

Applications des modèles de rang faible en traitement du signal

Matthieu Puigt

Université du Littoral Côte d'Opale, LISIC – UR 4491, Calais & Longuenesse

18ème Édition de l'École d'Été en Traitement du Signal et des Images – Peyresq

juin 2024



Summary

- 1 Foreword
- 2 Classical applications of low-rank models
 - Applications of low-rank approximations
 - Applications of low-rank + sparse approximations
 - Applications of low-rank data completion
- 3 In-situ sensor calibration
 - Fixed sensor calibration
 - Mobile sensor calibration
- 4 Joint multispectral image restoration and unmixing
 - Introduction, Motivation and Problem Statement
 - Proposed methods and Some Results
- 5 Low-rankness in the AI Era
- 6 Discussion
 - Matching Need and Solution

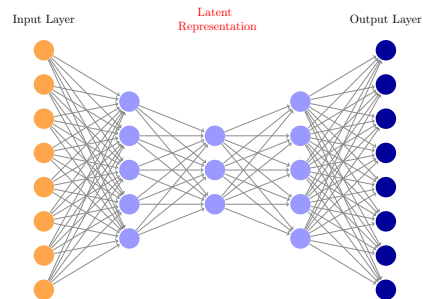
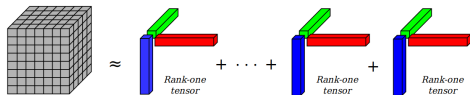
Summary

- 1 Foreword
- 2 Classical applications of low-rank models
- 3 In-situ sensor calibration
- 4 Joint multispectral image restoration and unmixing
- 5 Low-rankness in the AI Era
- 6 Discussion



Low-rank approximations

- Matrix and tensor decompositions
- Neural nets (e.g., (V)AEs, CNNs)



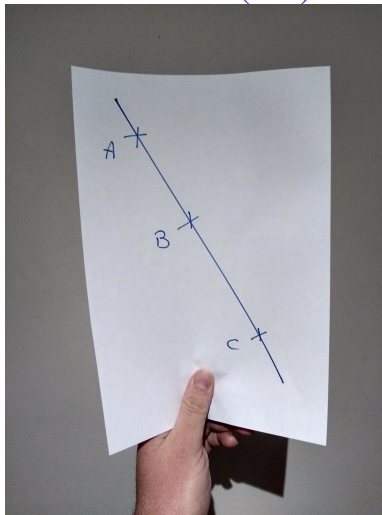
Foreword (2/4)

Global vs local low-rankness

$$X_{(n \times m)} \approx G_{(n \times k)} \cdot F_{(k \times m)}$$

- Global low-rankness : $k < \min(n, m)$

Foreword (2/4)

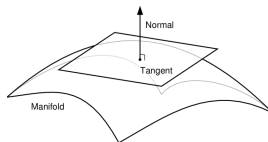
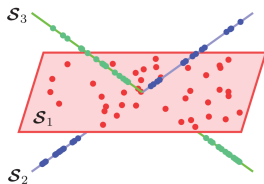


Foreword (2/4)

Global vs local low-rankness

$$X_{(n \times m)} \approx G_{(n \times k)} \cdot F_{(k \times m)}$$

- Global low-rankness : $k < \min(n, m)$
- Local low-rankness :
 - ◊ Latent variable sparsity (k can be greater than $\min(n, m)$) $\iff$ union of low-rank subspaces
 - ◊ Manifold approximation $\iff$ nonlinear problems



Left : Union of subspaces (source : (QU et al. 2023)). Right : Manifold tangent approximation (source : (EDELMAAN, ARIAS et SMITH 1998))

Foreword (3/4)

The Great Convergence

- Low-rank models are ubiquitous (e.g., in signal/image processing and in machine learning)
- Examples with a matrix factorization expression : $X \approx G \cdot F$

Method	$\mathcal{D}(X, G \cdot F)$	Constraint on G	Constraint on F
SVD	$\ X - G \cdot F\ _F^2$	$G^T G = I$	$F^T F = \Lambda$
k -means	$\ X - G \cdot F\ _F^2$	none	$F^T F = I$ $f_{ij} = \{0, 1\}$
k -medians	$\ X - G \cdot F\ _1$	none	$F^T F = I$ $f_{ij} = \{0, 1\}$
NMF (D. D. LEE et H. S. SEUNG 2001)	$\ X - G \cdot F\ _F^2$ or $KL(X, G \cdot F)$	$g_{ij} \geq 0$	$f_{ij} \geq 0$
MMV (TROPP 2006)	$\ X - G \cdot F\ _F^2$	none	structured sparsity ($\ell_{2,1}$)
CX (X. LI et PANG 2010)	$\ X - G \cdot F\ _F^2$	columns of G extracted from X	none
Convex NMF (DING, T. LI et JORDAN 2008)	$\ X - G \cdot F\ _F^2$	columns of G extracted from X	$f_{ij} \geq 0$
Robust convex NMF (KUMAR et SINDHWANI 2015)	$\ X - G \cdot F\ _1$	columns of G extracted from X	$f_{ij} \geq 0$

Foreword (4/4)

What this talk is and **is not** about

- ✓ Some applications of low-rank models/approximations
 - Fast overview of “classical” applications
 - Some insights on a few problems
- ✗ An exhaustive and detailed list of methods for each application

Summary

- 1 Foreword
- 2 Classical applications of low-rank models
 - Applications of low-rank approximations
 - Applications of low-rank + sparse approximations
 - Applications of low-rank data completion
- 3 In-situ sensor calibration
- 4 Joint multispectral image restoration and unmixing
- 5 Low-rankness in the AI Era
- 6 Discussion

Introduction

“Classical” applications of low-rank models

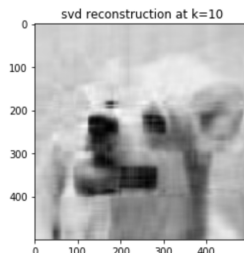
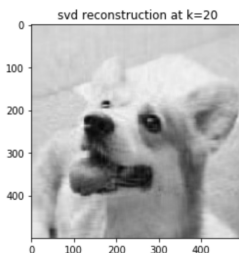
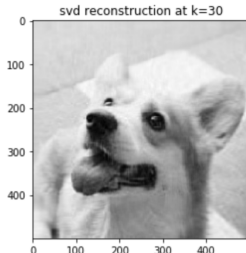
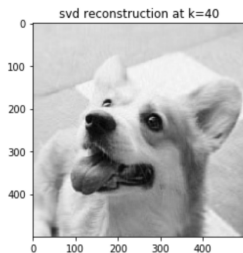
- ① Low-rank approximations (e.g., SVD, NMF) : $X \approx G \cdot F$
- ② Low-rank + sparse approximation : $X \approx G \cdot F + S$
- ③ Weighted low-rank approximations : $W \circ X \approx W \circ (G \cdot F)$

Data compression

- e.g., SVD-based data compression

$$\boxed{X} = \sigma_1 \cdot \boxed{\underline{u}_1 \cdot \underline{v}_1^T} + \sigma_2 \cdot \boxed{\underline{u}_2 \cdot \underline{v}_2^T} + \dots + \sigma_k \cdot \boxed{\underline{u}_k \cdot \underline{v}_k^T}$$

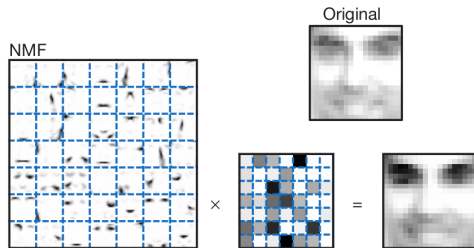
Patterns : Most important 2nd most k -th most



Dictionary/Feature learning

$$X \approx G \cdot F$$

- X matrix of (patches of) unfolded data
- G matrix of “atoms” or “features”
 - ↳ atoms may be sparse (RUBINSTEIN, ZIBULEVSKY et ELAD 2009)
- F matrix of weights
 - ↳ In dictionary learning, weights are sparse
- X , G , and F may be non-negative → NMF



Part-based decomposition with NMF (source : (D. LEE et H. SEUNG 1999))

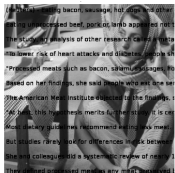
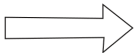
Dictionary/Feature learning

$$X \approx G \cdot F$$

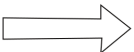
- Once G is learnt → many applications : denoising, inpainting, compression, etc



denoising

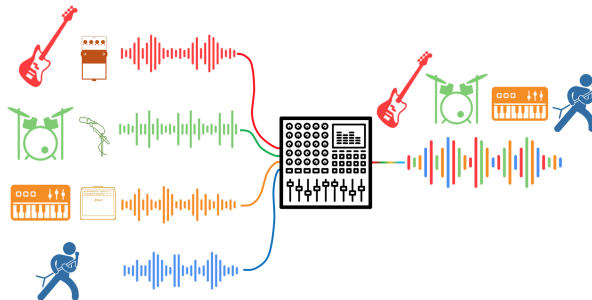


inpainting



Source : (GRIBONVAL 2014)

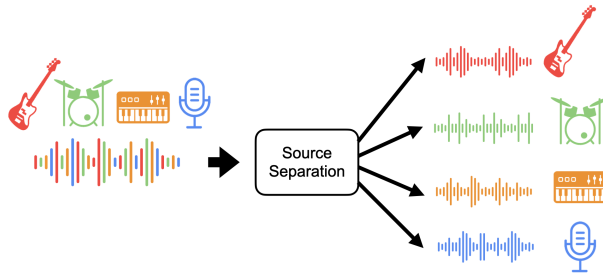
Blind source separation



Several source signals are mixed (Source : (MANILOW, SEETHARMAN et SALAMON 2020))

- Several unknown “source signals” f_j
- We only observe “mixture(s)” of them x_i , e.g., $x_i(t) \approx \sum_j g_{ij} \cdot f_j(t)$

Blind source separation



Source separation aims to estimate them from the mixture(s) (Source : (MANILOW, SEETHARMAN et SALAMON 2020))

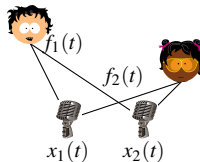
- Several unknown “source signals” f_j
- We only observe “mixture(s)” of them x_i , e.g., $x_i(t) \approx \sum_j g_{ij} \cdot f_j(t)$

⇒ BSS aims to estimate the f_j s from the x_i s

Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories :

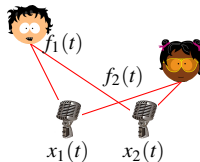
- 1 Linear instantaneous mixtures : $x_i(t) = \sum_{j=1}^k g_{ij}f_j(t)$



Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories :

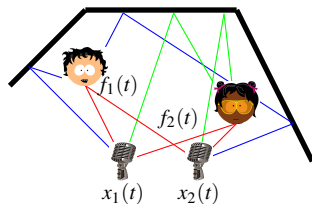
- 1 Linear instantaneous mixtures : $x_i(t) = \sum_{j=1}^k g_{ij} f_j(t)$
- 2 Anechoic mixtures : $x_i(t) = \sum_{j=1}^k g_{ij} f_j(t - t_{ij})$



Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories :

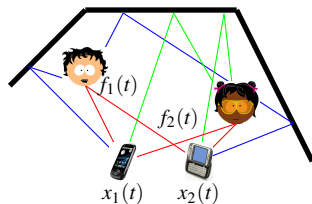
- 1 Linear instantaneous mixtures : $x_i(t) = \sum_{j=1}^k g_{ij} f_j(t)$
- 2 Anechoic mixtures : $x_i(t) = \sum_{j=1}^k g_{ij} f_j(t - t_{ij})$
- 3 Convolutive mixtures : $x_i(t) = \sum_{j=1}^k \sum_{l=-\infty}^{+\infty} f_{ijl} s_j(t - t_{ijl}) = \sum_{j=1}^N g_{ij}(t) * f_j(t)$



Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories :

- ① Linear instantaneous mixtures : $x_i(t) = \sum_{j=1}^k g_{ij} f_j(t)$
- ② Anechoic mixtures : $x_i(t) = \sum_{j=1}^k g_{ij} f_j(t - t_{ij})$
- ③ Convulsive mixtures : $x_i(t) = \sum_{j=1}^k \sum_{l=-\infty}^{+\infty} f_{ijl} s_j(t - t_{ijl}) = \sum_{j=1}^N g_{ij}(t) * f_j(t)$



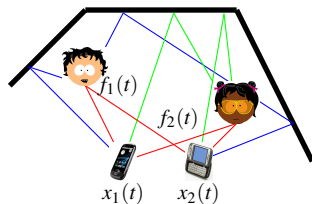
But also interest :

- ④ Post-nonlinear (convolutive) mixtures : $x_i(t) = \mathcal{H}_i \left(\sum_{j=1}^k g_{ij}(t) * f_j(t) \right)$

Classes of mixtures

Most of the approaches process linear mixtures which are divided in three categories :

- ① Linear instantaneous mixtures : $x_i(t) = \sum_{j=1}^k g_{ij} f_j(t)$
- ② Anechoic mixtures : $x_i(t) = \sum_{j=1}^k g_{ij} f_j(t - t_{ij})$
- ③ Convulsive mixtures : $x_i(t) = \sum_{j=1}^k \sum_{l=-\infty}^{+\infty} f_{ijl} s_j(t - t_{ijl}) = \sum_{j=1}^N g_{ij}(t) * f_j(t)$



But also interest :

- ④ Post-nonlinear (convolutive) mixtures : $x_i(t) = \mathcal{H}_i \left(\sum_{j=1}^k g_{ij}(t) * f_j(t) \right)$
- ⑤ Other classes of nonlinear mixtures (linear quadratic, polynomial, etc)

A very fast overview of source separation approaches

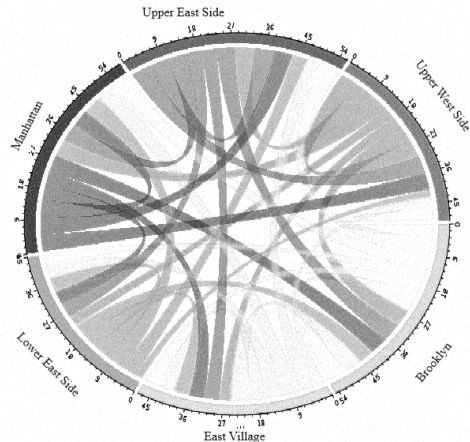
- 1 Sources f_j are statistically independent ➡ Independent Component/Vector Analysis (ICA, IVA) (COMON et JUTTEN 2010)
- 2 Sources f_j are sparse ➡ Sparse Component Analysis (SCA) (DEVILLE 2014)
- 3 Mixtures x_i , sources f_j and mixing parameters g_{ij} are non-negative ➡ NMF (GILLIS 2020) / NTF
- 4 Large amount of training data (i.e., data “quite similar” to the sources) are available ➡ Deep learning

Applications

- Sources can be of any kind (audio, image, biomedical, hyperspectral, [multimodal](#), etc)
- More information on less-classical BSS applications later

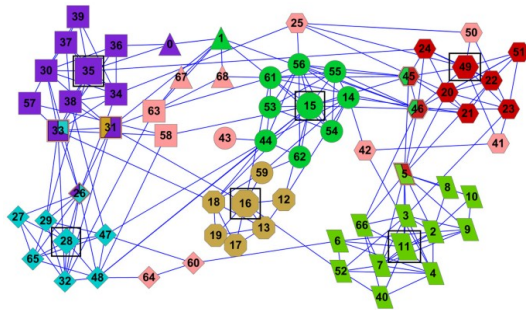
Graph analysis and applications

- Many data seen as graphs
 - bike sharing analysis, NYC cab traffic analysis
 - Social networks, Sociology
 - etc



Graph analysis and applications

- Many data seen as graphs
 - bike sharing analysis, NYC cab traffic analysis
 - Social networks, Sociology
 - etc
- One classical problem : community detection
 - Let the unweighted graph defined by $\{V, E\}$ where V and E are the sets of vertices and edges, resp.
 - Adjacency matrix A :
$$a_{ij} = \begin{cases} 1 & \text{if } V_i \text{ and } V_j \text{ are linked,} \\ 0 & \text{otherwise.} \end{cases}$$
 - Community detection using NMF :
$$A \approx G \cdot F \text{ (HE et al. 2021)}$$



(source : (CAO et al. 2013))

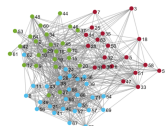
Graph analysis and applications

- Many data seen as graphs
 - bike sharing analysis, NYC cab traffic analysis
 - Social networks, Sociology
 - etc

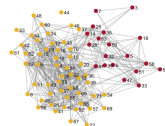
- One classical problem : community detection

- Let the unweighted graph defined by $\{V, E\}$ where V and E are the sets of vertices and edges, resp.
- Adjacency matrix A :
$$a_{ij} = \begin{cases} 1 & \text{if } V_i \text{ and } V_j \text{ are linked,} \\ 0 & \text{otherwise.} \end{cases}$$
- Community detection using NMF :
$$A \approx G \cdot F \text{ (HE et al. 2021)}$$

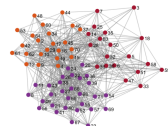
Going further : multiplex graph analysis (ORTIZ-BOUZA et AVIYENTE 2024)



(a) Advice



(b) Friendship

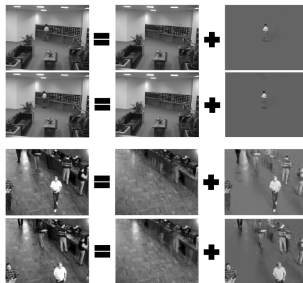


(c) Co-work



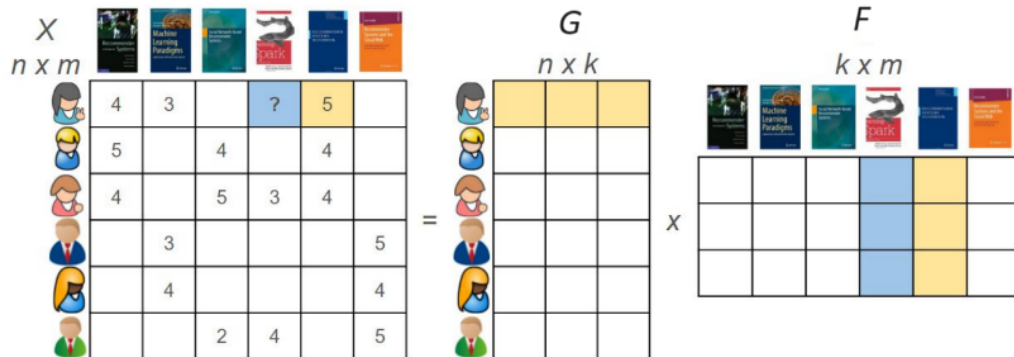
Video background subtraction

- In video surveillance:
 - Background is usually constant
 - Objects of interest are moving
- ⇒ Background subtraction $\Rightarrow X \approx G \cdot F + S$ aka **Robust PCA** or **low-rank plus sparse decomposition** (SOBRAL, BOUWMANS et ZAHZAH 2015; T. ZHOU et TAO 2011)



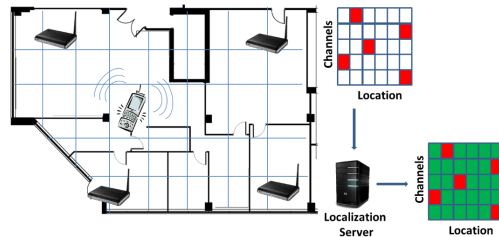
Source : (T. ZHOU et TAO 2011)

Collaborative filtering



Sensor Localization from Fingerprints

- Indoor localization is a hard problem (GNSS not very accurate)
- Sensor localization based on signal strength (e.g., RFID)
- Proposed technique :
 - 1 Get “fingerprints” from beacons and receivers with known positions
 - 2 Estimate signal strengths in non-sensed positions : $W \circ X \approx W \circ M$ where W is binary, X known values of signal strengths, M completed low-rank matrix of signal strengths
 - 3 Infer a mobile device position using the completed map of signal strengths



Source : (NIKITAKI, Grigorios TSAGKATAKIS et
TSAKALIDES 2015)

Summary

- 1 Foreword
- 2 Classical applications of low-rank models
- 3 In-situ sensor calibration**
 - Fixed sensor calibration
 - Mobile sensor calibration
- 4 Joint multispectral image restoration and unmixing
- 5 Low-rankness in the AI Era
- 6 Discussion

Context



- Air pollution remains an issue $\Rightarrow \approx 40000$ premature deaths per year in France
- Need to monitor air quality
- ! Local effects not sensed and hard to model with a sparsely distributed sensor network
- Tremendous development of miniaturized sensors
- Allow a much denser deployment than authoritative sensing stations
- 📦 Some local effects become observable
- ! But **sensor drift** is an issue

The “why” of in situ sensor calibration

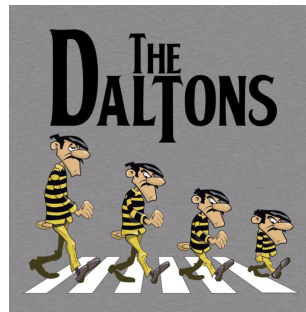


- Sensed phenomenon ⇨ voltage
- Voltage ⇨ phenomenon ?

The “why” of in situ sensor calibration



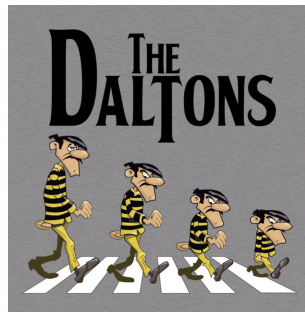
- Sensed phenomenon ⇨ voltage
- Voltage ⇨ phenomenon ?



The “why” of in situ sensor calibration



- Sensed phenomenon ⇨ voltage
- Voltage ⇨ phenomenon ?
 - In-lab calibration not possible
 - ⇨ data driven techniques (i.e., *in situ* sensor calibration)

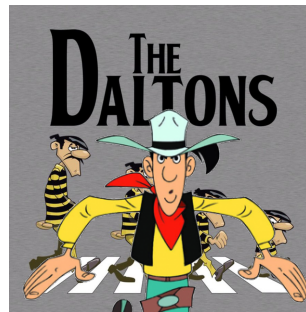




The “why” of in situ sensor calibration



- Sensed phenomenon ⇨ voltage
- Voltage ⇨ phenomenon ?
 - In-lab calibration not possible
 - ⇨ data driven techniques (i.e., *in situ* sensor calibration)
 - Some reference data might be available



The "how" of in situ sensor calibration (1)

- Many proposed methods
- ↳ Sensor network architecture
- In this talk :
 - ① Fixed sensors (BALZANO et NOWAK 2007)
 - ② Mobile sensors (DORFFER et al. 2018)

Fixed sensor calibration (1) – Problem Statement and Assumptions

- Network composed of N fixed and synchronized sensors observed at Times $t = t_1, \dots, t_d$.

