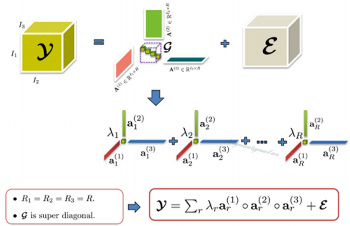
Tensor approaches for source separations: from applications to algorithms (and vice versa)

Xavier Luciani

Université de Toulon, CNRS, LSIS, UMR 7296, 83957 La Garde, France.
Aix Marseille Université, CNRS, ENSAM, LSIS, UMR 7296, 13397 Marseille, France.


I_3
I_1
I_2
$\mathcal{Y}$
$\mathcal{G}$
$\mathbf{A}^{(2)} \in \mathbb{R}^{I_2 \times R}$
$\mathbf{A}^{(1)} \in \mathbb{R}^{I_1 \times R}$
$\mathbf{A}^{(3)} \in \mathbb{R}^{I_3 \times R}$
$\mathcal{E}$
λ_1 $\mathbf{a}_1^{(2)}$ $\mathbf{a}_1^{(1)}$ $\mathbf{a}_1^{(3)}$
λ_2 $\mathbf{a}_2^{(2)}$ $\mathbf{a}_2^{(1)}$ $\mathbf{a}_2^{(3)}$
λ_R $\mathbf{a}_R^{(2)}$ $\mathbf{a}_R^{(1)}$ $\mathbf{a}_R^{(3)}$

- $R_1 = R_2 = R_3 = R$.
- $\mathcal{G}$ is super diagonal.

$$\mathcal{Y} = \sum_r \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \mathbf{a}_r^{(3)} + \mathcal{E}$$

Workshop *Tensor decompositions and covariance matrix estimation*
November 27th, 2015, Marseille

A brief personal introduction

Currently

- Maitre de conférence at the Laboratoire des Sciences de l'Information et des Systèmes in the team SIgnal et IMage and at University of Toulon depuis 2013.

Research interests

- Algorithms for tensor decompositions
- Joint diagonalization
- Source separation based on tensor methods
- Numerical analysis of fluorescence spectroscopy signals

A common thread: the tensor decompositions.

A brief personal introduction

Thesis

- *Analyse numérique des spectres de fluorescence 3D issus de mélanges non linéaires*, PROTEE Laboratory, UTLN.
  → Linear and non-linear tensor modelization and decompositions of fluorescence signals.

Post doctoral positions

- *Algorithmes de décomposition tensorielle et séparation de sources, application aux télécommunications*, I3S Laboratory, UNSA.
  → Algorithms for tensors decompositions and new tonsorial methods for BSS (BI) of mixture of telecommunications signals.
- *Algorithmes de décomposition tensorielle et de diagonalisation conjointe*, LTSI Laboratory, University of Rennes 1.
  → Development of original algorithms for tensor decompositions and joint diagonalization.
- *Séparation des spectres de fluorescence*, PROTEE Laboratory, UTLN.
  → Non-linear tensor modelization and decompositions of fluorescence signals.

OUTLINE

1. Tensor decompositions in sources separation

2. Example of applications of tensor decompositions in sources separation
   - Application example in analytical chemistry
   - Application example in telecommunications
   - Application example in Independent Components Analysis

3. Algorithms for Canonical Polyadic Decomposition (CPD)

4. Solving the overfactoring problem with DIAG: application to fluorescence spectroscopy

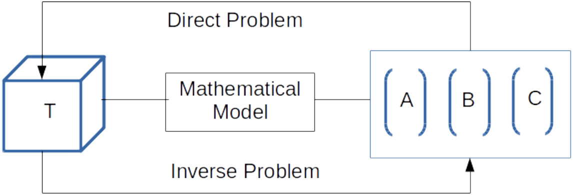
OUTLINE

1. Tensor decompositions in sources separation

What do we call a tensor decomposition ?

Basic definitions

- Tensor: for us a tensor is simply a multiway array of order greater than 2.
- Tensor decomposition: decompose a tensor as a sum of other tensors.
- For practical applications : known data are stored in an order Q tensor and we fit a mathematical model involving Q factor matrices containing the desired information.



WHAT DO WE CALL A TENSOR DECOMPOSITION ?

In the sources separation context

- Each column of the factors matrices is linked to a source. Factor matrices have the same number of columns which is equal to the number of sources.
- Thus, an order Q tensor of size $(I_1, \cdots, I_Q)$ will be decomposed as

$$\mathcal{T}_{i_1 i_2 \cdots i_Q} = \sum_{n=1}^{N} \left[f_n \left(\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(Q)} \right) \right]_{i_1 i_2 \cdots i_Q} \tag{1}$$

- Where $\mathbf{A}^{(1)}$, $\cdots$, $\mathbf{A}^{(Q)}$ are the factor matrices of sizes (I_1, N), $\cdots$, (I_Q, N).
- N defines the rank of the decomposition and is equal to the number of sources.

WHY USING TENSOR DECOMPOSITIONS ?

- They allow to exploit the multidimensional structure of some data set.
- Conversely to the "matrix case", in many practical situation the inverse problem (*i.e.* finding the elements of the factor matrices of the decomposition from the element of the tensor) has an unique solution !
- They allow to deal with more sources than observations

SOME TD USED IN SOURCE SEPARATION PROBLEMS

Canonical Polyadic Decomposition (CPD)

$$\mathcal{T}_{i_1 i_2 \cdots i_Q} = \sum_{n=1}^{N} \mathbf{A}^{(1)}_{i_1 n} \mathbf{A}^{(2)}_{i_2 n} \cdots \mathbf{A}^{(Q)}_{i_Q n} \tag{2}$$

- Introduced in [Hitchcock, 1927] and popularized in [Harshman, 1970].
- The simplest tensor decomposition, also known as PARAFAC and CANDECOMP.
- Has an unique solution within some weak conditions.
- Every tensor has an exact CPD and in this case N also defines the tensor rank.
- Used in most applications: Data analysis, telecommunications (MIMO systems), Independent Component Analysis (ICA) biomedical (EEG, fMRI...), spectroscopy (fluorescence, chromatography...)...

Some TD used in source separation problems

Non linear Fluorescence Decomposition (NLFD)

$$\mathcal{T}_{ijk} = \sum_{p=1}^{P} A_{ip} B_{jp} C_{kp} \prod_{p=1}^{P} e^{-C_{kp}(A_{ip}+A_{jp})} \tag{3}$$

- Introduced in [Luciani, 2009] and deeply studied in [Cohen, Luciani and Comon, 2015]
- Specific to the decomposition of non linear fluorescence measurements.
- Have (locally) an unique solution within some weak conditions.

Some TD used in source separation problems

Tucker Decomposition (of order 3)

$$\mathcal{T}_{ijk} = \sum_{f_1=1}^{F_1} \sum_{f_2=1}^{F_2} \sum_{f_3=1}^{F_3} A_{if_1} B_{jf_2} C_{kf_3} \mathcal{X}_{f_1 f_2 f_3} \tag{4}$$

- Introduced in [Hitchcock, 1927] and popularized in [Tucker, 1966].
- $\mathcal{X}$ is called the core tensor.
- Does generally not have an unique solution.
- Used in data analysis, data compression...

Some TD used in source separation problems

CONFAC Decomposition (of order 3)

$$\mathcal{T}_{ijk} = \sum_{f_1=1}^{F_1} \sum_{f_2=1}^{F_2} \sum_{f_3=1}^{F_3} A_{if_1} B_{jf_2} C_{kf_3} \mathcal{X}_{f_1 f_2 f_3}(\Theta, \Psi, \Omega), \tag{5}$$

where

$$\mathcal{X}_{f_1 f_2 f_3}(\Theta, \Psi, \Omega) = \sum_{n=N}^{F} \Theta_{f_1 n} \Psi_{f_2 n} \Omega_{f_3 n}. \tag{6}$$

- Introduced in [De Almeida, 2008]
- Constraint matrices, Θ, Ψ and Γ are full row-rank matrices.
- Their columns are canonical vectors possibly multiplied by -1.
- Particular case of the Tucker decomposition.
- Has an unique solution within some weak conditions
- Used in telecommunications (MIMO systems), ICA...

TENSOR METHODS FOR SOURCE SEPARATIONS

- We have to distinguish two main kinds of tensor methods for source separation.
- This entirely depends of the nature of the data.

