“Underdetermined Sparse Component Analysis (SCA)”

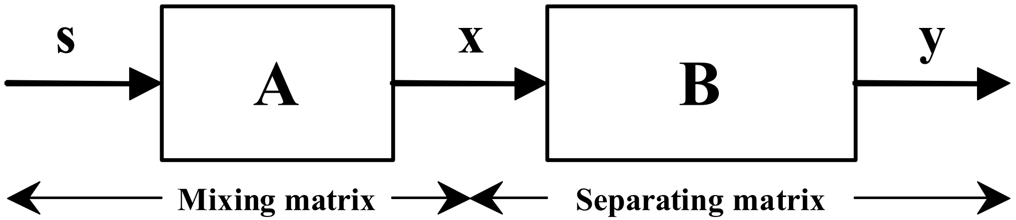
Massoud BABAIE-ZADEH

Outline

- Introduction to Blind Source Separation
- Geometrical Interpretation
- Sparse Component Analysis (SCA), underdetermined case
  - Identifying mixing matrix
  - Source restoration
- Finding sparse solutions of an Underdetermined System of Linear Equations (USLE):
  - Minimum L0 norm
  - Method of Frames
  - Matching Pursuit
  - Minimum L1 norm or Basis Pursuit (→Linear Programming)
  - Iterative Detection-Estimation (IDE) - our method
- Simulation results
- Conclusions and Perspectives

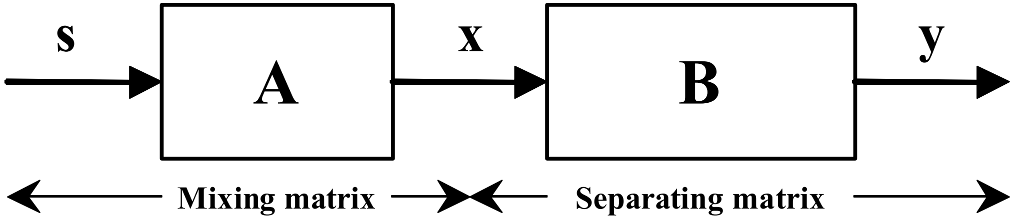
Blind Source Separation (BSS)

- 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 $\rightarrow$ $\mathbf{x} = \mathbf{A}\mathbf{s}$



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

Blind Source Separation (*cont.*)

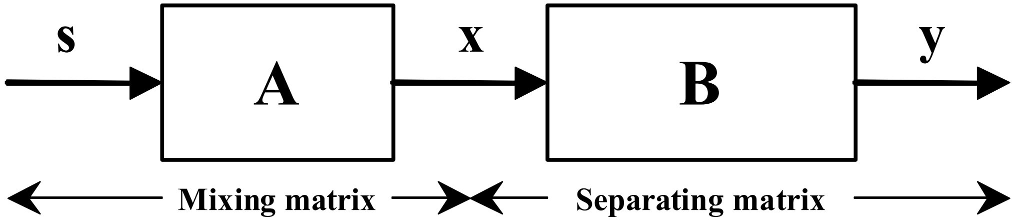


- Assumption:
  - N=M (#sensors = #sources), or N >=M (#sensors >= #sources)
  - A is full-rank (invertible)

- *prior* information: Statistical "Independence" of sources

- Main idea: Find "**B**" to obtain "independent" outputs (⇒ Independent Component Analysis=ICA)

Blind Source Separation (*cont.*)



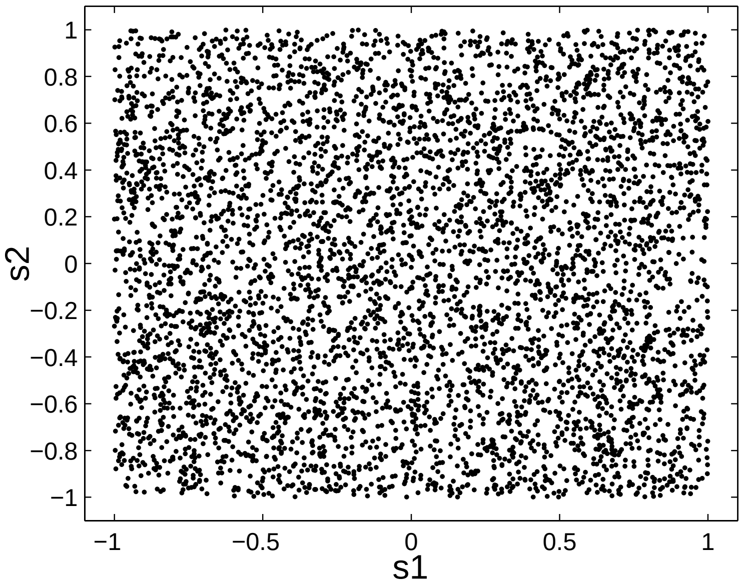
- Separability Theorem [Comon 1994,Darmois 1953]: If at most 1 source is Gaussian: statistical independence of outputs ⇒ source separation (⇒ ICA: a method for BSS)

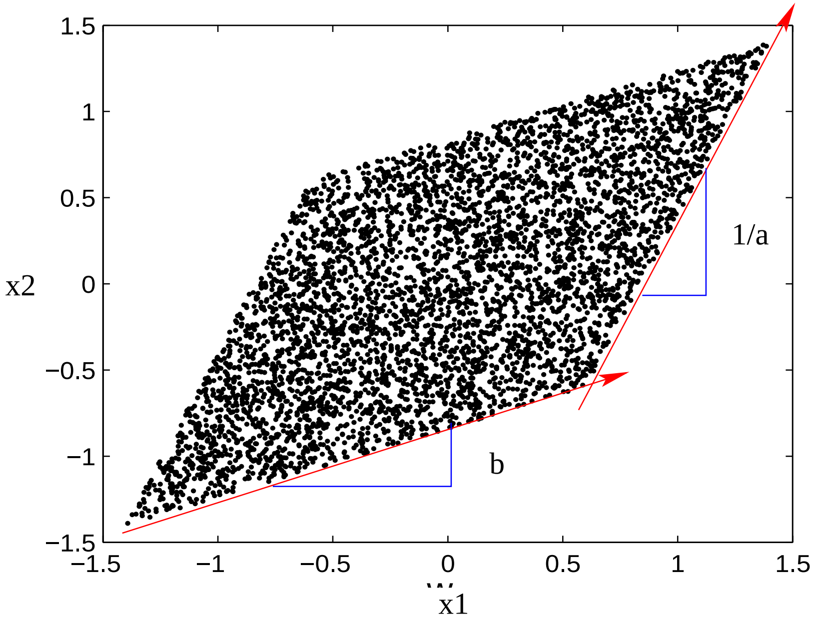
- Indeterminacies: permutation, scale

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M] \text{ , } \mathbf{x}=\mathbf{As} \quad \Rightarrow$$

$$\mathbf{x} = s_1\, \mathbf{a}_1 + s_2\, \mathbf{a}_2 + \ldots + s_M\, \mathbf{a}_M$$