Fixed sensor calibration (1) – Problem Statement and Assumptions

- Network composed of N fixed and synchronized sensors observed at Times $t = t_1, \dots, t_d$.
- Calibrated and uncalibrated sensor readings denoted $x_n(t)$ and $y_n(t)$

Fixed sensor calibration (1) – Problem Statement and Assumptions

- Network composed of N fixed and synchronized sensors observed at Times $t = t_1, \dots, t_d$.
- Calibrated and uncalibrated sensor readings denoted $x_n(t)$ and $y_n(t)$
- **Affine** response model :

$$x_n(t) \approx \alpha_n y_n(t) + \beta_n$$

where α_n (resp. β_n) is the n-th sensor gain (resp. offset) correction

Fixed sensor calibration (1) – Problem Statement and Assumptions

- Network composed of N fixed and synchronized sensors observed at Times $t = t_1, \dots, t_d$.
- Calibrated and uncalibrated sensor readings denoted $x_n(t)$ and $y_n(t)$
- **Affine** response model :

$$x_n(t) \approx \alpha_n y_n(t) + \beta_n$$

where α_n (resp. β_n) is the n-th sensor gain (resp. offset) correction

- **No sensor gain is zero**

$$y_n(t) \approx \frac{x_n(t) - \beta_n}{\alpha_n}$$

Fixed sensor calibration (1) – Problem Statement and Assumptions

- Network composed of N fixed and synchronized sensors observed at Times $t = t_1, \dots, t_d$.
- Calibrated and uncalibrated sensor readings denoted $x_n(t)$ and $y_n(t)$
- **Affine** response model :

$$x_n(t) \approx \alpha_n y_n(t) + \beta_n$$

where α_n (resp. β_n) is the n-th sensor gain (resp. offset) correction

- **No sensor gain is zero**

$$y_n(t) \approx \frac{x_n(t) - \beta_n}{\alpha_n}$$

- Sensors are **pre-calibrated** before deployment
 - ◊ First sensor readings of $y_i(t)$ thus correspond to $x_i(t)$

Fixed sensor calibration (1) – Problem Statement and Assumptions

- Network composed of N fixed and synchronized sensors observed at Times $t = t_1, \dots, t_d$.
- Calibrated and uncalibrated sensor readings denoted $x_n(t)$ and $y_n(t)$
- **Affine** response model :

$$x_n(t) \approx \alpha_n y_n(t) + \beta_n$$

where α_n (resp. β_n) is the n-th sensor gain (resp. offset) correction

- **No sensor gain is zero**

$$y_n(t) \approx \frac{x_n(t) - \beta_n}{\alpha_n}$$

- Sensors are **pre-calibrated** before deployment
 - ◊ First sensor readings of $y_i(t)$ thus correspond to $x_i(t)$
- The network is dense enough to oversampling the signal space
 - ◊ Rank k of the observed signal is known with $k \ll N$

Fixed sensor calibration (1) – Problem Statement and Assumptions

- Network composed of N fixed and synchronized sensors observed at Times $t = t_1, \dots, t_d$.
- Calibrated and uncalibrated sensor readings denoted $x_n(t)$ and $y_n(t)$
- **Affine** response model :

$$x_n(t) \approx \alpha_n y_n(t) + \beta_n$$

where α_n (resp. β_n) is the n -th sensor gain (resp. offset) correction

- **No sensor gain is zero**

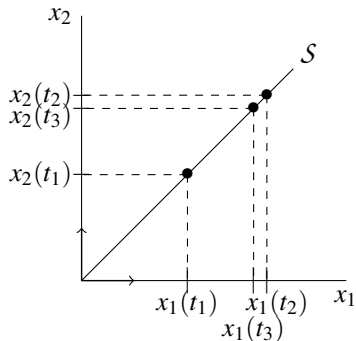
$$y_n(t) \approx \frac{x_n(t) - \beta_n}{\alpha_n}$$

- Sensors are **pre-calibrated** before deployment
 - ◊ First sensor readings of $y_i(t)$ thus correspond to $x_i(t)$
- The network is dense enough to oversampling the signal space
 - ◊ Rank k of the observed signal is known with $k \ll N$
- Observed signal subspace $\mathcal{S}$ is **known**, e.g.,
 - we can learn the subspace in which lie the "calibrated" data from the first readings,
 - the subspace can be provided by experts in case of no pre-calibration.

Fixed sensor calibration (2) – Example

Example with known rank-1 subspace $\mathcal{S}$

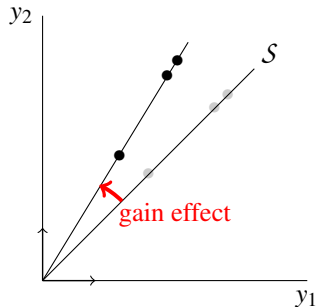
$$\begin{bmatrix} x_1(t) = \alpha_1 \cdot y_1(t) + \beta_1 \\ x_2(t) = \alpha_2 \cdot y_2(t) + \beta_2 \end{bmatrix} \in \mathcal{S}, \forall t$$



Fixed sensor calibration (2) – Example

Example with known rank-1 subspace $\mathcal{S}$

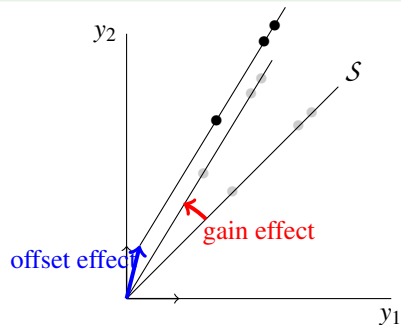
$$\begin{bmatrix} x_1(t) = \alpha_1 \cdot y_1(t) + \beta_1 \\ x_2(t) = \alpha_2 \cdot y_2(t) + \beta_2 \end{bmatrix} \in \mathcal{S}, \forall t$$



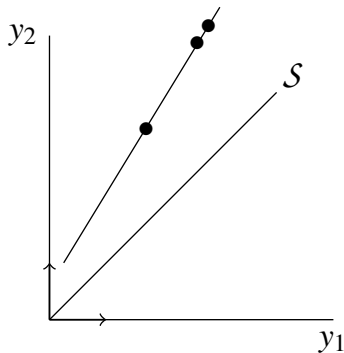
Fixed sensor calibration (2) – Example

Example with known rank-1 subspace $\mathcal{S}$

$$\begin{bmatrix} x_1(t) = \alpha_1 \cdot y_1(t) + \beta_1 \\ x_2(t) = \alpha_2 \cdot y_2(t) + \beta_2 \end{bmatrix} \in \mathcal{S}, \forall t$$

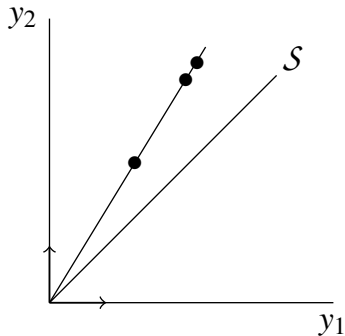


Fixed sensor calibration (3) – General strategy



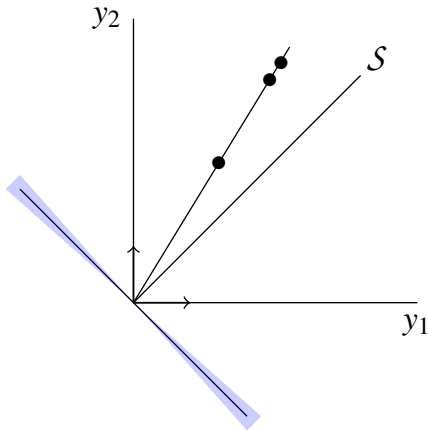
- 1 Either assume the offsets to be null or remove the offset contributions (by centering the signals)

Fixed sensor calibration (3) – General strategy



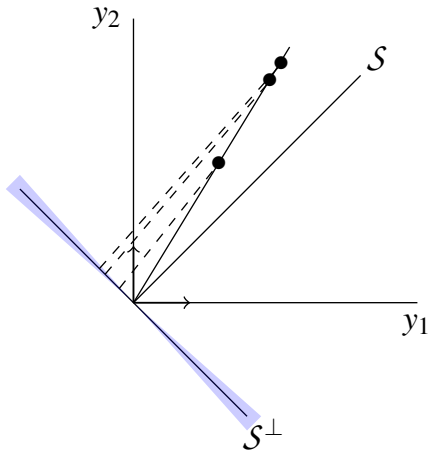
- 1 Either assume the offsets to be null or remove the offset contributions (by centering the signals)

Fixed sensor calibration (3) – General strategy



- 1 Either assume the offsets to be null or remove the offset contributions (by centering the signals)
- 2 Projecting data onto $S^\perp$

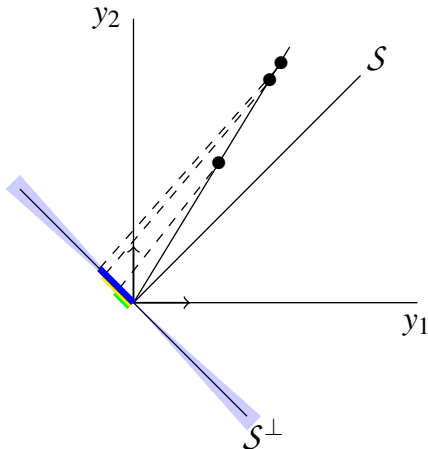
Fixed sensor calibration (3) – General strategy



- 1 Either assume the offsets to be null or remove the offset contributions (by centering the signals)
- 2 Projecting data onto $S^\perp$

Fixed sensor calibration (3) – General strategy

- 1 Either assume the offsets to be null or remove the offset contributions (by centering the signals)
- 2 Projecting data onto $\mathcal{S}^\perp$
- 3 Estimating sensor gains by **nullspace projection**, i.e.

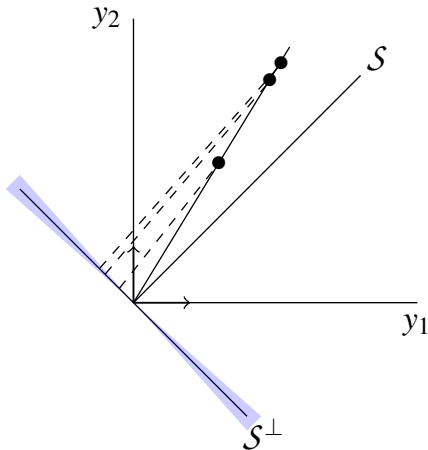


$$\underbrace{\begin{bmatrix} \mathcal{P} \cdot \mathcal{Y}_1 \\ \vdots \\ \mathcal{P} \cdot \mathcal{Y}_d \end{bmatrix}}_c \cdot \underline{\alpha} \approx 0,$$

where $\mathcal{P}$ is the projection operator onto $\mathcal{S}^\perp$ and

$$\mathcal{Y}_k = \begin{bmatrix} y_1(t_k) - \overline{y_1} & & 0 \\ 0 & \ddots & 0 \\ & & y_N(t_k) - \overline{y_N} \end{bmatrix}$$

Fixed sensor calibration (3) – General strategy



- 1 Either assume the offsets to be null or remove the offset contributions (by centering the signals)
- 2 Projecting data onto $\mathcal{S}^\perp$
- 3 Estimating sensor gains by **nullspace projection**, i.e.

$$\underbrace{\begin{bmatrix} \mathcal{P} \cdot \mathcal{Y}_1 \\ \vdots \\ \mathcal{P} \cdot \mathcal{Y}_d \end{bmatrix}}_c \cdot \underline{\alpha} \approx 0,$$

where $\mathcal{P}$ is the projection operator onto $\mathcal{S}^\perp$ and

$$\mathcal{Y}_k = \begin{bmatrix} y_1(t_k) - \overline{y_1} & & 0 \\ 0 & \ddots & 0 \\ & & y_N(t_k) - \overline{y_N} \end{bmatrix}$$

- 4 Estimating sensor offsets (e.g., by least squares if true data are zero-mean)

Fixed sensor calibration (4)

Are we sure it works ?

