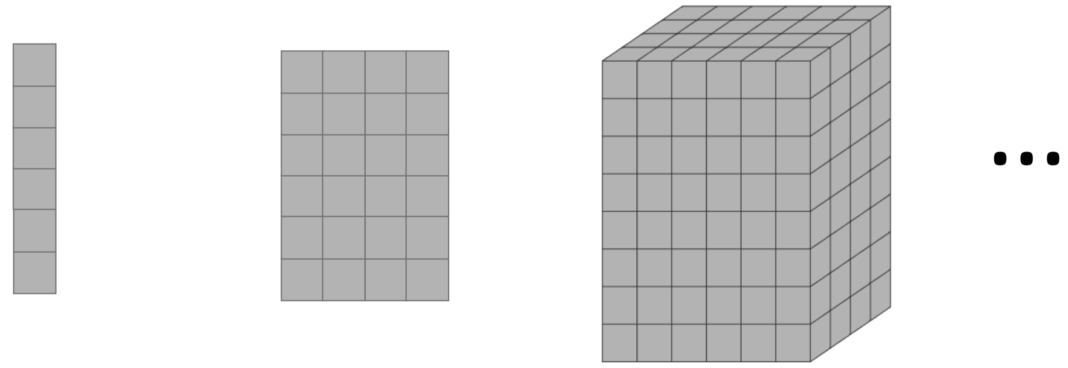
EE613 - Machine Learning for Engineers

TENSOR REGRESSION

Sylvain Calinon
Robot Learning and Interaction Group
Idiap Research Institute
Oct 7, 2021

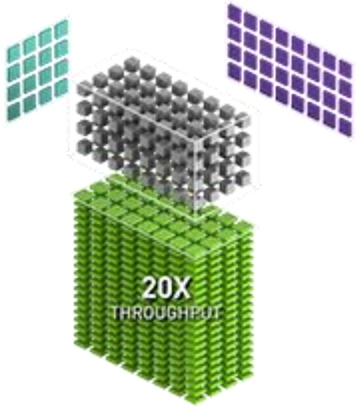
Tensors



1st-order tensors

2nd-order tensors

3rd-order tensors



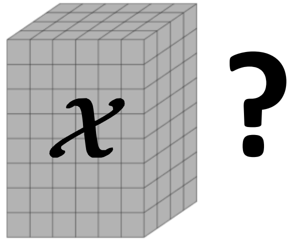
Images: 3D tensors
(width, height, color channels)

Videos: 4D tensors
(frame, width, height, color channels)

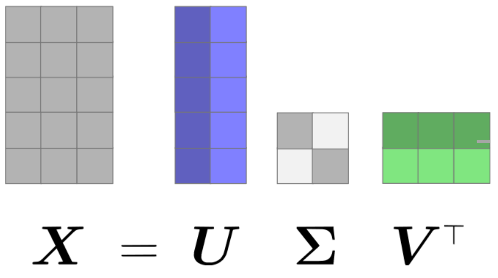
Tensors appear in various forms:

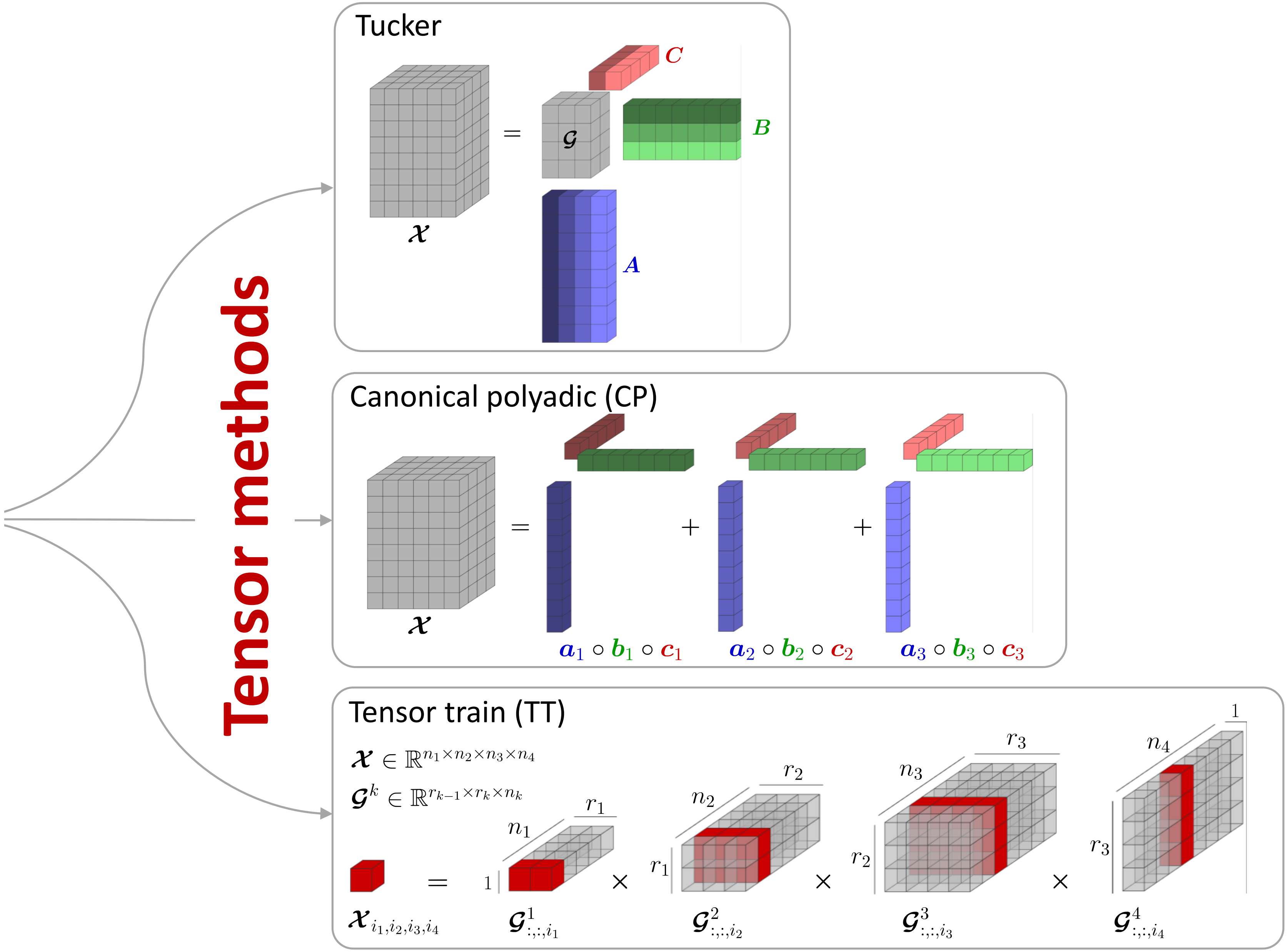
- Raw data
  (arrays of sensors, multidimensional channels)
- Data evolution over time window
  (sets of short sequences)
- Data in multiple coordinate systems
- Basis functions expansion

Tensor methods – Motivation



Matrix factorization with standard linear algebra:





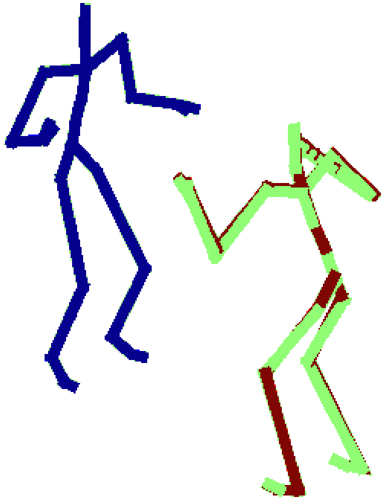
Tensor methods – Motivation

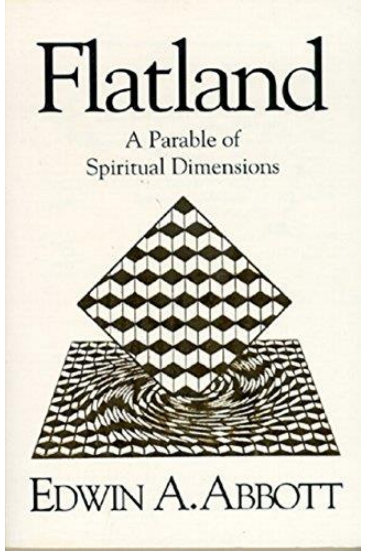
joint
agent
coordinate
sample
time step

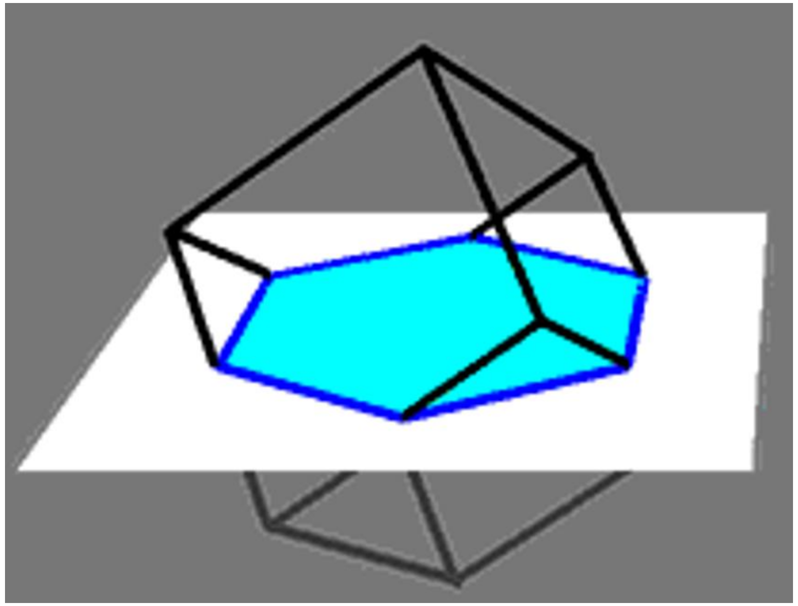
$$\mathcal{X} \in \mathbb{R}^{10 \times 2 \times 31 \times 3 \times 100} \qquad \boldsymbol{X} \in \mathbb{R}^{10 \times 18600}$$

Tensor factorization

→ Multiway analysis of the data







Tensors can reveal simpler underlying structures in the data

Tensor methods - Motivation

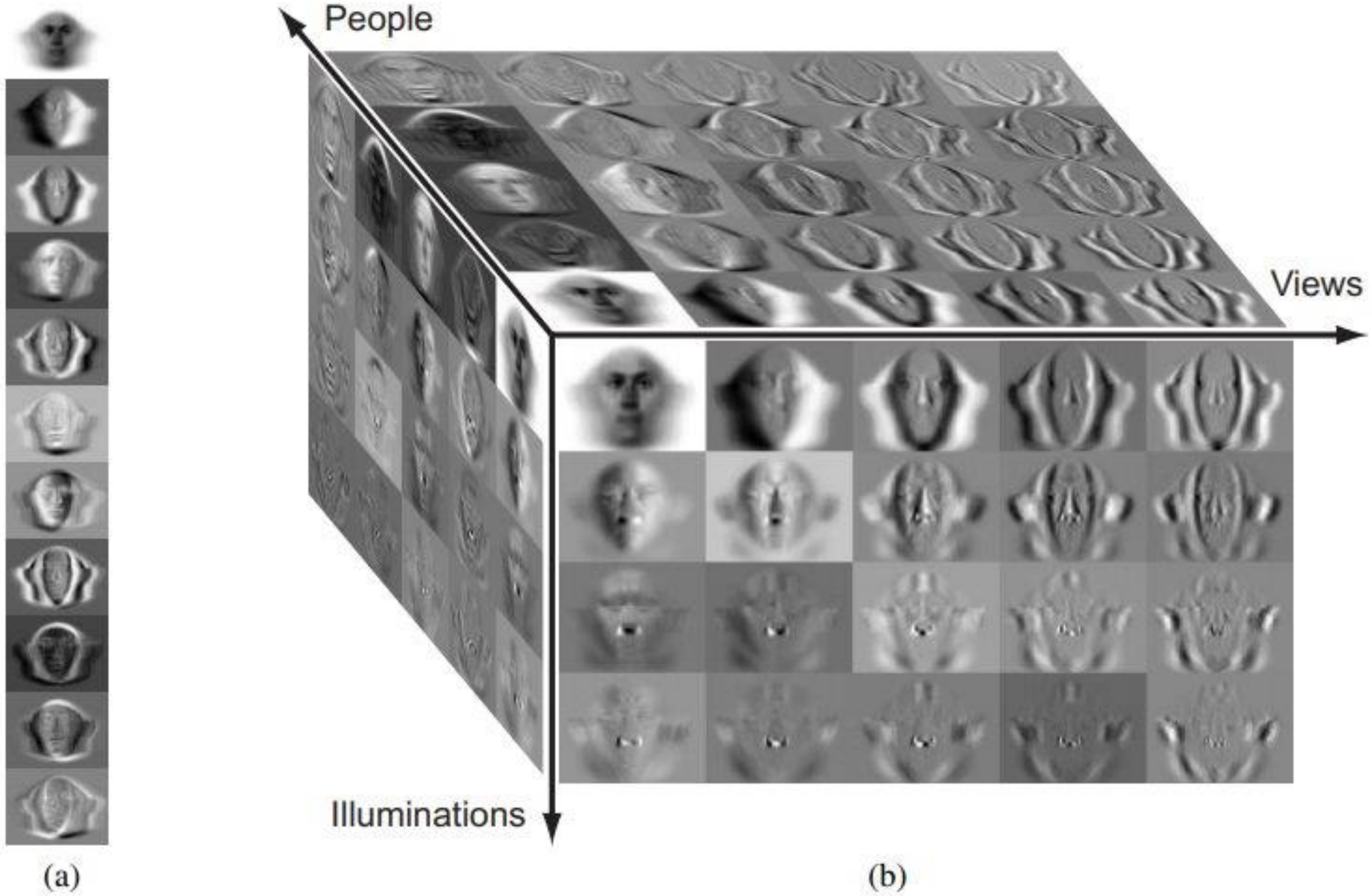
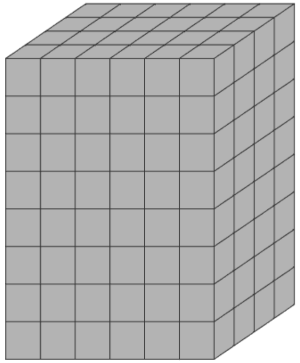


Figure 1: Eigenfaces and TensorFaces bases for an ensemble of 2,700 facial images spanning 75 people, each imaged under 6 viewing and 6 illumination conditions (see Section 5). (a) PCA eigenvectors (eigenfaces), which are the principal axes of variation across all images. (b) A partial visualization of the $75 \times 6 \times 6 \times 8560$ TensorFaces representation of $\mathcal{D}$, obtained as $\mathcal{T} = \mathcal{Z} \times_4 \mathbf{U}_{\text{pixels}}$.

