



## Inverse problems and sparse models (2/6)

Rémi Gribonval

INRIA Rennes - Bretagne Atlantique, France

[remi.gribonval@inria.fr](mailto:remi.gribonval@inria.fr)

# 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
- Slides of this course:
  - ◆ <http://www.irisa.fr/metiss/gribonval/Teaching/SISEA-2012/>

# PDF of the slides

- <http://www.irisa.fr/metiss/gribonval/Teaching/>

# Reminder of last session

- Session 1: Introduction
  - ✓ sparsity & data compression
  - ✓ inverse problems in signal and image processing
    - ◆ image deblurring, image inpainting,
    - ◆ channel equalization, signal separation,
    - ◆ tomography, MRI
  - ✓ sparsity & under-determined inverse problems
    - ◆ relation to subset selection problem

# Sparsity: definition

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

# Sparsity: definition

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

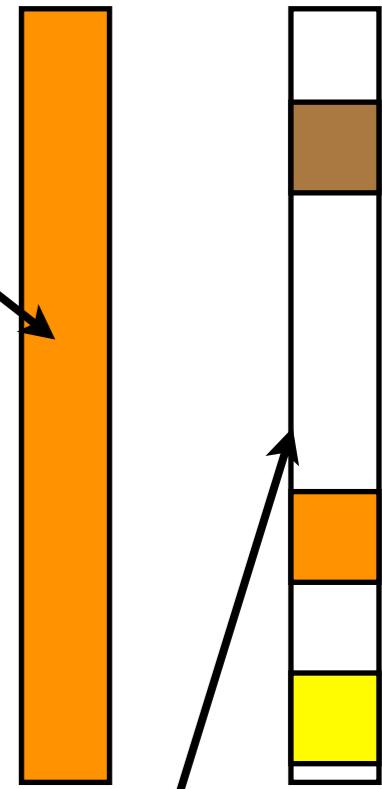
Not sparse



# Sparsity: definition

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

Not sparse

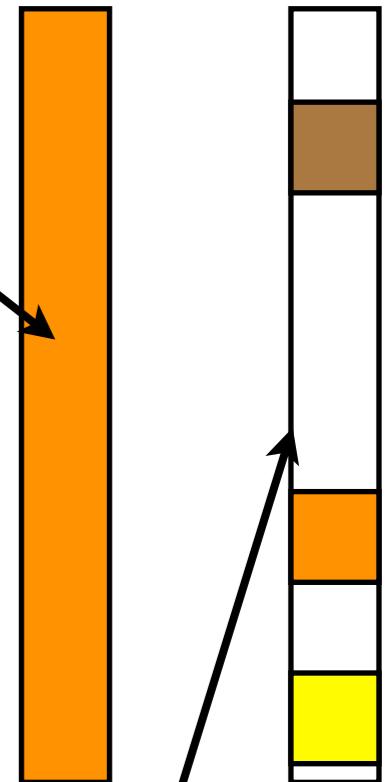


3-sparse

# Sparsity: definition

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

Not sparse



3-sparse

# Sparsity: definition

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

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

**Convention here**

$$a^0 = 1(a > 0); 0^0 = 0$$

Not sparse



3-sparse



# Sparsity: definition

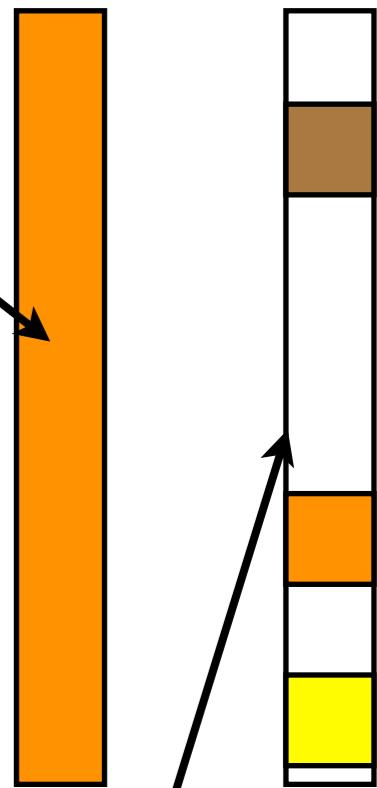
- A vector is
  - ✓ **sparse** if it has (many) zero coefficients
  - ✓  **$k$ -sparse** if it has *at most  $k$*  nonzero coefficients
- Symbolic representation as column vector
- **Support** = indices of nonzero components
- Sparsity measured with **L0 pseudo-norm**

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

- *In french:*
  - ◆ sparse
  - ◆ sparsity, sparseness

-> «creux», «parcimonieux»  
-> «parcimonie», ~~«sparsité»~~

Not sparse



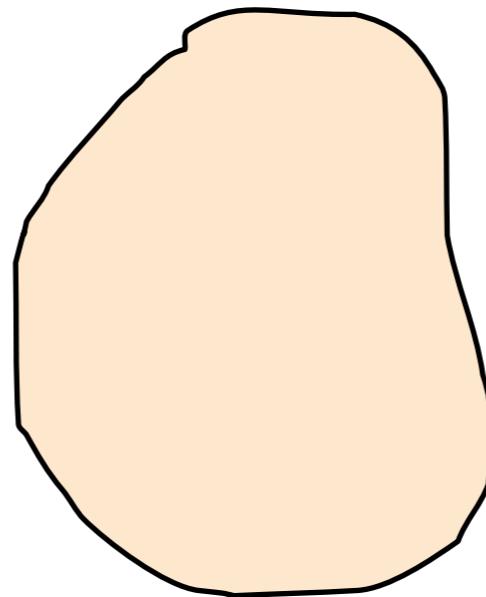
3-sparse

**Convention here**

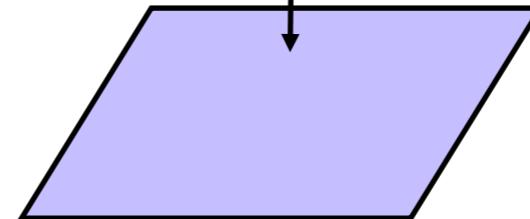
$$a^0 = 1(a > 0); 0^0 = 0$$

# Inverse problems

Signal space  $\sim \mathbb{R}^N$



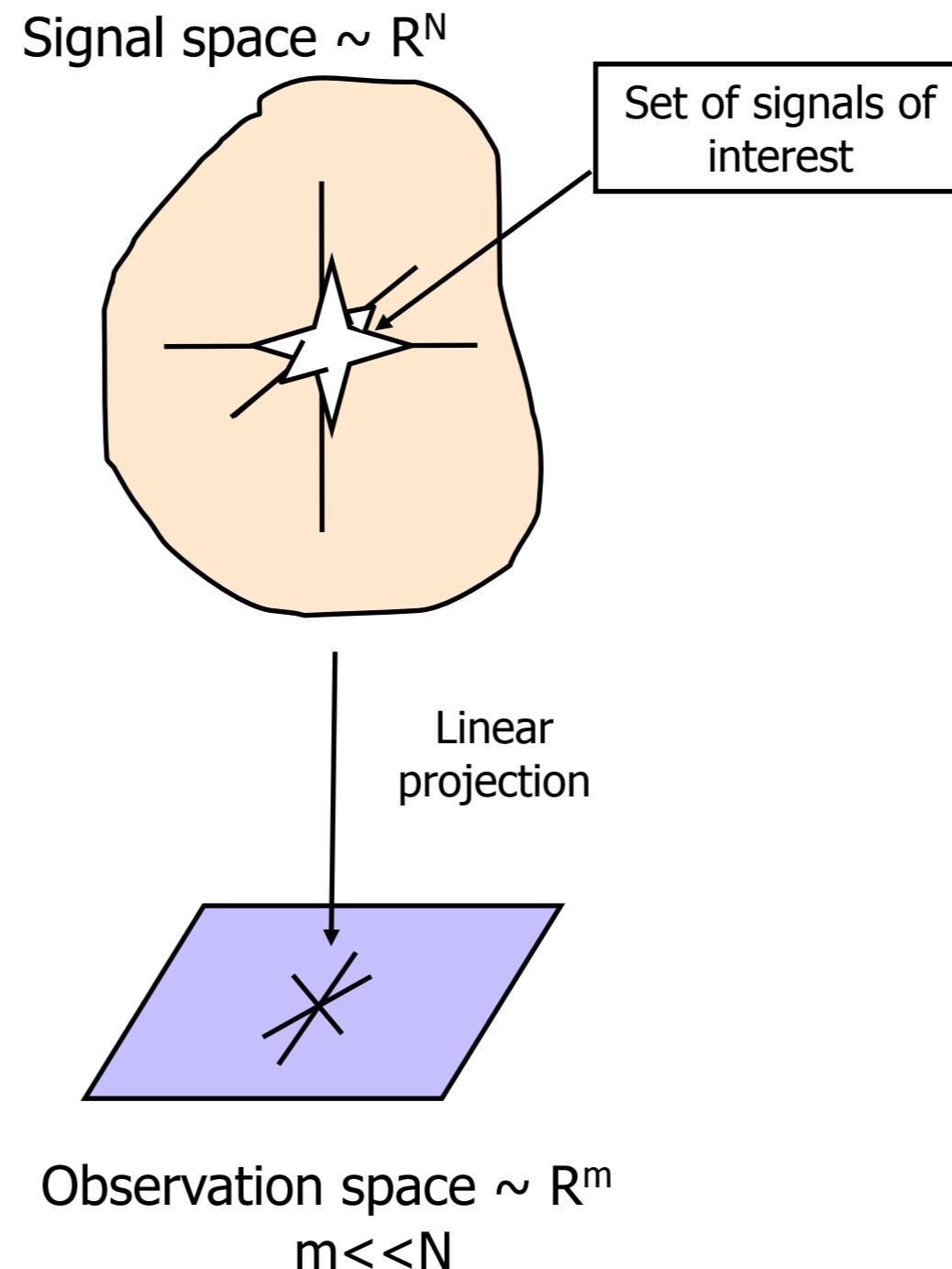
Linear  
projection



Observation space  $\sim \mathbb{R}^m$   
 $m \ll N$

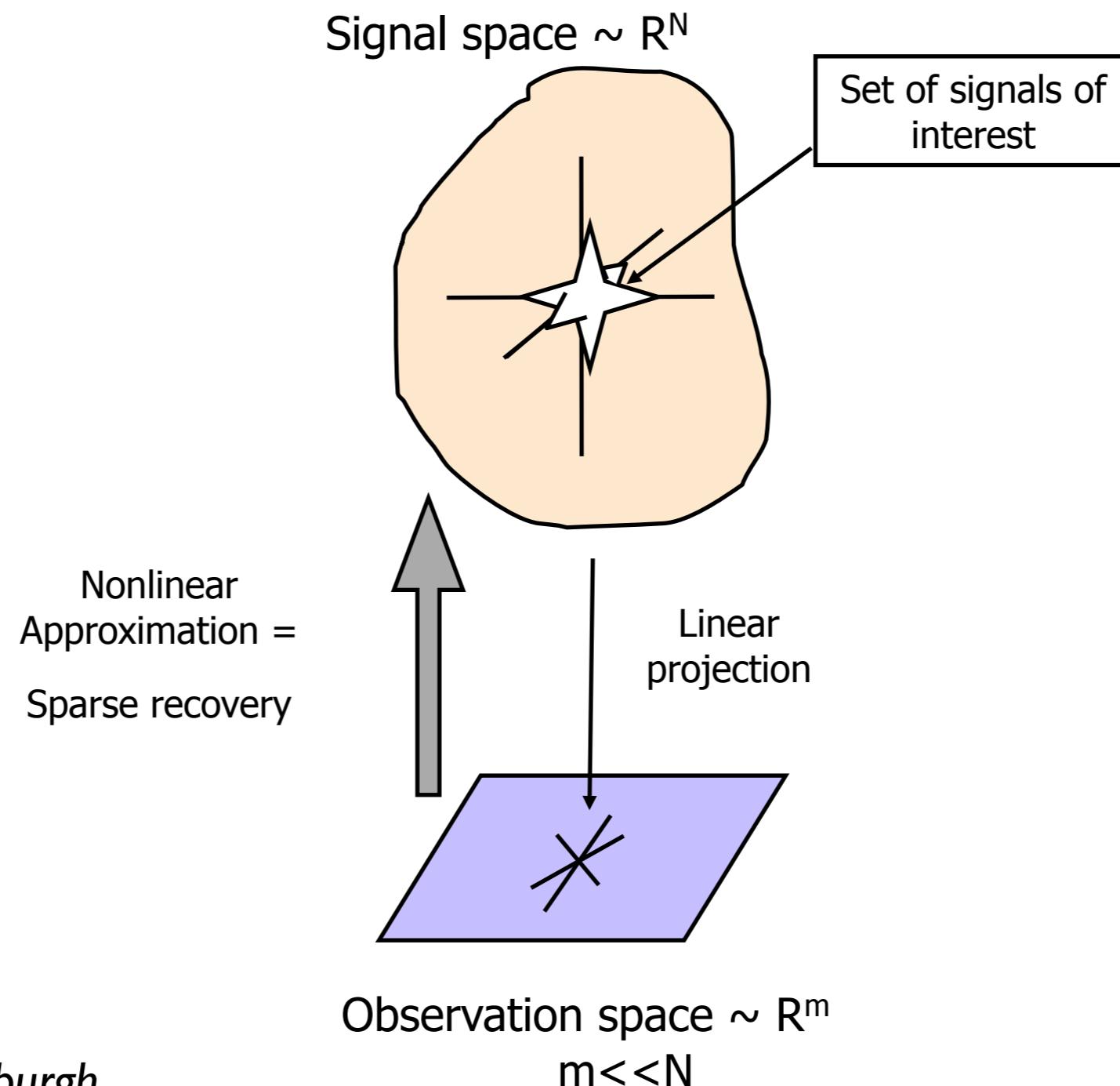
Courtesy: M. Davies, U. Edinburgh

# Inverse problems



Courtesy: M. Davies, U. Edinburgh

# Inverse problems



Courtesy: M. Davies, U. Edinburgh

# Overview of the course

## ● Introduction

- ✓ sparsity & data compression
- ✓ inverse problems in signal and image processing
  - ◆ image deblurring, image inpainting,
  - ◆ channel equalization, signal separation,
  - ◆ tomography, MRI
- ✓ sparsity & under-determined inverse problems
  - ◆ relation to subset selection problem

## ● Pursuit Algorithms

- ✓ Greedy algorithms: Matching Pursuit & al
- ✓ L1 minimization principles
- ✓ L1 minimization algorithms
- ✓ Complexity of Pursuit Algorithms

