Introduction to tensors: from applications to contemporary challenges

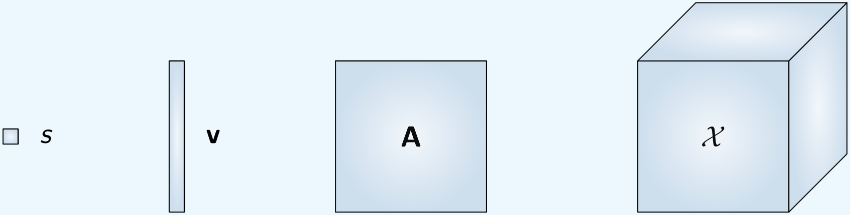
Martina Iannacito

Pysanum
Pisa, November 27, 2023

KU LEUVEN kulak
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics
erc European Research Council Established by the European Commission

Overview

From scalars to tensors



Matrix

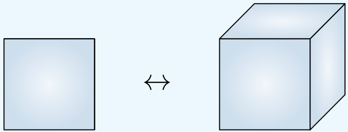
- object in $\mathbb{K}^{n_1 \times n_2}$
- set of n_2 elements in $\mathbb{K}^{n_1}$
- linear operator from $\mathbb{K}^{n_2}$ to $\mathbb{K}^{n_1}$

Tensor

- object in $\mathbb{K}^{n_1 \times \cdots \times n_d}$
- set of $(n_{i_1} \times n_{i_k})$ elements in $\mathbb{K}^{n_{j_1} \times \cdots \times n_{j_\ell}}$
- multilinear operator from $\mathbb{K}^{n_{j_1} \times \cdots \times n_{j_\ell}}$ to $\mathbb{K}^{n_{i_1} \times \cdots \times n_{i_k}}$

with $k + \ell = d$

Where and why tensors?



Examples of tensor data

- Color images, video, ...
- Text mining: term $\times$ document $\times$ author
- (Social) networks: score $\times$ object $\times$ referee $\times$ criterion
- Event Related Potential (ERP): subject $\times$ time $\times$ frequency $\times$ channel $\times \dots$
- Face recognition: people $\times$ pose $\times$ illumination $\times$ angle

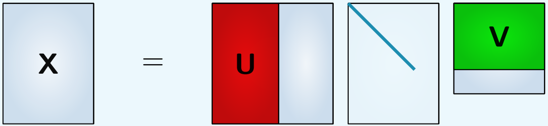
Tensor advantages

- better representation of intricate phenomena
- compression by factorization techniques
- uniqueness for some decomposition

Overview

SVD and EVD: numerical aspects

SVD: $\mathbf{X} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^{\mathsf{T}}$

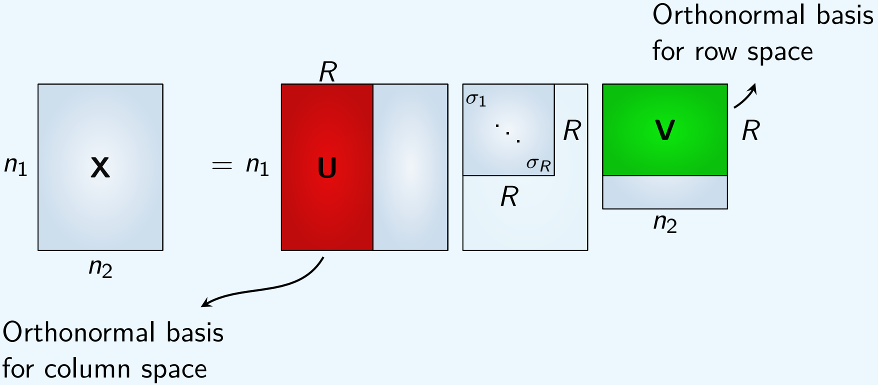


- gap between singular values

EVD: $\mathbf{Y} = \mathbf{E} \cdot \mathbf{D} \cdot \mathbf{E}^{-1}$

- does not always exist! (alg multiplicity $>$ geom multiplicity)
- gap between eigenvalues
- linear independence eigenvectors

Matrix Singular Value Decomposition



- (Numerical) rank revealing
- Best rank-r approximation $\prec$ truncation SVD
- Dominant subspace generated by first part of **U**, **V**

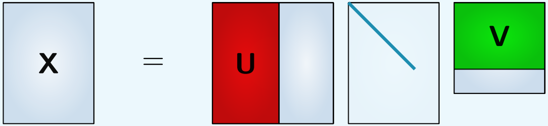
$$\mathbf{X} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^{\mathsf{T}} = \sum_{i=1}^{R} \sigma_i \mathbf{u}_i \otimes \mathbf{v}_i$$

Best low rank approximation of matrix

Matrix

$$\min_{\text{rank}(\hat{\mathbf{X}})\leqslant R} \|\mathbf{X}-\hat{\mathbf{X}}\|^2 \iff \max \|\hat{\mathbf{U}}^{\mathsf{T}}\cdot\mathbf{A}\cdot\hat{\mathbf{V}}\|^2 \qquad (\hat{\mathbf{U}},\hat{\mathbf{V}} \quad \text{orthogonal})$$

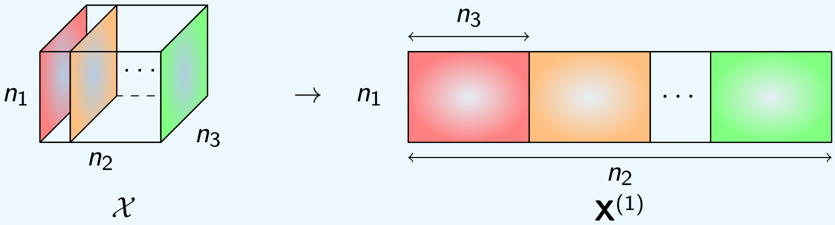
$$\|\mathbf{X}-\hat{\mathbf{X}}\|^2_{\min} = \sum_{r=R+1}^{\min(I_1,I_2)} \sigma_r^2$$



Overview

Unfolding and contraction

Let $\mathcal{X}$ be a $(n_1 \times \cdots \times n_d)$ tensor



Definition: the k-th mode matricization $\mathbf{X}^{(k)}$ is a $(n_k \times n_1 \cdots n_{k-1} n_{k+1} \cdots n_d)$ matrix, obtained stacking the vectors

$$\mathbf{x}_{i_k} = \mathrm{vec}(\mathcal{X}(\cdot, \ldots, i_k, \ldots, \cdot)).$$

Matrix-tensor product

Let $\mathcal{X}$ be a $(n_1 \times \cdots \times n_d)$ tensor. If $\mathbf{G}$ is a $(n_k \times m_k)$ matrix

Definition: the k-th mode matrix-tensor product

$$\mathcal{Y} = \mathcal{X} \times_k \mathbf{G}$$

a $(n_1 \times \cdots \times m_k \times \cdots n_d)$ such that

$$\mathcal{Y}(i_1, \ldots, j_k, \ldots, i_d) = \sum_{i_k=1}^{n_k} \mathcal{X}(i_1, \ldots, i_k, \ldots, i_d)\mathbf{G}(i_k, j_k)$$

Computational: a straightforward way of getting $\mathcal{Y}$ the matrix-tensor product is computing

$$\mathbf{Y} = \mathbf{G}^T \mathbf{X}^{(k)}$$

and then tensorizing $\mathbf{Y}$ into $\mathcal{Y}$.

Overview

Tucker's decomposition

Definition:

$$\mathcal{X} = \mathcal{C} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_d \mathbf{U}_d = (\mathbf{U}_1, \ldots, \mathbf{U}_d)\mathcal{S}$$

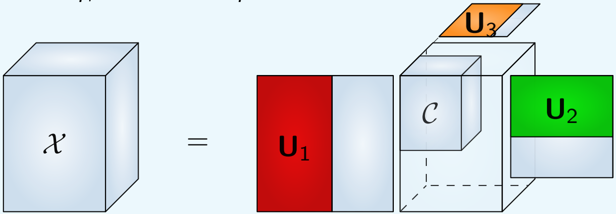
with $\mathcal{C}$ a $(R_1 \times \cdots \times R_d)$ tensor and $\mathbf{U}_i$ a $(n_i \times R_i)$ orthogonal matrix.

Remark:

- the core tensor $\mathcal{C}$ is is all-orthogonal, ordered and not diagonal;
- $\mathbf{U}_i$ is said i-th factor matrix.
- the memory cost is

$$\mathcal{O}(R^d + NR)$$

  where $R = \max R_i$, $N = \max n_i$.



