Nonnegative Tensor Factorization using a proximal algorithm: application to 3D fluorescence spectroscopy

Caroline Chaux

Joint work with X. Vu, N. Thirion-Moreau and S. Maire (LSIS, Toulon)

Aix-Marseille Univ. I2M

IHP, Feb. 2 2017

Outline

Introduction
- 3D fluorescence spectroscopy
- Tensor definition
- Goal

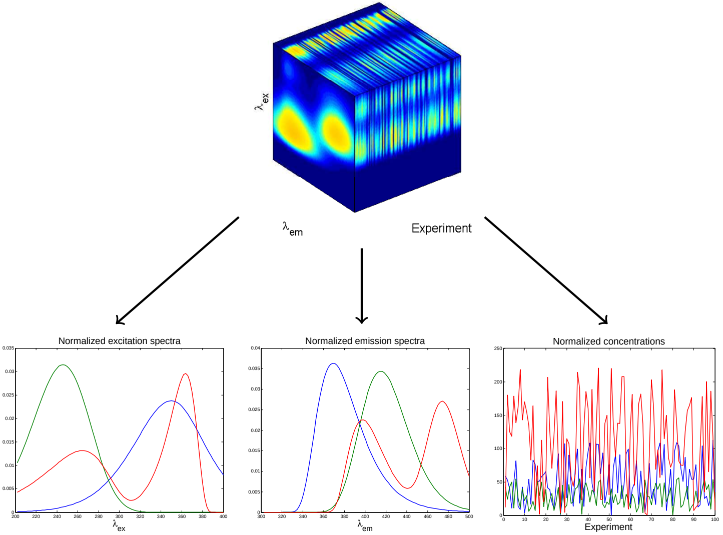
Proximal tools
- Criterion formulation
- Proximity operator
- Proximal algorithm

Application to CPD
- Minimization problem
- Algorithm

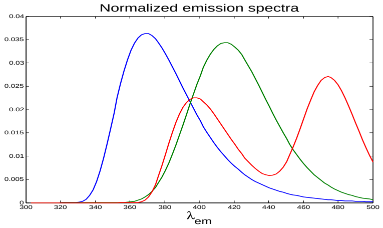
Numerical simulations
- Synthetic case
- Real case: water monitoring

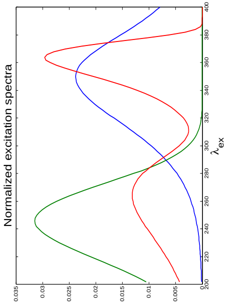
Conclusion and future work

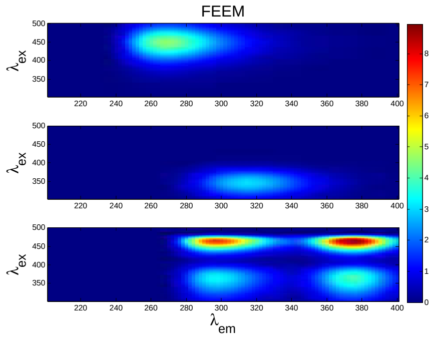
3D fluorescence spectroscopy



Coumpounds characterisation







Tensor

What is a tensor?

An Nth-order tensor is represented by an N-way array in a chosen basis.

Example:

- $N = 1$: a vector.
- $N = 2$: a matrix.

Third-order tensors

- A special case: nonnegative third-order tensors ($N = 3$)

$$\overline{\mathcal{T}} = (\bar{t}_{i_1 i_2 i_3})_{i_1, i_2, i_3} \in \mathbb{R}^{+I_1 \times I_2 \times I_3}.$$

- The Canonical Polyadic (CP) decomposition:

Tensor rank

$$\overline{\mathcal{T}} = \sum_{r=1}^{\overline{R}} \bar{\mathbf{a}}_r^{(1)} \circ \bar{\mathbf{a}}_r^{(2)} \circ \bar{\mathbf{a}}_r^{(3)} = [\![\bar{\mathbf{A}}^{(1)}, \bar{\mathbf{A}}^{(2)}, \bar{\mathbf{A}}^{(3)}]\!]$$

Loading vectors

Loading matrices

$\forall n \in \{1, 2, 3\}$, $\bar{\mathbf{a}}_r^{(n)} \in \mathbb{R}^{+I_n}$ and $\bar{\mathbf{A}}^{(n)} \in \mathbb{R}^{I_n \times \overline{R}}$

$\circ$: the outer product.

- Entry-wise form:

$$\bar{t}_{i_1 i_2 i_3} = \sum_{r=1}^{\overline{R}} \bar{a}_{i_1 r}^{(1)} \bar{a}_{i_2 r}^{(2)} \bar{a}_{i_3 r}^{(3)}, \quad \forall (i_1, i_2, i_3)$$

Standard operations

- Outer product: let $\mathbf{u} \in \mathbb{R}^I, \mathbf{v} \in \mathbb{R}^J$,

$$\boxed{\mathbf{u} \circ \mathbf{v} = \mathbf{u}\mathbf{v}^\top} \in \mathbb{R}^{I \times J}$$

