Is Nonnegative Tucker Decomposition the new NMF?

Jeremy E. Cohen, CREATIS, CNRS, France

Workshop on Tensor theory and Methods
23 November 2022

Credits

Roadmap

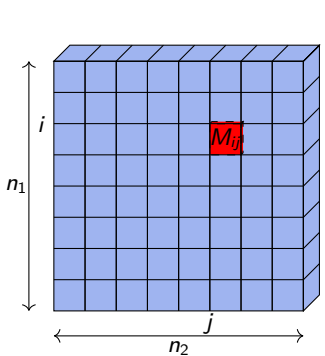
- Tensor Decompositions 101
- An illustration of NTD to Music Information Retrieval
- Some theory on NMF/NTD and open questions
- Numerical optimization methods for NTD
- Off topic: Tensorly

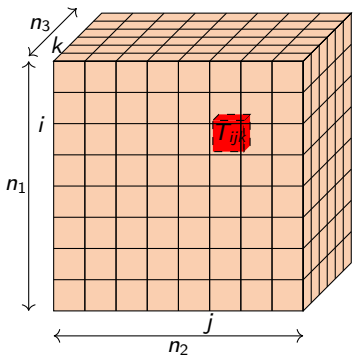
Matrices/Tensors as multiway arrays

Let $\mathcal{T}$ a tensor in $\mathbb{R}^{n_1 \times n_2 \times \ldots \times n_d}$

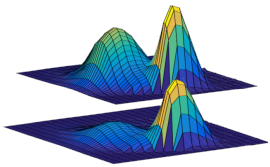
modes: indexes of the tensor from 1 to d. e.g. i is the first mode index.

order: d. e.g. the tensor below is a third order tensor.

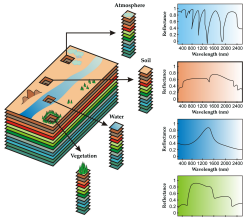




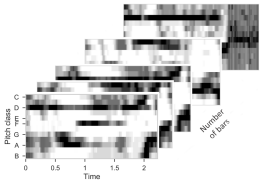
Examples of tensors in data science



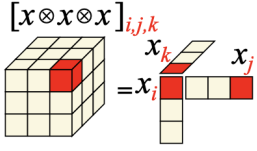
Tensor as Raw Data
Excitation Emission Matrices



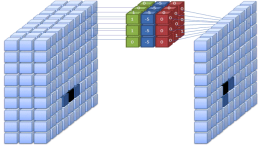
Tensor as Raw Data
Hyperspectral Images
[courtesy of J Chanussot]



Tensor as Processed Data
Tensor spectrogram



Tensor as Data Properties
Data Moments

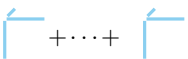


Tensor as Model Parameters
Convolutional Neural Networks
[figure from commons.wikimedia.org]

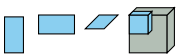
Tensors and dimensionality reduction

Number of parameters:

$\mathcal{O}(n^d)$



$\mathcal{O}(dnr)$



$\mathcal{O}(dnr + r^d)$

$\mathcal{O}(dnr^2)$

Consequently, tensor models can be used for:

Inverse Problems

- Matrix-Tensor completion
- Blind Source separation
- Denoising, deconvolution
- Phase retrieval
- ...

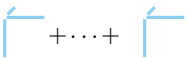
Compression, Low Complexity Model

- Big Data
- Data mining
- Neural Networks
- Partial Differential Equations
- ...

Tensors and dimensionality reduction

Number of parameters:

$\mathcal{O}(n^d)$



$\mathcal{O}(dnr)$

$\mathcal{O}(dnr + r^d)$

$\mathcal{O}(dnr^2)$

Consequently, tensor models can be used for:

Inverse Problems

- Matrix-Tensor completion
- Blind Source separation
- Denoising, deconvolution
- Phase retrieval
- ...

Compression, Low Complexity Model

- Big Data
- Data mining
- Neural Networks
- Partial Differential Equations
- ...

