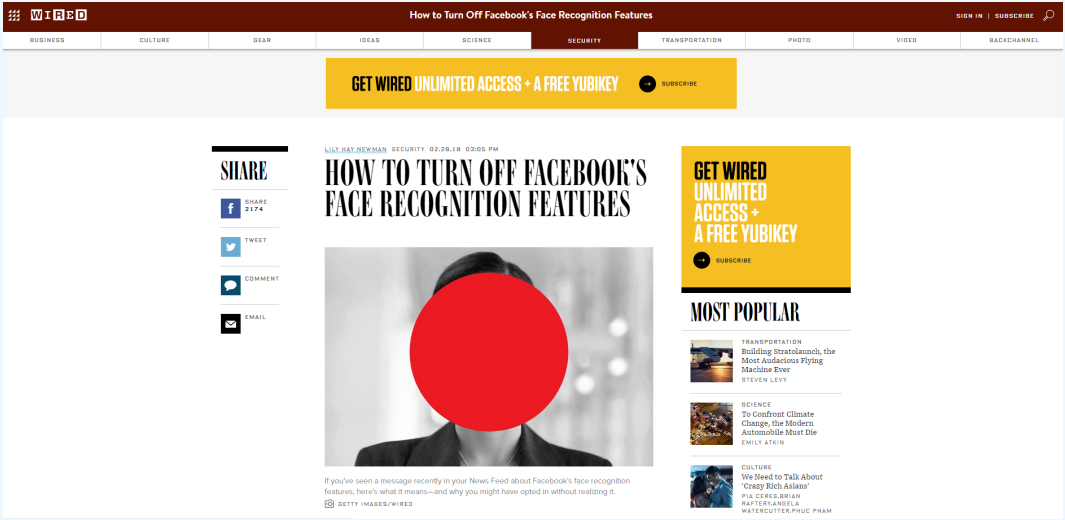
Linear Systems with a CPD Constrained Solution

Martijn Boussé

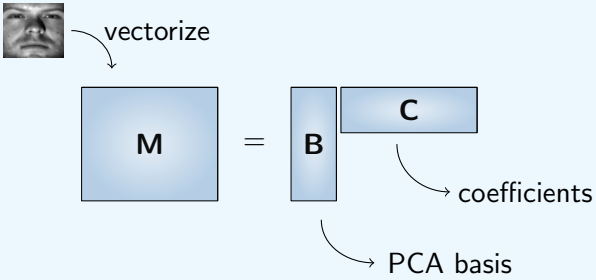
Joint work with N. Vervliet, I. Domanov, O. Debals, and L. De Lathauwer

May 6, 2018

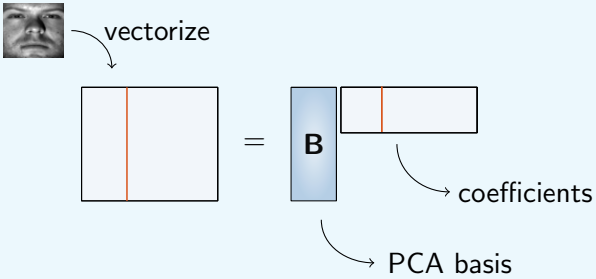


If you would like to turn it off, go to the Face Recognition tab in Settings.

EigenFaces is a simple matrix-based method for face recognition

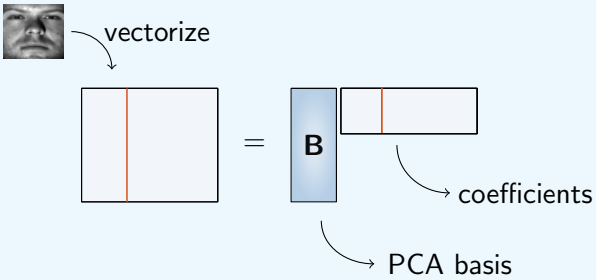


EigenFaces is a simple matrix-based method for face recognition



Each (vectorized) image can be expressed as a linear combination of the PCA basis

EigenFaces is a simple matrix-based method for face recognition



Each (vectorized) image can be expressed as a linear combination of the PCA basis

Hence, face recognition reduces to solving a linear system of equations:

$$\mathbf{d}^{(\text{new})} = \mathbf{B}\mathbf{c}^{(\text{new})}$$

and comparing $\mathbf{c}^{(\text{new})}$ with the rows of $\mathbf{C}$ to find the closest match and use the corresponding label.