- Khatri-Rao product: let $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_J] \in \mathbb{R}^{I \times J}$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_J] \in \mathbb{R}^{K \times J}$

$$\boxed{\mathbf{U} \odot \mathbf{V} = [\mathbf{u}_1 \otimes \mathbf{v}_1, \mathbf{u}_2 \otimes \mathbf{v}_2, \ldots, \mathbf{u}_J \otimes \mathbf{v}_J]} \in \mathbb{R}^{IK \times J}.$$

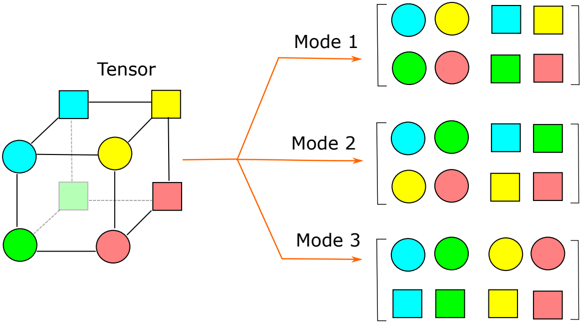
where $\mathbf{u} \otimes \mathbf{v} = [u_1\mathbf{v}; \ldots; u_I\mathbf{v}] \in \mathbb{R}^{IK}$ (Kronecker product).

- Hadamard division: let $\mathbf{U} \in \mathbb{R}^{I \times J}, \mathbf{V} \in \mathbb{R}^{I \times J}$,

$$\boxed{\mathbf{U} \oslash \mathbf{V} = (u_{ij}/v_{ij})_{i,j}} \in \mathbb{R}^{I \times J}$$

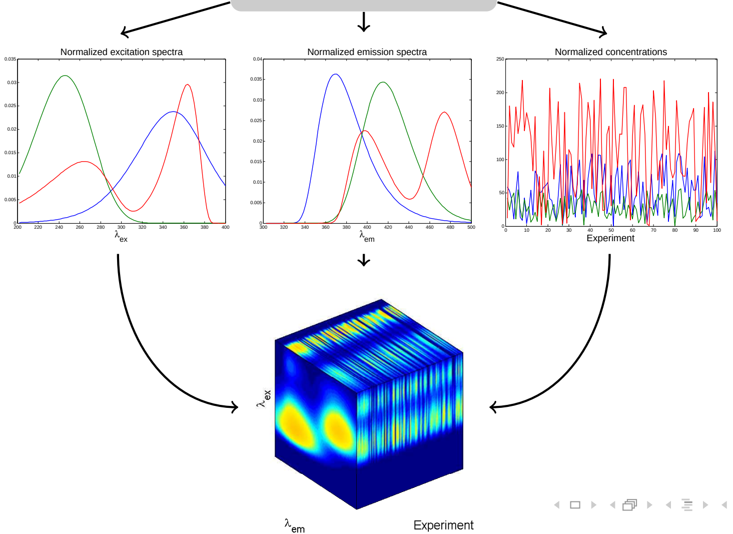
Tensor flattening: example

Objective: to handle matrices instead of tensors.



3D fluorescence spectroscopy and tensors

$$\overline{\mathcal{T}} = \sum_{r=1}^{\overline{R}} \bar{\mathbf{a}}_r^{(1)} \circ \bar{\mathbf{a}}_r^{(2)} \circ \bar{\mathbf{a}}_r^{(3)}$$



Objective: tensor decomposition

- Input:
  - Observed tensor $\mathcal{T}$: observation of an original (unknown) tensor $\overline{\mathcal{T}}$ possibly degraded (noise).
- Output:
  - Estimated loading matrices $\widehat{\mathbf{A}}^{(n)}$ for all $n \in \{1, 2, 3\}$
- Difficulty:
  - Rank $\overline{R}$ unknown (*i.e.* $\widehat{R} \neq \overline{R}$): generally i) estimated or ii) overestimated.

Proposed approach

Formulate the problem under a variational approach.

Minimization problem

- Standard problem:

$$\underset{\mathbf{x}\in\mathbb{R}^L}{\text{minimize}} \quad \underbrace{\mathcal{F}(\mathbf{x})}_{\text{Fidelity}} \quad + \quad \underbrace{\mathcal{R}(\mathbf{x})}_{\text{Regularization}} \quad .$$

- Taking into account several regularizations (J terms):

$$\mathcal{R}(\mathbf{x}) = \sum_{j=1}^{J} \mathcal{R}_j(\mathbf{x})$$

- For large size problem or for other reasons, can be interesting to work on data blocks $\mathbf{x}^{(j)}$ of size L_j ($\mathbf{x} = (\mathbf{x}^{(j)})_{1\leq j\leq J}$)

$$\mathcal{R}(\mathbf{x}) = \sum_{j=1}^{J} \mathcal{R}_j(\mathbf{x}^{(j)})$$

Technical assumptions: $\mathcal{F}$, $\mathcal{R}$ and $\mathcal{R}_j$ are proper lower semi-continuous functions. $\mathcal{F}$ is differentiable with a β-Lipschitz gradient. $\mathcal{R}_j$ is assumed to be bounded from below by an affine function, and its restriction to its domain is continuous.