Geometrical Interpretation



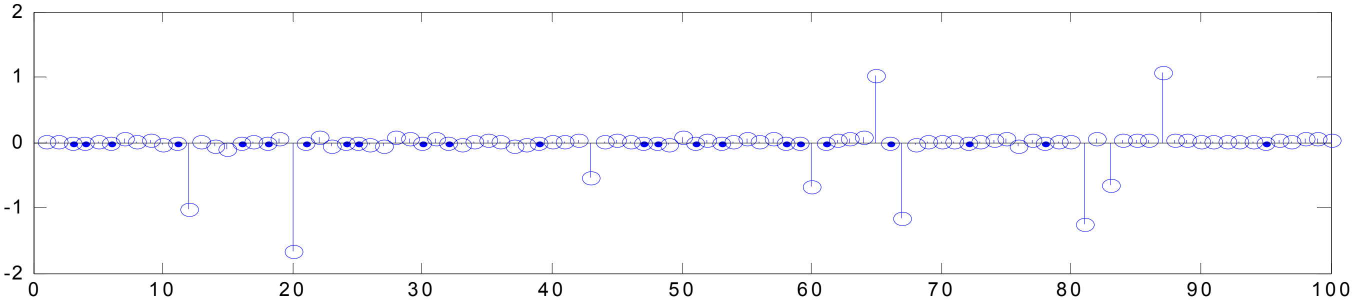


x=As

$$\mathbf{A} = \begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix}$$

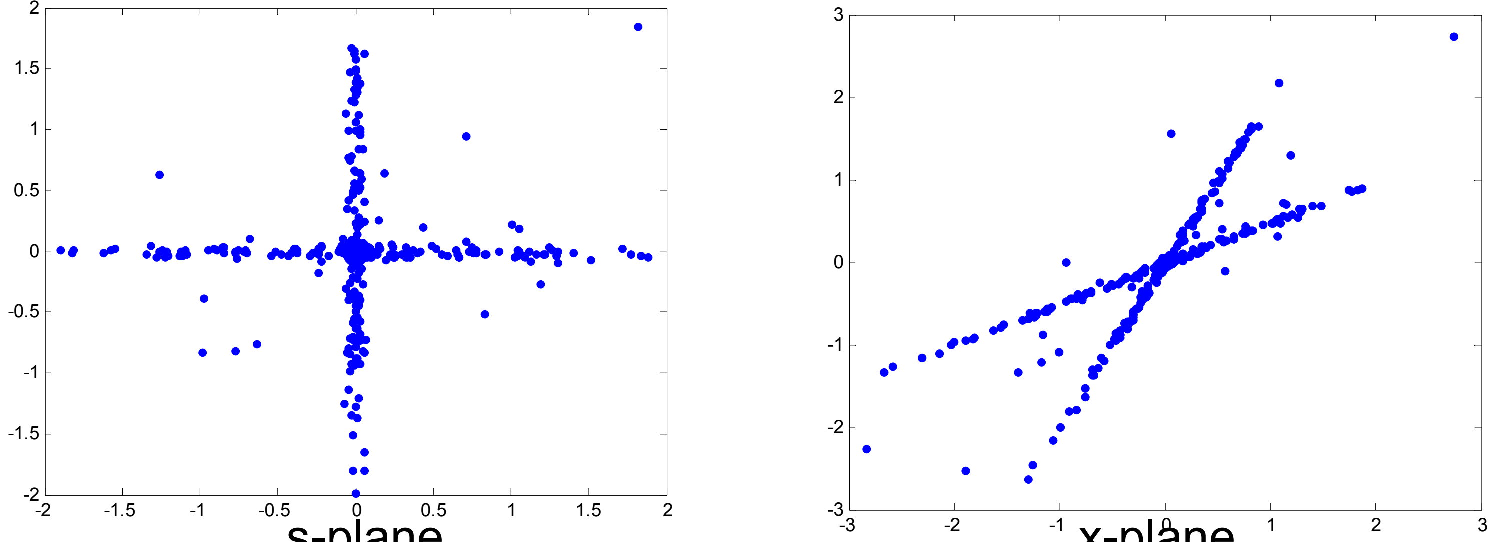
Statistical Independence of s1 and s2 $\Rightarrow$ rectangular scatter plot of (s1,s2)

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:



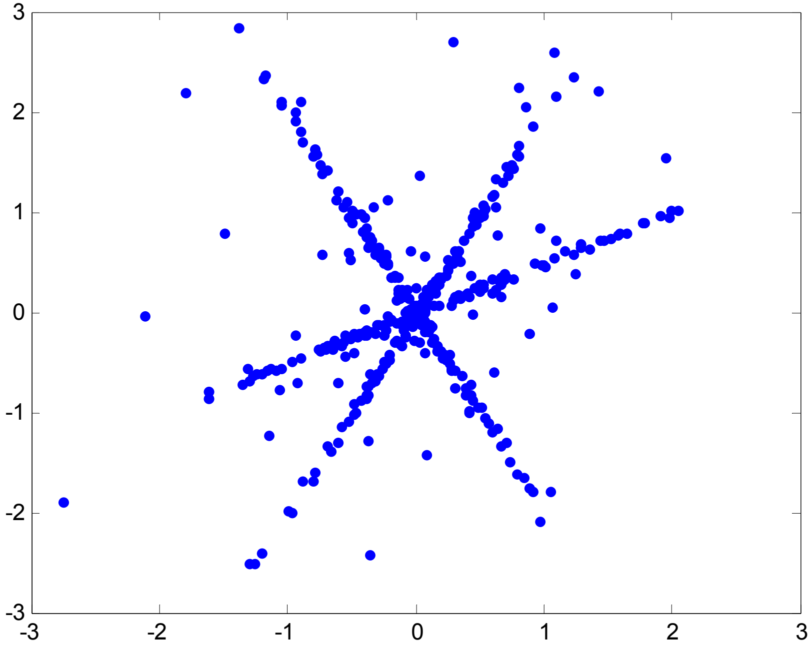
s-plane

x-plane

Sparse sources (*cont.*)

- 3 sparse sources, 2 sensors

Sparsity ⇒ Source Separation, with more sensors than sources?