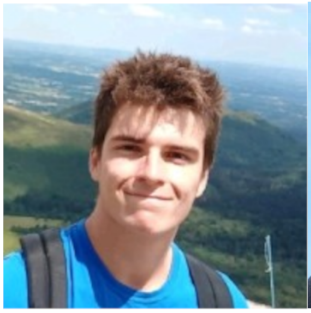
Segmenting a song?

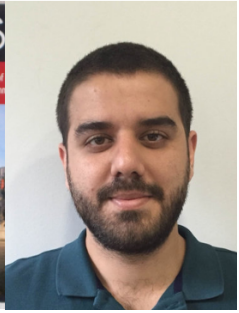
Organisation of the song:	Verse	Chorus	Verse	Solo	Chorus

Large scale structure:	A	B	A	C	B’

Small scale structure:	a	b	c	c	a	b	d	e	f	c	c’

A team effort

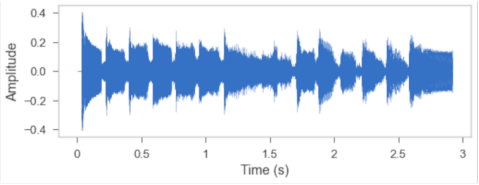


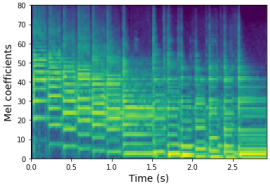



Axel Marmoret	Nancy Bertin	Frederic Bimbot	Caglayan Tuna
PhD student	CR CNRS	DR CNRS	Inria Engineer

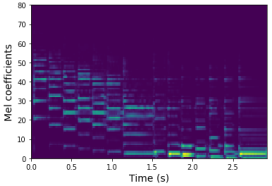
Axel Marmoret, Jérémy Cohen, Nancy Bertin, Frédéric Bimbot. Uncovering Audio Patterns in Music with Nonnegative Tucker Decomposition for Structural Segmentation. ISMIR 2020 - 21st International Society for Music Information Retrieval, Oct 2020, Montréal (Online), Canada. pp.1-7

From audio to time-frequency signals

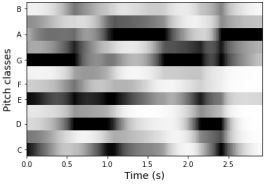




Log-Mel spectrogram



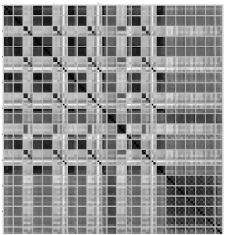
NN-Log-Mel spectrogram



chromagram

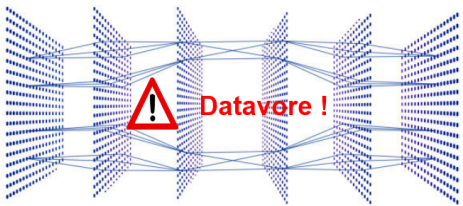
A word on the state-of-the-art

Unsupervised



Signal Autosimilarity + post-processing

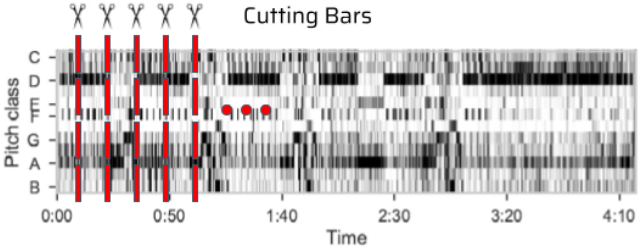
Supervised



Deep learning

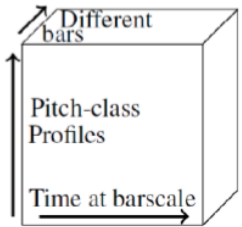
An idea: form a time-frequency tensor...

Chromagram



Chromagram of "Come Together", by The Beatles.