However, face recognition is inherently a higher-order problem due to variations in illumination, pose, facial geometry, expression, etc.

different
illuminations

different
persons

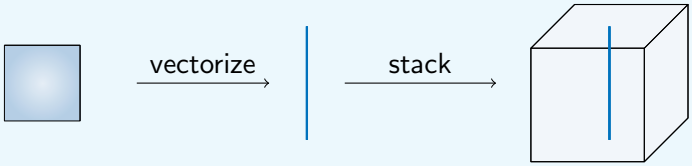
While matrix-based methods such as EigenFaces are limited to single-mode variations, tensors can explicitly accommodate for the higher-order nature of facial images

While matrix-based methods such as EigenFaces are limited to single-mode variations, tensors can explicitly accommodate for the higher-order nature of facial images

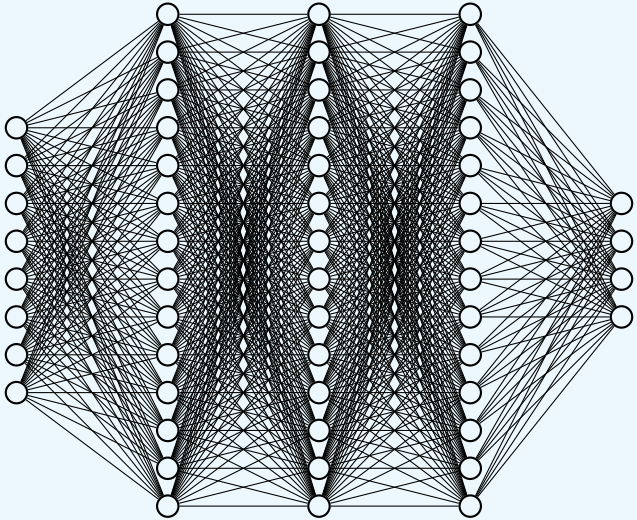
vectorize

image

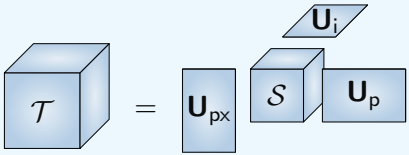
While matrix-based methods such as EigenFaces are limited to single-mode variations, tensors can explicitly accommodate for the higher-order nature of facial images



Additionally, tensor tools are typically highly interpretable in contrast to nonlinear models such as neural nets which are highly uninterpretable and often complex

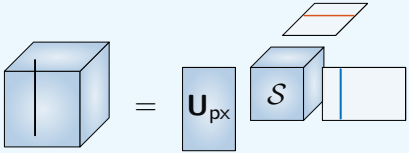


While in EigenFaces we compute a PCA ($\sim$SVD), here we compute a multilinear SVD, allowing us to exploit the higher-order nature of the problem



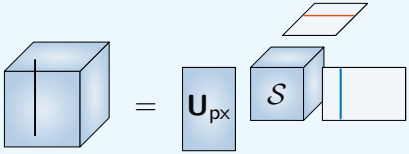
$$\mathcal{T} \approx \mathcal{S} \cdot_1 \mathbf{U}_{px} \cdot_2 \mathbf{U}_p \cdot_3 \mathbf{U}_i$$

Every vectorized facial image can then be expressed as a linear system with a rank-1 (or, in general, a low-rank) structured solution, which is the topic of this presentation!



$$\mathbf{d} = (\mathcal{S} \cdot_1 \mathbf{U}_{\text{px}}) \cdot_2 \mathbf{c}_{\text{p}}^{\mathsf{T}} \cdot_3 \mathbf{c}_{\text{i}}^{\mathsf{T}}$$

Every vectorized facial image can then be expressed as a linear system with a rank-1 (or, in general, a low-rank) structured solution, which is the topic of this presentation!



$$\mathbf{d} = (\mathcal{S} \cdot_1 \mathbf{U}_{\text{px}}) \cdot_2 \mathbf{c}_{\text{p}}^{\mathsf{T}} \cdot_3 \mathbf{c}_{\text{i}}^{\mathsf{T}}$$

$$\updownarrow$$

$$\mathbf{d} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}} \otimes \mathbf{c}_{\text{p}})$$

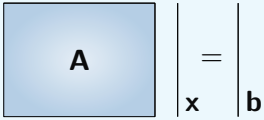
What to expect?

Definitions and links

Uniqueness and algorithms

Various applications

We consider a linear system of equations with a CPD-constrained solution

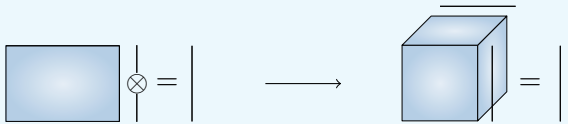


$$\text{with } \mathbf{x} = \text{vec}\left(\left[\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \cdots, \mathbf{U}^{(N)}\right]\!\right]\right)$$

Evidently, one can also consider other decompositions such as the multilinear SVD, tensor trains (TTs) and hierarchical Tucker (hT) models.

