From blind source separation to tensor decomposition: an algebraic algorithm

Martina Iannacito

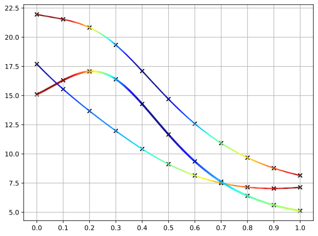
joint work with Ignat Domanov and Lieven De Lathauwer

Numa seminar
Leuven, Belgium, October 26, 2023

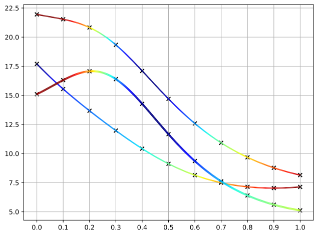
KU LEUVEN kulak
STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics
erc European Research Council Established by the European Commission

Overview

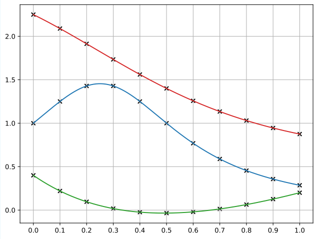
Blind Source Separation problem



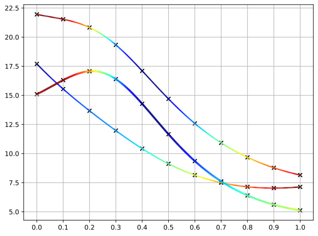
Blind Source Separation problem



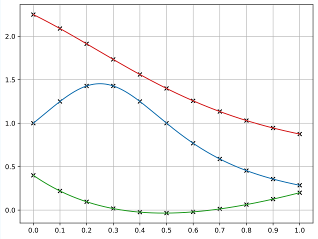
$$= \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix} \times$$

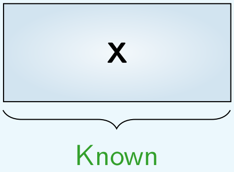


Blind Source Separation problem

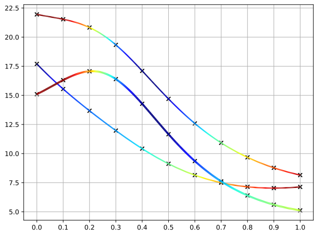


$$= \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix} \times$$

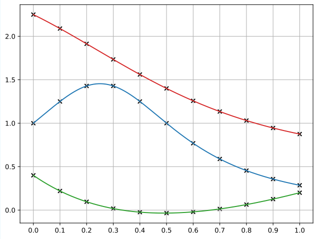




Blind Source Separation problem



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



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

Overview

Constraints for uniqueness

Let $\mathbf{X}$ be a matrix

Definition: a *deterministic condition of* $\mathbf{X}$ is a particular matrix property which is always true.

Definition: a *generic condition of* $\mathbf{X}$ depends on a parameter $\mathbf{z} \in \Omega$ and holds almost everywhere, i.e., if Σ is the set of $\mathbf{z}$ values for which the condition doesn't hold, then $\mu(\Sigma) = 0$ where μ is a measure absolute continuous w.r.t. the Lebesgue one.

Deterministic conditions

- Statistical independence → Independent Component Analysis;

$$\mathbf{X} = \mathbf{M} \underset{\text{Ind}}{\mathbf{S}^T}$$

[Comon and Jutten 2009]

Deterministic conditions

- Statistical independence → Independent Component Analysis;

$$\mathbf{X} = \mathbf{M}\,\underset{\text{Ind}}{\mathbf{S}^T}$$

- Nonnegativity → Nonnegative Matrix Factorization;

$$\mathbf{X} = \underset{+}{\mathbf{M}}\,\underset{+}{\mathbf{S}^T}$$

[Comon and Jutten 2009]

Deterministic conditions

- Statistical independence → Independent Component Analysis;

$$\mathbf{X} = \mathbf{M} \, \underset{\text{Ind}}{\mathbf{S}^T}$$

- Nonnegativity → Nonnegative Matrix Factorization;

$$\mathbf{X} = \underset{+}{\mathbf{M}} \, \underset{+}{\mathbf{S}^T}$$

- Sparsity → Sparse Component Analysis;

$$\mathbf{X} = \mathbf{M} \, \underset{\text{Max0}}{\mathbf{S}^T}$$

[Comon and Jutten 2009]

Deterministic conditions

- Statistical independence → Independent Component Analysis;

$$\mathbf{X} = \mathbf{M} \underset{\text{Ind}}{\mathbf{S}^T}$$

- Nonnegativity → Nonnegative Matrix Factorization;

$$\mathbf{X} = \underset{+}{\mathbf{M}} \underset{+}{\mathbf{S}^T}$$

- Sparsity → Sparse Component Analysis;

$$\mathbf{X} = \mathbf{M} \underset{\text{Max0}}{\mathbf{S}^T}$$

- ...

[Comon and Jutten 2009]

General case

$$\mathbf{X} = \mathbf{M} \quad \mathbf{S}^T$$

General case

$$\mathbf{X} = \mathbf{M} \; \mathbf{S}^T$$

$$\underbrace{\mathbf{X}}_{\mathbf{X}} = \underbrace{\mathbf{M} \; \mathbf{G}}_{\mathbf{M}_1} \; \underbrace{\mathbf{G}^{-1} \; \mathbf{S}^T}_{\mathbf{S}_1^T}$$

Uniqueness isn't guaranteed!

Overview

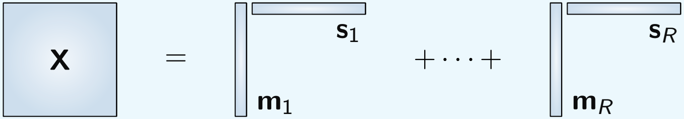
Constraints for uniqueness

Let $\mathbf{X}$ be a matrix

Definition: a *deterministic condition of* $\mathbf{X}$ is a particular matrix property which is always true.

Definition: a *generic condition of* $\mathbf{X}$ depends on a parameter $\mathbf{z} \in \Omega$ and holds almost everywhere, i.e., if Σ is the set of $\mathbf{z}$ values for which the condition doesn't hold, then $\mu(\Sigma) = 0$ where μ is a measure absolute continuous w.r.t. the Lebesgue one.

Problem statement

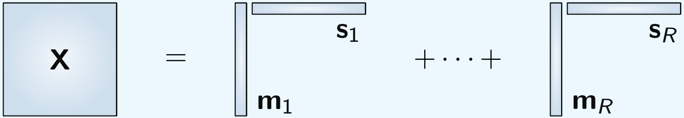


$$\mathbf{X} = \sum_{r=1}^{R} \mathbf{m}_r(\mathbf{z}) \otimes \mathbf{s}_r(\mathbf{z})$$

where

- $\mathbf{z} \in \Omega$ a subset of $\mathbb{R}^n$;

Problem statement

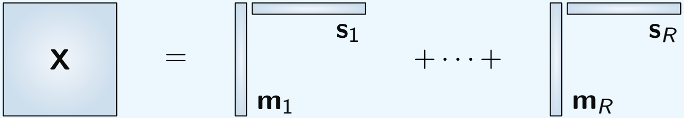


$$\mathbf{X} = \sum_{r=1}^{R} \mathbf{m}_r(\mathbf{z}) \otimes \mathbf{s}_r(\mathbf{z})$$

where

- $\mathbf{z} \in \Omega$ a subset of $\mathbb{R}^n$;
- $\mathbf{m}_r(\mathbf{z})$ are linearly independent;

Problem statement

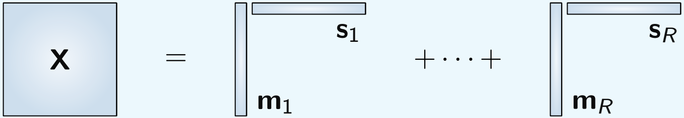


$$\mathbf{X} = \sum_{r=1}^{R} \mathbf{m}_r(\mathbf{z}) \otimes \mathbf{s}_r(\mathbf{z}) = \sum_{r=1}^{R} \mathbf{m}_r(\mathbf{z}) \otimes \mathbf{s}_r(\boldsymbol{\xi}_r)$$

where

- $\mathbf{z} \in \Omega$ a subset of $\mathbb{R}^n$;
- $\mathbf{m}_r(\mathbf{z})$ are linearly independent;
- each $\mathbf{s}_r(\mathbf{z})$ depends on ℓ independent parameters, entries of $\boldsymbol{\xi}_r \in \mathbb{R}^\ell$;

Problem statement



$$\mathbf{X} = \sum_{r=1}^{R} \mathbf{m}_r(\mathbf{z}) \otimes \mathbf{s}_r(\boldsymbol{\xi}_r) = \sum_{r=1}^{R} \mathbf{m}_r(\mathbf{z}) \otimes \mathbf{s}(\boldsymbol{\xi}_r)$$

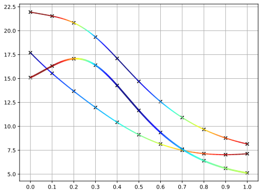
where

- $\mathbf{z} \in \Omega$ a subset of $\mathbb{R}^n$;
- $\mathbf{m}_r(\mathbf{z})$ are linearly independent;
- each $\mathbf{s}_r(\mathbf{z})$ depends on ℓ independent parameters, entries of $\boldsymbol{\xi}_r \in \mathbb{R}^\ell$;
- each $\mathbf{s}_r(\boldsymbol{\xi}_r)$ has the structure

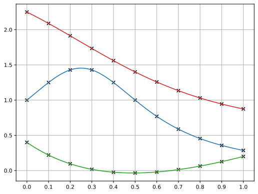
$$\mathbf{s}(\boldsymbol{\xi}_r) = \left[\frac{p_1}{q_1} \circ \mathbf{f}(\boldsymbol{\xi}_r) \quad \ldots \quad \frac{p_N}{q_N} \circ \mathbf{f}(\boldsymbol{\xi}_r) \right]$$

with p_h, q_h polynomials and $\mathbf{f} = [f_1, \ldots, f_\ell]$ a vectorial function.

