IMA 207

Hyperspectral Imaging and Blind Source Separation

Christophe Kervazo
christophe.kervazo@telecom-paris.fr

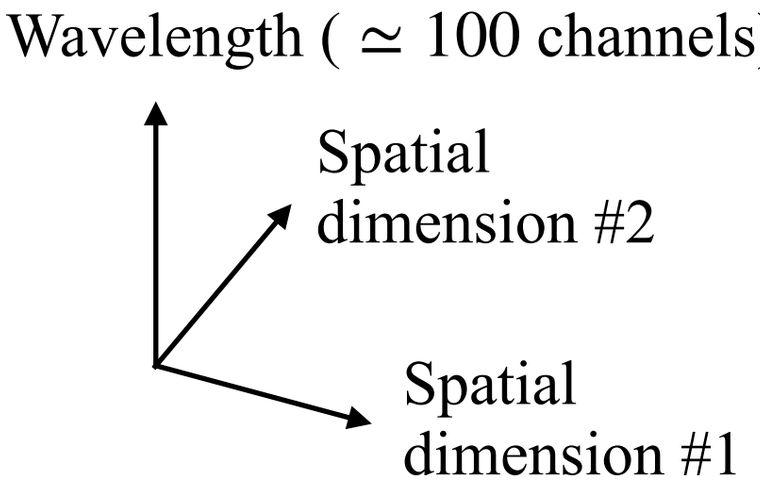
Class modalities

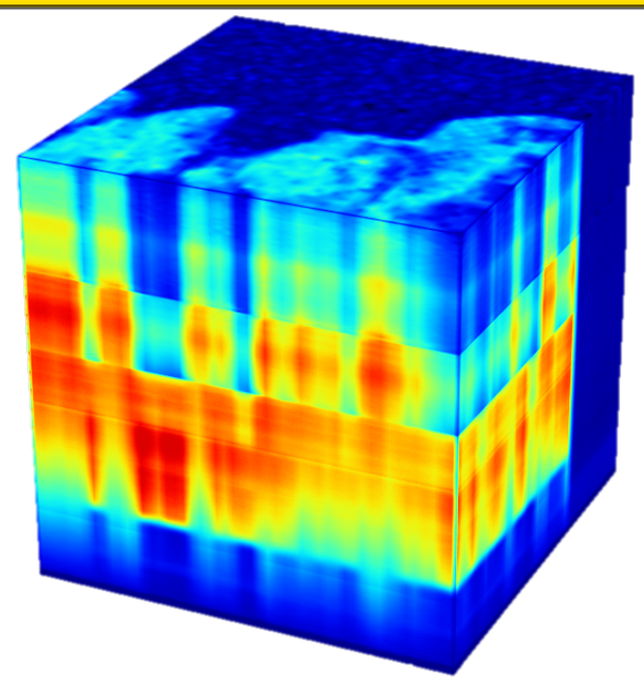
- 1 TH about sparse hyperspectral unmixing
- 1 TH of TP
- 1 TH about nonnegative hyperspectral unmixing
- 1 TH of TP

Evaluation :

- TP (50 %) : each report/notebook must be submitted within one week (on Tuesday evening) after the TP in e-campus
- Exam (50%) : the slides and TP should be mastered

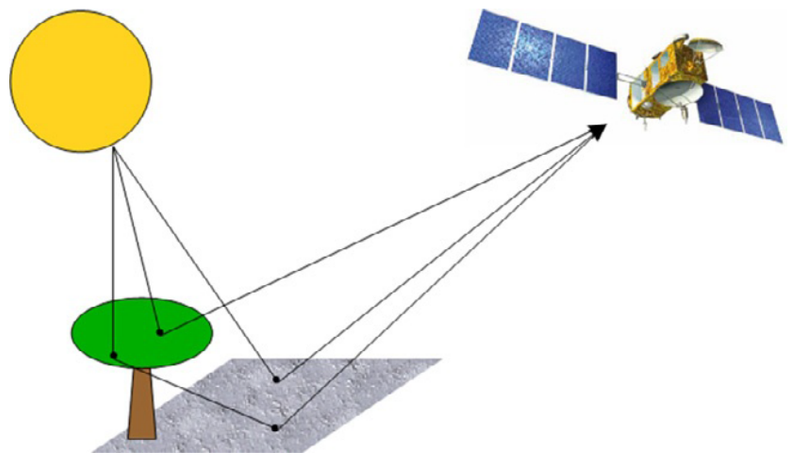
Context : hyperspectral imaging





Hyperspectral images:

- measure electromagnetic energy coming from the scene within many wavelength bands
- high spectral resolution (≃ 100, 1000 channels): generalization of multi-spectral sensors (in RGB image, 3 channels)



Interest :
- Yields a very precise spectral information about the scene

=> enable the identification of scene materials
=> might be more invariant to object shapes

Drawbacks :
- low spatial resolution (typically about 10-30 meters)
- some channels might be very noisy (not even useful: water absorption bands)

Applications

- Roughly speaking, every application that can benefit from the knowledge of the spectra :
  - Military (detection do not rely on shape)
  - Vegetation monitoring (determine species of trees, impact of pollution)
  - Urban monitoring (enables to separate road, buildings)
  - Harvest monitoring (classification of types of corns)
  - Soil inquiries (determination of the materials)

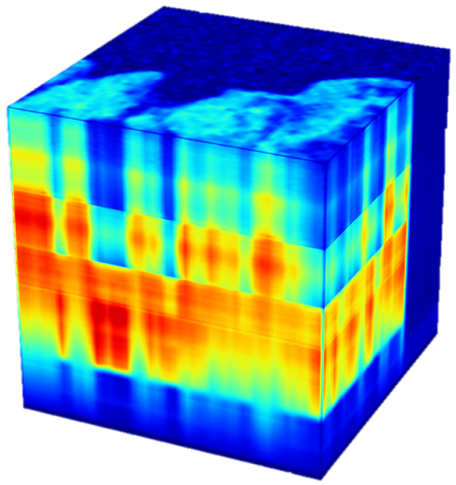
Sensor (Org) Air/Space	Spectra (μm)	No. Bands	Year*
AIS (JPL) A	1.2–2.4	128	1980
AVIRIS (JPL) A	0.4–2.5	224	1987–Present
GERIS-63 (US) A	0.47–2.44	63	1986
CASI (Can) A	0.48–0.805	288	1986
ROSIS (Ger) A	0.45–0.85	128	1992
HYDICE (NRL) A	0.4–2.5	210	1995
LEWIS (NASA) S	0.4–2.5	220	1997**
SEBASS (US) A	8.0–13.0	128	1996–Present
HYPERION (NASA) S	0.42–2.4	224	2000–Present
PROBA/CHRIS (ESA) S	0.41–1.06	63	2001
HYMAP (Aus) A	0.45–2.5	126	1998–Present
ARTEMIS (AFRL) S	0.4–2.5	420	2009–2012
HICO (NRL)	0.38–0.96	128	2009
EnMAP (DLR) S	0.4–2.5	200	2017*
HYSUI (JAXA) S	0.4–2.5	200	2017*
HYSPIRI (NASA) S	0.4–2.5 & 8–10	240	2020
Shalom (Israel–Italy) S	0.4–2.5	220	2018*

Data processing for hyperspectral imaging

- Hyperspectral imaging rises several important data processing challenges, especially as the high number of spectral bands precludes a mere visual inspection of the data :

- A lot of pre-processing :
  - Data calibration : convert the data to data with physical unit
  - Atmosphere compensation : remove the effets (absorption, scattering) of atmosphere, among others
  - …
- Processing for data exploitation :
  - Denoising
  - Compression
  - Target detection
  - Material classification
  - Super-resolution
  - **Unmixing**
  - …
- In this class, we focus on hyperspectral unmixing (HSU). This is due to the fact that HSU relies on the low-rank structure of the hyperspectral image, which is also a cornerstone for other data processing (super-resolution, compression…)