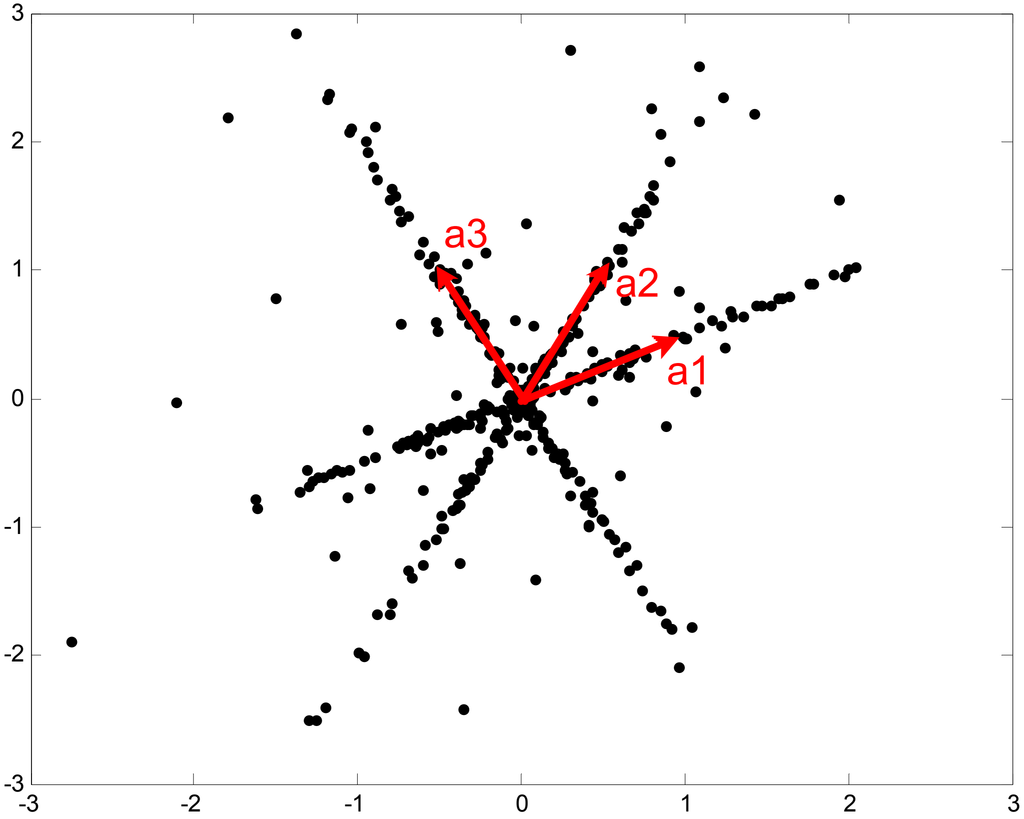
Estimating the mixing matrix

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

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

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

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



Restoration of the sources

- How to find the sources, after having found the mixing matrix (**A**)?

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

2 equations, 3 unknowns $\Rightarrow$ infinitely many solutions!

Underdertermined SCA, underdetermined system of equations

Identification vs Separation

- Case #Sources <= #Sensors: (determined or overdtermined)

  Identifying A ⇒ source Separation

- Underdetermined case: #Sources > #Sensors

  Two different problems:

  - Identifying the mixing matrix (relatively easy)
  - Restoring the sources (difficult)

Is it possible?

- A is known, at eash instant (n_0), we should solve un underdetermined linear system of equations:

$$\mathbf{A}\,\mathbf{s}(n_0) = \mathbf{x}(n_0) \quad or \quad \begin{cases} a_{11}s_1(n_0) + a_{12}s_2(n_0) + a_{13}s_3(n_0) = x_1(n_0) \\ a_{21}s_1(n_0) + a_{22}s_2(n_0) + a_{23}s_3(n_0) = x_2(n_0) \end{cases}$$

- Infinite number of solutions **s**(n_0) → Is it possible to recover the sources?

‘Sparse’ solution

- $s_i(n)$ sparse in time ⇒ The vector $\mathbf{s}(n_0)$ is most likely a ‘sparse vector’

- $\mathbf{A}.\mathbf{s}(n_0) = \mathbf{x}(n_0)$ has infinitely many solutions, but not all of them are sparse!

- Idea: For restoring the sources, take the sparsest solution (most likely solution)

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:

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

Sparsest

The idea of solving underdetermined SCA

$$\mathbf{A}\,\mathbf{s}(n) = \mathbf{x}(n)\ ,\ n=0,1,\ldots,T$$

- Step 1 (identification): Estimate A (relatively easy)
- Step 2 (source restoration): At each instant n_0, find the sparsest solution of

$$\mathbf{A}\,\mathbf{s}(n_0) = \mathbf{x}(n_0),\ n_0=0,\ldots,T$$

Main question: HOW to find the sparsest solution of an Underdetermined System of Linear Equations (USLE)?