[Tucker 1964; De Lathauwer, De Moor, et al. 2000a]

Multilinear rank of a tensor

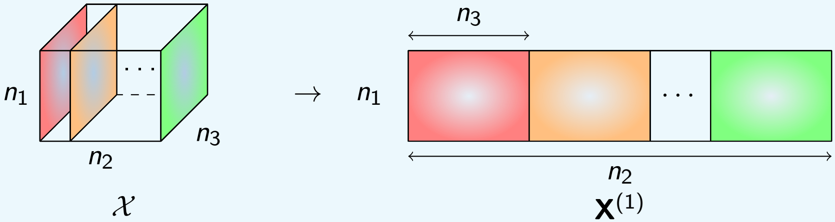
The Tucker's decomposition factorizes a tensor $\mathcal{A}$ using its8 multilinear rank
Definition: the multilinear rank $\text{rank}_{\boxplus}(\mathcal{X}) = (R_1, \ldots, R_d)$ is such that

$$R_i = \text{rank}(\mathbf{X}^{(i)}) \qquad \text{for} \qquad i = 1, \ldots, d.$$

- Surprising fact: each component of $\text{rank}_{\boxplus}$ can be different (new possibilities)
- Property:

$$R_i \leqslant \prod_{j \neq i} R_i$$

Compute Tucker's decomposition



- Mode-1 vector space (columns)/sing. values:

  SVD: $$\mathbf{X}^{(1)} = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1{}^\top$$

- Mode-2 vector space (rows)/sing. values:

  SVD: $$\mathbf{X}^{(2)} = \mathbf{U}_2 \mathbf{\Sigma}_2 \mathbf{V}_2{}^\top$$

- Mode-3 vector space/sing. values:

  SVD: $$\mathbf{X}^{(3)} = \mathbf{U}_3 \mathbf{\Sigma}_3 \mathbf{V}_3{}^\top$$

- Core tensor:

$$\mathcal{C} = \mathcal{X} \times_1 \mathbf{U}_1{}^\top \times_2 \mathbf{U}_2{}^\top \times_3 \mathbf{U}_3{}^\top$$

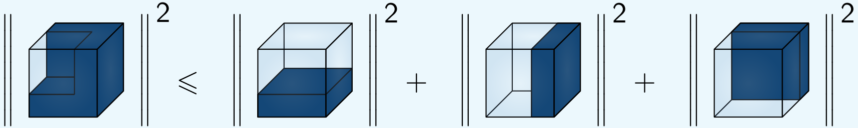
This algorithm is also called *High-Order Singular Value Decomposition*

Truncated HOSVD

What if $(R_1, \ldots, R_d)$ is not the exact multilinear rank?

Truncation error:

$$\|\mathcal{X} - \hat{\mathcal{X}}_\mathrm{T}\|^2 \leqslant \sum_{i_1=R_1+1}^{n_1} \sigma_{i_1}^2 + \sum_{i_2=R_2+1}^{n_2} \sigma_{i_2}^2 + \sum_{i_3=R_3+1}^{n_3} \sigma_{i_3}^2$$



Suboptimality: $\|\mathcal{X} - \hat{\mathcal{X}}_\mathrm{T}\|^2 \leqslant d \, \|\mathcal{X} - \hat{\mathcal{X}}\|_{\min}^2$ (order d)

Alternative: compute the multilinear approximation as

$$\hat{\mathcal{X}} = \min_{\mathrm{rank}_{\boxplus}(\mathcal{Y}) \leqslant (R_1, R_2, R_3)} \|\mathcal{X} - \mathcal{Y}\|^2$$

Algorithms for low multilinear rank approximation

HO orthogonal iteration: [Kroonenberg 1983; De Lathauwer, De Moor, et al. 2000b]

Optimization on manifolds:

- Newton [Eldén and Savas 2009; Ishteva, De Lathauwer, et al. 2009]
- Quasi-Newton [Savas and Lim 2010]
- Trust region [Ishteva, Absil, et al. 2010]
- Conjugate gradient [Ishteva, De Lathauwer, et al. 2009]

Krylov methods: [Savas and Eld'en 2013; Goreinov, I. V. Oseledets, et al. 2012]

Initialization: truncated HOSVD, random inits

Approximation vs truncation

Optimal approximation:

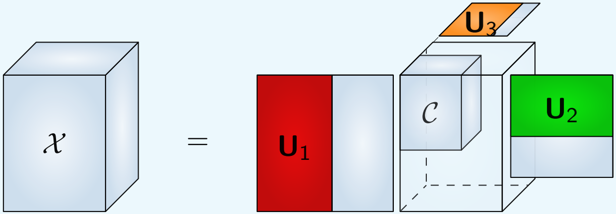
- relatively expensive
- not always necessary

Truncation:

- Truncation may suffice: multilinear rank can be increased, if higher accuracy is desired
- Precise values of $R_1, \ldots, R_N$ not important, if size subspace does not matter
- Discard small multilinear singular values
- How to choose $R_1, \ldots, R_N$?
  - for each mode separately (cf. PCA)
  - all modes together: check bound on error; compute norm residual; heuristic procedures

Numerical pros and cons

- Well-posed
- (Approximate) computation via matrix SVD
- Error bounds
- Curse not broken
- Numerically reliable but limited to modest order



Order 3 $= n^3 \to \mathcal{O}\left(3nR + R^3\right)$

Order d $= n^d \to \mathcal{O}\left(nR + R^d\right)$

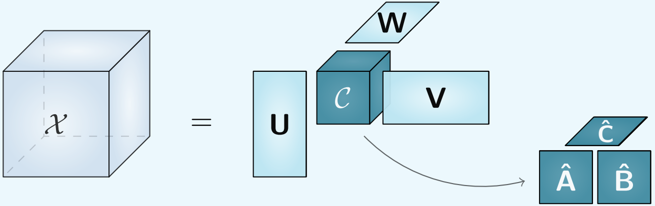
Applications of low multilinear rank approximation

Dimensionality reduction:

- acts like a compression and reduced storage
- denoising
- following computation in compressed format and then re-expansion

Subspaces:

- finds principal directions *in each mode separately*)
- distinguishes between information subspace and noise ones



Overview

Optical character recognition

- topic in **machine learning**
- recognizing and converting handwritten or printed text into electronic document
- widely used, e.g., *Oletko koskaan kokeillut Google-Translate?*

Studying example: creating a classifier, that estimates the probability a handwritten digit is a specific digit.

Workflow:

1. produce a training set, i.e., a set of handwritten digits labeled;
2. reduce the data dimensions;
3. finds features to discriminate the data;
4. choose a classification algorithm, e.g., k-nearest neighbors, SVM, NN....
5. learn the classifier parameters from the traing data
6. estimate its capacity on the test data

 $\Rightarrow$

0	1	2	3	4	5	6	7	8	9
0%	24%	5%	0%	0%	1%	0%	36%	0%	33%

 $\Rightarrow$

0	1	2	3	4	5	6	7	8	9
0%	0%	0%	0%	97%	0%	0%	0%	0%	3%

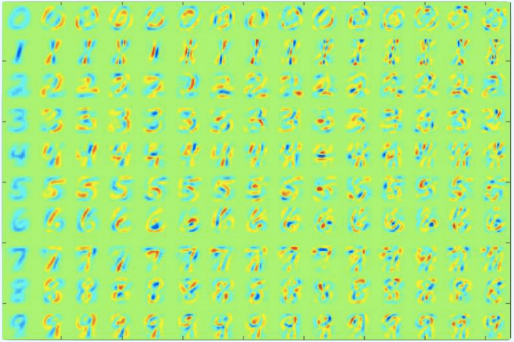
MNIST dataset

- 28 × 28 grayscale images of handwritten digits
- 60′000 training images
- 10′000 test images

Input data-set compression

Let $\mathcal{A}_\ell$ be a tensor of size $(28 \times 28 \times n_\ell)$ collecting all the handwritten images with tag $\ell = 0, \ldots, 9$. The compression step consists in

- computing the HOSVD of $\mathcal{A}_\ell$ at multilinear rank $(8, 8, 40)$
- define $\mathbf{f}_{\ell_k} = ((\mathbf{U}_1, \mathbf{U}_2, \mathbb{I})\mathcal{S})(\cdot, k)$ is the k-th element of the feature basis for $k = 1, \ldots, 40$
- pass $\{\{\mathbf{f}_{0_k}\}, \ldots, \{\mathbf{f}_{9_k}\}\}_k$ to 3-nearest neighbors classifier



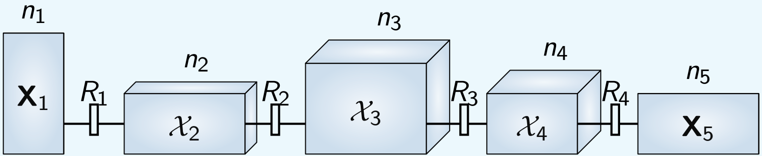
Classification results

The confusion matrix after this procedure gets

Actual class \ Predicted class	0	1	2	3	4	5	6	7	8	9
0	974	1	1	0	0	0	2	2	0	0
1	0	1132	2	0	0	0	1	0	0	0
2	6	1	1013	0	1	0	2	8	0	1
3	2	0	2	988	1	5	0	5	4	3
4	2	0	0	0	955	0	6	2	0	17
5	3	0	0	5	2	869	6	1	4	2
6	5	3	0	0	1	2	947	0	0	0
7	2	9	7	0	2	0	0	1005	0	3
8	4	0	2	5	0	3	3	4	950	3
9	3	5	1	2	6	2	1	6	1	982

Overview

Tensor Train or Matrix Product States



Let $\mathcal{X}$ a tensor of order d and dimensions $(N_1 \times \cdots \times N_d)$, then its TT-representation is given by d TT-cores s.t.

