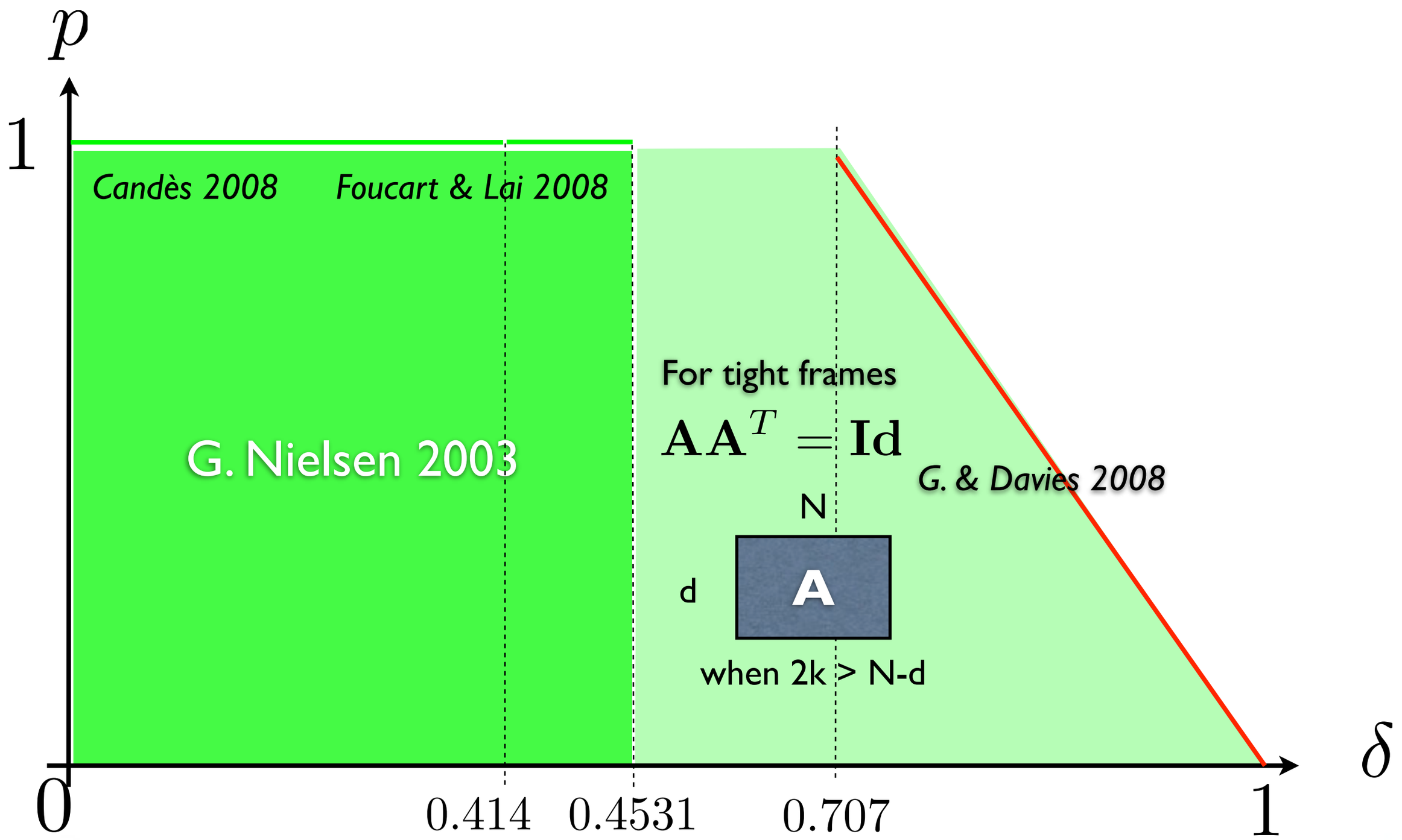
When does $\delta_{2k}(\mathbf{A}) < \delta$ imply $k \leq k_p(\mathbf{A})$?