Proximity operator

- let $\varphi : \mathbb{R} \to]-\infty, +\infty]$ be a proper lower semi-continuous function. The proximity operator is defined as

$$\boxed{\operatorname{prox}_{\varphi} : \mathbb{R} \to \mathbb{R} : v \mapsto \arg\min_{u \in \mathbb{R}} \frac{1}{2} \|u - v\|^2 + \varphi(u),}$$

- let $\varphi : \mathbb{R}^L \to]-\infty, +\infty]$ be a proper lower semi-continuous function. The proximity operator associated with a Symmetric Positive Definite (SPD) matrix $\mathbf{P}$ is defined as

$$\boxed{\operatorname{prox}_{\mathbf{P},\varphi} : \mathbb{R}^L \to \mathbb{R}^L : \mathbf{v} \mapsto \arg\min_{\mathbf{u} \in \mathbb{R}^L} \frac{1}{2} \|\mathbf{u} - \mathbf{v}\|_{\mathbf{P}}^2 + \varphi(\mathbf{u}),}$$

where $\forall \mathbf{x} \in \mathbb{R}^L$, $\|\mathbf{x}\|_{\mathbf{P}}^2 = \langle \mathbf{x}, \mathbf{P}\mathbf{x} \rangle$ and $\langle \cdot, \cdot \rangle$ is the inner product.

Remark : Note that if $\mathbf{P}$ reduces to the identity matrix, then the two definitions coincides.

Criterion to be minimized

$$\boxed{\underset{\mathbf{x}\in\mathbb{R}^L}{\text{minimize}} \quad \mathcal{F}(\mathbf{x}) \quad + \sum_{j=1}^{J} \mathcal{R}_j(\mathbf{x}^{(j)})}$$

Some solutions (non exhaustive list, CPD oriented):

- Proximal Alternating Linearized Minimization (PALM) [Bolte et al., 2014]
- A Block Coordinate Descent Method for both CPD and Tucker decomposition [Xu and Yin, 2013]
- An accelerated projection gradient based algorithm [Zhang et al., 2016]
- Block-Coordinate Variable Metric Forward-Backward (BC-VMFB) algorithm [Chouzenoux et al., 2016]

Advantages of the BC-VMFB: flexible, stable, integrates preconditionning, relatively fast.

Block coordinate proximal algorithm

```
Let $\mathbf{x}_0 \in \mathrm{dom}\mathcal{R}$, $k \in \mathbb{N}$ and $\gamma_k \in ]0, +\infty[$     // Initialization step
for $k = 0, 1, \ldots$ do     // k-th iteration of the algorithm
    Let $j_k \in \{1, \ldots, J\}$     // Processing of block number $j_k$ (chosen, here, according to a quasi cyclic rule)
    Let $\mathbf{P}_{j_k}(\mathbf{x}_k)$ be a SPD matrix     // Construction of the preconditioner $\mathbf{P}_{j_k}(\mathbf{x}_k)$
    Let $\nabla_{j_k}\mathcal{F}(\mathbf{x}_k)$ be the Gradient     // Calculation of Gradient
    $\tilde{\mathbf{x}}_k^{(j_k)} = \mathbf{x}_k^{(j_k)} - \gamma_k \mathbf{P}_{j_k}(\mathbf{x}_k)^{-1} \nabla_{j_k}\mathcal{F}(\mathbf{x}_k)$     // Updating of block $j_k$ according to a Gradient step
    $\mathbf{x}_{k+1}^{(j_k)} \in \mathrm{prox}_{\gamma_k^{-1}\mathbf{P}_{j_k}(\mathbf{x}_k), \mathcal{R}_{j_k}}\left(\tilde{\mathbf{x}}_k^{(j_k)}\right)$     // Updating of block $j_k$ according to a Proximal step
    $\mathbf{x}_{k+1}^{\bar{j}_k} = \mathbf{x}_k^{\bar{j}_k}$ where $\bar{j} = \{1, \ldots, J\} \setminus \{j\}$     // Other blocks are kept unchanged
end for
```

Prox for CP decomposition

CP decomposition: decompose a tensor into a (minimal) sum of rank-1 terms.
Order 3:

$$\overline{\mathcal{T}} = \sum_{r=1}^{\overline{R}} \bar{\mathbf{a}}_r^{(1)} \circ \bar{\mathbf{a}}_r^{(2)} \circ \bar{\mathbf{a}}_r^{(3)} = [\![\bar{\mathbf{A}}^{(1)}, \bar{\mathbf{A}}^{(2)}, \bar{\mathbf{A}}^{(3)}]\!], \tag{1}$$

Tensor structure: naturally leads to consider 3 blocks corresponding to the loading matrices $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$ and $\mathbf{A}^{(3)}$.

Proposed optimization problem

$$\underset{\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R},\, n \in \{1,2,3\}}{\text{minimize}} \quad \mathcal{F}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}) + \mathcal{R}_1(\mathbf{A}^{(1)}) + \mathcal{R}_2(\mathbf{A}^{(2)}) + \mathcal{R}_3(\mathbf{A}^{(3)}).$$

Some of the fastest classical approaches: Fast HALS [Phan et al., 2013] and N-Way [Bro, 1997].