Another application of USLE: Atomic decomposition over an overcompelete dictionary

- Decomposing a signal x, as a linear combination of a set of fixed signals (atoms)

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

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

- Terminology:
  - Atoms: $\underline{\varphi}_i$, i=1,…,M
  - Dictionary: $\{\underline{\varphi}_1 , \underline{\varphi}_2 ,\ldots, \underline{\varphi}_M\}$

Atomic decomposition (*cont.*)

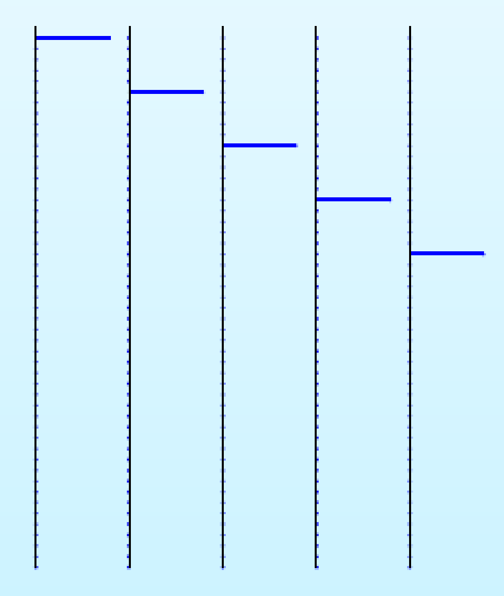
$$\begin{array}{c} \text{Time} \\ \downarrow \end{array} \begin{bmatrix} x(1)\\ x(2)\\ x(3)\\ \vdots\\ x(N)\end{bmatrix} = \alpha_1 \begin{bmatrix} \varphi_1(1)\\ \varphi_1(2)\\ \varphi_1(3)\\ \vdots\\ \varphi_1(N)\end{bmatrix} + \cdots + \alpha_M \begin{bmatrix} \varphi_M(1)\\ \varphi_M(2)\\ \varphi_M(3)\\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

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

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

Dirac Dictionary:

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

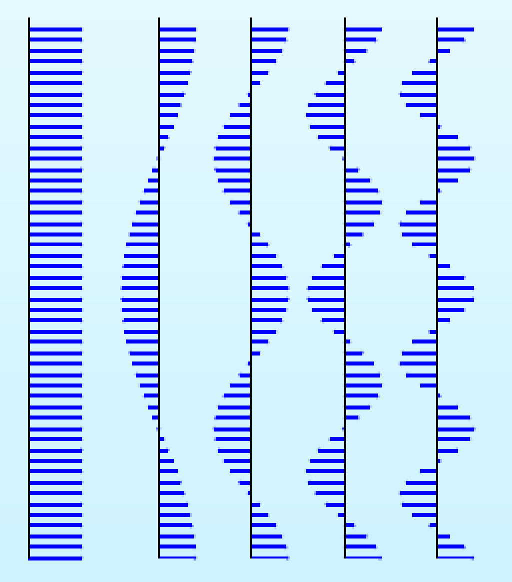


Atomic decomposition (*cont.*)

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

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

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



Fourier Dictionary:

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

Atomic decomposition (*cont.*)

$$\mathbf{x} = \alpha_1 \underline{\varphi}_1 + \cdots + \alpha_m \underline{\varphi}_m$$

- Matrix Form:

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

- If just a few number of coefficient are non-zero $\Rightarrow$ The underlying structure is very well revealed

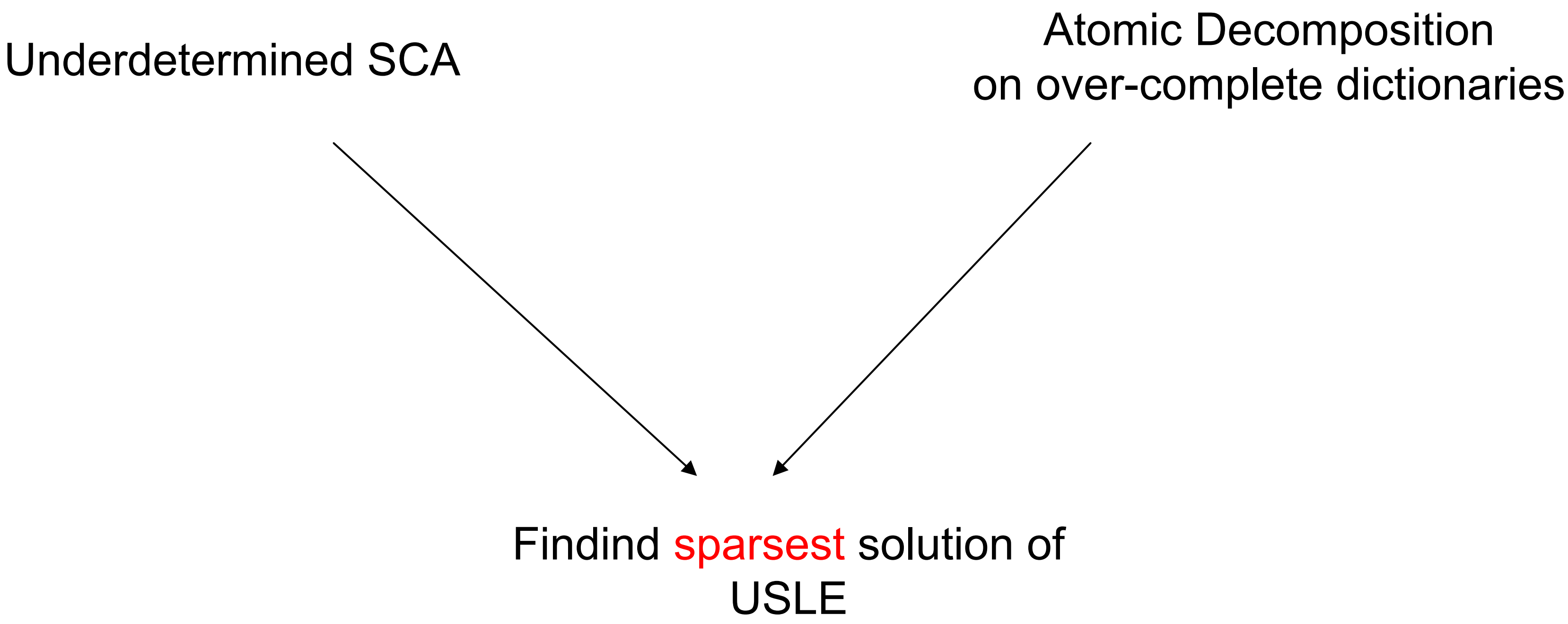
- Example.
  - signal has just a few non-zero samples in time → its decomposition over the Dirac dictionary reveals it
  - Signals composed of a few pure frequencies → its decomposition over the Fourier dictionary reveals it
  - How about a signals which is the sum of a pure frequency and a dirac?