Example



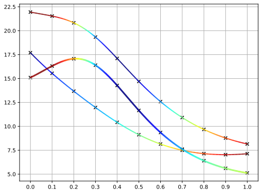
$$= \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix} \times$$



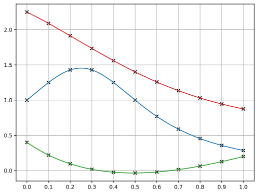
Given the observed mixtures, we assume that

- the mixture matrix $\mathbf{M}$ is constant and full rank;

Example



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

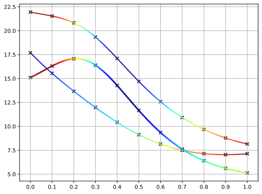


Given the observed mixtures, we assume that

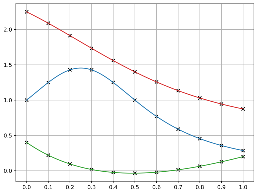
- the mixture matrix **M** is constant and full rank;
- the source signals can be modeled by rational functions, i.e., the columns of **S** are sampled of

$$s(t) = \frac{a_0 + a_1 t + \cdots + a_p t^p}{b_0 + b_1 t + \cdots + b_q t^q} \qquad \text{with} \qquad a_i, b_i \in \mathbb{R}, \quad t \in [t_b, t_e];$$

Example



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



Given the observed mixtures, we assume that

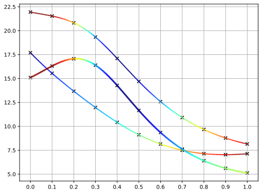
- the mixture matrix $\mathbf{M}$ is constant and full rank;
- the source signals can be modeled by rational functions, i.e., the columns of $\mathbf{S}$ are sampled of

$$s(t) = \frac{a_0 + a_1 t + \cdots + a_p t^p}{b_0 + b_1 t + \cdots + b_q t^q} \qquad \text{with} \qquad a_i, b_i \in \mathbb{R}, \quad t \in [t_b, t_e];$$

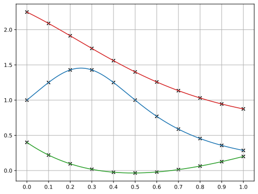
⇓

- $\xi = [a_0, \ldots, a_p, b_0, \ldots, b_q]$

Example



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



Given the observed mixtures, we assume that

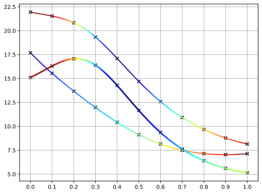
- the mixture matrix $\mathbf{M}$ is constant and full rank;
- the source signals can be modeled by rational functions, i.e., the columns of $\mathbf{S}$ are sampled of

$$s(t) = \frac{a_0 + a_1 t + \cdots + a_p t^p}{b_0 + b_1 t + \cdots + b_q t^q} \qquad \text{with} \qquad a_i, b_i \in \mathbb{R}, \quad t \in [t_b, t_e];$$

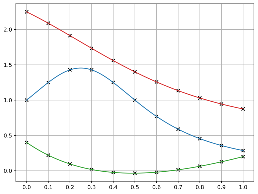
⇩

- $\xi = [a_0, \ldots, a_p, b_0, \ldots, b_q]$
- $\ell = p + q + 2$

Example



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



Given the observed mixtures, we assume that

- the mixture matrix $\mathbf{M}$ is constant and full rank;
- the source signals can be modeled by rational functions, i.e., the columns of $\mathbf{S}$ are sampled of

$$s(t) = \frac{a_0 + a_1 t + \cdots + a_p t^p}{b_0 + b_1 t + \cdots + b_q t^q} \qquad \text{with} \qquad a_i, b_i \in \mathbb{R}, \quad t \in [t_b, t_e];$$

⇓

- $\xi = [a_0, \ldots, a_p, b_0, \ldots, b_q]$
- $\ell = p + q + 2$
- $\mathbf{f}$ is the identity

A cookbook recipe for generic uniqueness [Domanov and De Lathauwer 2016]

Let $\mathbf{t}(\mathbf{x}) = \left[\frac{p_1}{q_1}(\mathbf{x}) \quad \dots \quad \frac{p_N}{q_N}(\mathbf{x})\right]^T$ for $\mathbf{x} \in \Theta = \{\mathbf{x} \in \mathbb{C}^\ell : q_1(\mathbf{x}) \cdots q_N(\mathbf{x}) \neq 0\}$, if

1. $\text{rank}\mathbf{M}(z) = R$ for a generic choice of $\mathbf{z}$;
2. each f_h is the ratio of two analytical functions on $\mathbb{C}^\ell$;
3. there exists $\boldsymbol{\xi}_0 \in \mathbb{C}^\ell$ s.t. $\det\mathbf{J}(\mathbf{f}, \boldsymbol{\xi}_0) \neq 0$;
4. the dimension of the span of $\mathbf{t}(\mathbf{x})$ for $x \in \Theta$ is at least $\hat{N}$;
5. $\text{rank}\mathbf{J}(\mathbf{t}, \mathbf{x}) > \hat{\ell}$ for a generic choice of $\mathbf{z}$;
6. $R \leq \hat{N} - \hat{\ell}$;

then

$$\mathbf{X} = \sum_{r=1}^{R} \mathbf{m}_r(\mathbf{z}) \otimes \mathbf{s}(\boldsymbol{\xi}_r)$$

is generically unique.

Remarks for BSS

It is assumed that the columns of $\mathbf{S}$ are values of the rational function

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

Composed with the function $\mathbf{f}$

Remarks for BSS

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

Composed with the function $\mathbf{f}$

Remarks for BSS

It is assumed that the columns of $\mathbf{S}$ are values of the rational function

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

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

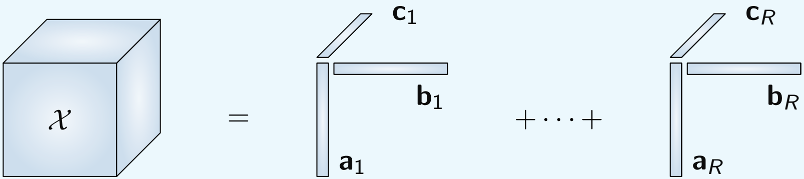
$$\mathcal{V} = \Big\{ (z_1, \ldots, z_N) \in \mathbb{C}^N : P_k(z_1, \ldots, z_N) = 0 \Big\}.$$

Composed with the function $\mathbf{f}$

Overview

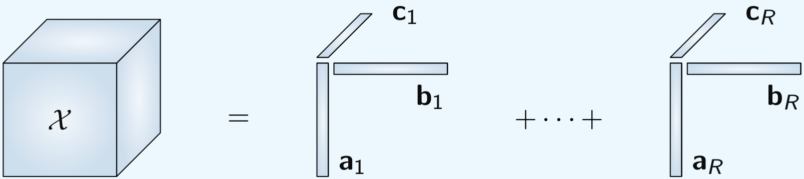
Link with the CPD

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

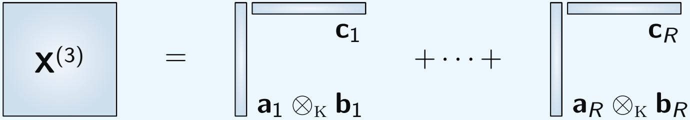


Link with the CPD

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

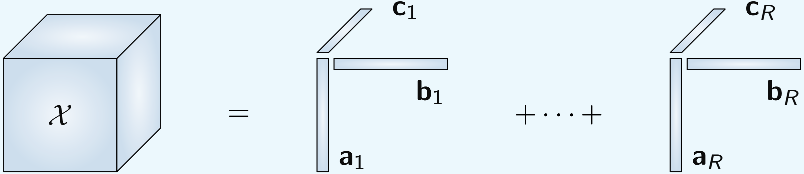


$$\text{then } \mathbf{X}^{(3)} = (\mathbf{a}_1 \otimes_{\mathrm{K}} \mathbf{b}_1) \otimes \mathbf{c}_1^T + \ldots + (\mathbf{a}_R \otimes_{\mathrm{K}} \mathbf{b}_R) \otimes \mathbf{c}_R^T$$



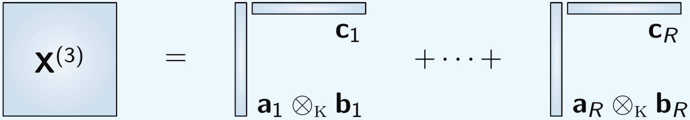
Link with the CPD

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



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

Overview

Algebraic algorithm: high view

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

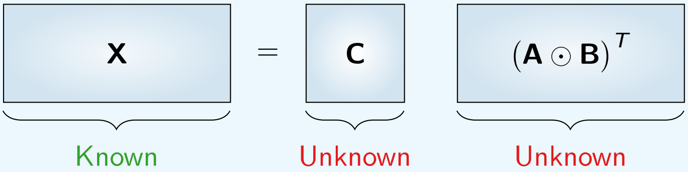
$$\mathbf{X}^{(3)} = \sum_{r=1}^{R} (\mathbf{a}_r \otimes_{\mathrm{K}} \mathbf{b}_r) \otimes \mathbf{c}_r^T = (\mathbf{A} \odot \mathbf{B})\mathbf{C}^T.$$

Algebraic algorithm: high view

Let $\mathcal{X}$ be a $(I \times J \times R)$ 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