TTs and hT models are popular in tensor-based scientific computing

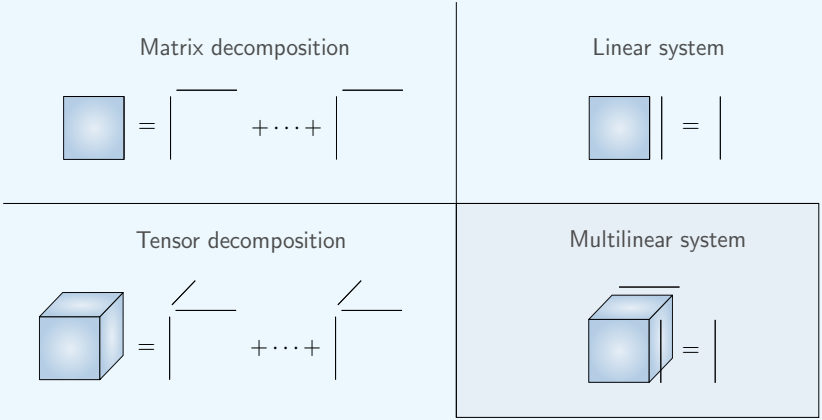
An LS-CPD can be interpreted as a multilinear system of equations.



$$\mathbf{A}(\mathbf{x} \otimes \mathbf{y}) = \mathbf{b} \qquad \mathcal{A} \cdot_2 \mathbf{y}^\mathsf{T} \cdot_3 \mathbf{x}^\mathsf{T} = \mathbf{b}$$

$\mathbf{A}$ equals the mode-1 unfolding of $\mathcal{A}$ denoted by $\mathbf{A}_{(1)}$. Remember: $(\mathcal{A} \cdot_n \mathbf{B})_{(n)} = \mathbf{B}\mathbf{A}_{(n)}$.

Multilinear systems are a generalization of linear systems in a similar way as tensor decompositions are a generalization of matrix decompositions



Link with compressed sensing

If $\mathbf{A}$ is diagonal and binary, LS-CPD reduces to:

$$\mathcal{B} = \mathcal{D} * \left[\!\left[\mathbf{U}^{(1)}, \mathbf{U}^{(2)} \cdots, \mathbf{U}^{(N)} \right]\!\right] \quad \text{with} \quad \begin{aligned} \operatorname{vec}(\mathcal{B}) &= \mathbf{b} \\ \operatorname{vec}(\mathcal{D}) &= \operatorname{diag}(\mathbf{A}) \end{aligned}$$

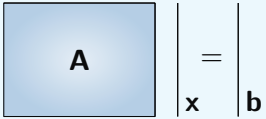
Although only a limited number of entries of $\mathbf{b}$ are known, $\mathbf{b}$ can be fully determined by computing the CPD.

Definitions and links

Uniqueness and algorithms

Various applications

For generic uniqueness, we need at least as many equations as free variables (compensated for scaling).



Lemma. *Let* $\mathbf{A}$ *be a generic* $M \times K$ *matrix with* $K = I_1 \cdots I_N$. *Define* $\mathbf{b} = \mathbf{A}vec(\mathcal{X}_0)$ *with* $\mathcal{X}_0$ *a* $I_1 \times \cdots \times I_N$ *tensor with rank less than or equal to* R. *In that case, the solution vector* $\mathbf{x}$ *is unique if* $M \geq (\underbrace{I_1 + \cdots + I_N}_{\text{free variables}} - \underbrace{N + 1}_{\text{scaling}})R + 1$.

Generic uniqueness means that we have uniqueness with probability one when the entries of $\mathbf{A}$ are drawn from absolutely continuous probability density functions.

Algebraic method to solve LS-CPDs when considering a rank-1 structure.

We use the well-known fact that:

$$\operatorname{rank}(\mathcal{X}) = 1 \iff \operatorname{rank}\left(\mathbf{X}_{(n)}\right) = R = 1, \quad \text{for } 1 \leq n \leq N. \quad (1) \tag{1}$$

In this particular case a solution $\mathbf{x}$ for LS-CPD is also a solution of the *relaxed* problem:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \text{with} \quad \mathbf{x} = \operatorname{vec}(\mathcal{X}), \text{ where } \mathcal{X} \text{ satisfies } (1)$$

and vice versa.

Hence, we can compute a solution for LS-CPD algebraically in two steps.