# Ideal sparse recovery

# Computing a sparse representation ?

- How to compute a  $k$ -sparse solution to

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



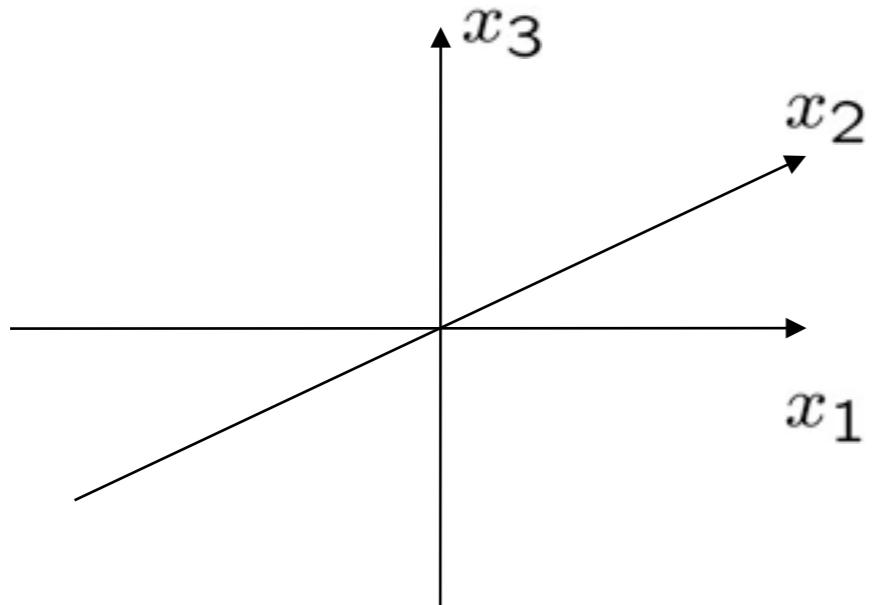
# Ideal sparse approximation

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

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

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search

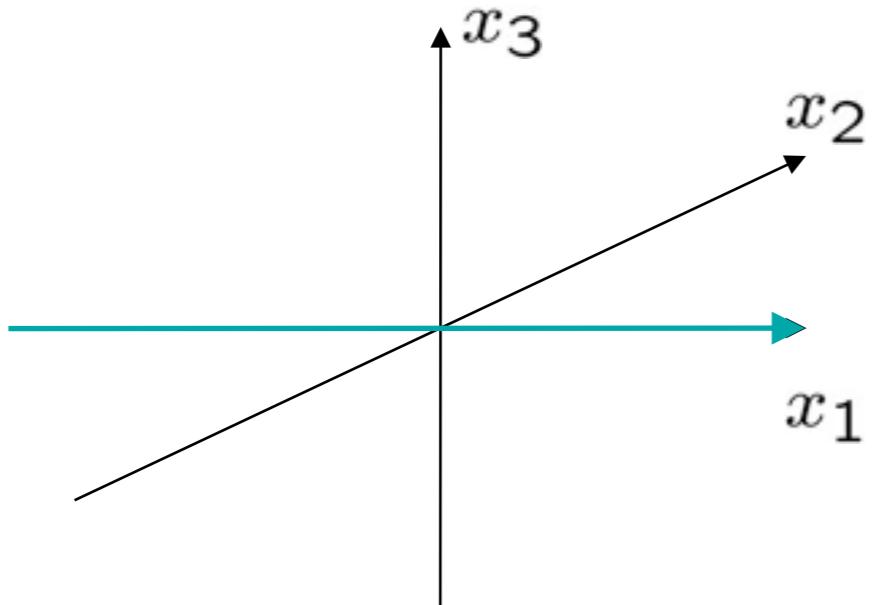


- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search

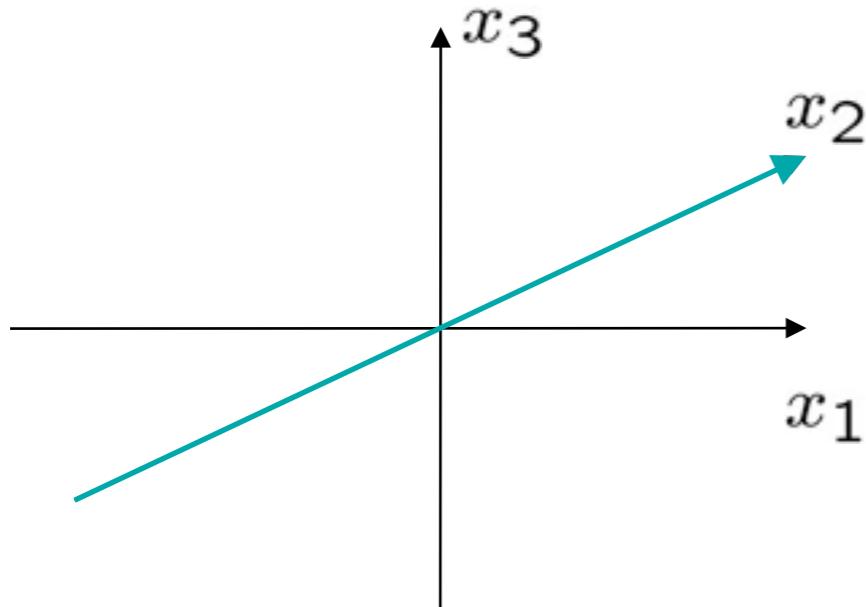


- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search

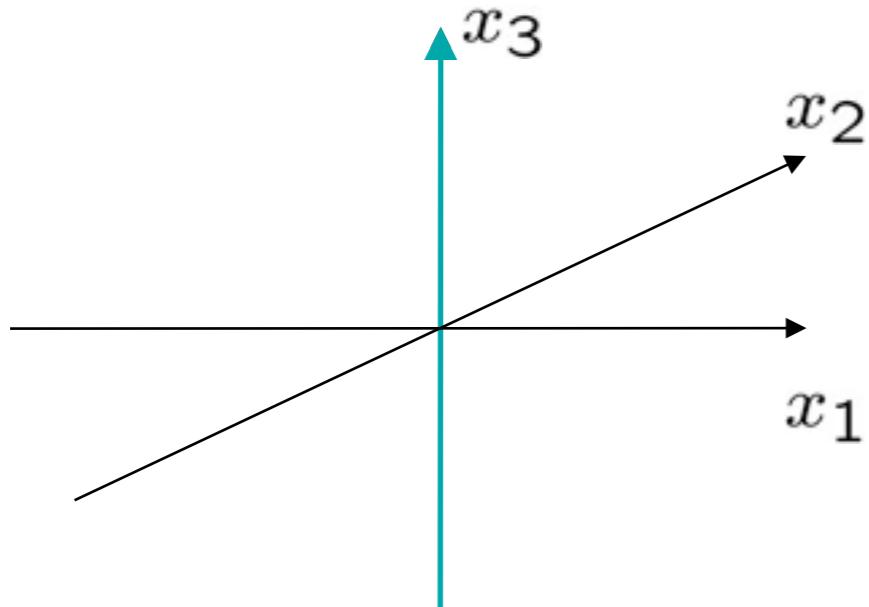


- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search

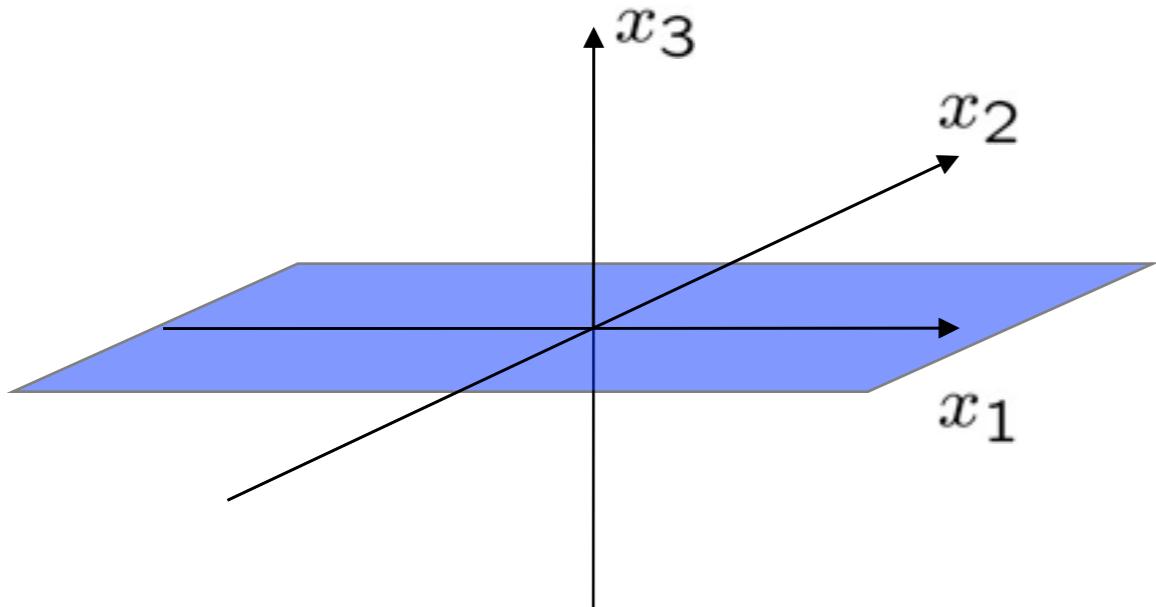


- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search

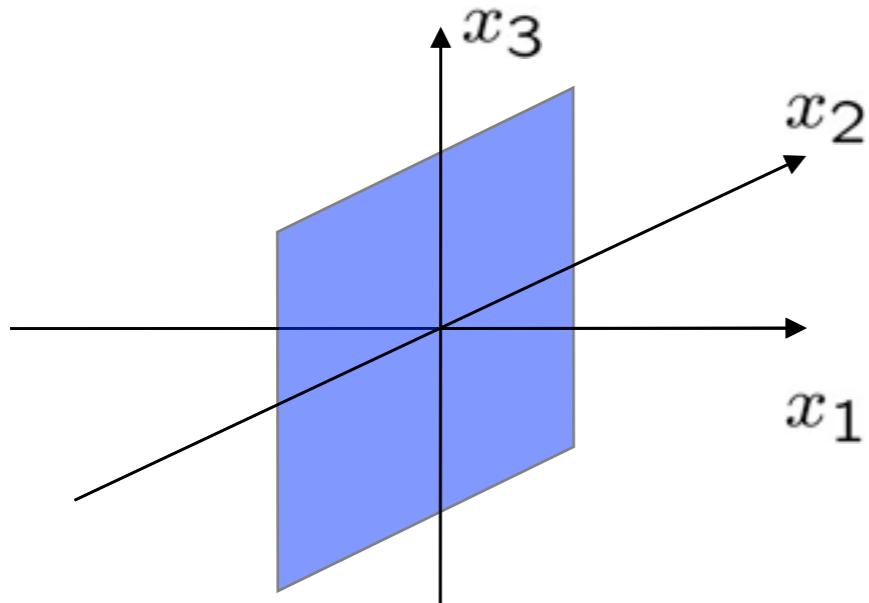


- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search

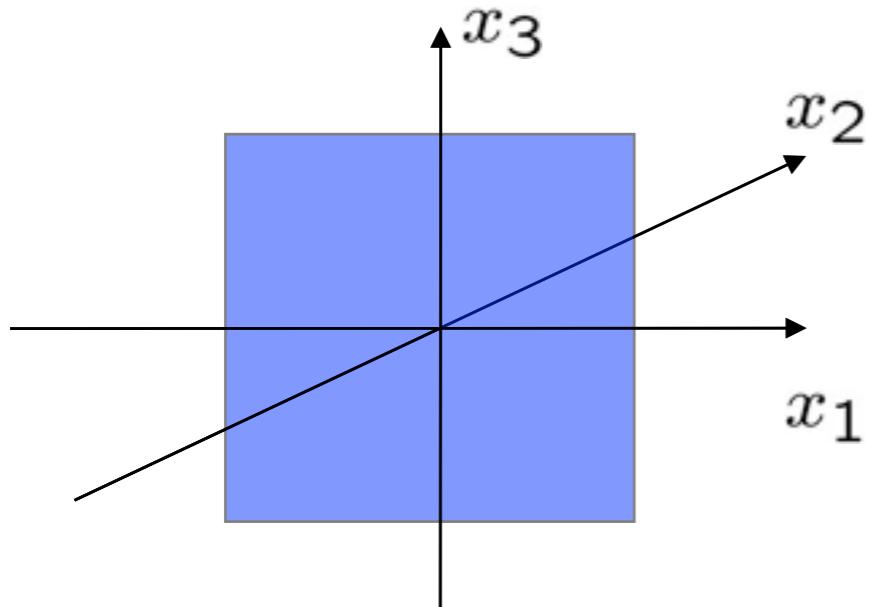


- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search

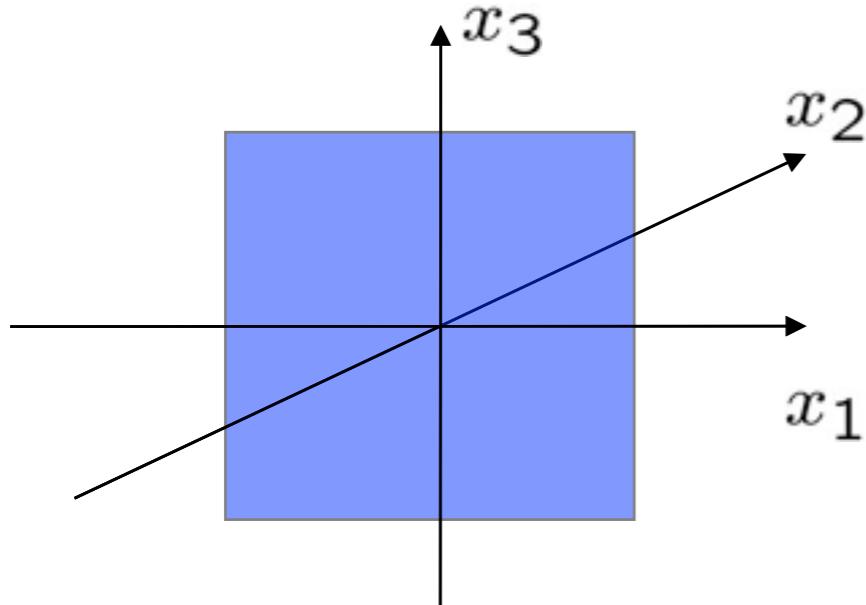


- **Theorem** (*Davies et al, Natarajan*)