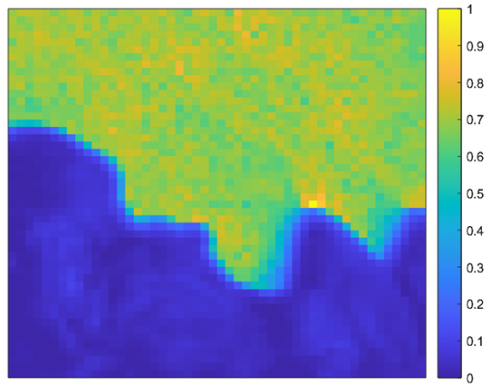
Hyperspectral unmixing (HSU)



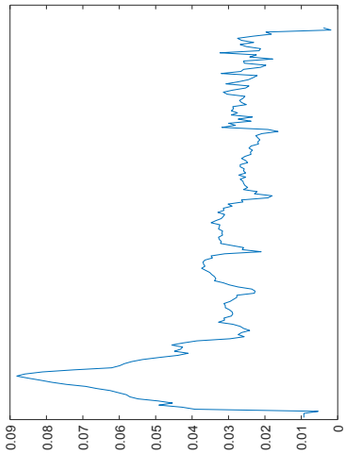
Images from MOFFETT

$=$

$\mathbf{s}^1$



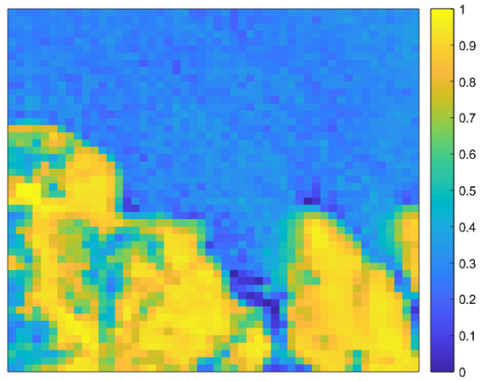
$\mathbf{x}$



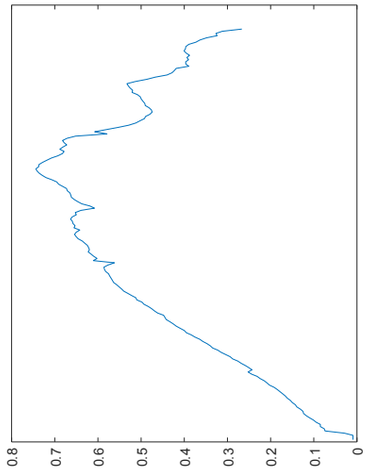
$\mathbf{a}_1$

$+$

$\mathbf{s}^2$



$\mathbf{x}$



$\mathbf{a}_2$

$+ \ldots + \mathbf{N}$

Linear model: $\mathbf{x}_i = \sum_{k=1}^{n} \mathbf{a}_k^* s_{ki}^* + \mathbf{n}_i$ with

$\mathbf{x}_i \in \mathbb{R}^m$: i*th* observation,

$\mathbf{a}_k^* \in \mathbb{R}^m$: k*th* endmember (signature)

$s_{ik}^* \in \mathbb{R}$: abundance of the k*th* endmember in the i*th* pixel

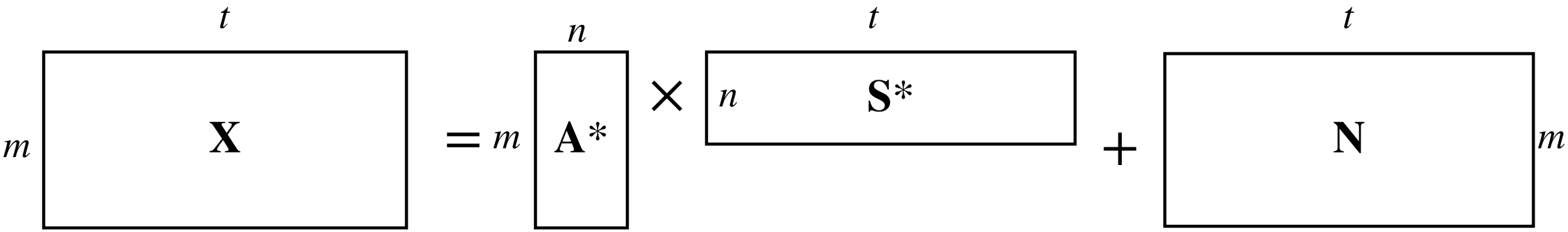
$\mathbf{n}_i \in \mathbb{R}^m$: noise

HSU as a Blind Source separation (BSS) problem

$$\mathbf{x}_i = \sum_{k=1}^{n} \mathbf{a}_k^* s_{ki}^* + \mathbf{n}_i \longrightarrow \boxed{\mathbf{X} = \mathbf{A}^*\mathbf{S}^* + \mathbf{N}}$$

X : m rows observations and t samples columns (m x t)
A* : mixing function (m x n)
S* : sources (n x t) - the sources will be assumed to be sparse
N : noise and model imperfections (m x t)

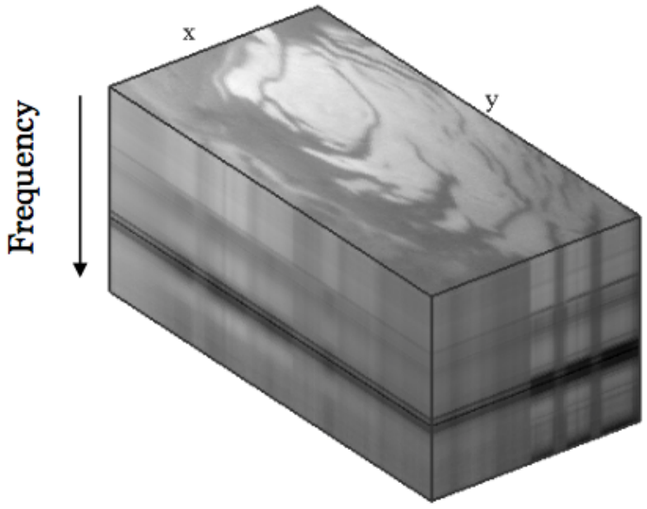
Goal : retrieve A*and S*from the sole knowledge of X
(or more generally: unmix some signals)



Assumption : $m \ll t$

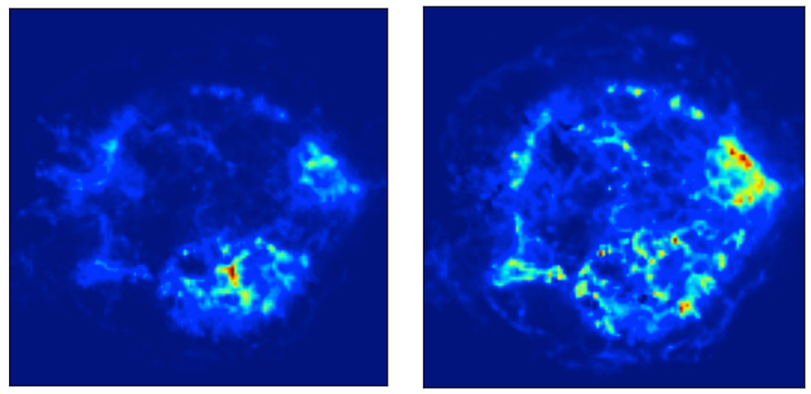
Two cases :
- **Exactly and over-determined** ($r \leq m$)
- Under-determined ($r > m$)