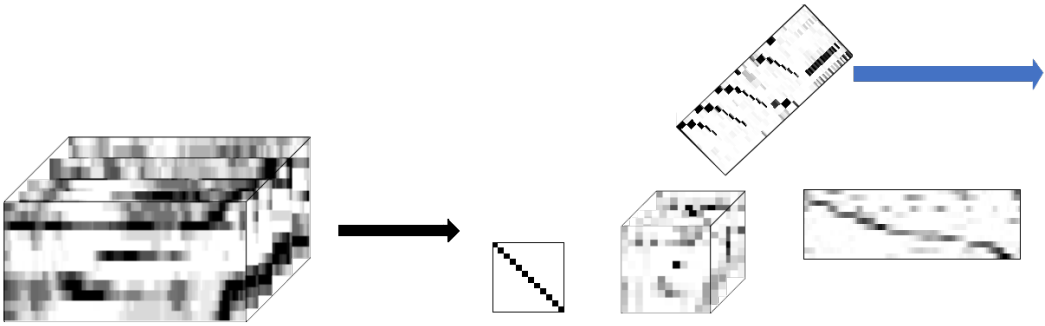
Stack



Decompose

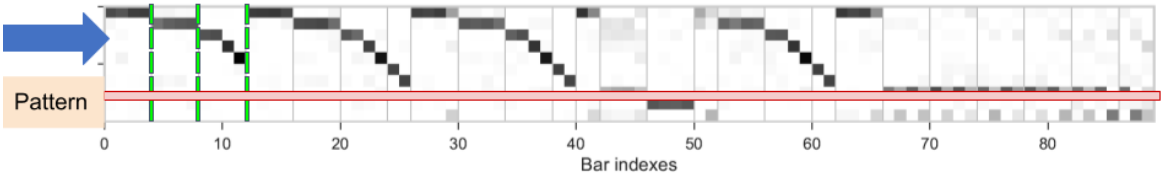
...and decompose it to find redundancies!

Approximate Nonnegative Tucker Decomposition

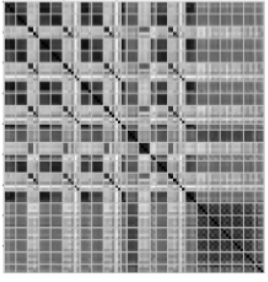


The inner dimensions are hyperparameters!!

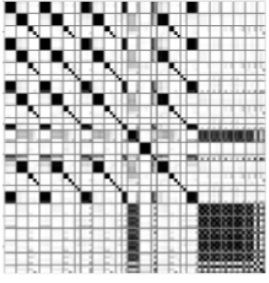
Back to segmentation



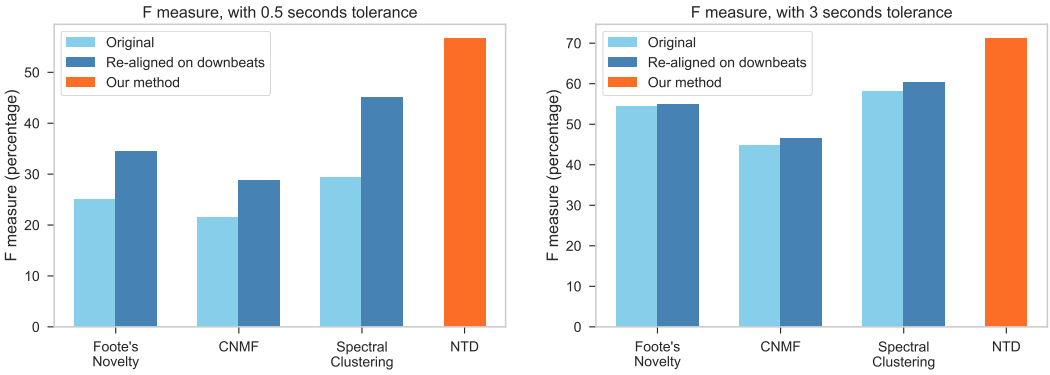
Signal Autosimilarity



Patterns autosimilarity



State-of-the-art unsupervised results!