Tensor matricization

- $\overline{\mathbf{T}}^{(n)}_{I_n,I_{-n}} \in \mathbb{R}_+^{I_n \times I_{-n}}$ the matrix obtained by unfolding the tensor $\overline{\mathcal{T}}$ in the n-th mode where the size I_{-n} is equal to $I_1 I_2 I_3 / I_n$
- Tensor expressed under matrix form as

$$\boxed{\overline{\mathbf{T}}^{(n)}_{I_n,I_{-n}} = \bar{\mathbf{A}}^{(n)} (\overline{\mathbf{Z}}^{(-n)})^\top, \quad n \in \{1, 2, 3\}}$$

where

$$\begin{aligned}
\overline{\mathbf{Z}}^{(-1)} &= \bar{\mathbf{A}}^{(3)} \odot \bar{\mathbf{A}}^{(2)} \in \mathbb{R}_+^{I_{-1} \times \overline{R}}, \\
\overline{\mathbf{Z}}^{(-2)} &= \bar{\mathbf{A}}^{(3)} \odot \bar{\mathbf{A}}^{(1)} \in \mathbb{R}_+^{I_{-2} \times \overline{R}}, \\
\overline{\mathbf{Z}}^{(-3)} &= \bar{\mathbf{A}}^{(2)} \odot \bar{\mathbf{A}}^{(1)} \in \mathbb{R}_+^{I_{-3} \times \overline{R}},
\end{aligned}$$

Function choice

- $\mathcal{F}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)})$: quadratic data fidelity term

$$\mathcal{F}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}) = \frac{1}{2}\|\mathcal{T} - [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}]\!]\|_F^2 = \frac{1}{2}\|\mathbf{T}^{(n)}_{I_n, I_{-n}} - \mathbf{A}^{(n)}\mathbf{Z}^{(-n)^\top}\|_F^2$$

- $\mathcal{R}_n(\mathbf{A}^{(n)})$: block dependent penalty terms enforcing sparsity and nonnegativity

$$\mathcal{R}_n(\mathbf{A}^{(n)}) = \sum_{i_n=1}^{I_n}\sum_{r=1}^{R} \rho_n(a^{(n)}_{i_n r}) \qquad \forall n \in \{1, 2, 3\}$$

where loading matrices are defined element wise as $\mathbf{A}^{(n)} = (a^{(n)}_{i_n r})_{(i_n, r) \in \{1,\ldots,I_n\} \times \{1,\ldots,R\}}$ and

$$\rho_n(\omega) = \begin{cases} \alpha^{(n)}|\omega|^{\pi^{(n)}} & \text{if } \eta^{(n)}_{\min} \leq \omega \leq \eta^{(n)}_{\max} \\ +\infty & \text{otherwise} \end{cases}$$

$\alpha^{(n)} \in]0, +\infty[$, $\pi^{(n)} \in \mathbb{N}^*$, $\eta^{(n)}_{\min} \in [-\infty, +\infty[$ and $\eta^{(n)}_{\max} \in [\eta^{(n)}_{\min}, +\infty]$.
$\Rightarrow$ block dependent but constant within a block regularization parameters.

Preconditionning

Preconditionning similar to the one used in NMF [Lee and Seung, 2001].
The matrix **P** for the n-th block can be defined as follows $\forall n \in \{1, 2, 3\}$

$$\boxed{\mathbf{P}^{(n)}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}) = \mathbf{A}^{(n)}(\mathbf{Z}^{(-n)^{\top}}\mathbf{Z}^{(-n)}) \oslash \mathbf{A}^{(n)}},$$

Remark: $\forall n \in \{1, 2, 3\}$, $\mathbf{A}^{(n)}$ must be non zero.

Gradient and proximity operator

- Gradient matrices of $\mathcal{F}$ with respect to $\mathbf{A}^{(n)}$ for all $n = 1, \ldots, 3$, defined as

$$\boxed{\nabla_n \mathcal{F}(\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}) = -(\mathbf{T}^{(n)}_{I_n, I_{-n}} - \mathbf{A}^{(n)} \mathbf{Z}^{(-n)^\top}) \mathbf{Z}^{(-n)}.}$$

- Proximity operator given by $(\forall y = (y^{(i)})_{i \in \{1,\ldots,RI_n\}} \in \mathbb{R}^{RI_n}))$

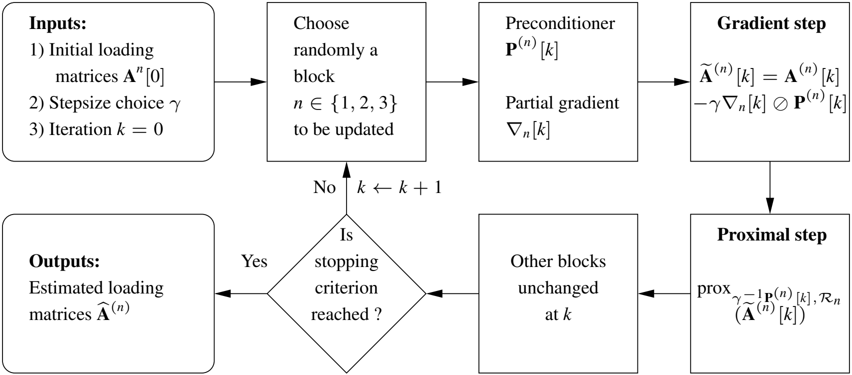
$$\operatorname{prox}_{\gamma[k]^{-1}\mathbf{P}^{(n)}[k], \mathcal{R}_n}(y) = \left(\operatorname{prox}_{\gamma[k]^{-1} p_i^{(n)}[k], \rho_n}(y^{(i)})\right)_{i \in \{1,\ldots,RI_n\}}.$$

where $\forall i \in \{1, \ldots, RI_n\}$, we have $(\forall \upsilon \in \mathbb{R})$

$$\boxed{\operatorname{prox}_{\gamma[k]^{-1} p_i^{(n)}, \rho_n}(\upsilon) = \min\left\{\eta^{(n)}_{\max}, \max\left\{\eta^{(n)}_{\min}, \operatorname{prox}_{\gamma[k]\alpha^{(n)}(p_i^{(n)}[k])^{-1} |\,.\,|^{\pi^{(n)}}}(\upsilon)\right\}\right\}}$$