A. Direct methods

Here the known data have an intrinsic tensor structure. This means that:

- Each source is a $(Q-1)$-dimensions signal with $Q > 2$.
- Data are actually a set of mixtures of the sources. Thereby, they vary according to Q variables and thus can be directly stored in a tensor of order Q.
- This tensor can be modelled using an appropriate tensor decomposition such that mixture parameters and source signals are directly obtained from the factor matrices of the decomposition.
- This case is of course very interesting since it allows to solve directly the problem (providing the tensor decomposition is unique and we dispose of an efficient decomposition algorithm)
- We will see two concrete applications in chemistry and in telecommunications.

TENSOR METHODS FOR SOURCE SEPARATIONS

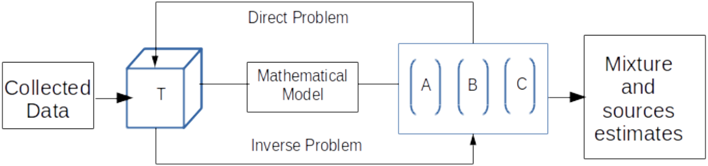
B. Indirect methods

Here the known data does not have an intrinsic tensor structure.

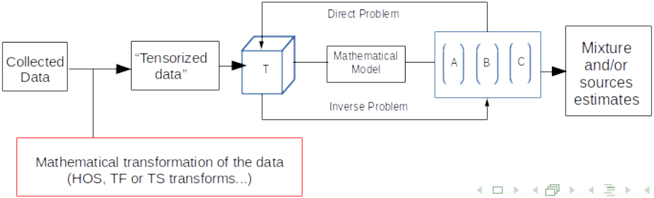
- Usually this means that the data vary according to only 1 or 2 and can only be stored in a vector or a matrix.
- Thereby we have to perform some mathematical transformation of the data in order to increase their diversities and store them in a tensor.
- If the transformation is suitable this tensor can be modelled using a tensor decomposition such that mixture parameters and/or source signals are directly obtained from the factor matrices of the decomposition.
- This is notably the case of ICA and traditional BSS and BI of Mixtures.
- According to their nature, there are several ways to increase the diversities of the data: Fourier transform, Cumulants, Derivatives of the characteristic function...
- These methods usually require some assumptions about the source signal.

TENSOR METHODS FOR SOURCE SEPARATIONS

A. Direct methods



B. Indirect methods


Direct Problem
Collected Data
"Tensorized data"
T
Mathematical Model
A
B
C
Mixture and/or sources estimates
Inverse Problem
Mathematical transformation of the data
(HOS, TF or TS transforms...)

Outline

2. Example of applications of tensor decompositions in sources separation
   - Application example in analytical chemistry
   - Application example in telecommunications
   - Application example in Independent Components Analysis

Outline

2 Example of applications of tensor decompositions in sources separation

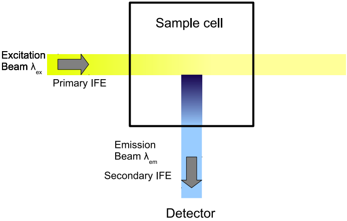
Application example in analytical chemistry

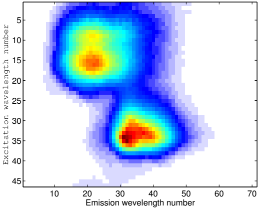
Application example in telecommunications

Application example in Independent Components Analysis

FLUORESCENCE SPECTROSCOPY

Excitation and Emission Matrix of fluorescence (EEM)





- We measure the fluorescence intensity of a solution for different couples of exciting and emitting wavelengths.
- 2 wavelength diversities ⇒ **2 way data**: EEM $\mathbf{X}$

THE SOURCE SIGNALS

- A solution is generally a mixture of several fluorescent components (fluorophores)
- Each fluorophore is characterized by its own EEM.
- We consider I excitation wavelengths and J emission wavelengths
- The EEM, $\mathbf{Y}_n$, of a fluorophore n is a rank 1 matrix of size (I, J) so that $\mathbf{Y}_n = \mathbf{a}_n \mathbf{b}_n^{\top}$ where:
  - $a_n[i]$ is the capacity of fluorophore n to absorb light at excitation wavelength number i.
  - $b_n[j]$ is the capacity of fluorophore n to emit light at emission wavelength number j.
- Thus, $\mathbf{a}_n$ and $\mathbf{b}_n$ define the excitation and emission spectra of fluorophore n.
- Matrix $\mathbf{Y}_n$ can be seen as the source signal associated to fluorophore n.

FLUORESCENT MIXTURES

- Of course the fluorescence intensity of a fluorophore is also a function of its concentration c_n
- A fluorescent solution can be seen as a mixture of N fluorophores and we can show that a linear approximation gives:

$$\mathbf{X} = \sum_{n=1}^{N} c_n \mathbf{Y}_n \tag{7}$$

where **X** is the EEM of the mixture .

CPD OF THE FLUORESCENCE TENSOR [BRO, 1997]

- In practice we dispose of a set of $K > 1$ solutions of the same N fluorophores and we have:

$$\mathbf{X}_k = \sum_{n=1}^{N} C_{nk} \mathbf{Y}_n \tag{8}$$

  where $\mathbf{X}_k$ is the EEM of solution k and C_{nk} is the concentration of fluorophore n in solution k.

- We can then store all matrices $\mathbf{X}_k$ in an order 3 tensor $\mathcal{T}$ of size (I, J, K), recalling that $\mathbf{Y}_n = \mathbf{a}_n \mathbf{b}_n^\intercal$ and defining $A_{i,n} = a_n[i]$ and $B_{j,n} = b_n[j]$ we have:

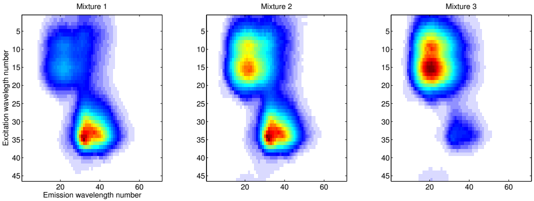
$$\mathcal{T}_{i,j,k} = \sum_{n=1}^{N} A_{i,n} B_{j,n} C_{k,n} \tag{9}$$

- Thereby, the CPD of $\mathcal{T}$ gives directly the excitation spectra (columns of $\mathbf{A}$), the emission spectra (columnd of $\mathbf{B}$) and the concentration profile (columnd of $\mathbf{C}$) of each fluorophore without any other knowledge.

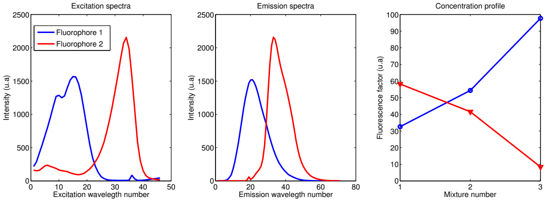
Example 1

EEM set

We have 3 EEM corresponding to 3 different mixtures of 2 fluorophores



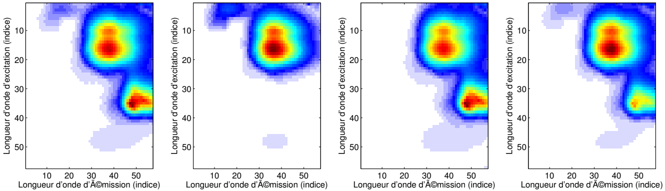
CP decomposition of $\mathcal{X}$



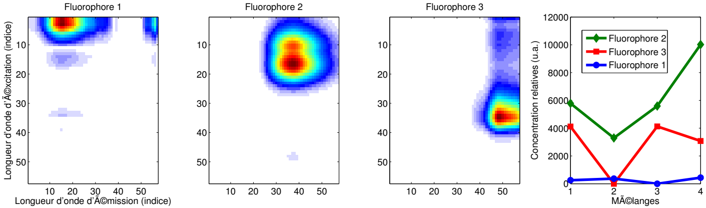
EXAMPLE 2

EEM set

We have 4 EEM corresponding to 4 different mixtures of 3 fluorophores



CP decomposition of $\mathcal{X}$

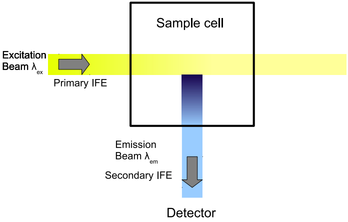


INNER FILTER EFFECTS (IFE)

Definition

Gradual absorption of the exciting and emitted (fluorescence) lights throughout the sample cell during the FEEM acquisition [Parker and Barnes, 1957] [lakowicz, 1983].

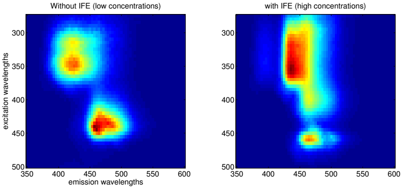
IFE dramatically increases with the concentration !