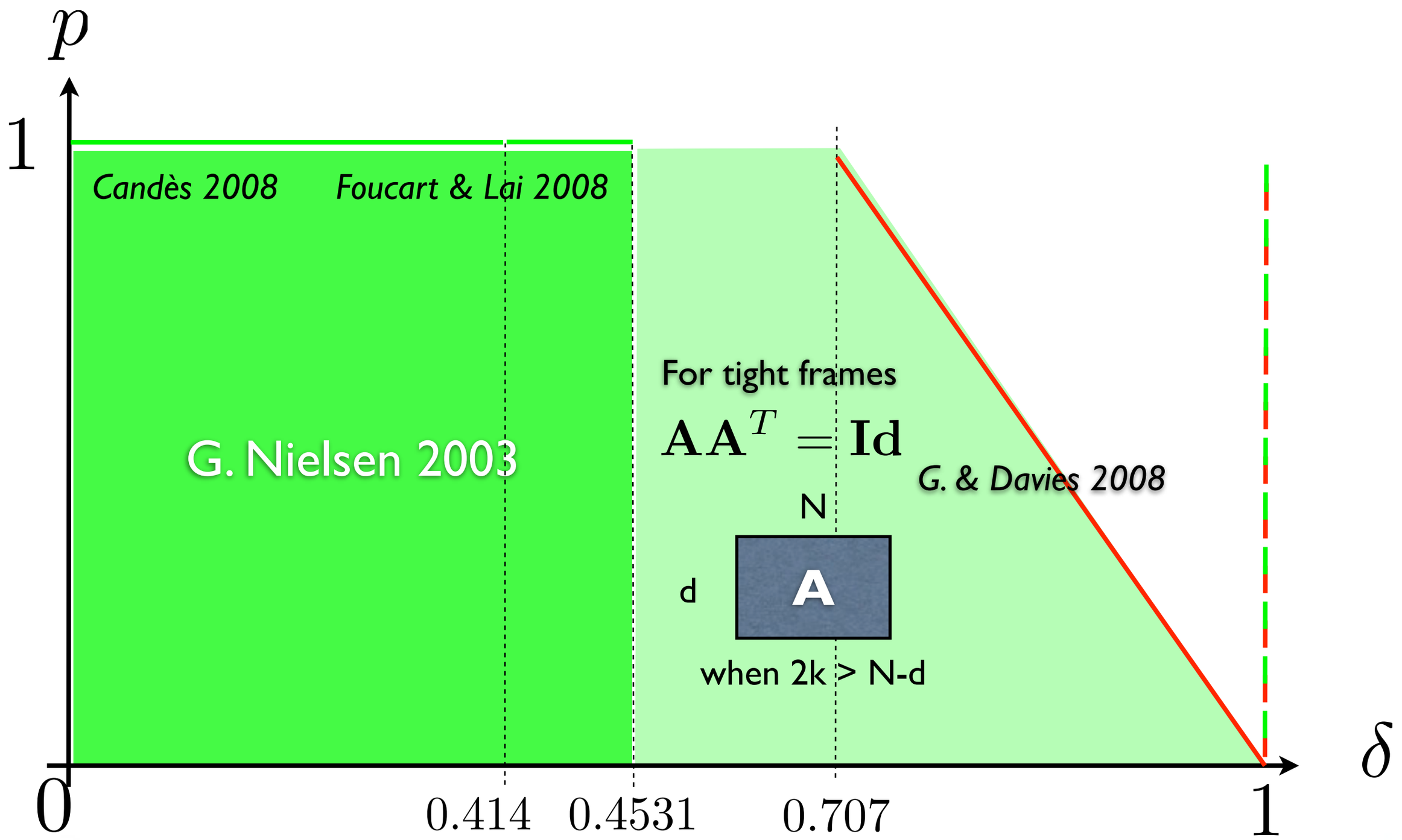
When does $\delta_{2k}(\mathbf{A}) < \delta$ imply $k \leq k_p(\mathbf{A})$?