Blind Source Separation (BSS): applications

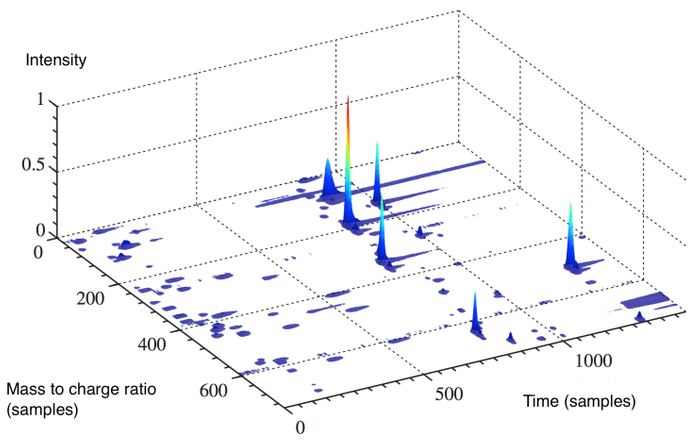


Remote sensing***

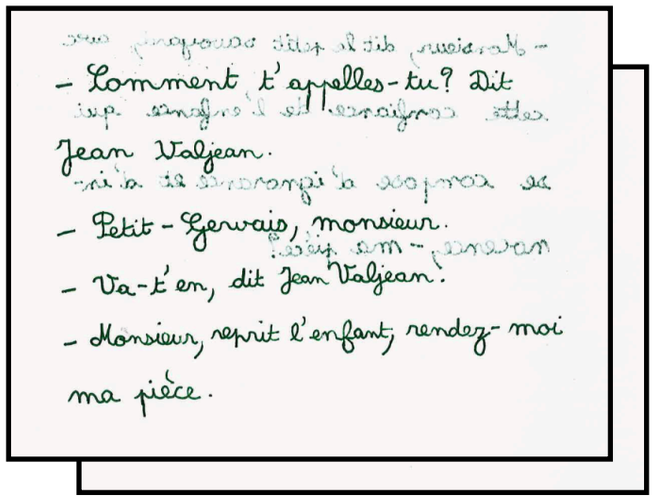
/



Astrophysics****



Spectroscopy**



Show-through removal

Cocktail party problem*

• • •

Figures: courtesy from: *https://cacm.acm.org/news/190656-3d-printed-device-helps-computers-solve-cocktail-party-problem/fulltext, ** Rapin, J. (2014). Décompositions parcimonieuses pour l'analyse avancée de données en spectrométrie pour la Santé, *** Chenot, C. (2017). Parcimonie, diversité morphologique et séparation robuste de sources.**** Courtesy of F. Acero

HSU/BSS vs mere dimensionality reduction

(or: why is this class harder than what we saw in IMA 205?)

IMA 205	**Here**
<u>General goal of unsupervised machine learning:</u> *« Describe the data with a smaller set of latent variables »*	<u>Goal of HSU/BSS</u> *« Recover the (small dimensional) set of latent variables that generated the data set »*

Difference : let $\mathbf{A}^*, \mathbf{S}^*$ be such that

$$\mathbf{X} = \mathbf{A}^*\mathbf{S}^*$$

$$\Rightarrow \mathbf{X} = \mathbf{A}^*\mathbf{P}\mathbf{P}^{-1}\mathbf{S}^* \quad \text{with } \mathbf{P} \text{ any invertible matrix}$$

$$\Rightarrow \mathbf{X} = \mathbf{A}\mathbf{S} \quad \text{with } \mathbf{A} = \mathbf{A}^*\mathbf{P} \text{ and } \mathbf{S} = \mathbf{P}^{-1}\mathbf{S}^*$$

Thus, there is a infinite number of possible solutions which do not correspond to the true generating $\mathbf{A}^*, \mathbf{S}^*$ factors.

Introducing additional priors

We just saw that there was an infinite number of solutions

=> How to determine which ones are the true generating factors ?

Said differently, BSS is an ill-posed problem.

=> need to introduce additional information, or also additional *priors*, on the sought after factors **A***, **S***.

Three main families of priors in BSS :

- ➢ Assume the independence of **S** (ICA)
- ➢ Assume the sparsity of **S** (SBSS)
- ➢ Use non-negativity (NMF) of **A** and **S**
- ➢ (+ deep learning methods)

=> each family has its strengths and weaknesses

Independent component analysis

- Historically, this is the first BSS method

Main principle :

$$\mathbf{X} = \mathbf{A}^*\mathbf{S}^*$$

- The sources $(\mathbf{s}_k^*)_{k=1..n}$ are assumed to be mutually statistically independent but their mixtures are not

=> look for independent estimated sources $(\hat{\mathbf{s}}_k)_{k=1..n}$

- Roughly speaking, two types of methods to solve the problem:
  - Minimize the mutual information
  - Maximize the non-Gaussianity

- A bunch of methods : infomax, JADE, FastICA, EFICA…

- You have seen FastICA in IMA 205

Pros and cons of ICA

Pros :

- **Darmois theorem** : provided that :
  - All the sources $\mathbf{s}_k^*$ are statistically independent
  - There is at most a source following a Gaussian law
  - The mixing matrix $\mathbf{A}$ is (square and) full rank

then, the estimated sources $\hat{\mathbf{S}}$ correspond to the ones $\mathbf{S}^*$ having generated the dataset, up to a (generally inconsequential) scaling and permutation indeterminacy.

Cons

- The Darmois theorem result only holds in the absence of noise, which is impractical
- The statistical independence of the sources can be a too strong assumption, especially for HSU (why?)

Outline of the class

I - Sparse Blind Source Separation / Hyperspectral unmixing

II - Nonnegative Matrix Factorization

Sparse BSS / HSU

Sparsity

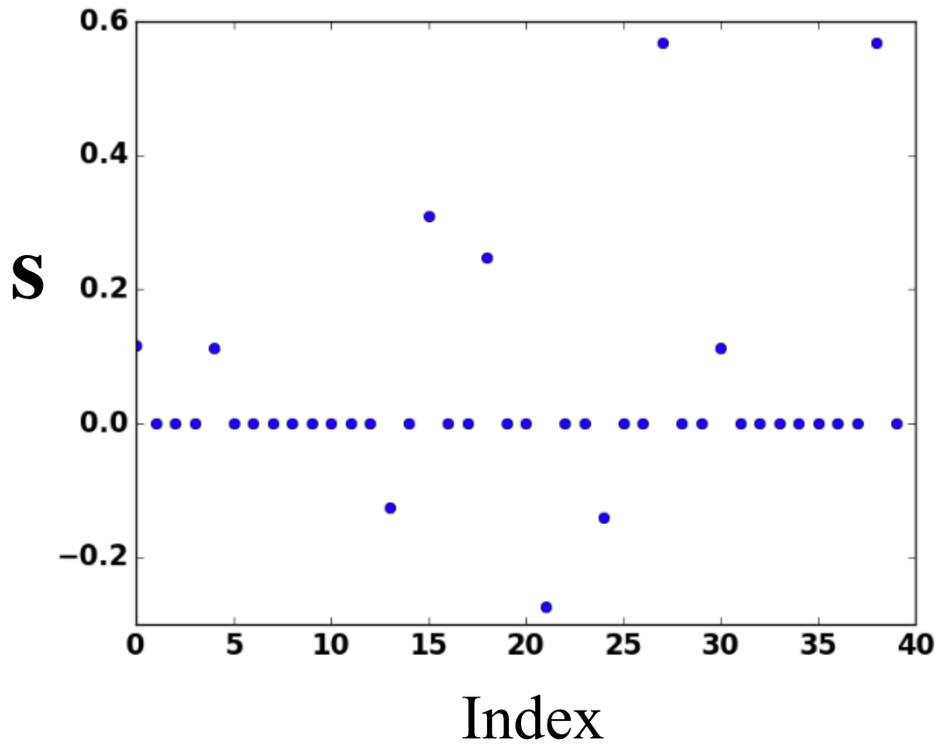
- Generally speaking, sparsity amounts to represent a signal with as few variables as possible.