- $\mathbf{X}_1$ a (N_1, R_1) matrix
- $\mathcal{X}_i$ is a $(R_{i-1} \times N_i \times R_i)$ tensor
- $\mathbf{X}_d$ is a $(R_{d-1} \times N_d)$ matrix

i.e., a *train of matrix - third-order tensors - matrix*.

The $(i_1, \ldots, i_d)$ element of $\mathcal{X}$ is

$$\mathcal{X}(i_1, \ldots, i_d) = \sum_{i=1}^{d} \sum_{r_i=1}^{R_i} \mathbf{X}(i, r_1) \mathcal{X}_1(i_1, r_2, i_2) \cdots \mathbf{X}_d(i_{d-1}, i_d).$$

TT-decomposition idea

$$\mathcal{X}(i_1, \ldots, i_d) = \sum_{i=1}^{d} \sum_{r_i=1}^{R_i} \mathbf{X}(i, r_1)\mathcal{X}_1(i_1, r_2, i_2) \cdots \mathbf{X}_d(i_{d-1}, i_d).$$

$\{1, 2, 3, 4, 5\}$

$⓪$

$\swarrow \quad \searrow$

$\{1\} \quad \{2, 3, 4, 5\}$

$①$

$\swarrow \quad \searrow$

$\{2\} \quad \{3, 4, 5\}$

$②$

$\swarrow \quad \searrow$

$\{3\} \quad \{4, 5\}$

$③$

$\swarrow \quad \searrow$

$\{4\} \quad \{5\}$

$(n_1 \times n_2 n_3 n_4 n_5)$

$\swarrow \quad \searrow$

$(n_1 \times R_1)(R_1 \times n_2 n_3 n_4 n_5)$

$(R_1 n_2 \times n_3 n_4 n_5)$

$\swarrow \quad \searrow$

$(R_1 n_2 \times R_2)(R_2 \times n_3 n_4 n_5)$

$(R_2 n_3 \times n_4 n_5)$

$\swarrow \quad \searrow$

$(R_2 n_3 \times R_3)\ (R_3 \times n_4 n_5)$

$(R_3 n_4 \times n_5)$

$\swarrow \quad \searrow$

$(R_3 n_4 \times R_4)\ (R_4 \times n_5)$

TT-SVD [I. Oseledets 2011]

Algorithm 1: $\mathcal{X}_{\mathrm{TT}} = \texttt{TT-SVD}(\mathcal{X}, \varepsilon)$

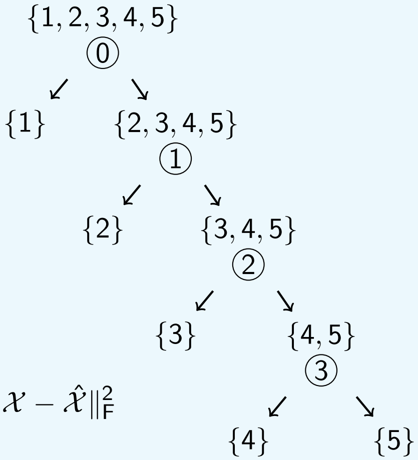
Input: $\mathcal{X}$ order d tensor, $\varepsilon \in \mathbb{R}_+$

```
δ = (ε/√(d − 1))‖𝒳‖
set 𝒞 equal to 𝒳 and R_0 = 1
for i = 1, ..., d − 1 do
    reshape 𝒞 as a matrix C of size (n_i R_{i−1} × m) with m = (∏_{j≠i} n_i)/(n_i R_{i−1})
    compute Ĉ_i = Û_i Σ̂_i V̂_i^T the SVD of C truncated at rank R_i s.t. ‖C − Ĉ_i‖ ≤ δ
    reshape Û_i as a tensor 𝒳_i of dimension (R_{i−1} × n_i × R_i)
    set C = Σ̂_i V̂_i^T
set X_d = C
```

$$
\begin{aligned}
&\delta = (\varepsilon/\sqrt{d-1})\|\mathcal{X}\| \\
&\text{set } \mathcal{C} \text{ equal to } \mathcal{X} \text{ and } R_0 = 1 \\
&\textbf{for } i = 1, \dots, d-1 \textbf{ do} \\
&\quad \text{reshape } \mathcal{C} \text{ as a matrix } \mathbf{C} \text{ of size } (n_i R_{i-1} \times m) \text{ with } m = \left(\prod_{j \neq i} n_i\right)/(n_i R_{i-1}) \\
&\quad \text{compute } \hat{\mathbf{C}}_i = \hat{\mathbf{U}}_i \hat{\boldsymbol{\Sigma}}_i \hat{\mathbf{V}}_i^{\mathsf{T}} \text{ the SVD of } \mathbf{C} \text{ truncated at rank } R_i \text{ s.t. } \|\mathbf{C} - \hat{\mathbf{C}}_i\| \leq \delta \\
&\quad \text{reshape } \hat{\mathbf{U}}_i \text{ as a tensor } \mathcal{X}_i \text{ of dimension } (R_{i-1} \times n_i \times R_i) \\
&\quad \text{set } \mathbf{C} = \hat{\boldsymbol{\Sigma}}_i \hat{\mathbf{V}}_i^{\mathsf{T}} \\
&\text{set } \mathbf{X}_d = \mathbf{C}
\end{aligned}
$$

Output: $\{\mathbf{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_{d-1}, \mathbf{X}_d\}$

TT-properties



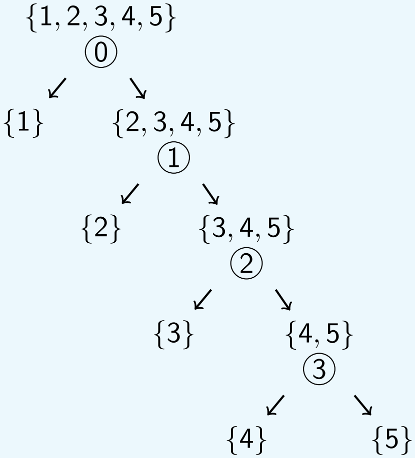
- graph structure below;
- total number of entries: $n^d \leftrightarrow$ number of variables: $\mathcal{O}\left(dnR^2\right)$, i.e., **curse broken** by means of QR/SVD
- truncation error bound

$$\|\mathcal{X} - \hat{\mathcal{X}}_{\text{TT, trunc}}\|_{\text{F}}^2 \leqslant (N-1) \min_{\text{rank}_{\text{TT}}(\hat{\mathcal{X}}) \leqslant (R_1, R_2, \ldots, R_{N-1})} \|\mathcal{X} - \hat{\mathcal{X}}\|_{\text{F}}^2$$

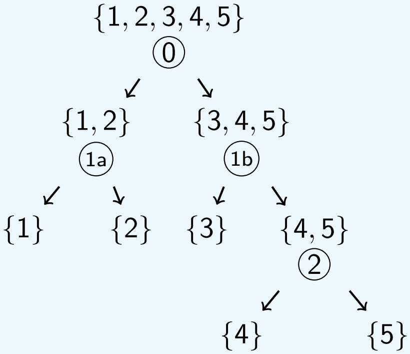
Remark: the TT-ranks increase with linear combinations of tensors in TT-format or contractions.

