“Overcomplete Sparse Decomposition”

Massoud BABAIE-ZADEH

Signal Decomposition

- Decomposition of a signal x(t) as a linear combination of a set of known signals:

$$x(t) = \alpha_1 \varphi_1(t) + \cdots + \alpha_m \varphi_m(t)$$

- Examples:
  - Fourier Transform ($\varphi_i \rightarrow$ complex sinusoids)
  - Wavelet Transform
  - DCT
  - …

Signal Decomposition

- Decomposition of a signal x(t) as a linear combination of a set of known signals:

$$x(t) = \alpha_1 \varphi_1(t) + \cdots + \alpha_M \varphi_M(t)$$

- Terminology:
  - Atomic Decomposition (=Signal Decomposition)
  - Atoms → φ_i
  - Dictionary → Set of all atoms: $\{\varphi_1, \varphi_2, \ldots\}$

Discrete Case

$$x(t) = \alpha_1\,\varphi_1(t) + \cdots + \alpha_M\,\varphi_M(t), \qquad t = 1,\dots,N$$

$$\begin{array}{c} \\ \text{Time} \\ \\ \downarrow \\ \\ \end{array} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(N) \end{bmatrix} = \alpha_1 \begin{bmatrix} \varphi_1(1) \\ \varphi_1(2) \\ \varphi_1(3) \\ \vdots \\ \varphi_1(N) \end{bmatrix} + \cdots + \alpha_M \begin{bmatrix} \varphi_M(1) \\ \varphi_M(2) \\ \varphi_M(3) \\ \vdots \\ \varphi_M(N) \end{bmatrix}$$

$$\mathbf{x} \quad = \alpha_1 \quad \underline{\varphi}_1 \quad + \cdots + \quad \alpha_M \quad \underline{\varphi}_M$$

Matrix form

$$\begin{array}{c} \text{Time} \\ \\ \downarrow \end{array} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(N) \end{bmatrix} = \alpha_1 \begin{bmatrix} \varphi_1(1) \\ \varphi_1(2) \\ \varphi_1(3) \\ \vdots \\ \varphi_1(N) \end{bmatrix} + \cdots + \alpha_M \begin{bmatrix} \varphi_M(1) \\ \varphi_M(2) \\ \varphi_M(3) \\ \vdots \\ \varphi_M(N) \end{bmatrix}$$

$$\mathbf{x} = \alpha_1 \underline{\varphi}_1 + \cdots + \alpha_M \underline{\varphi}_M$$

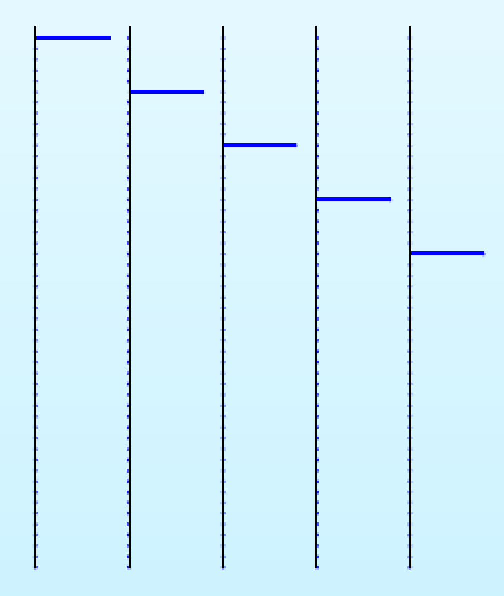
$$\mathbf{x} = \begin{bmatrix} \varphi_1 & \cdots & \varphi_M \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_M \end{bmatrix} \rightarrow \boxed{\underset{N\times M}{\mathbf{\Phi}}\,\underset{M\times 1}{\boldsymbol{\alpha}} = \underset{N\times 1}{\mathbf{x}}}$$

Complete decomposition: M=N

$$\begin{array}{c} \text{Time} \\ \downarrow \end{array} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(N) \end{bmatrix} = \alpha_1 \begin{bmatrix} \varphi_1(1) \\ \varphi_1(2) \\ \varphi_1(3) \\ \vdots \\ \varphi_1(N) \end{bmatrix} + \cdots + \alpha_M \begin{bmatrix} \varphi_M(1) \\ \varphi_M(2) \\ \varphi_M(3) \\ \vdots \\ \varphi_M(N) \end{bmatrix}$$

$$\mathbf{x} = \alpha_1 \underline{\varphi}_1 + \cdots + \alpha_M \underline{\varphi}_M$$