Yes! Proof of convergence if the number d of "snapshots" satisfies

$$d > \left\lceil \frac{N-1}{N-k} \right\rceil + 1$$

But a scale indeterminacy remains. Possibility to solve it by, e.g., assuming that $\alpha_1 = 1$

Fixed sensor calibration (4)

How to solve $C \cdot \underline{\alpha} = 0$? (BALZANO et NOWAK 2007; DORFFER et al. 2017)

① Least-squares :

- Remove α_1 from $\underline{\alpha}$ and obtain $\underline{\alpha}'$
- Remove the first column $\underline{c}_1$ from C and obtain C'
- The problem to solve reads $C' \cdot \underline{\alpha}' = -\underline{c}_1$
- $\underline{\alpha}' = -C'^{-\dagger} \cdot \underline{c}_1$ where $^{-\dagger}$ denotes the pseudo-inverse (`pinv` in Matlab)

Fixed sensor calibration (4)

How to solve $C \cdot \underline{\alpha} = 0$? (BALZANO et NOWAK 2007; DORFFER et al. 2017)

1 Least-squares :

- Remove α_1 from $\underline{\alpha}$ and obtain $\underline{\alpha}'$
- Remove the first column $\underline{c}_1$ from C and obtain C'
- The problem to solve reads $C' \cdot \underline{\alpha}' = -\underline{c}_1$
- $\underline{\alpha}' = -C'^{-\dagger} \cdot \underline{c}_1$ where $^{-\dagger}$ denotes the pseudo-inverse (`pinv` in Matlab)

2 SVD :

- Compute the economy-size SVD of C (`svd(C, 0)` in Matlab)
- The last singular vector v_{end} of V is proportional to $\underline{\alpha}$
- $\underline{\alpha}$ can be derived by dividing v_{end} by its first value $v_{\text{end}}(1)$.

Fixed sensor calibration (4)

How to solve $C \cdot \underline{\alpha} = 0$? (BALZANO et NOWAK 2007; DORFFER et al. 2017)

1 Least-squares :

- Remove α_1 from $\underline{\alpha}$ and obtain $\underline{\alpha}'$
- Remove the first column $\underline{c}_1$ from C and obtain C'
- The problem to solve reads $C' \cdot \underline{\alpha}' = -\underline{c}_1$
- $\underline{\alpha}' = -C'^{-\dagger} \cdot \underline{c}_1$ where $^{-\dagger}$ denotes the pseudo-inverse (`pinv` in Matlab)

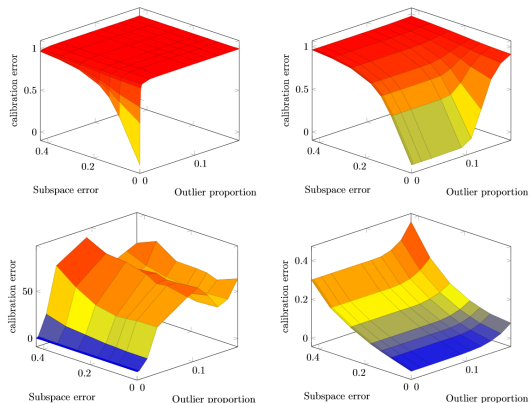
2 SVD :

- Compute the economy-size SVD of C (`svd(C, 0)` in Matlab)
- The last singular vector v_{end} of V is proportional to $\underline{\alpha}$
- $\underline{\alpha}$ can be derived by dividing v_{end} by its first value $v_{\text{end}}(1)$.

3 Outlier-robust extensions are immediate (DORFFER et al. 2017)

Experiments

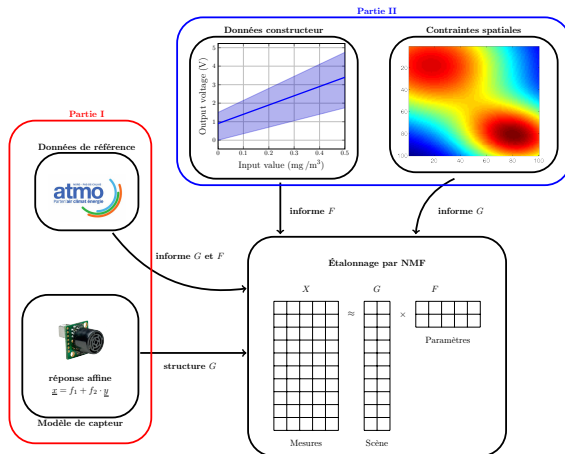
- Simulations where we investigate the error of estimation of the low-rank space (and its orthogonal) and the proportion of outliers
- Tested methods from (BALZANO et NOWAK 2007) (left) and (DORFFER et al. 2017) (right)



Motivation

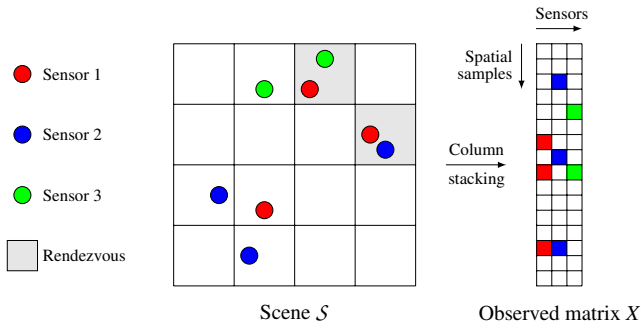
- When sensors are moving, most approaches specifically developed for a considered sensor network architecture (MAAG, Z. ZHOU et THIELE 2018)
 - By combining assumptions of several methods, we proposed in (DORFFER et al. 2018) a versatile technique
- ➡ Mobile sensor calibration $\iff$ collaborative filtering

Motivation



Definitions

- A **rendezvous** is a temporal and spatial vicinity between two sensors (SAUKH, HASENFRATZ, WALSER et al. 2014).
- A **scene** $\mathcal{S}$ is a discretized area observed during a time interval $[t, t + \Delta t)$. A spatial pixel has a size lower than Δd , where Δt and Δd define the vicinity of the rendezvous (DORFFER et al. 2018).



Assumptions & Problem Formalism (1)

- Sensor response (calibration function $\mathcal{F}(\cdot)$ of Sensor j)

$$\underbrace{x(i,j)}_{\text{sensor-output voltage}} \simeq \underbrace{g_1(i)}_{\text{physical phenomenon}} \cdot \underbrace{f_{1,j}}_{\text{unknown gain and offset}} + f_{0,j}$$

- ⇒ Matrix form (if **each** of the m sensor senses **all** the scene)

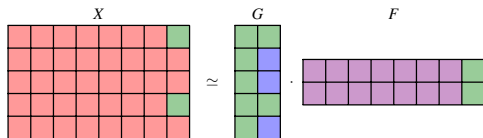
$$\underbrace{\begin{bmatrix} x(1,1) & \cdots & x(1,m) \\ \vdots & & \vdots \\ x(n,1) & \cdots & x(n,m) \end{bmatrix}}_X \simeq \underbrace{\begin{bmatrix} 1 & g_1(1) \\ \vdots & \vdots \\ 1 & g_1(n) \end{bmatrix}}_G \cdot \underbrace{\begin{bmatrix} f_{0,1} & f_{0,2} & \cdots & f_{0,m} \\ f_{1,1} & f_{1,2} & \cdots & f_{1,m} \end{bmatrix}}_F$$

- In practice, irregular sampling : $W \circ X \simeq W \circ (G \cdot F)$ with w_{ij} not binary (ρ_j is a weight coefficient associated with Sensor j)

Assumptions & Problem Formalism (2)

- X , G , and F are nonnegative (air quality application)
- A known reference
- $\forall i = 1, \dots, n, \quad x(i, m) = g_1(i)$ (i.e., $f_{1,m} = 1, f_{0,m} = 0$)
- Blind calibration revisited as an informed nonnegative matrix factorization problem

$$W \circ \underbrace{\begin{bmatrix} x(1,1) & \cdots & x(1,m-1) & g_1(1) \\ x(2,1) & \cdots & x(2,m-1) & g_1(2) \\ \vdots & & \vdots & \vdots \\ x(n,1) & \cdots & x(n,m-1) & g_1(n) \end{bmatrix}}_X \simeq W \circ \left(\underbrace{\begin{bmatrix} 1 & g_1(1) \\ 1 & g_1(2) \\ \vdots & \vdots \\ 1 & g_1(n) \end{bmatrix}}_G \cdot \underbrace{\begin{bmatrix} f_{0,1} & f_{0,2} & \cdots & f_{0,m-1} & 0 \\ d_{1,1} & f_{1,2} & \cdots & f_{1,m-1} & 1 \end{bmatrix}}_F \right)$$



$$G = \phi_G + \Delta_G$$

$$F = \phi_F + \Delta_F$$

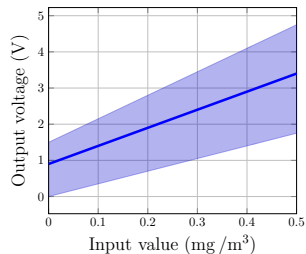
Calibration $\iff$ Estimating F

Alternative Models and Extra-Assumptions

- For some sensors, calibration model is polynomial (DORFFER et al. 2016), multilinear (VU THANH et al. 2021 ; YAHAYA 2021), etc ➡ **Structure on G**

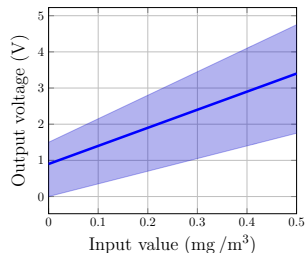
Alternative Models and Extra-Assumptions

- For some sensors, calibration model is polynomial (DORFFER et al. 2016), multilinear (VU THANH et al. 2021 ; YAHAYA 2021), etc $\Rightarrow$ Structure on G
- For some sensors, average behavior of sensors is known $\Rightarrow$ Penalization on the rows of F



Alternative Models and Extra-Assumptions

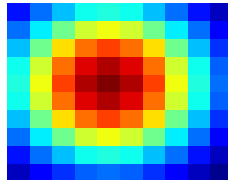
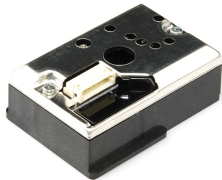
- For some sensors, calibration model is polynomial (DORFFER et al. 2016), multilinear (VU THANH et al. 2021 ; YAHAYA 2021), etc $\Rightarrow$ Structure on G
- For some sensors, average behavior of sensors is known $\Rightarrow$ Penalization on the rows of F
- Possibility to learn a dictionary of spatial distribution of sensed phenomena $\Rightarrow$ **Free row(s) of G are sparse wrt this dictionary**



Solving such problems

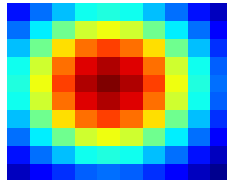
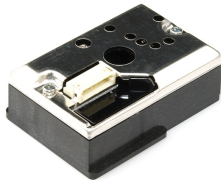
- Naive approach :
 - 1 Applying low-rank matrix completion on X
 - 2 Deriving G
 - 3 Estimating F (non-negative least-squares)
- Multiplicative updates for WNMF (using MM strategy) (DORFFER et al. 2018)
- Nesterov updates within an EM strategy for (semi-)WNMF (VU THANH et al. 2021 ; YAHAYA 2021)