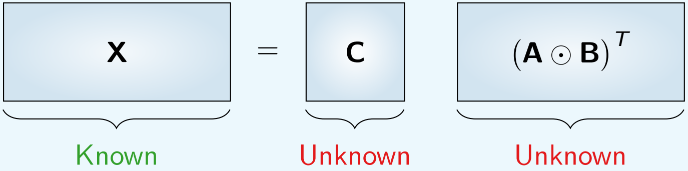


Algebraic algorithm: high view

Let $\mathcal{X}$ be a $(I \times J \times R)$ 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



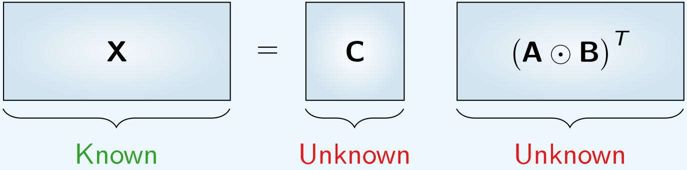
1. compute $\mathbf{C}^{-1}$ from $\mathbf{X}$;

Algebraic algorithm: high view

Let $\mathcal{X}$ be a $(I \times J \times R)$ 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



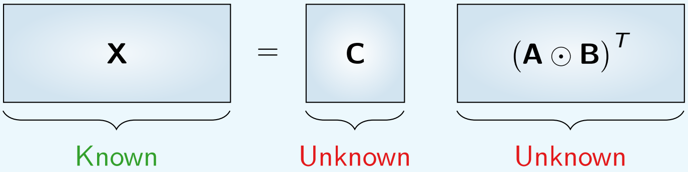
1. compute $\mathbf{C}^{-1}$ from $\mathbf{X}$;
2. compute $(\mathbf{A} \odot \mathbf{B})$ as the transposed product of $\mathbf{C}^{-1}\mathbf{X}$;

Algebraic algorithm: high view

Let $\mathcal{X}$ be a $(I \times J \times R)$ 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



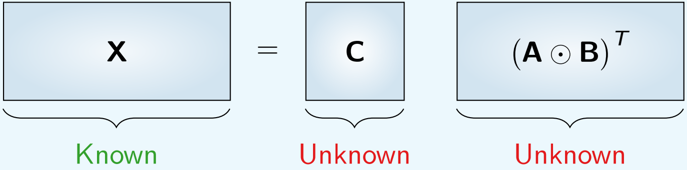
1. compute $\mathbf{C}^{-1}$ from $\mathbf{X}$;
2. compute $(\mathbf{A} \odot \mathbf{B})$ as the transposed product of $\mathbf{C}^{-1}\mathbf{X}$;
3. factorize $(\mathbf{A} \odot \mathbf{B}) = [\mathbf{a}_1 \otimes_{\mathrm{K}} \mathbf{b}_1, \ldots, \mathbf{a}_R \otimes_{\mathrm{K}} \mathbf{b}_R]$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

Algebraic algorithm: high view

Let $\mathcal{X}$ be a $(I \times J \times R)$ 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



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

Equivalent condition I

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

Equivalent condition I

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

⇩

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

Equivalent condition I

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

⇓

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

⇕

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

Equivalent condition I

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

⇓

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

⇕

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

⇕

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

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.

Equivalent condition II

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

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

Equivalent condition II

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

⇓

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

⇕

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

Equivalent condition II

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

$\Downarrow$

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

$\Updownarrow$

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

$\Updownarrow$

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

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

Overview

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 $\mathbf{X}$;

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 $\mathbf{Q}$;

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 $\mathcal{E}_0 = \text{null}(\mathbf{Q}) \cap \text{vec}(\text{Sym}_R^d)$

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 $\mathcal{E}_0 = \mathrm{null}(\mathbf{Q}) \cap \mathrm{vec}(\mathrm{Sym}_R^d)$
        - 1.2.1 if $\dim \mathcal{E}_0 = R$, then compute $\mathbf{C}$ by a CPD of $\{\mathbf{c}_1^{\otimes d}, \ldots, \mathbf{c}_R^{\otimes d}\}$ basis of $\mathcal{E}_0$;

Algebraic algorithm outline

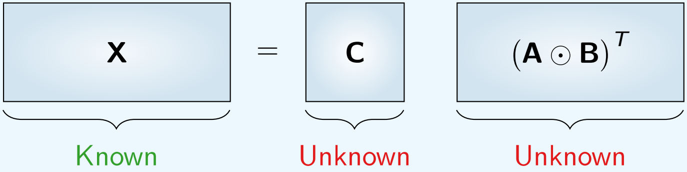
$$\underbrace{\mathbf{X}}_{\text{Known}} = \underbrace{\mathbf{C}}_{\text{Unknown}} \underbrace{(\mathbf{A} \odot \mathbf{B})^T}_{\text{Unknown}}$$

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 $\mathbf{C}$ by a CPD of $\{\mathbf{c}_1^{\otimes d}, \ldots, \mathbf{c}_R^{\otimes d}\}$ 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;

Algebraic algorithm outline

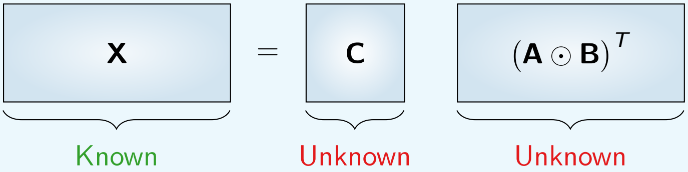


1. compute the factor matrix $\mathbf{C}^{-1}$ from $\mathbf{X}$;
   1.1 compute $\mathbf{Q}$;
   1.2 compute the space $\mathcal{E}_0 = \text{null}(\mathbf{Q}) \cap \text{vec}(\text{Sym}_R^d)$
      1.2.1 if dim $\mathcal{E}_0 = R$, then compute $\mathbf{C}$ by a CPD of $\{\mathbf{c}_1^{\otimes d},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

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

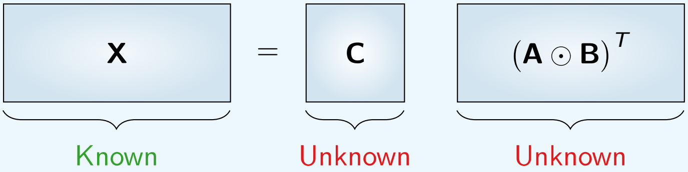
      until dim $\mathcal{E}_{h+1} = R^{h+1}$ and go to step 1.2.1;
2. compute $(\mathbf{A} \odot \mathbf{B})$ as $\mathbf{C}^{-1}\mathbf{X}$ transposed;

Algebraic algorithm outline



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 $\mathbf{C}$ by a CPD of $\{\mathbf{c}_1^{\otimes d}, \ldots, \mathbf{c}_R^{\otimes d}\}$ 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;

Algebraic algorithm outline



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

Overview

Numerical and computational obstacles

- Computational complexity: large and structured objects;

Numerical and computational obstacles

- Computational complexity: large and structured objects;
- NP-hardness;

Numerical and computational obstacles

- Computational complexity: large and structured objects;
- NP-hardness;
- Numerical conditioning and stability;

Numerical and computational obstacles

- Computational complexity: large and structured objects;
- NP-hardness;
- Numerical conditioning and stability;

⇓

Work in progress

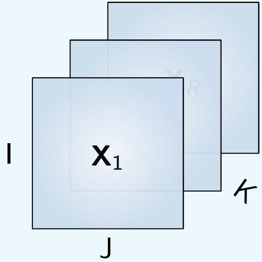
Advice is welcome!

Overview

Q for CPD

[Domanov and De Lathauwer 2013]

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times J \times R)$, then



Definition: **Q** is a $C_I^2 C_J^2 \times C_{R+1}^2$ matrix whose k-th column can be written as

$$\mathbf{Q}(\cdot, k) = \text{vec}\left(\mathcal{C}_2(\mathbf{X}_{k_1} + \mathbf{X}_{k_2}) - \mathcal{C}_2(\mathbf{X}_{k_1}) - \mathcal{C}_2(\mathbf{X}_{k_2})\right)$$

where

- (k_1, k_2) is the k-th element of $\mathcal{Q}_R^2 = \{(k_1, k_2) : 1 \leq k_1 \leq k_2 \leq R\}$;
- C_N^2 is the binomial of N over 2;
- $\mathcal{C}_2(\mathbf{X}_h)$ is the matrix with the determinants of every 2×2 minors of $\mathbf{X}_h$.

Example

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times 3 \times 2)$ such that

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \qquad \text{and} \qquad \mathbf{X}_2 = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}.$$

Example

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times 3 \times 2)$ such that

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \qquad \text{and} \qquad \mathbf{X}_2 = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}.$$

The matrix $\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \end{bmatrix}$ associated with $\mathcal{X}$ slices is such that

Example

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times 3 \times 2)$ such that

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \qquad \text{and} \qquad \mathbf{X}_2 = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}.$$

The matrix $\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \end{bmatrix}$ associated with $\mathcal{X}$ slices is such that

- $\mathbf{q}_1 = \text{vec}(\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1)) = 2\text{vec}(\mathcal{C}_2(\mathbf{X}_1))$;

Example

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times 3 \times 2)$ such that

$$\mathbf{X}_1 = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3\end{bmatrix} \qquad \text{and} \qquad \mathbf{X}_2 = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3\end{bmatrix}.$$

The matrix $\mathbf{Q} = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3\end{bmatrix}$ associated with $\mathcal{X}$ slices is such that

- $\mathbf{q}_1 = \text{vec}(\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1)) = 2\text{vec}(\mathcal{C}_2(\mathbf{X}_1))$;
- $\mathbf{q}_2 = \text{vec}(\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_2))$;

Example

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times 3 \times 2)$ such that

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \qquad \text{and} \qquad \mathbf{X}_2 = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}.$$

The matrix $\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \end{bmatrix}$ associated with $\mathcal{X}$ slices is such that

- $\mathbf{q}_1 = \text{vec}(\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1)) = 2\text{vec}(\mathcal{C}_2(\mathbf{X}_1))$;
- $\mathbf{q}_2 = \text{vec}(\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_2))$;
- $\mathbf{q}_3 = \text{vec}(\mathcal{C}_2(\mathbf{X}_2 + \mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_2)) = 2\text{vec}(\mathcal{C}_2(\mathbf{X}_2))$.