1. Solve the relaxed problem to recover $\mathcal{X}$
2. and compute the (exact) rank-1 CPD of $\mathcal{X}$.

In the trivial case, null $\left(\begin{bmatrix}\mathbf{A} & \mathbf{b}\end{bmatrix}\right) = 1$, allowing one to solve the relaxed problem by ignoring the multilinear structure.

If null $\left(\begin{bmatrix}\mathbf{A} & \mathbf{b}\end{bmatrix}\right) > 1$, one can recast the relaxed problem to the trivial case and solve an LS-CPD.

Nonlinear least-squares algorithm to solve LS-CPDs

Suppose we have a LS-CPD of the form (with $\mathbf{A}_{(1)} = \mathbf{A}$):

$$\mathbf{A}(\mathbf{x} \otimes \mathbf{y}) = \mathbf{b} \quad \leftrightarrow \quad \mathcal{A} \cdot_2 \mathbf{y}^\mathsf{T} \cdot_3 \mathbf{x}^\mathsf{T} = \mathbf{b}.$$

Then we minimize:

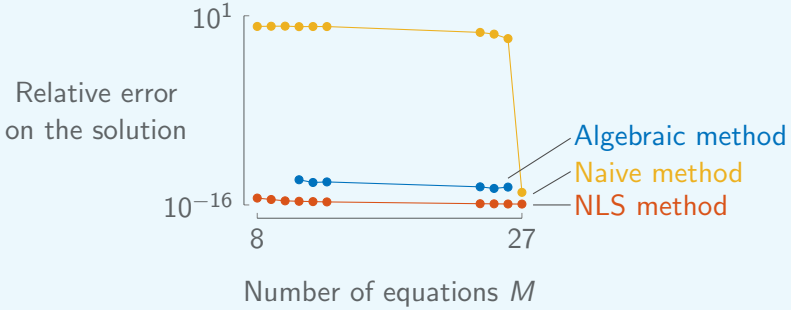
$$\min_{\mathbf{x},\mathbf{y}} \quad \frac{1}{2} ||\mathcal{A} \cdot_2 \mathbf{y}^\mathsf{T} \cdot_3 \mathbf{x}^\mathsf{T} - \mathbf{b}||_\mathsf{F}^2$$

Then, use the optimization framework from Nico Vervliet's talk.

Optimization-based framework to solve LS-CPDs when considering low-rank structure.

Input: $\mathbf{A}$, $\mathbf{b}$, and $\{\mathbf{U}^{(n)}\}_{n=1}^{N}$
Output: $\{\mathbf{U}^{(n)}\}_{n=1}^{N}$
while *not converged* **do**
- Compute gradient $\mathbf{g}$.
- Use PCG to solve $\mathbf{H}\mathbf{p} = -\overline{\mathbf{g}}$ for $\mathbf{p}$ using Gramian-vector products using a (block)-Jacobi preconditioner.
- Update $\mathbf{U}^{(n)}$, for $1 \leq n \leq N$, using dogleg trust region from $\mathbf{p}$, $\mathbf{g}$, and function evaluation.

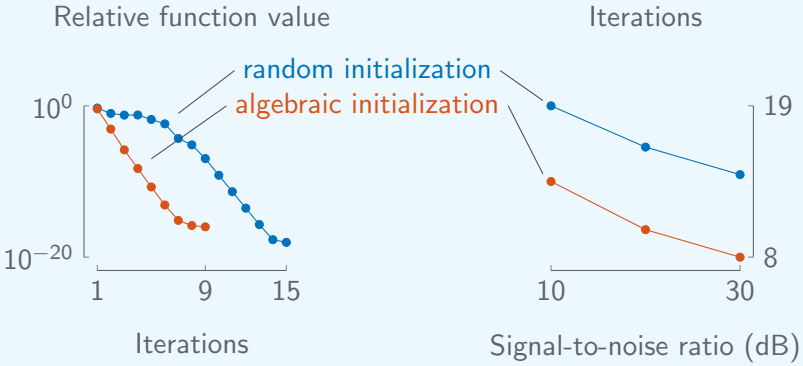
The naive method fails for an underdetermined LS-CPD while the NLS and algebraic method both perform well.



NLS typically needs many random initializations when M is close to $M^{(min)}$.

$N = 3, I = 3, R = 1, K = 27, M^{(min)} = 8$

By starting the NLS algorithm from the algebraic solution, fewer iterations are needed to achieve convergence.



Importantly, the algebraic method can still find a solution in the noisy case, but the accuracy is typically low.