[Vasilescu and Terzopoulos, CVPR'2005]

Tensor indexing - Slices and fibers



$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{8\times6\times4}$

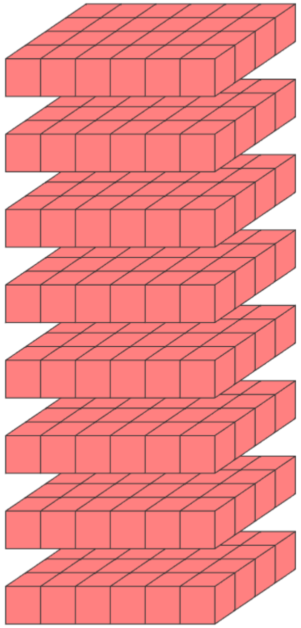
$\boldsymbol{\mathcal{X}}$ tensor

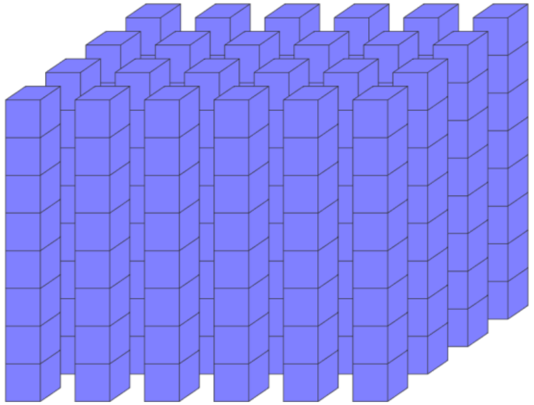
$\boldsymbol{X}$ matrix

$\boldsymbol{x}$ vector

x scalar

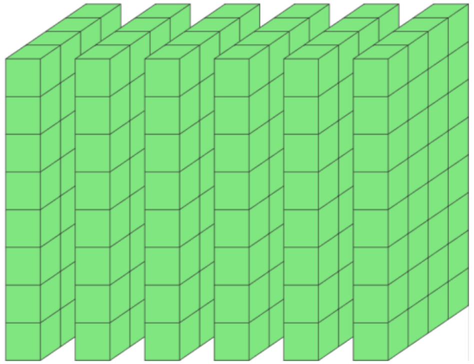
$\boldsymbol{X}_{i,:,:}$ (horizontal slice)

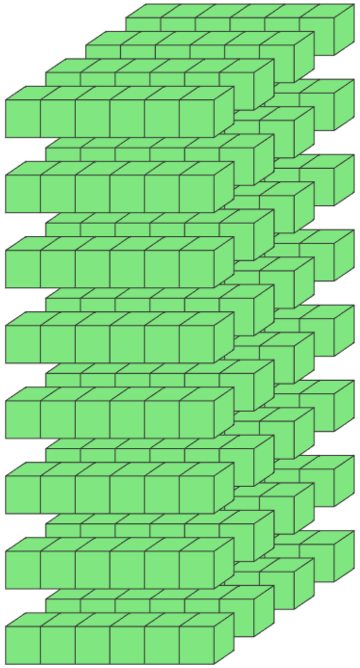




$\boldsymbol{x}_{:,j,k}$ (column fiber)

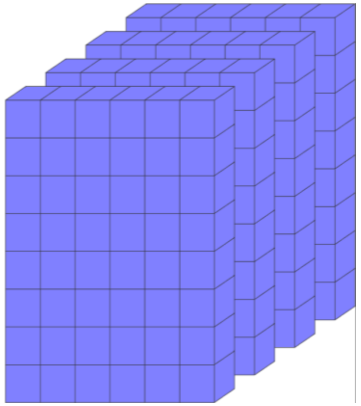
$\boldsymbol{X}_{:,j,:}$ (lateral slice)

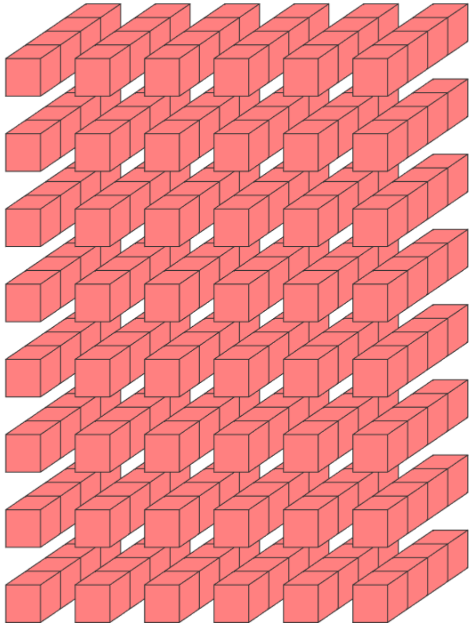




$\boldsymbol{x}_{i,:,k}$ (row fiber)

$\boldsymbol{X}_{:,:,k}$ (frontal slice)