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search



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

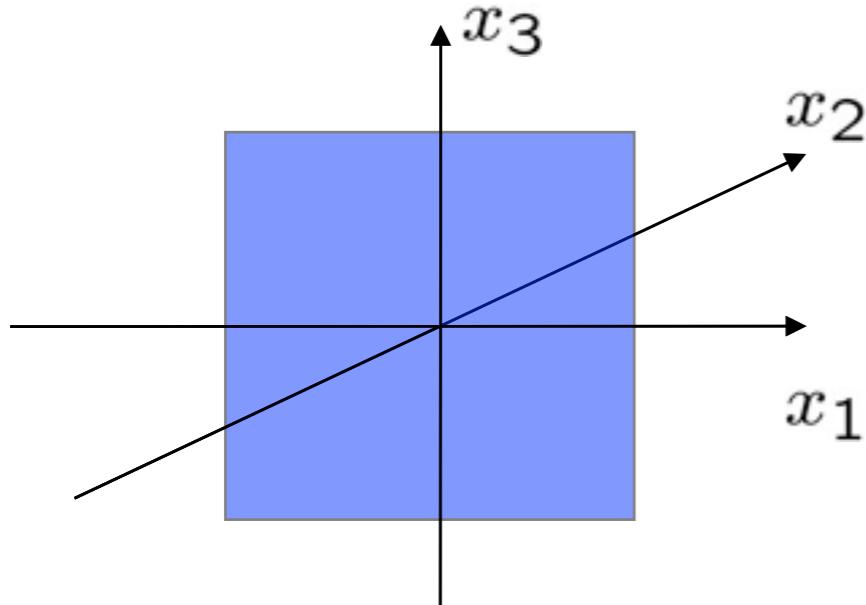
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \mathbf{b}$$

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

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search



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

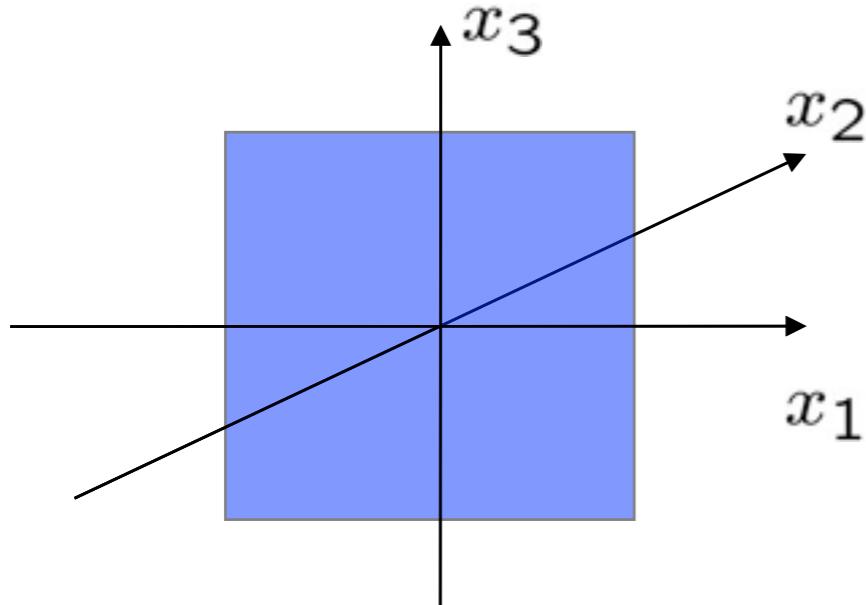
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ 0 \\ x_3 \end{pmatrix} = \mathbf{b}$$

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

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search



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

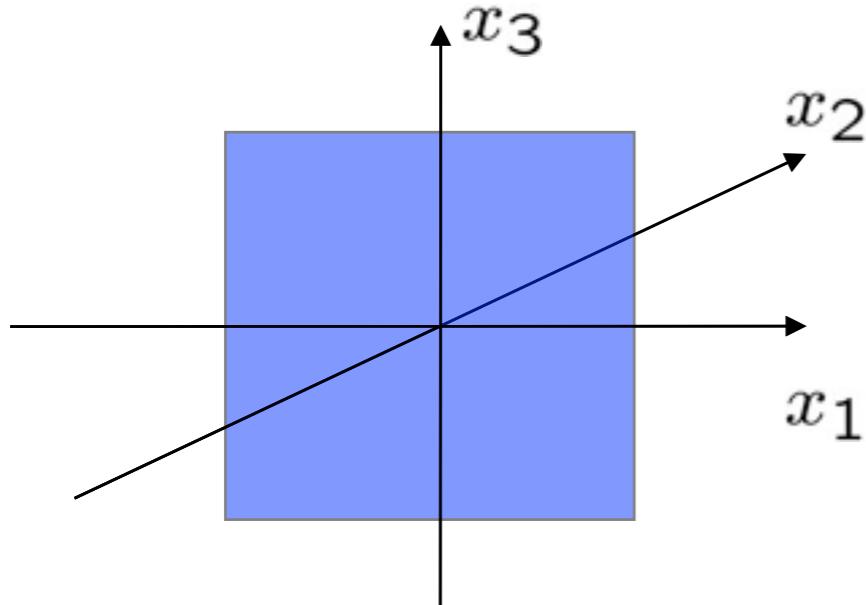
$$\begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{bmatrix} \cdot \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \mathbf{b}$$

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

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search



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

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

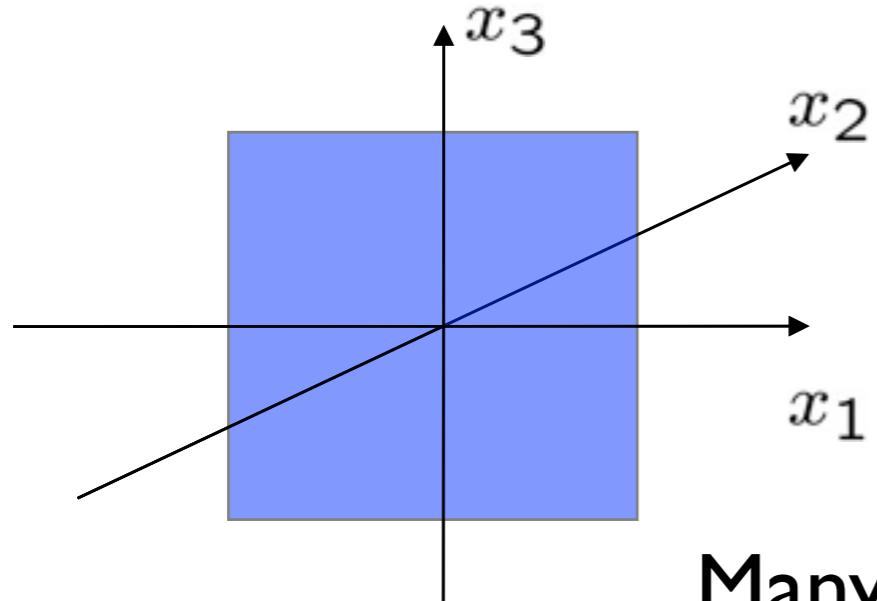
$$\begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{bmatrix}^{-1} \cdot \mathbf{b}$$

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

Solving the L0 optimization problem is **NP-complete**

# Complexity of Ideal Sparse Approximation

- Naive: Brute force search



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

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

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

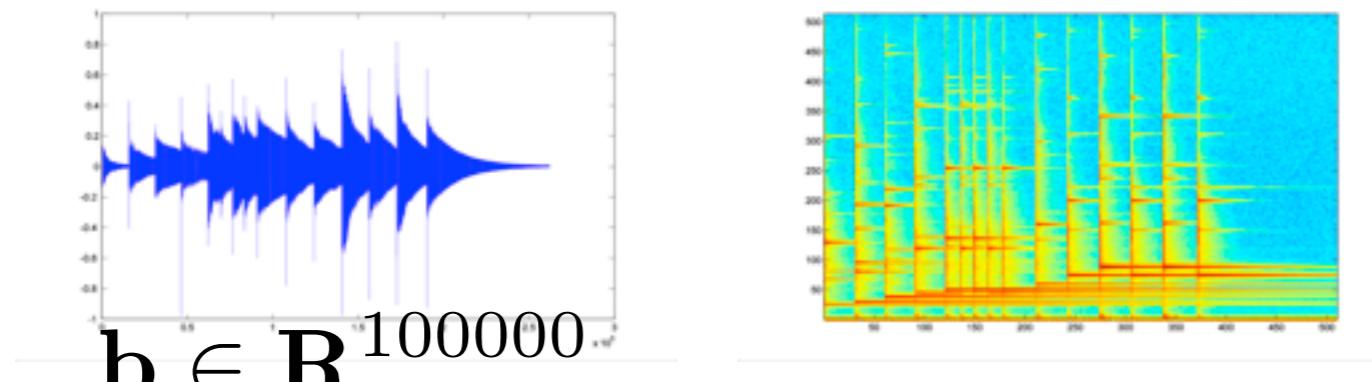
Many  $k$ -tuples to try!

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

Solving the L0 optimization problem is **NP-complete**

# Efficient Sparse Approximation With Time-Frequency Atoms

- Audio = superimposition of structures
- Example : glockenspiel



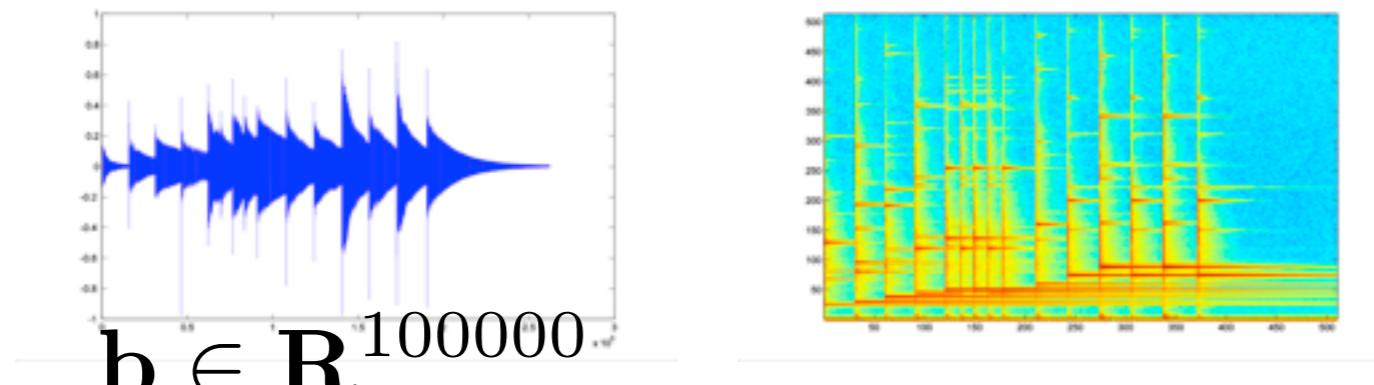
- ◆ transients = short, small scale
- ◆ harmonic part = long, large scale

- Two-layer sparse model with Gabor atoms

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

# Efficient Sparse Approximation With Time-Frequency Atoms

- Audio = superimposition of structures
- Example : glockenspiel



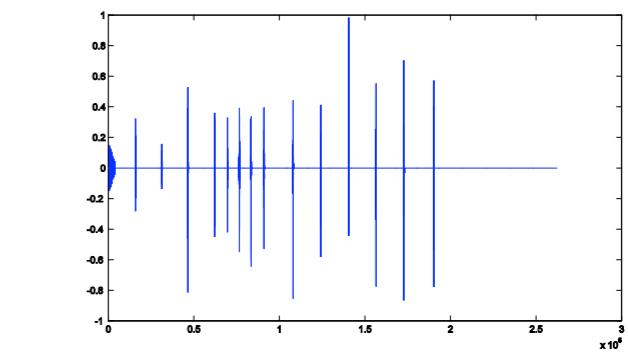
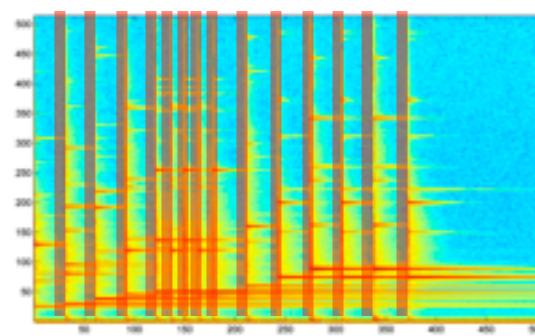
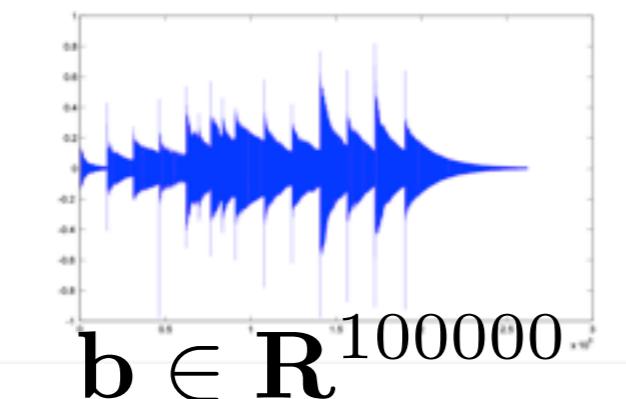
- ◆ transients = short, small scale
- ◆ harmonic part = long, large scale

- Two-layer sparse model with Gabor atoms

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

# Efficient Sparse Approximation With Time-Frequency Atoms

- Audio = superimposition of structures
- Example : glockenspiel



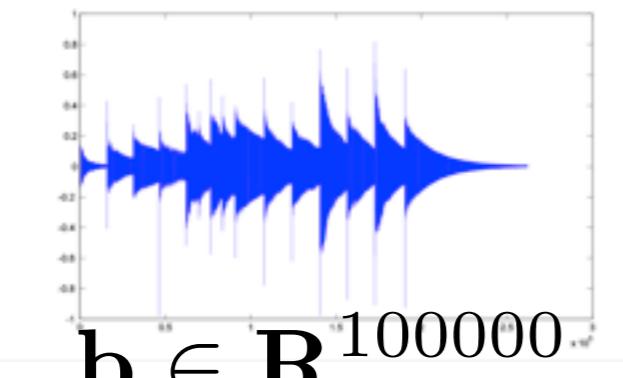
- ◆ transients = short, small scale
- ◆ harmonic part = long, large scale