Performance of some of these methods

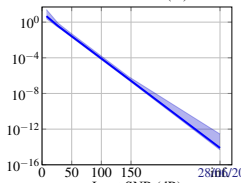
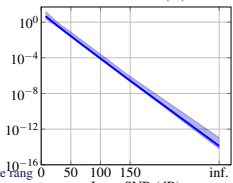
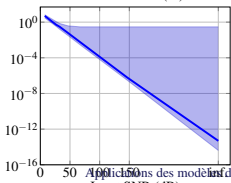
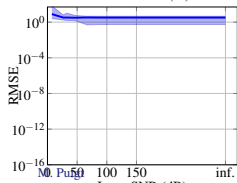
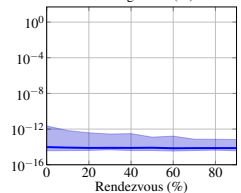
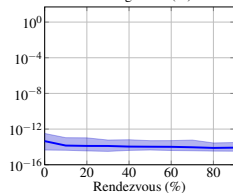
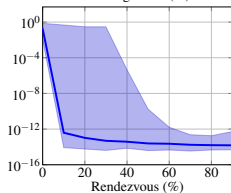
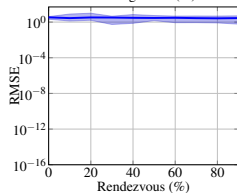
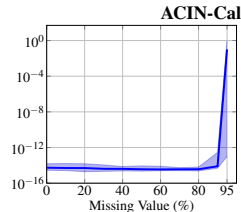
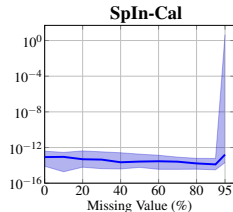
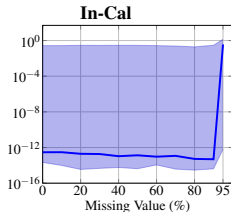
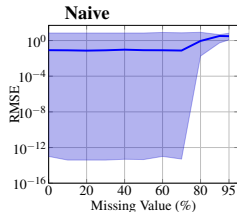


- Crowdsensing-like particulate matter sensing
- Compact Optical Dust Sensor (Sharp GP2Y1010AU0F)
- Scene = 10×10 discretized area ($n = 100$) observed by $m = 26$ sensors
- The values in y range between 0 and 0.5 mg/m^3 (no sensor saturation)
- Dictionary with 62 atoms
- y is 2-sparse

Performance of some of these methods

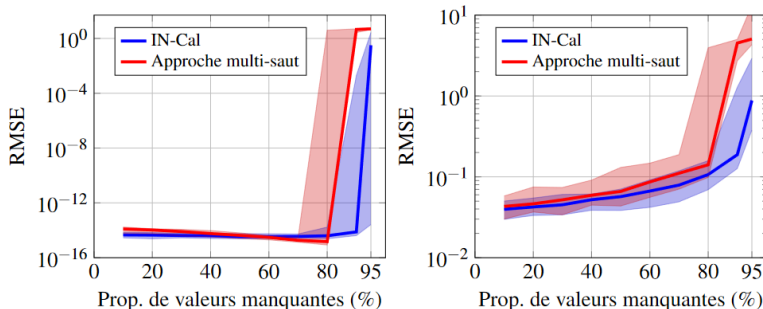


- We aim to test the influence of :
 - 1 the proportion of uncalibrated sensors to have **one unique rendezvous** with calibrated ones (default 30%),
 - 2 the proportion of missing entries in X (default 90%)
 - 3 the input SNR (default noiseless)
- For each test condition, we launch 25 random runs with **5e5** iterations
- Perf. criterion : RMSE over rows of F



Comparison with state-of-the-art method (SAUKH, HASENFRATZ et THIELE 2015)

- New simulations with more rendezvous to see how the approaches perform
 - Randomly remove a proportion of missing entries in X (much more calibrated data than in the previous tests)
 - Perf criterion : RMSE, success rate (RMSE below $1e-10$) in noiseless and noisy cases ($\text{SNR} \approx 30 \text{ dB}$)



Summary

- 1 Foreword
- 2 Classical applications of low-rank models
- 3 In-situ sensor calibration
- 4 Joint multispectral image restoration and unmixing**
 - Introduction, Motivation and Problem Statement
 - Proposed methods and Some Results
- 5 Low-rankness in the AI Era
- 6 Discussion

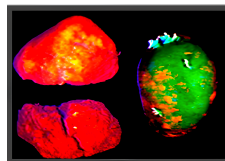
Evolution of Machine Vision



Monochromatic Image



Color Image



Hyperspectral Image



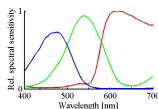
Monochromatic Camera

M. Puigt



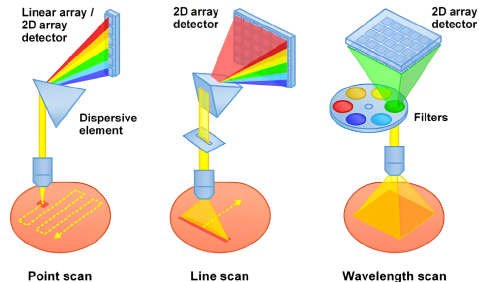
RGB Camera

Applications des modèles de faible rang



Hyperspectral Camera

Hyperspectral Cameras



Hyperspectral Cameras Cons

- Current spectral camera systems are **slow**, usually taking seconds to minutes to scan an object.
- **Not portable** to be placed on drones or unmanned aerial vehicles (UAVs).
- **Expensive !**

Compact Hyperspectral Cameras

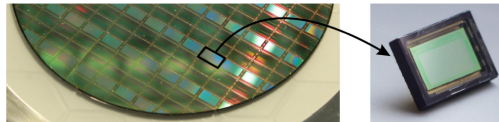
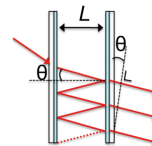
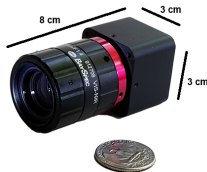


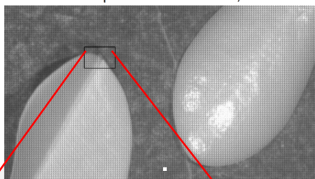
FIGURE 1 – Wafer including CMOS image sensors with integrated filter mosaics (left) and a packaged sensor (right). (Source (GEELEN, TACK et LAMBRECHTS 2014))



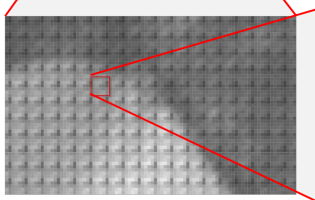
Fabry Perot Interferometer

Snapshot Spectral Cameras

Snapshot mosaic 5X5-NIR, 675-975nm

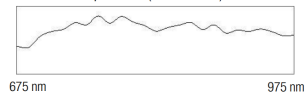


Single 5X5 pattern, wavelength peaks [nm]



900	909	892	882	683
809	821	797	784	693
759	772	746	732	708
943	949	935	927	975
861	873	852	840	955

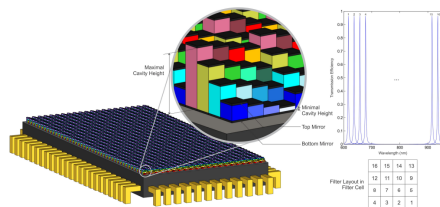
Spectrum (Raw-values)



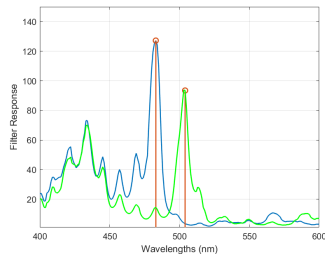
Snapshot Spectral Camera

- Theoretically, the SSI cameras associate each spatial pixel with a specific spectral band.
- In reality, the value at each pixel is a **filtered** version of the materials that exist in the pixel.

$$y_i(\lambda_i) = \sum_{j=1}^k h_i(\lambda_j) \cdot x_i(\lambda_j) + \omega_i, \quad (1)$$

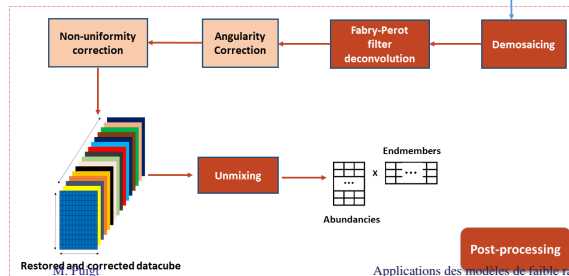
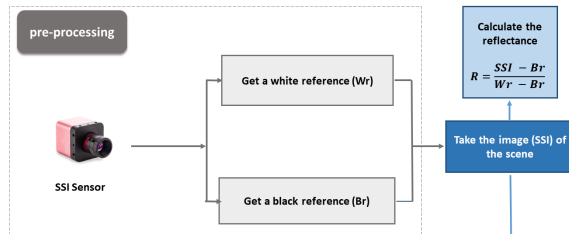


Snapshot mosaic filter (Source (GEELLEN, TACK et LAMBRECHTS 2014))



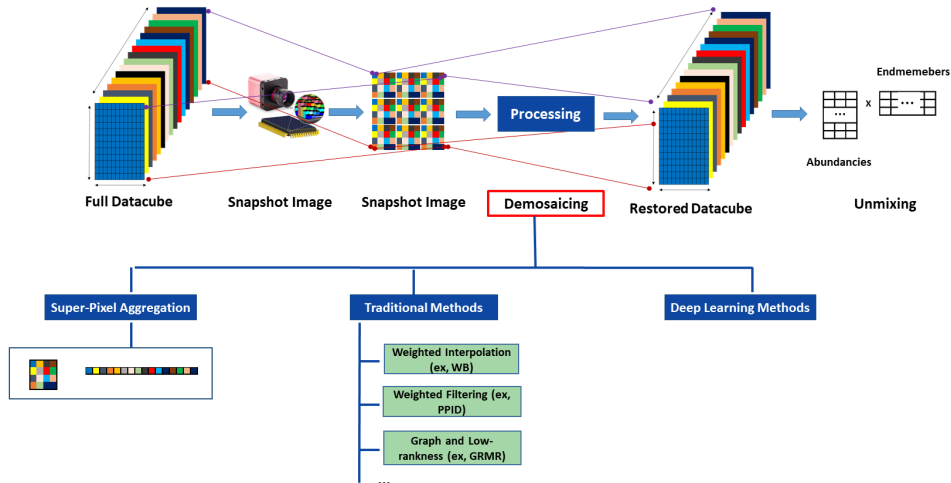
Ideal (in red) and real (in blue and green) spectral response of two Fabry-Perot filters of the 4×4 IMEC SSI camera.

Processing the SSI Image

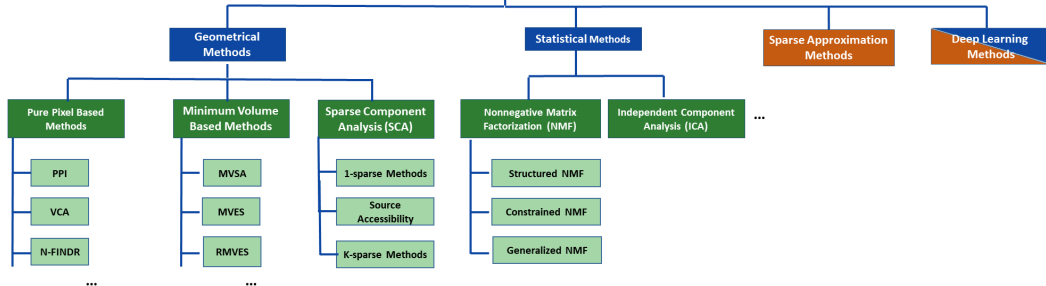
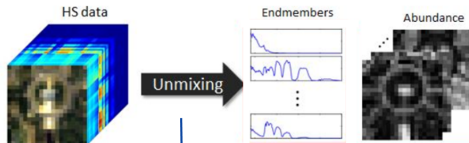


Assuming a Linear Mixture Model (LMM), deconvolution can be performed either before or after unmixing.