- **M=N** → Complete dictionary → Unique set of coefficients
- Examples: Dirac dictionary, Fourier Dictionary



Dirac Dictionary:

$$\underline{\varphi}_k(n) = \begin{cases} 1 & n = k \\ 0 & n \neq k \end{cases}$$

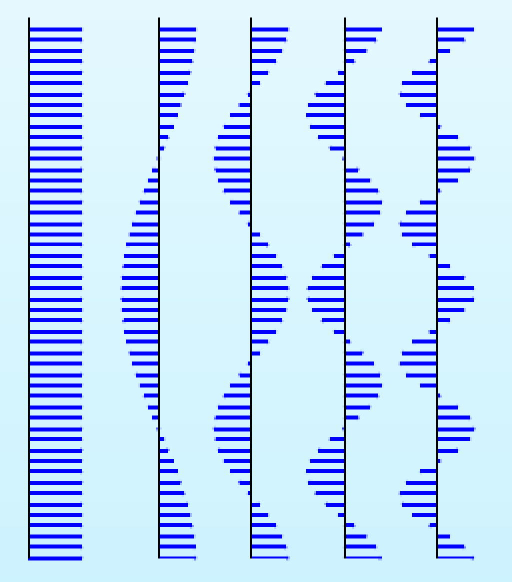
$$\Rightarrow \alpha_k = x(k)$$

Complete decomposition: M=N

$$\begin{array}{c} \text{Time} \\ \downarrow \end{array} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(N) \end{bmatrix} = \alpha_1 \begin{bmatrix} \varphi_1(1) \\ \varphi_1(2) \\ \varphi_1(3) \\ \vdots \\ \varphi_1(N) \end{bmatrix} + \cdots + \alpha_M \begin{bmatrix} \varphi_M(1) \\ \varphi_M(2) \\ \varphi_M(3) \\ \vdots \\ \varphi_M(N) \end{bmatrix}$$

$$\mathbf{x} = \alpha_1 \underline{\varphi}_1 + \cdots + \alpha_M \underline{\varphi}_M$$

- **M=N** → Complete dictionary → Unique set of coefficients
- Examples: Dirac dictionary, Fourier Dictionary



Fourier Dictionary:

$$\underline{\varphi}_k = \left(1,\ e^{\frac{2k\pi}{N}},\ e^{\frac{2k\pi}{N}2}, \ldots,\ e^{\frac{2k\pi}{N}(N-1)} \right)^T$$

Over-complete decomposition: M>N

$$\begin{array}{c} \text{Time} \\ \downarrow \end{array} \begin{bmatrix} x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(N) \end{bmatrix} = \alpha_1 \begin{bmatrix} \varphi_1(1) \\ \varphi_1(2) \\ \varphi_1(3) \\ \vdots \\ \varphi_1(N) \end{bmatrix} + \cdots + \alpha_M \begin{bmatrix} \varphi_M(1) \\ \varphi_M(2) \\ \varphi_M(3) \\ \vdots \\ \varphi_M(N) \end{bmatrix}$$

$$\mathbf{x} = \alpha_1 \underline{\varphi}_1 + \cdots + \alpha_M \underline{\varphi}_M$$

- M > N
- Over-complete dictionary
- Under-determined linear system: $\Phi\alpha = \mathbf{x}$
- Non-unique α

Overcomplete Sparse Decomposition: Motivation

$$\mathbf{x} = \alpha_1 \underline{\varphi}_1 + \cdots + \alpha_m \underline{\varphi}_m = \left[\underline{\varphi}_1, \ldots, \underline{\varphi}_m\right] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} = \boldsymbol{\Phi}\,\boldsymbol{\alpha}$$

<u>Example:</u>

- A sinusoidal signal, $\sin(\omega_0 t)$, → Fourier Dictionary
- A signal with just one non-zero value, $\delta(t-t_0)$, → Dirac Dictionary
- How about the signal: $\sin(\omega_0 t)+\delta(t-t_0)$?