- Two-layer sparse model with Gabor atoms

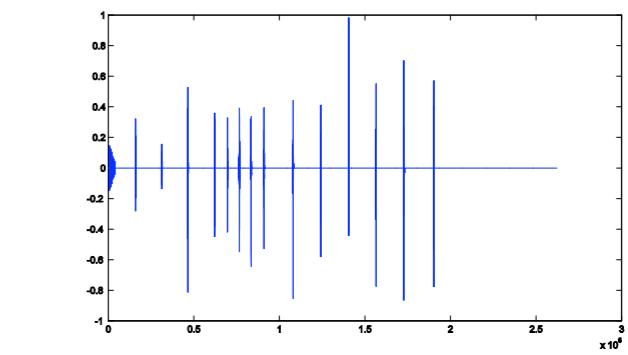
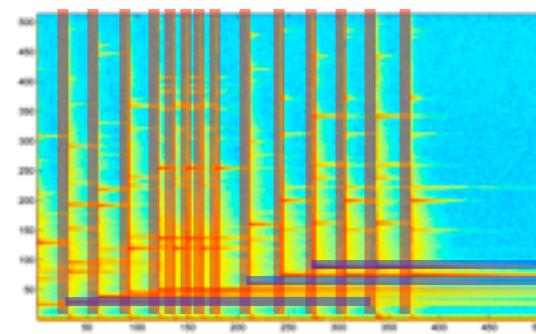
$$\left\{ \varphi_{s,\tau,f}(t) = \frac{1}{\sqrt{s}} w\left(\frac{t-\tau}{s}\right) e^{2i\pi f t} \right\}_{s,\tau,f} \quad b \approx \Phi_1 x_1 + \Phi_2 x_2$$

# Efficient Sparse Approximation With Time-Frequency Atoms

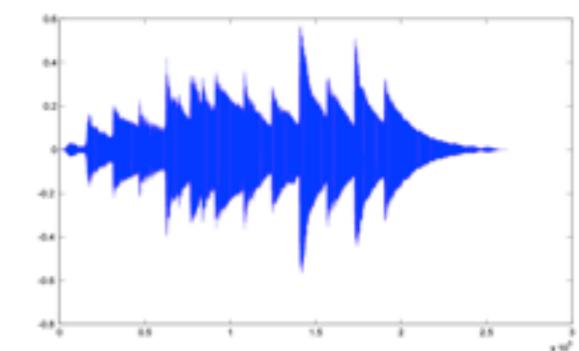
- Audio = superimposition of structures
- Example : glockenspiel



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



$\Phi_1 x_1$



$\Phi_2 x_2$

- Two-layer sparse model with Gabor atoms

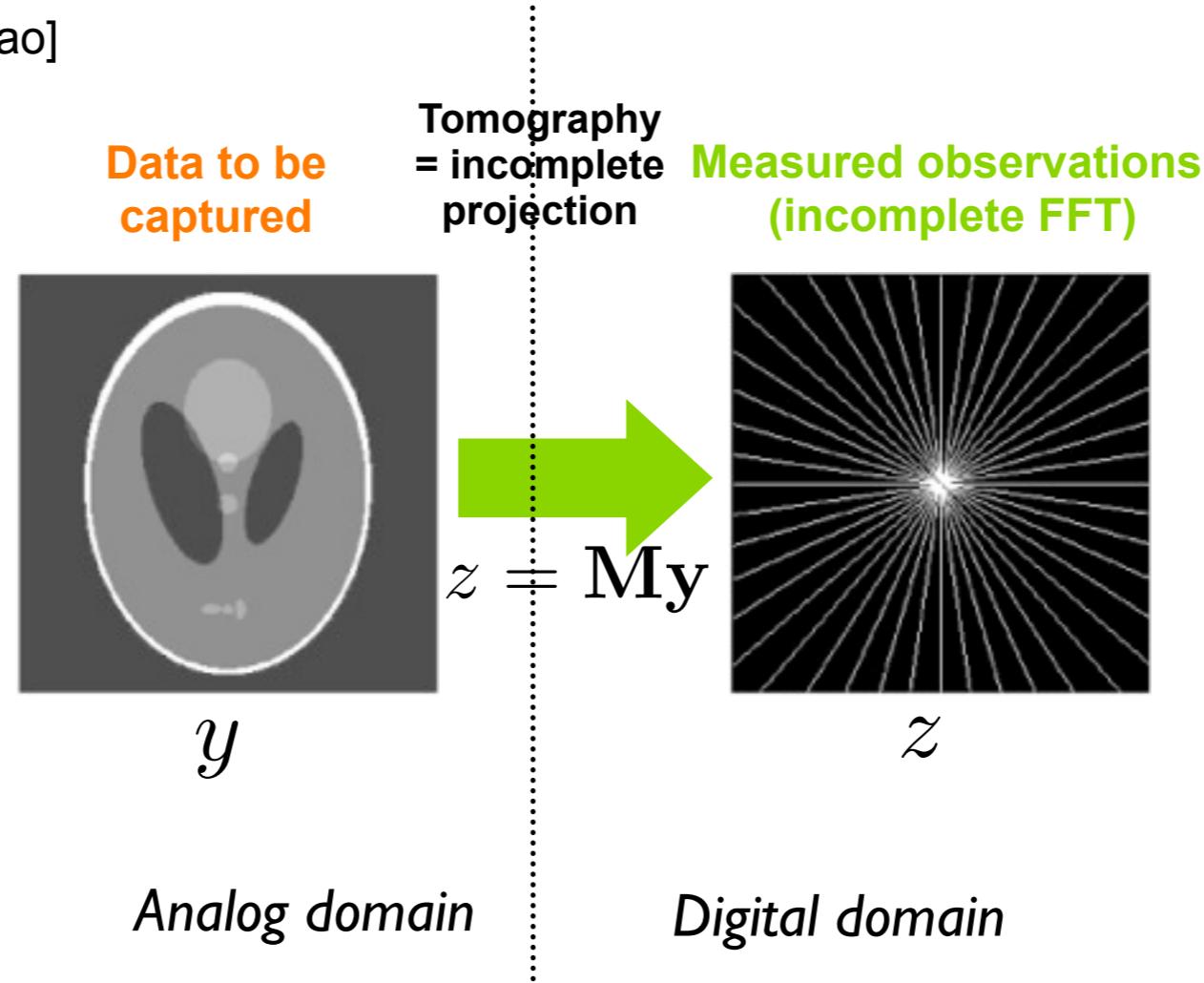
$$\left\{ \varphi_{s,\tau,f}(t) = \frac{1}{\sqrt{s}} w\left(\frac{t-\tau}{s}\right) e^{2i\pi f t} \right\}_{s,\tau,f}$$

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

# Example: tomography

- MRI from incomplete data

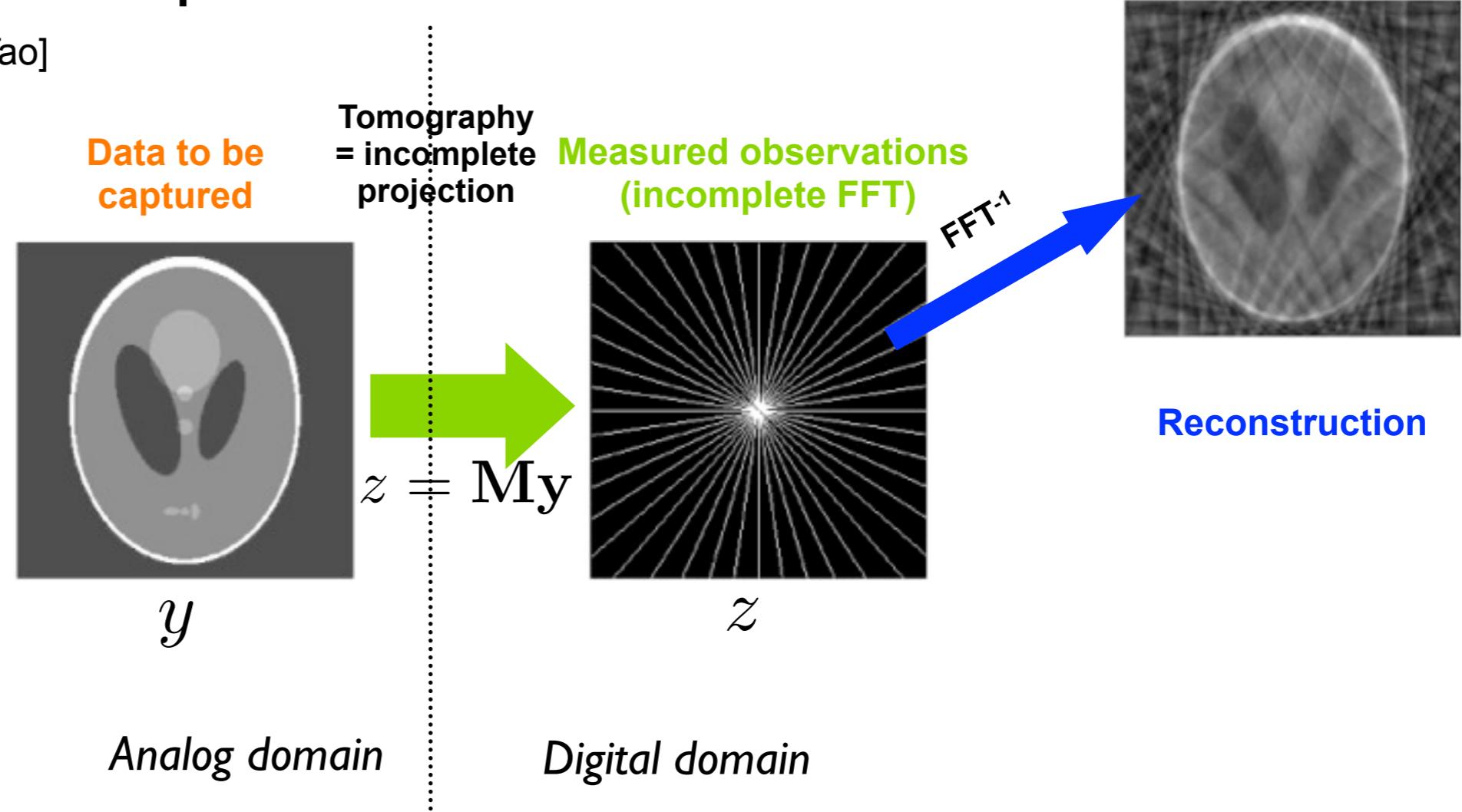
[Candès, Romberg & Tao]



# Example: tomography

- MRI from incomplete data

[Candès, Romberg & Tao]



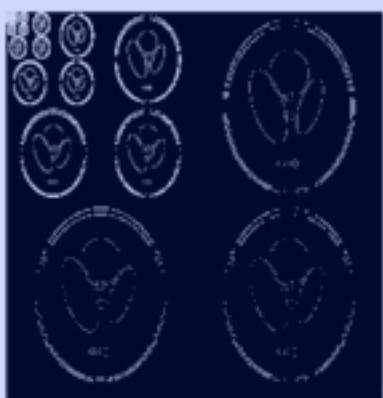
# Example: tomography

- MRI from incomplete data

[Candès, Romberg & Tao]

## Model / knowledge

The (unknown) wavelet transform is sparse



$x$

$$y = \Phi x$$

Data to be captured



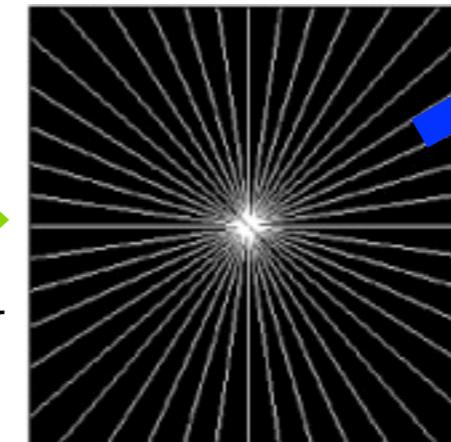
$y$

Tomography  
= incomplete projection

$$z = M y$$

Analog domain

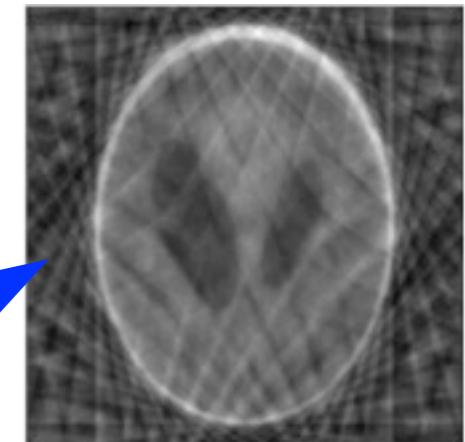
Measured observations  
(incomplete FFT)



$z$

Digital domain

$$\text{FFT}^{-1}$$



Reconstruction

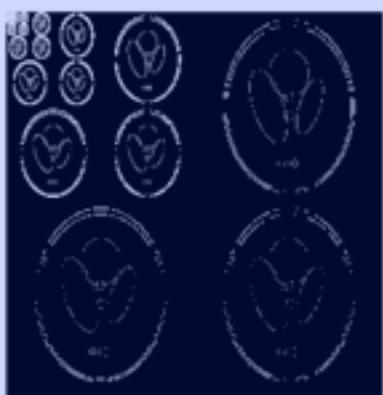
# Example: tomography

- MRI from incomplete data

[Candès, Romberg & Tao]

## Model / knowledge

The (unknown) wavelet transform is sparse



$x$

$$y = \Phi x$$

Data to be captured



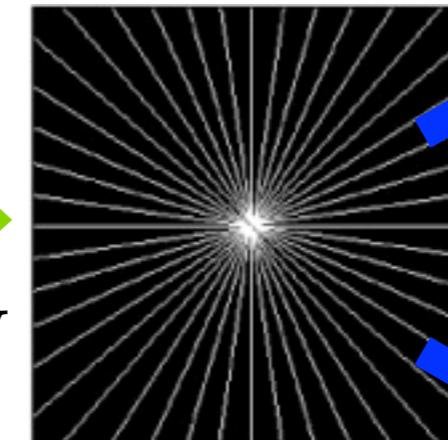
$y$

Analog domain

Tomography  
= incomplete projection

$$z = M y$$

Measured observations  
(incomplete FFT)