Atomic decomposition (*cont.*)

$$\mathbf{x} = \alpha_1 \underline{\varphi_1} + \cdots + \alpha_m \underline{\varphi_m} = \left[\underline{\varphi_1}, \ldots, \underline{\varphi_m} \right] \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} = \mathbf{\Phi} \boldsymbol{\alpha}$$

- Solution: consider a larger dictionary, containing both Dirac and Fourier atoms
- **M>N** → Overcomplete dictionary.
- Problem: Non-uniqueness of α (→ USLE)
- However: we are looking for sparse solution

Sparse solution of USLE



Underdetermined SCA

Atomic Decomposition
on over-complete dictionaries

Findind sparsest solution of
USLE

Uniqueness of sparse solution

- **x=As**, n equations, m unknowns, m>n

- Question: Is the sparse solution unique?

- Theorem (Donoho 2004): if there is a solution **s** with less than n/2 non-zero components, then it is unique with probability 1 (that is, for almost all **A**'s).

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:

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

- Example. m=50, n=30,

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

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 method (IDE)

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 $\hat{\mathbf{s}} = \mathbf{B}\mathbf{x},\ \ \mathbf{B}\mathbf{A} \overset{LS}{\approx} \mathbf{I}$
  - Linear MMSE Estimator
  - MAP estimator under a Gaussian prior $\mathbf{s} \sim N\left(0, \sigma_s^2 \mathbf{I}\right)$

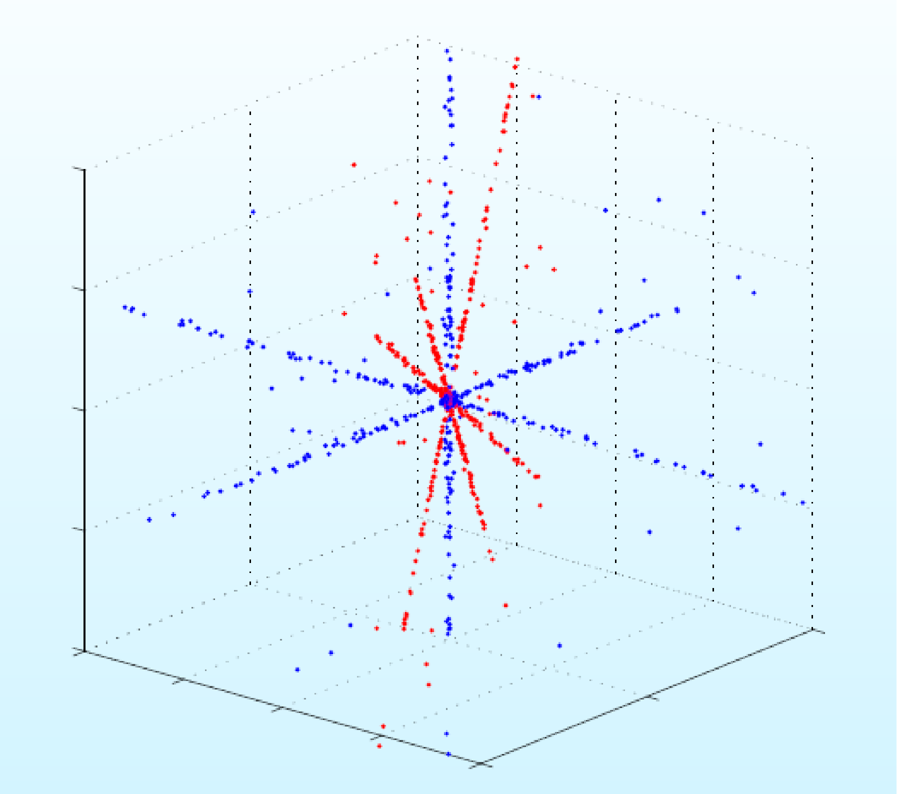
Drawback of MoF

- It is a ‘linear’ method: **s=Bx**

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

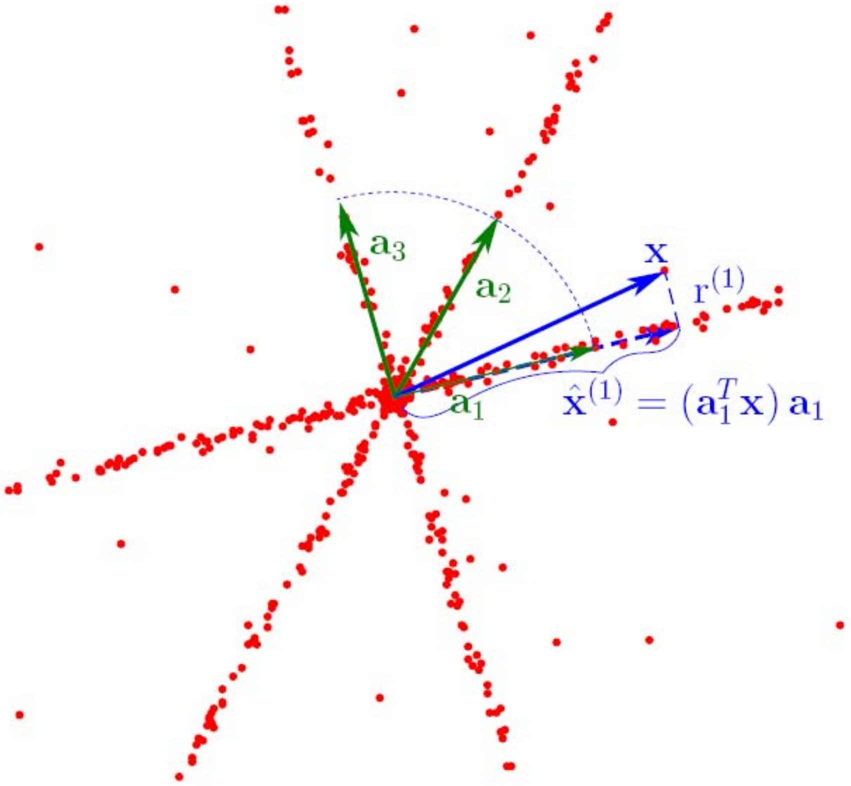
- Example:

  3 sources, 2 sensors:



- ⇒ Never can produce original sources

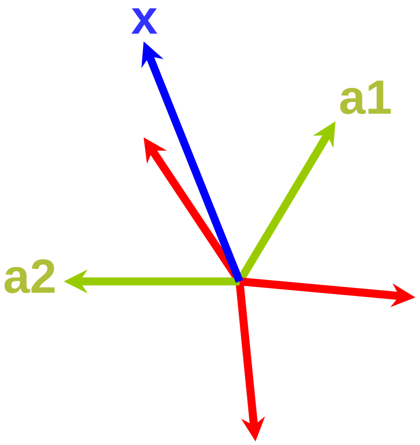
Matching Pursuit (MP) [Mallat & Zhang, 1993]



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

- Recent 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

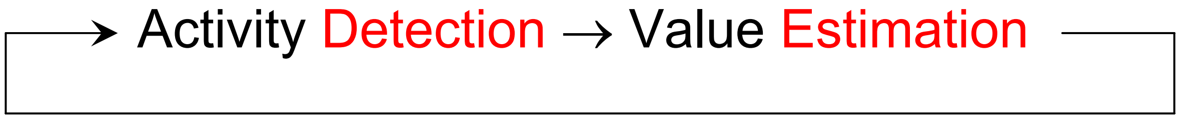
Minimal L^1 norm (*cont.*)

- 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

IDE *(cont.)*

- Detection Step (resulted from binary hypothesis testing, with a Mixture of Gaussian source model):

$$g_i(\mathbf{x},\hat{\mathbf{s}}) = \left| \mathbf{a}_i^T \left(\mathbf{x} - \sum_{j \neq i}^{m} \hat{s}_j \mathbf{a}_j \right) \right| > \varepsilon$$

$$or \quad \mathbf{g}(\mathbf{x},\hat{\mathbf{s}}) = \left| \mathbf{A}^T(\mathbf{x} - \mathbf{A}\hat{\mathbf{s}}) + \hat{\mathbf{s}} \right|$$

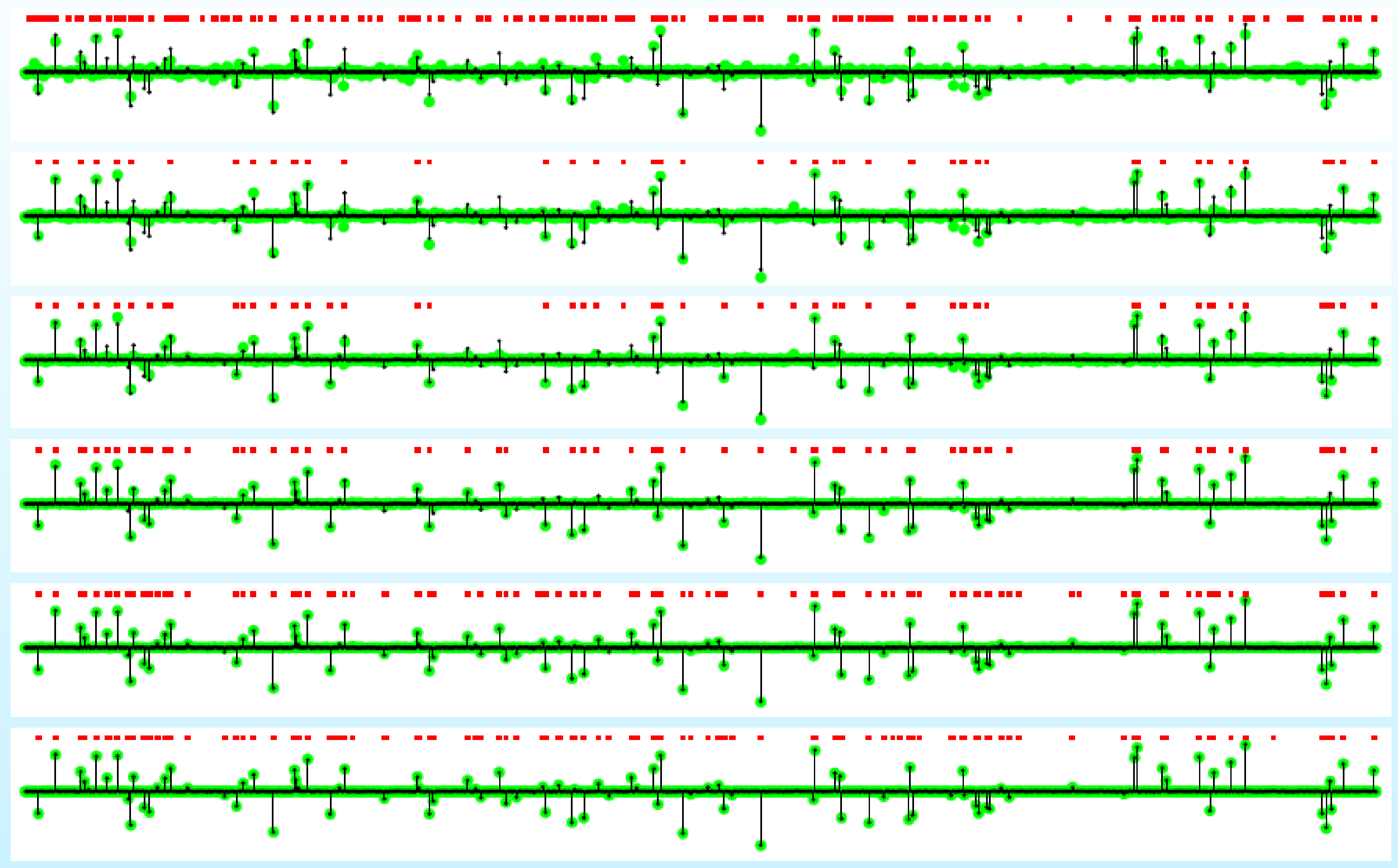
- Estimation Step:

$$\text{(IDE-s)} \quad \text{minimize} \sum_{i \in I_{inactive}} s_i^2 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{A}\mathbf{s}$$

$$\text{(IDE-x)} \quad \text{Let } \mathbf{s}_{inactive} = \mathbf{0}, \text{ and minimize } \left\| \mathbf{x} - \mathbf{A}_{act}\mathbf{s}_{act} \right\|_2$$