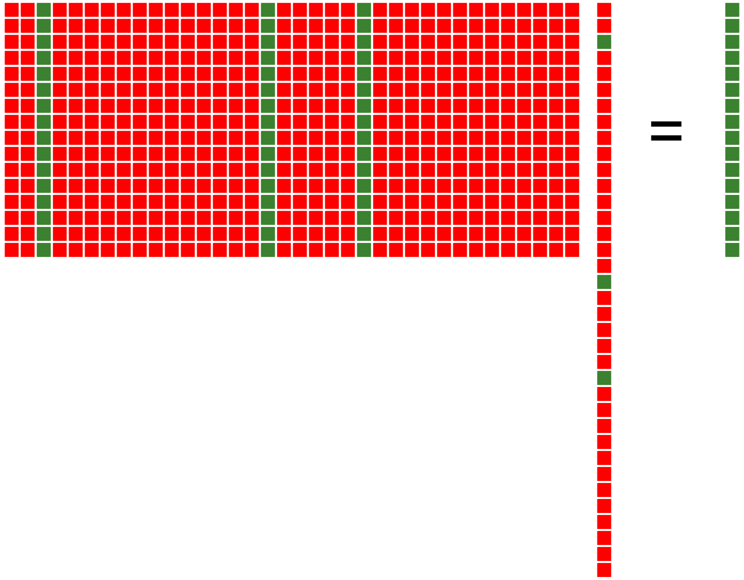
- A larger dictionary, containing both Dirac and Fourier atoms?
  → Non-unique α ☹

- Sparse solution of **Φα=x**

Overcomplete Sparse Decomposition

$$\mathbf{\Phi}\,\boldsymbol{\alpha} = \mathbf{x}$$

$$\alpha_1\varphi_1 + \cdots + \alpha_M\,\varphi_M = \mathbf{x}$$



Mathematical Abstraction

- Under-determined System of Linear Equations (USLE)

$$\mathbf{As=x}$$

- M unknowns
- N equations
- M>N
- Sparse solutions?

Example (2 equations, 4 unknowns)

$$\begin{bmatrix}1 & 2 & 1 & 1\\ 1 & -1 & 2 & -2\end{bmatrix}\begin{bmatrix}s_1\\ s_2\\ s_3\\ s_4\end{bmatrix}=\begin{bmatrix}4\\ -2\end{bmatrix}$$

- Some of solutions:

$$\begin{bmatrix}0\\ 0\\ 1.5\\ 2.5\end{bmatrix},\quad \begin{bmatrix}5\\ 1\\ -3\\ 0\end{bmatrix},\quad \begin{bmatrix}2\\ 1\\ -0.75\\ 0.75\end{bmatrix},\quad \underset{\substack{\downarrow\\ \text{Sparsest}}}{\begin{bmatrix}0\\ 2\\ 0\\ 0\end{bmatrix}},\quad \begin{bmatrix}2\\ 0\\ 0\\ -2\end{bmatrix},\quad \begin{bmatrix}6\\ 0\\ -3\\ 1\end{bmatrix}$$

Two main issues

- Uniqueness?
- How to find the sparse solution?

Uniqueness of the sparse solution

- **x=As**, n equations, m unknowns, m>n
- Theorem (Donoho 2004): if there is a solution **s** with less than n/2 non-zero components, then it is unique under some mild conditions.
- Sparsity Revolution!

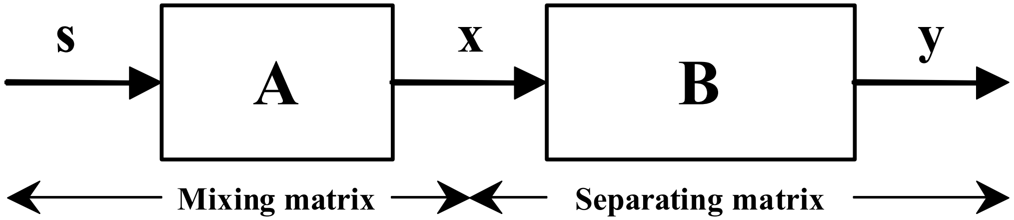
Examples of Applications

Application 1:

Blind Source Separation (BSS) and Sparse Component Analysis (SCA)