A different tree? Hierarchical Tucker [Hackbusch 2012; Grasedyck 2010]

TT



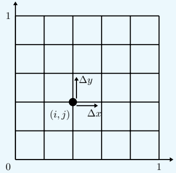
hT



Application: scientific computing

The problem

$$\begin{cases} \mathcal{L}(u) = f & \text{in} \quad \Omega \\ u = f_0 & \text{in} \quad \partial\Omega \end{cases} \quad \text{for} \quad \Omega \subseteq \mathbb{R}^{n_1 \times \cdots \times n_d}.$$



$$\boldsymbol{\mathcal{A}}\mathcal{X} = \mathcal{B}$$

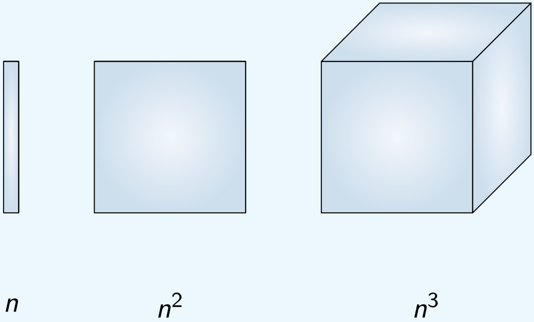
where $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a multilinear operator and $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ a tensor.

For large scale-simulations we have to take into account

- memory costs $\mathcal{O}(N^d)$
- computational model
- numerical method

Tensors and scientific computing

Scientific computing: discretization of functions in many variables: “curse of dimensionality”



Exponential increase of entries: n^d

Computation: in terms of parameterized approximation

Overview

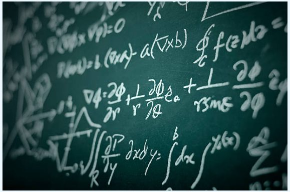
Maths vs computer science

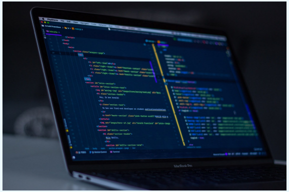
Mathematical world

- $\pi = 3.141592653589793238462643\ldots$

Computer world

```
>>> π = 3.141592653589793
```





Maths vs computer science

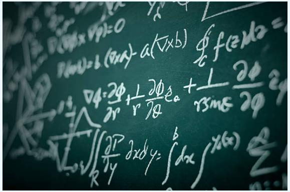
Mathematical world

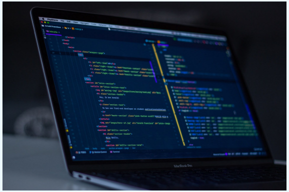
- $\pi = 3.14159265358979323846264 33\ldots$
- $x = 0.1$ and $y = 0.2$, then $x + y = 0.3$

Computer world

>>> $\overline{\pi}$ = 3.141592653589793

>>> $\overline{\text{x}}$ = 0.1 and $\overline{\text{y}}$ = 0.2, then $\overline{\text{x+y}}$ = 0.30000000000000004





Computational model

Denoting by u the **unit roundoff** of the working precision

Standard IEEE model [Higham 2002]

$$fl(x) = x(1+\xi) \qquad \text{[storage perturbation]}$$
$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1+\varepsilon) \qquad \text{[computational perturbation]}$$

with $|\xi| \le u$, $|\varepsilon| \le u$ and $\operatorname{op} \in \{+,-,\times,\div\}$.

Computational model

Denoting by u the **unit roundoff** of the working precision

Standard IEEE model [Higham 2002]

$$fl(x) = x(1+\xi) \qquad \text{[storage perturbation]}$$
$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1+\varepsilon) \qquad \text{[computational perturbation]}$$

with $|\xi| \le u$, $|\varepsilon| \le u$ and $\operatorname{op} \in \{+,-,\times,\div\}$.

Example

Assuming to work in floating point 64, with $u_{64} = 10^{-16}$

- $\overline{\pi} = 3.141592653589793 = \pi(1+\xi)$ with $|\xi| \le u_{64}$
- $\overline{\mathtt{x}} = \mathtt{0.1}$ and $\overline{\mathtt{y}} = \mathtt{0.2}$, then

$$\overline{\mathtt{x + y}} = \mathtt{0.30000000000000004} = (0.2+0.1)(1+\varepsilon)$$

with $|\varepsilon| \le u_{64}$

What happens when objects are compressed through a tensor techniques?

Assumptions when using both u the computational precision and δ a storage one

- use TT-formalism, so that storage cost is linear in d
- compress objects at precision δ
- perform operation with computational precision u

new computational framework

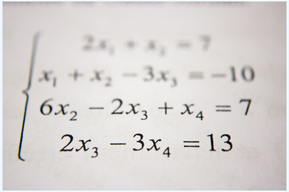
$$fl_\delta(\mathcal{X} \operatorname{op} \mathcal{Y}) = \delta\text{-storage}\big(fl(\mathcal{X} \operatorname{op} \mathcal{Y})\big)$$

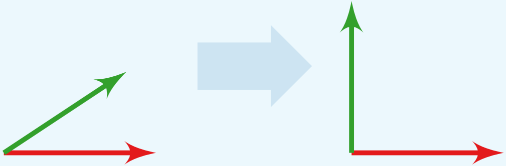
$$\delta\text{-storage}(\mathcal{Z}) = \overline{\mathcal{Z}} \qquad \text{s.t.} \qquad \frac{||\mathcal{Z} - \overline{\mathcal{Z}}||}{||\mathcal{Z}||} \leq \delta$$

with fl is the classical floating point computational function dependent on u.

Iterative solver

- Generalized Minimal RESidual (GMRES)





Orthogonalization kernels

- Classical and Modified Gram-Schmidt (CGS, MGS)
- Gram approach
- Householder transformation

Orthogonalization schemes

Let $\mathbf{Q}_k = [\mathbf{q}_1, \ldots, \mathbf{q}_k]$ be the orthogonal basis produced by an orthogonalization kernel, then the **Loss Of Orthogonality** is

$$||\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k||.$$

It measures the quality in terms of orthogonality of the computed basis. It is linked with the linearly dependency of the input vectors $\mathbf{A}_k = [\mathbf{a}_1, \ldots, \mathbf{a}_k]$, estimated through $\kappa(\mathbf{A}_k)$.

Matrix		
Source	*Algorithm*	$\left\|\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k\right\|$
[Stathopoulos and Wu 2002]	**Gram**	$\mathcal{O}(u\kappa^2(\mathbf{A}_k))$
[L. Giraud, Langou, et al. 2005]	**CGS**	$\mathcal{O}(u\kappa^2(\mathbf{A}_k))$
[Björck 1967]	**MGS**	$\mathcal{O}(u\kappa(\mathbf{A}_k))$
[L. Giraud, Langou, et al. 2005]	**CGS2**	$\mathcal{O}(u)$
[L. Giraud, Langou, et al. 2005]	**MGS2**	$\mathcal{O}(u)$
[Wilkinson 1965]	**Householder**	$\mathcal{O}(u)$

Classical and Modified Gram-Schmidt

Algorithm 2: $\boldsymbol{\mathcal{Q}}, \mathbf{R} = \texttt{TT-CGS}(\boldsymbol{\mathcal{A}}, \delta)$

Input: $\boldsymbol{\mathcal{A}} = [\mathcal{A}_1, \ldots, \mathcal{A}_m]$, $\delta \in \mathbb{R}_+$

for $i = 1, \ldots, m$ **do**
- $\mathcal{P} = \mathcal{A}_i$
- **for** $j = 1, \ldots, i-1$ **do**
  - $\mathbf{R}(i,j) = \langle \mathcal{A}_i, \mathcal{Q}_j \rangle$
  - $\mathcal{P} = \mathcal{P} - \mathbf{R}(i,j)\mathcal{Q}_j$
- $\mathcal{P} = \texttt{TT-rounding}(\mathcal{P}, \delta)$
- $\mathbf{R}(i,i) = ||\mathcal{P}||$
- $\mathcal{Q}_i = \mathcal{P}/\mathbf{R}(i,i)$

Output: $\boldsymbol{\mathcal{Q}} = \{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$, $\mathbf{R}$

Algorithm 3: $\boldsymbol{\mathcal{Q}}, \mathbf{R} = \texttt{TT-MGS}(\boldsymbol{\mathcal{A}}, \delta)$

Input: $\boldsymbol{\mathcal{A}} = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$, $\delta \in \mathbb{R}_+$

for $i = 1, \ldots, m$ **do**
- $\mathcal{P} = \mathcal{A}_i$
- **for** $j = 1, \ldots, i-1$ **do**
  - $\mathbf{R}(i,j) = \langle \mathcal{P}, \mathcal{Q}_j \rangle$
  - $\mathcal{P} = \mathcal{P} - \mathbf{R}(i,j)\mathcal{Q}_j$
- $\mathcal{P} = \texttt{TT-rounding}(\mathcal{P}, \delta)$
- $\mathbf{R}(i,i) = ||\mathcal{P}||$
- $\mathcal{Q}_i = \mathcal{P}/\mathbf{R}(i,i)$

Output: $\boldsymbol{\mathcal{Q}} = \{\mathcal{Q}_1, \ldots, \mathcal{Q}_m\}$, $\mathbf{R}$

They readily write in TT-format.