Algorithm	$P_{0.5}$	$R_{0.5}$	$F_{0.5}$	P_3	R_3	F_3
NTD, with "oracle ranks" for each song	67.1%	78.2%	71.5%	78.5%	90.2%	83.1%
Neural Networks[Grill2015]	80.4%	62.7%	69.7%	91.9%	71.1%	79.3%

Table: Averaged segmentation scores in the "oracle ranks" condition, compared to the current state-of-the-art (non-blind) method.

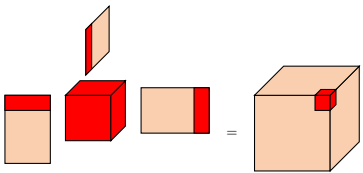
What is Tucker Decomposition

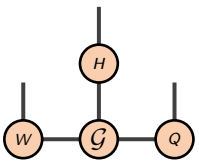
The Tucker format (3d order)

Input: Data tensor $\mathcal{T}$, core dimensions r_1, r_2, r_3
Parameters: $W \in \mathbb{R}^{n_1 \times r_1}$, $H \in \mathbb{R}^{n_2 \times r_2}$, $Q \in \mathbb{R}^{n_3 \times r_3}$ and $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$

$$\mathcal{T}_{ijk} = \sum_{q_1}^{r_1} \sum_{q_2}^{r_2} \sum_{q_3}^{r_3} W_{ir_1} H_{jr_2} Q_{kr_3} G_{r_1 r_2 r_3}$$

$$\mathcal{T} = (W \otimes H \otimes Q)\,\mathcal{G}$$





What is Tucker Decomposition

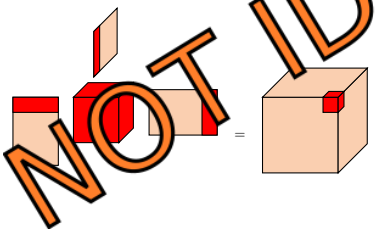
The Tucker format (3d order)

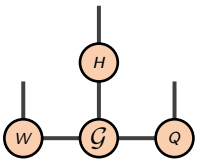
Input: Data tensor $\mathcal{T}$, core dimensions r_1, r_2, r_3
Parameters: $W \in \mathbb{R}^{n_1 \times r_1}$, $H \in \mathbb{R}^{n_2 \times r_2}$, $Q \in \mathbb{R}^{n_3 \times r_3}$ and $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$

$$\mathcal{T}_{ijk} = \sum_{q_1}^{r_1} \sum_{q_2}^{r_2} \sum_{q_3}^{r_3} W_{iq_1} H_{jq_2} Q_{kq_3} \mathcal{G}_{q_1 q_2 q_3}$$

$$\mathcal{T} = (W \otimes H \otimes Q)\,\mathcal{G}$$

NOT IDENTIFIABLE





Why Nonnegativity in Tucker decomposition, the NMF case

$$M = WH = WPP^{-1}H$$

but if $W \geq 0$ and $H \geq 0$, sometimes

$$WP \geq 0 \text{ and } P^{-1}H \geq 0 \implies P = \Pi\Sigma$$

with Π a permutation matrix and Σ a positive diagonal matrix.

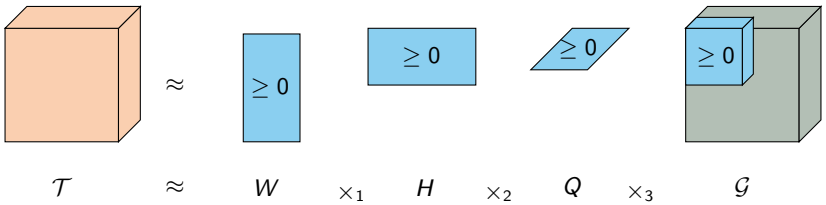
A collection of sufficient conditions for NMF identifiability