Exact sparsity

- A signal $\mathbf{s} \in \mathbb{R}^n$ is said to be k-sparse if only $k \ll t$ of its element are non-zeros:

$$\|\mathbf{s}\|_0 = k \ll t$$

with the pseudo-norm $\|.\|_0$ being the cardinal of the support of $\mathbf{s}$.



Sparsity

- Example of real-life exactly sparse signal:

Emission spectrum of hydrogen

Approximately sparse signals

- Most real-life signals are not exactly sparse :

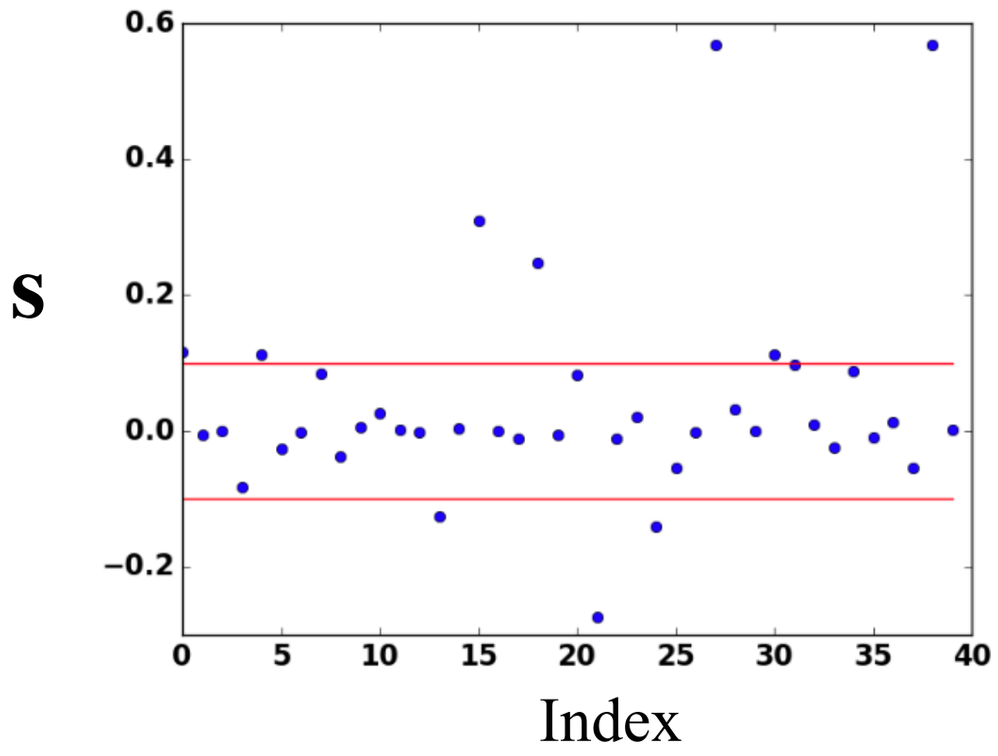
$$||\mathbf{s}||_0 \simeq t$$

Approximate sparsity

- Only a small number k of the signal samples have a *large* amplitude.

=> the signal can be well approximated by a k-sparse signal

- This is for instance, if the sorted magnitudes of the signal samples follow a power law

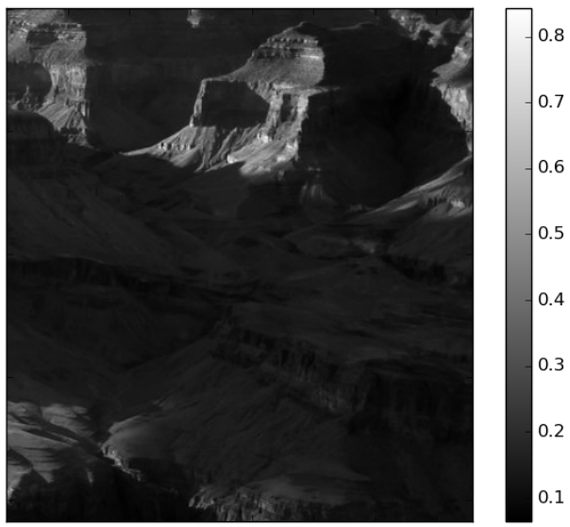


Approximately sparse signals in transformed domains

Most signals are not (exactly of approximately) sparse in the direct domain

Approximate sparsity in transformed domain

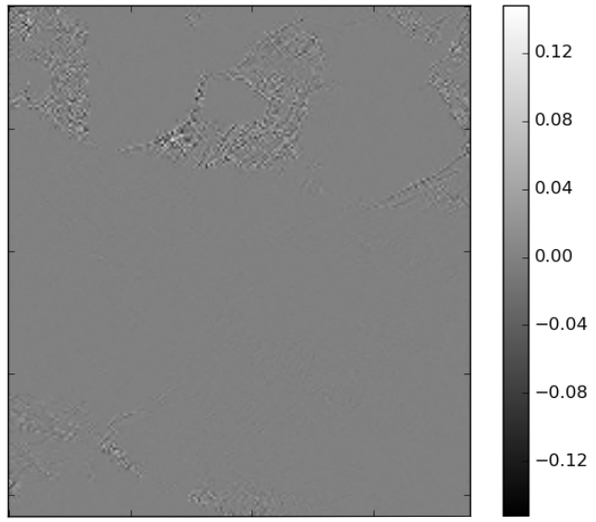
- However, they admit an approximately sparse representation in a *transformed* domain $\mathbf{\Phi}$
- Simple example: a sinus is represented in the Fourier domain by two Dirac.
- In practice, it is often better to retain the spatial information (*e.g.* by using a multi-scale transform - wavelet)
- It is also possible to *learn* $\mathbf{\Phi}$.

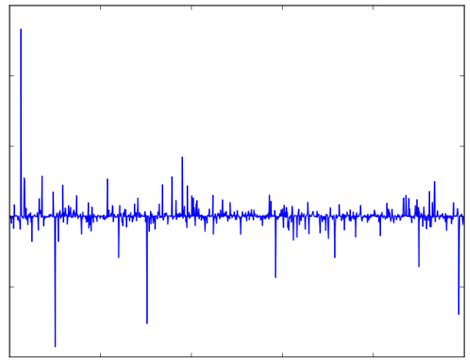


$\mathbf{s}$

Direct domain

$\mathbf{\Phi}$



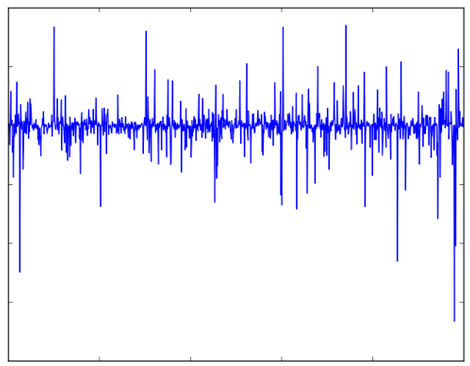


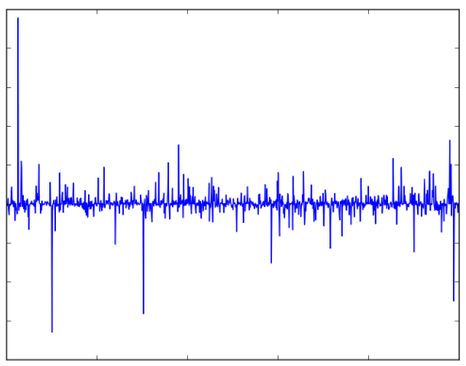
$\mathbf{s\Phi}$

Direct

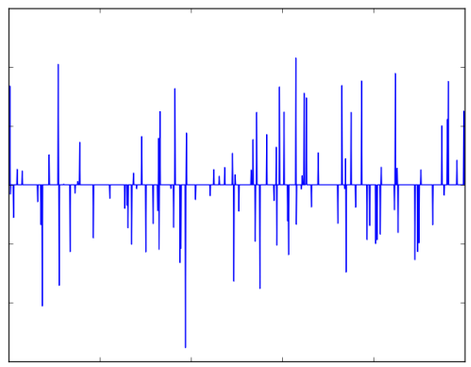
Interpretation of sparse BSS (1/3) : intuition

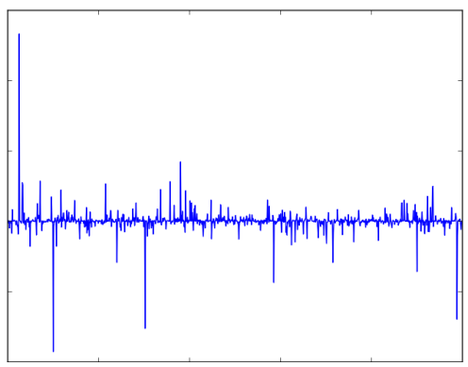
Now that we introduced sparsity, how can we use it for BSS?