$N = 3, I = 4, R = 1, K = 64, M = 60$

Complexity analysis teaches us the importance of exploiting coefficient structure

	Calls per iteration	Complexity
Objective function	$1 + \text{it}_{\text{TR}}$	$\mathcal{O}(MRI^N)$
Jacobian	1	$\mathcal{O}(MRNI^N)$
Gradient	1	$\mathcal{O}(MRNI)$
Gramian-vector	it_{CG}	$\mathcal{O}(MRNI)$

The per-iteration complexity of the NLS algorithm is dominated by the computation of the Jacobian.

Assume all dimensions are equal to I.

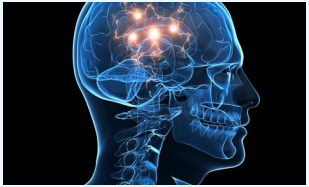
Definitions and links

Uniqueness and algorithms

Various applications

The power of the LS-CPD framework is in its wide applicability

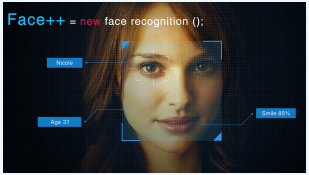
Biomedical signal processing



Array processing & Telecom

Large-scale problems

Computer vision



System identification

Tensor algebra

Quick recap of the face recognition problem

Quick recap of the face recognition problem

Tensorize (labeled) dataset into $\mathcal{T}$

Quick recap of the face recognition problem

Tensorize (labeled) dataset into $\mathcal{T}$

Compute multilinear SVD of $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_{\text{px}} \cdot_2 \mathbf{U}_{\text{p}} \cdot_3 \mathbf{U}_{\text{i}}$

Quick recap of the face recognition problem

Tensorize (labeled) dataset into $\mathcal{T}$

Compute multilinear SVD of $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_{\text{px}} \cdot_2 \mathbf{U}_{\text{p}} \cdot_3 \mathbf{U}_{\text{i}}$

Every image admits $\mathbf{d} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}} \otimes \mathbf{c}_{\text{p}})$

Quick recap of the face recognition problem

Tensorize (labeled) dataset into $\mathcal{T}$

Compute multilinear SVD of $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_{\text{px}} \cdot_2 \mathbf{U}_{\text{p}} \cdot_3 \mathbf{U}_{\text{i}}$

Every image admits $\mathbf{d} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}} \otimes \mathbf{c}_{\text{p}})$

Hence, for a new image we solve

$$\mathbf{d}^{(\text{new})} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}}^{(\text{new})} \otimes \mathbf{c}_{\text{p}}^{(\text{new})})$$

Quick recap of the face recognition problem

Tensorize (labeled) dataset into $\mathcal{T}$

Compute multilinear SVD of $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_{\text{px}} \cdot_2 \mathbf{U}_{\text{p}} \cdot_3 \mathbf{U}_{\text{i}}$

Every image admits $\mathbf{d} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}} \otimes \mathbf{c}_{\text{p}})$

Hence, for a new image we solve

$$\mathbf{d}^{(\text{new})} = (\mathbf{U}_{\text{px}}\mathbf{S}_{(1)})(\mathbf{c}_{\text{i}}^{(\text{new})} \otimes \mathbf{c}_{\text{p}}^{(\text{new})})$$

and we compare the estimate of $\mathbf{c}_{\text{p}}^{(\text{new})}$ with the rows of $\mathbf{U}_{\text{p}}$
to find the closest match and use the corresponding label.

We use the extended Yale B dataset to illustrate the LS-CPD-based approach

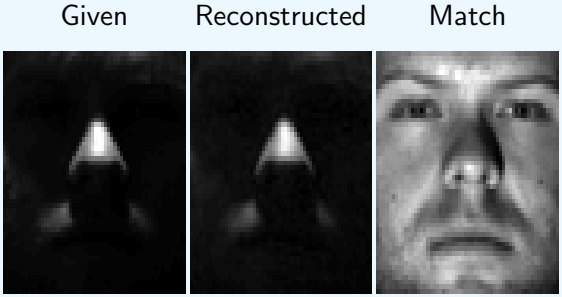
37 persons
64 illuminations
51×58 image

After preprocessing, we obtain a tensor of size $2958 \times 36 \times 57$

We use Tensorlab for all computations in MATLAB

Details can be found in [Boussé et al., 2017d]

We correctly classify a known person even though the image is almost completely dark



The multilinear SVD model generalizes quite well for a new person using only one image with a neutral illumination

Given Reconstructed

We correctly classify the new person using a different illumination

LS-CPD-based face recognition enables better recognition rates (%) than a matrix and another tensor-based method

	EigenFaces	TensorFaces	LS-CPD method
Accuracy	93.3	93.5	95.7
Precision	90.6	94.4	96.6
Recall	88.4	90.9	95.8
Time (s) of PCA/MLSVD	2.97	3.29	3.29
Time of recognition	0.004	0.135	0.097

Precision: What proportion of positive identifications was actually correct?
Recall: What proportion of actual positives was identified correctly?