(separable structure, diagonal preconditionning matrices, componentwise calculation)

Proximal algorithm for tensor decomposition



Inputs:
1) Initial loading matrices $\mathbf{A}^{n}[0]$
2) Stepsize choice γ
3) Iteration $k = 0$

Choose randomly a block $n \in \{1, 2, 3\}$ to be updated

Preconditioner $\mathbf{P}^{(n)}[k]$

Partial gradient $\nabla_n[k]$

Gradient step

$\widetilde{\mathbf{A}}^{(n)}[k] = \mathbf{A}^{(n)}[k]$
$-\gamma \nabla_n[k] \oslash \mathbf{P}^{(n)}[k]$

Proximal step

$\operatorname{prox}_{\gamma^{-1}\mathbf{P}^{(n)}[k], \mathcal{R}_n}(\widetilde{\mathbf{A}}^{(n)}[k])$

Other blocks unchanged at k

Is stopping criterion reached ?

No $k \leftarrow k+1$

Yes

Outputs:
Estimated loading matrices $\widehat{\mathbf{A}}^{(n)}$

Figure: BC-VMFB algorithm for CPD.

Computer simulation: simulated spectroscopy-like data

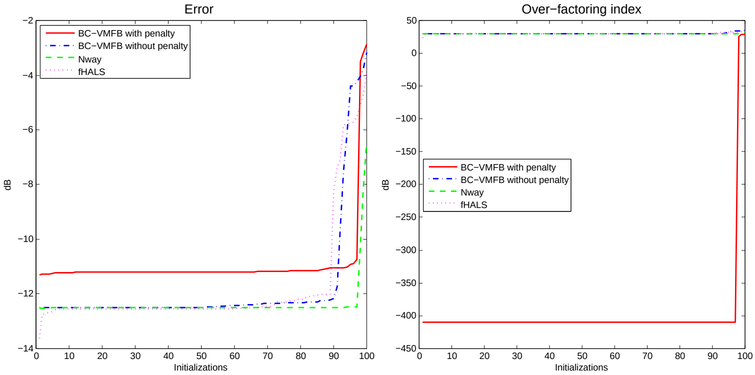
- Simulated tensor: (uni or bimodal type) emission and excitation spectra, random concentrations $\Rightarrow \overline{\mathcal{T}} \in \mathbb{R}_{+}^{100\times 100\times 100}$ and $\overline{R} = 5$.
- Simulated observed tensor: $\mathcal{T} = \overline{\mathcal{T}} + \mathcal{B}$ where $\mathcal{B}$ stands for an additive white Gaussian noise
- 2 considered cases :
  1. Perturbed case (noiseless): no noise added and $\widehat{R} = 6$ (overestimation).
  2. Perturbed case (noisy): $\mathcal{B}$ fixed such that $\mathsf{SNR} = 17.6$ dB and $\widehat{R} = 6$ (overestimation).
- Error measures
  1. Signal to Noise Ratio defined as $\mathsf{SNR} = 20\log_{10} \dfrac{\|\overline{\mathcal{T}}\|_F}{\|\widehat{\mathcal{T}} - \overline{\mathcal{T}}\|_F}$
  2. Relative Reconstruction Error defined as $\mathsf{RRE} = 20\log_{10} \dfrac{\|\widehat{\mathcal{T}} - \overline{\mathcal{T}}\|_1}{\|\overline{\mathcal{T}}\|_1}$
  3. Estimation error: $\mathbf{E}_1 = 10\log_{10}\left(\dfrac{\sum_{n=1}^{3} \|\widehat{\mathbf{A}}^{(n)}(1:\overline{R}) - \bar{\mathbf{A}}^{(n)}\|_1}{\sum_{n=1}^{3} \|\bar{\mathbf{A}}^{(n)}\|_1}\right)$
  4. Over-factoring error: $\mathbf{E}_2 = 10\log_{10}\left(\|\sum_{r=\overline{R}+1}^{\widehat{R}} \widehat{\mathbf{a}}_r^{(1)} \circ \widehat{\mathbf{a}}_r^{(2)} \circ \widehat{\mathbf{a}}_r^{(3)}\|_1\right)$

Numerical results

	Elapsed time (s)	BC-VMFB without penalty	BC-VMFB with penalty	N-way	fast HALS
Noisy case	For 50 iterations	0.2	0.2	11	0.5
	To reach stopping conditions	102	75	8	8
	(actual number of iterations)	(48500)	(36500)	(43)	(1856)
	(SNR,$\mathbf{E}_1$, $\mathbf{E}_2$) dB	(31.3, -12.5, 30.6)	(32.7, -11.2, -409)	(31.3, -12.5, 30.6)	(31.3, -12.5, 30.6)
Noiseless case	To reach stopping conditions	202	74	80	3.7
	(actual number of iterations)	(100000)	(36500)	(838)	(308)
	(RRE,$\mathbf{E}_1$, $\mathbf{E}_2$) dB	(-75.1,-12.4,25.6)	(-44.7, -15, -409)	(-127.9,-8.7, 31.7)	(-63.9, -6.1, 31.7)

Computation time comparison of BC-VMFB in two cases: with or without penalty, with N-way [Bro, 1997] and fast HALS [Phan et al., 2013] using the same initial value in the noiseless and noisy cases.

