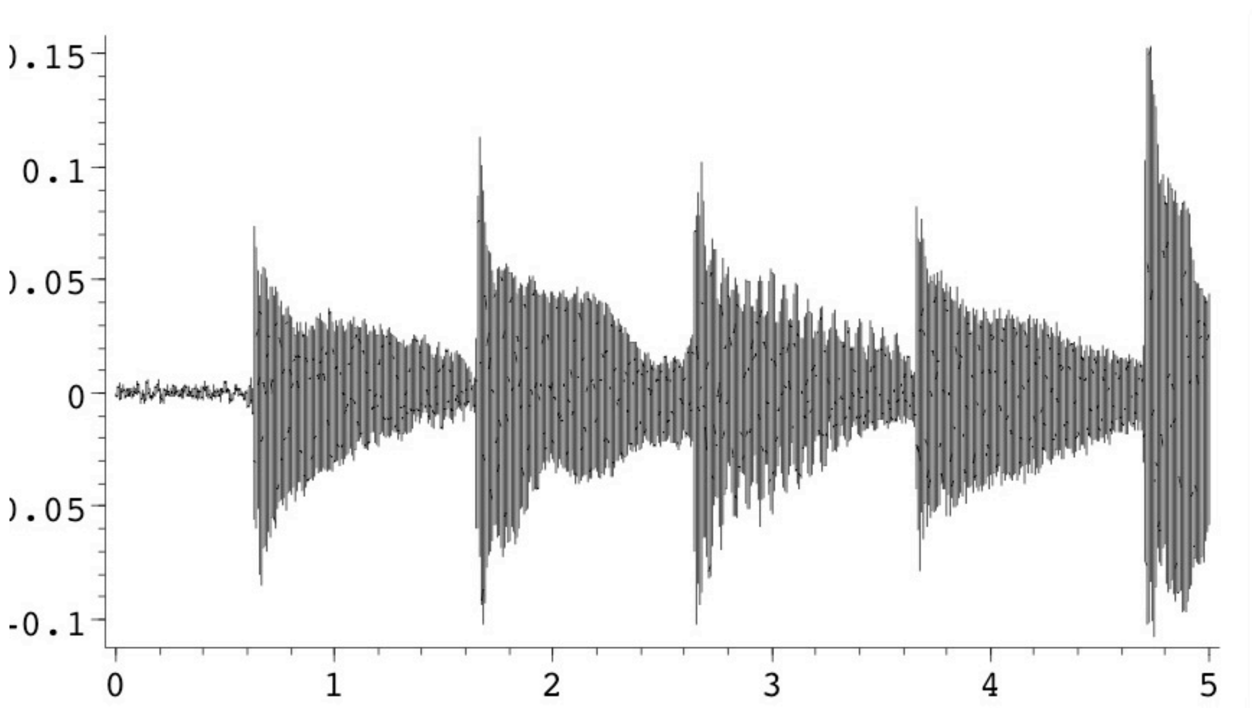
Sparse signal approximations

Alexandre Gramfort
alexandre.gramfort@telecom-paristech.fr

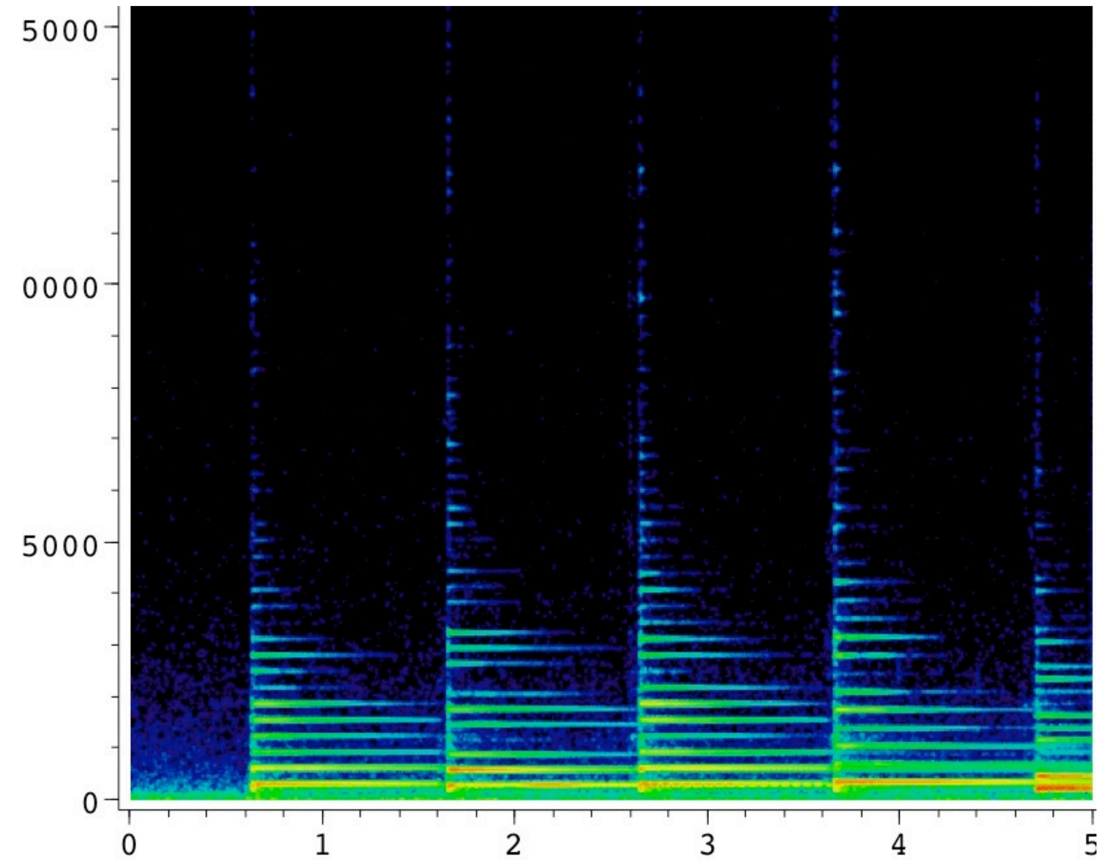
SIGMA 201b

Why sparsity?

Audio signal



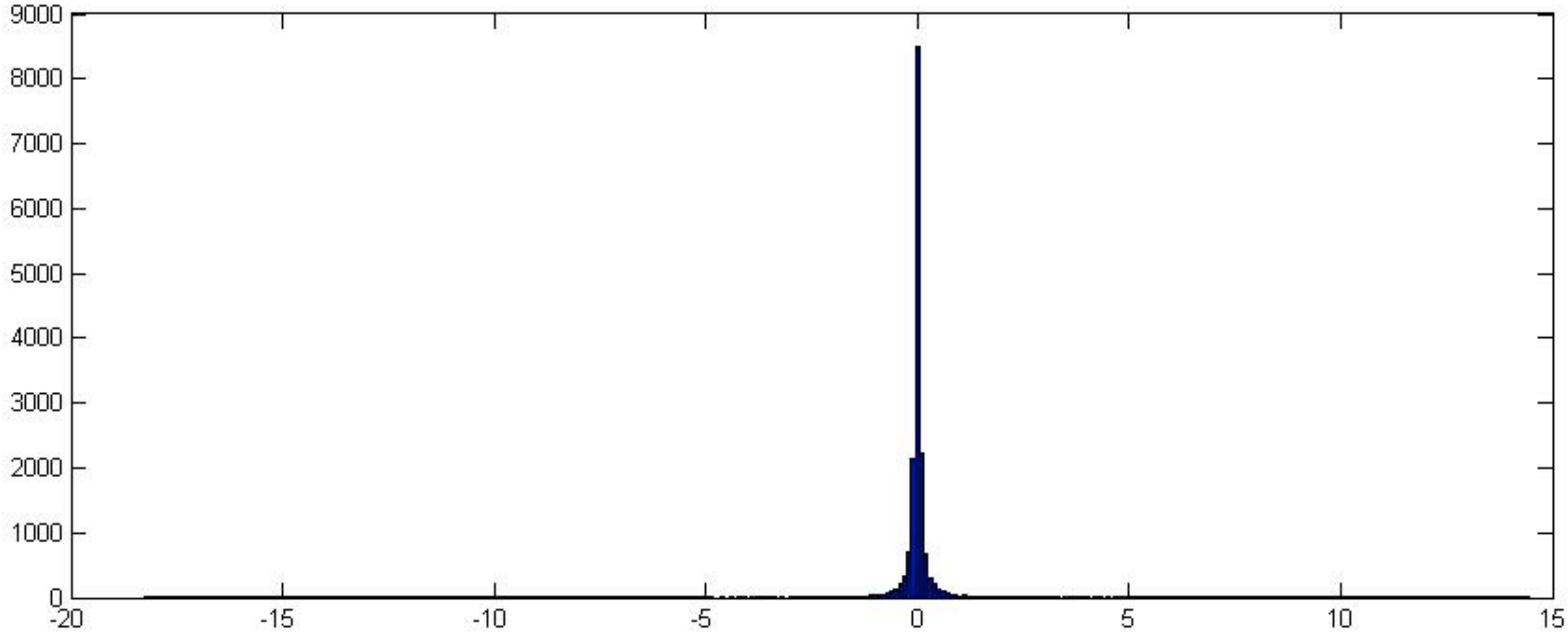
Its time-frequency representation (MDCT)



Black = zero

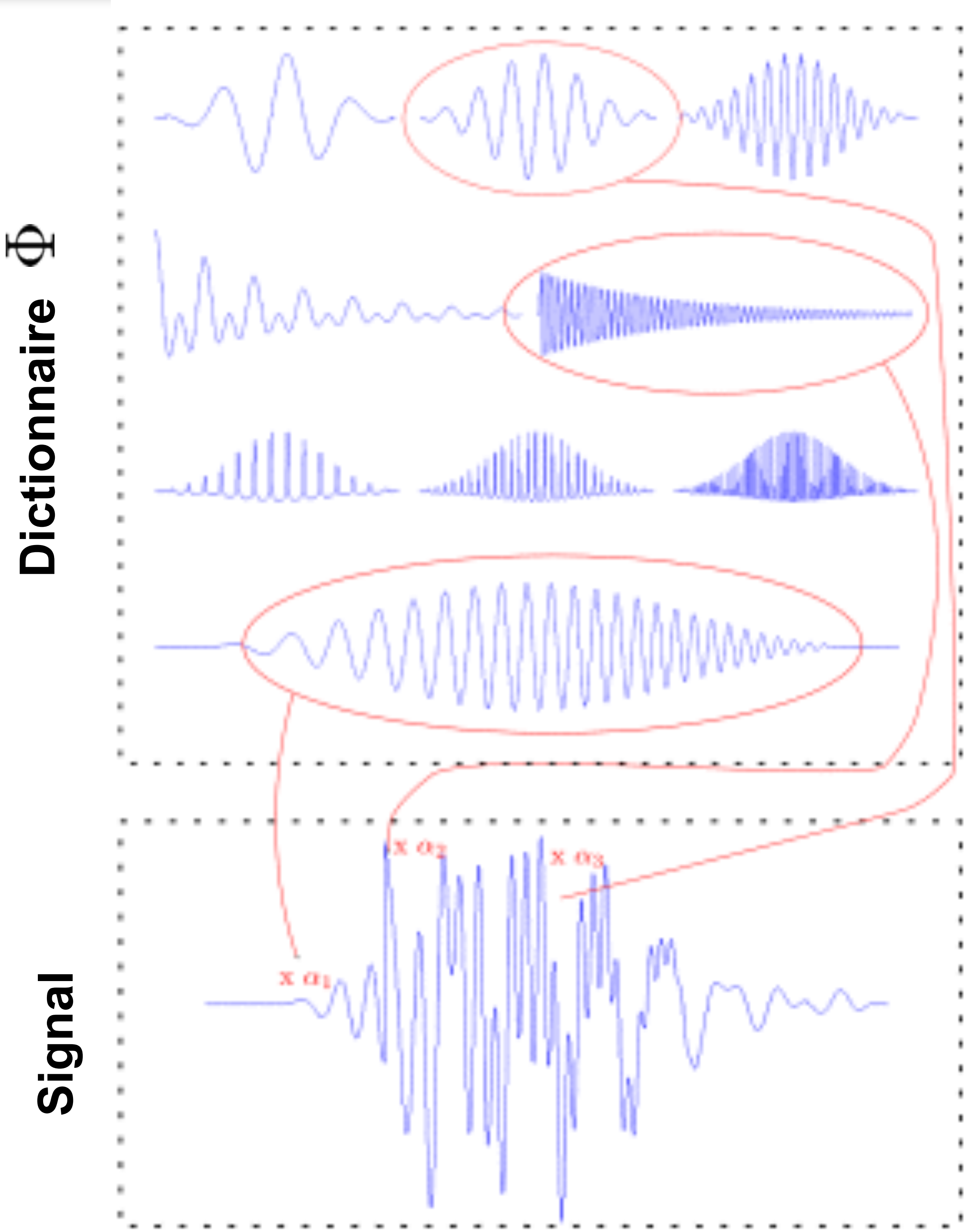
Why sparsity?