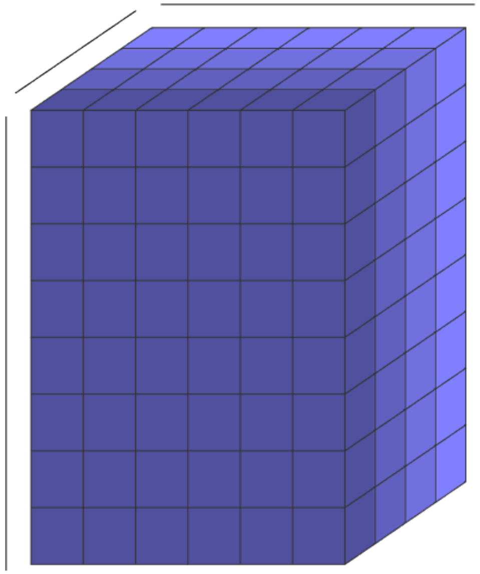




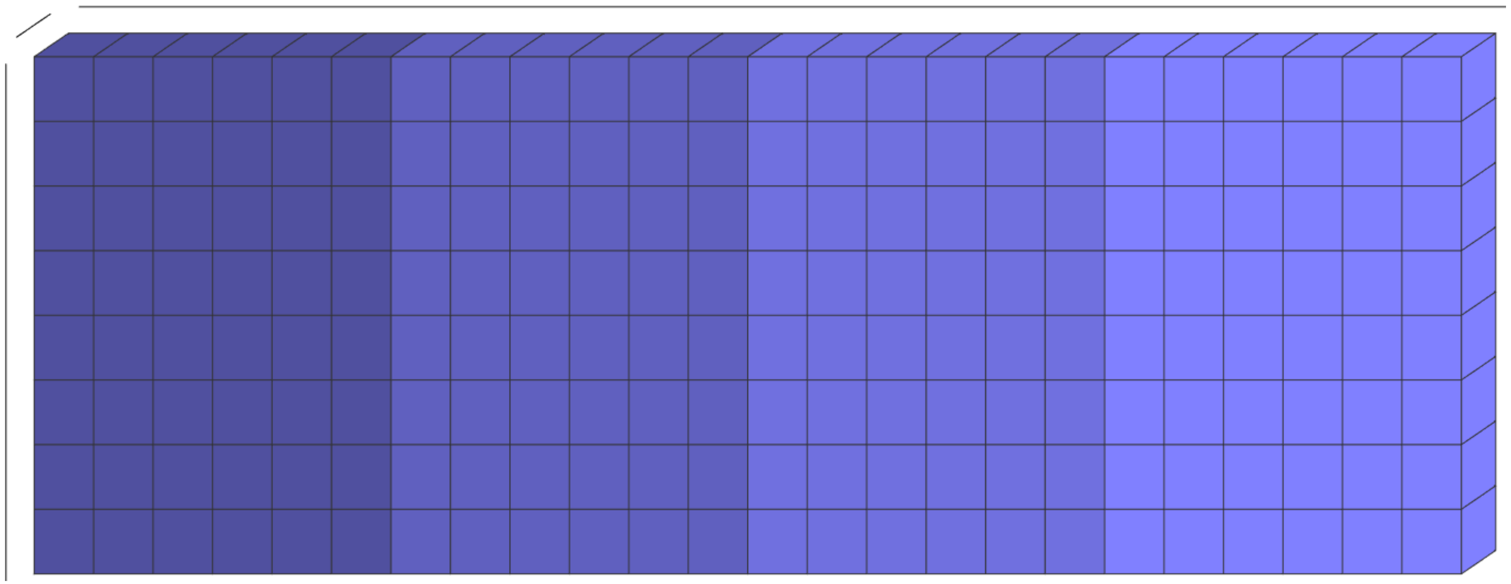
$\boldsymbol{x}_{i,j,:}$ (tube fiber)

Tensor matricization / unfolding

A matrix $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \cdots I_{n-1} I_{n+1} \cdots I_N)}$ results from the mode-n matricization (unfolding) of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, which consists of turning the mode-n fibers of $\boldsymbol{\mathcal{X}}$ into the columns of a matrix $\boldsymbol{X}_{(n)}$.



$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{8 \times 6 \times 4}$



$\boldsymbol{X}_{(1)} \in \mathbb{R}^{8 \times 24}$

(mode-1 unfolding)

Products (Hadamard, Kronecker, Khatri-Rao)

Hadamard (elementwise)

$$\boldsymbol{A} * \boldsymbol{B} = \begin{bmatrix} a_{1,1}b_{1,1} & a_{1,2}b_{1,2} & \cdots & a_{1,J}b_{1,J} \\ a_{2,1}b_{2,1} & a_{2,2}b_{2,2} & \cdots & a_{2,J}b_{2,J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I,1}b_{I,1} & a_{I,2}b_{I,2} & \cdots & a_{I,J}b_{I,J} \end{bmatrix}$$

$$\boldsymbol{A} \in \mathbb{R}^{I \times J} \quad \boldsymbol{B} \in \mathbb{R}^{I \times J} \quad \boldsymbol{A} * \boldsymbol{B} \in \mathbb{R}^{I \times J}$$

Kronecker

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{1,1}\boldsymbol{B} & a_{1,2}\boldsymbol{B} & \cdots & a_{1,J}\boldsymbol{B} \\ a_{2,1}\boldsymbol{B} & a_{2,2}\boldsymbol{B} & \cdots & a_{2,J}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I,1}\boldsymbol{B} & a_{I,2}\boldsymbol{B} & \cdots & a_{I,J}\boldsymbol{B} \end{bmatrix}$$

$$\boldsymbol{A} \in \mathbb{R}^{I \times J} \quad \boldsymbol{B} \in \mathbb{R}^{K \times L} \quad \boldsymbol{A} \otimes \boldsymbol{B} \in \mathbb{R}^{IK \times JL}$$

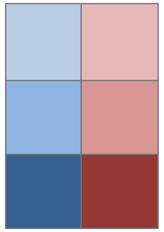
Khatri-Rao

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} a_{1,1}\boldsymbol{b}_1 & a_{1,2}\boldsymbol{b}_2 & \cdots & a_{1,K}\boldsymbol{b}_K \\ a_{2,1}\boldsymbol{b}_1 & a_{2,2}\boldsymbol{b}_2 & \cdots & a_{2,K}\boldsymbol{b}_K \\ \vdots & \vdots & \ddots & \vdots \\ a_{I,1}\boldsymbol{b}_1 & a_{I,2}\boldsymbol{b}_2 & \cdots & a_{I,K}\boldsymbol{b}_K \end{bmatrix}$$

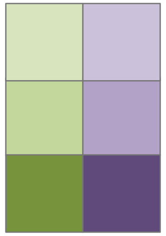
$$\boldsymbol{A} \in \mathbb{R}^{I \times K} \quad \boldsymbol{B} \in \mathbb{R}^{J \times K} \quad \boldsymbol{A} \odot \boldsymbol{B} \in \mathbb{R}^{IJ \times K}$$

Hadamard (elementwise) product - Example

$$A \in \mathbb{R}^{3\times 2}$$
$$B \in \mathbb{R}^{3\times 2}$$
$$A * B \in \mathbb{R}^{3\times 2}$$