Example of IFE

Mixture of Quinine sulphate and fluorescein:

- Left: low concentrate mixture
- Right: Highly concentrate mixture



THE NFLD: A NON LINEAR TENSOR MODEL

IFE modelisation

In order to simplified the notation, we assume that excitation and emission range are equal. IFE can be modelized using the beer-lamber law and after some simplifications of physical phenomena, it comes:

$$\mathcal{T}_{ijk} = \sum_{p=1}^{P} A_{ip}B_{jp}C_{kp} \prod_{p=1}^{P} e^{-C_{kp}(A_{ip}+A_{jp})} \tag{10}$$

We call this tensor decomposition the Non Linear Fluorescence Decomposition [Luciani *et al.*,2009] [Cohen and Comon, 2013][Cohen, Luciani and Comon, 2015]

Identifiability of the NLFD

- Matrix case (One FEEM) ⇒ the NLFD does not restore the identifiability.
- General case ⇒ We conjecture that the NLFD is generically not locally identifiable if and only if $P = \left\lfloor \frac{IJK}{I+J+K-1} \right\rfloor$ and $J \geq (K-1)(I-1)+3$ or any permutation of this inequality is true.
- Scaling indeterminacy is removed in the emission mode.

CORRECTION METHODS: FROM NLFD TO CPD

Each FEEM is linearized by cancelling the non linear term using an additional measurement. Then the set of linearized FEEM can be decomposed by CPD

Using the absorbance spectra

- Additional measurement : absorption spectrum of the mixture.
- Simple but provides poor results (notably when the absorbency is large).

Using a controlled dilution [Luciani *et al.*, 2009]

- Additional measurement : a second FEEM measured from a controlled (low) dilution of the mixture.
- Simple and efficient but requires a dilution and thus to modify the samples.

Using a mirrored cell [Luciani *et al.*, 2013]

- Additional measurement : a second FEEM measured using a mirrored cell.
- Numerically more sophisticated but does not require to modify the samples.

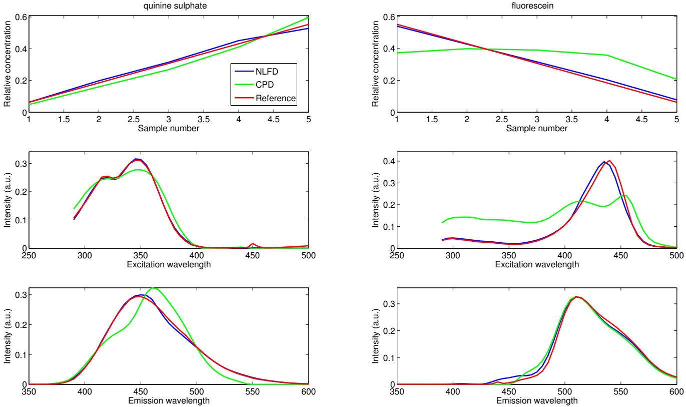
Direct computation of the NLFD

Proposed algorithm: NFLD-LM algorithm [Cohen, Luciani and Comon, 2015]

- The additional information is given by the sample diversity : the model is regarded as a tensor decomposition, locally unique.
- The NLFD can then be performed using the Levenberg-Marquardt optimization method ⇒ NFLD-LM algorithm
- The algorithm is sensitive to bad initialization. DIAG is then used for the initialization since it provides a good starting point very quickly.

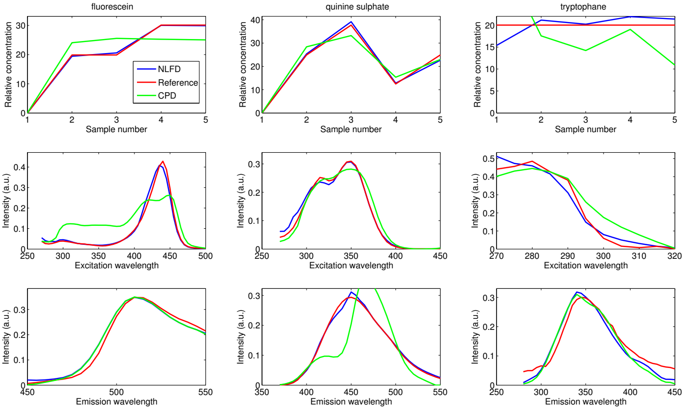
Some experimental results

NLFD and CPD of 5 concentrated mixtures of quinine sulphate and fluorescein.



SOME EXPERIMENTAL RESULTS

NLFD and CPD of 5 concentrated mixtures of tryptophane, quinine sulphate and fluorescein.



OUTLINE

2 Example of applications of tensor decompositions in sources separation

Application example in analytical chemistry

Application example in telecommunications

Application example in Independent Components Analysis

CPD APPLIED TO MIMO /DS-CDMA SYSTEM [SIDIROPOULOS, 2000]

- We consider a simplified MIMO (Multiple-Input Multiple-Output) system of N transmit antennas I receive antennas.
- Transmit antenna n sends a DS-CDMA signal $\mathbf{y}_n$.
- Each DS-CDMA signal is built by spreading an information sequence $s_n[j]$ of J symbols by a particular code sequence $c_n[k]$ of length K.
- Thereby we have $y_n[k+(j-1)K] = s_n[i]c_n[k]$
- Assuming a linear mixture model, receive antenna i receives a signal $\mathbf{x}_i$ so that $\mathbf{x}_i = \sum_{n=1}^{N} H_{i,n}\mathbf{y}_n$
- The mixture matrix $\mathbf{H}$ models the fading/gain between each couple of transmit and receive antennas.
- $\mathbf{x}_i$ and $\mathbf{y}_n$ are signals of length JK.

MIMO /DS-CDMA SYSTEM

- From the I signals $\mathbf{x}_i$ we can then build a third order tensor $\mathcal{X}$ of size (I, J, K) so that:

$$\mathcal{X}_{i,j,k} = x_i[k + (j-1)K] \tag{11}$$

- We then have

$$\mathcal{X}_{i,j,k} = \sum_{n=1}^{N} H_{i,n} Y_n[k + (j-1)K] \tag{12}$$

- Defining $S_{j,n} = s_n[j]$ and $C_{k,n} = c_n[k]$ yields:

$$\mathcal{X}_{i,j,k} = \sum_{n=1}^{N} H_{i,n} S_{j,n} C_{k,n} \tag{13}$$

- In other words, the CPD of $\mathcal{X}$ gives a direct and deterministic estimation of the mixing matrix, the source signals and the coding sequences.

OUTLINE

2. Example of applications of tensor decompositions in sources separation
   - Application example in analytical chemistry
   - Application example in telecommunications
   - Application example in Independent Components Analysis

Identification of underdetermined mixtures

Linear model of mixture

$$\mathbf{z}(m) = \mathbf{H}\mathbf{s}(m) + \mathbf{n}(m),\ m = 1 \cdots M$$

- $\mathbf{z} \in \mathbb{C}^N$ or $\mathbb{R}^N$: observation random vector
- $\mathbf{s} = [s_1, \ldots, s_K]^T \in \mathbb{C}^K$: source random vector
- $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_K] \in \mathbb{C}^{N \times K}$: mixing matrix
- $\mathbf{n} \in \mathbb{C}^N$: noise random vector

Underdetermined Blind Identification (BI) problem

- Estimate of $\mathbf{H}$ from $\mathbf{z}(m)$
- More sources than sensors! ($K > N$)

IDENTIFICATION OF UNDERDETERMINED MIXTURES

Linear model of mixture

$$\mathbf{z}(m) = \mathbf{H}\mathbf{s}(m) + \mathbf{n}(m),\ m = 1 \cdots M$$

- $\mathbf{z} \in \mathbb{C}^N$ or $\mathbb{R}^N$: observation random vector
- $\mathbf{s} = [s_1, \ldots, s_K]^T \in \mathbb{C}^K$: source random vector
- $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_K] \in \mathbb{C}^{N \times K}$: mixing matrix
- $\mathbf{n} \in \mathbb{C}^N$: noise random vector

Assumptions

1. **H** does not contain collinear columns
2. Sources are mutually independent and non-Gaussian
3. Sources and noise are statistically independent
4. The number of sources is known

TENSOR-BASED METHODS FOR BI

How to

1. Transform mathematically the data, using HOS such as Cumulants or derivatives of the characteristic function (CAF).
2. Store transformed data into an higher order array (tensor) whose appropriate decomposition provides an estimate of **H**,
3. Perform the tensor decomposition.