Histogram of MDCT (Modified Discrete Cosine Transform) coefficients of a musical sound



Most of the coefficients are 0 = Sparsity

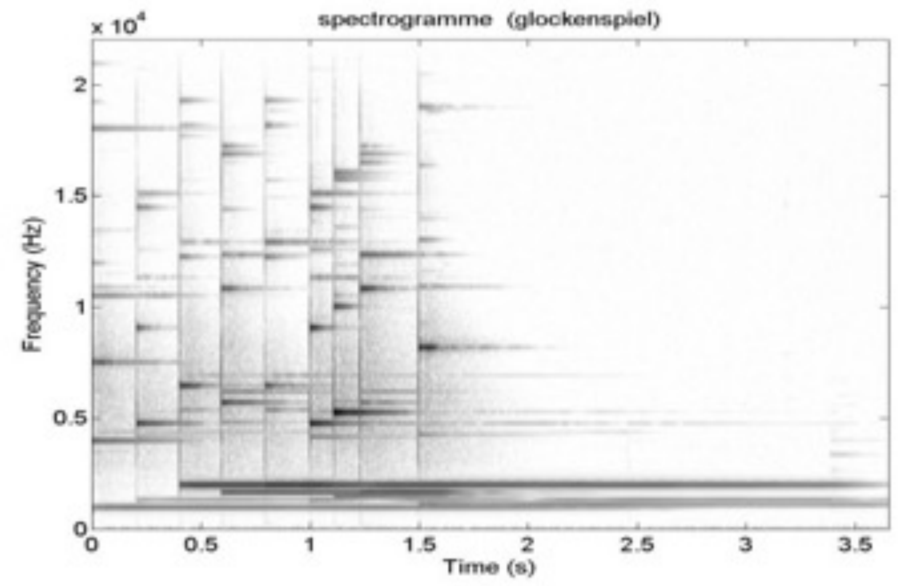
Dictionaries for signals



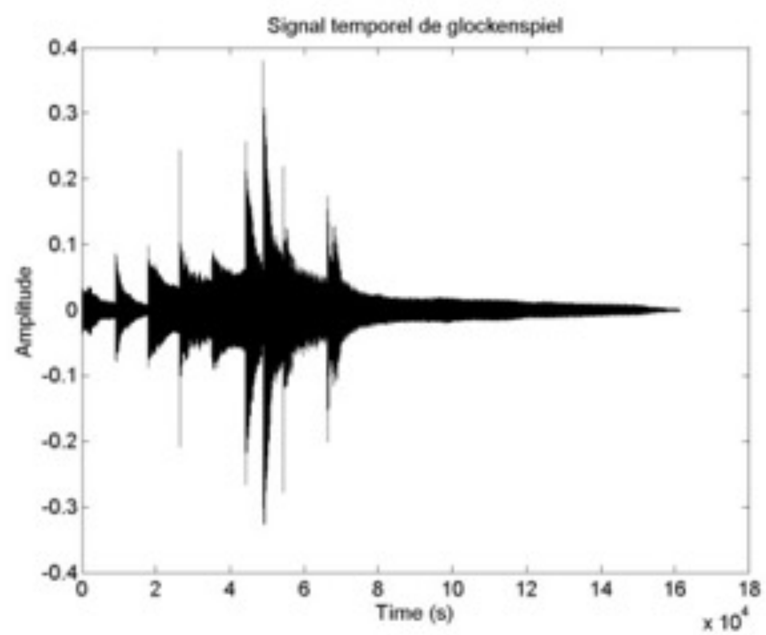
Multi scales dictionaries

- Spectrogram with Short Time Fourier Transform (STFT)

$$R_{TFCT}(x)[p,k] = \sum_{n=0}^{L-1} w[n].x[n-n_p].\exp(-2j\pi k\frac{n}{L})$$



$$x[n] = \frac{1}{P}\sum_{p=0}^{P} w^*[n-pL]\sum_{k=0}^{L-1} R_{TFCT}(x)[p,k].\exp(2j\pi k\frac{n}{L})$$



Remark: Complex values so we take the modulus

Representations for audio signals

- How to choose the window to obtain a perfect reconstruction?
  - One can show that the window should satisfy [Ravelli08, p12]:

$$
\begin{aligned}
w_0[u] &= 1 \\
w_p^2[u+\frac{L}{2}] + w_{p+1}^2[u] &= 1, \quad p \in [0..P-2] \\
w_{P-1}[u+\frac{L}{2}] &= 1
\end{aligned}
$$

  - One of the most popular window is the sinusoidal window:

$$
w[u] = sin[\frac{\pi}{L}(u+\frac{1}{2})]
$$

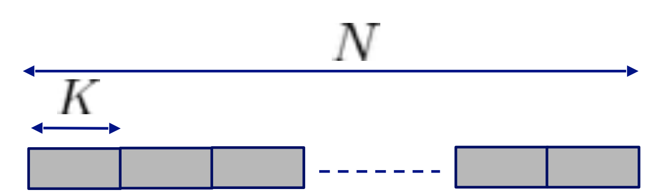
Representations for audio signals

- MDCT: Modified Discrete Cosine Transform
  - Based discrete cosine transform of type IV

$$R_{DCT}[k] = \sqrt{\frac{2}{M}} \sum_{n=0}^{L-1} x[n] \cos\left[(\frac{1}{2} + n).(\frac{1}{2} + k).\frac{\pi}{L} \right]$$

  - Perfect reconstruction

- In matrix form
  - Given x with a vector of size $N = PK$



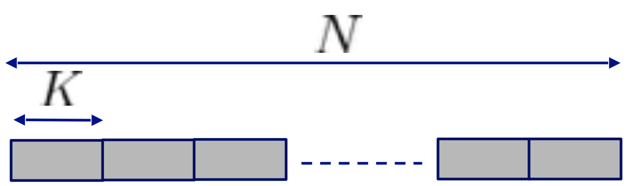
  - MDCT of size $L = 2K$ can be written as a transformation matrix T of size $N \times N$

Remark: Real values

Remark: the MDCT is employed in most modern lossy audio formats, including MP3, AC-3, Vorbis, Windows Media Audio, AAC etc.

MDCT : Matrix form

- In matrix form



  - Given x with a vector of size $N = PK$
  - MDCT of size $L = 2K$ can be written as a transformation matrix T of size $N \times N$

$$\mathbf{T}[n, pK + k] = \phi_{p,k}[n] \quad \text{pour} \quad p \in [0..P-1], k \in [0..K-1] \quad \text{et} \quad n \in [0..N-1]$$

- with $$\phi_{p,k}[n] = w_p[u]\sqrt{\frac{2}{K}} \cos\left[\frac{\pi}{K}\left(u + \frac{K+1}{2}\right)\left(k + \frac{1}{2}\right)\right]$$

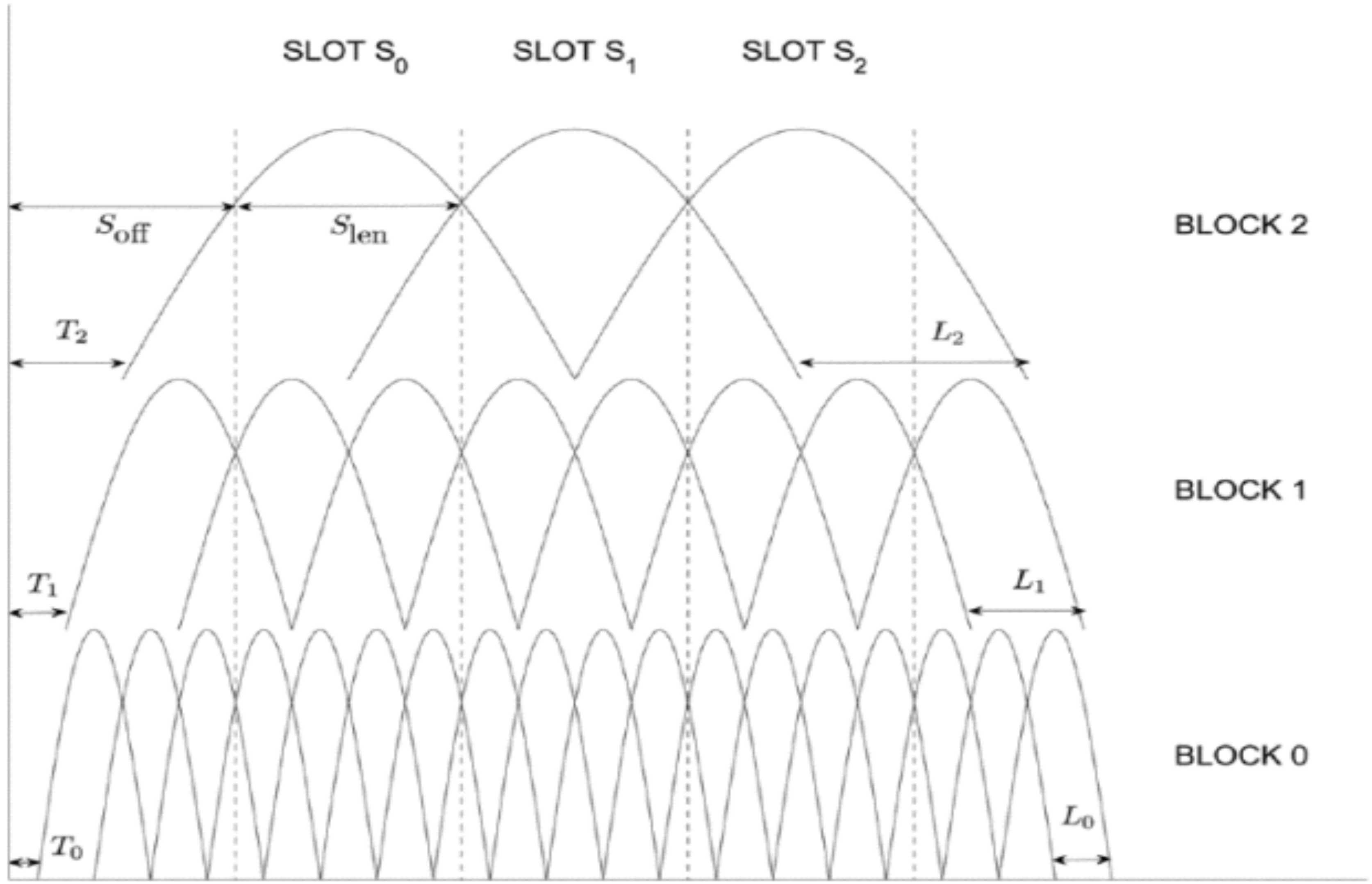
$$u = n - (p - \frac{1}{2})K$$

- Inverse condition for perfect reconstruction is given by the orthogonality of the matrix T:

$$\mathbf{T}\mathbf{T}^T = \mathbf{I}$$

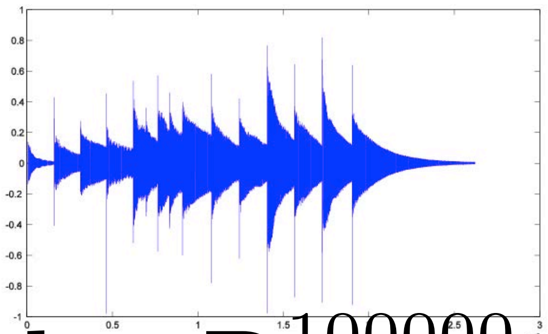
Union of MDCT basis

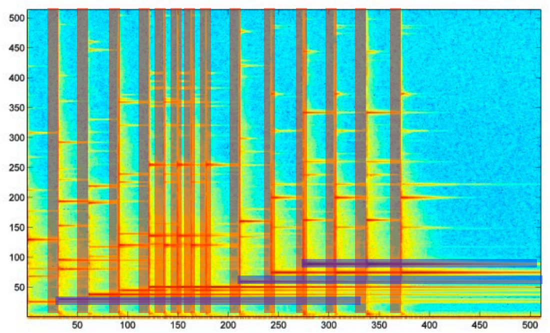
- Construct redundant representations: Union of MDCT dictionaries [E. Ravelli et al. 2008]

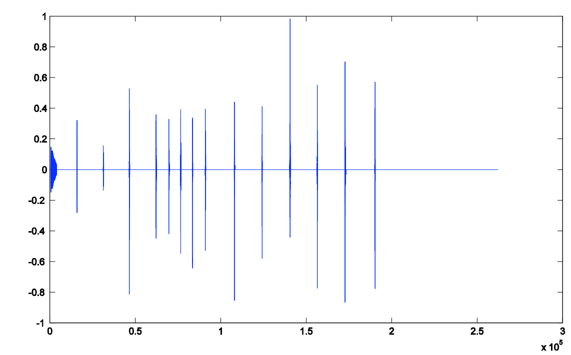


Union of MDCT basis

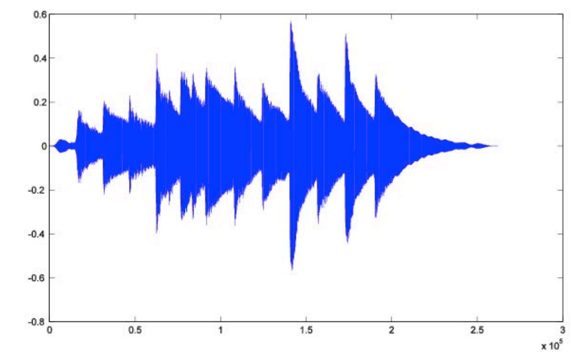
- Audio = superimposition of structures
- Example : glockenspiel







$\boldsymbol{\Phi}_1 x_1$



$\boldsymbol{\Phi}_2 x_2$

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

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

- Two-layer sparse model with Gabor atoms

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

$$\mathbf{b} \approx \boldsymbol{\Phi}_1 x_1 + \boldsymbol{\Phi}_2 x_2$$

Demo

sparsity on images

(a) (b)

Sparse representation of an image via a multiscale wavelet transform. (a) Original image. (b) Wavelet representation. Large coefficients are represented by light pixels, while small coefficients are represented by dark pixels. Observe that most of the wavelet coefficients are close to zero.

sparsity on images

(a) (b)

Sparse approximation of a natural image. (a) Original image.
(b) Approximation of image obtained by keeping only the largest 10% of the wavelet coefficients.