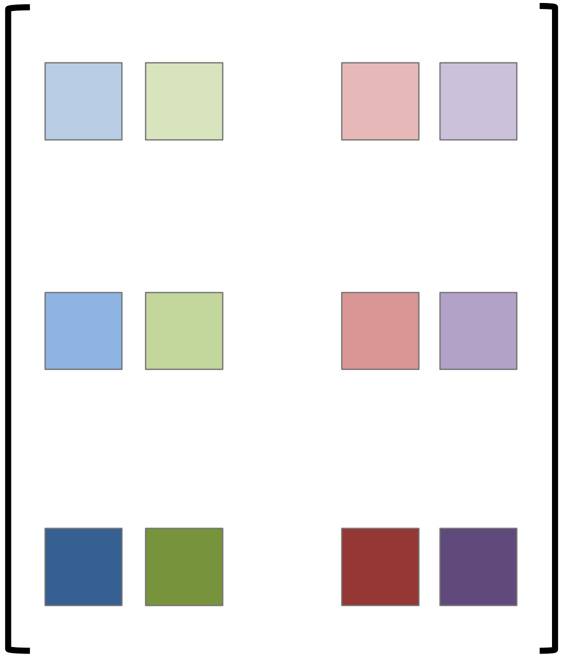


A



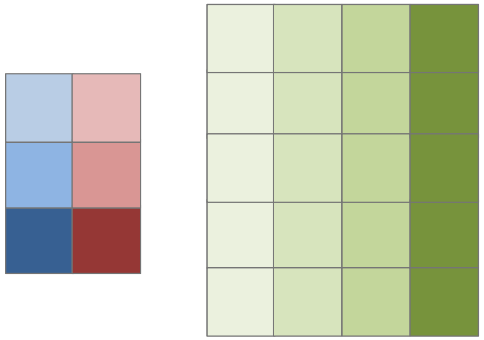
B

$A * B =$



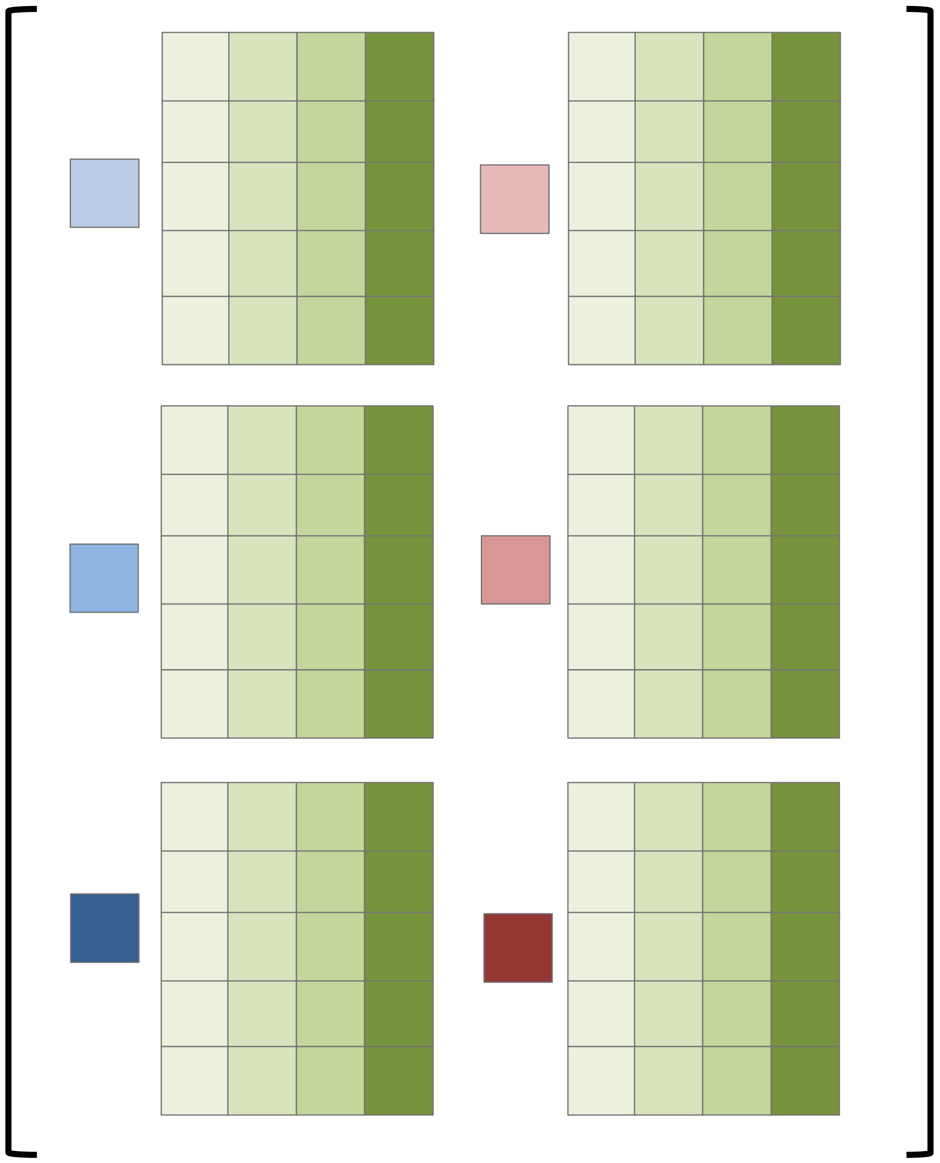
Kronecker product - Example

$$\boldsymbol{A} \in \mathbb{R}^{3\times 2}$$
$$\boldsymbol{B} \in \mathbb{R}^{5\times 4}$$
$$\boldsymbol{A} \otimes \boldsymbol{B} \in \mathbb{R}^{15\times 8}$$



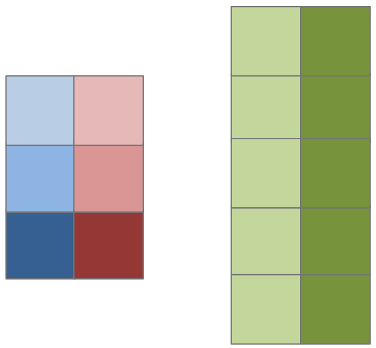
$\boldsymbol{A}$ $\boldsymbol{B}$

$\boldsymbol{A} \otimes \boldsymbol{B} =$

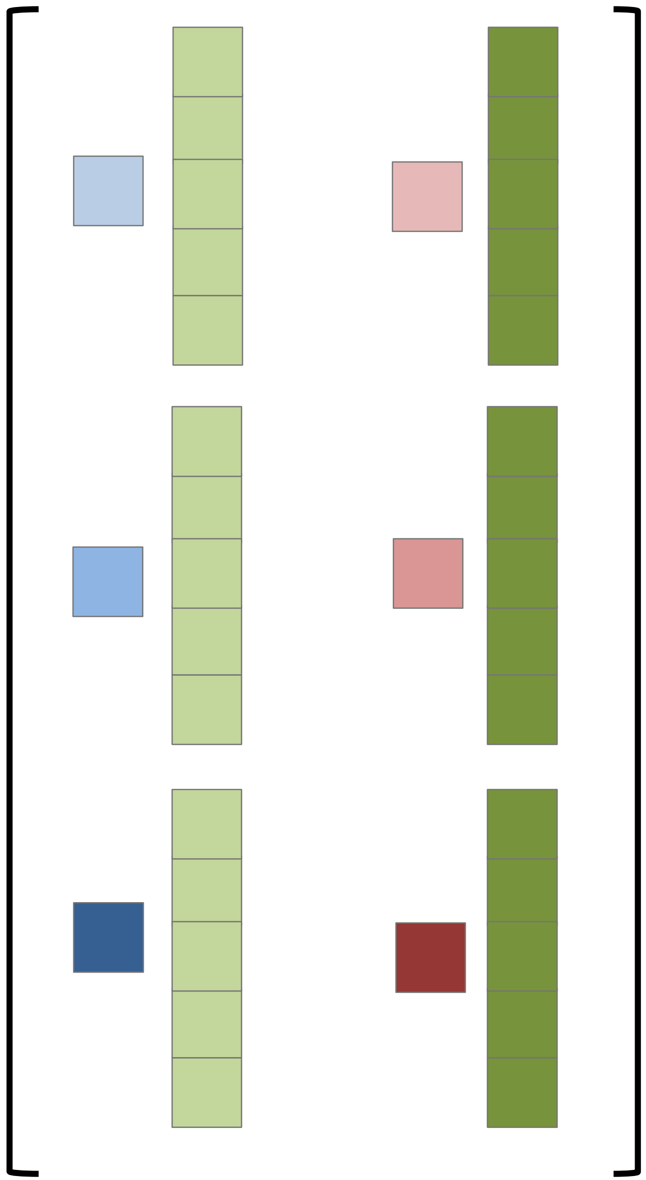


Khatri-Rao product - Example

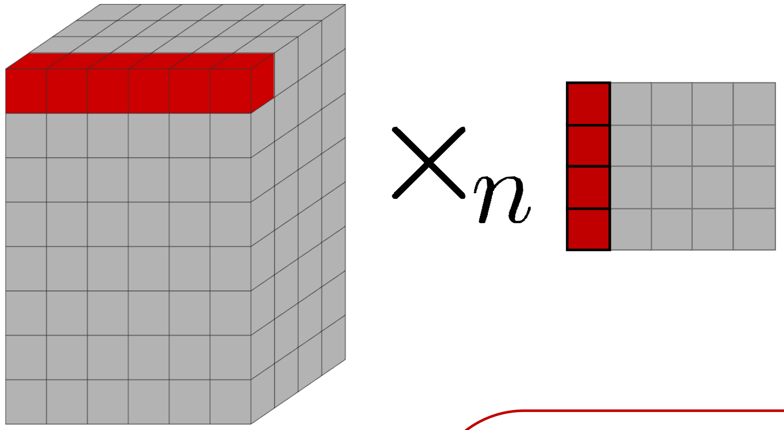
$$\boldsymbol{A} \in \mathbb{R}^{3\times 2}$$
$$\boldsymbol{B} \in \mathbb{R}^{5\times 2}$$
$$\boldsymbol{A} \odot \boldsymbol{B} \in \mathbb{R}^{15\times 2}$$