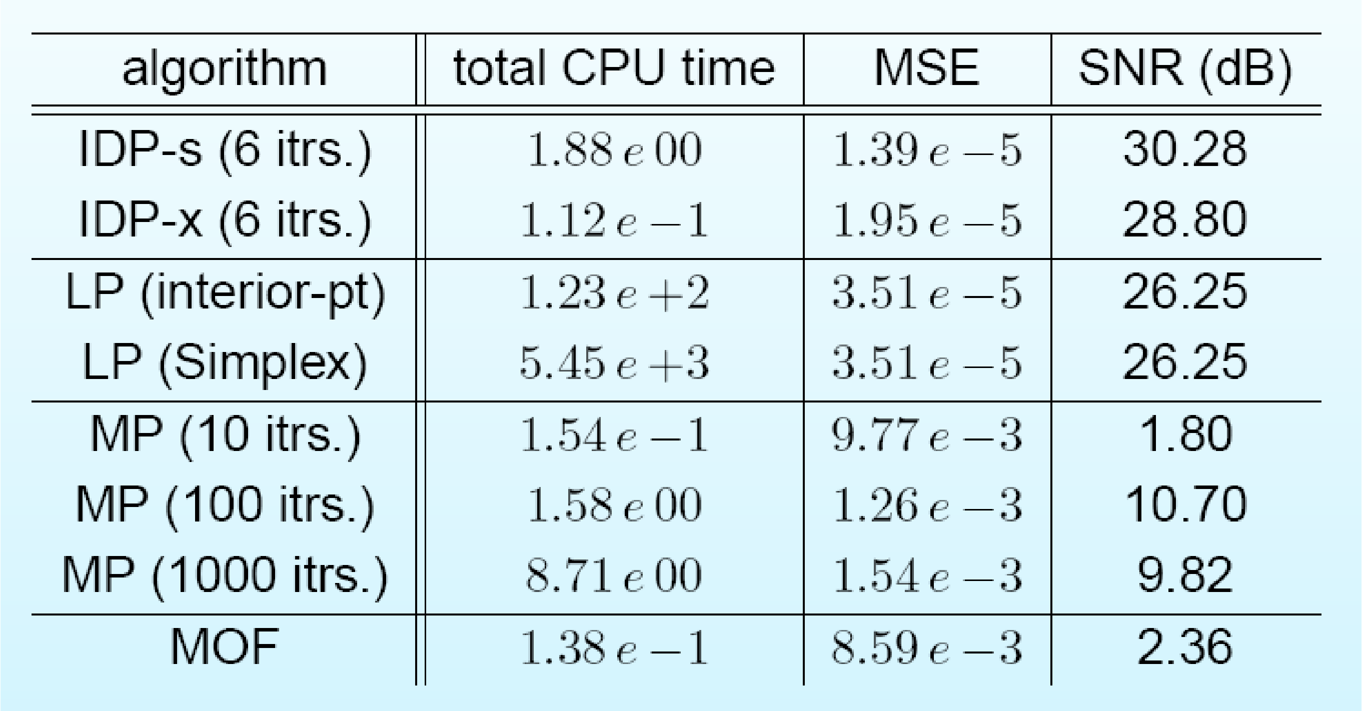
IDE (*cont.*)

m=1024, n=0.4m=409



IDE (*Simulation Results*)