Gram approach

Let $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]$, then we look for $\mathbf{A} = \mathbf{QR}$ with $\mathbf{Q}^\top\mathbf{Q} = \mathbb{I}_m$
compute the Gram matrix

$$\mathbf{A}^\top\mathbf{A} = (\mathbf{R}^\top\mathbf{Q}^\top)\mathbf{QR} = \mathbf{R}^\top\mathbf{R}$$

this is (almost) the **Cholesky** factorization of $A^\top A$ that can be written as

$$\mathbf{A}^\top\mathbf{A} = \mathbf{R}^\top\mathbf{R} = \mathbf{LL}^\top$$

with the Cholesky factor $\mathbf{L} = \mathbf{R}^\top$ and then $\mathbf{Q}$ gets

$$\mathbf{Q} = \mathbf{AR}^{-1} = \mathbf{A}(\mathbf{L}^\top)^{-1}$$

Gram approach

Algorithm 4: $\mathcal{Q}, \mathbf{R} = \texttt{TT-Gram}(\mathcal{A}, \delta)$

Input: $\mathcal{A} = \{\mathcal{A}_1, \dots, \mathcal{A}_m\}$, $\delta \in \mathbb{R}_+$

```
G be the Gram matrix from 𝓐
L = cholesky(G)
Q = {𝒬_1, ..., 𝒬_d} from 𝓐 and (L^T)^{-1}
for i = 1, ..., m do
    𝒬_i = δ-storage(𝒬_i)
```

Output: $\mathcal{Q} = \{\mathcal{Q}_1, \dots, \mathcal{Q}_m\}$, $\mathbf{R}$

In TT-format the following modifications occur

- $\mathbf{G}(i,j) = \langle \mathcal{A}_i, \mathcal{A}_j \rangle$
- $L^\top$ inverse is explicitly computed
- $\mathcal{Q}_i$ is constructed as a linear combination of $\mathcal{A}$ elements
- `TT-rounding` is used to compress at precision δ

Householder transformation

Given a vector $\mathbf{x} \in \mathbb{R}^n$ and a direction $\mathbf{y} \in \mathbb{R}^n$, the Householder reflector $\mathbf{H}$ reflects $\mathbf{x}$ along $\mathbf{y}$, i.e.,

$$\mathbf{H}\mathbf{x} = ||\mathbf{x}||\mathbf{y} \qquad \text{with} \qquad ||\mathbf{y}|| = 1.$$

Thanks to its properties, $\mathbf{H}$ writes as

$$\mathbf{H} = \mathbb{I}_n - \frac{2}{||\mathbf{u}||^2}\mathbf{u} \otimes \mathbf{u} \qquad \text{with} \qquad \mathbf{u} = (\mathbf{x} - ||\mathbf{x}||\mathbf{y}).$$

The practical implementation of the Householder transformation kernel uses the components of the input vectors.

Remark

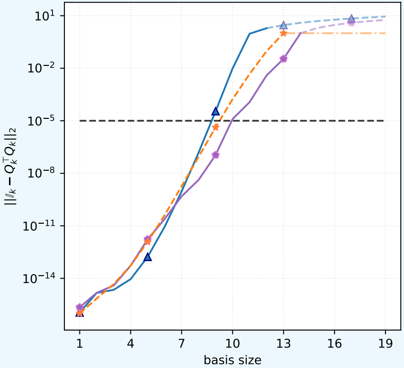
The Householder algorithm does **not** readily apply to tensor in TT-formats because of the compressed nature of this format.

TT-orthogonalization

$$\mathcal{X}_{k+1} = \texttt{TT-rounding}(\boldsymbol{\Delta}_d \mathcal{A}_k, \texttt{max\_rank} = 1) \quad \text{with} \quad \mathcal{A}_{k+1} = \frac{1}{||\mathcal{X}_{k+1}||}\mathcal{X}_{k+1}$$

TT-orthogonalization

$$\mathcal{X}_{k+1} = \texttt{TT-rounding}(\mathbf{\Delta}_d \mathcal{A}_k, \texttt{max\_rank} = 1) \quad \text{with} \quad \mathcal{A}_{k+1} = \frac{1}{||\mathcal{X}_{k+1}||}\mathcal{X}_{k+1}$$



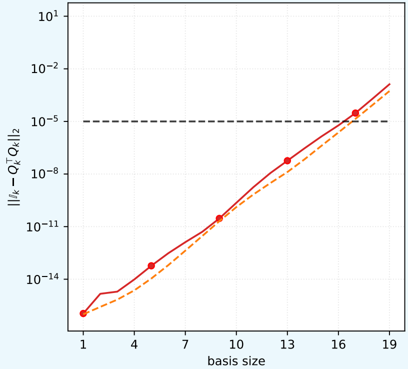
- **Gram approach**
- **CGS**
- $\kappa^2(\mathbf{A}_k)$

$\mathcal{O}(\delta\kappa^2(\mathbf{A}_k))$

Figure: Loss of orthogonality for $m = 20$ TT-vectors of order $d = 3$ and mode size $n = 15$, rounding precision $\delta = 10^{-5}$

TT-orthogonalization

$$\mathcal{X}_{k+1} = \texttt{TT-rounding}(\mathbf{\Delta}_d \mathcal{A}_k, \texttt{max\_rank} = 1) \quad \text{with} \quad \mathcal{A}_{k+1} = \frac{1}{||\mathcal{X}_{k+1}||}\mathcal{X}_{k+1}$$



- **MGS**
- $\kappa(\mathbf{A}_k)$

$\mathcal{O}(\delta\kappa(\mathbf{A}_k))$

Figure: Loss of orthogonality for $m = 20$ TT-vectors of order $d = 3$ and mode size $n = 15$, rounding precision $\delta = 10^{-5}$

TT-orthogonalization

$$\mathcal{X}_{k+1} = \texttt{TT-rounding}(\boldsymbol{\Delta}_d \mathcal{A}_k, \texttt{max\_rank} = 1) \quad \text{with} \quad \mathcal{A}_{k+1} = \frac{1}{||\mathcal{X}_{k+1}||} \mathcal{X}_{k+1}$$

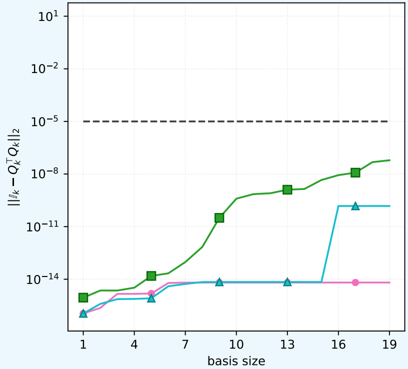


Figure: Loss of orthogonality for $m = 20$ TT-vectors of order $d = 3$ and mode size $n = 15$, rounding precision $\delta = 10^{-5}$

- **CGS2**
- **MGS2**
- **Householder transformation**

$\mathcal{O}(\delta)$

TT-orthogonalization

$$\mathcal{X}_{k+1} = \texttt{TT-rounding}(\mathbf{\Delta}_d \mathcal{A}_k, \texttt{max\_rank} = 1) \quad \text{with} \quad \mathcal{A}_{k+1} = \frac{1}{||\mathcal{X}_{k+1}||}\mathcal{X}_{k+1}$$

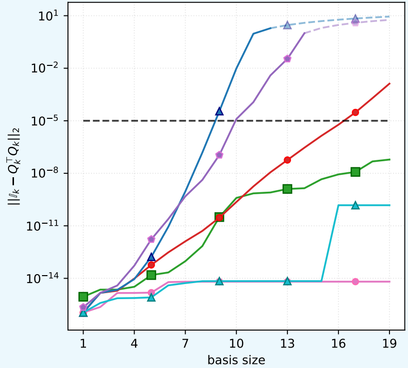


Figure: Loss of orthogonality for $m = 20$ TT-vectors of order $d = 3$ and mode size $n = 15$, rounding precision $\delta = 10^{-5}$

- **Gram approach** $\mathcal{O}(\delta\kappa^2(\mathbf{A}_k))$
- **CGS** $\mathcal{O}(\delta\kappa^2(\mathbf{A}_k))$
- **MGS** $\mathcal{O}(\delta\kappa(\mathbf{A}_k))$
- **CGS2** $\mathcal{O}(\delta)$
- **MGS2** $\mathcal{O}(\delta)$
- **Householder transformation** $\mathcal{O}(\delta)$