Underlying idea

- Uniqueness properties
- Able to deal with the underdetermined case
- Number of efficient algorithms

BI CAF-BASED ALGORITHMS

1. Find a relationship between observations and sources characteristic (or generating) functions → core equation
2. Choose S differentiation points
3. Compute P-th order partial derivatives of the core equation with respect to each point (here P=3)
4. Store the derivative results in a tensor $\mathcal{X}$
5. Estimate $\mathbf{H}$ from an appropriate decomposition of $\mathcal{X}$

Real sources vs. Complex sources

- Real case: CP decomposition [Comon 2004]
- Complex case: CONFAC decomposition [Luciani, 2010] [de Almeida, 2012]

BI CAF-BASED ALGORITHMS IN THE REAL CASE

- Let us denote Φ_z and φ_k the second generating functions of the observations and source k respectively:

$$\varphi_k(x) \overset{\text{def}}{=} \log \mathsf{E}[\exp(x s_k)], \quad x \in \mathbb{R},$$

$$\Phi_z(\mathbf{u}^s) \overset{\text{def}}{=} \log \mathsf{E}[\exp(\mathbf{u}^{s\mathsf{T}}\mathbf{z})], \quad \mathbf{u}^s \in \mathbb{R}^N.$$

- Replacing $\mathbf{z}$ by its model and neglecting the noise contribution yield:

$$\Phi_z(\mathbf{u}^s) = \log \mathsf{E}[\exp\left(\sum_{n,k} u_n^s H_{nk} s_k\right)].$$

- Using the source independence property, we get:

$$\Phi_z(\mathbf{u}^s) = \sum_k \varphi_k\left(\sum_n H_{nk} u_n^s\right). \tag{14}$$

BI CAF-based algorithms in the real case

$$\Phi_z(\mathbf{u}^s) = \sum_k \varphi_k \left(\sum_n H_{nk} u_n^s \right). \tag{15}$$

- Differentiating (15) 2 times in S points $(\mathbf{u}^1 \cdots \mathbf{u}^S)$ of $\mathbb{R}^N$ yields:

$$\frac{\partial^2 \Phi_z(\mathbf{u}^s)}{\partial u_i \partial u_j} = \sum_{k=1}^{K} H_{ik} H_{jk} G_{sk}, \tag{16}$$

- where $G_{sk} \overset{\text{def}}{=} \frac{\partial^2 \varphi_k(\sum_n H_{nk} u_n^s)}{\partial u_i \partial u_j}$
- We can thus construct a third order tensor, $\mathcal{T}$, of size (N,N,S) so that $\mathcal{T}_{ijs} = \frac{\partial^2 \Phi_z(\mathbf{u}^s)}{\partial u_i \partial u_j}$
- Equation (16) shows that the CPD of $\mathcal{T}$ allows to directly estimate the mixing matrix $\mathbf{H}$.

BI CAF-based algorithms in the real case

- In practice, in order to build **c** we can easily calculate formal estimators of the partial derivative of Φ_z at a given point based on sample means.
- Taking various differentiation points allows to apply a tensor method using SO statistics only.
- Since the maximal CPD rank keeping the unicity property is greater than the smallest tensor dimension, if we take at least N differentiation points we can deal with more sources (CPD rank) than observations (tensor smallest dimensions).

BI CAF-based algorithms in the real case

- Of course this approach can be generalized at higher differentiation order:

$$\frac{\partial^p \Phi_z(\mathbf{u}^s)}{\partial u_{n_1} \partial u_{n_2} \cdots \partial u_{n_P}} = \sum_{k=1}^{K} H_{n_1 k} H_{n_2 k} \cdots H_{n_P k} G_{sk}, \tag{17}$$

- with $n_p = 1, \ldots, N$ and $p = 1, \ldots, P$ and where $G_{sk} \overset{\text{def}}{=} \frac{\partial^p \varphi_k(\sum_n H_{nk} u_n^s)}{\partial u_{n_1} \partial u_{n_2} \cdots \partial u_{n_P}}$.
- This allows to increase the tensor order hence the CPD maximal rank, hence the maximal number of sources.
- The counter part is that the statistic require more samples to be correctly estimated.
- Algorithms family based on this approach: ALESCAF, LEMACAF, lEMACAFC, CONFAC-ELS.

OUTLINE

3 Algorithms for Canonical Polyadic Decomposition (CPD)

CPD ALGORITHMS

Iterative algorithms

- Alternating Least Squares (ALS) [Harshman, 1970] [Bro, 1998]
- Gradient, conjugate gradient, LM, BFGS...[Tomasi and Bro, 2006] [Comon, Luciani and de Almeida, 2009][Acar *et al.*, 2011]
- Enhanced Line Search (ELS) [Rajih *et al.*, 2008] [Comon, 2009]
- Pro: Precision, implementation of positive constraints...
- Cons: Convergence problems, over-factoring, correlated factors

Semi-algebraic approaches

Idea: rewrite the CPD as a Joint Diagonalization problem

- By congruence [de Lathauwer *et al.*, 2004]
- By similarity (JEVD): Closed Form Solution (CFS) [Roemer and Haardt 2006, 2008] and DIrect AlGorithm [DIAG, Luciani and Albera 2011, 2012]

Algorithms for the CPD

Alternating Least Squares

- $\mathbf{Z}_A = \mathbf{C} \odot \mathbf{B}$
- $\mathbf{Z}_B = \mathbf{A} \odot \mathbf{C}$
- $\mathbf{Z}_C = \mathbf{B} \odot \mathbf{A}$
- $\mathbf{X}_{I,JK} = \mathbf{A}\mathbf{Z}^{\top}{}_{A} + \mathbf{E}_{I,JK}$
- $\mathbf{X}_{J,KI} = \mathbf{B}\mathbf{Z}^{\top}{}_{B} + \mathbf{E}_{J,KI}$
- $\mathbf{X}_{K,IJ} = \mathbf{C}\mathbf{Z}^{\top}{}_{C} + \mathbf{E}_{K,IJ}$
- $\mathcal{C}_{I,JK} = ||\mathbf{X}_{I,JK} - \widehat{\mathbf{A}}(\widehat{\mathbf{C}} \odot \widehat{\mathbf{B}})^{\top}||^2$

1. Initialize **B** and **C**
2. Estimate of **A**, **B** and **C** :

$$\widehat{\mathbf{A}} = \mathbf{X}_{I,JK}\mathbf{Z}_A(\mathbf{Z}^{\top}{}_{A}\mathbf{Z}_A)^{-1}$$

Compute $\mathbf{Z}_B$ and estimate **B**:

$$\widehat{\mathbf{B}} = \mathbf{X}_{J,KI}\mathbf{Z}_B(\mathbf{Z}^{\top}{}_{B}\mathbf{Z}_B)^{-1}$$

Compute $\mathbf{Z}_C$ and estimate **C**:

$$\widehat{\mathbf{C}} = \mathbf{X}_{K,IJ}\mathbf{Z}_C(\mathbf{Z}^{\top}{}_{C}\mathbf{Z}_C)^{-1}$$

3. Compute: $\mathcal{C}_{I,JK}$
4. restart from 2 till convergence.

ALGORITHMS FOR THE CPD

Gradient descent

Vector of parameters **p** and gradient **g** of the quadratic criterion:

$$\mathbf{p} = \begin{bmatrix} \mathbf{vec}[\mathbf{A}^\top] \\ \mathbf{vec}[\mathbf{B}^\top] \\ \mathbf{vec}[\mathbf{C}^\top] \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} \mathbf{g}_\mathbf{A} \\ \mathbf{g}_\mathbf{B} \\ \mathbf{g}_\mathbf{C} \end{bmatrix}$$

update **p**:

$$\mathbf{p}(t+1) = \mathbf{p}(t) - \mu(t)\,\mathbf{g}(t)$$

We have:

$$\begin{aligned}
\mathbf{g}_\mathbf{A} &= [\mathbf{I}_A \otimes (\mathbf{C}^\top\mathbf{C} \boxdot \mathbf{B}^\top\mathbf{B})]\mathbf{vec}[\mathbf{A}^\top] - [\mathbf{I}_A \otimes (\mathbf{C} \odot \mathbf{B})]\mathbf{vec}\left[\mathbf{X}_{KJ\times I}\right] \\
\mathbf{g}_\mathbf{B} &= [\mathbf{I}_B \otimes (\mathbf{A}^\top\mathbf{A} \boxdot \mathbf{C}^\top\mathbf{C})]\mathbf{vec}[\mathbf{B}^\top] - [\mathbf{I}_A \otimes (\mathbf{A} \odot \mathbf{C})]\mathbf{vec}\left[\mathbf{X}_{IK\times J}\right] \\
\mathbf{g}_\mathbf{C} &= [\mathbf{I}_C \otimes (\mathbf{B}^\top\mathbf{B} \boxdot \mathbf{A}^\top\mathbf{A})]\mathbf{vec}[\mathbf{C}^\top] - [\mathbf{I}_C \otimes (\mathbf{B} \odot \mathbf{A})]\mathbf{vec}\left[\mathbf{X}_{JI\times K}\right]
\end{aligned}$$