- m=1024, n=0.4m=409

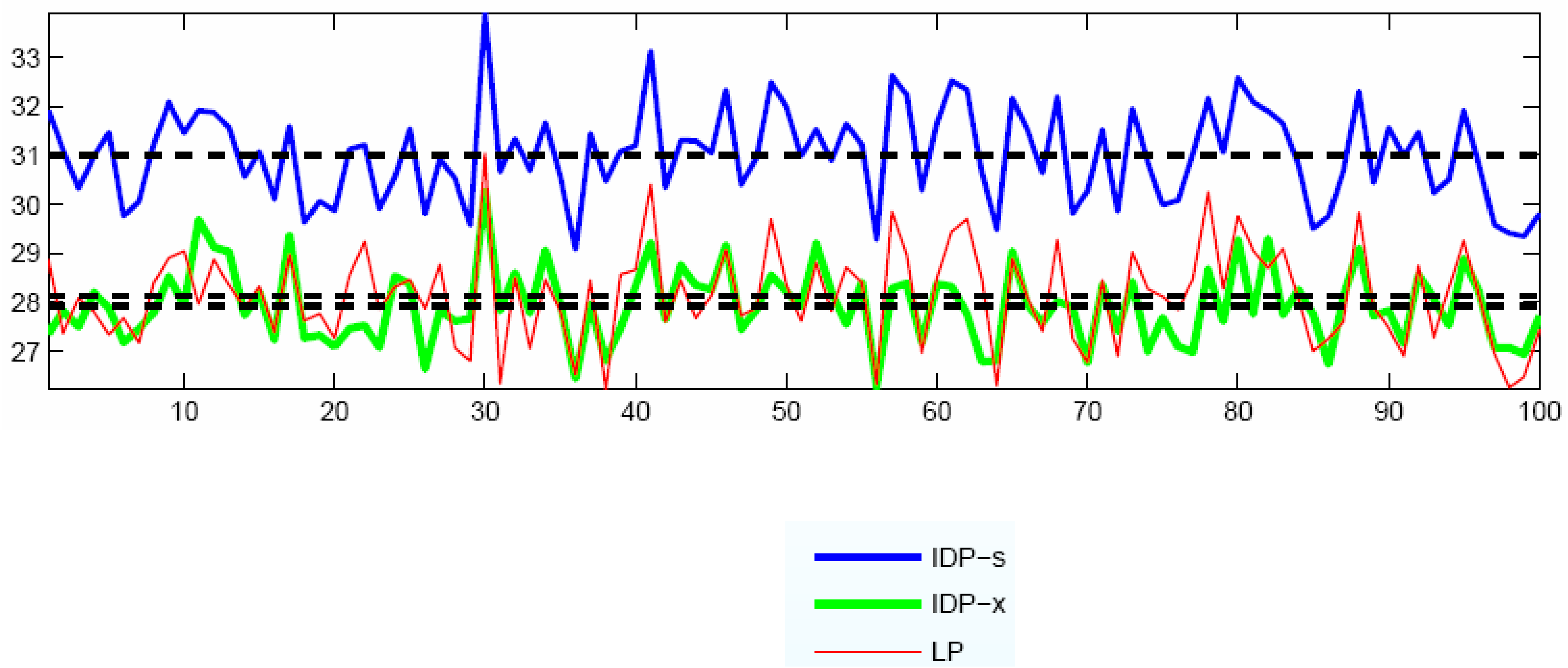


algorithm	total CPU time	MSE	SNR (dB)
IDP-s (6 itrs.)	$1.88\,e\,00$	$1.39\,e-5$	30.28
IDP-x (6 itrs.)	$1.12\,e-1$	$1.95\,e-5$	28.80
LP (interior-pt)	$1.23\,e+2$	$3.51\,e-5$	26.25
LP (Simplex)	$5.45\,e+3$	$3.51\,e-5$	26.25
MP (10 itrs.)	$1.54\,e-1$	$9.77\,e-3$	1.80
MP (100 itrs.)	$1.58\,e\,00$	$1.26\,e-3$	10.70
MP (1000 itrs.)	$8.71\,e\,00$	$1.54\,e-3$	9.82
MOF	$1.38\,e-1$	$8.59\,e-3$	2.36

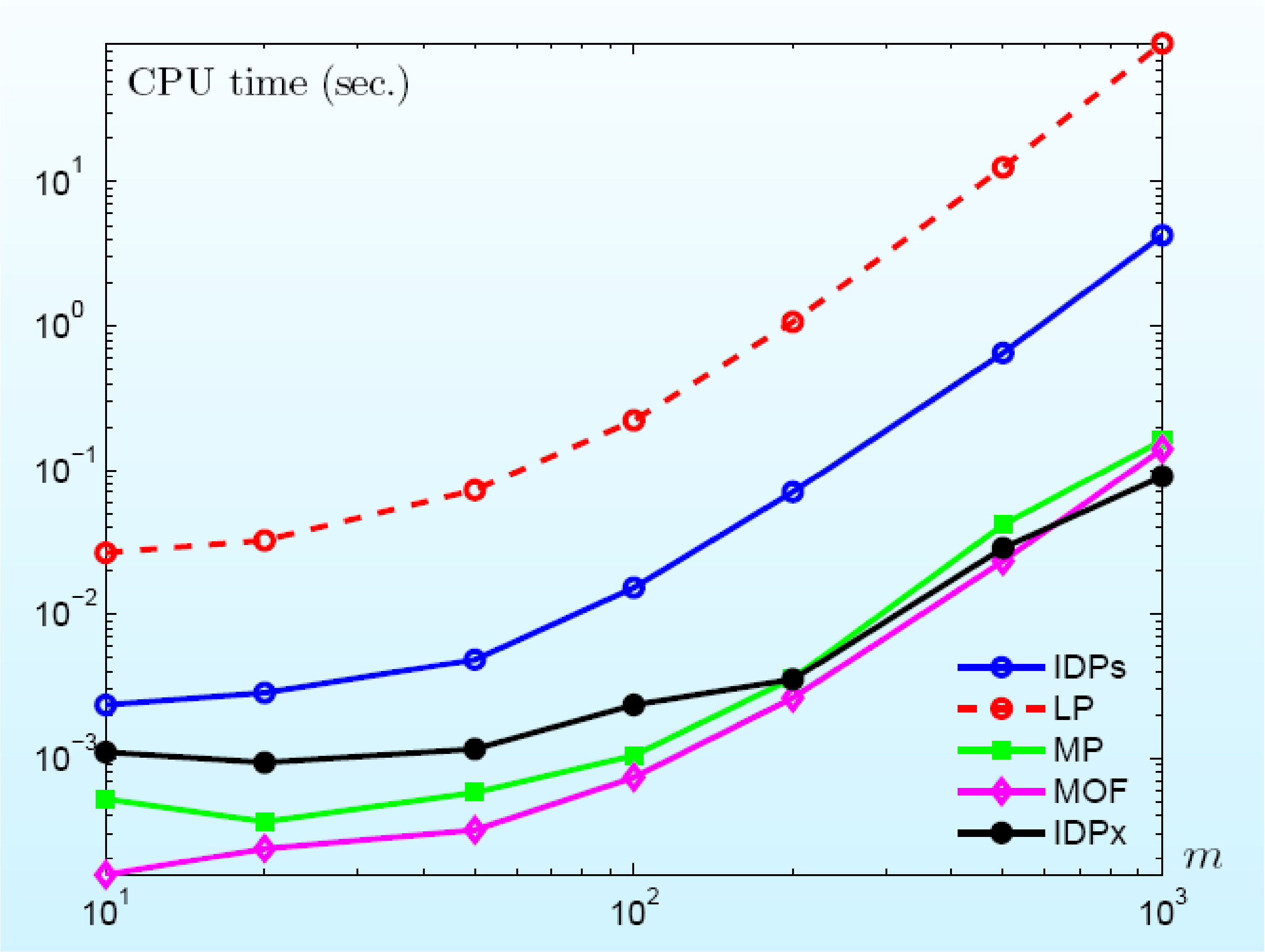
- IDE-x is about two order of magnitudes faster than LP method.

IDE (*Simulation Results*)

m=100, n=0.6m, Averaged SNRs (on 1000 simulations)



Speed/Complexity comparision



Conclusion and Perspectives

- Two problems of Underdetermined SCA:
  - Identifying mixing matrix
  - Restoring sources
- Two applications of finding sparse solution of USLE’s:
  - Source restoration in underdetermined SCA
  - Atomic Decomposition on over-complete dictionaries
- 5 methods:
  - Minimum L0 norm (→Combinatorial search)
  - Method of Frames
  - Minimum L1 norm or Basis Pursuit (→Linear Programming)
  - Matching Pursuit
  - Iterative Detection-Estimation (IDE)
- Perspectives:
  - Better activity detection (removing thresholds?)
  - Applications in other domains

Thank you very much for your attention