$\boldsymbol{A}$ $\quad$ $\boldsymbol{B}$

$\boldsymbol{A} \odot \boldsymbol{B} =$ 

Mode-n product



$$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$$
$$\boldsymbol{M} \in \mathbb{R}^{J \times I_n}$$
$$\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$$

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \boldsymbol{M}$$

$$\boldsymbol{Y}_{(n)} = \boldsymbol{M}\boldsymbol{X}_{(n)} \quad \text{(matricized form)}$$

$$y_{i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1,\ldots,i_N}\, m_{j,i_n} \quad \text{(elementwise)}$$

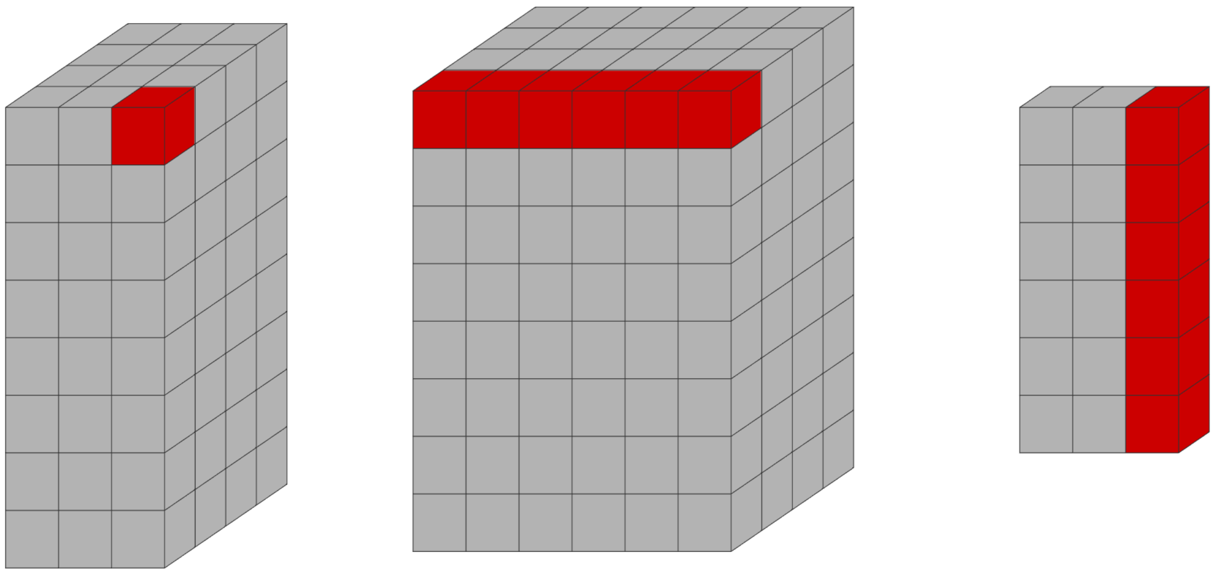
Intuitively, the operation corresponds to multiplying each mode-n fiber of $\boldsymbol{\mathcal{X}}$ by the matrix $\boldsymbol{M}$.

Mode-n product - Example

$\mathcal{X} \in \mathbb{R}^{8\times 6\times 4}$

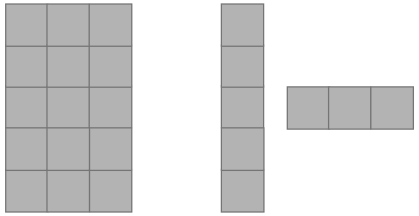
$\boldsymbol{M} \in \mathbb{R}^{6\times 3}$

$\mathcal{Y} \in \mathbb{R}^{8\times 3\times 4}$



$$\mathcal{Y} = \mathcal{X} \times_2 \boldsymbol{M}$$

Outer product and inner product

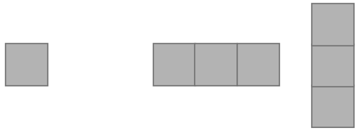


$$\boldsymbol{X} = \boldsymbol{a} \quad \boldsymbol{b}^\top$$
$$= \boldsymbol{a} \circ \boldsymbol{b}$$

(outer product)

The **outer product** of two vectors $\boldsymbol{a} \in \mathbb{R}^I$ and $\boldsymbol{b} \in \mathbb{R}^J$ results in a matrix $\boldsymbol{X} \in \mathbb{R}^{I \times J}$ denoted by $\boldsymbol{X} = \boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^\top$.

The **outer product** of three (or more) vectors $\boldsymbol{a} \in \mathbb{R}^I$, $\boldsymbol{b} \in \mathbb{R}^J$ and $\boldsymbol{c} \in \mathbb{R}^K$ results in a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ denoted by $\boldsymbol{\mathcal{X}} = \boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c}$ with elements $x_{i,j,k} = a_i \, b_j \, c_k$.



$$x = \boldsymbol{a}^\top \quad \boldsymbol{b}$$
$$= \langle \boldsymbol{a} \, , \, \boldsymbol{b} \rangle$$

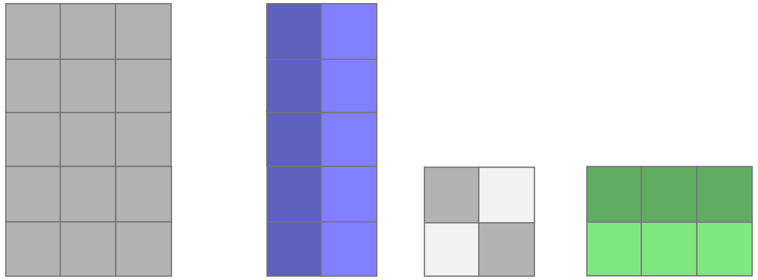
(inner product)

The **inner product** of two vectors $\boldsymbol{a} \in \mathbb{R}^I$ and $\boldsymbol{b} \in \mathbb{R}^I$ results in a scalar $x = \langle \boldsymbol{a}, \boldsymbol{b} \rangle = \boldsymbol{a}^\top \boldsymbol{b} = \sum_{i=1}^{I} a_i b_i$.