The LS-CPD framework offers a tensor-based classification methodology with many possible applications

The LS-CPD framework offers a tensor-based classification methodology with many possible applications

Obtain a (labeled) tensor $\mathcal{T}$ in some way

Use the MLSVD model $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_1 \cdot_2 \mathbf{U}_2 \cdot_3 \mathbf{U}_3$

Every mode-1 fiber satisfies $\mathbf{t} = (\mathbf{U}_1 \mathbf{S}_{(2)})(\mathbf{c}_3 \otimes \mathbf{c}_2)$

Hence, for a new fiber we solve

$$\mathbf{t}^{(\text{new})} = (\mathbf{U}_1 \mathbf{S}_{(2)})(\mathbf{c}_3^{(\text{new})} \otimes \mathbf{c}_2^{(\text{new})})$$

and compare the coefficient vector of interest with the rows of the corresponding factor matrix, using the corresponding label to classify the new fiber.

Biomedical examples, see [Boussé et al., 2017c; Van Eyndhoven et al., 2018]

The LS-CPD framework offers a tensor-based classification methodology with many possible applications

Obtain a (labeled) tensor $\mathcal{T}$ in some way

Use the MLSVD model $\mathcal{T} = \mathcal{S} \cdot_1 \mathbf{U}_1 \cdot_2 \mathbf{U}_2 \cdot_3 \mathbf{U}_3$

Every mode-1 fiber satisfies $\mathbf{t} = (\mathbf{U}_1 \mathbf{S}_{(2)})(\mathbf{c}_3 \otimes \mathbf{c}_2)$

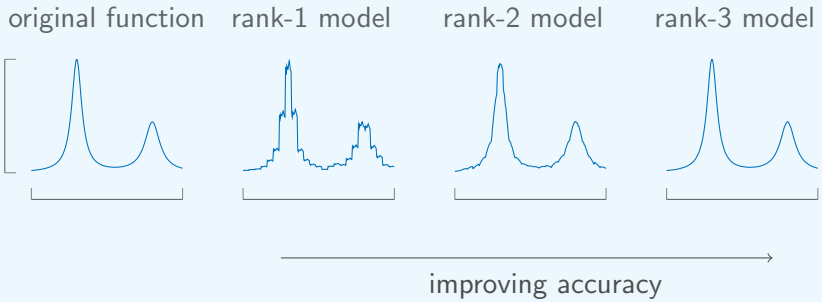
Hence, for a new fiber we solve

$$\mathbf{t}^{(\text{new})} = (\mathbf{U}_1 \mathbf{S}_{(2)})(\mathbf{c}_3^{(\text{new})} \otimes \mathbf{c}_2^{(\text{new})})$$

and compare the coefficient vector of interest with the rows of the corresponding factor matrix, using the corresponding label to classify the new fiber.

Biomedical examples, see [Boussé et al., 2017c; Van Eyndhoven et al., 2018]

The optimization-based method exactly reconstructs a rational solution vector in the noiseless case.



The exact rank is known for various cases such as exponential polynomials, rational functions, and periodic functions.

In general, smooth signals can often be well approximated by a low-rank tensor model.

[Boussé et al., 2017b,a; Debals et al., 2016; De Lathauwer, 2011; Grasedyck, 2010; Grasedyck et al., 2013; Khoromskij, 2011]

The construction of a tensor with particular multilinear singular values can be written as a LS-CPD.

Consider an all-orthogonal tensor $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$ with multilinear singular values $\boldsymbol{\sigma}^{(n)}$, for $1 \leq n \leq 3$, or, equivalently, one can write:

$$\mathbf{T}_{(n)}\mathbf{T}_{(n)}^{\mathsf{T}} = \boldsymbol{\Sigma}^{(n)} \quad \text{with} \quad \boldsymbol{\Sigma}^{(n)} = \operatorname{diag}\left(\boldsymbol{\sigma}^{(n)}\right) \quad \text{for} \quad 1 \leq n \leq 3.$$

By taking $n = 1$ and exploiting symmetry, one obtains:

$$\sum_{j,k} t_{ijk}\, t_{ijk} = \left(\sigma_i^{(1)}\right)^2, \text{ for } 1 \leq i \leq I,$$

$$\sum_{j,k} t_{i_1 jk}\, t_{i_2 jk} = 0, \text{ for } 1 \leq i_1 < i_2 \leq I,$$

and similarly for $n = 2$ and 3.