Example

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times 3 \times 2)$ such that

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \quad \text{and} \quad \mathbf{X}_2 = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}.$$

The matrix $\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 \end{bmatrix}$ associated with $\mathcal{X}$ slices is such that

- $\mathbf{q}_1 = \text{vec}(\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1)) = 2\text{vec}(\mathcal{C}_2(\mathbf{X}_1))$;
- $\mathbf{q}_2 = \text{vec}(\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_2))$;
- $\mathbf{q}_3 = \text{vec}(\mathcal{C}_2(\mathbf{X}_2 + \mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_2)) = 2\text{vec}(\mathcal{C}_2(\mathbf{X}_2))$.

The compound matrices are

- $\mathcal{C}_2(\mathbf{X}_1) = \begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_3 \end{bmatrix}$
- $\mathcal{C}_2(\mathbf{X}_2) = \begin{bmatrix} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 & \boldsymbol{\beta}_3 \end{bmatrix}$
- $\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_2) = \begin{bmatrix} \boldsymbol{\gamma}_1 & \boldsymbol{\gamma}_2 & \boldsymbol{\gamma}_3 \end{bmatrix}$

Example

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times 3 \times 2)$ such that

$$\mathbf{X}_1 = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3\end{bmatrix} \qquad \text{and} \qquad \mathbf{X}_2 = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3\end{bmatrix}.$$

The matrix $\mathbf{Q} = \begin{bmatrix}\mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3\end{bmatrix}$ associated with $\mathcal{X}$ slices is such that

- $\mathbf{q}_1 = \text{vec}(\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_1)) = 2\text{vec}(\mathcal{C}_2(\mathbf{X}_1));$
- $\mathbf{q}_2 = \text{vec}(\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_1) - \mathcal{C}_2(\mathbf{X}_2));$
- $\mathbf{q}_3 = \text{vec}(\mathcal{C}_2(\mathbf{X}_2 + \mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_2) - \mathcal{C}_2(\mathbf{X}_2)) = 2\text{vec}(\mathcal{C}_2(\mathbf{X}_2)).$

The compound matrices are

- $\mathcal{C}_2(\mathbf{X}_1) = \begin{bmatrix}\boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_3\end{bmatrix}$
- $\mathcal{C}_2(\mathbf{X}_2) = \begin{bmatrix}\boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 & \boldsymbol{\beta}_3\end{bmatrix}$
- $\mathcal{C}_2(\mathbf{X}_1 + \mathbf{X}_2) = \begin{bmatrix}\boldsymbol{\gamma}_1 & \boldsymbol{\gamma}_2 & \boldsymbol{\gamma}_3\end{bmatrix}$

$$\Rightarrow \quad \mathbf{Q} = \begin{bmatrix} 2\boldsymbol{\alpha}_1 & \boldsymbol{\gamma}_1 - (\boldsymbol{\alpha}_1 + \boldsymbol{\beta}_1) & 2\boldsymbol{\beta}_1 \\ 2\boldsymbol{\alpha}_2 & \boldsymbol{\gamma}_2 - (\boldsymbol{\alpha}_2 + \boldsymbol{\beta}_2) & 2\boldsymbol{\beta}_2 \\ 2\boldsymbol{\alpha}_3 & \boldsymbol{\gamma}_3 - (\boldsymbol{\alpha}_3 + \boldsymbol{\beta}_3) & 2\boldsymbol{\beta}_3 \end{bmatrix}$$

Possible choices

- *Ignore structure* and compute $\mathbf{Q}$ by its definition;

Possible choices

- *Ignore structure* and compute $\mathbf{Q}$ by its definition;
- *Use structure* and compute $\mathbf{Q}^T\mathbf{Q}$;

Possible choices

- *Ignore structure* and compute $\mathbf{Q}$ by its definition;
- *Use structure* and compute $\mathbf{Q}^T\mathbf{Q}$;
- *Use structure* and compute $\mathbf{Q}$;

Possible choices

- *Ignore structure* and compute $\mathbf{Q}$ by its definition;
- *Use structure* and compute $\mathbf{Q}^T\mathbf{Q}$;
- *Use structure* and compute $\mathbf{Q}$;

All options present limitations!

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $(C_I^2 C_J^2 \times C_{R+1}^2)$

- each entry requires two multiplications;
- the total number of entries is $(IJR(I-1)(J-1)(R+1))/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $(C_{R+1}^2 \times C_{R+1}^2)$

Let $\mathbf{a}_{h_i,k_j} = \text{vec}(\mathbf{A}_{h_i,k_j}) = \text{vec}(\mathbf{X}_{h_i}^T \mathbf{X}_{k_j})$ be a vector of length J^2, then

- $\text{tr}(\mathbf{A}_{h_i,k_j}^T \mathbf{A}_{k_\ell,h_m}) = \mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $(C_I^2 C_J^2 \times C_{R+1}^2)$

- each entry requires two multiplications;
- the total number of entries is $(IJR(I-1)(J-1)(R+1))/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $(C_{R+1}^2 \times C_{R+1}^2)$

Let $\mathbf{a}_{h_i,k_j} = \text{vec}(\mathbf{A}_{h_i,k_j}) = \text{vec}(\mathbf{X}_{h_i}^T \mathbf{X}_{k_j})$ be a vector of length J^2, then

- $\text{tr}(\mathbf{A}_{h_i,k_j}^T \mathbf{A}_{k_\ell,h_m}) = \mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $(C_I^2 C_J^2 \times C_{R+1}^2)$

- each entry requires two multiplications;
- the total number of entries is $\big(IJR(I-1)(J-1)(R+1)\big)/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $(C_{R+1}^2 \times C_{R+1}^2)$

Let $\mathbf{a}_{h_i,k_j} = \text{vec}(\mathbf{A}_{h_i,k_j}) = \text{vec}(\mathbf{X}_{h_i}^T \mathbf{X}_{k_j})$ be a vector of length J^2, then

- $\text{tr}(\mathbf{A}_{h_i,k_j}^T \mathbf{A}_{k_\ell,h_m}) = \mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $(C_I^2 C_J^2 \times C_{R+1}^2)$

- each entry requires two multiplications;
- the total number of entries is $\big(IJR(I-1)(J-1)(R+1)\big)/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $(C_{R+1}^2 \times C_{R+1}^2)$

Let $\mathbf{a}_{h_i,k_j} = \text{vec}(\mathbf{A}_{h_i,k_j}) = \text{vec}(\mathbf{X}_{h_i}^T \mathbf{X}_{k_j})$ be a vector of length J^2, then

- $\text{tr}(\mathbf{A}_{h_i,k_j}^T \mathbf{A}_{k_\ell,h_m}) = \mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Structure highlight

Example

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times 3 \times 2)$ such that

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \end{bmatrix} \qquad \text{and} \qquad \mathbf{X}_2 = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3 \end{bmatrix}.$$

The matrix $\mathbf{Q}$ is

$$\mathbf{Q} = \begin{bmatrix} 2\alpha_1 & \gamma_1 - (\alpha_1 + \beta_1) & 2\beta_1 \\ 2\alpha_2 & \gamma_2 - (\alpha_2 + \beta_2) & 2\beta_2 \\ 2\alpha_3 & \gamma_3 - (\alpha_3 + \beta_3) & 2\beta_3 \end{bmatrix}.$$

† twice with opposite sign

Structure highlight

Example

Let $\mathcal{X}$ be an order-3 tensor of dimension $(I \times 3 \times 2)$ such that

$$\mathbf{X}_1 = \begin{bmatrix}\mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3\end{bmatrix} \quad \text{and} \quad \mathbf{X}_2 = \begin{bmatrix}\mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3\end{bmatrix}.$$

The matrix $\mathbf{Q}$ is

$$\mathbf{Q} = \begin{bmatrix} 2\alpha_1 & \gamma_1 - (\alpha_1 + \beta_1) & 2\beta_1 \\ 2\alpha_2 & \gamma_2 - (\alpha_2 + \beta_2) & 2\beta_2 \\ 2\alpha_3 & \gamma_3 - (\alpha_3 + \beta_3) & 2\beta_3 \end{bmatrix}.$$

We can remark that

- the entries of α_1 appears† in $\mathbf{a}_1\mathbf{a}_2^T - \mathbf{a}_2\mathbf{a}_1^T$
- the entries of β_1 appears† in $\mathbf{b}_1\mathbf{b}_2^T - \mathbf{b}_2\mathbf{b}_1^T$
- the entries of $\gamma_1 - (\alpha_1 + \beta_1)$ appears† in $\mathbf{a}_1\mathbf{b}_2^T + \mathbf{b}_1\mathbf{a}_2^T - \mathbf{b}_2^T\mathbf{a}_1 - \mathbf{a}_2\mathbf{b}_1^T$

† twice with opposite sign

Exterior algebra and $\mathcal{C}_2$

Definition: The exterior product $\wedge : \mathbb{R}^I \times \mathbb{R}^J \to \mathbb{R}^{I \times J}$ is defined as

$$\mathbf{a} \wedge \mathbf{b} = \mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T.$$

Exterior algebra and $\mathcal{C}_2$

Definition: The exterior product $\wedge : \mathbb{R}^I \times \mathbb{R}^J \to \mathbb{R}^{I \times J}$ is defined as

$$\mathbf{a} \wedge \mathbf{b} = \mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T.$$

Example

Let $\mathbf{a}^T = \begin{bmatrix} a & b & c \end{bmatrix}$ and $\mathbf{b}^T = \begin{bmatrix} d & e & f \end{bmatrix}$ be two vectors, then

$$\mathbf{a} \wedge \mathbf{b} = \begin{bmatrix} 0 & ae - bd & af - cd \\ -ae + bd & 0 & bf - ce \\ -af + cd & -bf + ce & 0 \end{bmatrix}.$$