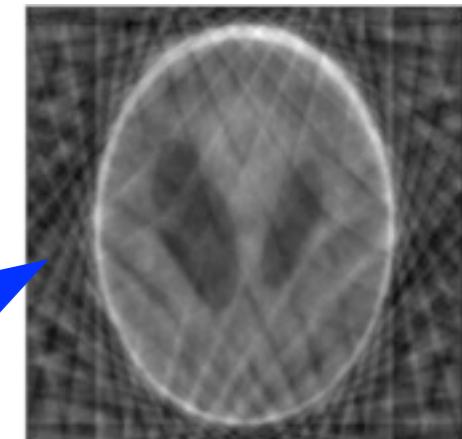


$z$

Digital domain

$$\text{FFT}^{-1}$$

Reconstruction



Sparse reconstruction  
(Candes et al 2004)



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

# Greedy Pursuit Algorithms

# Special case k=1

- Objective

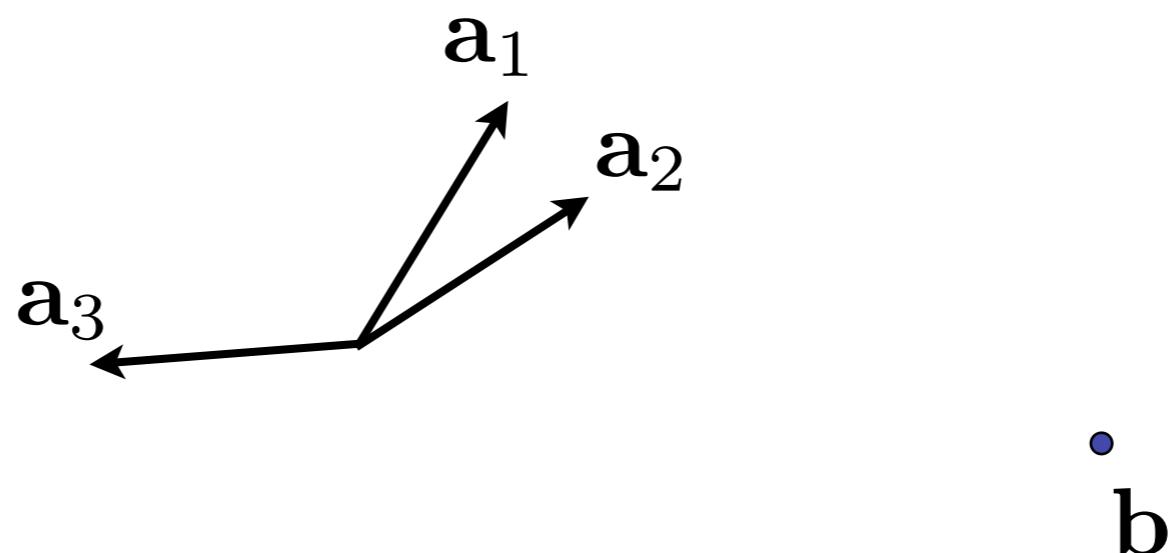
$$\min_x \|\mathbf{b} - \mathbf{A}x\|_2^2 \text{ s.t. } \|x\|_0 \leq 1$$

?

# Special case k=1

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

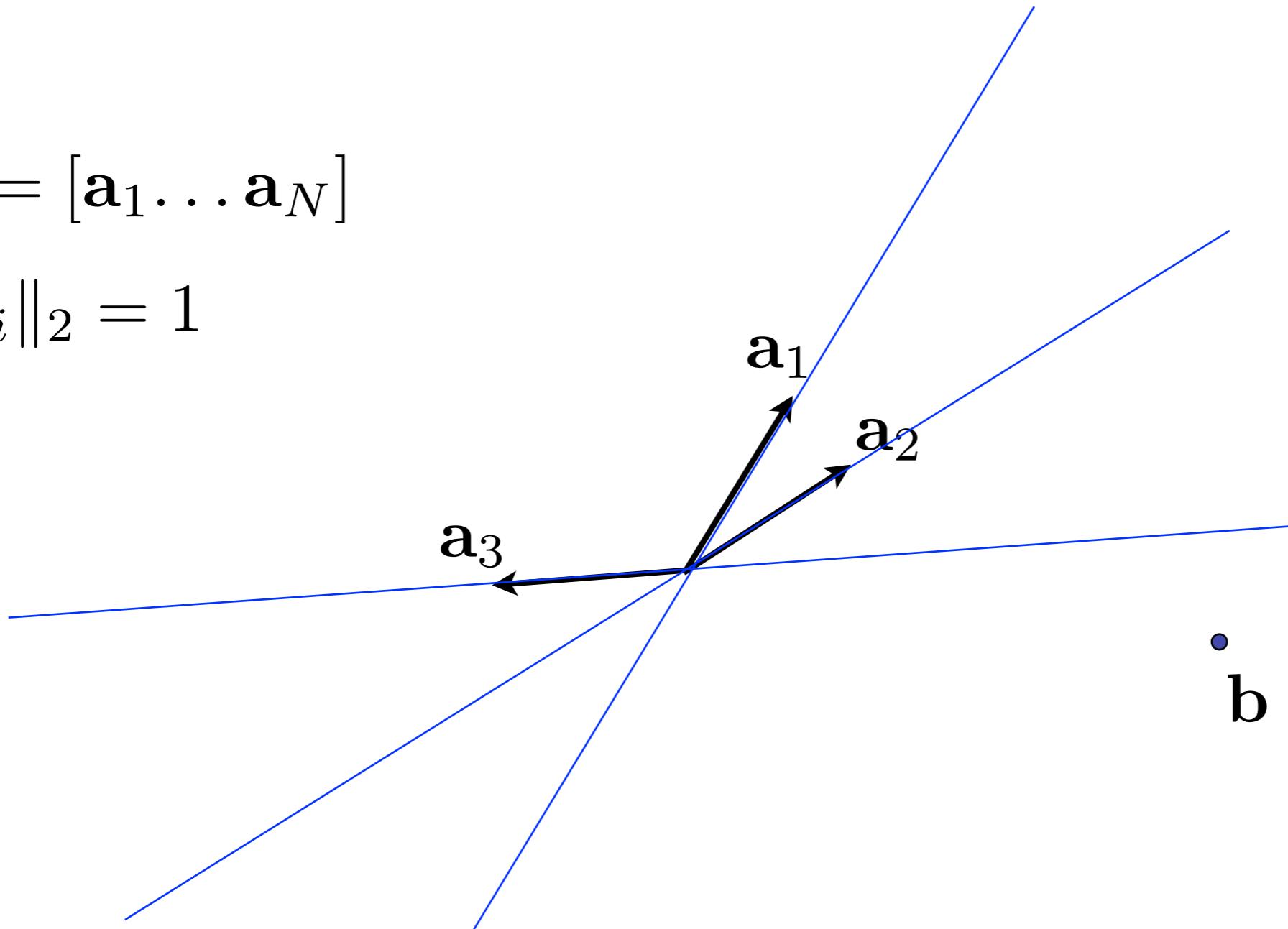
$$\|\mathbf{a}_i\|_2 = 1$$



# Special case k=1

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

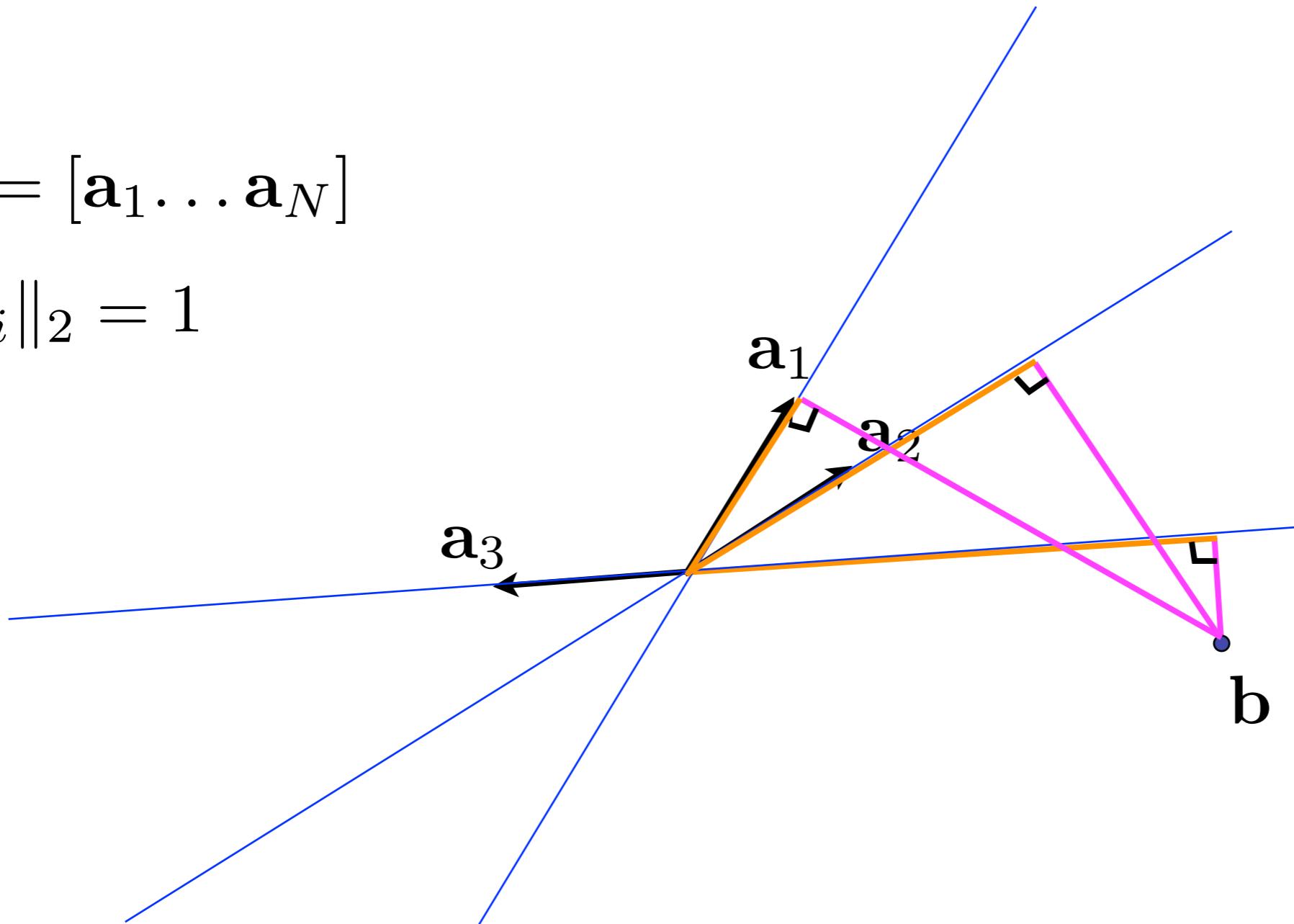
$$\|\mathbf{a}_i\|_2 = 1$$



# Special case k=1

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

$$\|\mathbf{a}_i\|_2 = 1$$



# Special case k=1

- Objective

$$\min_x \|\mathbf{b} - \mathbf{A}x\|_2^2 \text{ s.t. } \|x\|_0 \leq 1$$

- Assuming normalized columns:

- ✓ Best atom selection:

$$n^* = \arg \max_n |\mathbf{a}_n^T \mathbf{b}| ?$$

$$x_{n^*} = \mathbf{a}_{n^*}^T \mathbf{b}; \quad x_n = 0, n \neq n^*$$

# Special case k=1

- Objective

$$\min_x \|\mathbf{b} - \mathbf{A}x\|_2^2 \text{ s.t. } \|x\|_0 \leq 1$$

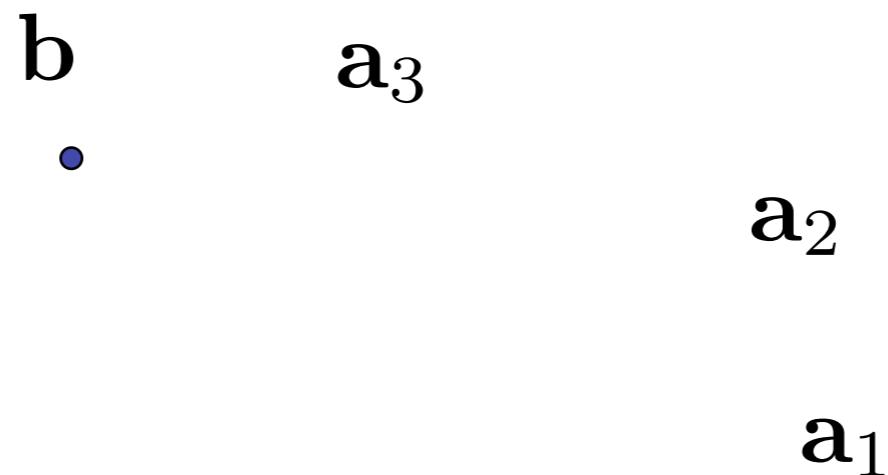
- Assuming normalized columns:

- ✓ Best atom selection:

$$n^* = \arg \max_n |\mathbf{a}_n^T \mathbf{b}|$$

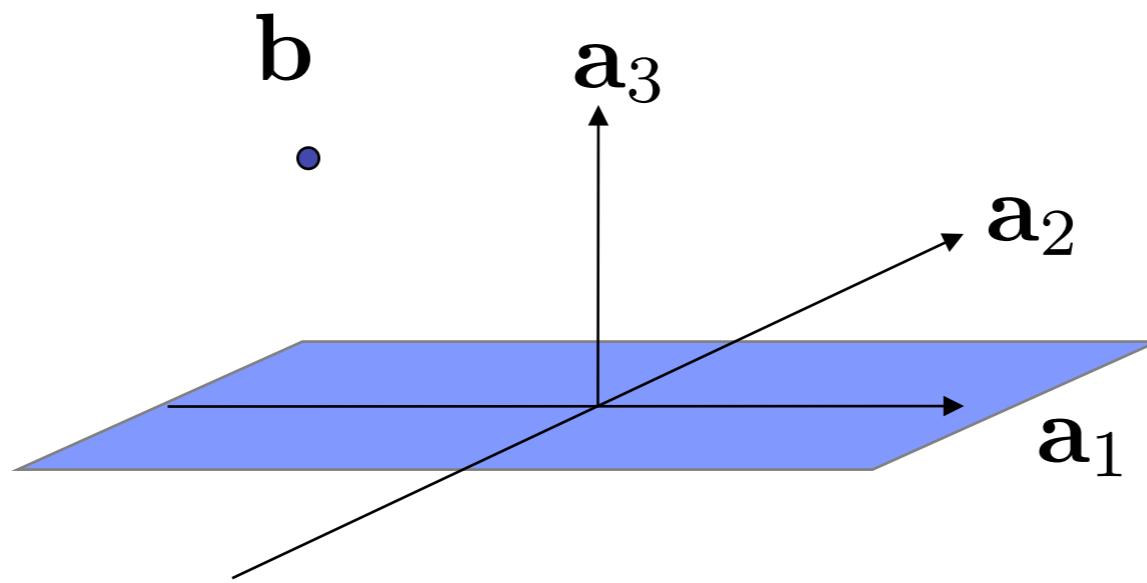
$$x_{n^*} = \mathbf{a}_{n^*}^T \mathbf{b}; \quad x_n = 0, n \neq n^*$$