ALGORITHMS FOR THE CPD

Levenberg-Marquardt

Jacobian matrix (**J**) of residual error :

$$\mathbf{J} = [\mathbf{J}_\mathbf{A}, \mathbf{J}_\mathbf{B}, \mathbf{J}_\mathbf{C}]$$

update **p**:

$$\mathbf{p}(t+1) = \mathbf{p}(t) - [\mathbf{J}(t)^\intercal \mathbf{J}(t) + \eta(t)\,\mathbf{I}]^{-1}\,\mathbf{g}_(t)$$

We have:

$$\begin{aligned} \mathbf{J}_\mathbf{A} &= \mathbf{I}_A \otimes (\mathbf{C} \odot \mathbf{B}) \\ \mathbf{J}_\mathbf{B} &= \Pi_1 [\mathbf{I}_B \otimes (\mathbf{A} \odot \mathbf{C})] \\ \mathbf{J}_\mathbf{C} &= \Pi_2 [\mathbf{I}_C \otimes (\mathbf{B} \odot \mathbf{A})] \end{aligned}$$

Where matrices Π_i are permutation matrices.

Algorithms for the CPD, Enhanced Line Search (Rajih, 2005)

Relevance

Speed up iterative algorithms, get out of local minima.

Algorithm

1. Choose a research direction Δ_p in the parameter space (for instance, $\mathbf{p}(t+1)-\mathbf{p}(t)$, $\mathbf{g}$ or $[\mathbf{J}^\mathsf{T}\mathbf{J}+\eta\,\mathbf{I}]^{-1}\,\mathbf{g}$)
2. find the optimal step μ in the direction Δ_p by minimizing polynomial:

$$||\mathbf{X}_{I,JK}-(\widehat{\mathbf{A}}+\mu\Delta_{p_A})((\widehat{\mathbf{C}}+\mu\Delta_{p_C})\odot(\widehat{\mathbf{B}}+\mu\Delta_{p_B}))^\mathsf{T}||^2$$

DIAG ALGORITHM (AT ORDER 3)

In the following we describe the real version of the algorithm but the complex version is almost identical.

- Let $\mathcal{T}$ be 3 way tensor of size $I \times J \times K$ and $\mathbf{T}$ the $I \times JK$ unfolding matrix of $\mathcal{T}$ such that:

$$\mathcal{T}_{i,j,k} = \sum_{n=N}^{P} A_{in} B_{jn} C_{kn} \quad \text{and} \quad \mathbf{T} = A(C \odot B)^{\intercal} \tag{18}$$

- Denoting $\mathbf{USV}^{\intercal}$ the SVD of $\mathbf{T}$ truncated at order P then $\exists$ $\mathbf{W}$ such that:

$$\mathbf{A} = \mathbf{UW} \ \text{ and } \ (\mathbf{C} \odot \mathbf{B})^{\intercal} = \mathbf{W}^{-1}\mathbf{SV}^{\intercal}.$$

- Let $\phi^{(1)}, \cdots, \phi^{(K)}$ be the diagonal matrices built from the rows of $\mathbf{C}$, then:

$$(\mathbf{C} \odot \mathbf{B})^{\intercal} = \left[\Phi^{(1)}\mathbf{B}^{\intercal}, \cdots, \Phi^{(K)}\mathbf{B}^{\intercal}\right] \tag{19}$$

$$\mathbf{SV}^{\intercal} = \left[\underbrace{\mathbf{W}\Phi^{(1)}\mathbf{B}^{\intercal}}_{\Gamma^{(1)\intercal}}, \cdots, \underbrace{\mathbf{W}\Phi^{(K)}\mathbf{B}^{\intercal}}_{\Gamma^{(K)\intercal}}\right] \tag{20}$$

DIAG ALGORITHM (AT ORDER 3)

JEVD rewriting

- From $\mathbf{S}\mathbf{V}^{\intercal}$ we have obtained K matrices $\Gamma^{(k)} = \mathbf{B}\Phi^{(k)}\mathbf{W}^{\intercal}$.
- Defining, $\mathbf{M}^{(k_1,k_2)} = \Gamma^{(k_1)\sharp}\Gamma^{(k_2)}$ and $\Lambda^{(k_1,k_2)} = \Phi^{(k_1)\sharp}\Phi^{(k_2)}$ where $\sharp$ denotes the Moore-Penrose matrix inverse, we have:

$$\boxed{\forall k_1 = 1\cdots K-1,\ k_2 = 2\cdots K,\ k_2 > k_1 \qquad \mathbf{M}^{(i_1,i_2)} = \mathbf{W}^{-\intercal}\Lambda^{k_1,k_2}\mathbf{W}^{\intercal}.} \tag{21}$$

Joint EigenValue Decomposition problem

- Iterative JEVD algorithm $\Rightarrow \mathbf{W}^{-\intercal}$
- $\mathbf{W}^{-\intercal} \Rightarrow \mathbf{C}\odot\mathbf{B}$ and $\mathbf{A}$ since $(\mathbf{C}\odot\mathbf{B})^{\intercal} = \mathbf{W}^{-1}\mathbf{S}\mathbf{V}^{\intercal}$ and $\mathbf{A} = \mathbf{U}\mathbf{W}$
- $(\mathbf{C}\odot\mathbf{B}) \Rightarrow \mathbf{B}$ and $\mathbf{C}$.

Necessary conditions

H1. $N \leq \min(I,J)$

H2. Matrix $\mathbf{C}$ has at least two rows whose entries are non-zero.

EXTENSION TO HO TENSORS

$$\mathcal{T}_{i_1\cdots i_Q} = \sum_{n=1}^{N} H^{(1)}_{i_1 r} H^{(2)}_{i_2 r} \cdots H^{(Q)}_{i_Q p}. \tag{22}$$

- Tensor $\mathcal{T}$ of order Q, rank P and dimensions $I_1 \times I_2 \times \cdots \times I_Q$
- Q factor matrices $\mathbf{H}^{(q)}$ of size $I_q \times N$

We define:

- numbers: $\pi_a^b = I_a I_{a+1} \cdots I_b,\ (b > a)$
- matrices: $\mathbf{Y}_{(b)}^{(a)} = \mathbf{H}^{(b)} \odot \mathbf{H}^{(b-1)} \odot \cdots \odot \mathbf{H}^{(a)},\ (b > a)$

and the unfolded matrices $\mathbf{T}(q)$ of size $\pi_1^q \times \pi_{q+1}^Q$ so that:

$$\mathbf{T}(q) = \mathbf{Y}_{(q)}^{(1)} \mathbf{Y}_{(Q)}^{(q+1)\mathsf{T}}. \tag{23}$$

Extension to HO tensors

Let $\mathbf{USV}^\mathsf{T}$ be the truncated SVD of $\mathbf{T}(q)$ at rank N, then, it exist an invertible matrix $\mathbf{W}$ of size $N \times N$ such that:

$$\mathbf{Y}_{(q)}^{(1)} = \mathbf{UW} \;\text{ and }\; \mathbf{Y}_{(Q)}^{(q+1)\mathsf{T}} = \mathbf{W}^{-1}\mathbf{SV}^\mathsf{T}. \tag{24}$$

We then define I_Q diagonal matrices $\mathbf{\Phi}^{(1)}, \cdots, \mathbf{\Phi}^{(I_Q)}$ from the I_Q rows of $\mathbf{H}^{(Q)}$.

$$\mathbf{Y}_{(Q)}^{(q+1)\mathsf{T}} = \left[\mathbf{\Phi}^{(1)}\mathbf{Y}_{(Q-1)}^{(q+1)\ \mathsf{T}}, \cdots, \mathbf{\Phi}^{(I_Q)}\mathbf{Y}_{(Q-1)}^{(q+1)\ \mathsf{T}}\right] \tag{25}$$

$$\mathbf{SV}^\mathsf{T} = \left[\underbrace{\mathbf{W\Phi}^{(1)}\mathbf{Y}_{(Q-1)}^{(q+1)\ \mathsf{T}}}_{\Gamma^{(1)\mathsf{T}}}, \cdots, \underbrace{\mathbf{W\Phi}^{(I_Q)}\mathbf{Y}_{(Q-1)}^{(q+1)\ \mathsf{T}}}_{\Gamma^{(I_Q)\mathsf{T}}}\right] \tag{26}$$