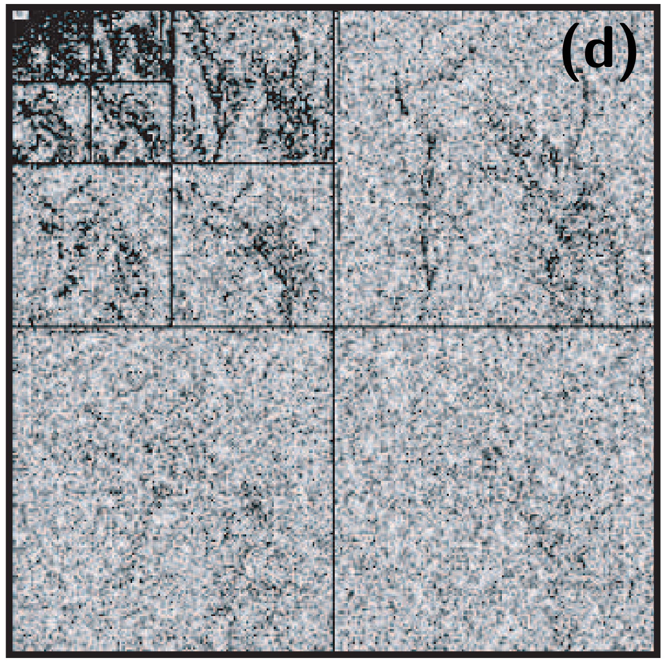
sparsity on images

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

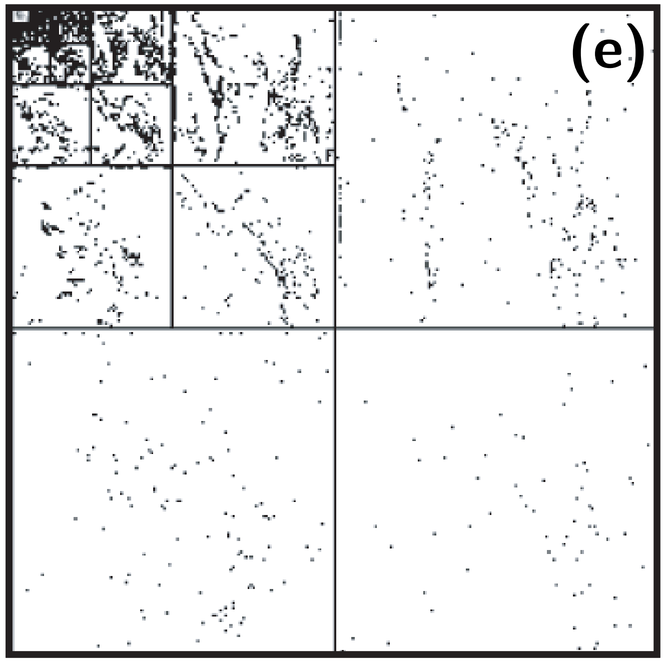
Original

Noisy

Smoothed



Coefficients

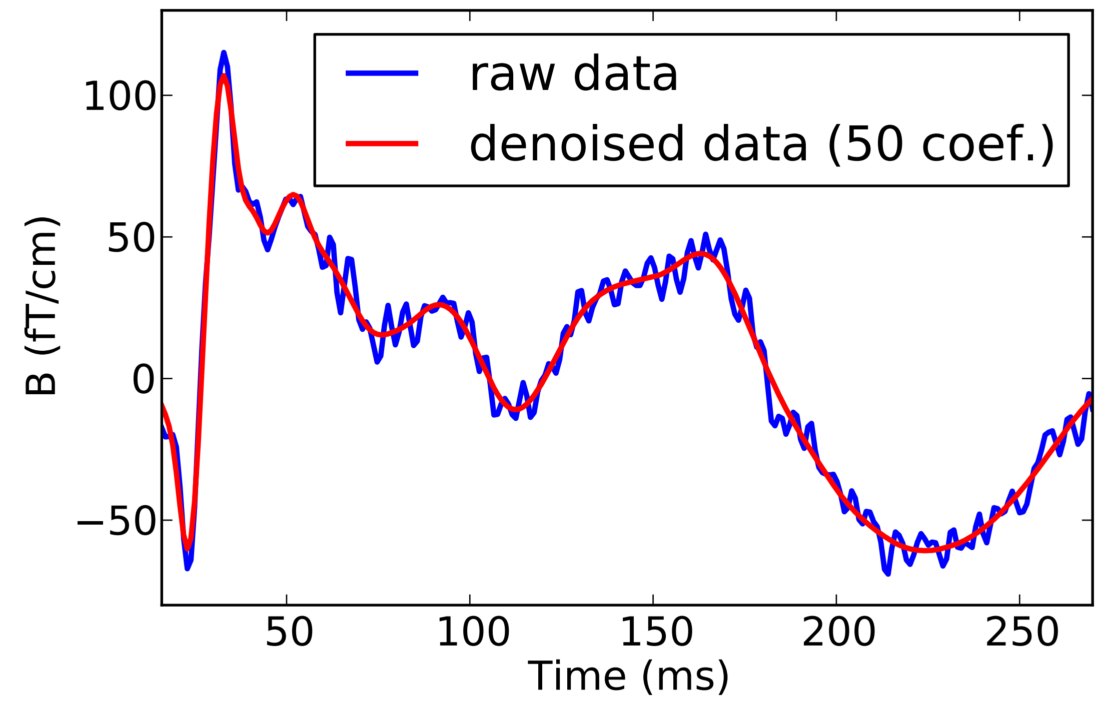


Thresholded coefficients

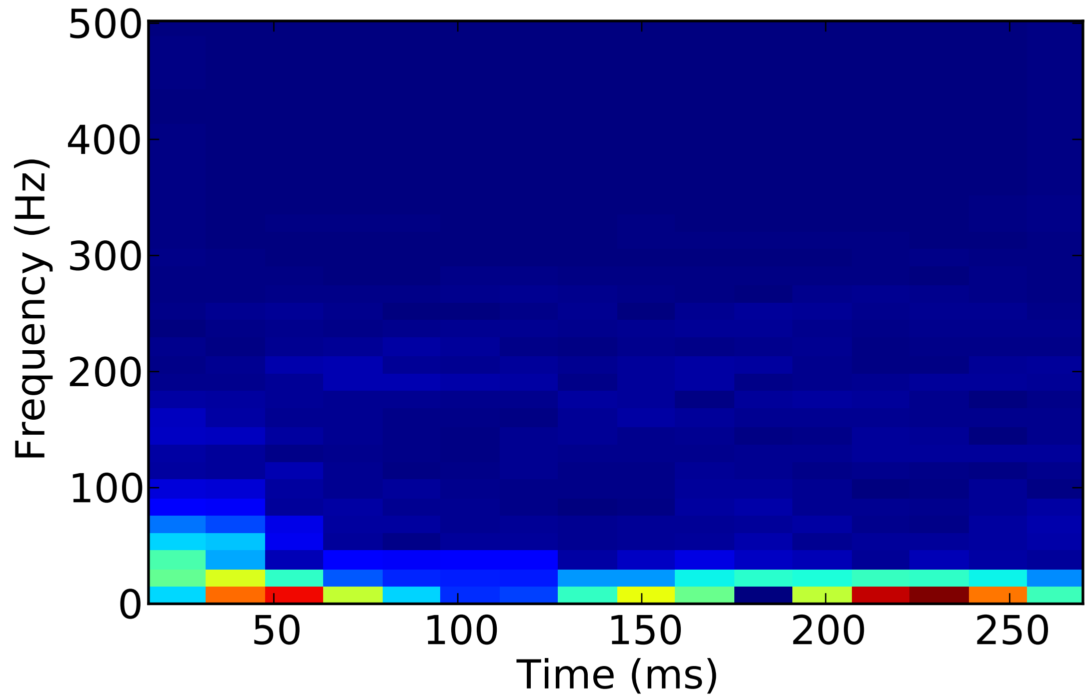
Denoised

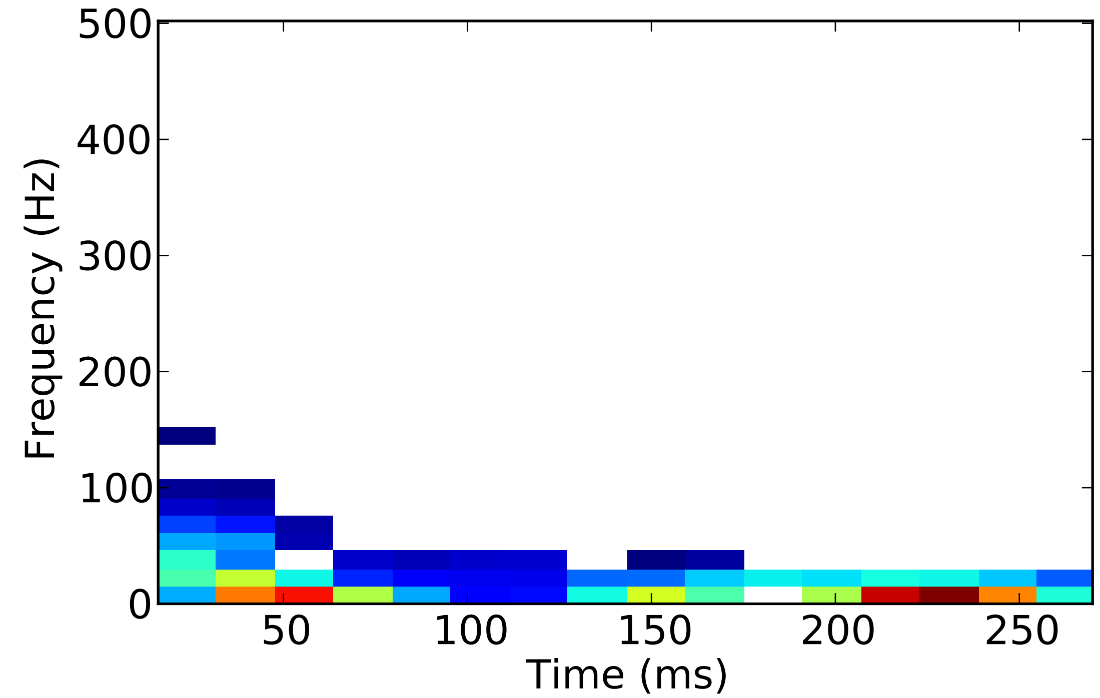
sparsity on neuroscience signals

Example of MEG data



STFT





Outline

- Sparse approximations why and how?
- Sparse decomposition with *greedy* algorithms
- Sparse decomposition with *convex* algorithms
- Some theory of exact recovery

Lp 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

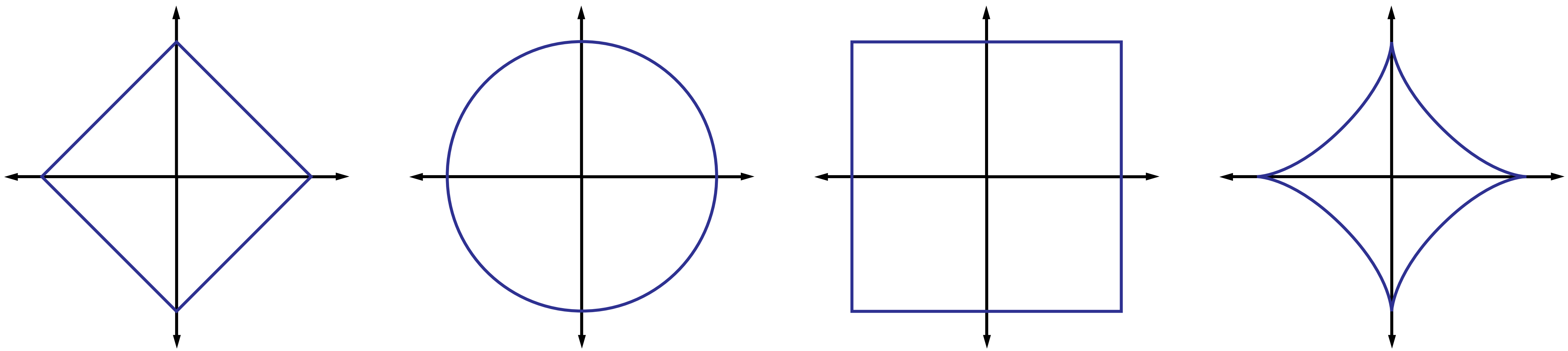
$$\|x + y\|_p \leq 2^{1/p} \big(\|x\|_p + \|y\|_p \big), \forall x, y$$

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

- "*Pseudo*"-norm for p=0

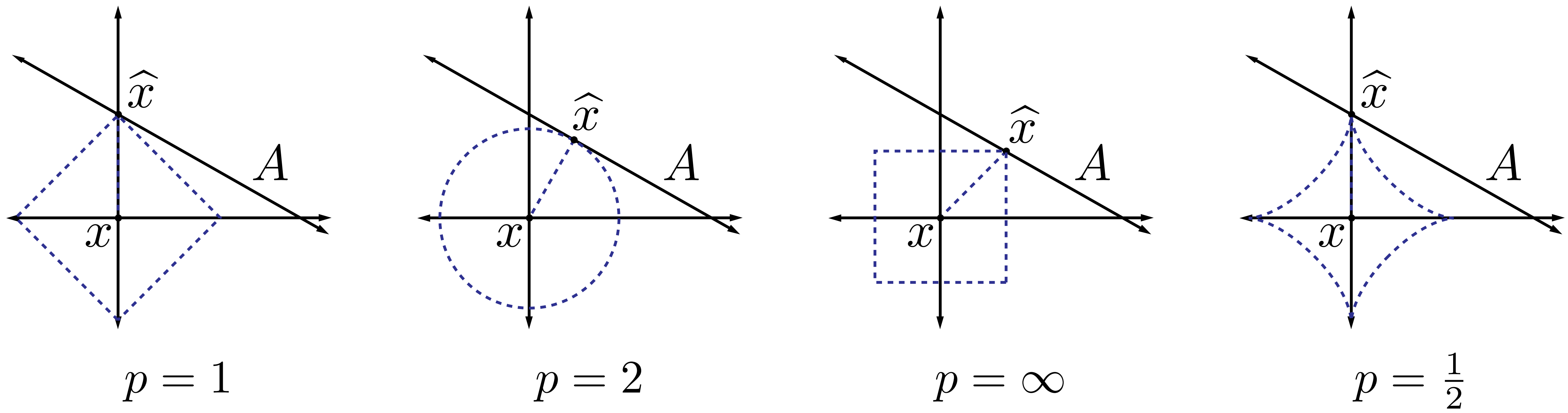
$$\|x + y\|_0 \leq \|x\|_0 + \|y\|_0, \forall x, y$$