# Special case: *canonical basis*, k=2



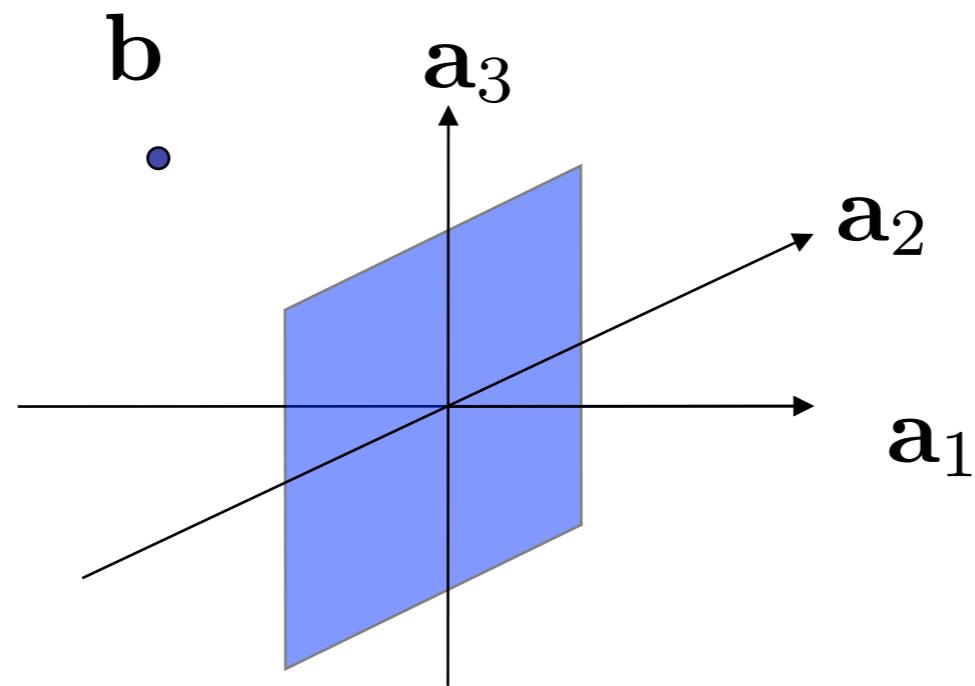
$$\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 = (b_1 - x_1)^2 + (b_2 - x_2)^2 + (b_3 - x_3)^2$$

## Special case: *canonical basis*, k=2



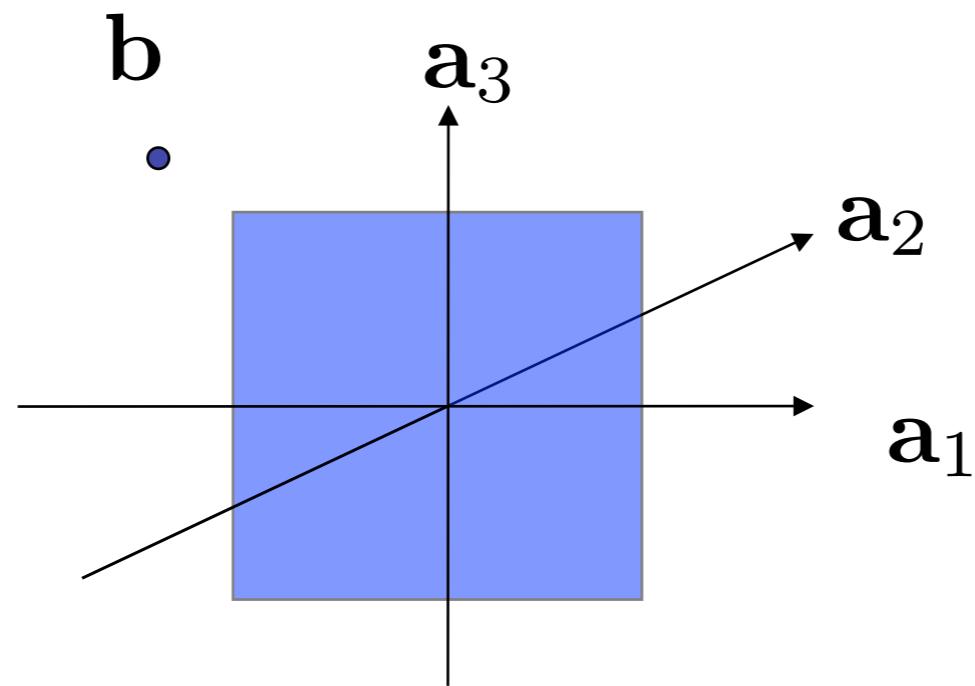
$$\|\mathbf{b} - \mathbf{Ax}\|_2^2 = (b_1 - x_1)^2 + (b_2 - x_2)^2 + (b_3 - x_3)^2$$

## Special case: *canonical basis*, k=2



$$\|\mathbf{b} - \mathbf{Ax}\|_2^2 = (b_1 - x_1)^2 + (b_2 - x_2)^2 + (b_3 - x_3)^2$$

## Special case: *canonical basis*, k=2



$$\|\mathbf{b} - \mathbf{Ax}\|_2^2 = (b_1 - x_1)^2 + (b_2 - x_2)^2 + (b_3 - x_3)^2$$

# Special case: $\mathbf{A}$ is orthonormal

- Assumption :  $m=N$  and  $\mathbf{A}$  is *orthonormal*

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

$$\begin{aligned}\|\mathbf{b} - \mathbf{A}x\|_2^2 &= \|\sum_{i=1}^N (\mathbf{a}_i^T \mathbf{b} - x_i) \mathbf{a}_i\|_2^2 \\ &= \sum_{i=1}^N (\mathbf{a}_i^T \mathbf{b} - x_i)^2\end{aligned}$$

# The orthonormal case: thresholding

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

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

- ✓ is equivalent to

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

- Let  $\Lambda_k$  index the  $k$  largest inner products

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

- ✓ an optimum solution is

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

# Matching Pursuit (MP)

- Iterative algorithm (aka Projection Pursuit, CLEAN)

- ✓ Initialization  $\mathbf{r}_0 = \mathbf{b}$   $i = 1$
- ✓ Atom selection:

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

- ✓ Residual update

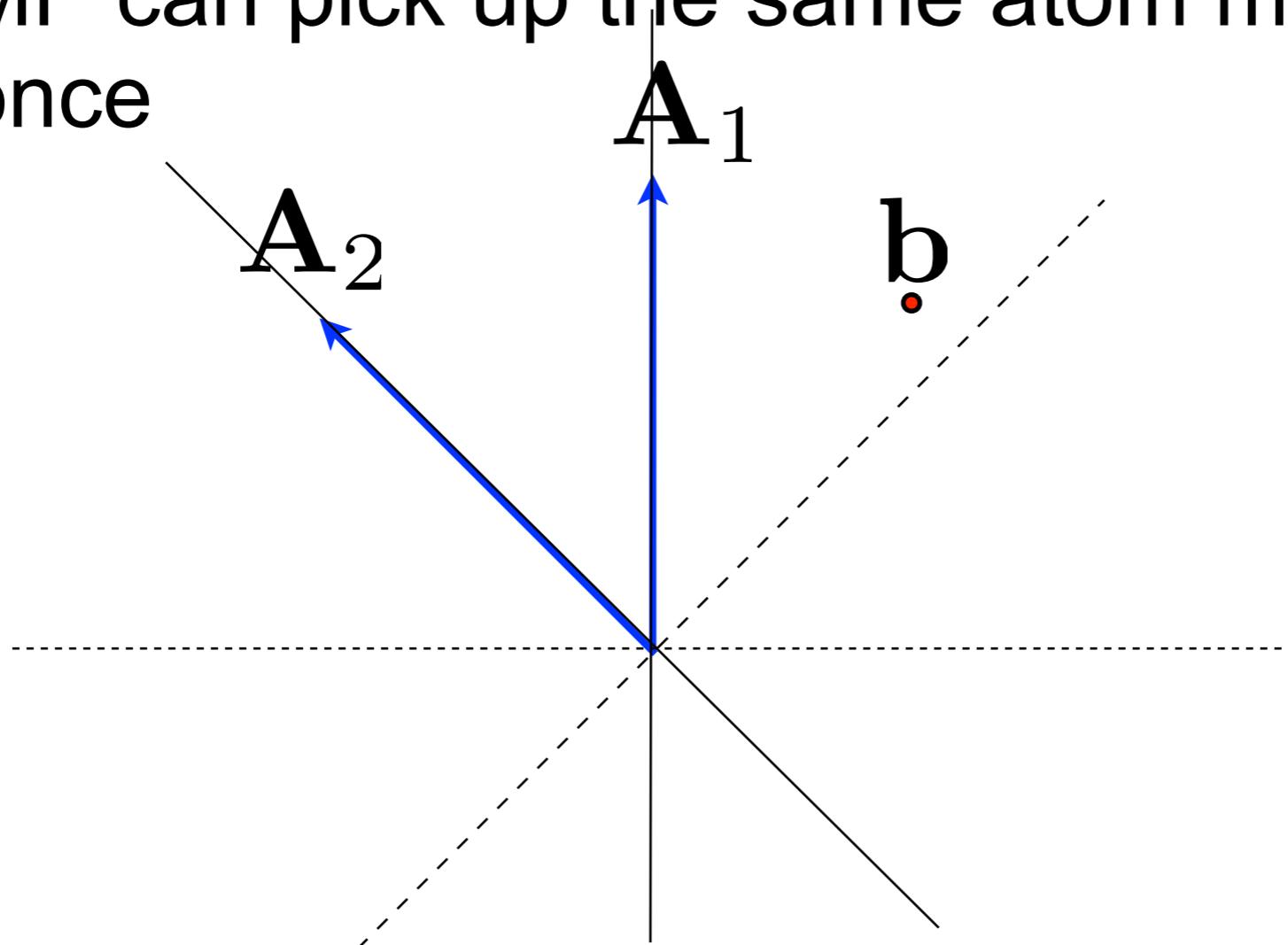
$$\mathbf{r}_i = \mathbf{r}_{i-1} - (\mathbf{a}_{n_i}^T \mathbf{r}_{i-1}) \mathbf{a}_{n_i}$$

- Sparse approximation after k steps

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

# Caveats (1)

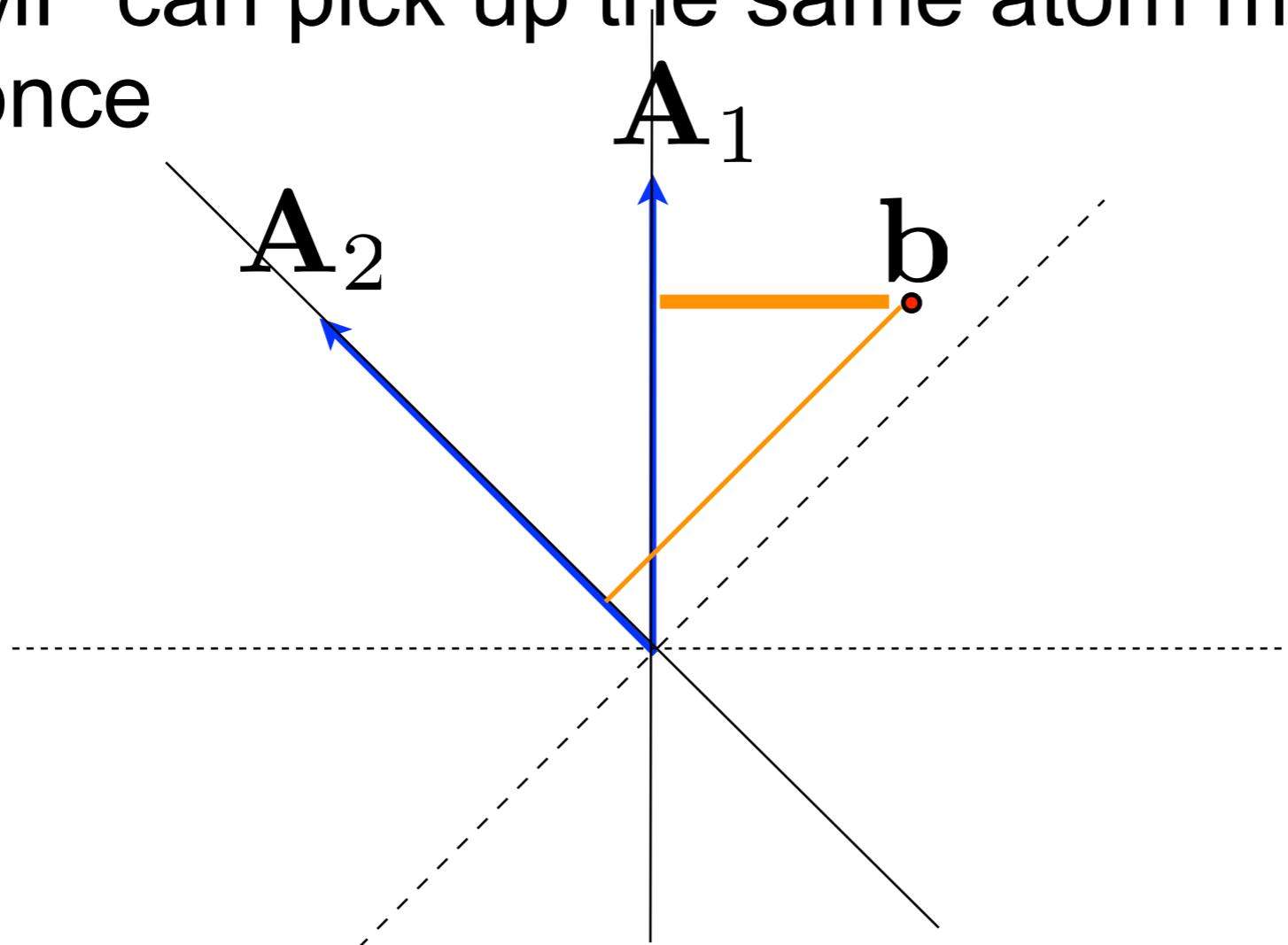
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