EXTENSION TO HO TENSORS

JEVD rewriting

- From $\mathbf{SV}^\intercal$ we have obtained I_Q matrices $\Gamma^{(p)} = \mathbf{Y}_{(Q-1)}^{(q+1)} \Phi^{(p)} \mathbf{W}^\intercal$.
- Let, $\mathbf{M}^{(p_1,p_2)} = \Gamma^{(p_1)\sharp}\Gamma^{(p_2)}$ and $\Lambda^{(p_1,p_2)} = \Phi^{(p_1)\sharp}\Phi^{(p_2)}$ ($\sharp$: pseudo inverse), then we have

$$\boxed{\forall p_1 = 1 \cdots I_Q - 1,\ p_2 = 2 \cdots I_Q,\ p_2 > p_1 \qquad \mathbf{M}^{(p_1,p_2)} = \mathbf{W}^{-\intercal}\Lambda^{(p_1,p_2)}\mathbf{W}^\intercal.} \tag{27}$$

JEVD of matrices $\mathbf{M}^{(p_1,p_2)}$

- Iterative JEVD algorithm $\Rightarrow \mathbf{W}^{-\intercal}$
- $\mathbf{W}^{-\intercal} \Rightarrow \mathbf{Y}_{(Q)}^{(q+1)}$ and $\mathbf{Y}_{(q)}^{(1)}$ since $\mathbf{Y}_{(Q)}^{(q+1)\intercal} = \mathbf{W}^{-1}\mathbf{SV}^\intercal$ and $\mathbf{Y}_{(q)}^{(1)} = \mathbf{UW}$
- $\mathbf{Y}_{(Q)}^{(q+1)}$ et $\mathbf{Y}_{(q)}^{(1)} \Rightarrow \mathbf{H}^{(1)} \cdots \mathbf{H}^{(Q)}$ (rank 1 HOSVDs)

Generic necessary conditions

- $\mathcal{C}_{DIAG}$: $\exists$ a modes order and $q \in [2;Q-1]_{\mathbb{N}}$ so that $N \leq \min(\pi_1^q, \pi_{q+1}^{Q-1})$
- $\mathcal{C}_{CFS}$: $\exists\, (q_1, q_2) \in [1;Q]_{\mathbb{N}}^2$, $q_1 \neq q_2$ so that $I_{q_1} \geq P$ and $I_{q_2} \geq N$
- $\mathcal{C}_{ALS}$: $\forall\, q \in [1;Q]_{\mathbb{N}}$, $\prod_{\substack{i=1 \\ i \neq q}}^{Q} I_i \geq N$

Proposition:

$$\mathcal{C}_{CFS} \Rightarrow \mathcal{C}_{DIAG} \Rightarrow \mathcal{C}_{ALS}$$

C_{DIAG} at low orders

$Q = 3.$ At least two tensor dimension is greater or equal to the CPD rank N.

$Q = 4.$ At least one tensor dimension is greater than N and at least one product of two of the remaining dimensions is also greater than N.

$Q = 5.$
- At least one tensor dimension is greater than N and at least one product of three of the remaining dimensions is also greater than N.
- Or at least one product between two tensor dimensions and another product between two of the remaining dimensions are greater than N.

CHOICE OF THE UNFOLDING MATRIX

All the mode don't play the same role in the algorithm but they can be permuted. Hence we have to choose carefully:

1. The value of q (at order 3, $q = 1$).
2. The permutation of tensor modes.

- Choosing $\mathbf{T}(q)$ so that $N << \min\left(\prod_{m=1}^{q} I_m, \prod_{m=q+1}^{Q-1} I_m\right)$ allows to relax C_{DIAG}. Usually it gives the more accurate results in noisy situations.
- When it is possible, putting the mode with the smallest dimension at the end decreases the overall complexity and thus can dramatically speed up the algorithm.

DIAG MAIN FEATURES

- A customizable algorithm (depend on the choice of the unfolding matrix and of the JEVD algorithm).
- Can be very fast.
- Can be used to initialize an iterative algorithm in order to improve the accuracy of the solution while keeping a fast result.
- Able to deal with very correlated factors.
- Can help to solve the overfactoring problem.

THE JOINT EIGENVALUE DECOMPOSITION

Problem formulation

Find a non-singular matrix $\mathbf{A} \in \mathbb{R}^{P \times P}$ from a set of non-defective matrices $\mathbf{M}^{(k)}$ so that:

$$\forall k \in [1;K]_{\mathbb{N}}, \qquad \mathbf{M}^{(k)} = \mathbf{A}\mathbf{D}^{(k)}\mathbf{A}^{-1} \tag{28}$$

where the K matrices $\mathbf{D}^{(k)} \in \mathbb{R}^{P \times P}$ are diagonal and unknown.

State of the art

All these algorithm resort to a Jacobi-like iterative procedure.

- sh-rt [Fu, 2006] based on the polar decomposition of **A**.
- JUST [Iferroudjene, 2009] based on the polar decomposition of **A**.
- JDTM [Luciani and Albera, 2010] based on the polar decomposition of **A**.
- JET (JET-U and JET-O) [Luciani and Albera, 2011 and 2015] based on the **LU** decomposition of **A**.
- JDTE [André, 2015] global estimation of **A** at each iteration.

THE JET ALGORITHM

LU decomposition

Due to the indeterminacies of the JEVD problem, the matrix $\mathbf{A}$ (56) can be chosen of the form $\mathbf{A} = \mathbf{L}\mathbf{U}$ (without any loss of generality) with:

- $\mathbf{L}$: unit lower triangular matrix (1 on the diagonal)
- $\mathbf{U}$: unit upper triangular matrix (1 on the diagonal)

Joint triangularization

Let $\mathbf{R}^{(k)}$ be given by $\mathbf{R}^{(k)} = \mathbf{U}\mathbf{D}^{(k)}\mathbf{U}^{-1}$ for any $k \in [1;K]_{\mathbb{N}}$.

- Joint triangularization of the K matrices $\mathbf{M}^{(k)}$ by $\mathbf{L}$:

$$\forall k \in [1;K]_{\mathbb{N}}, \qquad \mathbf{M}^{(k)} = \mathbf{L}\,\mathbf{R}^{(k)}\,\mathbf{L}^{-1}$$

- Direct computation of the unit upper triangular matrix $\mathbf{U}$ from the set of matrices $\mathbf{R}^{(k)}$ (component by component).

The JET algorithm

elementary lower triangular matrix

An elementary lower triangular matrix $\mathbf{L}^{(i,j)}(a)$ is a unit lower triangular matrix with only one non-null off-diagonal component a located at the i-th row and the j-th column.

LU factorization

Any unit lower triangular matrix $\mathbf{L}$ of size $(N \times N)$ can be factorized as a product of $M = N(N-1)/2$ elementary lower triangular matrices:

$$\mathbf{L} = \prod_{j=1}^{N-1} \prod_{i=j+1}^{N} \mathbf{L}^{(i,j)}(\ell_{i,j})$$

Jacobi-like procedure

Repeat several times a series (called "sweep") of M sequential optimizations with respect to only one parameter.

THE JET ALGORITHM

Update of the matrices to be triangularized

$$\forall(i,j) \in [1;N]_{\mathbb{N}}^2,\ i>j,\ \forall k \in [1;K]_{\mathbb{N}},\ \mathbf{M}^{(k)} \leftarrow \left(\mathbf{L}^{(ij)}(x_{i,j})\right)^{-1} \mathbf{M}^{(k)} \mathbf{L}^{(ij)}(x_{i,j}) \quad (29)$$

- Each of these updates only depends on one parameter $x_{i,j}$;
- Each parameter $x_{i,j}$ is computed in order to sequentially improve the upper triangular structure of the $\mathbf{L}^{(i,j)}(x_{i,j})$-updated matrices.