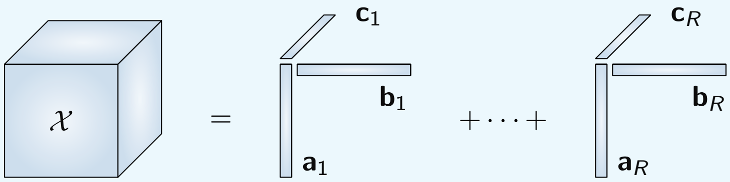
Loss of orthogonality: matrix vs tensor [Coulaud, Luc Giraud, et al. 2022]

	Matrix, theoretical	**TT-format, conjecture**
Algorithm	$\|\|\mathbb{I}_k - \mathbf{Q}_k^\top \mathbf{Q}_k\|\|$	$\|\|\mathbb{I}_k - \boldsymbol{\mathcal{Q}}_k^\top \boldsymbol{\mathcal{Q}}_k\|\|$
Gram	$\mathcal{O}(u\kappa^2(\mathbf{A}_k))$	$\mathcal{O}(\delta\kappa^2(\boldsymbol{\mathcal{A}}_k))$
CGS	$\mathcal{O}(u\kappa^2(\mathbf{A}_k))$	$\mathcal{O}(\delta\kappa^2(\boldsymbol{\mathcal{A}}_k))$
MGS	$\mathcal{O}(u\kappa(\mathbf{A}_k))$	$\mathcal{O}(\delta\kappa(\boldsymbol{\mathcal{A}}_k))$
CGS2	$\mathcal{O}(u)$	$\mathcal{O}(\delta)$
MGS2	$\mathcal{O}(u)$	$\mathcal{O}(\delta)$
Householder	$\mathcal{O}(u)$	$\mathcal{O}(\delta)$

Overview

Canonical polyadic decomposition [F. L. Hitchcock 1927; Richard A. Harshman 1970; Carroll and J.-J. Chang 1970]

Definition: decomposition in minimal number of rank-1 terms [R. A. Harshman 1970; Carrol and J. J. Chang 1970]

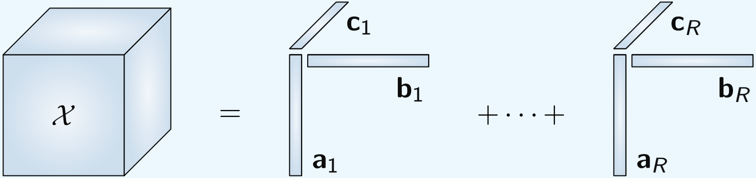


Surprising fact: unique under mild conditions on number of terms and differences between terms

Additional constraints such as orthogonality, triangularity, ... are not required, but may be imposed.

Uniqueness

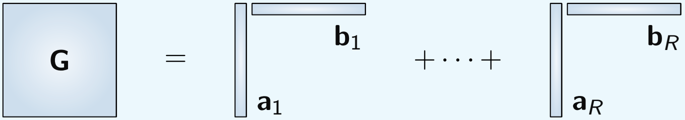
Trivial indeterminacies: permutation and scaling



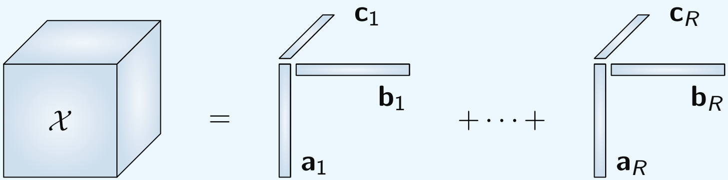
Scaling: $\mathbf{a}_r \leftarrow \mathbf{a}_r \cdot \alpha_r \qquad \mathbf{b}_r \leftarrow \mathbf{b}_r \cdot \beta_r \qquad \mathbf{c}_r \leftarrow \mathbf{c}_r \cdot \alpha_r^{-1} \cdot \beta_r^{-1}$

Rank of a tensor

- The rank R of a matrix $\mathbf{X}$ is minimal number of rank-1 matrices that yield $\mathbf{X}$ in a linear combination.



- The rank R of an Nth-order tensor $\mathcal{X}$ is the minimal number of rank-1 tensors that yield $\mathcal{X}$ in a linear combination.



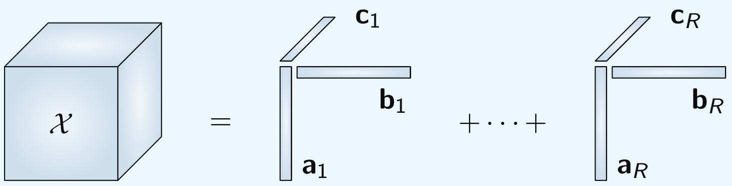
[F. Hitchcock 1927]

Rank and dimension

Remark 1: $(n \times n \times \ldots \times n)$: rank can $> n$ (new possibilities)

Remark 2: expected rank $> n$

Remark 3: is NP-hard [Håstad 1990]



Partial explanation: number of free tensor parameters: n^d

number of parameters in expansion: dnR

Rank and multilinear rank: $R \geqslant \max(R_1, R_2, \ldots, R_N)$

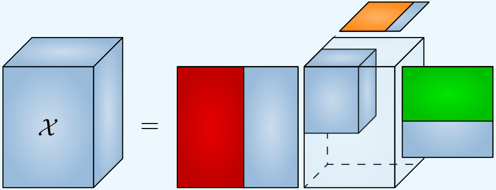
Tucker vs CPD

exact Tucker's decomposition

matrix SVD

low multilinear rank approximation

numerically reliable

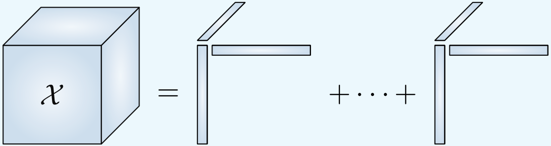


exact CPD

matrix EVD

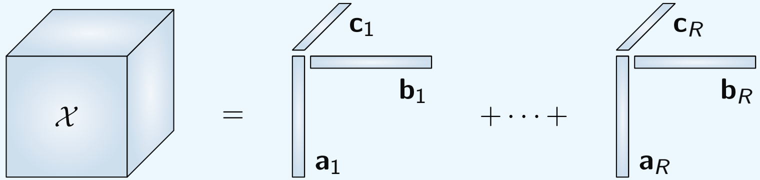
low rank approximation

numerically less reliable



CPD / low rank approximation: numerical pros and cons

- Possibly ill-posed
- Possibly ill-conditioned
- Curse broken
- Powerful tool but not always numerically reliable



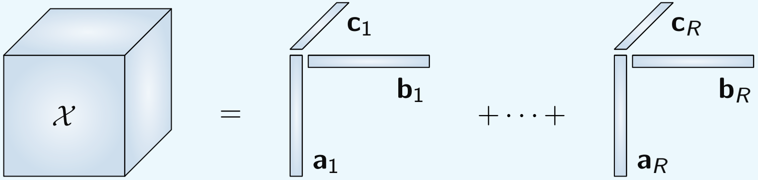
Order 3 $= n^3 \to \mathcal{O}(3nR)$
Order d $= n^d \to \mathcal{O}(dnR)$

Algorithm basics: CPD using ALS

$$\text{Step } k, \text{substep 1}: \quad \min_{\mathbf{A}_k} \frac{1}{2} \left\| \mathbf{X}^{(1)} - \mathbf{A}_k (\mathbf{C}_{k-1} \odot \mathbf{B}_{k-1})^{\mathsf{T}} \right\|_F^2$$

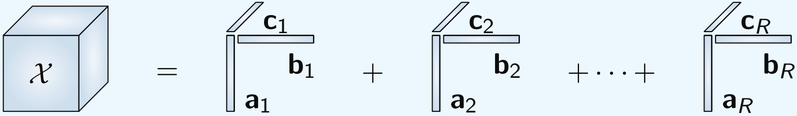
$$\text{Step } k, \text{substep 2}: \quad \min_{\mathbf{B}_k} \frac{1}{2} \left\| \mathbf{X}^{(2)} - \mathbf{B}_k (\mathbf{C}_{k-1} \odot \mathbf{A}_k)^{\mathsf{T}} \right\|_F^2$$

$$\text{Step } k, \text{substep 3}: \quad \min_{\mathbf{C}_k} \frac{1}{2} \left\| \mathbf{X}^{(3)} - \mathbf{C}_k (\mathbf{B}_k \odot \mathbf{A}_k)^{\mathsf{T}} \right\|_F^2$$



Pencil-based computation: numerical implication

CPD:



(G)EVD:

$$\mathcal{X}(:,:,1)\mathcal{X}(:,:,2){-1} = \begin{bmatrix}\mathbf{a}_1 & \dots & \mathbf{a}_R\end{bmatrix} \begin{bmatrix} c_{11}/c_{21} & & \\ & \ddots & \\ & & c_{1R}/c_{2R} \end{bmatrix} \begin{bmatrix}\mathbf{a}_1 & \dots & \mathbf{a}_R\end{bmatrix}^{-1}$$

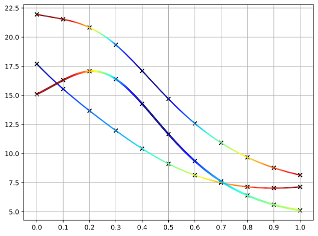
Algebraically equivalent but computational differences

- init optimization algorithm
- quantization noise $\rightarrow$ condition number [Beltrán, Breiding, et al. 2019]

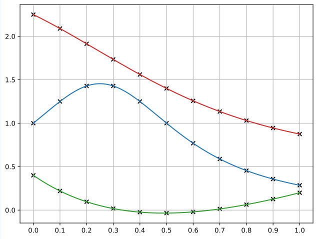
CPD structure is collapsed into matrix pencil

Overview

Blind Source Separation problem



$$= \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix} \times$$

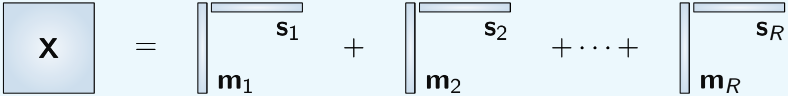


$$\underbrace{\mathbf{X}}_{\text{Known}} = \underbrace{\mathbf{M}}_{\text{Unknown}} \underbrace{\mathbf{S}^T}_{\text{Unknown}}$$

Factor Analysis and Blind Source Separation

- Decompose a data matrix in rank-1 terms that can be interpreted
  E.g. statistics, telecommunication, biomedical applications, chemometrics, data analysis, . . .

$$\mathbf{X} = \mathbf{M}\mathbf{S}^{\top}$$



- **M**: mixing matrix **S**: source signals
- Matrix decomposition in rank-1 terms is not unique!

$$\mathbf{X} = (\mathbf{M}\mathbf{G})(\mathbf{G}^{-1}\mathbf{S}^{\top}) = \tilde{\mathbf{M}}\tilde{\mathbf{S}}^{\top}$$

Additional hypothesis [Domanov and De Lathauwer 2016]

It is assumed that the columns of **S** are values of the rational function

$$\mathbf{t} : \mathbf{x} \to \begin{bmatrix} \frac{p_1}{q_1}(\mathbf{x}) & \dots & \frac{p_N}{q_N}(\mathbf{x}) \end{bmatrix}^T.$$

⇓

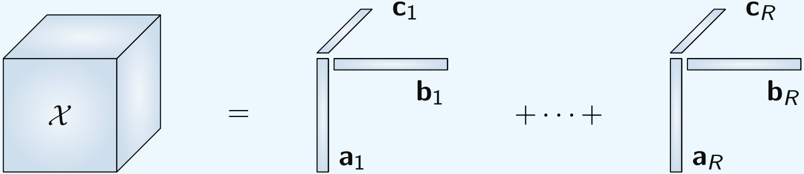
The columns of **S** belong to an algebraic variety $\mathcal{V}$ which is described by a finite system of polynomials $\{P_k\}_{k=1}^K$

$$\mathcal{V} = \Big\{(z_1, \dots, z_N) \in \mathbb{C}^N : P_k(z_1, \dots, z_N) = 0\Big\}.$$

Overview

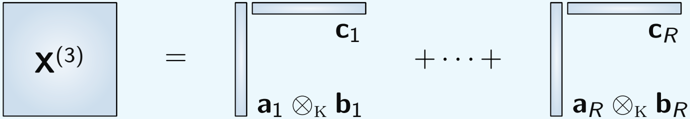
CPD reformulation

if $\mathcal{X} = \mathbf{a}_1 \otimes \mathbf{b}_1 \otimes \mathbf{c}_1 + \ldots + \mathbf{a}_R \otimes \mathbf{b}_R \otimes \mathbf{c}_R$



---

then $\mathbf{X}^{(3)} = (\mathbf{a}_1 \otimes_{\mathrm{K}} \mathbf{b}_1) \otimes \mathbf{c}_1^T + \ldots + (\mathbf{a}_R \otimes_{\mathrm{K}} \mathbf{b}_R) \otimes \mathbf{c}_R^T$



$$(\mathbf{a}_r \otimes_{\mathrm{K}} \mathbf{b}_r) \in \mathcal{V} = \left\{ \mathrm{vec}(\mathbf{Z}) : \begin{vmatrix} z_{i_1 j_1} & z_{i_1 j_2} \\ z_{i_2 j_1} & z_{i_2 j_2} \end{vmatrix} = 0 \right\} \text{ algebraic variety}$$

Algebraic algorithm: high view

Let $\mathcal{X}$ be a $(N_1 \times N_2 \times R)$ tensor, then

$$\mathbf{X}^{(3)} = \sum_{r=1}^{R} (\mathbf{a}_r \otimes_{\mathrm{K}} \mathbf{b}_r) \otimes \mathbf{c}_r^T = (\mathbf{A} \odot \mathbf{B})\mathbf{C}^T.$$

If $\mathbf{X} = \left(\mathbf{X}^{(3)}\right)^T$, then

$$\underbrace{\boxed{\quad \mathbf{X} \quad}}_{\text{Known}} = \underbrace{\boxed{\ \mathbf{C}\ }}_{\text{Unknown}} \ \underbrace{\boxed{\quad (\mathbf{A} \odot \mathbf{B})^T \quad}}_{\text{Unknown}}$$

1. compute $\mathbf{C}^{-1}$ from $\mathbf{X}$ using algebraic geometry properties;
2. compute $(\mathbf{A} \odot \mathbf{B})$ as the transposed product of $\mathbf{C}^{-1}\mathbf{X}$;
3. factorize $(\mathbf{A} \odot \mathbf{B}) = [\mathbf{a}_1 \otimes_{\mathrm{K}} \mathbf{b}_1, \ldots, \mathbf{a}_R \otimes_{\mathrm{K}} \mathbf{b}_R]$ to recover $\mathbf{A}$ and $\mathbf{B}$;
4. compute $\mathbf{C}$ by solving $(\mathbf{A} \odot \mathbf{B})\mathbf{C} = \mathbf{X}$.

Using algebraic geometry I

$\mathbf{e}$ is a column of $\mathbf{C}^{-1}$ if and only if $\mathbf{X}^T\mathbf{e}$ is equal to a column of $(\mathbf{A} \odot \mathbf{B})$

$$\Downarrow$$

$$\mathbf{X}^T\mathbf{e} = (\mathbf{x}_1^T\mathbf{e}, \ldots, \mathbf{x}_N^T\mathbf{e}) = (z_1, \ldots, z_N) \in \mathcal{V}$$

$$\Updownarrow$$

$$P_k(\mathbf{x}_1^T\mathbf{e}, \ldots, \mathbf{x}_N^T\mathbf{e}) = 0 \text{ for } k = 1, \ldots, K$$

$$\Updownarrow$$

$$P_k^{\otimes}(\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T)(\mathbf{e} \otimes \cdots \otimes \mathbf{e}) = 0 \text{ for } k = 1, \ldots, K$$

where $P_k^{\otimes}(\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T)$ is the vector obtained by formal substitution of $(z_1, \ldots, z_N)$ by $\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T$ and the scalar multiplication by the tensor product.