The matrix $\mathbf{a} \wedge \mathbf{b}$ has the properties:

Exterior algebra and $\mathcal{C}_2$

Definition: The exterior product $\wedge : \mathbb{R}^I \times \mathbb{R}^J \to \mathbb{R}^{I \times J}$ is defined as

$$\mathbf{a} \wedge \mathbf{b} = \mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T.$$

Example

Let $\mathbf{a}^T = \begin{bmatrix} a & b & c \end{bmatrix}$ and $\mathbf{b}^T = \begin{bmatrix} d & e & f \end{bmatrix}$ be two vectors, then

$$\mathbf{a} \wedge \mathbf{b} = \begin{bmatrix} 0 & ae - bd & af - cd \\ -ae + bd & 0 & bf - ce \\ -af + cd & -bf + ce & 0 \end{bmatrix}.$$

The matrix $\mathbf{a} \wedge \mathbf{b}$ has the properties:

- the diagonal entries are zeros;

Exterior algebra and $\mathcal{C}_2$

Definition: The exterior product $\wedge : \mathbb{R}^I \times \mathbb{R}^J \to \mathbb{R}^{I \times J}$ is defined as

$$\mathbf{a} \wedge \mathbf{b} = \mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T.$$

Example

Let $\mathbf{a}^T = \begin{bmatrix} a & b & c \end{bmatrix}$ and $\mathbf{b}^T = \begin{bmatrix} d & e & f \end{bmatrix}$ be two vectors, then

$$\mathbf{a} \wedge \mathbf{b} = \begin{bmatrix} 0 & ae - bd & af - cd \\ -ae + bd & 0 & bf - ce \\ -af + cd & -bf + ce & 0 \end{bmatrix}.$$

The matrix $\mathbf{a} \wedge \mathbf{b}$ has the properties:

- the diagonal entries are zeros;
- it is skew-symmetric;

Exterior algebra and $\mathcal{C}_2$

Definition: The exterior product $\wedge : \mathbb{R}^I \times \mathbb{R}^J \to \mathbb{R}^{I \times J}$ is defined as

$$\mathbf{a} \wedge \mathbf{b} = \mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T.$$

Example

Let $\mathbf{a}^T = \begin{bmatrix} a & b & c \end{bmatrix}$ and $\mathbf{b}^T = \begin{bmatrix} d & e & f \end{bmatrix}$ be two vectors, then

$$\mathbf{a} \wedge \mathbf{b} = \begin{bmatrix} 0 & ae - bd & af - cd \\ -ae + bd & 0 & bf - ce \\ -af + cd & -bf + ce & 0 \end{bmatrix}.$$

The matrix $\mathbf{a} \wedge \mathbf{b}$ has the properties:

- the diagonal entries are zeros;
- it is skew-symmetric;
- the elements highlighted are the entries of $\mathcal{C}_2(\begin{bmatrix} \mathbf{a} & \mathbf{b} \end{bmatrix})$.

Gram matrix formulation

Property: $$\left\langle \mathrm{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])), \mathrm{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])) \right\rangle = 2\left\langle \mathrm{vec}(\mathbf{a} \wedge \mathbf{b}), \mathrm{vec}(\mathbf{a} \wedge \mathbf{b}) \right\rangle$$
$$= 4||\mathbf{a}||^2||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b} \rangle^2$$

Gram matrix formulation

Property: $$\Big\langle \mathrm{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])), \mathrm{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])) \Big\rangle = 2\Big\langle \mathrm{vec}(\mathbf{a} \wedge \mathbf{b}), \mathrm{vec}(\mathbf{a} \wedge \mathbf{b}) \Big\rangle = 4||\mathbf{a}||^2||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b} \rangle^2$$

Idea: Using the exterior and exploiting the structure.

Gram matrix formulation

$$\text{Property: } \left\langle \text{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])), \text{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])) \right\rangle = 2\left\langle \text{vec}(\mathbf{a} \wedge \mathbf{b}), \text{vec}(\mathbf{a} \wedge \mathbf{b}) \right\rangle$$
$$= 4||\mathbf{a}||^2||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b} \rangle^2$$

Idea: Using the exterior and exploiting the structure.

- $\mathbf{Q}$ is not explicitly constructed

Gram matrix formulation

Property: $$\left\langle \mathrm{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])), \mathrm{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])) \right\rangle = 2\left\langle \mathrm{vec}(\mathbf{a} \wedge \mathbf{b}), \mathrm{vec}(\mathbf{a} \wedge \mathbf{b}) \right\rangle$$
$$= 4||\mathbf{a}||^2||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b} \rangle^2$$

Idea: Using the exterior and exploiting the structure.

- $\mathbf{Q}$ is not explicitly constructed
- $\langle \mathbf{q}_k, \mathbf{q}_h \rangle = \langle \mathrm{vec}(\mathcal{C}_2(\mathbf{Y})), \mathrm{vec}(\mathcal{C}_2(\mathbf{Z})) \rangle$ with $\mathbf{Y}, \mathbf{Z}$ slices or a linear combination of slices of the input tensor $\mathcal{X}$.

Gram matrix formulation

Property: $$\left\langle \text{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])), \text{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])) \right\rangle = 2\left\langle \text{vec}(\mathbf{a} \wedge \mathbf{b}), \text{vec}(\mathbf{a} \wedge \mathbf{b}) \right\rangle$$
$$= 4||\mathbf{a}||^2||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b} \rangle^2$$

Idea: Using the exterior and exploiting the structure.

- $\mathbf{Q}$ is not explicitly constructed
- $\langle \mathbf{q}_k, \mathbf{q}_h \rangle = \langle \text{vec}(\mathcal{C}_2(\mathbf{Y})), \text{vec}(\mathcal{C}_2(\mathbf{Z})) \rangle$ with $\mathbf{Y}, \mathbf{Z}$ slices or a linear combination of slices of the input tensor $\mathcal{X}$.

⇓

$$\langle \mathbf{q}_k, \mathbf{q}_h \rangle = \text{tr}(\mathbf{A}_{h_1 k_1})\text{tr}(\mathbf{A}_{h_2 k_2}) + \text{tr}(\mathbf{A}_{h_1 k_2})\text{tr}(\mathbf{A}_{h_2 k_1}) - \text{tr}(\mathbf{A}_{h_1 k_1}^T \mathbf{A}_{h_1 k_2}) - \text{tr}(\mathbf{A}_{h_1 k_2}^T \mathbf{A}_{h_2 k_1})$$

Gram matrix formulation

$$\text{Property: } \left\langle \text{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])), \text{vec}(\mathcal{C}_2([\mathbf{a} \quad \mathbf{b}])) \right\rangle = 2\left\langle \text{vec}(\mathbf{a} \wedge \mathbf{b}), \text{vec}(\mathbf{a} \wedge \mathbf{b}) \right\rangle$$
$$= 4||\mathbf{a}||^2||\mathbf{b}||^2 - \langle \mathbf{a}, \mathbf{b} \rangle^2$$

Idea: Using the exterior and exploiting the structure.

- $\mathbf{Q}$ is not explicitly constructed
- $\langle \mathbf{q}_k, \mathbf{q}_h \rangle = \langle \text{vec}(\mathcal{C}_2(\mathbf{Y})), \text{vec}(\mathcal{C}_2(\mathbf{Z})) \rangle$ with $\mathbf{Y}, \mathbf{Z}$ slices or a linear combination of slices of the input tensor $\mathcal{X}$.

⇩

$$\langle \mathbf{q}_k, \mathbf{q}_h \rangle = \text{tr}(\mathbf{A}_{h_1 k_1})\text{tr}(\mathbf{A}_{h_2 k_2}) + \text{tr}(\mathbf{A}_{h_1 k_2})\text{tr}(\mathbf{A}_{h_2 k_1}) - \text{tr}(\mathbf{A}_{h_1 k_1}^T \mathbf{A}_{h_1 k_2}) - \text{tr}(\mathbf{A}_{h_1 k_2}^T \mathbf{A}_{h_2 k_1})$$

where $\mathbf{A}_{h_i k_j} = \mathbf{X}_{h_i}^T \mathbf{X}_{k_j}$ for $h_i \leq k_j$.

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $(C_I^2 C_J^2 \times C_{R+1}^2)$

- each entry requires two multiplications;
- the total number of entries is $\big(IJR(I-1)(J-1)(R+1)\big)/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $(C_{R+1}^2 \times C_{R+1}^2)$

Let $\mathbf{a}_{h_i,k_j} = \text{vec}(\mathbf{A}_{h_i,k_j}) = \text{vec}(\mathbf{X}_{h_i}^T \mathbf{X}_{k_j})$ be a vector of length J^2, then

- $\text{tr}(\mathbf{A}_{h_i,k_j}^T \mathbf{A}_{k_\ell,h_m}) = \mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $(C_I^2 C_J^2 \times C_{R+1}^2)$

- each entry requires two multiplications;
- the total number of entries is $\big(IJR(I-1)(J-1)(R+1)\big)/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $(C_{R+1}^2 \times C_{R+1}^2)$

Let $\mathbf{a}_{h_i,k_j} = \text{vec}(\mathbf{A}_{h_i,k_j}) = \text{vec}(\mathbf{X}_{h_i}^T\mathbf{X}_{k_j})$ be a vector of length J^2, then

- $\text{tr}(\mathbf{A}_{h_i,k_j}^T \mathbf{A}_{k_\ell,h_m}) = \mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $(C_I^2 C_J^2 \times C_{R+1}^2)$

- each entry requires two multiplications;
- the total number of entries is $\big(IJR(I-1)(J-1)(R+1)\big)/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $(C_{R+1}^2 \times C_{R+1}^2)$