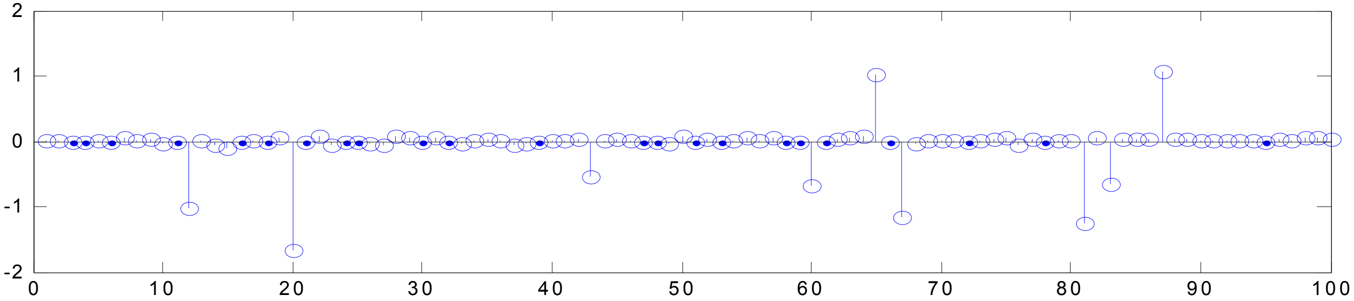
Blind Source Separation (BSS)

- Source signals $s_1, s_2, \ldots, s_M$
- Source vector: $\mathbf{s}=(s_1, s_2, \ldots, s_M)^T$
- Observation vector: $\mathbf{x}=(x_1, x_2, \ldots, x_N)^T$
- Mixing system → $\mathbf{x} = \mathbf{A}\mathbf{s}$



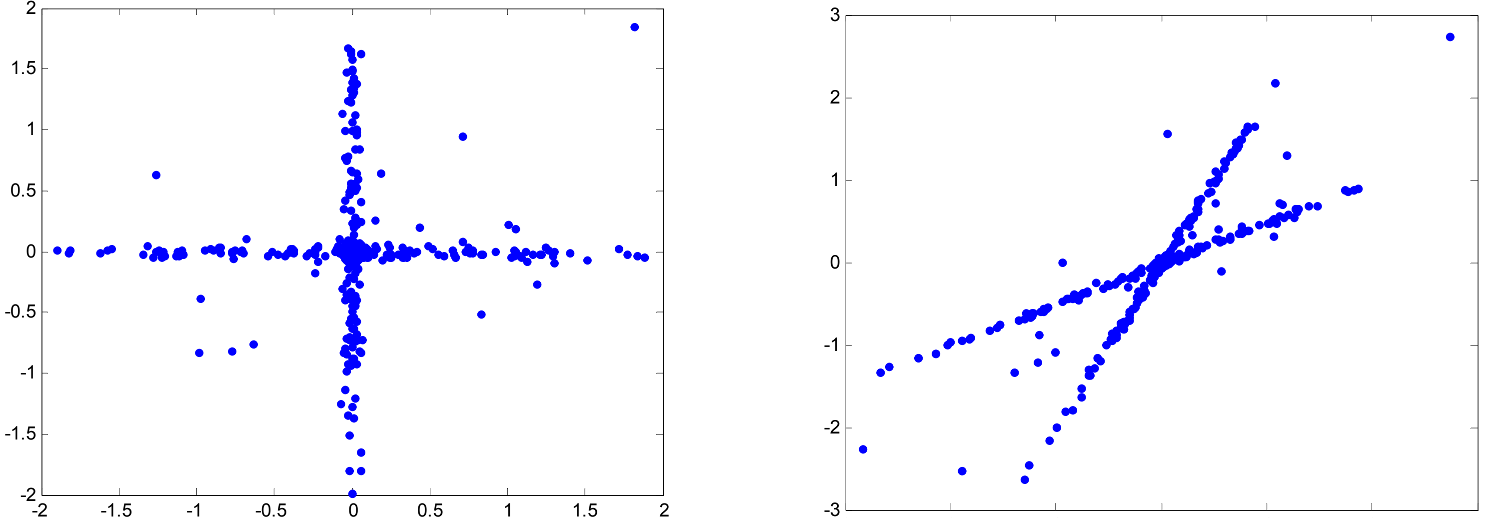
- Goal → Finding a separating matrix $\mathbf{y} = \mathbf{B}\mathbf{x}$

Sparse Sources



Note: The sources may be not sparse in time, but sparse in another domain (frequency, time-frequency, time-scale)

2 sources, 2 sensors:

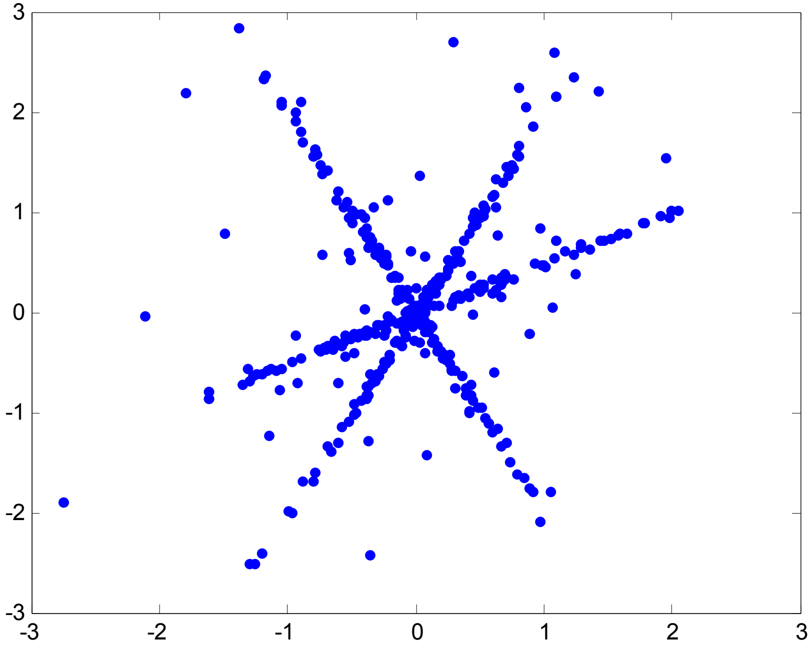


s-plane x-plane

Sparse sources (*cont.*)

- 3 sparse sources, 2 sensors

Sparsity ⇒ Source Separation, with more sensors than sources?