$\mathbf{X}_{1,:}$ 

$\mathbf{X}_{2,:}$ 

$=$ $\left[\mathbf{a}_{:,1}, \mathbf{a}_{:,2}\right] \times$





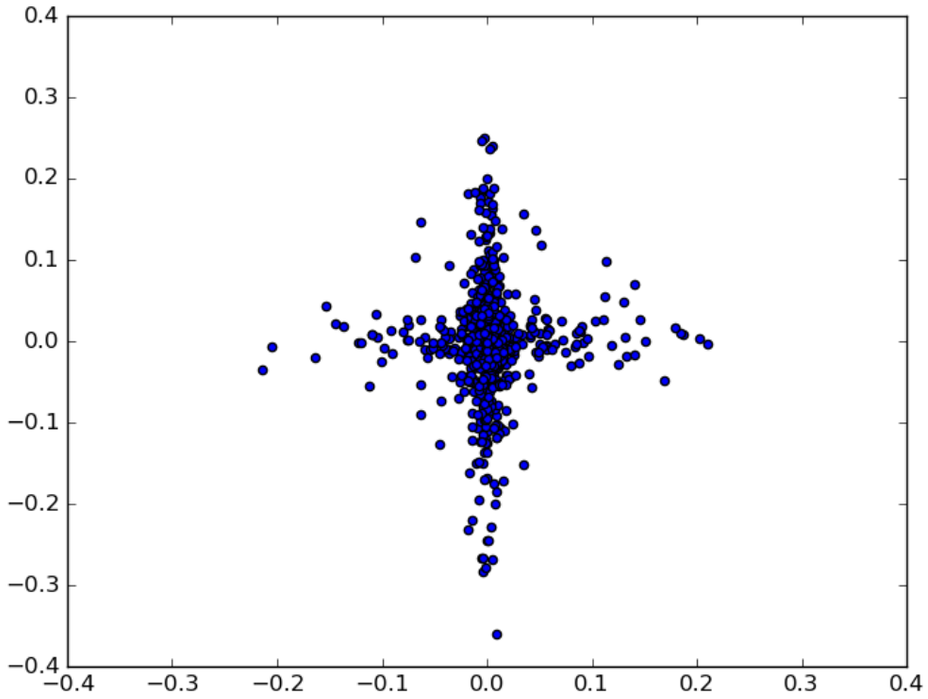
$\mathbf{S}_{1,:}$

$\mathbf{S}_{2,:}$

=> mixing reduces the sparsity. To recover the sources, we must find the sparsest signals.

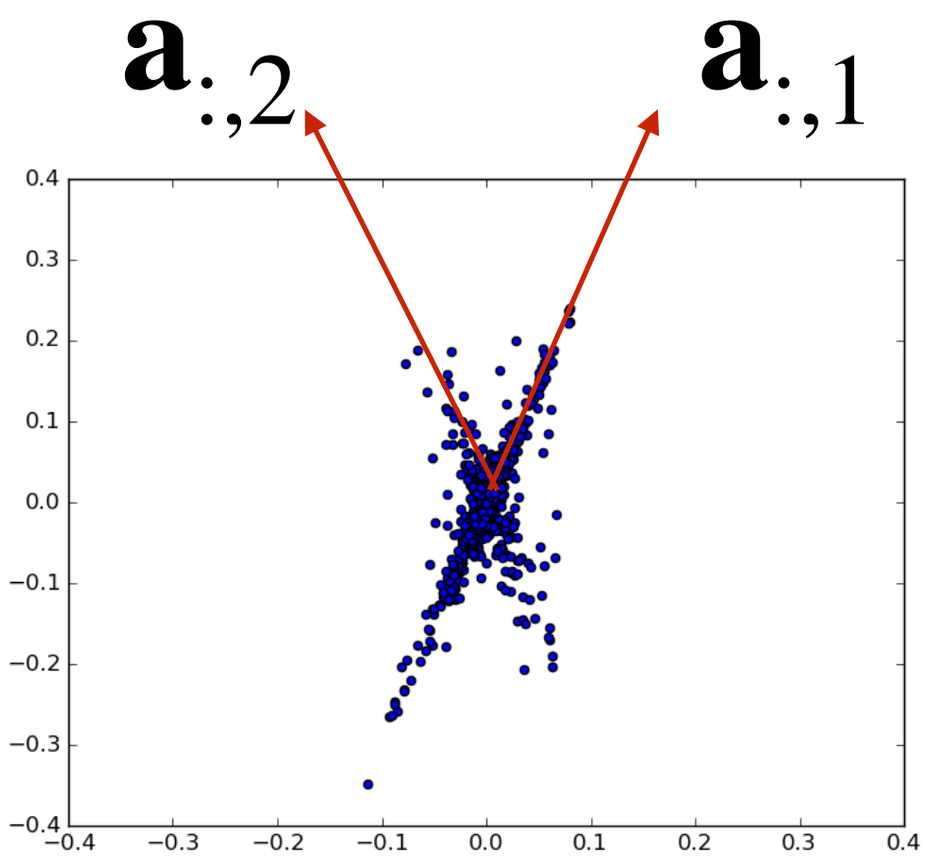
Interpretation of sparse BSS (2/3) : geometrical view

$\mathbf{s}_{2,:}$



$\mathbf{s}_{1,:}$

$\mathbf{x}_{2,:}$



$\mathbf{x}_{1,:}$

- Due to sparsity, the scatter plot of the sources has a star shape
- Multiplying the sources by $\mathbf{A}$* changes the direction of the axes

=> recovering the sources amounts to back-project the observations on the canonical axes

Interpretation of sparse BSS (3/3) : statistical view (1/3)

Recovering the sources amounts to back-project the observations on the canonical axes

- **How to do that?**

=> rewrite sparse BSS as a Maximum A Posteriori (MAP) estimation.

$$\arg\max_{\mathbf{A},\mathbf{S}} P(\mathbf{A},\mathbf{S}\,|\,\mathbf{X})$$

But :

$$P(\mathbf{A},\mathbf{S}\,|\,\mathbf{X}) \propto P(\mathbf{X}\,|\,\mathbf{A},\mathbf{S})P(\mathbf{A},\mathbf{S}) \qquad \text{(Bayes' rule)}$$

$$= P(\mathbf{X}\,|\,\mathbf{A},\mathbf{S})P(\mathbf{A})P(\mathbf{S}) \qquad (\mathbf{A},\mathbf{S} \text{ independent})$$

Thus, the MAP writes as

$$\arg\max_{\mathbf{A},\mathbf{S}} P(\mathbf{X}\,|\,\mathbf{A},\mathbf{S})P(\mathbf{A})P(\mathbf{S})$$