Lp Norms & Quasi-Norms

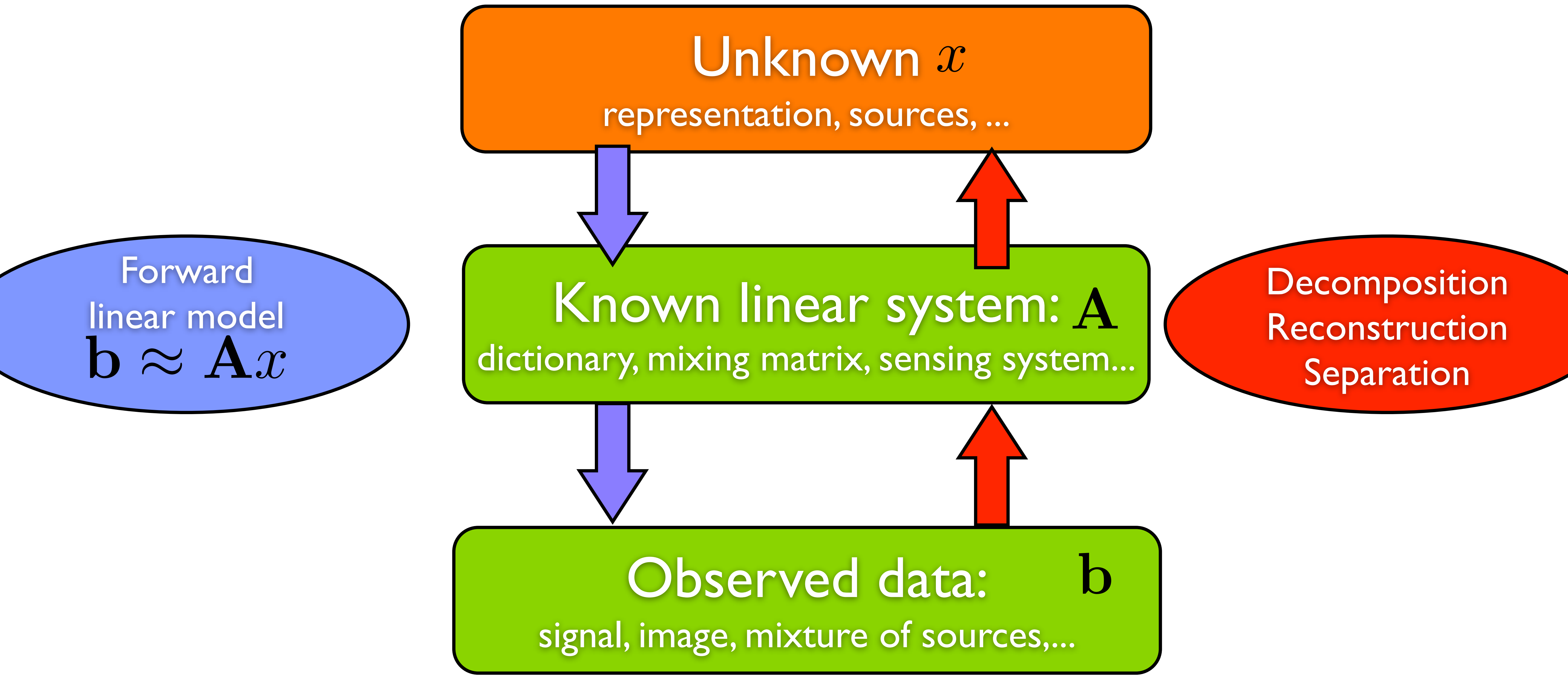


$p = 1$ $p = 2$ $p = \infty$ $p = \frac{1}{2}$

Lp Norms & Quasi-Norms



Notations & Vocabulary



Notations & Vocabulary

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

Inpainting

Wavelets

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

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

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

Ideal sparse representation

- Input:

m x N matrix $\mathbf{A}$, with $m < N$, m-dimensional vector $\mathbf{b}$

- 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

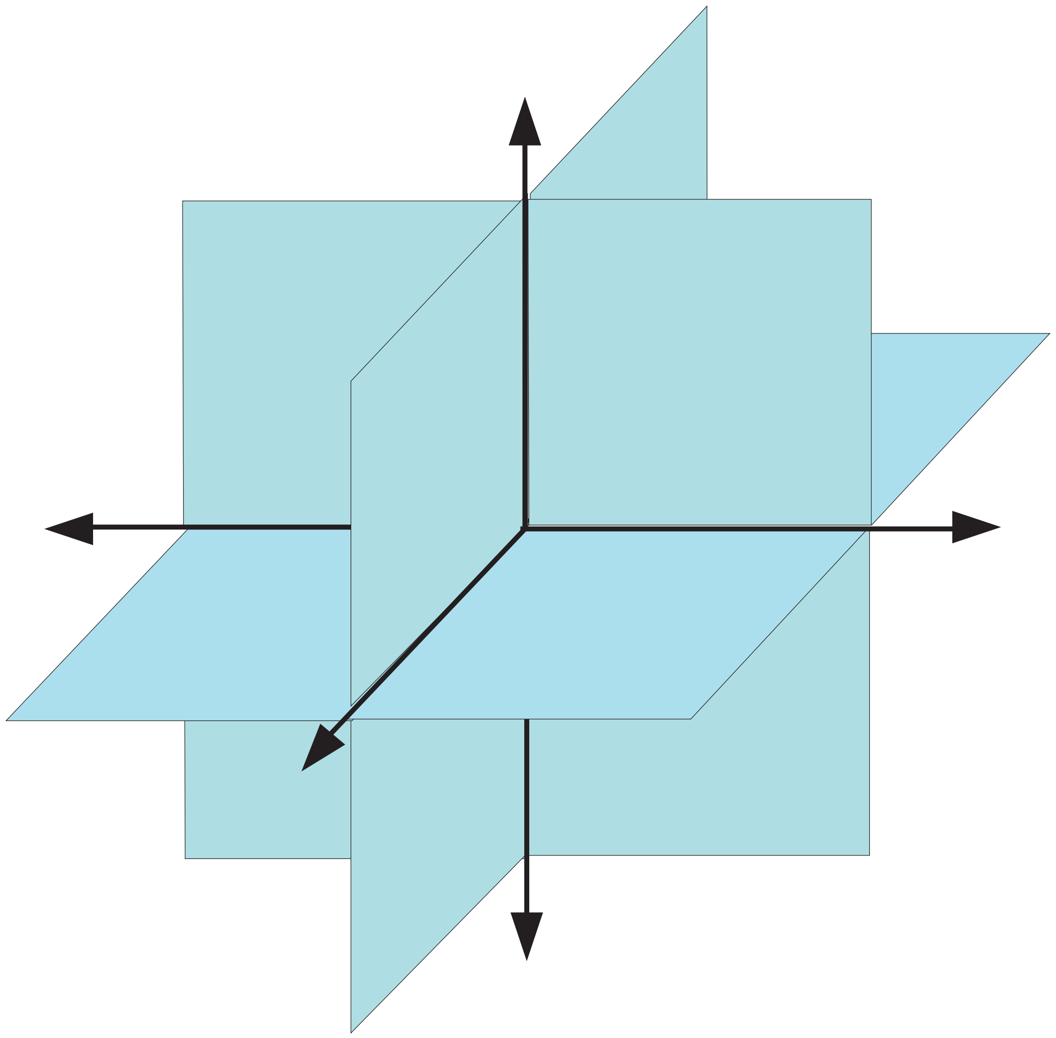
$$\arg\min_x \|\mathbf{b} - \mathbf{A}x\|, \ \text{s.t.} \|x\|_0 \leq k$$

find a solution *x* to

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

Remark: For compression, compromise between quality of the approximation and the compression level

Why is it hard? Geometric interpretation



Union of subspaces,
set of 2-sparse signals in R^3

More generally C_N^k
possible subspaces for a k-sparse signal in a space of dimension N

Combinatorial search is untractable in high dimension!
NP-complete problem.

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

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

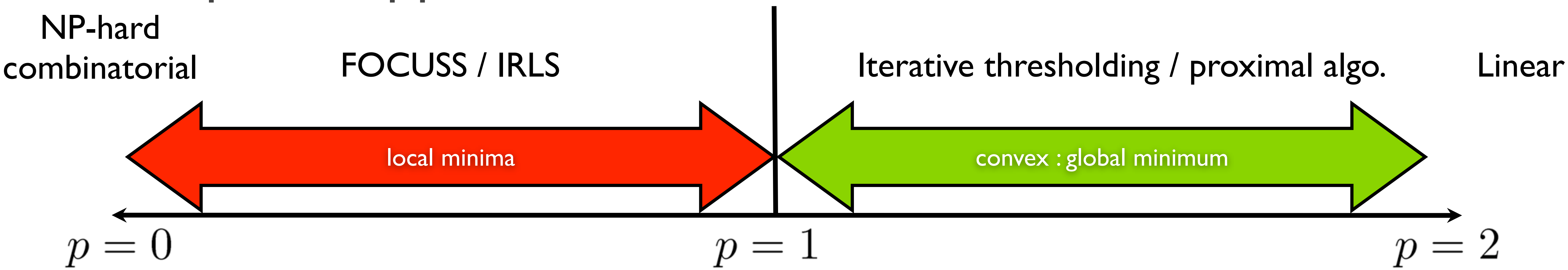
Optimization

- Optimization principle

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

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

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



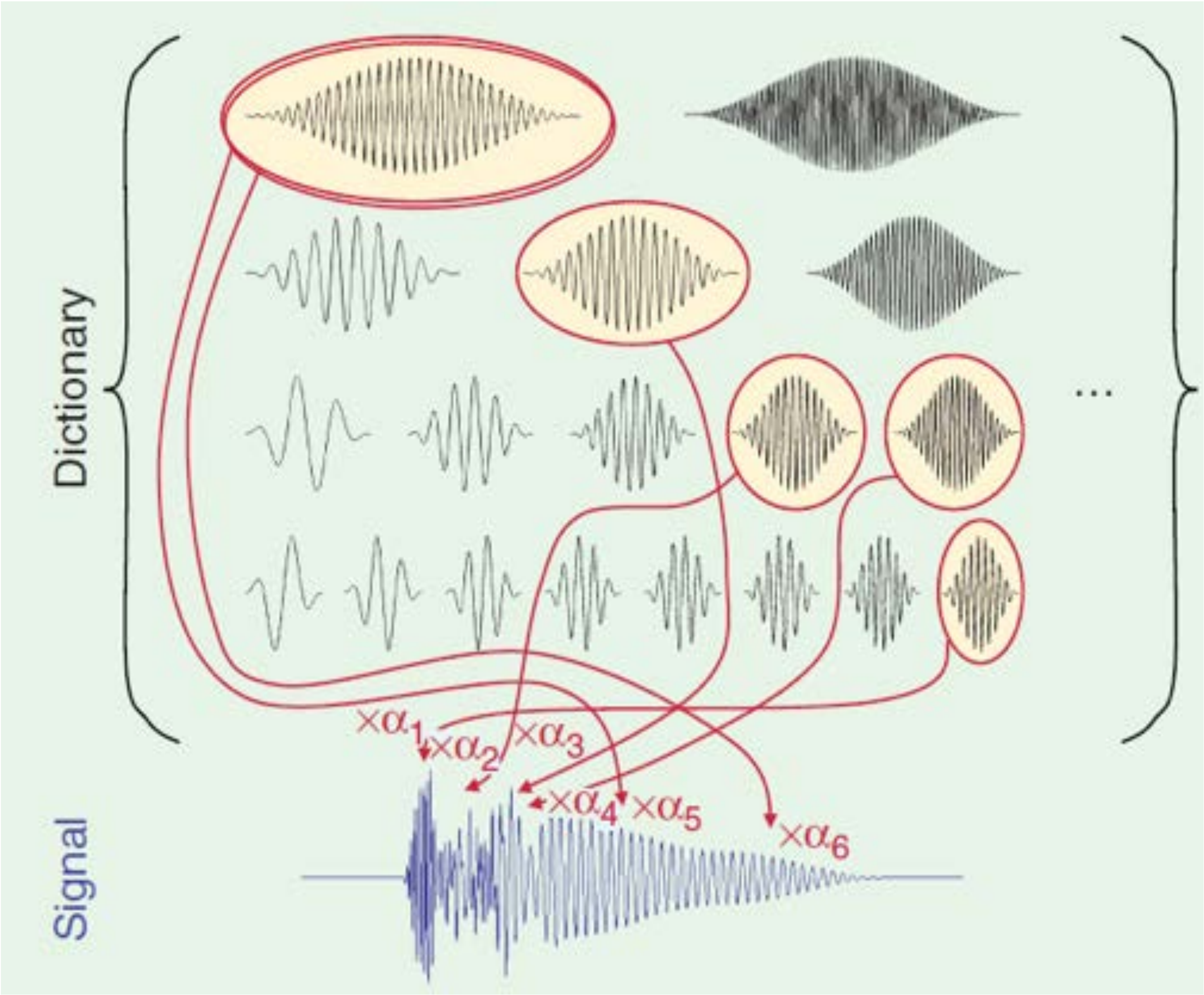
Lasso [Tibshirani 1996], *Basis Pursuit (Denoising)* [Chen, Donoho & Saunders, 1999]
Linear/Quadratic programming (interior point, etc.)
Homotopy method [Osborne 2000] / Least Angle Regression [Efron &al 2002]
Iterative / proximal algorithms [Daubechies, de Frise, de Mol 2004, Combettes & Pesquet 2008, ...]