Influence of the initialization



Performance versus different initializations (noisy, overestimated case): error index $\mathbf{E}_1$, overfactoring error index $\mathbf{E}_2$

Visual results: noiseless case

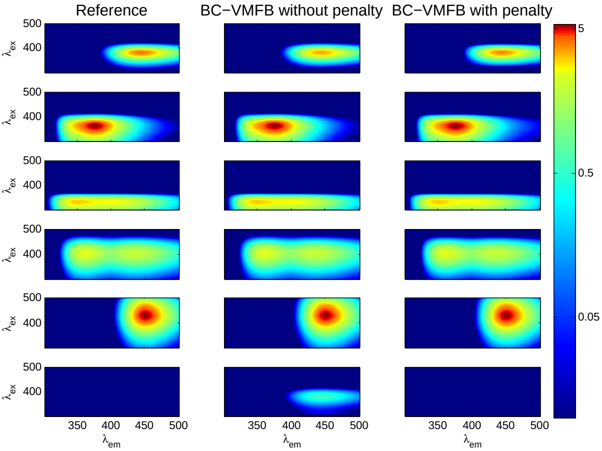


Figure: FEEM of reference (left) - FEEM reconstructed using BC-VMFB without regularization (middle) and with regularization $\alpha = 0.05$ (right).

Visual results: noiseless case

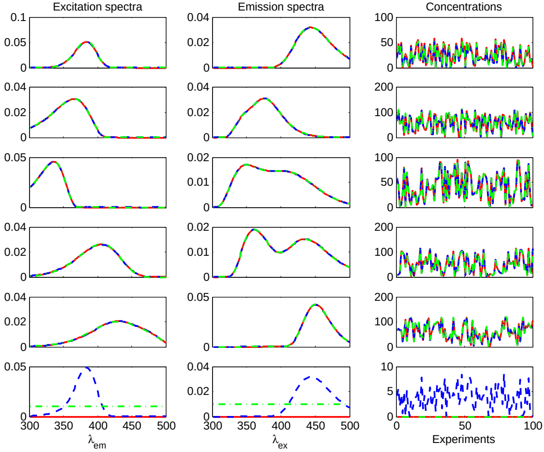


Figure: $\widehat{R} = 6$ - reference spectra / BC-VMFB without penalty / BC-VMFB with penalty $\alpha = 0.05$.

Visual results: noisy case

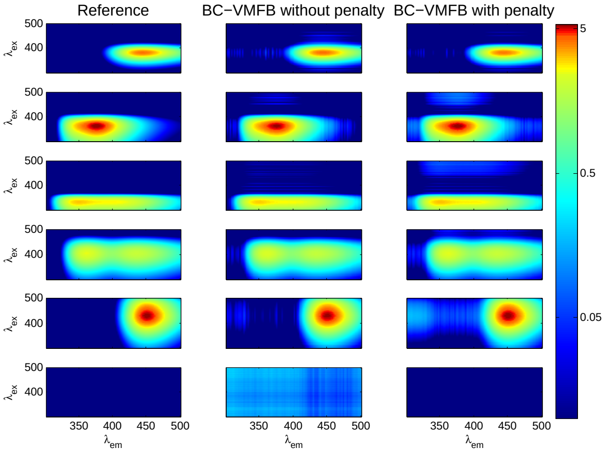


Figure: FEEM of reference (left) - FEEM reconstructed using BC-VMFB without regularization (middle) and with regularization $\alpha = 0.05$ (right).

Visual results: noisy case

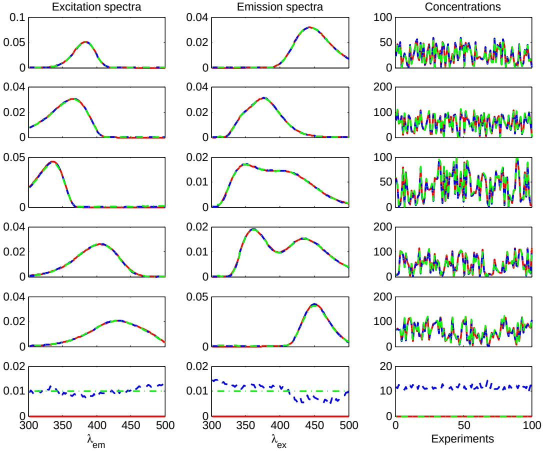


Figure: $\widehat{R} = 6$ - reference spectra / BC-VMFB without penalty / BC-VMFB with penalty $\alpha = 0.05$.