Interpretation of sparse BSS (3/3) : statistical view (2/3)

$$\arg\max_{\mathbf{A},\mathbf{S}} P(\mathbf{X}\,|\,\mathbf{A},\mathbf{S})P(\mathbf{A})P(\mathbf{S})$$

$$= \arg\min_{\mathbf{A},\mathbf{S}} -\log(P(\mathbf{X}\,|\,\mathbf{A},\mathbf{S})) - \log(P(\mathbf{A})) - \log(P(\mathbf{S})) \qquad \text{(rather use } -\log\text{)}:$$

Assuming:

- A Gaussian noise: $P(\mathbf{X}\,|\,\mathbf{A},\mathbf{S}) \propto e^{\frac{||-\mathbf{X}-\mathbf{AS}||_F^2}{2\sigma^2}}$
- An exponential distribution for $\mathbf{S} = e^{-\beta||\mathbf{S}||_1}$
- A uniform distribution for $\mathbf{A}$,

we obtain:

$$\arg\min_{\mathbf{A},\mathbf{S}} \frac{1}{2\sigma^2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \beta\|\mathbf{S}\|_1$$

$$\Rightarrow \arg\min_{\mathbf{A},\mathbf{S}} \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \lambda\|\mathbf{S}\|_1$$

Interpretation of sparse BSS (3/3) : statistical view (3/3)

$$\arg\min_{\mathbf{A},\mathbf{S}} \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \lambda\|\mathbf{S}\|_1$$

Unfortunately, for any $\mathbf{A}, \mathbf{S}$ and $\alpha > 1$,

$$\frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \lambda\|\mathbf{S}\|_1 > \frac{1}{2}\|\mathbf{X}-\mathbf{A}\alpha\alpha^{-1}\mathbf{S}\|_F^2 + \lambda\|\alpha^{-1}\mathbf{S}\|_1$$

- The solution $\mathbf{S}' = \alpha^{-1}\mathbf{S}$ and $\mathbf{A}' = \alpha\mathbf{A}$ has a lower cost than $\mathbf{S}, \mathbf{A}$
- The larger α, the better the results

  => this known as ***scaling indeterminacy***

More generally, we need to modify the cost function and limit the energy of the columns of $\mathbf{A}$ so that we do not obtain degenerated solutions.

Oblique constraint

We require each column of $\mathbf{A}$ to have a unit energy:

$$l_{\{\forall i \in [1,n]; \|\mathbf{a}_{:,j}\|_{\ell_2}^2 \leq 1\}}(\mathbf{A})$$

Sparse BSS : cost function

We obtain the following cost-function to minimize to solve the sparse BSS problem

$$\arg\min_{\mathbf{A}\in\mathbb{R}^{m\times n},\ \mathbf{S}\in\mathbb{R}^{n\times t}} \frac{1}{2}\|\mathbf{X}-\mathbf{A}\mathbf{S}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \iota_{\{\forall i\in[1,n];\|\mathbf{a}_{:,j}\|_{\ell_2}^2\leq 1\}}(\mathbf{A})$$

Data-fidelity **Sparsity** **Oblique constraint**

λ : Regularization parameters

$\mathbf{\Phi_S^T}$: is a sparsifying transform

X : m rows observations and t samples columns (m x t)

A : mixing function (m x n)

S : sources (n x t)

N : noise and model imperfections (m x t)

How to minimize the cost function?

We obtain the following cost-function to minimize to solve the sparse BSS problem

$$\arg\min_{\mathbf{A}\in\mathbb{R}^{m\times n},\ \mathbf{S}\in\mathbb{R}^{n\times t}} \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \iota_{\{\forall i\in[1,n];\|\mathbf{a}_{:,j}\|_{\ell_2}^2\leq 1\}}(\mathbf{A})$$

Data-fidelity **Sparsity** **Oblique constraint**

λ : Regularization parameters

$\mathbf{\Phi}_{\mathbf{S}}^{\mathbf{T}}$: is a sparsifying transform

Challenges:

1) **Non-smooth** (needs advanced optimization tools: proximal operators)
2) **Non-convex** (non-unique minima)

=> **Difficult optimization problem**

Challenge 1 : non-smooth problem

$$\arg\min_{\mathbf{A}\in\mathbb{R}^{m\times n},\ \mathbf{S}\in\mathbb{R}^{n\times t}} \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \iota_{\{\forall i\in[1,n];\|\mathbf{a}_{:,j}\|_{\ell_2}^2\leq 1\}}(\mathbf{A})$$

Let's simplify the problem by fixing $\mathbf{A}$:

$$\arg\min_{\mathbf{S}\in\mathbb{R}^{n\times t}} \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \lambda\|\mathbf{S}\|_1$$

- Good news: the problem is now convex ! (why?)
- It is still non-smooth because of the $\|.\|_1$ term…
- More generally, it is a problem of the form:

$$\arg\min_{\mathbf{S}\in\mathbb{R}^{n\times t}} h(\mathbf{A},\mathbf{S}) + \mathcal{G}(\mathbf{S})$$

With:

- The function $\mathbf{S}\rightarrow h(\mathbf{A},\mathbf{S})$ smooth and having a Lipschitz gradient
- The $\mathbf{S}\rightarrow\mathcal{G}(\mathbf{S})$ being a closed proper convex function:

  $\mathbf{epi}\ \mathcal{G} = \{(\mathbf{x},t)\in\mathbb{R}^n\times\mathbb{R}\,|\,\mathcal{G}(\mathbf{x})\leq t\}$ is a non-empty closed convex set

Recall : gradient descent for smooth optimization (1/2)

$$\arg \min_{\mathbf{S} \in \mathbb{R}^{n \times t}} h(\mathbf{A}, \mathbf{S}) + \mathcal{G}(\mathbf{S})$$

Let's us consider an even simpler problem:

$$\arg \min_{\mathbf{S} \in \mathbb{R}^{n \times t}} h(\mathbf{A}, \mathbf{S})$$

(Rem. : $\mathbf{A}$ could be omitted in the notations)

- That's now easy, we can use a **<u>gradient descent algorithm</u>**:

```
Initialize S^(0)
k = 0
while not converged do:
    S^(k+1) = S^(k) − γ ∇h(S^(k))
    k ← k + 1
end
return S^(k)
```

- With $\gamma < \frac{1}{L}$, the Lipschitz constant of ∇h

Recall : gradient descent for smooth optimization (2/2)

Initialize $\mathbf{S}^{(0)}$

$k = 0$

while **not** converged do:

$\mathbf{S}^{(k+1)} = \mathbf{S}^{(k)} - \gamma \nabla f(\mathbf{S}^{(k)})$

$k \leftarrow k + 1$

end

return $\mathbf{S}^{(k)}$

- If we go back to our specific example:

$$\arg \min_{\mathbf{S} \in \mathbb{R}^{n \times t}} h(\mathbf{A}, \mathbf{S}) = \frac{1}{2} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_F^2$$

- The gradient is :

$$\nabla f(\mathbf{S}^{(k)}) = \mathbf{A}^T(\mathbf{A}\mathbf{S} - \mathbf{X}) \qquad \text{(try to do the maths!)}$$

- And we can choose :

$$\gamma \simeq \frac{1}{\|\mathbf{A}^T\mathbf{A}\|_2} \qquad \text{(Why ?)}$$

Non-smooth optimization and proximal operators

Let us now consider the difficult part:

$$\arg \min_{\mathbf{S} \in \mathbb{R}^{n \times t}} \mathcal{G}(\mathbf{S})$$

This is a non-smooth problem, so the gradient does not exist at some points.

In this case, a widely used tool is the proximal operator.

$$\texttt{prox}_{\mathcal{G}(.)}(\mathbf{S}) = \arg\min_{\mathbf{Y}} \mathcal{G}(\mathbf{Y}) + \frac{1}{2}\|\mathbf{S} - \mathbf{Y}\|^2$$

- We do not solve the initial non-smooth problem at once, but rather a local minimization problem
- The regularization by the ℓ_2-norm makes the problem easier to solve

=> In practice, several non-smooth functions have a closed-form prox.