The greedy way: matching pursuit

Good read: Chapter 3 in
http://manuel.moussallam.net/docs/manuscrit_final.pdf

« Matching Pursuit »

- The atomic decomposition is obtained with "matching pursuit" :
    - The atom the most correlated with the signal is extracted and subtracted from the original signal.
    - Iterate the procedure until a convergence criteria is satisfied (based on number of atoms or goodness of fit / distortion)



Adapted from L. Daudet: *Audio Sparse Decompositions in Parallel,* IEEE Signal Processing Magazine, 2010

Algorithm

Require: : Signal: b, dictionary A.

return : List of coefficients: (a_n), the selected atoms (γ_n)

$R_1 \leftarrow b$;

$n \leftarrow 1$;

repeat

find $A_{\gamma_n} \in A$ with maximum inner product $|\langle R_n, A_{\gamma_n} \rangle|$;

$a_n \leftarrow \langle R_n, A_{\gamma_n} \rangle$;

$R_{n+1} \leftarrow R_n - a_n A_{\gamma_n}$;

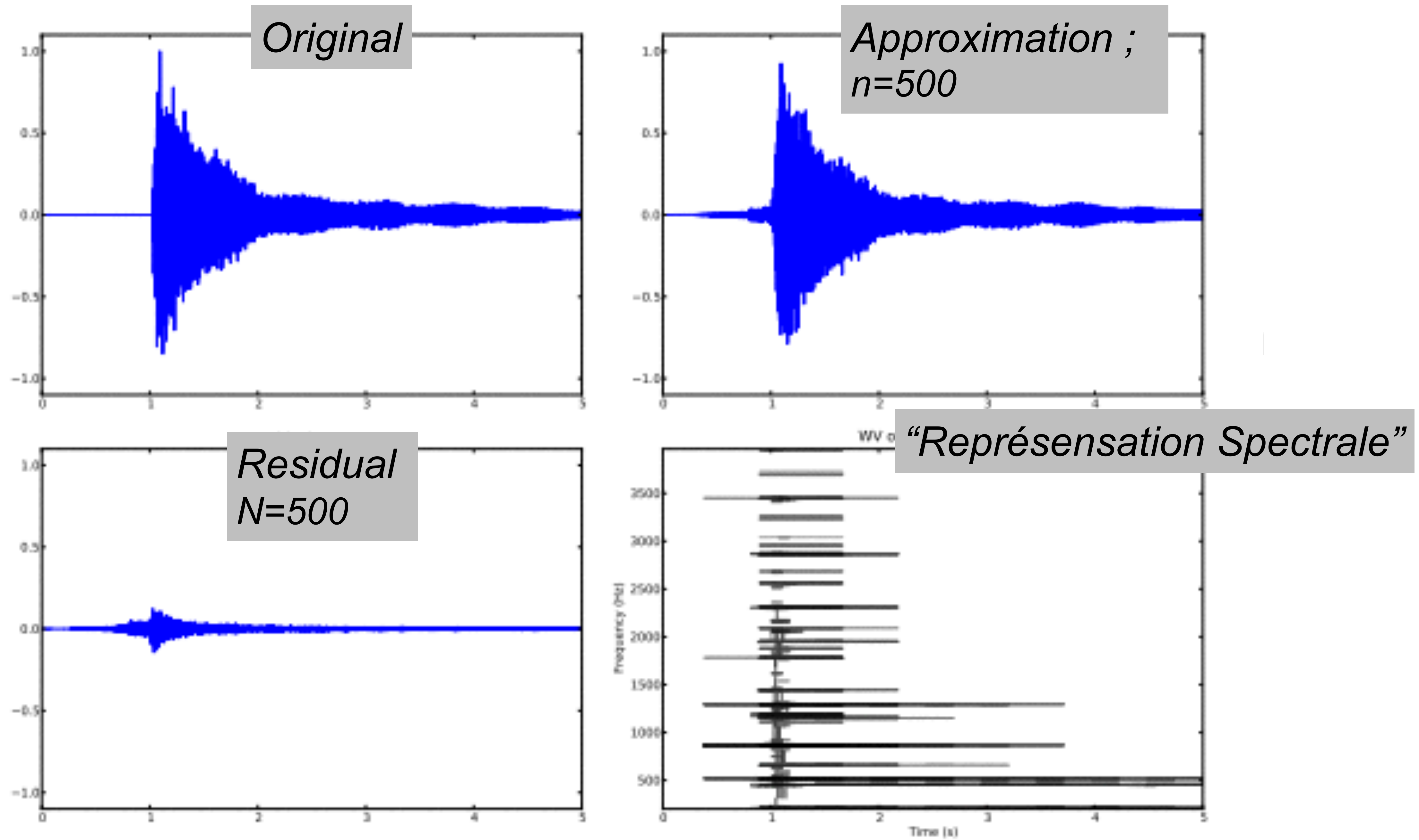
$n \leftarrow n + 1$;

until stop condition OK (for example: $\|R_n\| <$ threshold)

$b_{\text{approx}} \leftarrow \sum_n a_n A_{\gamma_n}$

Matching pursuit example

•



Convergence of Matching Pursuit (Mallat 93)

- Update rule with Pythagore

$$\|R^{n+1}x\|^2 = \|R^n x\|^2 - |\langle R^n x, d_{\gamma^n}\rangle|^2$$

residual after n iterations

- or

$$\frac{\|R^{n+1}x\|^2}{\|R^n x\|^2} = 1 - \left|\frac{\langle R^n x, d_{\gamma^n}\rangle}{\|R^n x\|}\right|^2$$

- Given $\mu(R^n \mathcal{D})$ a coherence term (maximal dot product between a normalized vector and the atoms of a dictionary) :

$$\mu(x, \mathcal{D}) = \max_{d_\gamma \in \mathcal{D}} \left|\langle \frac{x}{\|x\|}, d_\gamma \rangle\right| \leq 1$$

Convergence of Matching Pursuit (Mallat 93)

- One can show that

$$\mu_{inf}(\mathcal{D}) = \inf_{x \in \mathbb{C}^N, x \neq 0} \mu(x, \mathcal{D}) > 0$$

  - Where $\mu_{inf}(\mathcal{D})$ is the infimum of the max of the dot products
  - One obtains: $$\frac{\|R^{n+1}x\|^2}{\|R^n x\|^2} \leq 1 - \mu_{inf}^2(\mathcal{D})$$
  - Which leads to the bound: $$\|R^{n+1}x\|^2 \leq \left(1 - \mu_{inf}^2(\mathcal{D})\right)^n \|x\|^2$$

➡ **Convergence depends on the quality of the dictionary for the signal at hand**

Stability of MP algorithm

- Stability: Does MP find the best k terms approximation after k iterations ?

- To know more see [Tropp 2004]

Limitations of Matching Pursuit

- Local / greedy optimisation: does not take into account previous iterations and previously selected atoms
- It is possible to select multiple times the same atom and eventually "undo" some previous steps
- If one considers the subspace spanned by the atoms selected previously one can obtain a better approximation (smaller residual after k iterations) :

Solution: Update the coefficients of all selected atoms at each iteration (of course it's more costly)

Orthogonal Matching Pursuit (OMP)

Require: : Signal: b, dictionary A.

return : The estimated coefficients x, the selected atoms Γ

$R_1 \leftarrow b$;

$x \leftarrow 0_N$;

$n \leftarrow 1$;

repeat

find $A_{\gamma_n} \in A$ with maximum inner product $|\langle R_n, A_{\gamma_n} \rangle|$;

$\Gamma \leftarrow (\gamma_1, \ldots, \gamma_n)$;

$x_\Gamma \leftarrow \operatorname{argmin}_y \|b - A_\Gamma y\|^2$;

$R_{n+1} \leftarrow b - A_\Gamma x_\Gamma$;

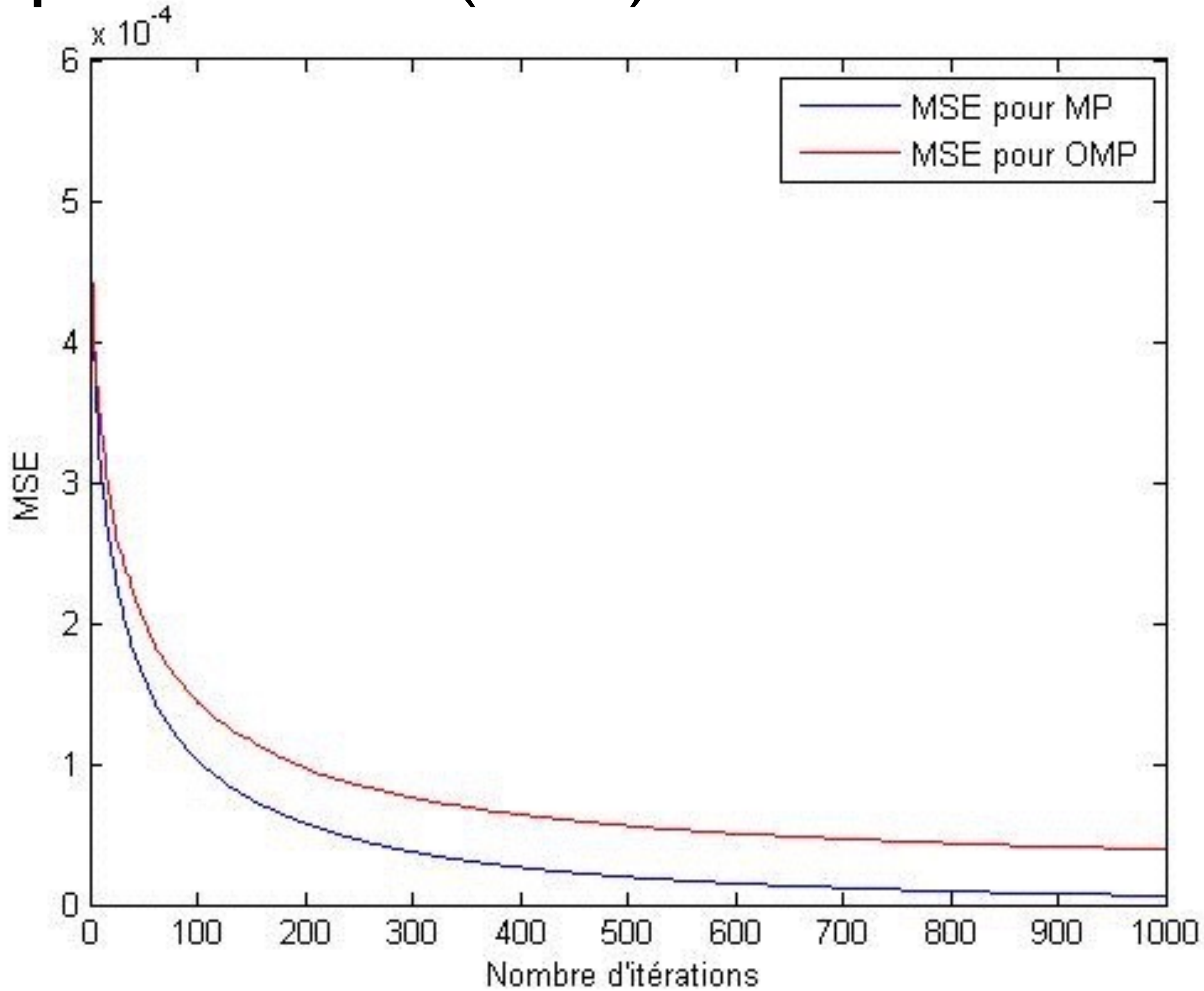
$n \leftarrow n + 1$;

until stop condition OK (for example: $\|R_n\| <$ threshold)

Remark: There exist much smarter implementations….

Orthogonal Matching Pursuit

- Comparison MP – OMP on Glockenspiel signal
- Mean-Square-Error (MSE)

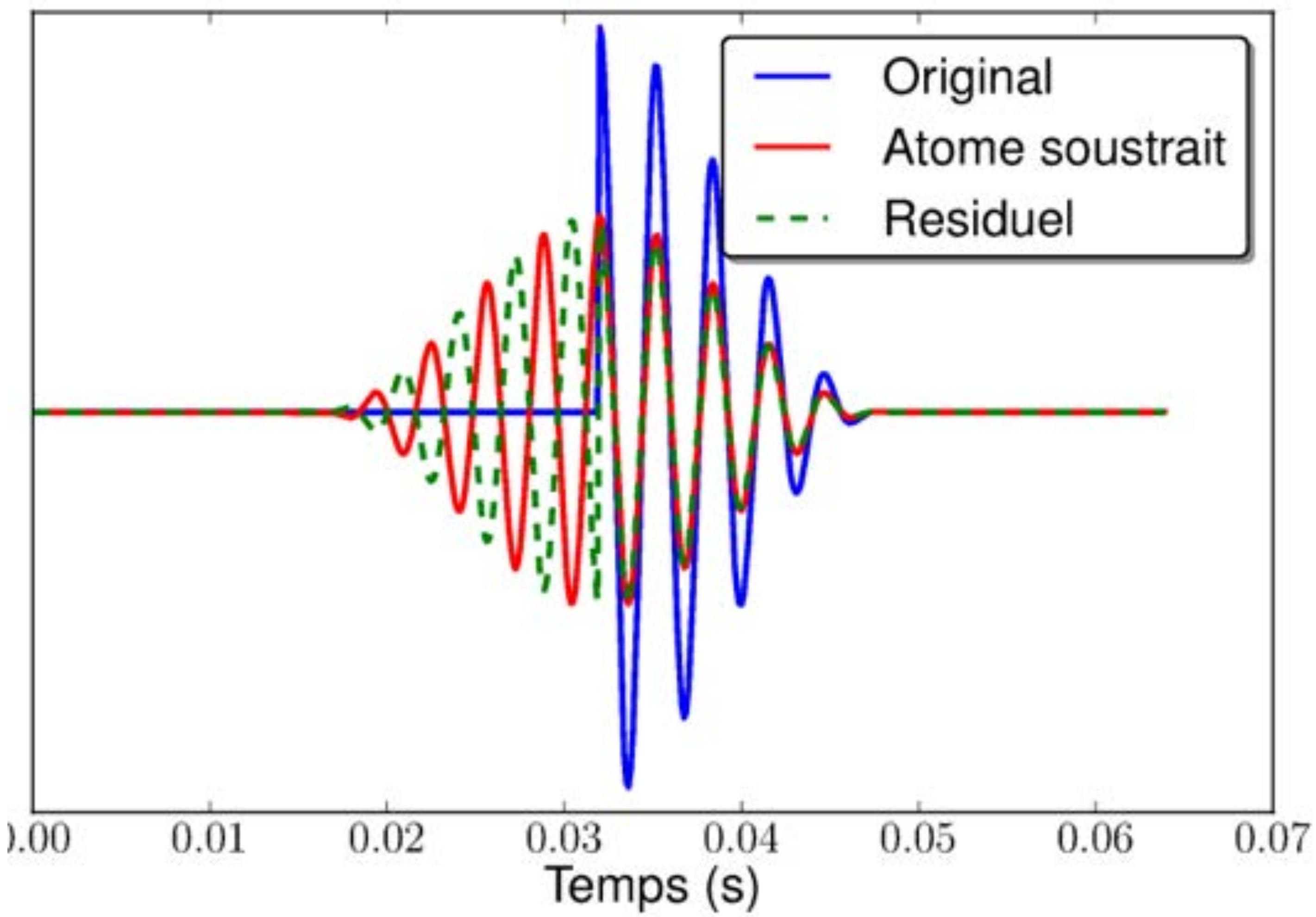


Many variants exist

- Cyclic Matching Pursuit (CMP)
- Weak Matching Pursuit
- Stagewise Greedy algorithms
- Stochastic Matching Pursuit
- Random Matching Pursuit

- ......