The construction of a tensor with particular multilinear singular values can be written as a LS-CPD.

More compactly, one can write this as a LS-CPD:

$$\mathbf{A}(\mathbf{u} \otimes \mathbf{u}) = \mathbf{b} \quad \text{with} \quad \mathbf{u} = \text{vec}(\mathcal{T})$$

in which:

- $\mathbf{A}$ is a binary and sparse matrix
- and each entry of $\mathbf{b}$ is either zero or a squared MLSV.

[Boussé et al., 2018b]

By exploiting the sparsity in the Jacobian, the NLS-based method is faster than the alternating projection method.

	$\alpha = 1$	$\alpha = 5$	$\alpha = 10$
Alternating projection method (APM)	0.100	34.4	1747
Construction of **A**	0.004	1.4	23
Initialization (i.e., one iteration of APM)	0.002	0.4	12
LS-CPD	0.023	15.4	444
Total computation time of LS-CPD	0.029	17.2	479

We report median computation time (in seconds) across 20 experiments for a tensor of size $10\alpha \times 10\alpha \times 5\alpha$.

LS-CPDs can be used to solve the blind deconvolution of a constant modulus (CM) signal.

Consider a single-input-single-output autoregressive system:

$$\sum_{l=0}^{L} w_l \cdot y[k-l] = s[k] + n[k], \text{ for } 1 \leq k \leq K.$$

In matrix form, we have:

$$\mathbf{Y}^{\mathsf{T}}\mathbf{w} = \mathbf{s}.$$

The goal of blind deconvolution is to find the filter coefficients using only the measured output values.

In order to make the problem identifiable, we assume the input signal has constant modulus (CM), i.e., each sample s_k satisfies:

$$|s_k|^2 = s_k \cdot \overline{s}_k = c, \text{ for } 1 \leq k \leq K$$

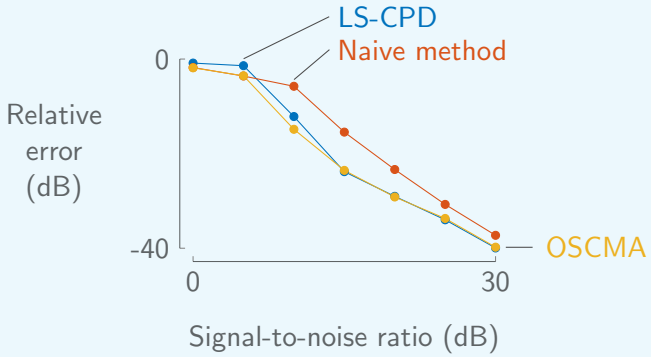
with c the squared constant modulus which is known *a priori*.

Taking into account all equations, we obtain an LS-CPD:

$$\left(\mathbf{Y} \odot \overline{\mathbf{Y}}\right)^{\mathsf{T}} (\mathbf{w} \otimes \overline{\mathbf{w}}) = c \cdot \mathbf{1}_K.$$

[Boussé et al., 2018b]

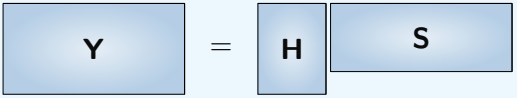
LS-CPD enables more accurate results than the naive method and achieves similar accuracy as OSCMA.



$L = 5$, $K = 100$, $c = 1$, using uniformly distributed coefficients in $[0, 1]$.

Blind signal separation of constant modulus signals can also be interpreted as a LS-CPD

Consider an instantaneous model with R constant modulus source signals:



with $|s_{rk}|^2 = 1 \ \forall r, k$

Using the same strategy as before, we obtain a LS-CPD with R possible solutions:

$$\left(\mathbf{Y} \odot \overline{\mathbf{Y}}\right)^{\mathsf{T}} \left(\mathbf{w}_r \otimes \overline{\mathbf{w}}_r\right) = \mathbf{1}_K \tag{2}$$

The well-known Analytical Constant Modulus Algorithm (ACMA) solves (2) as follows:

1. First, find R solutions $\mathbf{x}_r$ of the linear system while ignoring the structure.
2. Next, find $\mathbf{w}_r$ by simultaneous diagonalization of $\mathbf{X}_r$ for $1 \leq r \leq R$ ($\sim$CPD).