Example of proximal operators (1/2)

- If $\mathcal{G}(\,.\,)$ is a the **indicator function** of a closed non-empty convex set C :

$$I_C(x) = \begin{cases} 0 & x \in C \\ +\infty & x \notin C, \end{cases}$$

Then:

$$\mathbf{prox}_f(\mathbf{v}) = \Pi_C(\mathbf{v}) = \arg\min_{\mathbf{x} \in C} \|\mathbf{x} - \mathbf{v}\|_2$$

The proximal operator is the Euclidean projection onto C

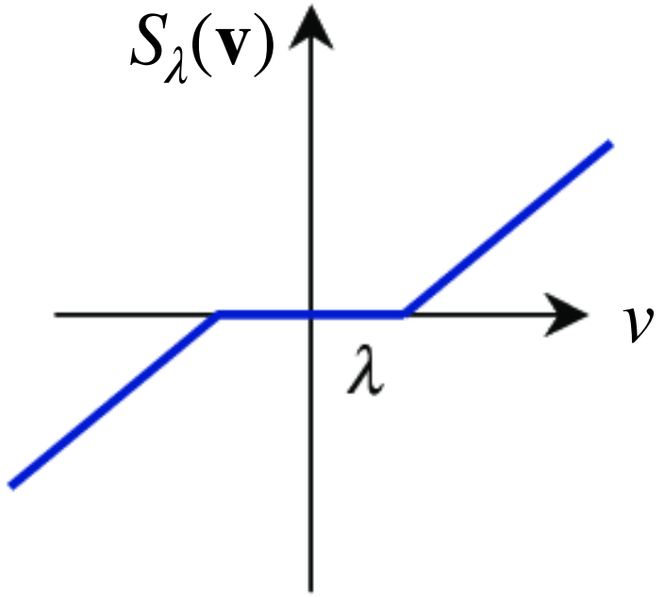
Example of proximal operators (2/2)

- If $\mathcal{G}(\,.\,)$ is the ℓ_1 norm :

$$\mathcal{G}(x) = \lambda \|x\|_1 \text{ for } x \in \mathbb{R}^n$$

Then:

$$(\mathbf{prox}_{\lambda\|.\|_1}(\mathbf{v}))_i = S_\lambda(\mathbf{v}) = \begin{cases} v_i - \lambda & v_i \geq \lambda \\ 0 & |v_i| \leq \lambda \\ v_i + \lambda & v_i \leq -\lambda. \end{cases}$$



The proximal operator of the ℓ_1-norm is the soft-thresholding operator

Proximal operators : properties

Fixed point property

The point $\mathbf{S}^*$ minimizes $\mathcal{G}(\mathbf{S})$ if and only if

$$\mathbf{prox}_{\mathcal{G}(.)}(\mathbf{S}^*) = \mathbf{S}^*$$

Firm non-expansiveness

Let $x, y \in \mathbb{R}^n$, then

$$\|\mathbf{prox}_{\mathcal{G}}(\mathbf{s}_1) - \mathbf{prox}_{\mathcal{G}}(\mathbf{s}_2)\|_2^2 \leq (\mathbf{s}_1 - \mathbf{s}_2)^T(\mathbf{prox}_{\mathcal{G}}(\mathbf{s}_1) - \mathbf{prox}_{\mathcal{G}}(\mathbf{s}_2))$$

N.B. : stronger than mere non-expansiveness :

$$\|\mathbf{prox}_{\mathcal{G}}(\mathbf{s}_1) - \mathbf{prox}_{\mathcal{G}}(\mathbf{s}_2)\|_2^2 \leq \|\mathbf{s}_1 - \mathbf{s}_2\|_2^2$$

Non-smooth minimization method: the proximal point method

Coming back to

$$\arg \min_{\mathbf{S} \in \mathbb{R}^{n \times t}} \mathcal{G}(\mathbf{S})$$

How to perform the minimization if we know the proximal operator of $\mathcal{G}(\,.\,)$?

Use for instance the **proximal point method**:

```
Initialize S^(0)
k = 0
while not converged do:
    S^(k+1) = prox_{G(.)}(S^(k))
    k ← k + 1
end
return S^(k)
```

This algorithm is based on the fact that the proximal operator is a firmly non-expensive operator, and on the fixed point property : if $\mathbf{s}^*$ a minimizer of $\mathcal{G}(\,.\,)$, then

$$\mathbf{prox}_{\mathcal{G}(.)}(\mathbf{s}^*) = \mathbf{s}^*$$

Non-smooth minimization method: summary so far

$$\arg \min_{\mathbf{S} \in \mathbb{R}^{n \times t}} h(\mathbf{A}, \mathbf{S})$$

- $h(\,.\,)$ smooth, convex as a function of $\mathbf{S}$

- Gradient descent algorithm:

```
Initialize S^(0)
k = 0
while not converged do:
    S^(k+1) = S^(k+1) − γ ∇h(S^(k))
    k ← k + 1
end
return S^(k)
```

$$\arg \min_{\mathbf{S} \in \mathbb{R}^{n \times t}} \mathcal{G}(\mathbf{S})$$

- $\mathcal{G}(\,.\,)$ non-smooth, convex

- Proximal operator:

$$\mathbf{prox}_{\mathcal{G}(.)}(\mathbf{S}) = \arg \min_{\mathbf{Y}} \mathcal{G}(\mathbf{Y}) + \frac{1}{2}\|\mathbf{S} - \mathbf{Y}\|^2$$

- Proximal point method:

```
Initialize S^(0)
k = 0
while not converged do:
    S^(k+1) = prox_{G(.)}(S^(k))
    k ← k + 1
end
return S^(k)
```

Non-smooth minimization method: sum of previous cases

$$\arg \min_{\mathbf{S} \in \mathbb{R}^{n \times t}} h(\mathbf{A}, \mathbf{S}) + \mathcal{G}(\mathbf{S})$$

- $h(\,.\,)$ smooth, convex as a function of $\mathbf{S}$, $\mathcal{G}(\,.\,)$ non-smooth, convex
- Forward-backward splitting method:

```
Initialize S^(0)
k = 0
while not converged do:
    S^(k+1) = prox_{γ G(.)}(S^(k) − γ ∇h(S^(k)))
    k ← k + 1
end
return S^(k)
```

- Also known as proximal gradient method :
  - Perform a gradient step on the smooth part of the cost function
  - Perform a proximal step on the non-smooth part

Non-smooth minimization method: example

$$\arg \min_{\mathbf{S} \in \mathbb{R}^{n \times t}} \frac{1}{2} \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_F^2 + \lambda \|\mathbf{S}\|_1$$

- Forward-backward splitting method = Iterative Shrinkage Thresholding Algorithm (ISTA)

Initialize $\mathbf{S}^{(0)}$

$k = 0$

Choose $\gamma = 0.9/\|\mathbf{A}^T\mathbf{A}\|_*$ (why?)

while **not** converged do:

$\mathbf{S}^{(k+1)} = S_{\lambda\gamma}(\mathbf{S}^{(k)} - \gamma\mathbf{A}^T(\mathbf{A}\mathbf{S} - \mathbf{X}))$

$k \leftarrow k + 1$

end

return $\mathbf{S}^{(k)}$

- It is also possible to accelerate this algorithm with inertial / extrapolation moments (see FISTA for an example)

How to minimize the cost function?

We obtain the following cost-function to minimize to solve the sparse BSS problem

$$\arg\min_{\mathbf{A}\in\mathbb{R}^{m\times n},\ \mathbf{S}\in\mathbb{R}^{n\times t}} \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \iota_{\{\forall i\in[1,n];\|\mathbf{a}_{:,j}\|_{\ell_2}^2=1\}}(\mathbf{A})$$

Data-fidelity **Sparsity** **Oblique constraint**

λ : Regularization parameter

Challenges:

1) **Non-smooth** (needs advanced optimization tools: proximal operators) **OK**
2) **Non-convex** (non-unique minima)

=> **Difficult optimization problem**

What can we say about the non-convexity?