High-Resolution Matching Pursuit (Jaggi et al 98, Gribonval et al. 96)



- MP can produce artifacts (pre-echo)

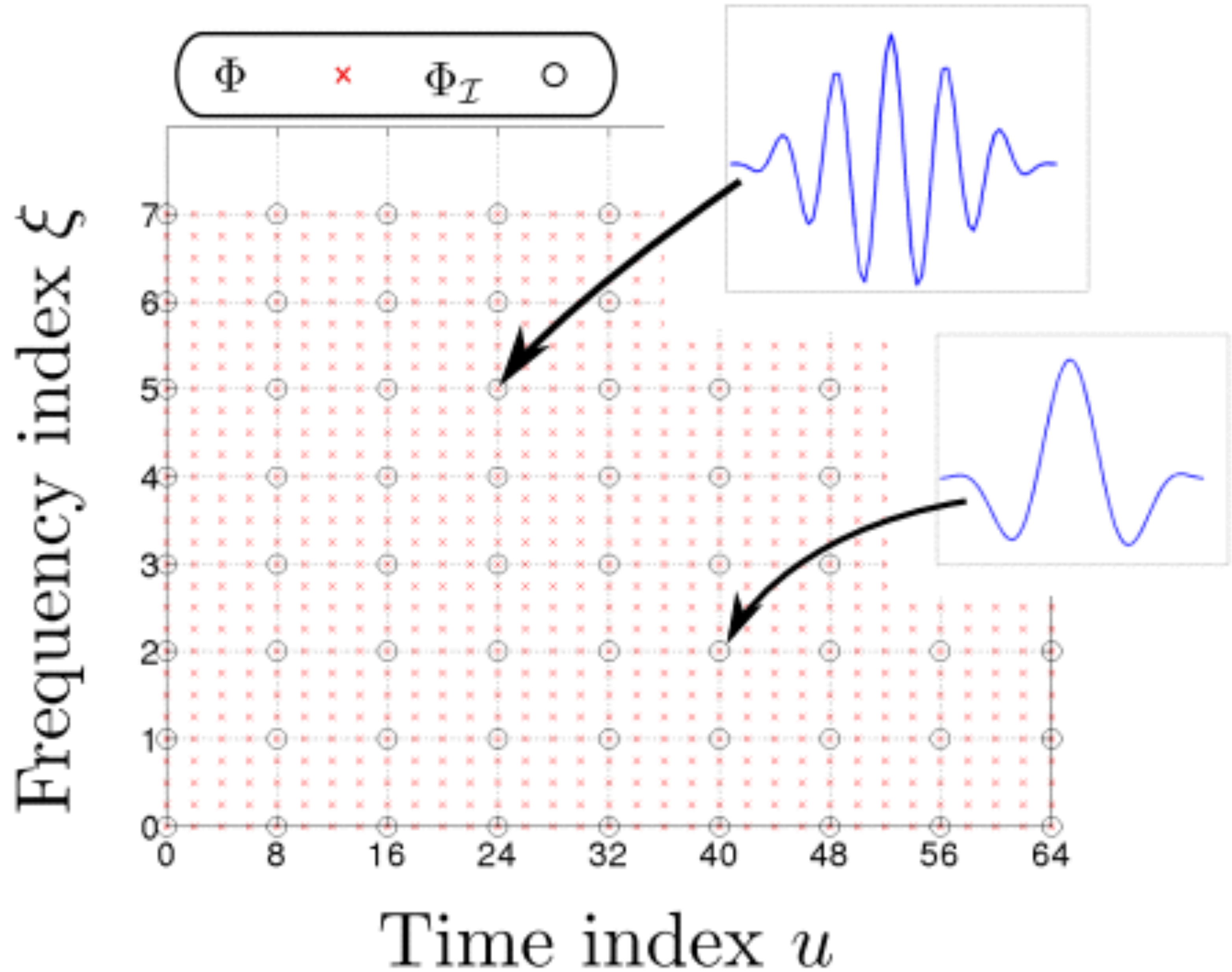
Adapted from [Moussalam 2012]

Weak Matching Pursuit

- The complexity of MP is due to the computation of the dot products between current residual and all dictionary atoms
- Idea: Restrict the search to a sub-dictionary
  - At iteration n, on chooses an atom d_{γ^n} such that: $t_n \in]0-1]$
  - With optimality control $$|\langle R^n x, d_{\gamma^n}\rangle| \geq t_n \sup_{d_i \in \mathcal{D}} |\langle R^n x, d_i\rangle|$$
  - In practice this is done by limiting the number of dot products to a "good" sub-dictionary and finding the best atom in this limited dictionary

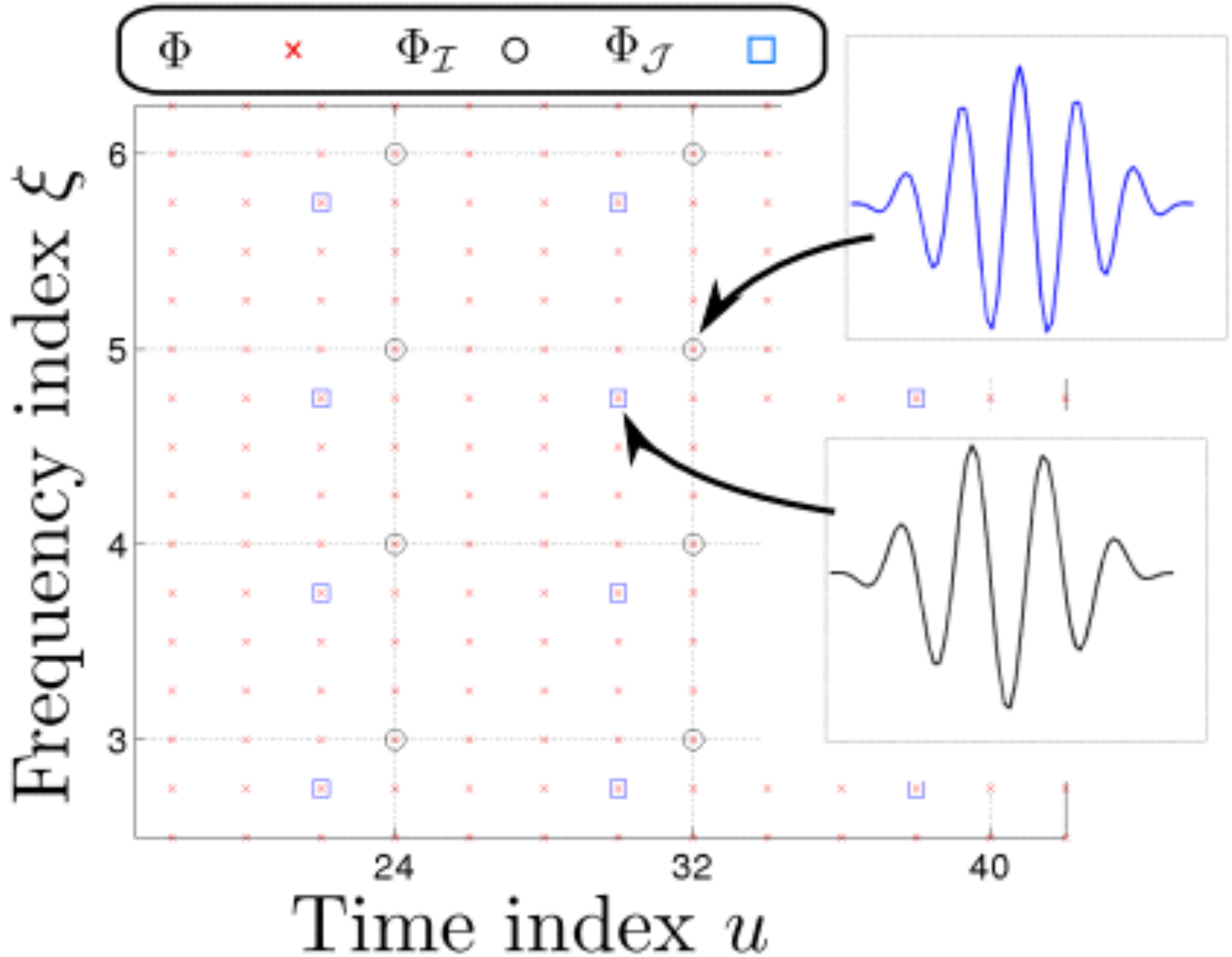
Weak Matching Pursuit

- Example with a sub-sampled time-frequency dictionary



Weak Matching Pursuit

- Using a different sub-dictionary yields a different selection of time-frequency atoms



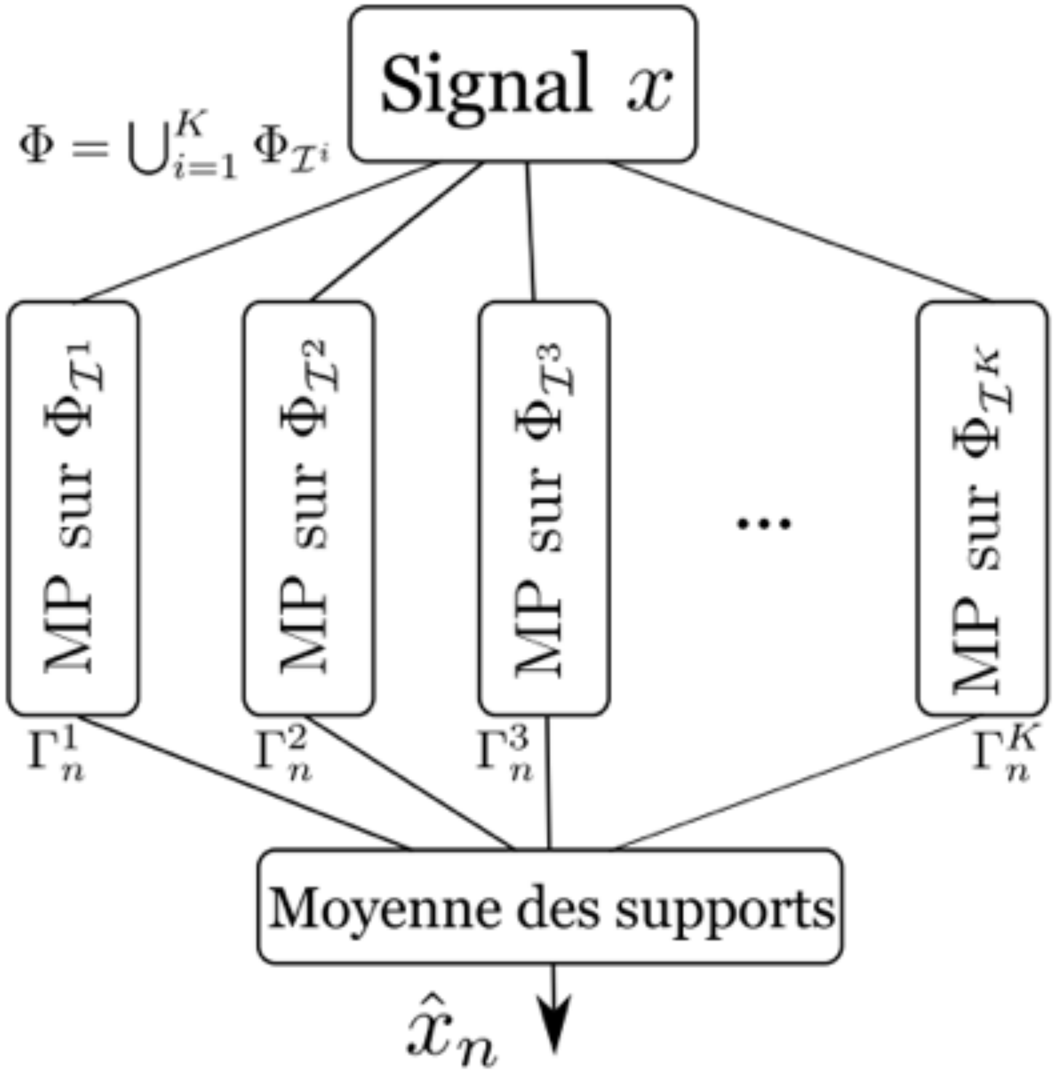
Adaptative Matching Pursuit

- Weak MP can be combined with a local search/ optimization of the best atom
  1. Search of the best atom in sub-dictionary
  2. Optimization (local) of the selected atom

- For example in a TF dictionary
  - Step 1 corresponds to a selection in a coarse TF grid
  - Step 2 corresponds to an optimization of the TF localization of the selected atom

Stochastic MP

- The idea is to compute multiple weak MP in parallel with random sub-dictionaries and to combine the decompositions (for example using simple averaging)



[Moussallam 2012]

The convex way

Convex

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

Basis Pursuit (BP)
LASSO

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

Exercise: Rewrite the problem as a linear program

Convex

- 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*

$$\min_{z \geq 0} \mathbf{c}^T z, \text{ s.t. } [\mathbf{A}, -\mathbf{A}]z = \mathbf{b}$$
$$\mathbf{c} = (c_i), \; c_i = 1, \forall i$$

Exercise: Prove that the set of minimizers is a convex set

Convex

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

Basis Pursuit Denoising
(BPDN)

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

Exercise: Rewrite the problem as a quadratic program (QP)

Convex

- 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 vs. Specific

- 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/`

CVX Matlab Toolbox

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2 \\ \text{subject to} & Cx = d \\ & \|x\|_\infty \leq e \end{array}$$

The following code segment generates and solves a random instance of this model:

```
m = 20; n = 10; p = 4;
A = randn(m,n); b = randn(m,1);
C = randn(p,n); d = randn(p,1); e = rand;
cvx_begin
    variable x(n)
    minimize( norm( A * x - b, 2 ) )
    subject to
        C * x == d
        norm( x, Inf ) <= e
cvx_end
```

http://cvxr.com/cvx/

CVX Matlab Toolbox

```
n = 10;
A = randn(n/2,n);
b = randn(n/2,1);
gamma = 1;

cvx_begin
  variable x(n)
  dual variable y
  minimize( 0.5 * norm( A * x - b, 2 ) + gamma *
norm( x, 1 ) )
cvx_end
```

http://cvxr.com/cvx/

CVX Matlab Toolbox

```
n = 10;
A = randn(n/2,n);
b = randn(n/2,1);
gamma = 1;

cvx_begin
  variable x(n)
  dual variable y
  minimize( norm( x, 1 ) )
  subject to
     y : norm(A*x - b, 2) <= 0.1;
cvx_end
```

http://cvxr.com/cvx/

What if A is orthogonal?