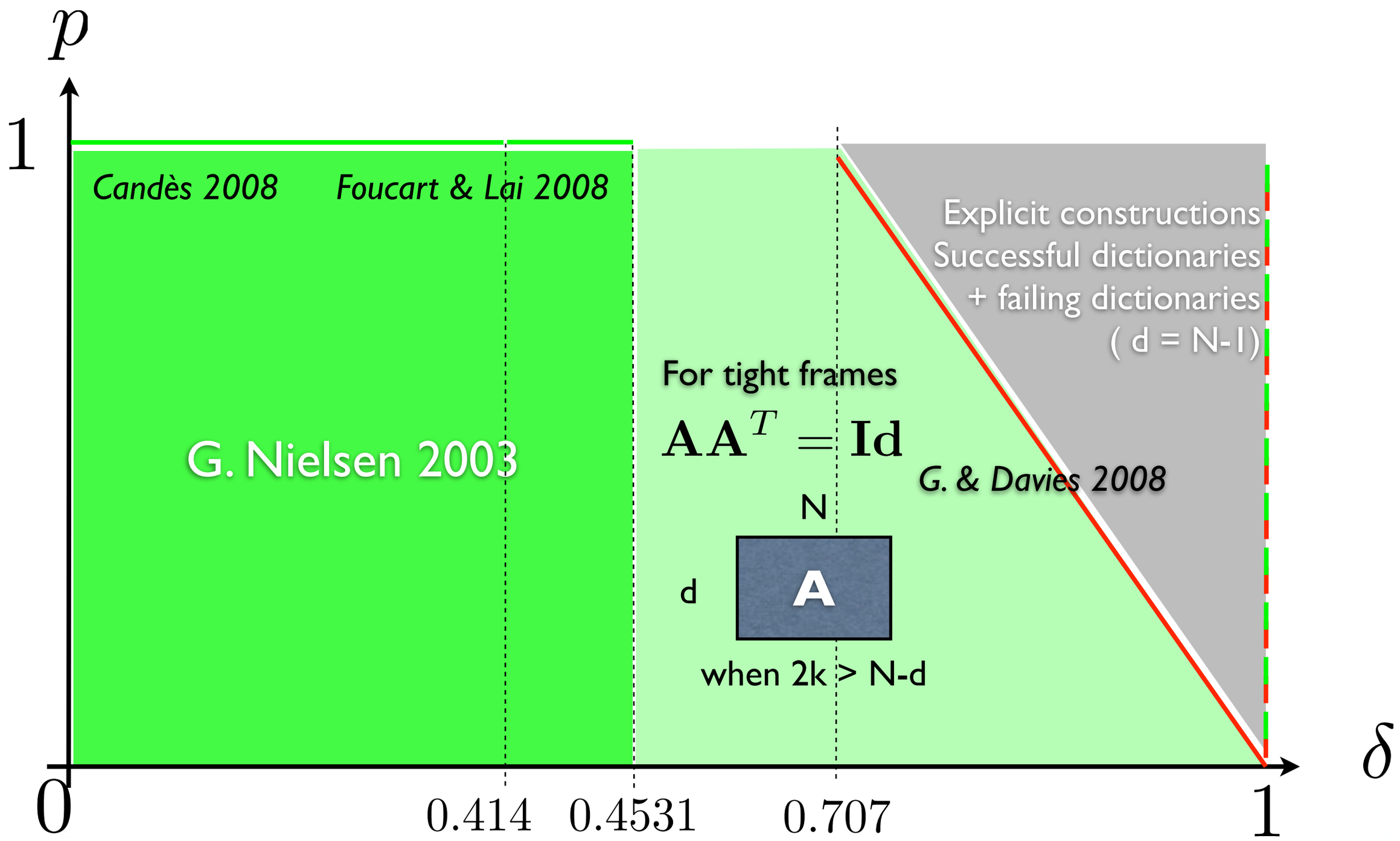
When does $\delta_{2k}(\mathbf{A}) < \delta$ imply $k \leq k_p(\mathbf{A})$?



When does $\delta_{2k}(\mathbf{A}) < \delta$ imply $k \leq k_p(\mathbf{A})$?