Let $\mathbf{a}_{h_i,k_j} = \text{vec}(\mathbf{A}_{h_i,k_j}) = \text{vec}(\mathbf{X}_{h_i}^T \mathbf{X}_{k_j})$ be a vector of length J^2, then

- $\text{tr}(\mathbf{A}_{h_i,k_j}^T \mathbf{A}_{k_\ell,h_m}) = \mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $(C_I^2 C_J^2 \times C_{R+1}^2)$

- each entry requires two multiplications;
- the total number of entries is $\big(IJR(I-1)(J-1)(R+1)\big)/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $(C_{R+1}^2 \times C_{R+1}^2)$

Let $\mathbf{a}_{h_i,k_j} = \mathrm{vec}(\mathbf{A}_{h_i,k_j}) = \mathrm{vec}(\mathbf{X}_{h_i}^T\mathbf{X}_{k_j})$ be a vector of length J^2, then

- $\mathrm{tr}(\mathbf{A}_{h_i,k_j}^T\mathbf{A}_{k_\ell,h_m}) = \mathbf{a}_{h_i,k_j}^T\mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T\mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $\left(C_I^2 C_J^2 \times C_{R+1}^2\right)$

- each entry requires two multiplications;
- the total number of entries is $\big(IJR(I-1)(J-1)(R+1)\big)/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $\left(C_{R+1}^2 \times C_{R+1}^2\right)$

Let $\mathbf{a}_{h_i,k_j} = \text{vec}\big(\mathbf{A}_{h_i,k_j}\big) = \text{vec}\big(\mathbf{X}_{h_i}^T \mathbf{X}_{k_j}\big)$ be a vector of length J^2, then

- $\text{tr}\big(\mathbf{A}_{h_i,k_j}^T \mathbf{A}_{k_\ell,h_m}\big) = \mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Computational complexity

Let $\mathcal{X}$ be a $(I \times J \times R)$ tensor of rank R

Direct construction: the matrix $\mathbf{Q}$ has size $(C_I^2 C_J^2 \times C_{R+1}^2)$

- each entry requires two multiplications;
- the total number of entries is $\big(IJR(I-1)(J-1)(R+1)\big)/8$;

assuming that $R \sim IJ$, then the computational complexity is $R^4/4$.

Gram construction: the matrix $\mathbf{Q}^T\mathbf{Q}$ has size $(C_{R+1}^2 \times C_{R+1}^2)$

Let $\mathbf{a}_{h_i,k_j} = \text{vec}(\mathbf{A}_{h_i,k_j}) = \text{vec}(\mathbf{X}_{h_i}^T \mathbf{X}_{k_j})$ be a vector of length J^2, then

- $\text{tr}(\mathbf{A}_{h_i,k_j}^T \mathbf{A}_{k_\ell,h_m}) = \mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$;
- compute $\mathbf{a}_{h_i,k_j}^T \mathbf{a}_{k_\ell,h_m}$ require J^2 multiplications;
- the total number of inner product to compute is C_{N+1}^2 with $N = C_{R+1}^2$;

assuming that $J \sim \sqrt{R}$, then the computational complexity is $R^5/8$.

Numerical evidences

Given an input random tensor $\mathcal{X}$ of dimensions $(I \times I \times R)$ such that its rank is

$$R = (I-1)^2$$

Size	$\mathbf{Q}$		$\mathbf{Q}^T\mathbf{Q}$
	`det`	`recursive`	
$(11, 11, 100)$	`1.364e+01`	`1.556e+00`	`7.373e-01`
$(12, 12, 121)$	`2.876e+01`	`3.613e+00`	`1.621e+00`
$(13, 13, 144)$	`5.814e+01`	`6.353e+00`	`3.477e+00`
$(14, 14, 169)$	`1.071e+02`	`1.130e+01`	`6.928e+00`
$(15, 15, 196)$	`1.937e+02`	`1.956e+01`	`8.954e+01`
$(16, 16, 225)$	`3.298e+02`	`3.291e+01`	`9.325e+02`

Table: Time estimates in seconds for computing $\mathbf{Q}$ and its Gram matrix.

The link between rank and dimensions is meant to satisfy [Domanov and De Lathauwer 2016] theorem.

Numerical evidences

Given an input random tensor $\mathcal{X}$ of dimensions $(I \times I \times R)$ such that its rank is

$$R = (I - 1)^2$$

Size	$\mathbf{Q}$		$\mathbf{Q}^T\mathbf{Q}$
	det	recursive	
$(11, 11, 100)$	1.364e+01	1.556e+00	7.373e-01
$(12, 12, 121)$	2.876e+01	3.613e+00	1.621e+00
$(13, 13, 144)$	5.814e+01	6.353e+00	3.477e+00
$(14, 14, 169)$	1.071e+02	1.130e+01	6.928e+00
$(15, 15, 196)$	1.937e+02	1.956e+01	8.954e+01
$(16, 16, 225)$	3.298e+02	3.291e+01	9.325e+02

Table: Time estimates in seconds for computing $\mathbf{Q}$ and its Gram matrix.

The link between rank and dimensions is meant to satisfy [Domanov and De Lathauwer 2016] theorem.

Numerical evidences

Given an input random tensor $\mathcal{X}$ of dimensions $(I \times I \times R)$ such that its rank is

$$R = (I - 1)^2$$

Size	$\mathbf{Q}$		$\mathbf{Q}^T\mathbf{Q}$
	`det`	`recursive`	
$(11, 11, 100)$	1.364e+01	1.556e+00	7.373e-01
$(12, 12, 121)$	2.876e+01	3.613e+00	1.621e+00
$(13, 13, 144)$	5.814e+01	6.353e+00	3.477e+00
$(14, 14, 169)$	1.071e+02	1.130e+01	6.928e+00
$(15, 15, 196)$	1.937e+02	1.956e+01	8.954e+01
$(16, 16, 225)$	3.298e+02	3.291e+01	9.325e+02

Table: Time estimates in seconds for computing $\mathbf{Q}$ and its Gram matrix.

The link between rank and dimensions is meant to satisfy [Domanov and De Lathauwer 2016] theorem.

Numerical evidences

Given an input random tensor $\mathcal{X}$ of dimensions $(I \times I \times R)$ such that its rank is

$$R = (I-1)^2$$

Size	$\mathbf{Q}$		$\mathbf{Q}^T\mathbf{Q}$
	det	recursive	
$(11, 11, 100)$	1.364e+01	1.556e+00	7.373e-01
$(12, 12, 121)$	2.876e+01	3.613e+00	1.621e+00
$(13, 13, 144)$	5.814e+01	6.353e+00	3.477e+00
$(14, 14, 169)$	1.071e+02	1.130e+01	6.928e+00
$(15, 15, 196)$	1.937e+02	1.956e+01	8.954e+01
$(16, 16, 225)$	3.298e+02	3.291e+01	9.325e+02

Table: Time estimates in seconds for computing $\mathbf{Q}$ and its Gram matrix.

The link between rank and dimensions is meant to satisfy [Domanov and De Lathauwer 2016] theorem.

Numerical evidences

Given an input random tensor $\mathcal{X}$ of dimensions $(I \times I \times R)$ such that its rank is

$$R = (I - 1)^2$$

Size	$\mathbf{Q}$		$\mathbf{Q}^T\mathbf{Q}$
	`det`	`recursive`	
$(11, 11, 100)$	`1.364e+01`	`1.556e+00`	`7.373e-01`
$(12, 12, 121)$	`2.876e+01`	`3.613e+00`	`1.621e+00`
$(13, 13, 144)$	`5.814e+01`	`6.353e+00`	`3.477e+00`
$(14, 14, 169)$	`1.071e+02`	`1.130e+01`	`6.928e+00`
$(15, 15, 196)$	`1.937e+02`	`1.956e+01`	`8.954e+01`
$(16, 16, 225)$	`3.298e+02`	`3.291e+01`	`9.325e+02`

Table: Time estimates in seconds for computing $\mathbf{Q}$ and its Gram matrix.

The link between rank and dimensions is meant to satisfy [Domanov and De Lathauwer 2016] theorem.

Numerical evidences

Given an input random tensor $\mathcal{X}$ of dimensions $(I \times I \times R)$ such that its rank is

$$R = (I - 1)^2$$

Size	$\mathbf{Q}$		$\mathbf{Q}^T\mathbf{Q}$
	`det`	`recursive`	
$(11, 11, 100)$	1.364e+01	1.556e+00	7.373e-01
$(12, 12, 121)$	2.876e+01	3.613e+00	1.621e+00
$(13, 13, 144)$	5.814e+01	6.353e+00	3.477e+00
$(14, 14, 169)$	1.071e+02	1.130e+01	6.928e+00
$(15, 15, 196)$	1.937e+02	1.956e+01	8.954e+01
$(16, 16, 225)$	3.298e+02	3.291e+01	9.325e+02

Table: Time estimates in seconds for computing $\mathbf{Q}$ and its Gram matrix.

The link between rank and dimensions is meant to satisfy [Domanov and De Lathauwer 2016] theorem.

Numerical evidences

Given an input random tensor $\mathcal{X}$ of dimensions $(I \times I \times R)$ such that its rank is

$$R = (I - 1)^2$$

Size	$\mathbf{Q}$		$\mathbf{Q}^T\mathbf{Q}$
	det	recursive	
$(11, 11, 100)$	1.364e+01	1.556e+00	7.373e-01
$(12, 12, 121)$	2.876e+01	3.613e+00	1.621e+00
$(13, 13, 144)$	5.814e+01	6.353e+00	3.477e+00
$(14, 14, 169)$	1.071e+02	1.130e+01	6.928e+00
$(15, 15, 196)$	1.937e+02	1.956e+01	8.954e+01
$(16, 16, 225)$	3.298e+02	3.291e+01	9.325e+02

Table: Time estimates in seconds for computing $\mathbf{Q}$ and its Gram matrix.