- Currently, no solution for solving general non-convex optimization problems => only heuristics

- But we can still obtain a few results, since our cost function is particular

$$\arg \min_{\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{S}\in\mathbb{R}^{n\times t}} h(\mathbf{A},\mathbf{S}) + \mathcal{J}(\mathbf{A}) + \mathcal{G}(\mathbf{S})$$

- It's non-convex but convex in **A** and **S** separately

=> this is called multi-convex

- Using the multi-convexity (and other hypotheses), some algorithms exist ensuring the convergence to a fixed point of the cost function (i.e. **local** min. / max. or saddle point)

PALM

$$\arg \min_{\mathbf{A}\in\mathbb{R}^{m\times n},\mathbf{S}\in\mathbb{R}^{n\times t}} h(\mathbf{A},\mathbf{S}) + \mathcal{J}(\mathbf{A}) + \mathcal{G}(\mathbf{S})$$

- Main idea: alternate between the two variables, for which the problem is convex

- Proximal Alternating Linear Minimization (PALM)

Initialize $\mathbf{S}^{(0)}, \mathbf{A}^{(0)}$

$k = 0$

while **not** converged do:

$\mathbf{S}^{(k+1)} = \mathbf{prox}_{\gamma\mathcal{G}(.)}(\mathbf{S}^{(k)} - \gamma \nabla h(\mathbf{A}^{(k)}, \mathbf{S}^{(k)}))$

$\mathbf{A}^{(k+1)} = \mathbf{prox}_{\eta\mathcal{J}(.)}(\mathbf{A}^{(k)} - \eta \nabla h(\mathbf{A}^{(k)}, \mathbf{S}^{(k+1)}))$

$k \leftarrow k + 1$

end

return $\mathbf{S}^{(k)}, \mathbf{A}^{(k)}$

PALM : application to sparse BSS

$$\arg\min_{\mathbf{A}\in\mathbb{R}^{m\times n},\ \mathbf{S}\in\mathbb{R}^{n\times t}} \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \lambda\|\mathbf{S}\|_1 + \iota_{\{\forall i\in[1,n];\|\mathbf{a}_{:,j}\|_{\ell_2}^2\leq 1\}}(\mathbf{A})$$

- Proximal Alternating Linear Minimization (PALM)

Initialize $\mathbf{S}^{(0)}, \mathbf{A}^{(0)}$

$k = 0$

while **not** converged do:

$$\mathbf{S}^{(k+1)} = S_{\gamma\lambda}\left(\mathbf{S}^{(k)} + \frac{0.9}{\|\mathbf{A}^{(k)T}\mathbf{A}^{(k)}\|_*}\mathbf{A}^{(k)T}\left(\mathbf{X}-\mathbf{A}^{(k)}\mathbf{S}^{(k)}\right)\right)$$

$$\mathbf{A}^{(k+1)} = \Pi_{\|.\|_2\leq 1}\left(\mathbf{A}^{(k)} + \frac{0.9}{\|\mathbf{S}^{(k+1)}\mathbf{S}^{(k+1)^T}\|_*}\left(\mathbf{X}-\mathbf{A}^{(k)}\mathbf{S}^{(k+1)}\right)\mathbf{S}^{(k+1)^T}\right)$$

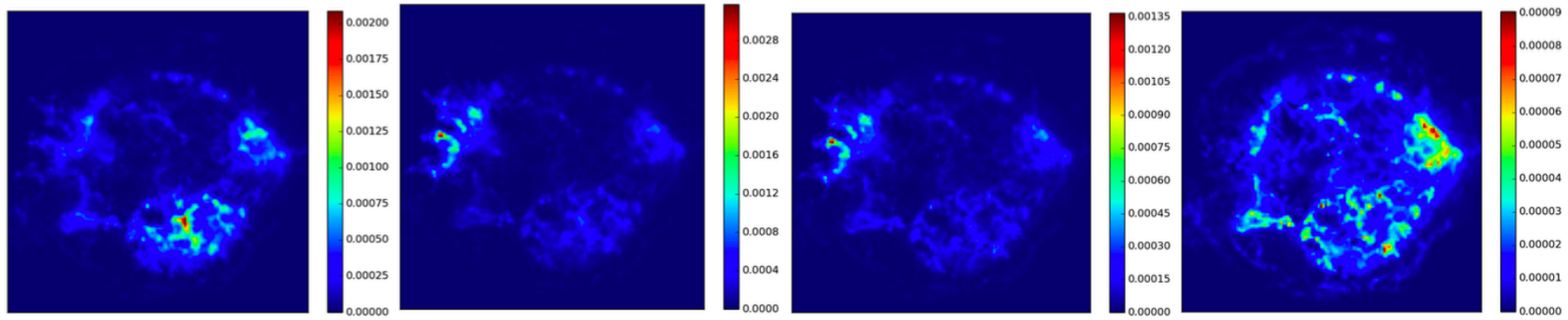
$k \leftarrow k+1$

end

return $\mathbf{S}^{(k)}, \mathbf{A}^{(k)}$

Exemple of application : Chandra data

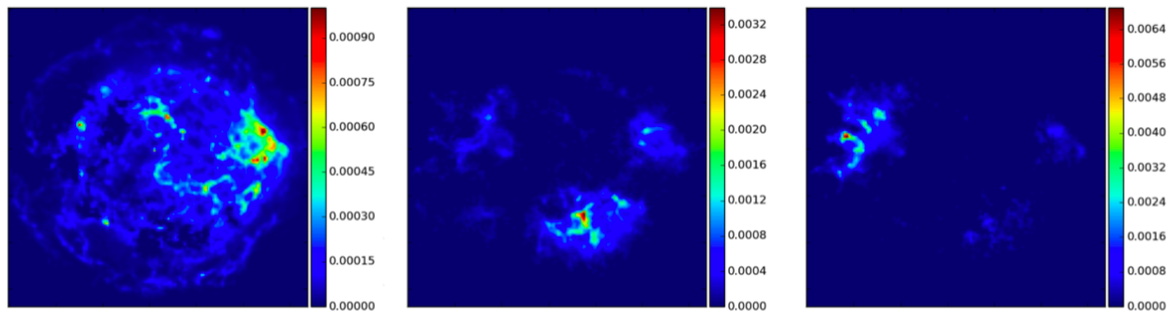
X



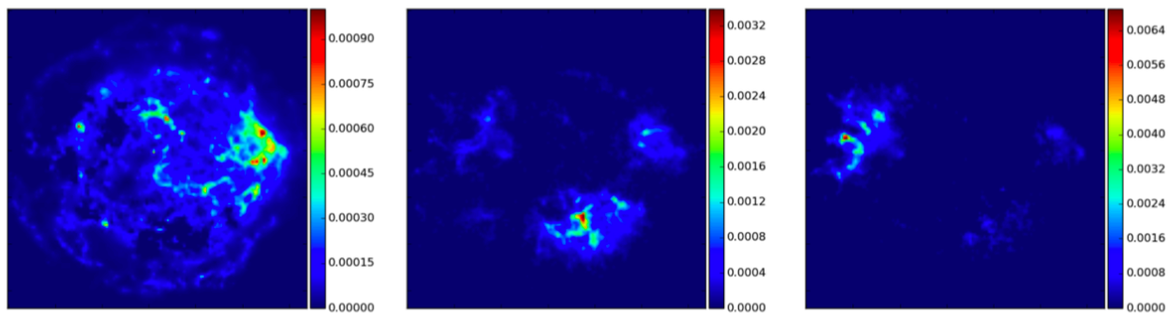
(a) $\mathbf{x}_1$ (b) $\mathbf{x}_5$ (c) $\mathbf{x}_9$ (d) $\mathbf{x}_{30}$

Synchrotron Fe 1 Fe 2

S*



S



Residual