The formulation can be extended to tensors $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ of the same size. We have

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \langle \boldsymbol{A}_{(n)}, \boldsymbol{B}_{(n)} \rangle = \langle \mathrm{vec}(\boldsymbol{\mathcal{A}}), \mathrm{vec}(\boldsymbol{\mathcal{B}}) \rangle.$$

Singular value decomposition (SVD)



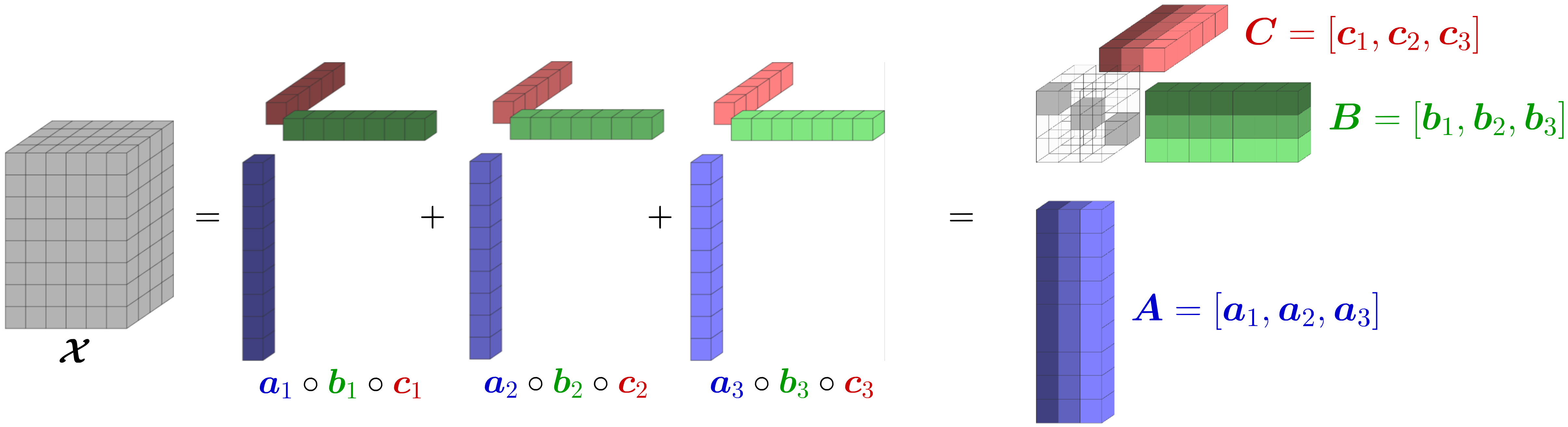
$$\boldsymbol{X} = \boldsymbol{U} \; \boldsymbol{\Sigma} \; \boldsymbol{V}^\top$$

$$\tilde{\boldsymbol{u}}_i = \sigma_i \boldsymbol{u}_i$$

$$\tilde{\boldsymbol{v}}_i = \sigma_i \boldsymbol{v}_i$$

$$
\begin{aligned}
&= \sigma_1^2 \, \boldsymbol{u}_1 \, \boldsymbol{v}_1^\top + \sigma_2^2 \, \boldsymbol{u}_2 \, \boldsymbol{v}_2^\top \\
&= \tilde{\boldsymbol{u}}_1 \, \tilde{\boldsymbol{v}}_1^\top + \tilde{\boldsymbol{u}}_2 \, \tilde{\boldsymbol{v}}_2^\top \\
&= \tilde{\boldsymbol{u}}_1 \circ \tilde{\boldsymbol{v}}_1 + \tilde{\boldsymbol{u}}_2 \circ \tilde{\boldsymbol{v}}_2
\end{aligned}
$$

CP decomposition



CP decomposition

$$\begin{aligned} \boldsymbol{\mathcal{X}} &= \textstyle\sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r \\ &= [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!] \end{aligned}$$

Matricized form:

$$\begin{aligned} \boldsymbol{X}_{(1)} &= \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^\top \\ \boldsymbol{X}_{(2)} &= \boldsymbol{B}(\boldsymbol{C} \odot \boldsymbol{A})^\top \\ \boldsymbol{X}_{(3)} &= \boldsymbol{C}(\boldsymbol{B} \odot \boldsymbol{A})^\top \end{aligned}$$

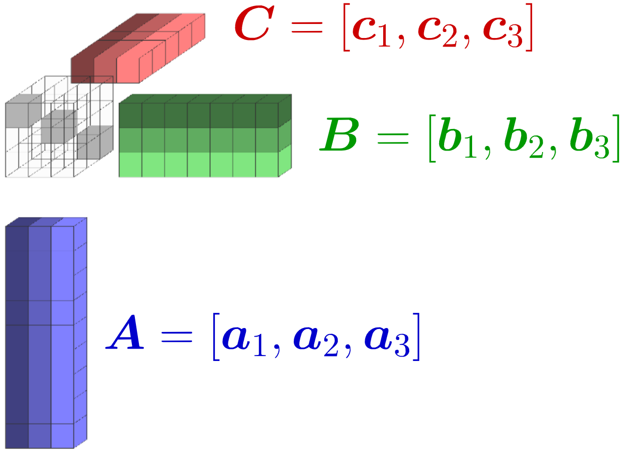
Vectorized form:

$$\mathrm{vec}(\boldsymbol{\mathcal{X}}) = (\boldsymbol{C} \odot \boldsymbol{B} \odot \boldsymbol{A})\,\mathbf{1}_R$$

Elementwise:

$$x_{i,j,k} = \textstyle\sum_{r=1}^{R} a_{i,r}\, b_{j,r}\, c_{k,r}$$

$$\boldsymbol{A} \in \mathbb{R}^{I \times R}$$
$$\boldsymbol{B} \in \mathbb{R}^{J \times R}$$
$$\boldsymbol{C} \in \mathbb{R}^{K \times R}$$



$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_R]$ is called a factor matrix.

The **tensor rank** R corresponds to the smallest number of components required in the CP decomposition.

Parameters estimation: Alternating least squares (ALS)

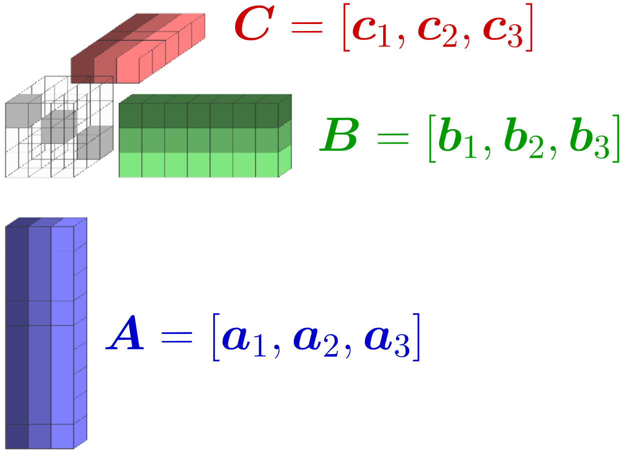
$$A \in \mathbb{R}^{I\times R}$$
$$B \in \mathbb{R}^{J\times R}$$
$$C \in \mathbb{R}^{K\times R}$$

The CP decomposition can be solved by alternating least squares (ALS), by repeating