Objective functions

$$\zeta_O(x_{i,j}) = \sum_{k=1}^{K} \sum_{q=1}^{N-1} \sum_{p=q+1}^{N} \left(M_{p,q}^{(k)}\right)^2 \quad (30)$$

$$\zeta_U(x_{i,j}) = \sum_{k=1}^{K} \left(M_{i,j}^{(k)}\right)^2 \quad (31)$$

THE JDTM ALGORITHM

Joint Diagonalization algorithm based on Targeting hyperbolic Matrices

Polar decomposition

$$\mathbf{A} = \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \mathbf{G}(\theta_{ij})\mathbf{H}(\phi_{ij}). \tag{32}$$

$$\mathbf{G}(\theta_{ij}) = \begin{array}{cccccccc} 1 & & i & & j & & P & \\ 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 1 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots & \\ 1 & \cdots & \cos(\theta_{ij}) & \cdots & \sin(\theta_{ij}) & \cdots & 0 & i \\ \vdots & \ddots & \vdots & & \vdots & & \vdots & \\ 1 & \cdots & -\sin(\theta_{ij}) & \cdots & \cos(\theta_{ij}) & \cdots & 0 & j \\ \vdots & \ddots & \vdots & & \vdots & & \vdots & \\ 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & P \end{array}$$

The JDTM algorithm

Joint Diagonalization algorithm based on Targeting hyperbolic Matrices

Polar decomposition

$$\mathbf{A} = \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} \mathbf{G}(\theta_{ij})\mathbf{H}(\phi_{ij}). \tag{32}$$

$$\mathbf{H}(\phi_{ij}) = \begin{array}{cc} \begin{array}{ccccccc} 1 & & i & & j & & P \end{array} & \\ \left(\begin{array}{ccccccc} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 1 & \cdots & \cosh(\phi_{ij}) & \cdots & \sinh(\phi_{ij}) & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 1 & \cdots & \sinh(\phi_{ij}) & \cdots & \cosh(\phi_{ij}) & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \end{array}\right) & \begin{array}{c} 1 \\ \\ i \\ \\ j \\ \\ P \end{array} \end{array}$$

we look for the optimal parameter (θ_{ij}, ϕ_{ij}) for each pair (i, j) successively.

THE JDTM ALGORITHM

$$\mathbf{D}^{(k)} = \left(\prod_{i=1}^{N-1}\prod_{j=i+1}^{N}\mathbf{H}(\phi_{ij})^{-1}\mathbf{G}(\theta_{ij})^{\intercal}\right)\mathbf{M}^{(k)}\left(\prod_{i=1}^{N-1}\prod_{j=i+1}^{N}\mathbf{G}(\theta_{ij})\mathbf{H}(\phi_{ij})\right), \tag{33}$$

Updating procedure

We perform successively the following updates for each $(i,j), i<j$:

$$\forall k = 1\cdots K \qquad \mathbf{N}^{(k)} \quad \leftarrow \quad \mathbf{G}(\theta_{ij})^{\intercal}\mathbf{M}^{(k)}\mathbf{G}(\theta_{ij}), \tag{34}$$

$$\forall k = 1\cdots K \qquad \mathbf{M}^{(k)} \quad \leftarrow \quad \mathbf{H}(\phi_{ij})^{-1}\mathbf{N}^{(k)}\mathbf{H}(\phi_{ij}). \tag{35}$$

And the whole process (sweep) is repeated iteratively till convergence.

Originality of JDTM

Resort to a simplified cost function for **H** that target two particular terms:

$$\zeta_G(\theta_{ij}) = \sum_{k}^{K}\sum_{\substack{p,q\\p\neq q}}^{N,N}\left(N_{pq}^{(k)}\right)^2, \quad \zeta_H^{JDTM}(\phi_{ij}) = \sum_{k}^{K}\left(M_{ij}^{(k)}\right)^2+\left(M_{ji}^{(k)}\right)^2. \tag{36}$$

JDTM versus JET

Real case

- JDTM : Good accuracy for the estimation of **A** especially in difficult situations, require very few sweep to reach the convergence, very versatile algorithm.
- JET-U : (by far) the less costly algorithm in all situations, very efficient in the most simple cases.
- JET-O : Usually the most costly but also the most accurate.

Complex case

- All algorithms were extended to the complex case. JET can be extended in a very natural way to $\mathbb{C}$ while JDTM complex version is much more tricky.
- For low SNR values: JET-U and JET-O are more efficient than JDTM for the estimation of **A**.
- Conversely, for high SNR values JDTM should be prefered.
- JDTM requires much more sweep to converge.

Outline

4 Solving the overfactoring problem with DIAG: application to fluorescence spectroscopy

Overfactoring : Definition and example

- In practice we have: $\mathcal{T}_{i,j,k} = \sum_{n=1}^{N} A_{i,n} B_{j,n} C_{k,n} + \mathcal{E}_{i,j,k}$
- Actual tensor rank is greater than N (physical rank), which is unknown and generally overestimated (overfactoring), 2 possibilities then:

Overfactoring : Definition and example

- In practice we have: $\mathcal{T}_{i,j,k} = \sum_{n=1}^{N} A_{i,n} B_{j,n} C_{k,n} + \mathcal{E}_{i,j,k}$
- Actual tensor rank is greater than N (physical rank), which is unknown and generally overestimated (overfactoring), 2 possibilities then:
  1. **extra-factors have no influence on the estimation of the real factors and their overall contribution can be neglected $\Rightarrow$ No problem**.

Example:

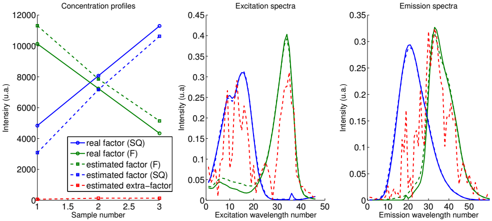


Figure : No problem of overfactoring.

OVERFACTORING : DEFINITION AND EXAMPLE

- In practice we have: $\mathcal{T}_{i,j,k} = \sum_{n=1}^{N} A_{i,n} B_{j,n} C_{k,n} + \mathcal{E}_{i,j,k}$
- Actual tensor rank is greater than N (physical rank), which is unknown and generally overestimated (overfactoring), 2 possibilities then:
  2. **extra-factors impact the estimation of the real factors and cannot be neglected $\Rightarrow$ pb to choose the appropriate model**.

Example:

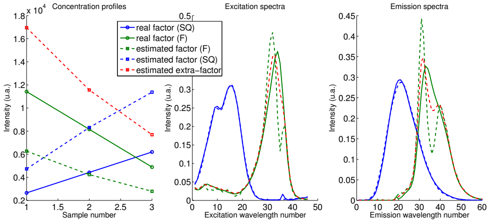
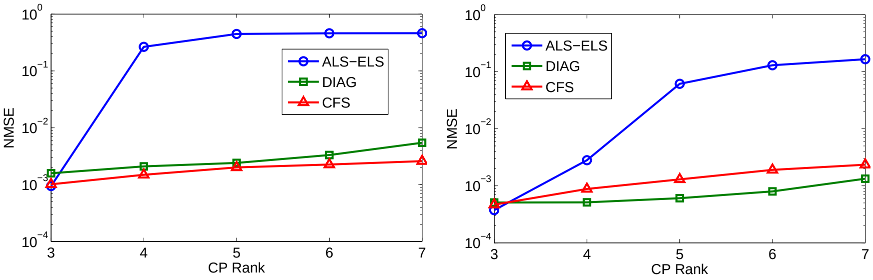


Figure : Overfactoring problem (ALS).

COMPARATIVE RESULTS 1

Synthetic tensors



(a) $7 \times 7 \times 7$ tensors of physical rank 3

(b) $7 \times 7 \times 7 \times 7$ tensors of physical rank 3

Figure : Factors estimation error vs model rank

Comparative results 2

Fluorescence tensor of physical rank 2

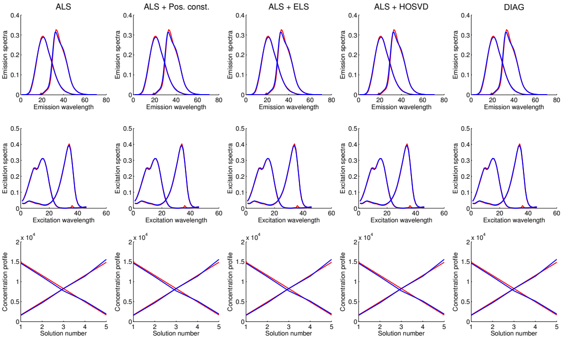
- 5 Mixtures of quinine sulphate and fluorescein, $5 \times 46 \times 61$ tensor
- We compare various CP algorithms for model rank 2 to 5

Comparative results 2

Fluorescence tensor of physical rank 2

- 5 Mixtures of quinine sulphate and fluorescein, $5 \times 46 \times 61$ tensor
- We compare various CP algorithms for model rank 2 to 5

Results for model rank 2, Red : actual factors, Blue : estimated factors

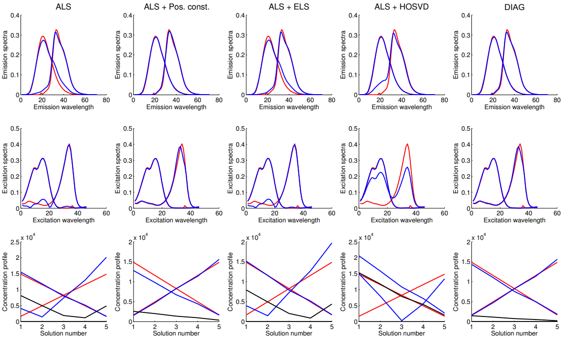


COMPARATIVE RESULTS 2

Fluorescence tensor of physical rank 2

- 5 Mixtures of quinine sulphate and fluorescein, 5 × 46 × 61 tensor
- We compare various CP algorithms for model rank 2 to 5