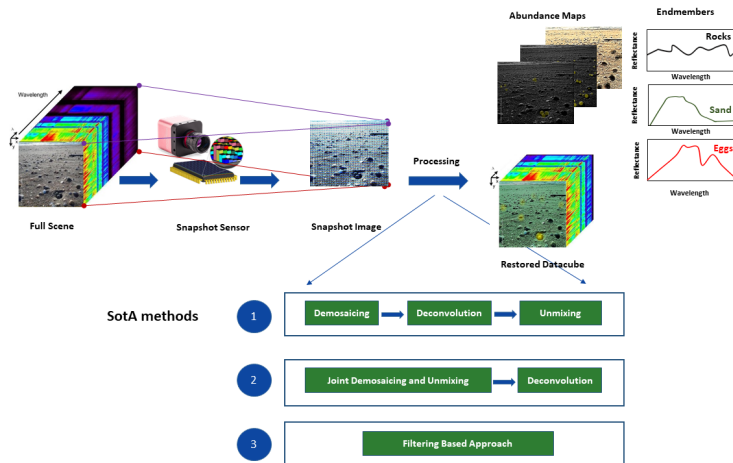
Demosaicing



Unmixing



Processing the SSI Image (ABBAS et al. 2024a,b)



Why Joint Demosaicing and Unmixing ?

- Low rank matrix completion has been used for demosaicing in (G. TSAGKATAKIS et al. 2019)

Why Joint Demosaicing and Unmixing ?

- Low rank matrix completion has been used for demosaicing in (G. TSAGKATAKIS et al. 2019)
- Performing **classification after demosaicing** provides a poor classification performance

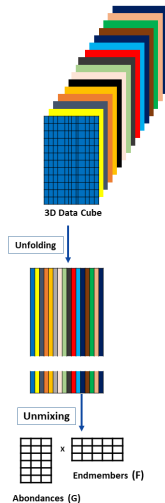
Why Joint Demosaicing and Unmixing ?

- Low rank matrix completion has been used for demosaicing in (G. TSAGKATAKIS et al. 2019)
- Performing **classification after demosaicing** provides a poor classification performance
- For another application, **joint** low-rank matrix completion and factorization **far more accurate** than a two-stage approach (DORFFER et al. 2018).

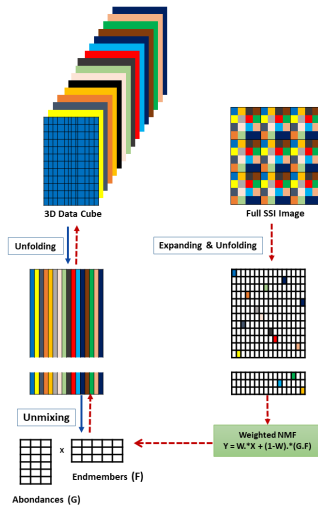
Why Joint Demosaicing and Unmixing ?

- Low rank matrix completion has been used for demosaicing in (G. TSAGKATAKIS et al. 2019)
- Performing **classification after demosaicing** provides a poor classification performance
- For another application, **joint** low-rank matrix completion and factorization **far more accurate** than a two-stage approach (DORFFER et al. 2018).

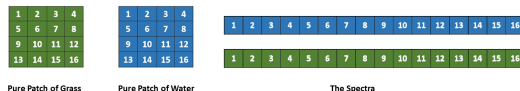
Naive approach for Joint Demosaicing and Unmixing



Naive approach for Joint Demosaicing and Unmixing



Assumptions required for the proposed methods (1/2)



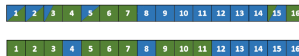
Assumption 1 (Pure Patch Assumption)

For each endmember, there exists at least one sensor “patch” where only this endmember is present.

Assumptions required for the proposed methods (1/2)



Grass and water abundances vary



Wavelength-dependent mixture of endmembers

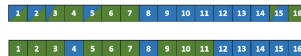
Assumption 2

In the patches where several endmembers are present, their abundances significantly vary over each patch.

Assumptions required for the proposed methods (1/2)



Grass and water abundances vary



Wavelength-dependent mixture of endmembers



Grass and Water abundances don't vary



Wavelength-independent mixture of endmembers

“Assumption” 3

In the patches where several endmembers are present, their abundances may or may not vary over each patch.

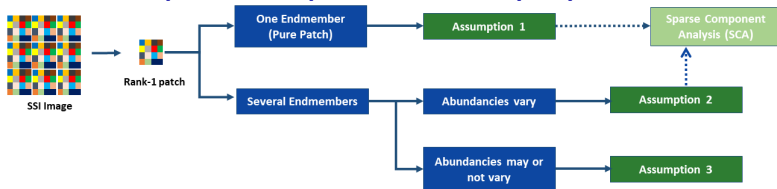
Assumptions required for the proposed methods (1/2)

Assumption 4

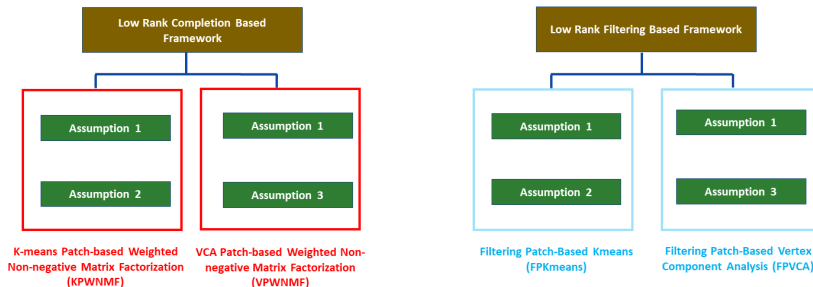
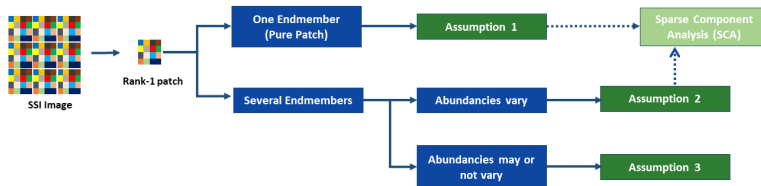
In each pixel, the mixture of the endmembers is expected to be linear.

$$x_i(\lambda) = \sum_l g_{il} f_l(\lambda)$$

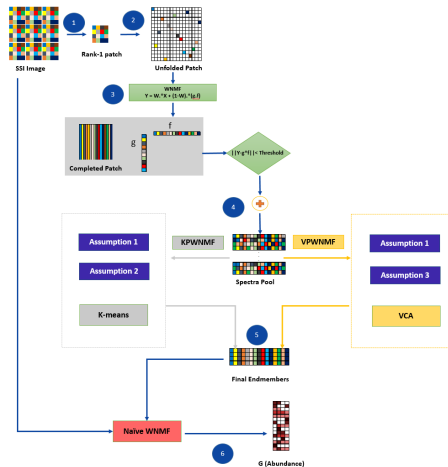
Assumptions required for the proposed methods (2/2)



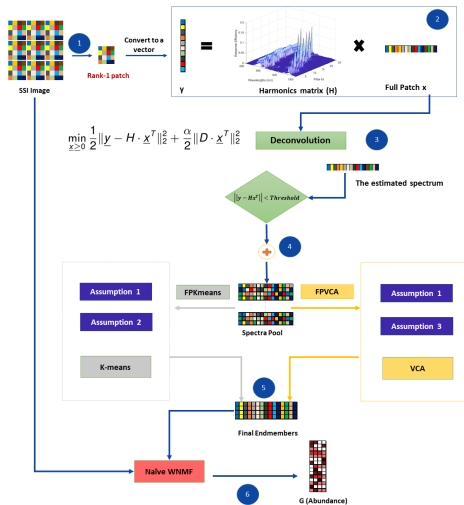
Assumptions required for the proposed methods (2/2)



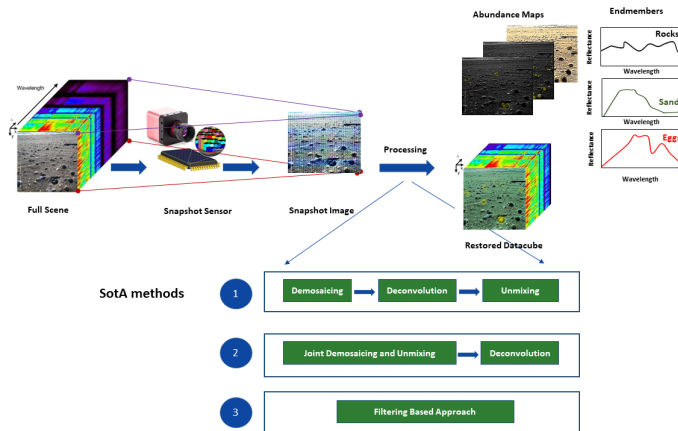
Low-rank Matrix Completion Framework



Filtering-based Framework



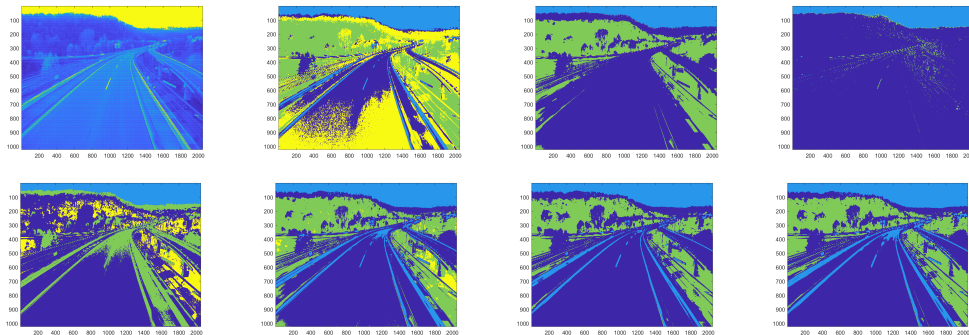
Experiments and Results (1/2)



Experiments Setup

- To assess the performance of our methods, we did extensive experiments on **synthetic** data and **real** images acquired by SSI camera.
- We compare with **3-stage** approaches involving separate demosaicing spectral correction and unmixing steps
 - 1 Demosaicing : SotA
 - 2 Spectral correction : IMEC correction
 - 3 Unmixing : VCA

Performance evaluation on real SSI images



Segmentation of a Hyko 2 database image for different unmixing methods. From left to right :
Top : Raw, PPID, GRMR, Naive WNMF.
Bottom : KPWNMF, VPWNMF, FPKmeans, FPVCA.

Experiments and Results (2/2)

Main Findings

- In ideal scenarios with varying noise levels, **KPWNMF** and **VPWNMF**, which belong to the **Low-rank framework**, exhibited the highest performance.
- When real filters were introduced, **FPKmeans** and **FPVCA** (**Filtering based framework**) demonstrated superior performance.
- Although the performance of both KPWNMF and VPWNMF methods declined compared to ideal situations, they still outperformed 3-stage approaches.

Take-home Messages

- Employing a joint unmixing and demosaicing approach within the low-rank completion framework proves superior to 3-stage approaches in both ideal and real-world scenarios.
- **Obviating** the spectral correction step and starting the deconvolution process directly from the raw SSI images improved unmixing and demosaicing results while simplifying the overall processing pipeline.

Summary

- 1 Foreword
- 2 Classical applications of low-rank models
- 3 In-situ sensor calibration
- 4 Joint multispectral image restoration and unmixing
- 5 Low-rankness in the AI Era**
- 6 Discussion

Random projections and compressed learning

- Big data tend to be **low-rank**
- ⇒ Draw a smaller **Sketch** of big data while keeping (almost) similar information
- ⇒ Random projections aka RandNLA (DRINEAS et MAHONEY 2016)

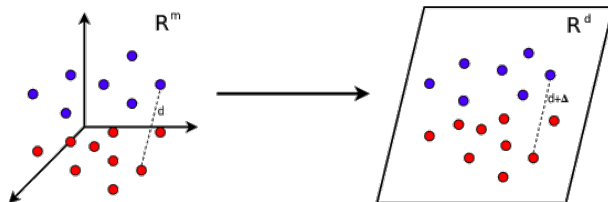
Random projections and compressed learning

Johnson-Lindenstrauss Lemma

Let $\epsilon \in]0, 1[$ and a set of n points $\{y_1 \cdots y_n\}$ in $\mathbb{R}^m$.