- [Donoho2003]: Separability
- [Huang2013]: sufficiently scattered condition
- [Miao2007], Fu2015/Lin2015: Minimum Volume

For NTD, rotation ambiguities on all modes!

$$\mathcal{T} = (WP_1 \otimes HP_2 \otimes QP_3)\left[\left(P_1^{-1} \otimes P_2^{-1} \otimes P_3^{-1}\right)\mathcal{G}\right]$$

(approximate) Nonnegative Tucker Decomposition



In practice, given data $\mathcal{T}$ and hyperparameters $r_{1,2,3}$, we solve an optimization problem of the form

$$\underset{W \in \mathbb{R}_+^{n_1 \times r_1},\, H \in \mathbb{R}_+^{n_2 \times r_2},\, Q \in \mathbb{R}_+^{n_3 \times r_3},\, \mathcal{G} \in \mathbb{R}_+^{r_1 \times r_2 \times r_3}}{\text{minimize}} \quad D(\mathcal{T} \mid (W \otimes H \otimes Q)\,\mathcal{G}) \tag{1}$$

NTD identifiability

The big open question: under which conditions is NTD identifiable/essentially unique?

A few empirical observations:

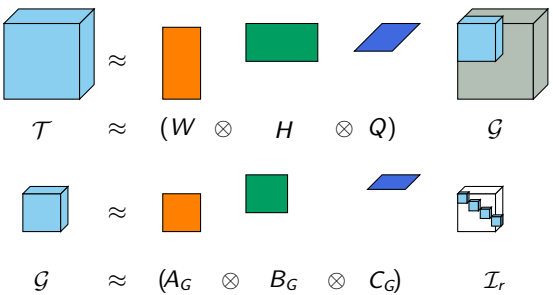
- NTD factors and core can be recovered when they are very sparse, even without explicit sparsity imposed
- Imposing sparsity helps a lot in recovering the true factors and core.
  - [Murakami,Smile 1998, Rocci,Ten Berge 2022 ...] Rotation Tucker models to maximize number of zeroes.
  - [Morup,Hansen,Arnfred 2008] Sparsity penalty imposed by default.

Existing litterature

- [Zhou,Cichocki 2014] claim to links NTD identifiability to NMF identifiability of the unfoldings.

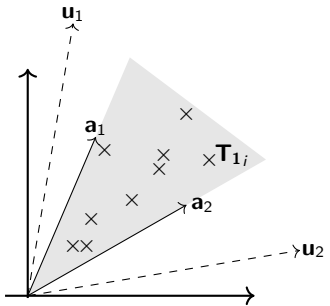
NTD for nnCANDELINC [C.2017]

CANDELINC: Tucker format then PARAFAC



Problems with nnCANDELINC

- Rank of core might increase
- Factors of $\mathcal{T}$ might not be recovered
- NTD is hard to compute anyway
- Does not work in (my) pratice



NTD for nnCANDELINC [Skau DeSantis 2022]

A few interesting concepts/facts:

- Nonnegative multilinear ranks

$$\text{rank}_+(\mathcal{T}_{[n]})$$

- Intersection of tensor cones and tensor product don't commute
- **Minimal NTD** has dimension equal to nonnegative multilinear ranks (may not exist)
- **Canonical NTD** when dimensions equal to nonnegative ranks of factors for a unique CPD tensor.

Proposition

Suppose $\mathcal{T}$ admits a unique CPD.

- Then there exists a canonical NTD which preserves its nonnegative rank.
- For any canonical NTD that preserves the rank, its factors have full nonnegative rank.

Core problem: selecting the right canonical NTD.

Back to NMF algorithms

NMF and numerical optimization

$$\underset{W \geq 0, H \geq 0}{\operatorname{argmin}} \; D(M, WH)$$

Usual loss functions:

- Frobenius loss $D(M, WH) = \|M - WH\|_F^2$
- Kullback-Leibler $D(M, WH) = \sum_{ij} KL(M_{ij}, [WH]_{ij}) = \sum_{ij} M_{ij} \log(\frac{M_{ij}}{[WH]_{ij}}) + [WH]_{ij} - M_{ij}$
- Beta-Divergence
- More exotic: Wasserstein distance [Rolet2016, Varol2019]0, ℓ_1 norm [Gillis2018] ...