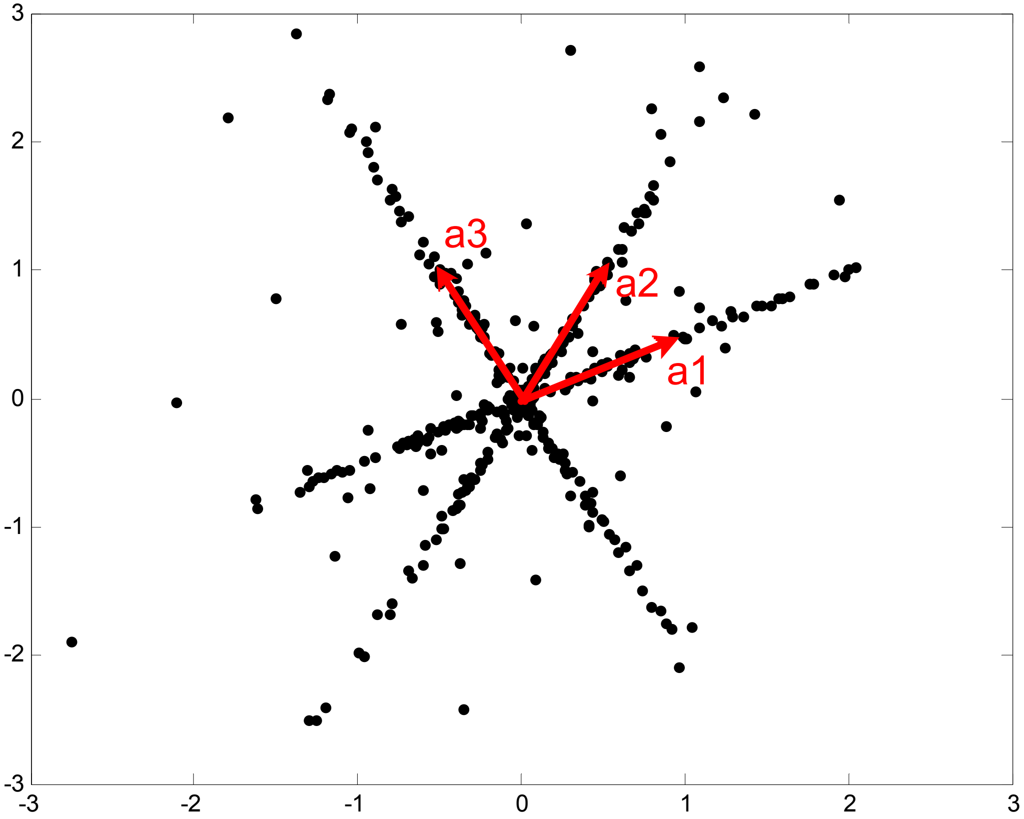
Estimating the mixing matrix

$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] \Rightarrow$

$\mathbf{x} = s_1\,\mathbf{a}_1 + s_2\,\mathbf{a}_2 + s_3\,\mathbf{a}_3$

$\Rightarrow$ Mixing matrix is easily identified for sparse sources

- Scale & Permutation indeterminacy
- $||\mathbf{a}_i||=1$



Restoration of the sources

- **A** known, how to find the sources?

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad or \quad \begin{cases} a_{11}s_1 + a_{12}s_2 + a_{13}s_3 = x_1 \\ a_{21}s_1 + a_{22}s_2 + a_{23}s_3 = x_2 \end{cases}$$

Underdertermined SCA

Application 2:

Error Correcting Codes

Coding problem

- **v** → code vector (length n)
- **H** → Parity check matrix, (n-k)×n
- **Hv**=0
- **e** → error
- **x=v+e** → received message
- **r=Hx=H**(**v**+**e**)=**He** → Syndrom

- Correcting errors: **He=r** → USLE

Application 3:

Compressed Sensing

Compressed Sensing

- Why to record a large samples of a signal, and then compress it? → requires Expensive A/D

- One-pixel camera (Rice university)

Other Applications

- Image Denoising
- OCR
- Sampling Theory
- …

How to find the sparse solution

How to find the sparsest solution

- **A.s** = **x**, n equations, m unknowns, m>n
- Goal: Finding the sparsest solution
- Note: at least m-n sources are zero.

- Direct method:
  - Set m-n (arbitrary) sources equal to zero
  - Solve the remaining system of n equations and n unknowns
  - Do above for all possible choices, and take sparsest answer.

- Another name: Minimum L^0 norm method
  - L^0 norm of s = number of non-zero components = $\Sigma|s_i|^0$

Example

$$\begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & -1 & 2 & -2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} 4 \\ -2 \end{bmatrix}$$

$$\binom{4}{2} = 6 \text{ different answers to be tested}$$

- s1=s2=0 $\Rightarrow$ **s**=$(0, 0, 1.5, 2.5)^T$ $\Rightarrow$ $L^0=2$
- s1=s3=0 $\Rightarrow$ **s**=$(0, 2, 0, 0)^T$ $\Rightarrow$ $L^0=1$
- s1=s4=0 $\Rightarrow$ **s**=$(0, 2, 0, 0)^T$ $\Rightarrow$ $L^0=1$
- s2=s3=0 $\Rightarrow$ **s**=$(2, 0, 0, 2)^T$ $\Rightarrow$ $L^0=2$
- s2=s4=0 $\Rightarrow$ **s**=$(10, 0, -6, 0)^T$ $\Rightarrow$ $L^0=2$
- s3=s4=0 $\Rightarrow$ **s**=$(0, 2, 0, 0)^T$ $\Rightarrow$ $L^0=2$
- $\Rightarrow$ Minimum L^0 norm solution $\rightarrow$ **s**=$(0, 2, 0, 0)^T$

Drawbacks of minimal norm L^0

$$(P_0)\ \text{Minimize } \|\mathbf{s}\|_0 = \sum_i |s_i|^0 \quad \text{s.t. } \mathbf{x} = \mathbf{A}\mathbf{s}$$

- Highly (unacceptably) sensitive to noise
- Need for a combinatorial search:

$$\binom{m}{n} \text{ diffetent cases should be tested separately}$$

- Example. m=50, n=30,

$$\binom{50}{30} \approx 5 \times 10^{13} \text{ cases should be tested.}$$

On our computer: Time for solving a 30 by 30 system of equation=$2\text{x}10^{-4}$

Total time ≈ $(5\text{x}10^{13})(2\text{x}10^{-4})$ ≈ 300 years! → Non-tractable

A few faster methods

- Method of Frames (MoF) [Daubechies, 1989]
- Matching Pursuit [Mallat & Zhang, 1993]
- Basis Pursuit (minimal L1 norm → Linear Programming) [Chen, Donoho, Saunders, 1995]
- Our methods

Method of Frames (Daubechies, 1989)