Results for model rank 3, Red : actual factors, Blue : estimated factors

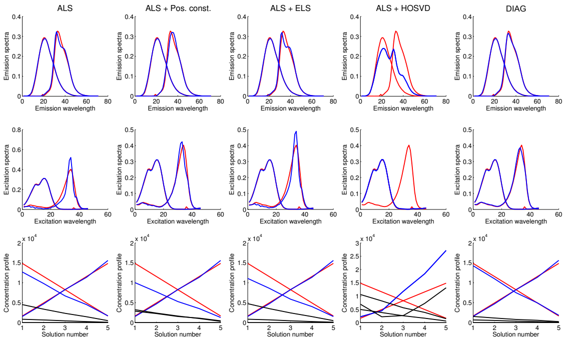


Comparative results 2

Fluorescence tensor of physical rank 2

- 5 Mixtures of quinine sulphate and fluorescein, $5 \times 46 \times 61$ tensor
- We compare various CP algorithms for model rank 2 to 5

Results for model rank 4, Red : actual factors, Blue : estimated factors

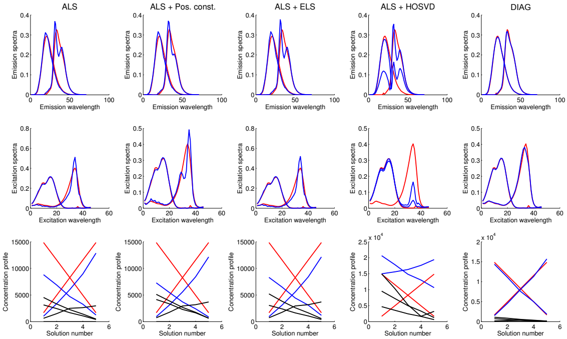


Comparative results 2

Fluorescence tensor of physical rank 2

- 5 Mixtures of quinine sulphate and fluorescein, $5 \times 46 \times 61$ tensor
- We compare various CP algorithms for model rank 2 to 5

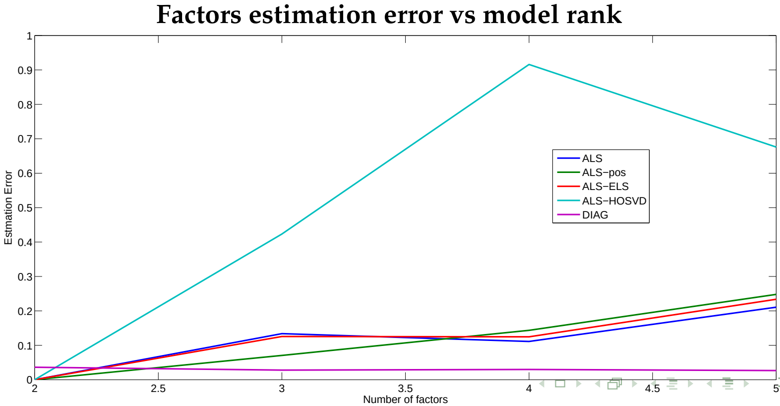
Results for model rank 5, Red : actual factors, Blue : estimated factors



Comparative results 2

Fluorescence tensor of physical rank 2

- 5 Mixtures of quinine sulphate and fluorescein, $5 \times 46 \times 61$ tensor
- We compare various CP algorithms for model rank 2 to 5



Comparative results 3

Fluorescence tensor of physical rank 3

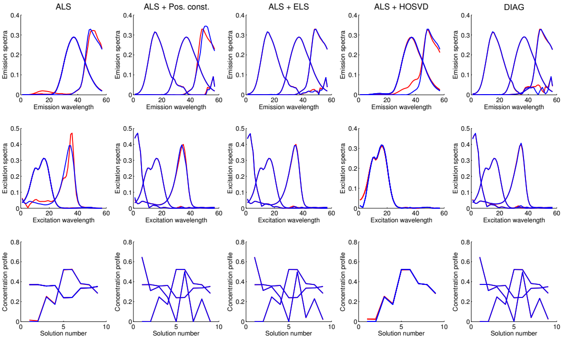
- 9 Mixtures of QS, fluorescein and tryptophan, $9 \times 57 \times 57$ tensor
- We compare various CP algorithms for model rank 4 to 6

Comparative results 3

Fluorescence tensor of physical rank 3

- 9 Mixtures of QS, fluorescein and tryptophan, $9 \times 57 \times 57$ tensor
- We compare various CP algorithms for model rank 4 to 6

Results for model rank 4, Red : actual factors, Blue : estimated factors

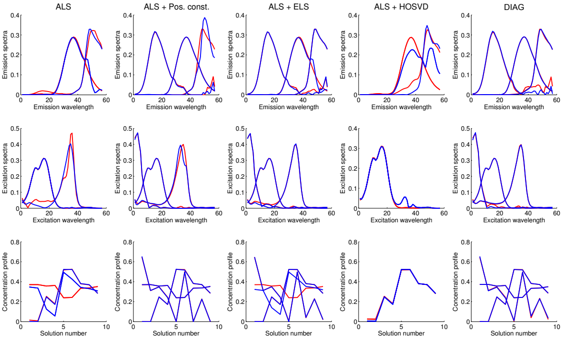


Comparative results 3

Fluorescence tensor of physical rank 3

- 9 Mixtures of QS, fluorescein and tryptophan, 9 × 57 × 57 tensor
- We compare various CP algorithms for model rank 4 to 6

Results for model rank 5, Red : actual factors, Blue : estimated factors

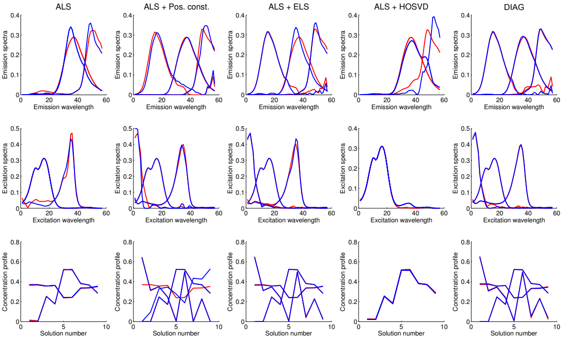


Comparative results 3

Fluorescence tensor of physical rank 3

- 9 Mixtures of QS, fluorescein and tryptophan, $9 \times 57 \times 57$ tensor
- We compare various CP algorithms for model rank 4 to 6

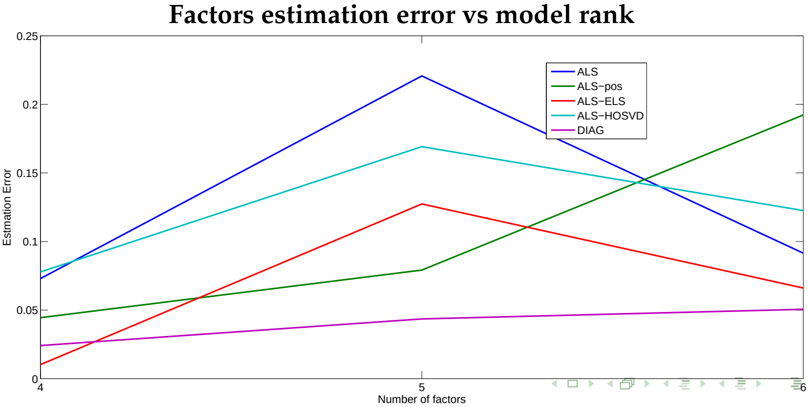
Results for model rank 6, Red : actual factors, Blue : estimated factors



COMPARATIVE RESULTS 3

Fluorescence tensor of physical rank 3

- 9 Mixtures of QS, fluorescein and tryptophan, $9 \times 57 \times 57$ tensor
- We compare various CP algorithms for model rank 4 to 6



CONCLUSION

Tensor decompositions for BBS and BI

- 2 kinds of approaches according the structure of the data.
- Tensor modelling of the data (direct problem) combined with a tensor decomposition algorithm (inverse problem).
- Some applications leads the TD algorithm choice...
- ...but existing TD and algorithms can also leads to new applications !

Algorithms

- The CPD can be rewritten as a JEVD problem ⇒ DIAG algorithm.
- This approach is fast and can also be used to initialize iterative algorithm.
- It allows to deal with very correlated factors and to deal with overfactoring.
- Future improvement: including positivity constraint, relax the necessary working conditions, improve speed and noise robustness => PhD thesis of Rémi André started in November 2014 directed by Eric Moreau.