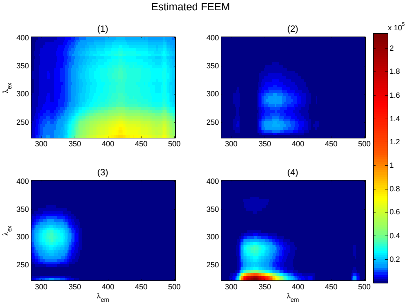
Computer simulation: real experimental data - water monitoring to detect pollutants

- Data were acquired automatically every 3 minutes, during a 10 days monitoring campaign performed on water extracted from an urban river $\Rightarrow$ tensor of size $36 \times 111 \times 2594$.
- The excitation wavelengths range from 225nm to 400nm with a 5nm bandwidth, whereas the emission wavelengths range from 280nm to 500nm with a 2nm bandwidth.
- The FEEM have been pre-processed using the Zepp's method (negative values were set to 0).

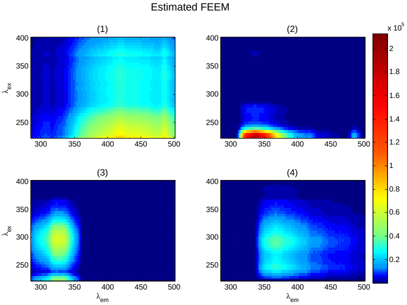
Contamination

During this experiment, a contamination with diesel oil appeared 7 days after the beginning of the monitoring.

Results: what about the rank ?



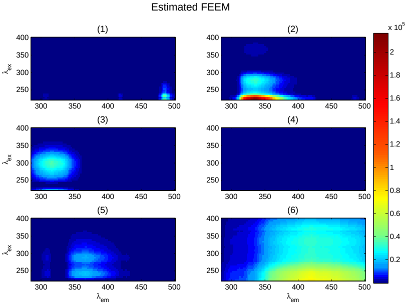
penalized BC-VMFB algorithm



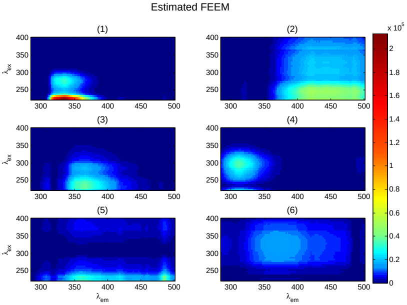
Bro's *N*-way algorithm

Case $\widehat{R} = 4$

Results: what about the rank ?