The link between rank and dimensions is meant to satisfy [Domanov and De Lathauwer 2016] theorem.

Possible new direction

Combining the two approaches

constructing $\mathbf{Q}$ benefiting from $\mathbf{Q}$ structure

⇓

develop† a more convenient matrix-vector product, using $\mathbf{Q}$ structure.

† possibly based on the exterior product

Overview

Intersection shrinking

Once $\mathbf{Q}^{\dagger}$ is computed, the intersection subspace

$$\mathcal{E}_0 = \mathrm{null}(\mathbf{Q}) \cap \mathrm{vec}(\mathrm{Sym}_R^N)$$

has to be computed, cheking its dimension:

- if $\dim \mathcal{E}_0 = R$, then the algorithm proceed;
- if $\dim \mathcal{E}_0 > R$, then the intersection is shrinked, computing $\mathcal{E}_{h+1}$ such that

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

  until $\dim \mathcal{E}_{h+1} = R^{h+1}$ and then proceed.

† or its Gram matrix $\mathbf{Q}^T\mathbf{Q}$.

$\mathcal{E}_0$ basis

If $\mathbf{Q}$ the $(M \times N)$ is matrix whose kernel is needed with $M = C_I^2 C_J^2$ and $N = C_{R+1}^2$

$\mathcal{E}_0$ basis

If $\mathbf{Q}$ the $(M \times N)$ is matrix whose kernel is needed with $M = C_I^2 C_J^2$ and $N = C_{R+1}^2$

1. compute the SVD as $\mathbf{Q} = \mathbf{U\Sigma V}^T$;

$\mathcal{E}_0$ basis

If $\mathbf{Q}$ the $(M \times N)$ is matrix whose kernel is needed with $M = C_I^2 C_J^2$ and $N = C_{R+1}^2$

1. compute the SVD as $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$;
2. define $m \in \mathbb{N}$ such that σ_m is the last singular value larger than $(\varepsilon \max\{M, N\})$;

$\mathcal{E}_0$ basis

If $\mathbf{Q}$ the $(M \times N)$ is matrix whose kernel is needed with $M = C_I^2 C_J^2$ and $N = C_{R+1}^2$

1. compute the SVD as $\mathbf{Q} = \mathbf{U\Sigma V}^T$;
2. define $m \in \mathbb{N}$ such that σ_m is the last singular value larger than $(\varepsilon \max\{M, N\})$;
3. define the basis of null($\mathbf{Q}$) as $\mathbf{F}_0 = [\mathbf{v}_{m+1}, \dots, \mathbf{v}_N]$ with $\mathbf{v}_\ell$ the ℓ-th column of $\mathbf{V}$

$\mathcal{E}_0$ basis

If $\mathbf{Q}$ the $(M \times N)$ is matrix whose kernel is needed with $M = C_I^2 C_J^2$ and $N = C_{R+1}^2$

1. compute the SVD as $\mathbf{Q} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$;
2. define $m \in \mathbb{N}$ such that σ_m is the last singular value larger than $(\varepsilon \max\{M, N\})$;
3. define the basis of null($\mathbf{Q}$) as $\mathbf{F}_0 = [\mathbf{v}_{m+1}, \dots, \mathbf{v}_N]$ with $\mathbf{v}_\ell$ the ℓ-th column of $\mathbf{V}$
4. define the basis of the intersection subspace

$$\mathcal{E}_0 = \text{null}(\mathbf{Q}) \cap \text{vec}(\text{Sym}_R^N)$$

   as $\mathbf{E}_0 = \mathbf{G}_0\mathbf{F}_0$ where the matrix $\mathbf{G}_0$ has size $(R^2 \times N)$ and comes from the symmetrization of the canonical basis of order-2 tensors of size $(R \times R)$;

$\mathcal{E}_0$ basis

If $\mathbf{Q}$ the $(M \times N)$ is matrix whose kernel is needed with $M = C_I^2 C_J^2$ and $N = C_{R+1}^2$

1. compute the SVD as $\mathbf{Q} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$;
2. define $m \in \mathbb{N}$ such that σ_m is the last singular value larger than $(\varepsilon \max\{M, N\})$;
3. define the basis of null($\mathbf{Q}$) as $\mathbf{F}_0 = [\mathbf{v}_{m+1}, \ldots, \mathbf{v}_N]$ with $\mathbf{v}_\ell$ the ℓ-th column of $\mathbf{V}$
4. define the basis of the intersection subspace

$$\mathcal{E}_0 = \mathrm{null}(\mathbf{Q}) \cap \mathrm{vec}(\mathrm{Sym}_R^N)$$

   as $\mathbf{E}_0 = \mathbf{G}_0\mathbf{F}_0$ where the matrix $\mathbf{G}_0$ has size $(R^2 \times N)$ and comes from the symmetrization of the canonical basis of order-2 tensors of size $(R \times R)$;
5. compute $\hat{R}$ the dimension of the intersection as $\hat{R} = N - m$

$\mathcal{E}_0$ basis

If $\mathbf{Q}$ the $(M \times N)$ is matrix whose kernel is needed with $M = C_I^2 C_J^2$ and $N = C_{R+1}^2$

1. compute the SVD as $\mathbf{Q} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$;
2. define $m \in \mathbb{N}$ such that σ_m is the last singular value larger than $(\varepsilon \max\{M, N\})$;
3. define the basis of null($\mathbf{Q}$) as $\mathbf{F}_0 = [\mathbf{v}_{m+1}, \ldots, \mathbf{v}_N]$ with $\mathbf{v}_\ell$ the ℓ-th column of $\mathbf{V}$
4. define the basis of the intersection subspace

$$\mathcal{E}_0 = \mathrm{null}(\mathbf{Q}) \cap \mathrm{vec}(\mathrm{Sym}_R^N)$$

   as $\mathbf{E}_0 = \mathbf{G}_0\mathbf{F}_0$ where the matrix $\mathbf{G}_0$ has size $(R^2 \times N)$ and comes from the symmetrization of the canonical basis of order-2 tensors of size $(R \times R)$;
5. compute $\hat{R}$ the dimension of the intersection as $\hat{R} = N - m$

   5.1 if $\hat{R}$ is equal to R, then reshape $\mathbf{E}_0$ as a $(R \times R \times R)$ and compute its CPD;

$\mathcal{E}_0$ basis

If $\mathbf{Q}$ the $(M \times N)$ is matrix whose kernel is needed with $M = C_I^2 C_J^2$ and $N = C_{R+1}^2$

1. compute the SVD as $\mathbf{Q} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$;
2. define $m \in \mathbb{N}$ such that σ_m is the last singular value larger than $(\varepsilon \max\{M, N\})$;
3. define the basis of null($\mathbf{Q}$) as $\mathbf{F}_0 = [\mathbf{v}_{m+1}, \dots, \mathbf{v}_N]$ with $\mathbf{v}_\ell$ the ℓ-th column of $\mathbf{V}$
4. define the basis of the intersection subspace

$$\mathcal{E}_0 = \text{null}(\mathbf{Q}) \cap \text{vec}(\text{Sym}_R^N)$$

   as $\mathbf{E}_0 = \mathbf{G}_0\mathbf{F}_0$ where the matrix $\mathbf{G}_0$ has size $(R^2 \times N)$ and comes from the symmetrization of the canonical basis of order-2 tensors of size $(R \times R)$;
5. compute $\hat{R}$ the dimension of the intersection as $\hat{R} = N - m$
   5.1 if $\hat{R}$ is equal to R, then reshape $\mathbf{E}_0$ as a $(R \times R \times R)$ and compute its CPD;
   5.2 if $\hat{R}$ is *not* equal to R, then shrink the intersection until $\hat{R} = R^{h+1}$.

$\mathcal{E}_{h+1}$ basis

If the matrix $\mathbf{E}_0$ of size $(R^2 \times \hat{R})$ span the intersection subspace $\mathcal{E}_0$ with $\hat{R} > R$ until $\hat{R}$ is equal to R^{h+1}, set $h = 0$ and repeat

$\mathcal{E}_{h+1}$ basis

If the matrix $\mathbf{E}_0$ of size $(R^2 \times \hat{R})$ span the intersection subspace $\mathcal{E}_0$ with $\hat{R} > R$ until $\hat{R}$ is equal to R^{h+1}, set $h = 0$ and repeat

1. compute $\mathbf{F}_{h+1}$ the basis of
$$\mathcal{F}_{h+1} = \mathbb{K}^R \otimes \mathcal{E}_h$$
by a Kronecker product between the identity of order R and $\mathbf{E}_h$ the basis of $\mathcal{E}_h$;
2. compute $\mathbf{E}_{h+1}$ the basis of
$$\mathcal{E}_{h+1} = \mathcal{F}_{h+1} \cap \mathrm{vec}\left(\mathrm{Sym}_R^{(h+1)+2}\right)$$
by symmetrizing the columns of $\mathbf{F}_{h+1}$;

$\mathcal{E}_{h+1}$ basis

If the matrix $\mathbf{E}_0$ of size $(R^2 \times \hat{R})$ span the intersection subspace $\mathcal{E}_0$ with $\hat{R} > R$ until $\hat{R}$ is equal to R^{h+1}, set $h = 0$ and repeat

1. compute $\mathbf{F}_{h+1}$ the basis of
$$\mathcal{F}_{h+1} = \mathbb{K}^R \otimes \mathcal{E}_h$$
by a Kronecker product between the identity of order R and $\mathbf{E}_h$ the basis of $\mathcal{E}_h$;
2. compute $\mathbf{E}_{h+1}$ the basis of
$$\mathcal{E}_{h+1} = \mathcal{F}_{h+1} \cap \mathrm{vec}\left(\mathrm{Sym}_R^{(h+1)+2}\right)$$
by symmetrizing the columns of $\mathbf{F}_{h+1}$;
3. orthogonalize $\mathbf{E}_{h+1}$;

$\mathcal{E}_{h+1}$ basis

If the matrix $\mathbf{E}_0$ of size $(R^2 \times \hat{R})$ span the intersection subspace $\mathcal{E}_0$ with $\hat{R} > R$ until $\hat{R}$ is equal to R^{h+1}, set $h = 0$ and repeat

1. compute $\mathbf{F}_{h+1}$ the basis of
$$\mathcal{F}_{h+1} = \mathbb{K}^R \otimes \mathcal{E}_h$$
by a Kronecker product between the identity of order R and $\mathbf{E}_h$ the basis of $\mathcal{E}_h$;
2. compute $\mathbf{E}_{h+1}$ the basis of
$$\mathcal{E}_{h+1} = \mathcal{F}_{h+1} \cap \mathrm{vec}\big(\mathrm{Sym}_R^{(h+1)+2}\big)$$
by symmetrizing the columns of $\mathbf{F}_{h+1}$;
3. orthogonalize $\mathbf{E}_{h+1}$;
4. check how many of $\mathbf{E}_{h+1}$ columns satisfy the symmetric conditions, updating $\hat{R}$.