- Take the minimum norm 2 (energy) solution:

$$(P_2)\ \text{Minimize } \|\mathbf{s}\|_2 = \sum_i |s_i|^2 \quad \text{s.t. } \mathbf{x} = \mathbf{A}\mathbf{s}$$

- Solution: pseudo inverse:

$$\hat{\mathbf{s}}_{MoF} = \mathbf{A}^T \left(\mathbf{A}\mathbf{A}^T\right)^{-1} \mathbf{x}$$

- Different view points resulting in the same answer:
  - Linear LS inverse $\hat{\mathbf{s}} = \mathbf{B}\mathbf{x},\quad \mathbf{B}\mathbf{A} \overset{LS}{\approx} \mathbf{I}$
  - Linear MMSE Estimator
  - MAP estimator under a Gaussian prior $\mathbf{s} \sim N\left(0, \sigma_s^2 \mathbf{I}\right)$

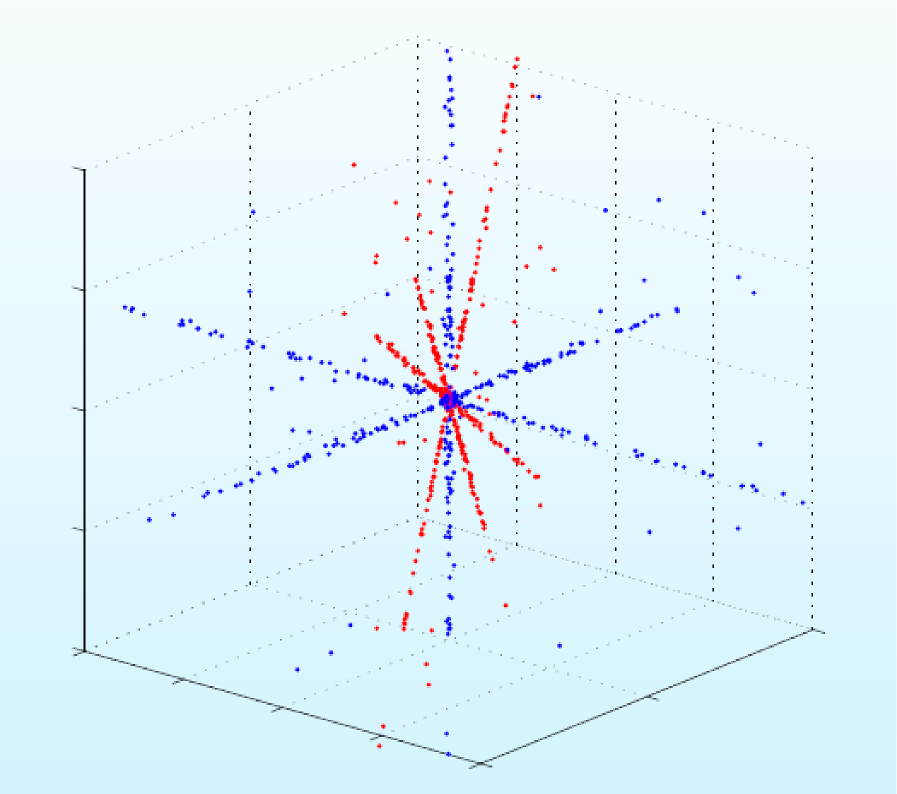
Drawback of MoF

- It is a ‘linear’ method: **s=Bx**

  ⇒ s will be an n-dim subspace of m-dim space

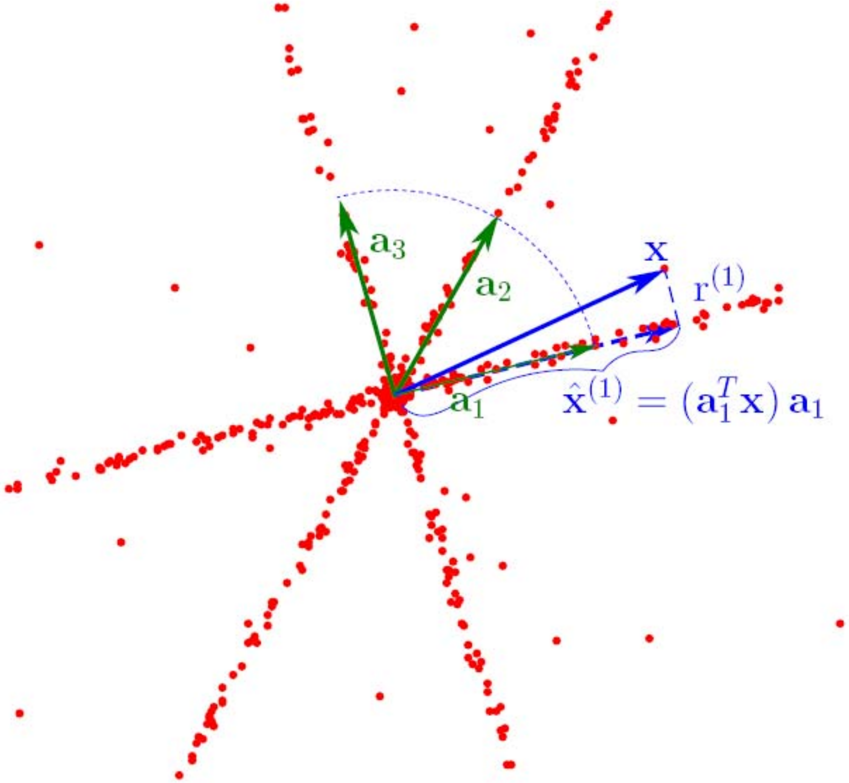
- Example:

  3 sources, 2 sensors:



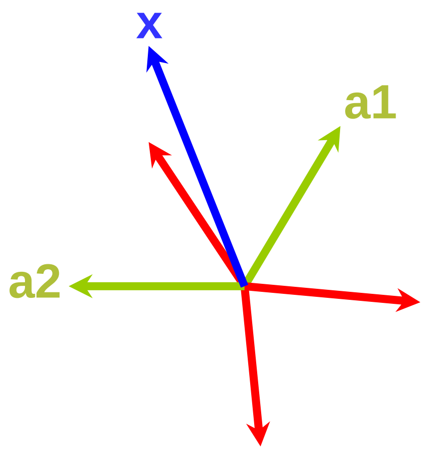
- ⇒ Never can produce original sources

Matching Pursuit (MP) [Mallat & Zhang, 1993]



Properties of MP

- Advantage:
  - Very Fast
- Drawback
  - A very ‘greedy’ algorithm → Error in a stage, can never be corrected → Not necessarily a sparse solution



Minimum L^1 norm or Basis Pursuit [Chen, Donoho, Saunders, 1995]

- Minimum norm L1 solution:

$$(P_1)\ \text{Minimize}\ \|\mathbf{s}\|_1 = \sum_i |s_i| \quad \text{s.t.}\ \ \mathbf{x} = \mathbf{A}\mathbf{s}$$

- MAP estimator under a Laplacian prior

- Theoretical support (Donoho, 2004):

For 'most' 'large' underdetermined systems of linear equations, the minimal L^1 norm solution is also the sparsest solution

Minimal L^1 norm (*cont.*)

$$(P_1)\ \ \text{Minimize}\ \|\mathbf{s}\|_1 = \sum_i |s_i| \quad \text{s.t.}\ \ \mathbf{x} = \mathbf{A}\mathbf{s}$$

- Minimal L^1 norm solution may be found by Linear Programming (LP)

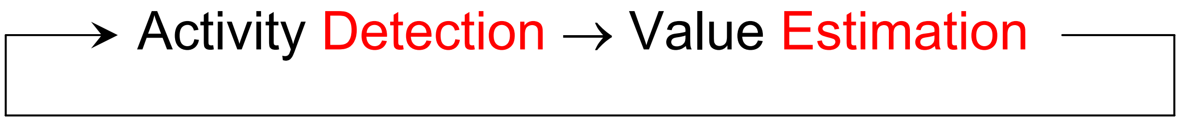
- Fast algorithms for LP:
  - Simplex
  - Interior Point method

Minimal L^1 norm (*cont.*)

- Advantages:
  - Very good practical results
  - Theoretical support
- Drawback:
  - Tractable, but still very time-consuming

Iterative Detection-Estmation (IDE)- Our method

- Main Idea:
  - Step 1 (Detection): Detect which sources are ‘active’, and which are ‘non-active’
  - Step 2 (Estimation): Knowing active sources, estimate their values
- Problem: Detection the activity status of a source, requires the values of all other sources!
- Our proposition: Iterative Detection-Estimation



Activity Detection → Value Estimation

Thank you very much for your attention