There exists a linear mapping $f : \mathbb{R}^m \mapsto \mathbb{R}^d$ with $d > 8(\log(n)/\epsilon^2)$, such that $\forall 1 \leq i \leq j \leq n, \forall u, v \in X$,

$$(1 - \epsilon)\|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \epsilon)\|u - v\|^2$$



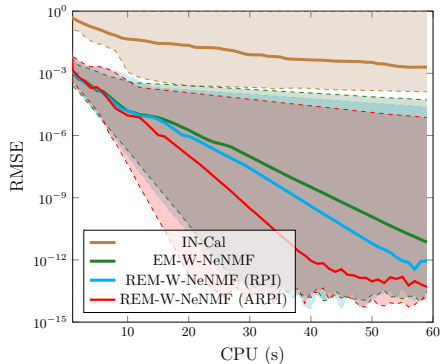
Random projections and compressed learning

- Big data tend to be **low-rank**
- ⇒ Draw a smaller **Sketch** of big data while keeping (almost) similar information
- ⇒ Random projections aka RandNLA (DRINEAS et MAHONEY 2016)
- Strategy proposed with (W)NMF (TEPPER et SAPIRO 2016; YAHAYA 2021), CP decomposition (ERICHSON et al. 2020), deep learning (WÓJCIK et KURDZIEL 2019)...

Random projections and compressed learning

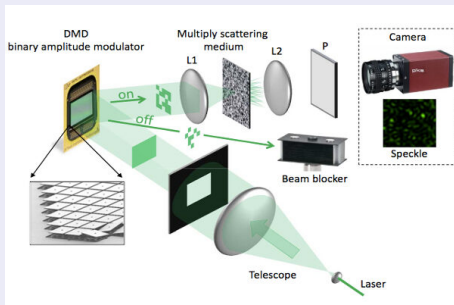
Illustration : Randomized WNMF (YAHAYA 2021)

- $W \circ X \approx W \circ (G \cdot F)$
- Repeat
 - 1 E-step : $X^{\text{comp}} = W \circ X + (\mathbb{1} - W) \circ (G \cdot F)$
Design projection matrices L and R
 - 2 M-step : Repeat
 - 1 Update F : $\underbrace{L \cdot X^{\text{comp}}}_{X_L} \approx \underbrace{L \cdot G \cdot F}_{G_L}$
 - 2 Update G : $\underbrace{X^{\text{comp}} \cdot R}_{X_R} \approx \underbrace{G \cdot F \cdot R}_{F_R}$



Random projections and compressed learning

Dedicated hardware (LightOn)



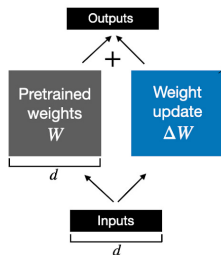
Source : (SAADE et al. 2016).

- Phase retrieval problem (RAJAEI et al. 2016)

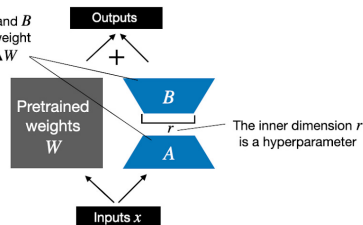
Low-rankness to optimize generative models

① Fine-tuning Large Language Models through Low-rank Adaptation (LoRA) (HU et al. 2021)

Weight update in **regular finetuning**

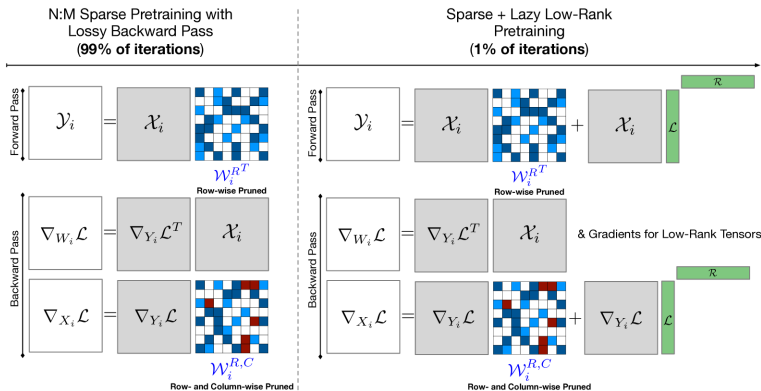


Weight update in **LoRA**



Low-rankness to optimize generative models

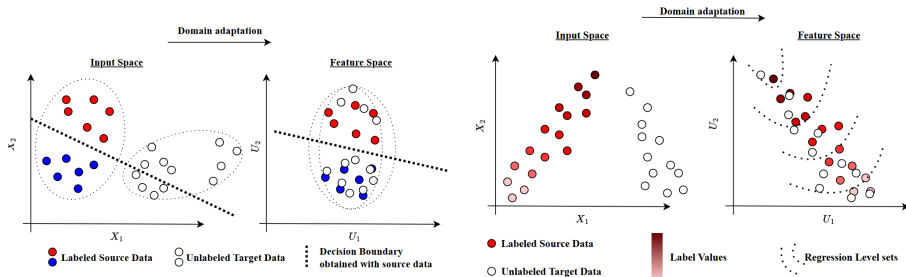
- 1 **Fine-tuning** Large Language Models through Low-rank Adaptation (LoRA) (HU et al. 2021)
- 2 **Sparse + low-rank pre-training LLM** (HAN et al. 2024; MOZAFFARI et al. 2024)



Sparse vs Sparse + Lazy Low-rank pre-training (source : (MOZAFFARI et al. 2024))

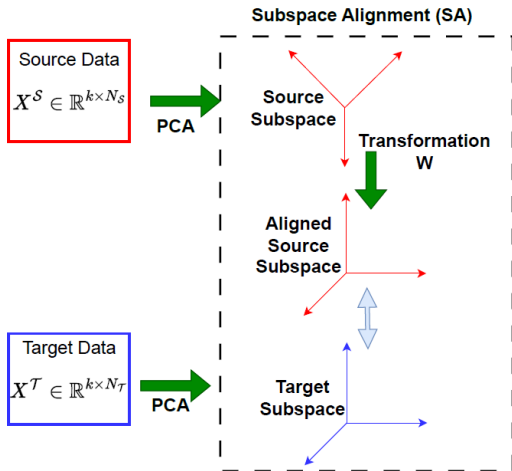
Low-rankness for Data Adaptation (1/2)

- Deep-learning architecture accurate for data close to the training data
- Data too different → Train again the network or apply **data adaptation**



Examples of data augmentation problems. Left : Classification. Right : Regression. Source : (DHAINI 2024)

Low-rankness for Data Adaptation (2/2)



- There is a feature space in which both the target and the source spaces are low-rank
- Data alignment through subspace alignment using :
 - SVDs (CHEN et al. 2021)
 - NMFs (DHAINI 2024)
 - Dictionary learning (ibid.)

Summary

- 1 Foreword
- 2 Classical applications of low-rank models
- 3 In-situ sensor calibration
- 4 Joint multispectral image restoration and unmixing
- 5 Low-rankness in the AI Era
- 6 **Discussion**
 - **Matching Need and Solution**

Discussion

- We covered a very few examples of existing applications of low-rank models
- 📖 **Take-home message #1** : These models are ubiquitous in our digitized world
- 📖 **Take-home message #2** : You will find jobs after your Ph.D. thesis (academics, private sector)

Discussion

- We covered a very few examples of existing applications of low-rank models
- **Take-home message #1** : These models are ubiquitous in our digitized world
- **Take-home message #2** : You will find jobs after your Ph.D. thesis (academics, private sector)
 - Still, objectives in a company might be different from those in research labs

Examples of differences

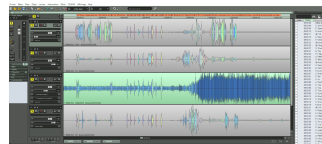
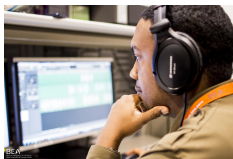
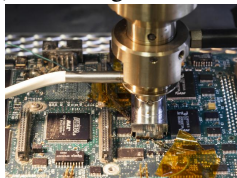
- Different performance criteria (e.g., costs, speed/energy optimization, environment ➤ \$\$\$)
- Matching the need, the constraints, and the solution (example with BEA-LISIC collaboration)

Matching Need and Solution (1/4)

- In case of civil aircraft incident or accident, BEA looks for and analyzes both aircraft “black boxes”



- One black box dedicated to the sound signals recorded in the cockpit : *Cockpit Voice Recorder* (CVR) whose signals are transcribed by CVR audio analysts



Matching Need and Solution (2/4)

Capturing sounds in the cockpit



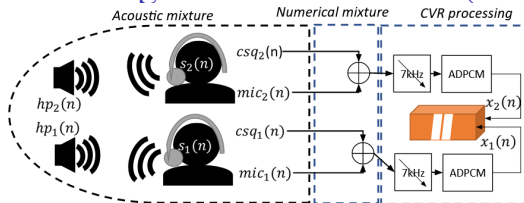
- Headset microphones (up to 3 channels) :
 - unidirectional,
 - mixed to headset signals in the CVR data,
 - sampled at 7 kHz

- Cockpit Area Microphone (CAM) :
 - omnidirectional,
 - sampled at 12 kHz
- ⇒ The CAM channel is not considered in this work

Problem statement

- In case of an incident or accident, the analysis of CVR is a **cocktail party problem**
- ⇒ **Does blind source separation (BSS) can help analyzing these data ?**

Matching Need and Solution (3/4)



Reverse-engineering of the CVR mixing model

- Depending on the active sources, most of the time in between simple (instantaneous) or complex (convolutive) linear models
- SotA methods can be applied (BIGOT, DEVULDER et PUIGT 2022; PUIGT, BIGOT et DEVULDER 2022)

Tested BSS families

- Sparse Component Analysis (SCA)
- Multichannel-NMF

Matching Need and Solution (4/4)

- Case with only 2 sources (pilot / pilot, pilot / radio)
- 46 audio mixtures (25 pilot/pilot, 21 pilot/air traffic control) from 15 CVRs.
- Performance criterion : Word Recognition Rate (WRR)
- Percentage of initially unintelligible words transcribed after applying BSS

		DEMIX	UCBSS	NMF	3 methods
Improved segment rate	pilot / pilot	28%	16%	20%	44%
	pilot / radio	33%	28.5%	38%	66%
WRR	pilot / pilot	50%	55%	57.5%	80%
	pilot / radio	56%	63.6%	70%	89.6%

Matching Need and Solution (4/4)

- Case with only 2 sources (pilot / pilot, pilot / radio)
- 46 audio mixtures (25 pilot/pilot, 21 pilot/air traffic control) from 15 CVRs.
- Performance criterion : Word Recognition Rate (WRR)
- Percentage of initially unintelligible words transcribed after applying BSS

		DEMIX	UCBSS	NMF	3 methods
Improved segment rate	pilot / pilot	28%	16%	20%	44%
	pilot / radio	33%	28.5%	38%	66%
WRR	pilot / pilot	50%	55%	57.5%	80%
	pilot / radio	56%	63.6%	70%	89.6%

- Transcription without BSS

Pilot : première à gauche ouais c'est là on va on va se garer et puis après on (?)

CTRL : three two zero push back approved

CDB : euh (?) (?)

FO : ça on avait (?)