When does $\delta_{2k}(\mathbf{A}) < \delta$ imply $k \leq k_p(\mathbf{A})$?



L1 minimization fo Laplacian data ?

Bayesian modeling

- Observation : $\mathbf{b} = \mathbf{A}x + \mathbf{n}$
- Prior model $P(x_k) \propto \exp(-f(|x_k|)) \qquad \mathbf{n} \sim \mathcal{N}(0, \sigma^2 \mathbf{Id})$
- Probability vs «energy»

$$\max_x \prod_k P(x_k) \Leftrightarrow \min_x \sum_k f(|x_k|)$$

- L1 minimization «equivalent to MAP with Laplacian model»

$$\hat{P}(x_k) \propto \exp(-|x_k|)$$

Experiment: L1 minimization for Laplacian data ...

- Gaussian matrix

$$\mathbf{A} \in \mathbb{R}^{m \times N} \qquad N = 128 \qquad 1 \leq m \leq 100$$

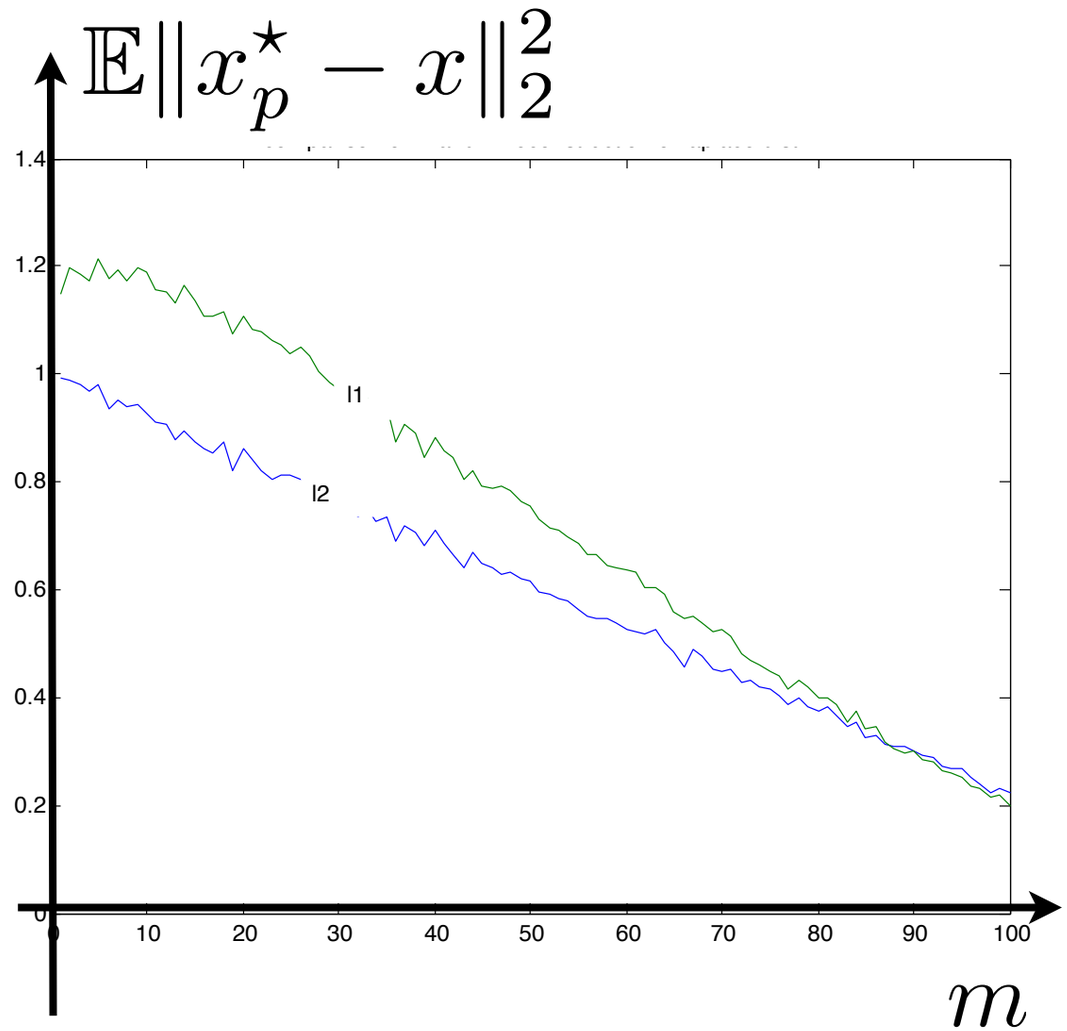
- Laplacian data, 500 draws

$$x \in \mathbb{R}^N \longrightarrow \mathbf{b} = \mathbf{A}x$$

- Reconstruction L1 or L2

$$x_p^\star := \arg\min \|x\|_p, \;\; p = 1, 2$$

= ML with Laplacian / Gaussian prior



cf also Seeger and Nickish, ICML 2008