Using algebraic geometry II

$\mathbf{e}$ is a column of $\mathbf{C}^{-1}$ if and only if $\mathbf{X}^T\mathbf{e}$ is equal to a column of $\mathbf{A} \odot \mathbf{B}$

⇓

$$P_k^{\otimes}(\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T)(\mathbf{e} \otimes \cdots \otimes \mathbf{e}) = 0 \text{ for } k = 1, \ldots, K$$

⇕

$$\mathbf{Q}\text{vec}(\mathbf{e}^{\otimes d}) = \begin{bmatrix} P_1^{\otimes}(\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T) \\ \vdots \\ P_K^{\otimes}(\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T) \end{bmatrix} \text{vec}(\mathbf{e} \otimes \cdots \otimes \mathbf{e}) = 0$$

⇕

The columns of $\mathbf{C}^{-1}$ belong to the intersection of $\mathbf{Q}$ kernel and $\text{vec}(\text{Sym}_R^N)$ the subspace of vectorized order N symmetric tensors, i.e.,

$$\mathbf{e} \in \text{null}(\mathbf{Q}) \cap \text{vec}(\text{Sym}_R^d).$$

Algebraic algorithm outline

$$\underbrace{\mathbf{X}}_{\text{Known}} = \underbrace{\mathbf{C}}_{\text{Unknown}} \underbrace{(\mathbf{A} \odot \mathbf{B})^T}_{\text{Unknown}}$$

1. compute the factor matrix $\mathbf{C}^{-1}$ from $\mathbf{X}$;
   1.1 compute $\mathbf{Q}$;
   1.2 compute the space $\mathcal{E}_0 = \text{null}(\mathbf{Q}) \cap \text{vec}(\text{Sym}_R^d)$
      1.2.1 if dim $\mathcal{E}_0 = R$, then compute $\mathbf{C}^{-1}$ by a CPD of $\{\mathbf{e}_1^{\otimes d}, \dots, \mathbf{e}_R^{\otimes d}\}$ basis of $\mathcal{E}_0$;
      1.2.2 if dim $\mathcal{E}_0 > R$, then compute $\mathcal{E}_{h+1}$ such that

$$\mathcal{E}_{h+1} = (\mathbb{K} \otimes \mathcal{E}_h) \cap \text{vec}(\text{Sym}_R^{d+h})$$

      until dim $\mathcal{E}_{h+1} = R^{h+1}$ and go to step `1.2.1`;
2. compute $(\mathbf{A} \odot \mathbf{B})$ as $\mathbf{C}^{-1}\mathbf{X}$ transposed;
3. factorize each column of $(\mathbf{A} \odot \mathbf{B})$ at rank-1 to retrieve $\mathbf{A}$ and $\mathbf{B}$ by SVD;
4. compute $\mathbf{C}$ solving $(\mathbf{A} \odot \mathbf{B})^T\mathbf{C} = \mathbf{X}$.

Challenges

- efficiently construct $\mathbf{Q}$ and its kernel [Domanov and De Lathauwer 2013]
- estimate the dimension of the intersection with Sym_R^{d+h}
- efficiently construct a basis for $\mathfrak{E}_h$
- compute the CPD of $\{\mathbf{e}_1^{\otimes(h+d)}, \dots, \mathbf{e}_d^{\otimes(h+d)}\}$
- estimate the quality of the algorithm and its robustness

Overview

Wrap up

Tensor methods used in

- data analysis problem as compression methods
  - by the Tucker's decomposition
- scientific computing as new policy for computational methods
  - by the Tensor-Train decomposition
- signal processing
  - by the Canonical Polyadic Decomposition

Research

General theme: tensor tools for mathematical engineering:

- algebraic foundations
- numerical algorithms and software
- signal processing/data analysis/machine learning/modelling/...: concepts
- specific applications: array processing, telecom, biomedical applications, materials science, chemical science, ...

Thank you for the attention!
Questions?

References I

- Beltrán, Carlos, Paul Breiding, and Nick Vannieuwenhoven (Jan. 2019). "Pencil-Based Algorithms for Tensor Rank Decomposition are not Stable". In: *SIAM J. Matrix Anal. Appl.* 40.2, pp. 739–773.
- Björck, Åke (Mar. 1967). "Solving linear least squares problems by Gram-Schmidt orthogonalization". In: *BIT Numerical Mathematics* 7.1, pp. 1–21.
- Carrol, J. D. and J. J. Chang (Sept. 1970). "Analaysis of individual differences in multidimensional scaling via an *n*-way generalization of "Eckart–Young" decomposition". In: *Psychometrika* 35.3, pp. 283–319.
- Carroll, J. Douglas and Jih-Jie Chang (Sept. 1970). "Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition". In: *Psychometrika* 35.3, pp. 283–319.

References II

- Coulaud, Olivier, Luc Giraud, and Martina Iannacito (Nov. 2022). *On some orthogonalization schemes in Tensor Train format*. Tech. rep. RR-9491. Inria Bordeaux - Sud-Ouest.
- De Lathauwer, L., B. De Moor, and J. Vandewalle (Apr. 2000a). "A multilinear singular value decomposition". In: *SIAM J. Matrix Anal. Appl.* 21.4, pp. 1253–1278.
- De Lathauwer, L., B. De Moor, and J. Vandewalle (July 2000b). "On the best rank-1 and rank-$(R_1, R_2, \cdots, R_N)$ approximation of higher-order tensors". In: *SIAM J. Matrix Anal. Appl.* 21.4, pp. 1324–1342.
- Domanov, I. and L. De Lathauwer (July 2013). "On the uniqueness of the canonical polyadic decomposition of third-order tensors — Part I: Basic results and uniqueness of one factor matrix". In: *SIAM J. Matrix Anal. Appl.* 34.3, pp. 855–875.

References III

— (June 2016). "Generic Uniqueness of a Structured Matrix Factorization and Applications in Blind Source Separation". In: *IEEE J. Sel. Topics Signal Process.* 10.4, pp. 701–711.

Eldén, L. and B. Savas (2009). "A Newton–Grassmann Method for Computing the Best Multilinear Rank-(r_1, r_2, r_3) Approximation of a Tensor". In: *SIAM J. Matrix Anal. Appl.* 31.2, pp. 248–271.

Giraud, L., J. Langou, and M. Rozloznik (2005). "The loss of orthogonality in the Gram-Schmidt orthogonalization process". In: *Computers & Mathematics with Applications* 50.7. Numerical Methods and Computational Mechanics, pp. 1069–1075.

Goreinov, S. A., I. V. Oseledets, and D. V. Savostyanov (2012). "Wedderburn rank reduction and Krylov subspace method for tensor approximation. Part 1 Tucker case". In: *SIAM Journal on Scientific Computing* 34.1, A1–A27.

References IV

- Grasedyck, Lars (Jan. 2010). "Hierarchical Singular Value Decomposition of Tensors". In: *SIAM J. Matrix Anal. Appl.* 31.4, pp. 2029–2054.
- Hackbusch, W. (2012). *Tensor Spaces and Numerical Tensor Calculus*. Vol. 42. New York, NY, USA: Springer Science & Business Media.
- Harshman, R. A. (1970). "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis". In: *UCLA Working Papers in Phonetics* 16, pp. 1–84.
- Harshman, Richard A. (1970). "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-model factor analysis". In: vol. 16. UCLA Working Papers in Phonetics. University Microfilms, Ann Arbor, Michigan, pp. 1–84.
- Håstad, J. (1990). "Tensor Rank is NP-complete". In: *Journal of Algorithms* 11.4, pp. 644–654.

References V

- Higham, N.J. (2002). *Accuracy and Stability of Numerical Algorithms: Second Edition*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics.
- Hitchcock, F.L. (1927). "The Expression of a Tensor or a Polyadic as a Sum of Products". In: *J. Math. Phys.* 6, pp. 164–189.
- Hitchcock, Frank L. (1927). "The Expression of a Tensor or a Polyadic as a Sum of Products". In: *Journal of Mathematics and Physics* 6.1-4, pp. 164–189.
- Ishteva, M. et al. (June 2009). "Differential-geometric Newton method for the best rank-(R_1, R_2, R_3) approximation of tensors". In: *Numerical Algorithms* 51.2, pp. 179–194.

References VI

- Ishteva, M. et al. (2010). "On the Best Low Multilinear Rank Approximation of Higher-order Tensors". In: *Recent Advances in Optimization and its Applications in Engineering*. Ed. by M. Diehl et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 145–164.
- Kroonenberg, P. M. (1983). "Three-Mode Principal Component Analysis Theory and Applications". Ph.D. dissertation. Leiden University.
- Oseledets, I. (Sept. 2011). "Tensor-train decomposition". In: *SIAM J. Sci. Comput.* 33.5, pp. 2295–2317.
- Savas, B. and L. Eld'en (2013). "Krylov-type methods for tensor computations I". In: *Linear Algebra Appl.* 438.2. Tensors and Multilinear Algebra, pp. 891–918.
- Savas, B. and L.-H. Lim (2010). "Quasi-Newton Methods on Grassmannians and Multilinear Approximations of Tensors". In: *SIAM J. Sci. Comput.* 32.6, pp. 3352–3393.

- Stathopoulos, Andreas and Kesheng Wu (2002). "A Block Orthogonalization Procedure with Constant Synchronization Requirements". In: *SIAM Journal on Scientific Computing* 23.6, pp. 2165–2182.
- Tucker, L. R. (1964). "The extension of factor analysis to three-dimensional matrices". In: *Contributions to mathematical psychology*. Ed. by H. Gulliksen and N. Frederiksen. New York: Holt, Rinehart & Winston, pp. 110–127.
- Wilkinson, J. H. (1965). *The algebraic eigenvalue problem*. en. Numerical Mathematics and Scientific Computation. Oxford, England: Clarendon Press.