Possible tool

Currently to avoid forming $\mathbf{F}_{h+1}$ and the symmetrizer matrix $\mathbf{G}_{h+1}$

1. compute $\mathbf{C}_{h+1}$ matrix of size $(R\hat{R} \times C^{h+2}_{R+2+h})$ carrying their product information;

Possible tool

Currently to avoid forming $\mathbf{F}_{h+1}$ and the symmetrizer matrix $\mathbf{G}_{h+1}$

1. compute $\mathbf{C}_{h+1}$ matrix of size $(R\hat{R} \times C^{h+2}_{R+2+h})$ carrying their product information;
2. orthogonalize $\mathbf{C}_{h+1}$;

Possible tool

Currently to avoid forming $\mathbf{F}_{h+1}$ and the symmetrizer matrix $\mathbf{G}_{h+1}$

1. compute $\mathbf{C}_{h+1}$ matrix of size $(R\hat{R} \times C^{h+2}_{R+2+h})$ carrying their product information;
2. orthogonalize $\mathbf{C}_{h+1}$;
3. use $\mathbf{C}_{h+1}$ as projector to obtain $\mathbf{E}_{h+1}$;

Possible tool

Currently to avoid forming $\mathbf{F}_{h+1}$ and the symmetrizer matrix $\mathbf{G}_{h+1}$

1. compute $\mathbf{C}_{h+1}$ matrix of size $(R\hat{R} \times C^{h+2}_{R+2+h})$ carrying their product information;
2. orthogonalize $\mathbf{C}_{h+1}$;
3. use $\mathbf{C}_{h+1}$ as projector to obtain $\mathbf{E}_{h+1}$;

and this steps are repeated until R^{h+1} and $\hat{R}$ coincide.

Possible tool

Currently to avoid forming $\mathbf{F}_{h+1}$ and the symmetrizer matrix $\mathbf{G}_{h+1}$

1. compute $\mathbf{C}_{h+1}$ matrix of size $(R\hat{R} \times C^{h+2}_{R+2+h})$ carrying their product information;
2. orthogonalize $\mathbf{C}_{h+1}$;
3. use $\mathbf{C}_{h+1}$ as projector to obtain $\mathbf{E}_{h+1}$;

and this steps are repeated until R^{h+1} and $\hat{R}$ coincide.

⇩

Study theoretically the enlargement-shrinking process to estimate m a priori and orthogonalize just once[†]

† or design an heuristic method

Overview

Auxiliary CPD: current state

Let $\mathbf{E}_{h+1}$ the basis of $\mathcal{E}_{h+1}$ has size $(R^{h+1} \times R)$, then

Auxiliary CPD: current state

Let $\mathbf{E}_{h+1}$ the basis of $\mathcal{E}_{h+1}$ has size $(R^{h+1} \times R)$, then

1. reshape $\mathbf{E}_{h+1}$ as a $(R^h \times R^2)$ matrix;

Auxiliary CPD: current state

Let $\mathbf{E}_{h+1}$ the basis of $\mathcal{E}_{h+1}$ has size $(R^{h+1} \times R)$, then

1. reshape $\mathbf{E}_{h+1}$ as a $(R^h \times R^2)$ matrix;
2. compute $\mathbf{R}$ the `R` factor from the `QR-factorization` of $\mathbf{E}_{h+1}$;

Auxiliary CPD: current state

Let $\mathbf{E}_{h+1}$ the basis of $\mathcal{E}_{h+1}$ has size $(R^{h+1} \times R)$, then

1. reshape $\mathbf{E}_{h+1}$ as a $(R^h \times R^2)$ matrix;
2. compute $\mathbf{R}$ the `R` factor from the `QR-factorization` of $\mathbf{E}_{h+1}$;
3. truncate $\mathbf{R}$ to size $(R \times R^2)$ matrix;

Auxiliary CPD: current state

Let $\mathbf{E}_{h+1}$ the basis of $\mathcal{E}_{h+1}$ has size $(R^{h+1} \times R)$, then

1. reshape $\mathbf{E}_{h+1}$ as a $(R^h \times R^2)$ matrix;
2. compute $\mathbf{R}$ the `R` factor from the `QR-factorization` of $\mathbf{E}_{h+1}$;
3. truncate $\mathbf{R}$ to size $(R \times R^2)$ matrix;
4. reshape $\mathbf{R}$ as a tensor $\mathcal{R}$ of size $(R \times R \times R)$;

Auxiliary CPD: current state

Let $\mathbf{E}_{h+1}$ the basis of $\mathcal{E}_{h+1}$ has size $(R^{h+1} \times R)$, then

1. reshape $\mathbf{E}_{h+1}$ as a $(R^h \times R^2)$ matrix;
2. compute $\mathbf{R}$ the `R` factor from the `QR-factorization` of $\mathbf{E}_{h+1}$;
3. truncate $\mathbf{R}$ to size $(R \times R^2)$ matrix;
4. reshape $\mathbf{R}$ as a tensor $\mathcal{R}$ of size $(R \times R \times R)$;
5. compute the CPD of $\mathcal{R}$ getting the factor $\mathbf{C}$ of size $(R \times R)$.

Possible tool

- currently use Jenrich's algorithm, based on `GEVD`[†];

[†] problematic numerical stability Beltrán, Breiding, et al. 2019.

Possible tool

- currently use Jenrich's algorithm, based on `GEVD`†;
- use an optimization based CPD algorithm;

† problematic numerical stability Beltrán, Breiding, et al. 2019.

Possible tool

- currently use Jenrich's algorithm, based on `GEVD`[†];
- use an optimization based CPD algorithm;
- use an Generalized EigenSpace based CPD algorithm Evert E. 2020;

[†] problematic numerical stability Beltrán, Breiding, et al. 2019.

Possible tool

- currently use Jenrich's algorithm, based on `GEVD`[†];
- use an optimization based CPD algorithm;
- use an Generalized EigenSpace based CPD algorithm Evert E. 2020;
- try to preserve (partially)-symmetric properties and tune CPD algorithm.

[†] problematic numerical stability Beltrán, Breiding, et al. 2019.

Overview

Summary

Current state

- Blind Source Separation problem;

Summary

Current state

- Blind Source Separation problem;
- conditions that guarantee generic uniqueness;

Summary

Current state

- Blind Source Separation problem;
- conditions that guarantee generic uniqueness;
- conditions transferred from BSS to CPD context;

Summary

Current state

- Blind Source Separation problem;
- conditions that guarantee generic uniqueness;
- conditions transferred from BSS to CPD context;
- outlined a CPD algebraic algorithm;

Summary

Current state

- Blind Source Separation problem;
- conditions that guarantee generic uniqueness;
- conditions transferred from BSS to CPD context;
- outlined a CPD algebraic algorithm;
- designed a first implementation;

Summary

Current state

- Blind Source Separation problem;
- conditions that guarantee generic uniqueness;
- conditions transferred from BSS to CPD context;
- outlined a CPD algebraic algorithm;
- designed a first implementation;

Work in progress

Summary

Current state

- Blind Source Separation problem;
- conditions that guarantee generic uniqueness;
- conditions transferred from BSS to CPD context;
- outlined a CPD algebraic algorithm;
- designed a first implementation;

Work in progress

- using matrix structure in the computation of $\mathbf{Q}$;
  - exterior algebra could be a tool
  - Gram matrix doesn't seem to be beneficial

Summary

Current state

- Blind Source Separation problem;
- conditions that guarantee generic uniqueness;
- conditions transferred from BSS to CPD context;
- outlined a CPD algebraic algorithm;
- designed a first implementation;

Work in progress

- using matrix structure in the computation of $\mathbf{Q}$;
  - exterior algebra could be a tool
  - Gram matrix doesn't seem to be beneficial
- estimate the order of the symmetric tensor to product an auxiliary tensor of proper size;

Summary

Current state

- Blind Source Separation problem;
- conditions that guarantee generic uniqueness;
- conditions transferred from BSS to CPD context;
- outlined a CPD algebraic algorithm;
- designed a first implementation;

Work in progress

- using matrix structure in the computation of $\mathbf{Q}$;
  - exterior algebra could be a tool
  - Gram matrix doesn't seem to be beneficial
- estimate the order of the symmetric tensor to product an auxiliary tensor of proper size;
- using the (partially)-symmetric structure into CPD of the auxiliary tensor.

Thank you for the attention!
Questions? Advice?

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.
- Comon, P. and C. Jutten (2009). *Handbook of blind source separation: Independent component analysis and applications*. Academic press.
- 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.
- — (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.

References II

Evert E. Vandecappelle M., De Lathauwer L. (2020). *A recursive eigenspace computation for the canonical polyadic decomposition*. Tech. rep. 20-80. Leuven, Belgium: ESAT-STADIUS, KU Leuven.