# Caveats (1)

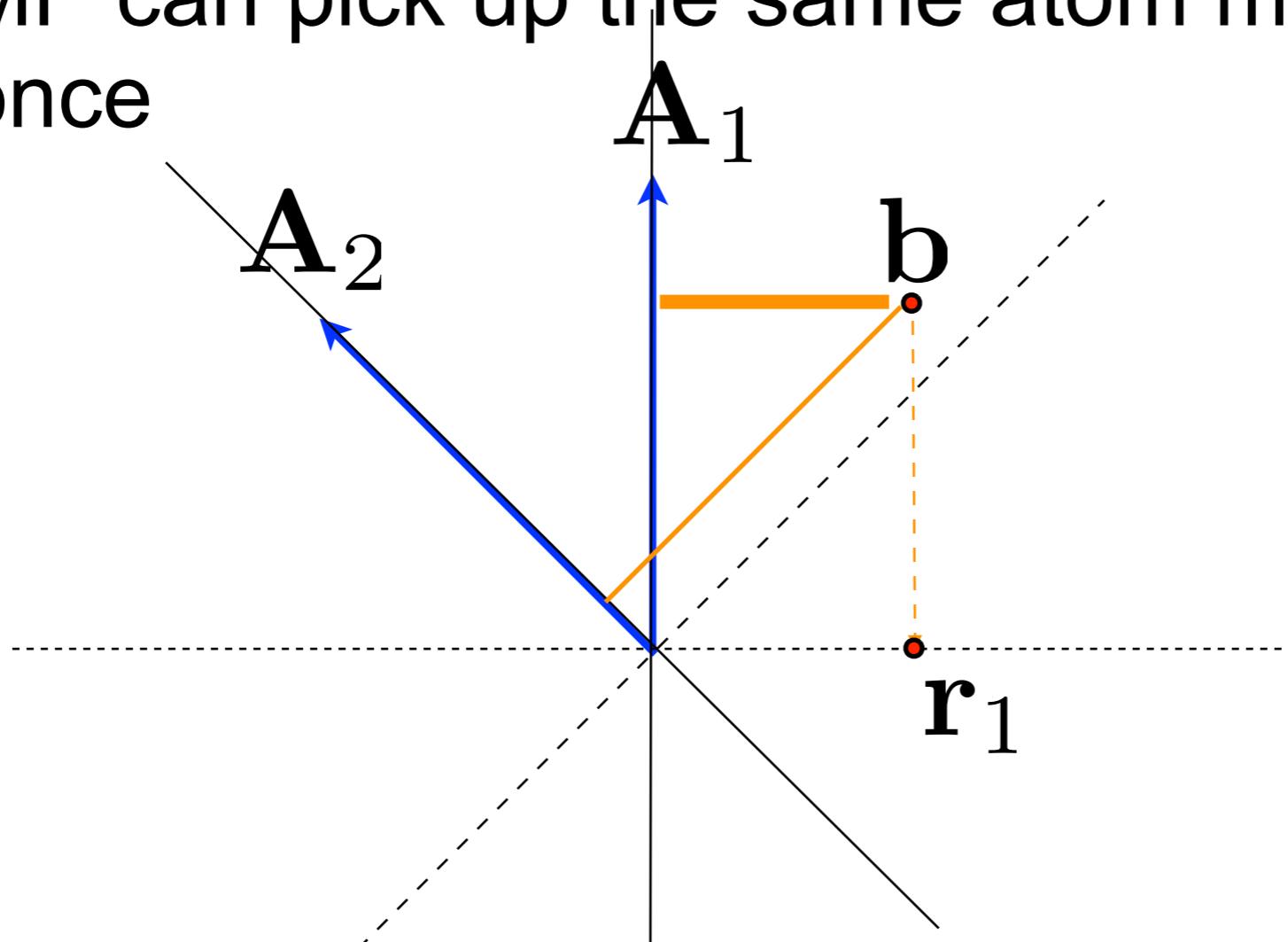
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

# Caveats (1)

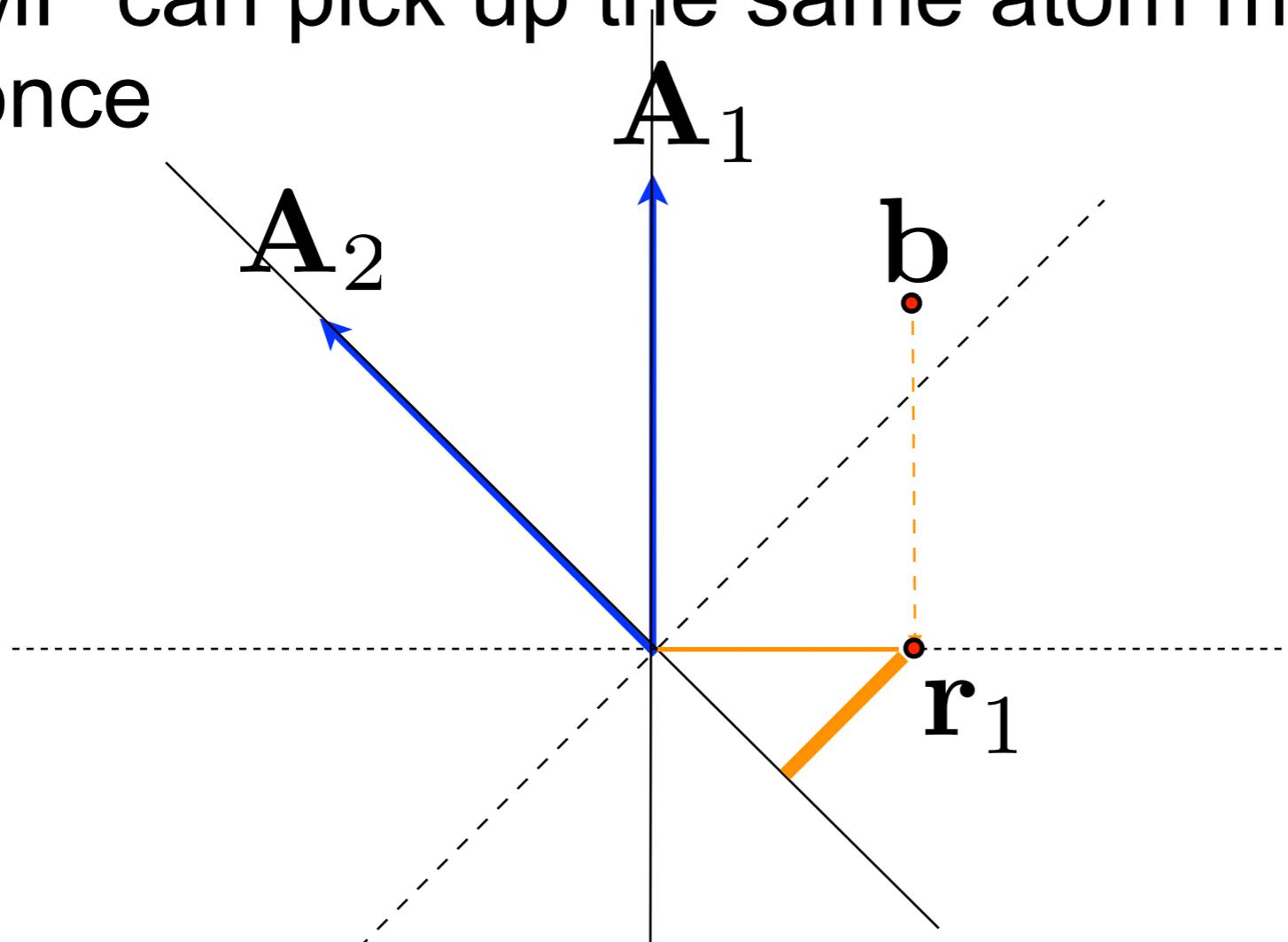
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

# Caveats (1)

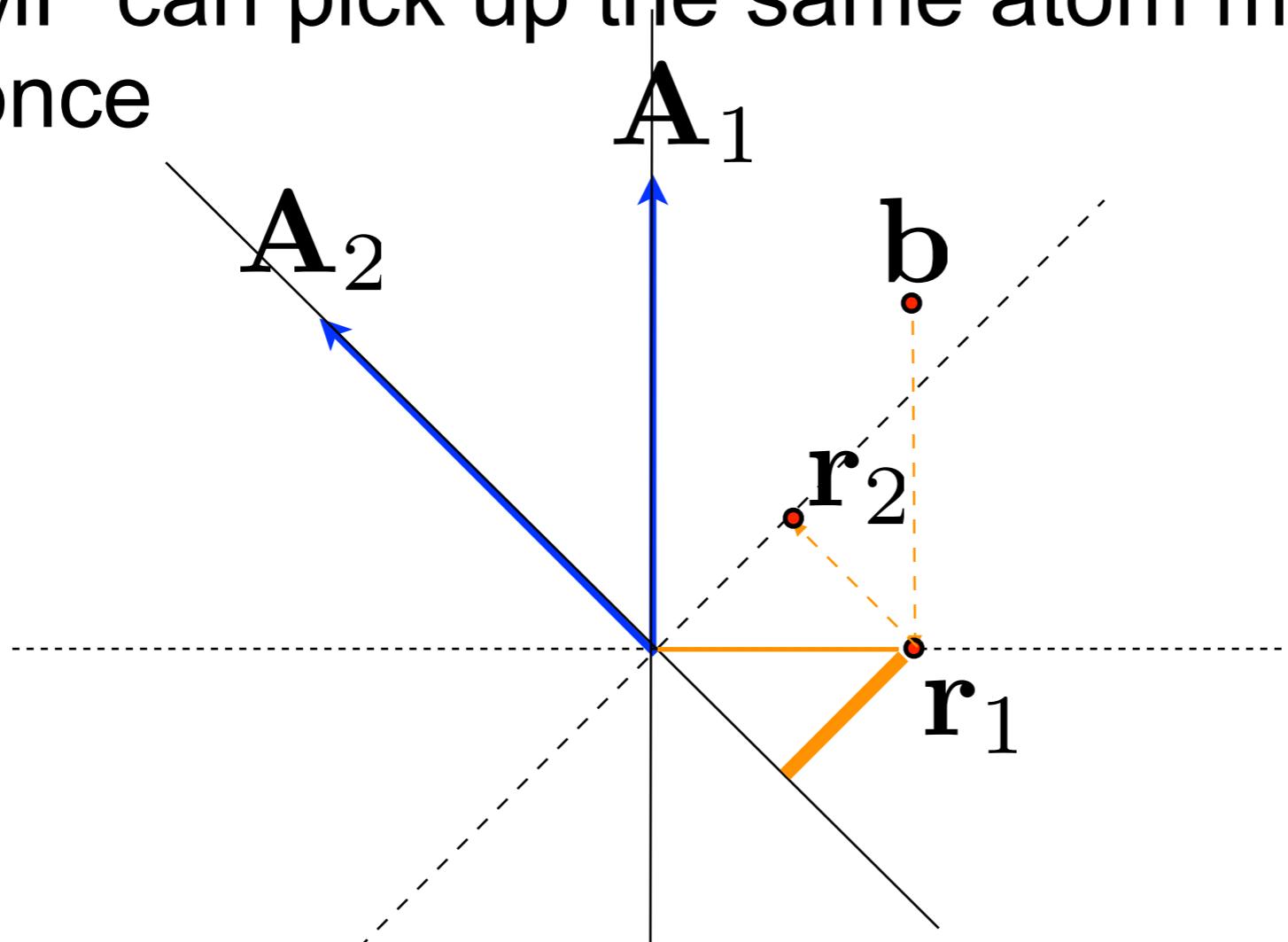
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

# Caveats (1)

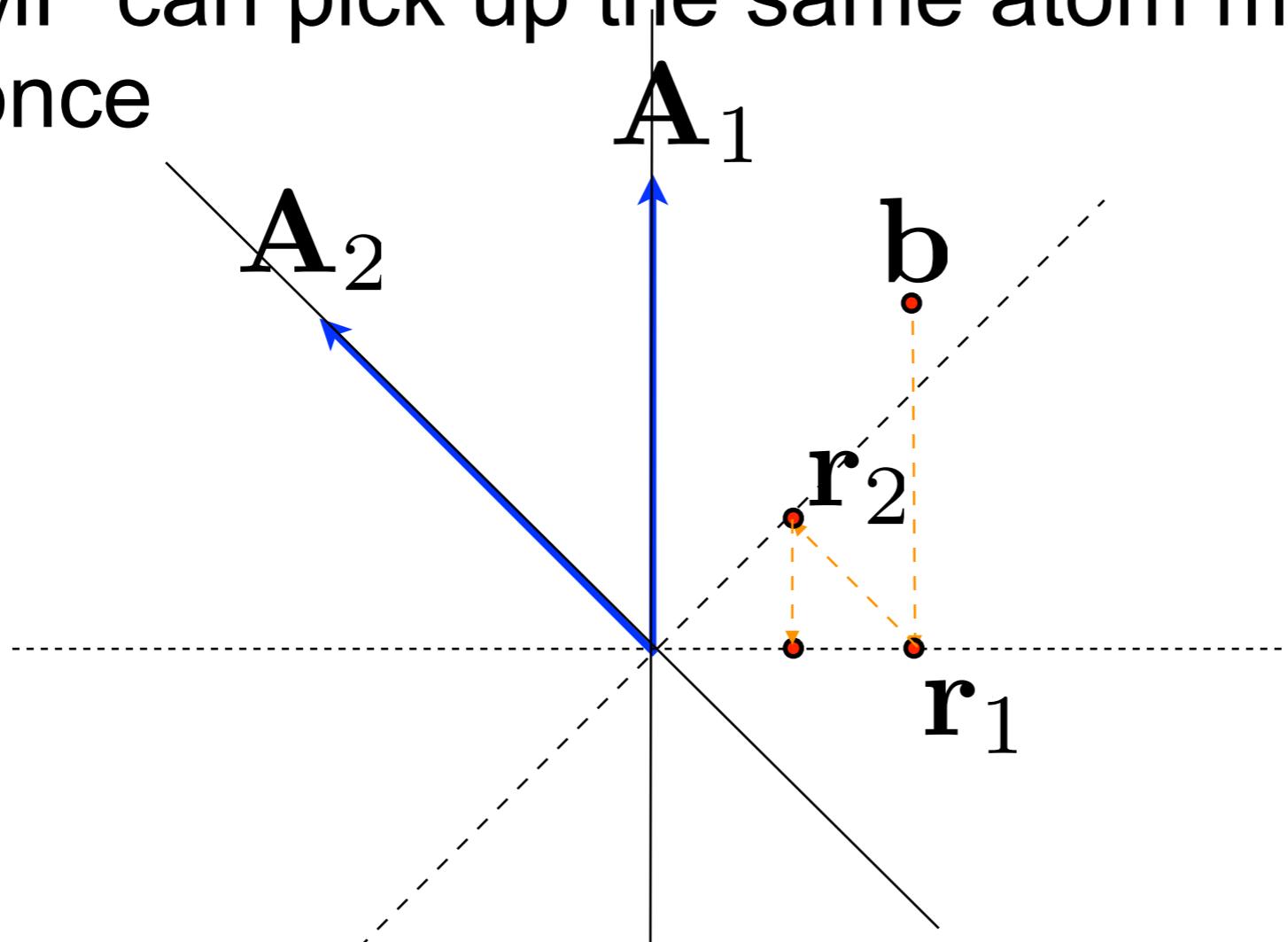
- MP can pick up the same atom more than once