A few remarks:

- Problem non-convex in general for (W, H) but "solvable" for fixed W or H.
- Beta-divergence loss is separable in columns of H (or rows of W).

This calls for block-coordinate descent methods:

- Hierarchical Alternating Least Squares (exact block-coordinate descent for ℓ_2 loss)
- Alternating Multiplicative Updates
- Alternating Proximal Gradient
- ...

NTD algorithms mimic NMF algorithms

NTD and numerical optimization

$$\underset{W\geq 0, H\geq 0, Q\geq 0, \mathcal{G}\geq 0}{\operatorname{argmin}} \; D(M, (W \otimes H \otimes Q)\,\mathcal{G})$$

Usual loss functions:

- Frobenius loss $D(M, (W \otimes H \otimes Q)\,\mathcal{G}) = \|M - (W \otimes H \otimes Q)\,\mathcal{G}\|_F^2$
- Kullback-Leibler $D(M, (W \otimes H \otimes Q)\,\mathcal{G}) = \sum_{ijk} KL(M_{ijk}, [(W \otimes H \otimes Q)\,\mathcal{G}]_{ijk})$

A few key points:

- The core update is a "vector" update (not matrix!)
- One must pay attention to update rules, to avoid computing big intermediate representations and Kronecker products.

Existing algorithms (sample):

- HALS + Proximal Gradient for $\mathcal{G}$
- Alternating MU

What about sparsity?

In the first NTD paper [Morup 2008], sparsity was already considered.

Sparsity?

Most papers impose sparsity with ℓ_1 norm.
Problem: Scale ambiguity!! For $\mu > 1$,

$$\|M - WH\|_F^2 + \lambda\|W\|_1 > \|M - \frac{1}{\mu}W\mu H\|_F^2 + \frac{\lambda}{\mu}\|W\|_1 = \|M - WH\|_F^2 + \lambda'\|W\|_1$$

with $\lambda' < \lambda$.
Consequence: minimal loss is the minimum of $\|M - WH\|_F^2$ but the minimum is not attained in general!

- Several work around for NMF
  - Constrain H on the hypersphere [Morup 2008][LeRoux2015]
  - Use a more complex sparsity metric [Hoyer2002/2004]
  - Use ℓ_2 on H [??][Roald2022] How to use in MU?
- Problems with multiple sparsity penalties?

Conclusion

Is NTD the new NMF?

Similarities between NMF and NTD

- Numerical Optimization
- Applications, to some extent
- Decomposition of data into a sum of parts
- Empirically, identifiability

Some major differences

- NTD theory requires multilinear algebra
- Almost no identifiability results available for NTD
- Connection between NTD and polytopes?
- NTD is hard to understand
- Few dedicated algorithms, e.g. efficient initialization

Tensorly ad 1: What is Tensorly

TensorLy **Open source and collaborative Python toolbox for tensors**

Contents:

- Tensor objects from Numpy, Pytorch, Tensorflow... and soon support for efficient contractions.
- Tensor manipulations (reshape, permute and so)
- Some tensor decompositions (CP, constrained CP, Generalized CP, Tucker, Nonnegative Tucker, TT, PARAFAC2, CMTF)
- Dataset loaders, visualisation tools (CP)

Code features:

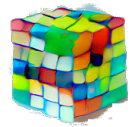
- Back-end transparent for users and devs
- Collaborative! Issues/Pull Requests with reasonable response time
- Automatic unit tests
- User guide, API, Examples at `tensorly.org`

Sparse tensor storage and sparse-dedicated algorithms are to be improved!

NTD in Tensorly

Notebook demo for NTD applied to HSI




Thank you for your attention!!