- 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

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

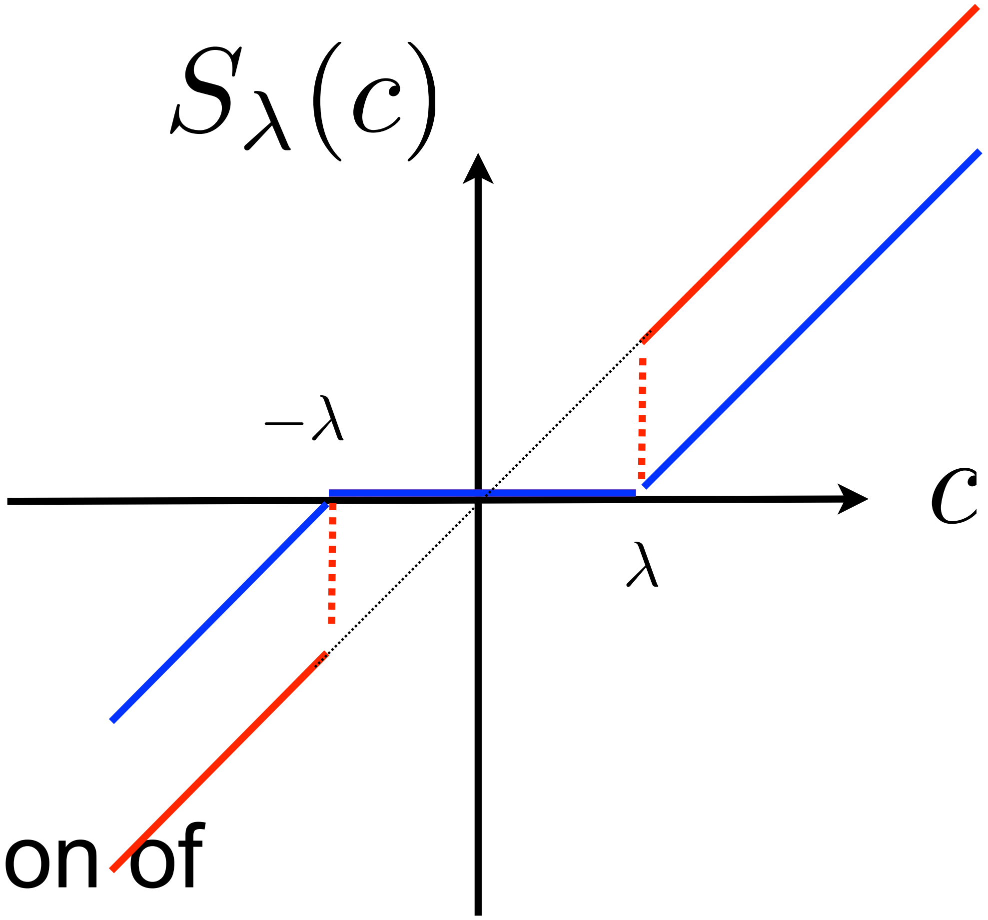
- Minimization can be done coordinate-wise

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

Exercise: What is the solution for p=2, 1 and 0?

What if A is orthogonal?

Soft-thresholding (p=1)

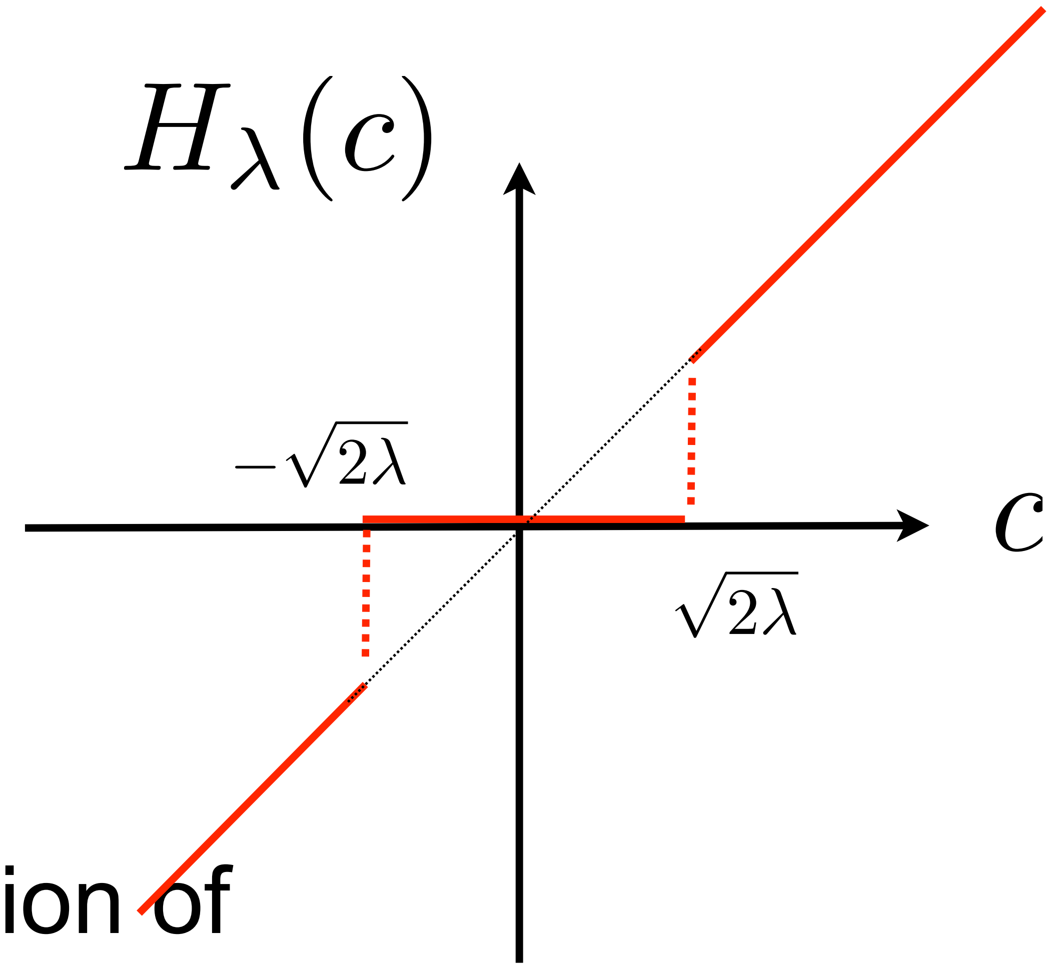


- Solution of

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

What if A is orthogonal?

Hard-thresholding (p=0)



- Solution of

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

What if A is orthogonal?

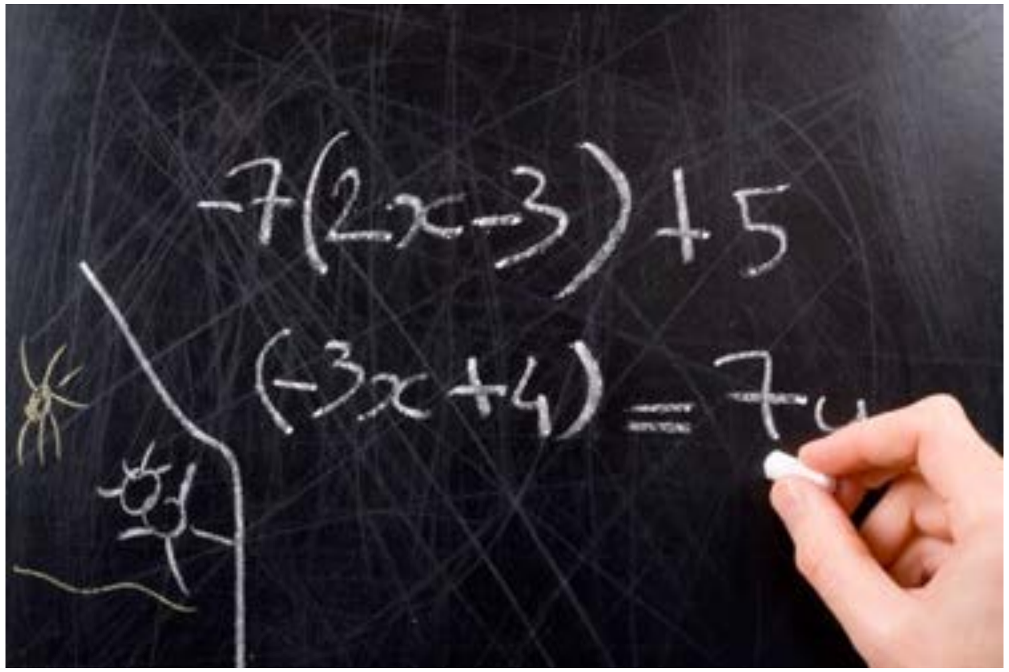
- Soft thresholding
- `@softthresh(c,lambda)(sign(c).*max(abs(c)-lambda,0))`
- `x = softthresh(c,lambda);`

```
def softthresh(c, lamdb):
    return np.sign(c) * np.maximum(np.abs(c) - lambd, 0)
```

- Hard-thresholding
- `@hardthresh(c,lambda)(c.*(abs(c)>=sqrt(2*lambda)))`
- `x = hardthresh(c,lambda);`

```
def hardthresh(c, lamdb):
    return c * (np.abs(c) >= (2 * lambd))
```

What if A is NOT orthogonal?



Hint: Use an iterative scheme and linearize the data term (smooth) with Taylor and an upper bounding the 2nd order term.

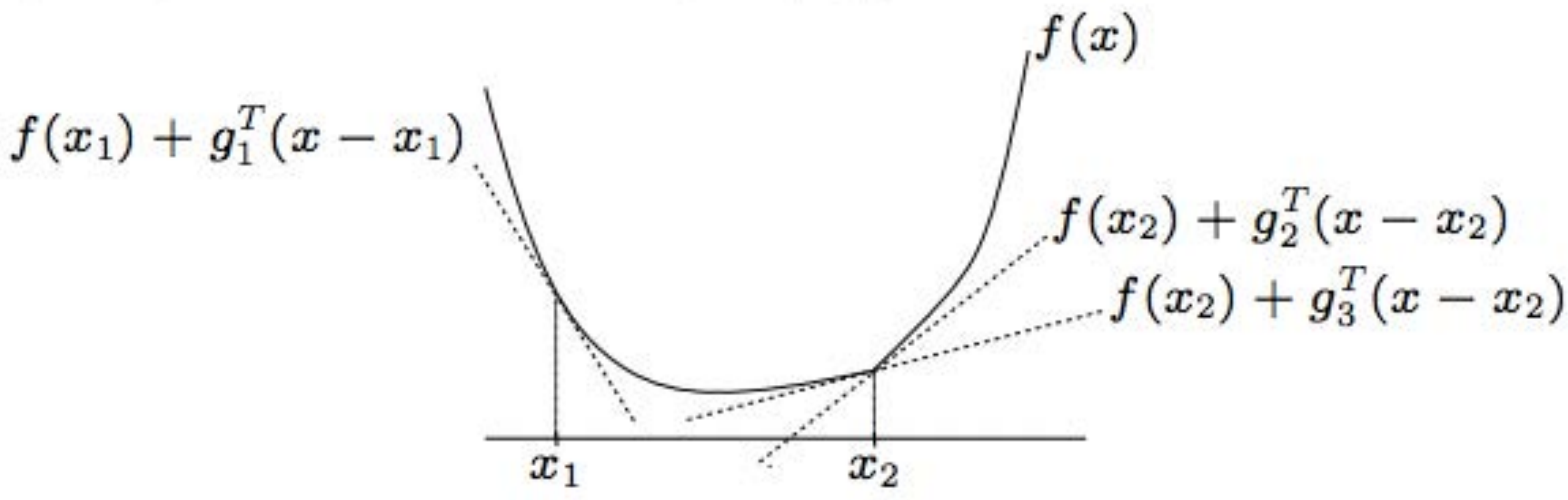
Subgradient and subdifferential

For a smooth convex function we have

$$f(x) - f(x_0) \geq \nabla f(x_0)^T (x - x_0) \qquad \forall x$$

For a non-smooth convex function, g is a **subgradient** at x_0 if:

$$f(x) - f(x_0) \geq g^T (x - x_0) \qquad \forall x$$



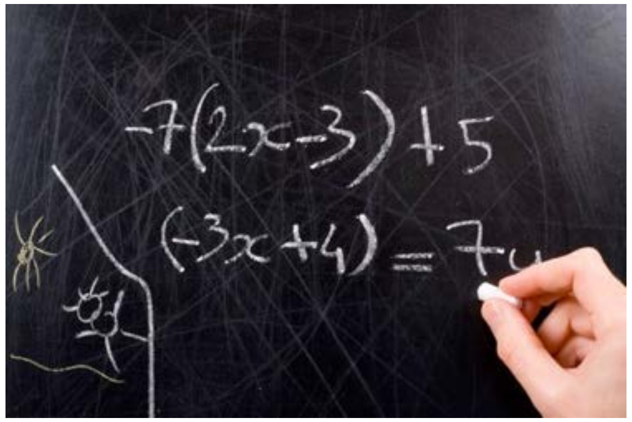
Subgradient and subdifferential

The subdifferential of f at x_0 is:

$$\partial f(x_0) = \{g \in \mathbb{R}^n / f(x) - f(x_0) \geq g^T(x - x_0), \forall x \in \mathbb{R}^n\}$$