- OMP will never select twice the same atom

# Caveats (1)

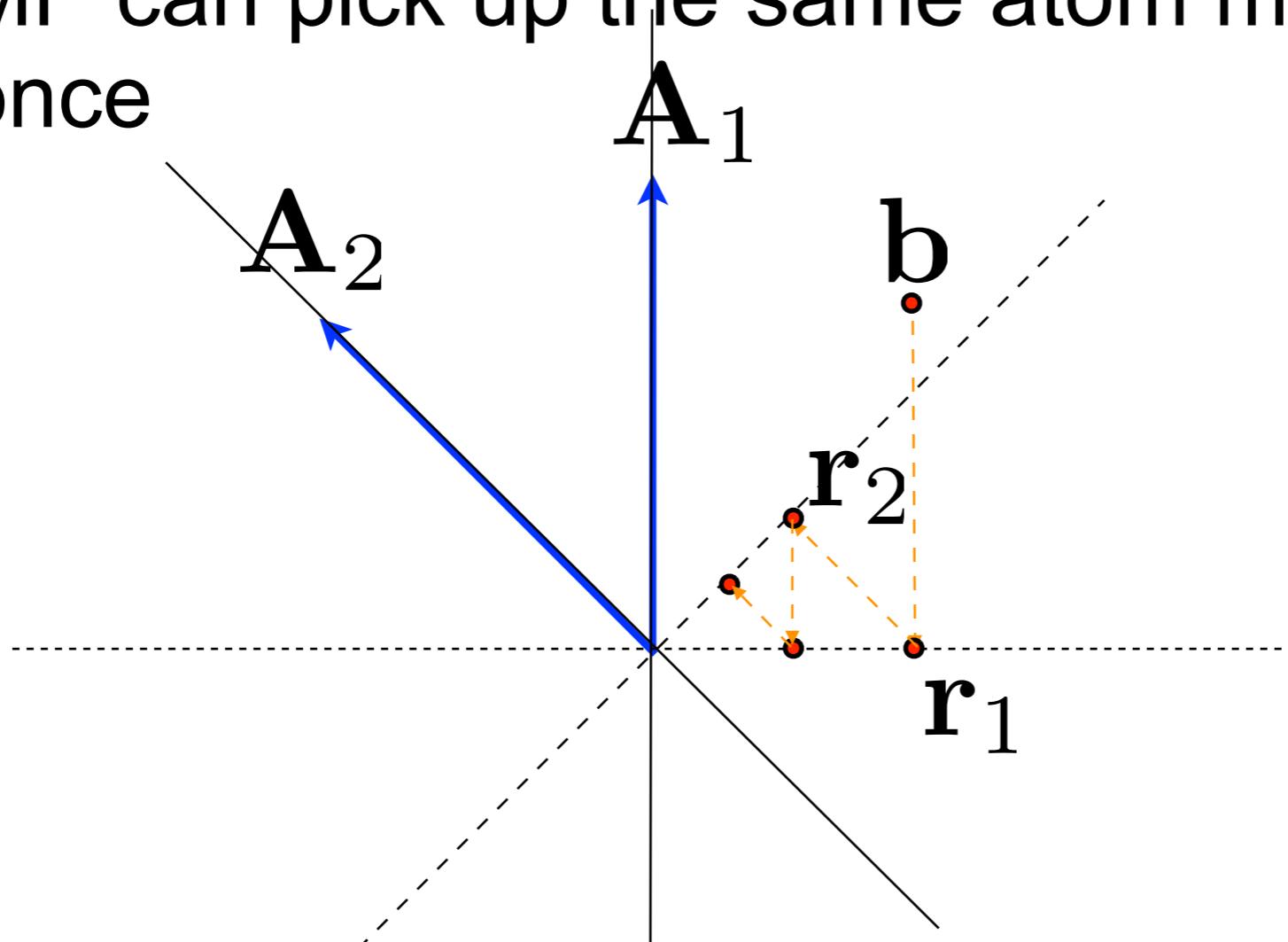
- MP can pick up the same atom more than once



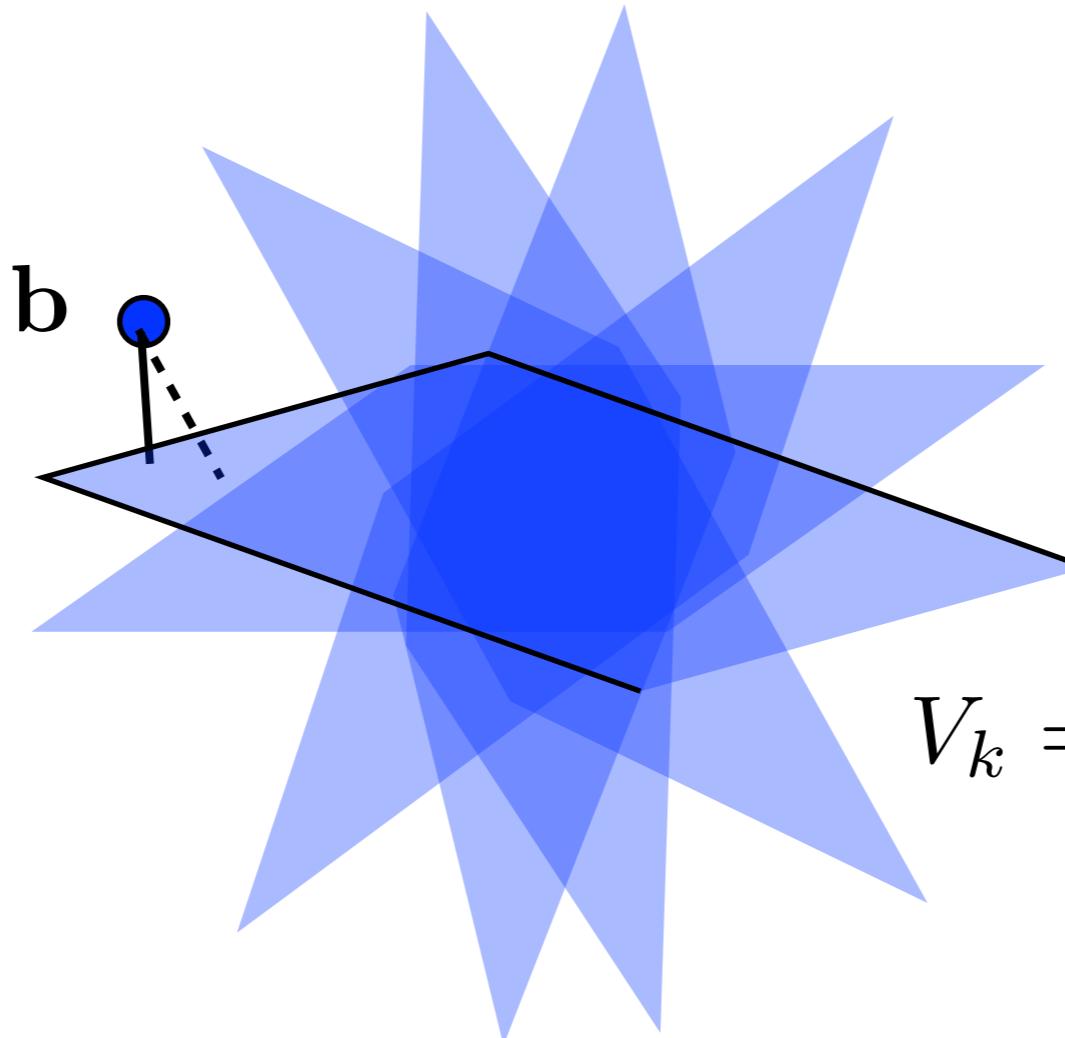
- OMP will never select twice the same atom

# Caveats (1)

- MP can pick up the same atom more than once



- OMP will never select twice the same atom

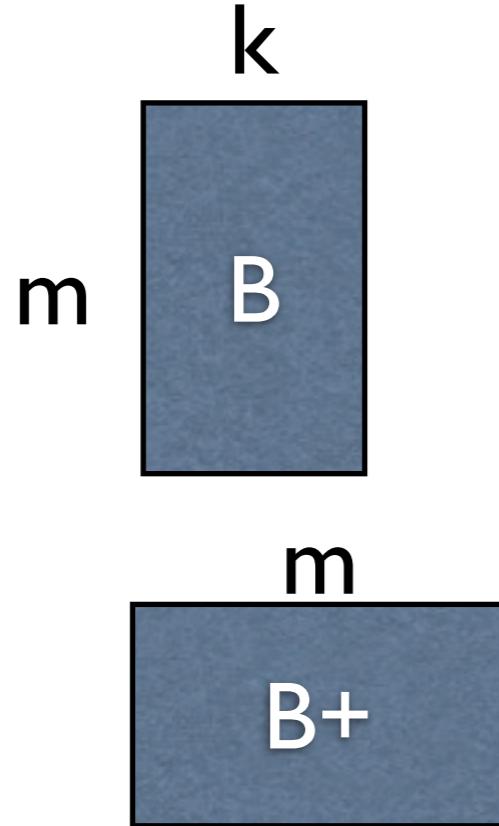


$$V_k = \text{span}(\mathbf{a}_{n_i}, 1 \leq i \leq k)$$

# Orthogonal projection onto a subspace

# Pseudo-inversion

- Over-determined case:  $m > k$

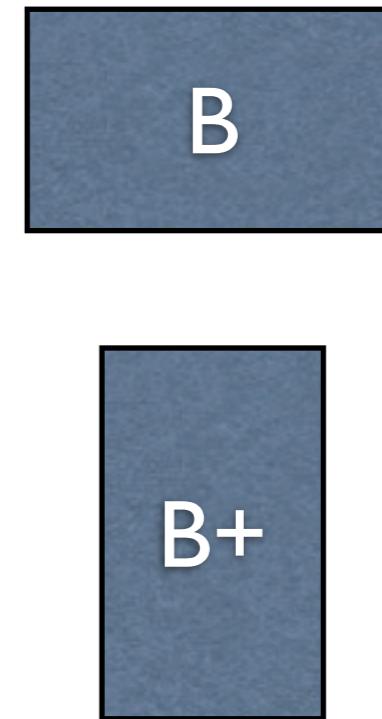


$$B^+ = (B^T B)^{-1} B^T$$

$$B^+ B = \text{Id}_k$$

$$B B^+ = P_{\text{span}(B)}$$

- Under-determined case  $m < k$



$$B^+ = B^T (B B^T)^{-1}$$

$$B B^+ = \text{Id}_m$$

# Orthonormal MP (OMP)

[Mallat & Zhang 93, Pati & al 94]

- Observation: after  $k$  iterations
- Approximant belongs to

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

- Best approximation from  $V_k$  = orthoprojection

$$P_{V_k} \mathbf{b} = \mathbf{A}_{\Lambda_k} \mathbf{A}_{\Lambda_k}^+ \mathbf{b} \quad \text{pseudo-inverse}$$

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

- OMP residual update rule

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

# OMP

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

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

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

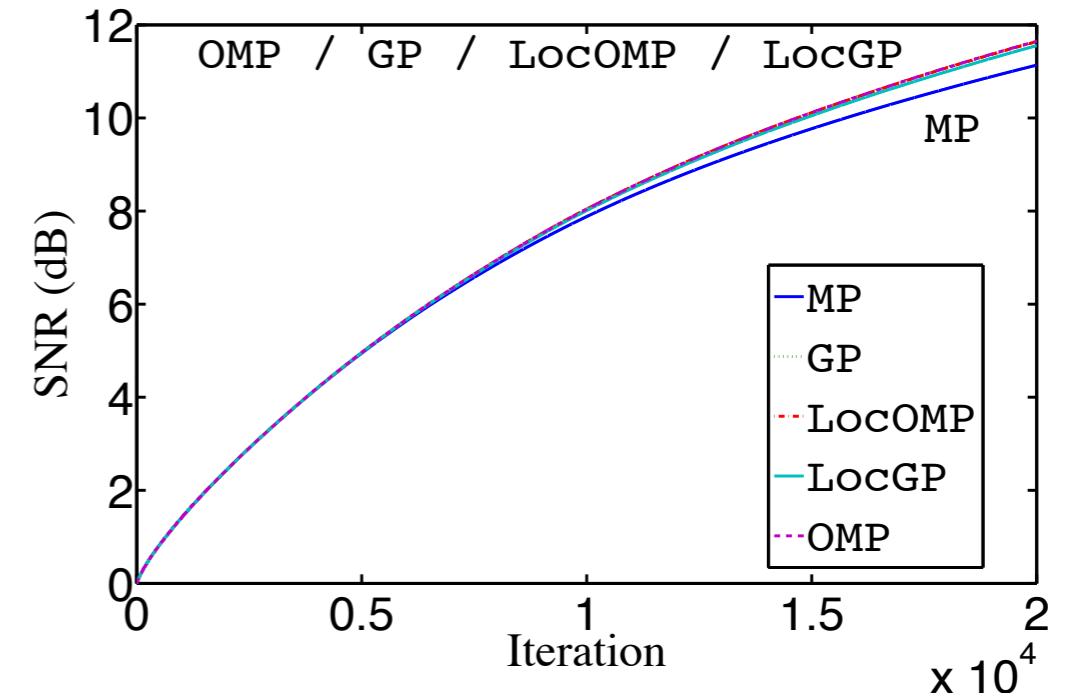
$$V_i = \text{span}(\mathbf{a}_n, n \in \Lambda_i)$$

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

# OMP versus MP

- SNR as a function of iteration number

$$\text{SNR} = 10 \log_{10} \frac{\|\mathbf{b}\|_2^2}{\|\mathbf{r}_i\|_2^2}$$



# 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

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

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

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

# Exercice

- Write Matlab pseudo-code for MP
- Idem for OMP

# Exercice: Matlab code for (O)MP

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

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

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