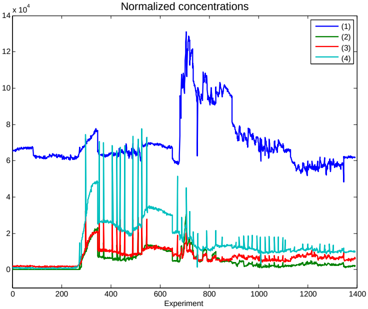
penalized BC-VMFB algorithm



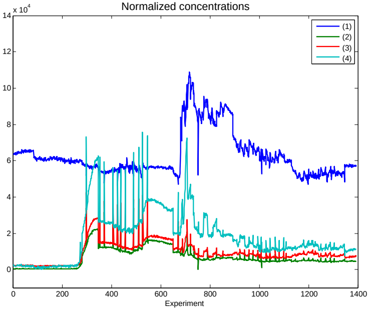
Bro's *N*-way algorithm

Case $\widehat{R} = 6$

Results: concentrations



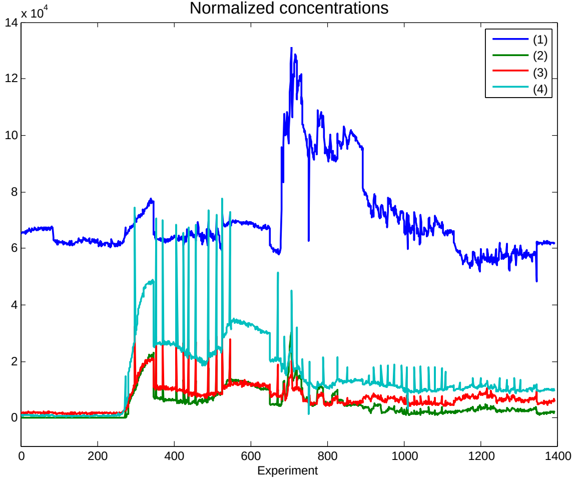
penalized BC-VMFB algorithm



Bro’s *N*-way algorithm

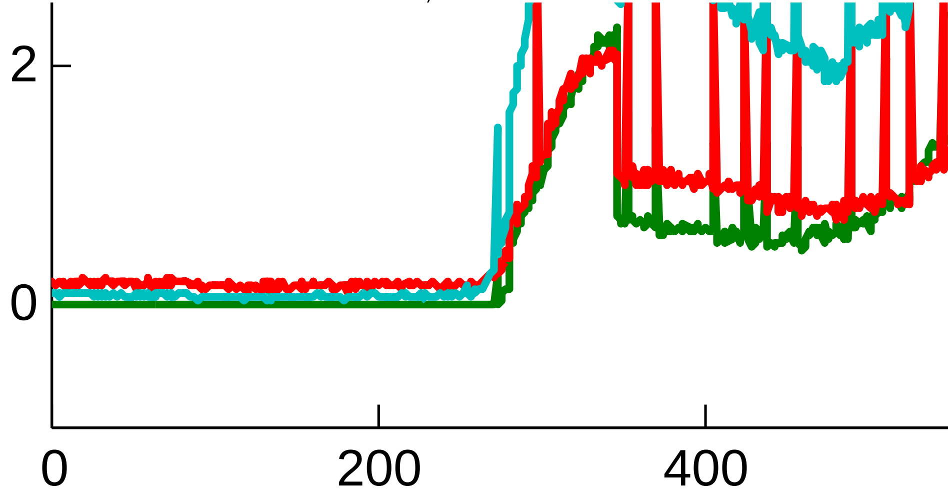
Case $\widehat{R} = 4$

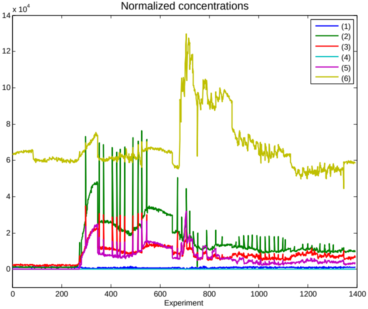
Concentrations estimated by BC-VMFB



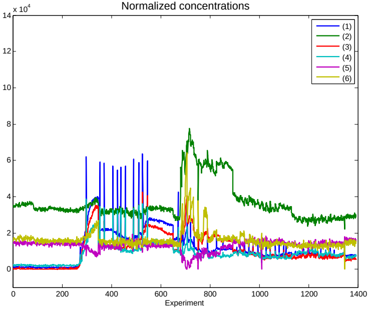
Case $\widehat{R} = 4$

Concentrations estimated by BC-VMFB





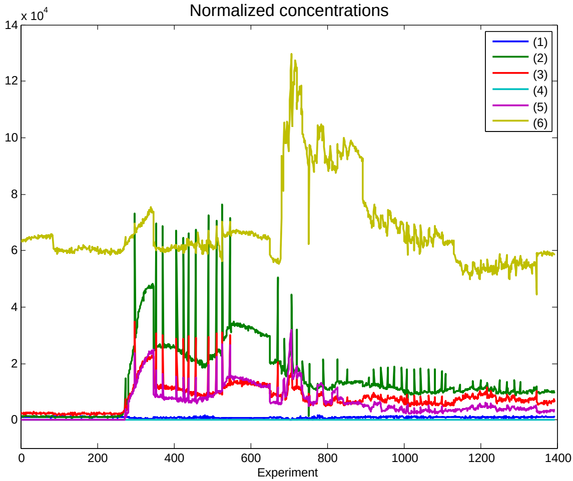
penalized BC-VMFB algorithm



Bro's N-way algorithm

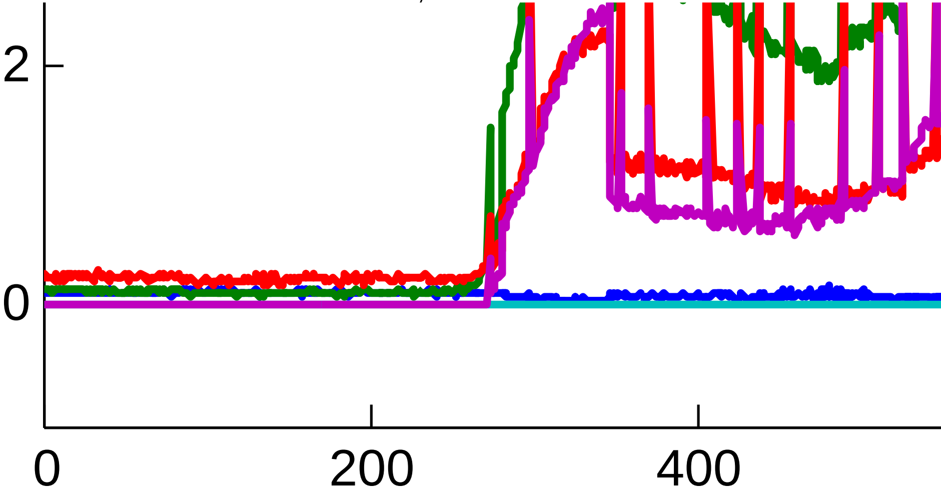
Case $\widehat{R} = 6$

Concentrations estimated by BC-VMFB



Case $\widehat{R} = 6$

Concentrations estimated by BC-VMFB



Conclusion

- clear theoretical and mathematical framework for CPD decomposition;
- interesting properties of the proposed approach: reliability, robustness versus noise and overestimation of the rank, good performance despite model errors and relative quickness;
- promising results on simulated and real data.

Perspectives:

- extension to higher order tensor (order N; LVA-ICA Grenoble 21-23 Feb. 2017);
- possibility of considering missing data;
- study other preconditionning stategies.

Conclusion

- clear theoretical and mathematical framework for CPD decomposition;
- interesting properties of the proposed approach: reliability, robustness versus noise and overestimation of the rank, good performance despite model errors and relative quickness;
- promising results on simulated and real data.

Perspectives:

- extension to higher order tensor (order N; LVA-ICA Grenoble 21-23 Feb. 2017);
- possibility of considering missing data;
- study other preconditionning stategies.

Thank you !

Questions ?

?