Properties

- The subdifferential is a convex set
- x_0 is a minimizer of f if $0 \in \partial f(x_0)$



Exercise:

What is $\partial|\cdot|(0) = ?$

Path of solutions

- **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

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

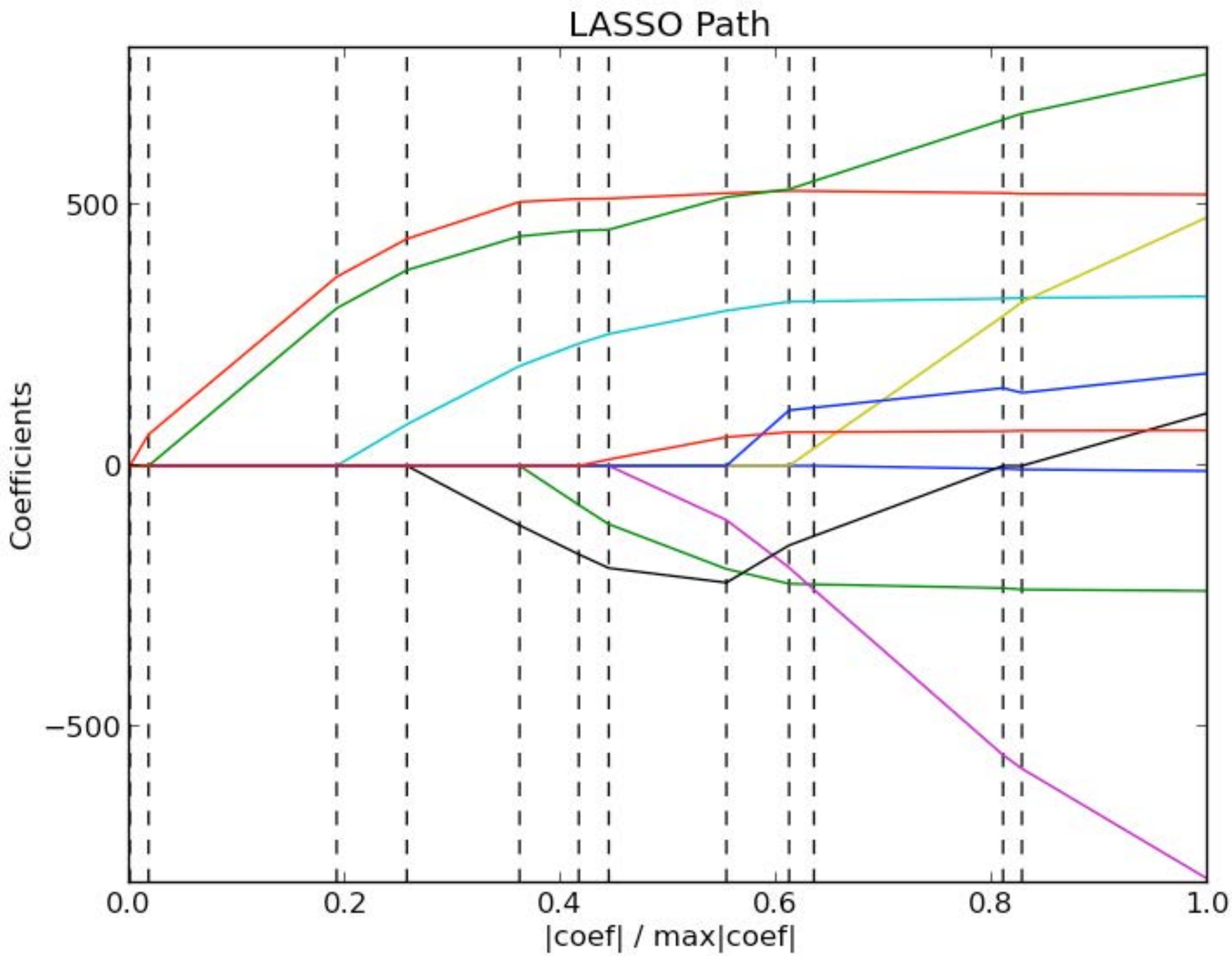
Homotopy

- **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

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

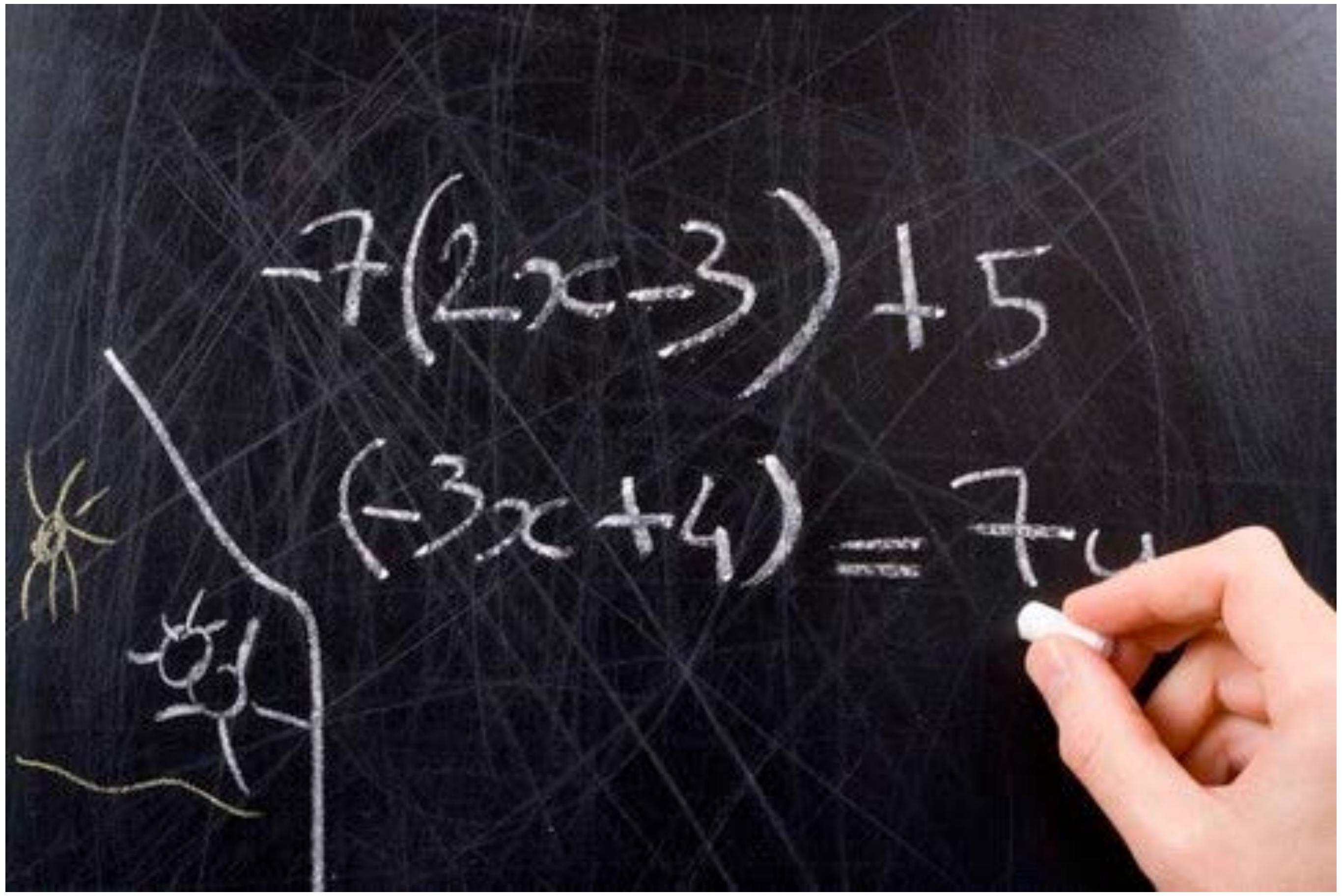
  - ✓ for given sign pattern, dependence on λ is affine
  - ✓ sign patterns are piecewise constant functions of λ
  - ✓ overall, the solution is piecewise affine
- **Method** = iteratively find *breakpoints*
  - ✓ [Osborne 2000; Efron & al 2004]

Piecewise affine solution



Coordinate Descent

Idea: Update each x_i sequentially



Coordinate Descent

```
from scipy import linalg
import numpy as np

def fsign(f):
    if f == 0:
        return 0
    elif f > 0:
        return 1.0
    else:
        return -1.0

def lasso(A, b, alpha, max_iter=200):
    n, m = A.shape
    norm_cols_A = (A ** 2).sum(axis=0)
    x = np.zeros(m)
    for n_iter in range(max_iter):
        for ii in xrange(m):
            x_ii = x[ii]
            # Get current residual
            R = b - np.dot(A, x) + A[:, ii] * x_ii
            tmp = np.dot(A[:, ii], R)
            # Soft thresholding
            x[ii] = fsign(tmp) * max(abs(tmp) - alpha, 0) / norm_cols_A[ii]
    return x
```

Some theory

Well posedness?

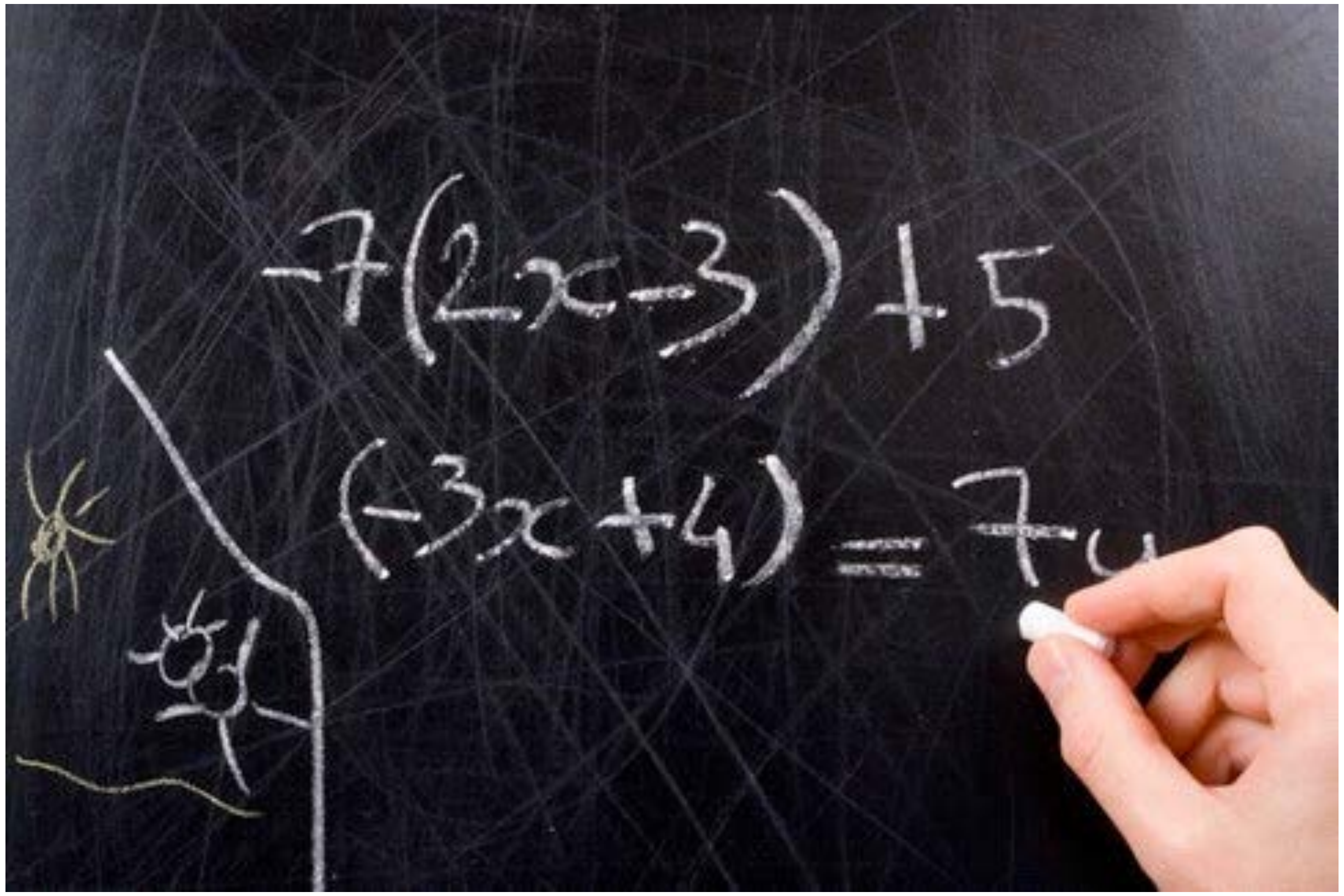
- What property should **A** satisfy such that, for any pair of k-sparse vectors x_0, x_1

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

Well posedness for 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 for k-sparse vectors

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

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

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

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

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

Well posedness for k-sparse vectors

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

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

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

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

$$m < N$$

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

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

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

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

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

Well posedness for k-sparse vectors

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

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

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

Conclusion

- Bottleneck 1990-2000 : under-determined = fewer equations than unknowns = ill-posed

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

- Novelty 2001-2006 :
  - ✓ **Uniqueness** of sparse solution:
    - ✦ if x_0, x_1 are "sufficiently sparse" (in an appropriate «domain»),
    - ✦ then $\mathbf{A}x_0 = \mathbf{A}x_1 \Rightarrow x_0 = x_1$
  - ✓ Recovery of x_0 with **efficient** algorithms
    - ✦ Thresholding, Matching Pursuits, Minimisation of Lp norms p<=1,...

Conclusion

- What if you don’t know a good representation of the signal?
- Let’s learn it : *Dictionary Learning*

Learn more

Reading list:

- *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*
- *Review paper:*
  - *Bruckstein, Donoho, Elad, SIAM Reviews, 2009*