Matching Need and Solution (4/4)

- Case with only 2 sources (pilot / pilot, pilot / radio)
- 46 audio mixtures (25 pilot/pilot, 21 pilot/air traffic control) from 15 CVRs.
- Performance criterion : Word Recognition Rate (WRR)
- Percentage of initially unintelligible words transcribed after applying BSS

		DEMIX	UCBSS	NMF	3 methods
Improved segment rate	pilot / pilot	28%	16%	20%	44%
	pilot / radio	33%	28.5%	38%	66%
WRR	pilot / pilot	50%	55%	57.5%	80%
	pilot / radio	56%	63.6%	70%	89.6%

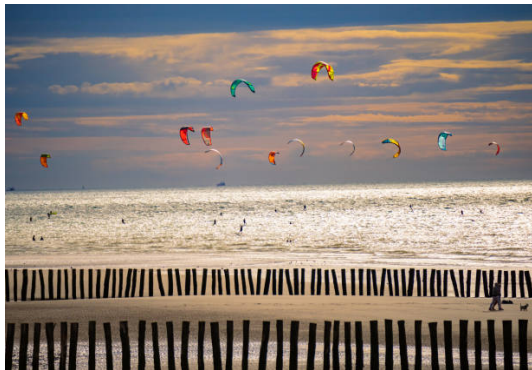
- Transcription with BSS

Pilot : première à gauche ouais c'est là on va on va se garer et puis après on fera le reste hein
CTRL : three two zero push back approved

CDB : euh l'APU est démarré
FO : ça on avait prévenu

Thank you for your attention

- Most papers, some codes, and some talks are available at :
<https://www-lisic.univ-littoral.fr/~puigt/>
- In case you do not find what you are looking for, e-mail-me !



References I



ABBAS, Kinan, Matthieu PUIGT, Gilles DELMAIRE et Gilles ROUSSEL (2024a).

“Locally-Rank-One-Based Joint Unmixing and Demosaicing Methods for Snapshot Spectral Images. Part I: a Matrix-Completion Framework”. In : *IEEE Trans. Comput. Imaging* 10, p. 848-862.



– (2024b). “Locally-Rank-One-Based Joint Unmixing and Demosaicing Methods for Snapshot Spectral Images. Part II: A Filtering-Based Framework”. In : *IEEE Trans. Comput. Imaging* 10, p. 806-817.



BALZANO, Laura et Robert NOWAK (2007). “Blind calibration of sensor networks”. In : *Proceedings of the 6th international conference on Information processing in sensor networks*, p. 79-88.



BIGOT, B., H. DEVULDER et M. PUIGT (juin 2022). “Amélioration de l’intelligibilité de la parole dans des enregistrements de « boîtes noires aéronautiques » à l’aide de méthodes de Séparation Aveugle de Sources”. In : *Actes des JEP 2022*. Noirmoutier, France.

References II

-  CAO, Xiaochun, Xiao WANG, Di JIN, Yixin CAO et Dongxiao HE (2013). “Identifying overlapping communities as well as hubs and outliers via nonnegative matrix factorization”. In : *Scientific reports* 3.1, p. 2993.
-  CHEN, Xinyang, Sinan WANG, Jianmin WANG et Mingsheng LONG (2021). “Representation Subspace Distance for Domain Adaptation Regression.”. In : *ICML*, p. 1749-1759.
-  COMON, P. et C. JUTTEN (2010). *Handbook of blind source separation. Independent component analysis and applications*. Academic press.
-  DEVILLE, Yannick (2014). “Sparse component analysis: A general framework for linear and nonlinear blind source separation and mixture identification”. In : *Blind Source Separation*. Springer, p. 151-196.
-  DHAINI, Mohamad (2024). “Apprentissage profond pour l’analyse de la pollution des sols par imagerie hyperspectrale”. Thèse de doct. Université de Rouen Normandie.
-  DING, Chris HQ, Tao LI et Michael I JORDAN (2008). “Convex and semi-nonnegative matrix factorizations”. In : 32.1, p. 45-55.

References III



DORFFER, C., M. PUIGT, G. DELMAIRE et G. ROUSSEL (juil. 2016). “Nonlinear Mobile Sensor Calibration Using Informed Semi-Nonnegative Matrix Factorization With a Vandermonde Factor”. In : *Proc. IEEE SAM’16*.



– (2017). “Outlier-robust calibration method for sensor networks”. In : *Proc. IEEE ECMSM’17*.



– (déc. 2018). “Informed Nonnegative Matrix Factorization Methods for Mobile Sensor Network Calibration”. In : 4.4, p. 667-682.



DRINEAS, Petros et Michael W MAHONEY (2016). “RandNLA: randomized numerical linear algebra”. In : *Communications of the ACM* 59.6, p. 80-90.



EDELMAN, Alan, Tomás A ARIAS et Steven T SMITH (1998). “The geometry of algorithms with orthogonality constraints”. In : *SIAM journal on Matrix Analysis and Applications* 20.2, p. 303-353.



ERICHSON, N Benjamin, Krithika MANOHAR, Steven L BRUNTON et J Nathan KUTZ (2020). “Randomized CP tensor decomposition”. In : *Machine Learning: Science and Technology* 1.2, p. 025012.

References IV



GEELLEN, B., N. TACK et A. LAMBRECHTS (mar. 2014). “A compact snapshot multispectral imager with a monolithically integrated per-pixel filter mosaic”. In : *Advanced Fabrication Technologies for Micro/Nano Optics and Photonics VII*. Sous la dir. de G. von FREYMAN, W. V. SCHOENFELD et R. C. RUMPF. SPIE.



GILLIS, Nicolas (2020). *Nonnegative matrix factorization*. SIAM.



GRIBONVAL, Rémi (2014). *Sparse dictionary learning in the presence of noise and outliers*. Keynote during IEEE MLSP conference.



HAN, Andi, Jiaxiang LI, Wei HUANG, Mingyi HONG, Akiko TAKEDA, Pratik JAWANPURIA et Bamdev MISHRA (2024). “SLTrain: a sparse plus low-rank approach for parameter and memory efficient pretraining”. In : *arXiv preprint arXiv:2406.02214*.



HE, Chaobo, Xiang FEI, Qiwei CHENG, Hanchao LI, Zeng HU et Yong TANG (2021). “A survey of community detection in complex networks using nonnegative matrix factorization”. In : *IEEE Transactions on Computational Social Systems* 9.2, p. 440-457.

References V



HU, Edward J, Yelong SHEN, Phillip WALLIS, Zeyuan ALLEN-ZHU, Yuanzhi LI, Shean WANG, Lu WANG et Weizhu CHEN (2021). “Lora: Low-rank adaptation of large language models”. In : *arXiv preprint arXiv:2106.09685*.



KUMAR, Abhishek et Vikas SINDHWANI (2015). “Near-separable non-negative matrix factorization with ℓ_1 and Bregman loss functions”. In : *Proc. SIAM ICDM 2015*. SIAM, p. 343-351.



LEE, D.D. et H.S. SEUNG (1999). “Learning the parts of objects by non negative matrix factorization”. In : *Nature* 401.6755, p. 788-791.



LEE, Daniel D. et H. Sebastian SEUNG (2001). “Algorithms for Non-negative Matrix Factorization”. In : *Proc. NIPS'01*. Sous la dir. de T. K. LEEN, T. G. DIETTERICH et V. TRESP. MIT Press, p. 556-562.



LI, Xuelong et Yawei PANG (jan. 2010). “Deterministic Column-Based Matrix Decomposition”. In : 22.1, p. 145-149.



MAAG, Balz, Zimu ZHOU et Lothar THIELE (2018). “A survey on sensor calibration in air pollution monitoring deployments”. In : *IEEE Internet of Things Journal* 5.6, p. 4857-4870.

References VI

-  MANILOW, Ethan, Prem SEETHARMAN et Justin SALAMON (2020). *Open Source Tools & Data for Music Source Separation*. <https://source-separation.github.io/tutorial>. URL : <https://source-separation.github.io/tutorial>.
-  MOZAFFARI, Mohammad, Amir YAZDANBAKSH, Zhao ZHANG et Maryam Mehri DEHNAVI (2024). “SLoPe: Double-Pruned Sparse Plus Lazy Low-Rank Adapter Pretraining of LLMs”. In : *arXiv preprint arXiv:2405.16325*.
-  NIKITAKI, Sofia, Grigorios TSAGKATAKIS et Panagiotis TSAKALIDES (2015). “Efficient multi-channel signal strength based localization via matrix completion and Bayesian sparse learning”. In : 14.11, p. 2244-2256.
-  ORTIZ-BOUZA, Meiby et Selin AVIYENTE (2024). “Community detection in multiplex networks based on orthogonal nonnegative matrix tri-factorization”. In : *IEEE Access*.
-  PUIGT, M., B. BIGOT et H. DEVULDER (2022). “Apport de la séparation de sources au traitement des signaux audiophoniques issus de « boîtes noires aéronautiques »”. In : *Actes du GRETSI*. Nancy, France, p. 1001-1004.

References VII

-  QU, Wentao, Xianchao XIU, Huangyue CHEN et Lingchen KONG (2023). “A survey on high-dimensional subspace clustering”. In : *Mathematics* 11.2, p. 436.
-  RAJAEI, Boshra, Eric W TRAMEL, Sylvain GIGAN, Florent KRZAKALA et Laurent DAUDET (2016). “Intensity-only optical compressive imaging using a multiply scattering material and a double phase retrieval approach”. In : *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, p. 4054-4058.
-  RUBINSTEIN, Ron, Michael ZIBULEVSKY et Michael ELAD (2009). “Double sparsity: Learning sparse dictionaries for sparse signal approximation”. In : 58.3, p. 1553-1564.
-  SAADE, A., F. CALTAGIRONE, I. CARRON, L. DAUDET, A. DRÉMEAU, S. GIGAN et F. KRZAKALA (2016). “Random projections through multiple optical scattering: Approximating Kernels at the speed of light”. In : *Proc. ICASSP'16*, p. 6215-6219.
-  SAUKH, Olga, David HASENFRATZ et Lothar THIELE (2015). “Reducing multi-hop calibration errors in large-scale mobile sensor networks”. In : *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, p. 274-285.

References VIII

-  SAUKH, Olga, David HASENFRATZ, Christoph WALSER et Lothar THIELE (2014). “On rendezvous in mobile sensing networks”. In : *Real-World Wireless Sensor Networks*. Springer, p. 29-42.
-  SOBRAL, Andrews, Thierry BOUWMANS et El-hadi ZAHZAH (2015). “LRSLibrary: Low-Rank and Sparse tools for Background Modeling and Subtraction in Videos”. In : *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. CRC Press, Taylor et Francis Group.
-  TEPPER, M. et G. SAPIRO (mai 2016). “Compressed Nonnegative Matrix Factorization Is Fast and Accurate”. In : 64.9, p. 2269-2283.
-  TROPP, Joel A (2006). “Algorithms for simultaneous sparse approximation. Part II: Convex relaxation”. In : *Signal Processing* 86.3, p. 589-602.
-  TSAGKATAKIS, G., M. BLOEMEN, B. GELEN, M. JAYAPALA et P. TSAKALIDES (juin 2019). “Graph and Rank Regularized Matrix Recovery for Snapshot Spectral Image Demosaicing”. In : *IEEE Trans. Comput. Imaging* 5.2, p. 301-316.

References IX



VU THANH, Olivier, Matthieu PUIGT, Farouk YAHAYA, Gilles DELMAIRE et Gilles ROUSSEL (juin 2021). “In situ calibration of cross-sensitive sensors in mobile sensor arrays using fast informed non-negative matrix factorization”. In : *Proc. IEEE ICASSP 2021*.



WÓJCIK, Piotr Iwo et Marcin KURDZIEL (2019). “Training neural networks on high-dimensional data using random projection”. In : *Pattern Analysis and Applications* 22.3, p. 1221-1231.



YAHAYA, F. (2021). “Compressive informed (semi-)non-negative matrix factorization methods for incomplete and large-scale data, with application to mobile crowd-sensing data”. *Thèse de doct.* Université du Littoral Côte d’Opale.



ZHOU, Tianyi et Dacheng TAO (2011). “Godec: Randomized low-rank & sparse matrix decomposition in noisy case”. In : *Proc. ICML 2011*.