MAP is bad when the model fits the data!
Mikolova 2007, Inverse Problems and Imaging

When to rely on sparse methods ?

- Laplacian distribution:
$$P(x_i) \quad \propto \quad \exp(-\lambda|x_i|)^{-1}$$

  - ✓ peak at the origin
  - ✓ not heavy tailed
  - ✓ x not well approximated by sparse vector

- Cauchy distribution:
$$P(x_i) \quad \propto \quad (1+\lambda x_i^2)^{-1}$$

  - ✓ no peak at the origin
  - ✓ heavy tailed (large values)
  - ✓ x very well approximated by sparse vector

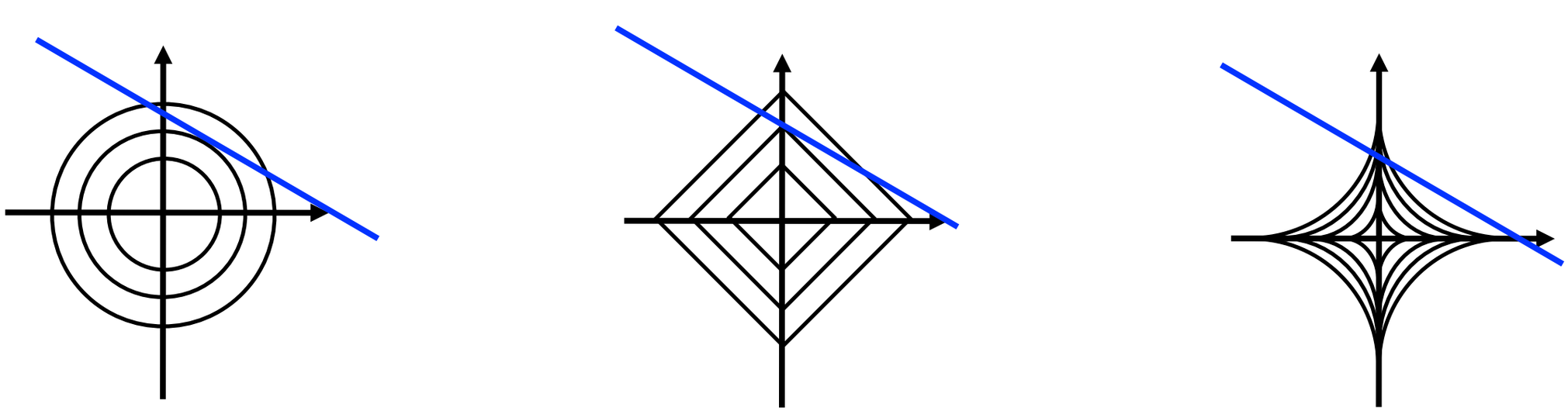
Bad match

Good match

Sparse methods always provide sparse estimates

L*p* "norms" level sets

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



Property: *the minimizer is sparse*

$\{x \text{ s.t.} \mathbf{b} = \mathbf{A}x\}$

Empirical observation : L*p* versus L1

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

reference — direct model — inverse problem

Typical observation *(e.g. Chartrand 2007)* +extrapolation