[van der Veen and Paulraj, 1996]

Large-scale autoregressive system identification can be handled via LS-CPD

- Consider an Lth order AR model with Q outputs and one exogenous input:

$$\sum_{l=0}^{L} g_q[l] y_q[k-l] = x[k] + n[k] \quad \text{for} \quad 1 \leq k \leq K$$

- Assuming we have $K + L$ samples, we have the following matrix notation:

$$\sum_{l=0}^{L} \mathbf{g}^{(l)\mathsf{T}} \mathbf{Y}^{(l)} = \mathbf{x}^{\mathsf{T}} + \mathbf{n}^{\mathsf{T}}$$

- Assuming $\mathbf{g}^{(l)} = \mathbf{b}^{(l)} \otimes \mathbf{a}^{(l)}$ and taking the transpose, results into a LS-CPD:

$$\sum_{l=0}^{L} \mathbf{Y}^{(l)\mathsf{T}} \left(\mathbf{b}^{(l)} \otimes \mathbf{a}^{(l)} \right) = \mathbf{x}$$

[Boussé et al., 2018a]

Linear Systems with a CPD Constrained Solution

Martijn Boussé

Joint work with N. Vervliet, I. Domanov, O. Debals, and L. De Lathauwer

May 6, 2018

References I

M. Boussé, O. Debals, and L. De Lathauwer. Tensor-based large-scale blind system identification using segmentation. *IEEE Transactions on Signal Processing*, Nov. 2017a.

M. Boussé, O. Debals, and L. De Lathauwer. A tensor-based method for large-scale blind source separation using segmentation. *IEEE Transactions on Signal Processing*, Jan. 2017b.

M. Boussé, G. Goovaerts, N. Vervliet, O. Debals, S. Van Huffel, and L. De Lathauwer. Irregular heartbeat classification using Kronecker product equations. In *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2017, Jeju Island, South-Korea)*, July 2017c.

References II

M. Boussé, N. Vervliet, O. Debals, and L. De Lathauwer. Face recognition as a Kronecker product equation. In *IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP 2017, Curaçao, Dutch Antilles)*, Dec. 2017d.

M. Boussé, , and L. De Lathauwer. Large-scale autoregressive system identification using Kronecker product equations. Technical Report 17-164, ESAT-STADIUS, KU Leuven, Leuven, Belgium, 2018a.

M. Boussé, N. Vervliet, I. Domanov, O. Debals, and L. De Lathauwer. Linear systems with a canonical polyadic decomposition constrained solution: Algorithms and applications. 2018b. Accepted for publication in Numerical Linear Algebra with Applications for a Special Issue on Matrix Equations and Tensor Techniques.

References III

L. De Lathauwer. Blind separation of exponential polynomials and the decomposition of a tensor in rank-$(L_r, L_r, 1)$ terms. *SIAM Journal on Matrix Analysis and Applications*, Dec. 2011.

O. Debals and L. De Lathauwer. The concept of tensorization. *Technical Report 17-99, ESAT-STADIUS, KU Leuven, Leuven, Belgium*, 2017.

O. Debals, M. Van Barel, and L. De Lathauwer. Löwner-based blind signal separation of rational functions with applications. *IEEE Transactions on Signal Processing*, Apr. 2016.

S. Geirnaert, G. Goovaerts, S. Padhy, M. Boussé, L. De Lathauwer, and S. Van Huffel. Tensor-based ECG signal processing applied to atrial fibrillation detection. Technical report, ESAT-STADIUS, KU Leuven, Leuven, Belgium, 2018. accepted for publication in Proceedings of the 52nd Asilomar Conference on Signals, Systems and Computers (Asilomar 2018, Pacific Grove, California, US).

References IV

L. Grasedyck. Polynomial approximation in hierarchical Tucker format by vector-tensorization. Apr. 2010. Preprint 43, DFG/SPP1324, RWTH Aachen.

L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, Feb. 2013.

B. N. Khoromskij. $\mathcal{O}(d\log N)$-quantics approximation of N-d tensors in high-dimensional numerical modeling. *Constructive Approximation*, Oct. 2011.

A.-J. van der Veen and A. Paulraj. An analytical constant modulus algorithm. *IEEE Transactions on Signal Processing*, May 1996.

S. Van Eyndhoven, M. Boussé, B. Hunyadi, L De Lathauwer, and S. Van Huffel. Single-channel EEG classification by multi-channel tensor subspace learning and regression. Technical report, ESAT-STADIUS, KU Leuven, Leuven, Belgium, 2018. accepted for publication in Proceedings of the 2018 IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2018, Aalborg, Denmark.

References V

M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the European Conference on Computer Vision (ECCV '02, Copenhagen, Denmark)*, May 2002.