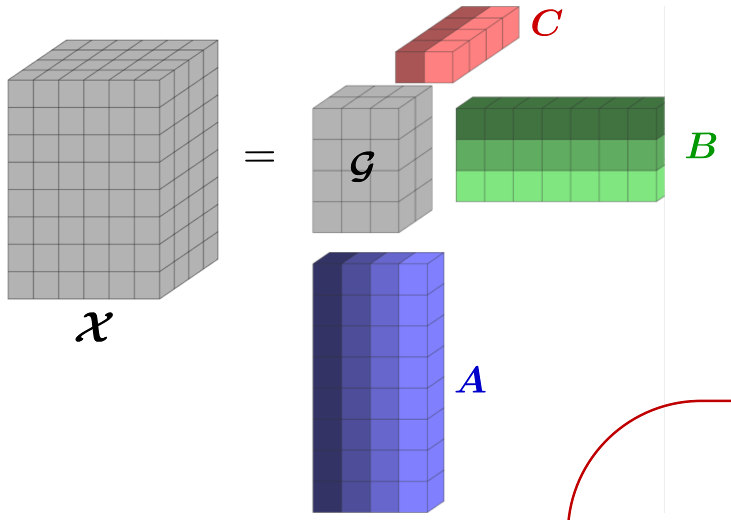
$$\boldsymbol{A} \leftarrow \arg\min_{\boldsymbol{A}} \left\|\boldsymbol{X}_{(1)} - \boldsymbol{A}(\boldsymbol{C}\odot\boldsymbol{B})^\top\right\|_F^2$$
$$\boldsymbol{B} \leftarrow \arg\min_{\boldsymbol{B}} \left\|\boldsymbol{X}_{(2)} - \boldsymbol{B}(\boldsymbol{C}\odot\boldsymbol{A})^\top\right\|_F^2$$
$$\boldsymbol{C} \leftarrow \arg\min_{\boldsymbol{C}} \left\|\boldsymbol{X}_{(3)} - \boldsymbol{C}(\boldsymbol{B}\odot\boldsymbol{A})^\top\right\|_F^2$$

until convergence, yielding the update rules

$$\boldsymbol{A} \leftarrow \boldsymbol{X}_{(1)}\left((\boldsymbol{C}\odot\boldsymbol{B})^\top\right)^\dagger$$
$$\boldsymbol{B} \leftarrow \boldsymbol{X}_{(2)}\left((\boldsymbol{C}\odot\boldsymbol{A})^\top\right)^\dagger$$
$$\boldsymbol{C} \leftarrow \boldsymbol{X}_{(3)}\left((\boldsymbol{B}\odot\boldsymbol{A})^\top\right)^\dagger$$



Tucker decomposition



Core tensor

$\boldsymbol{\mathcal{G}} \in \mathbb{R}^{P \times Q \times R}$

$\boldsymbol{A} \in \mathbb{R}^{I \times P}$

$\boldsymbol{B} \in \mathbb{R}^{J \times Q}$

$\boldsymbol{C} \in \mathbb{R}^{K \times R}$

$$
\begin{aligned}
\boldsymbol{\mathcal{X}} &= \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{p,q,r} \, \boldsymbol{a}_p \circ \boldsymbol{b}_q \circ \boldsymbol{c}_r \\
&= \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} \\
&= [\![\boldsymbol{\mathcal{G}}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]
\end{aligned}
$$

Matricized form:

$$
\begin{aligned}
\boldsymbol{X}_{(1)} &= \boldsymbol{A}\boldsymbol{G}_{(1)}(\boldsymbol{C} \otimes \boldsymbol{B})^\top \\
\boldsymbol{X}_{(2)} &= \boldsymbol{B}\boldsymbol{G}_{(2)}(\boldsymbol{C} \otimes \boldsymbol{A})^\top \\
\boldsymbol{X}_{(3)} &= \boldsymbol{C}\boldsymbol{G}_{(3)}(\boldsymbol{B} \otimes \boldsymbol{A})^\top
\end{aligned}
$$

Elementwise:

$$
x_{i,j,k} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{p,q,r} \, a_{i,p} \, b_{j,q} \, c_{k,r}
$$

Parameters estimation:
Higher-order orthogonal iteration (HOOI)

$$\min_{\boldsymbol{\mathcal{G}},\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \left\| \boldsymbol{\mathcal{X}} - [\![\boldsymbol{\mathcal{G}}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!] \right\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{I}_P,\ \boldsymbol{B}^\top\boldsymbol{B} = \boldsymbol{I}_Q,\ \boldsymbol{C}^\top\boldsymbol{C} = \boldsymbol{I}_R$$

which can be solved by repeating

$$\begin{aligned}
\boldsymbol{\mathcal{Y}}^A &\leftarrow \boldsymbol{\mathcal{X}} \times_2 \boldsymbol{B}^\top \times_3 \boldsymbol{C}^\top \\
\boldsymbol{\mathcal{Y}}^B &\leftarrow \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^\top \times_3 \boldsymbol{C}^\top \\
\boldsymbol{\mathcal{Y}}^C &\leftarrow \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^\top \times_2 \boldsymbol{B}^\top \\
\boldsymbol{A} &\leftarrow P \text{ leading singular vectors of } \boldsymbol{Y}^A_{(1)} \\
\boldsymbol{B} &\leftarrow Q \text{ leading singular vectors of } \boldsymbol{Y}^B_{(2)} \\
\boldsymbol{C} &\leftarrow R \text{ leading singular vectors of } \boldsymbol{Y}^C_{(3)}
\end{aligned}$$

until convergence, with $\boldsymbol{\mathcal{G}}$ finally evaluated as

$$\boldsymbol{\mathcal{G}} \leftarrow \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^\top \times_2 \boldsymbol{B}^\top \times_3 \boldsymbol{C}^\top$$

$\boldsymbol{\mathcal{G}} \in \mathbb{R}^{P\times Q\times R}$

$\boldsymbol{A} \in \mathbb{R}^{I\times P}$

$\boldsymbol{B} \in \mathbb{R}^{J\times Q}$

$\boldsymbol{C} \in \mathbb{R}^{K\times R}$

In contrast to CP, the Tucker decomposition is generally not unique

→ A, B and C constrained to be orthogonal matrices

Parameters estimation: Higher-order orthogonal iteration (HOOI)

The problem can be recast as a series of maximization subproblems

$$\begin{aligned} \boldsymbol{A} &\leftarrow \arg\max_{\boldsymbol{A}} \left\|\boldsymbol{A}^\top \boldsymbol{X}_{(1)}(\boldsymbol{C}\otimes\boldsymbol{B})\right\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{I}_P \\ \boldsymbol{B} &\leftarrow \arg\max_{\boldsymbol{B}} \left\|\boldsymbol{B}^\top \boldsymbol{X}_{(2)}(\boldsymbol{C}\otimes\boldsymbol{A})\right\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \boldsymbol{B}^\top\boldsymbol{B} = \boldsymbol{I}_Q \\ \boldsymbol{C} &\leftarrow \arg\max_{\boldsymbol{C}} \left\|\boldsymbol{C}^\top \boldsymbol{X}_{(3)}(\boldsymbol{B}\otimes\boldsymbol{A})\right\|_{\mathrm{F}}^2 \quad \text{s.t.} \quad \boldsymbol{C}^\top\boldsymbol{C} = \boldsymbol{I}_R \end{aligned}$$

$$\boldsymbol{\mathcal{G}} \in \mathbb{R}^{P\times Q\times R} \qquad \boldsymbol{A} \in \mathbb{R}^{I\times P} \qquad \boldsymbol{B} \in \mathbb{R}^{J\times Q} \qquad \boldsymbol{C} \in \mathbb{R}^{K\times R}$$

which can be solved by repeating

$$\begin{aligned} \boldsymbol{A} &\leftarrow P \text{ leading singular vectors of } \boldsymbol{X}_{(1)}(\boldsymbol{C}\otimes\boldsymbol{B}) \\ \boldsymbol{B} &\leftarrow Q \text{ leading singular vectors of } \boldsymbol{X}_{(2)}(\boldsymbol{C}\otimes\boldsymbol{A}) \\ \boldsymbol{C} &\leftarrow R \text{ leading singular vectors of } \boldsymbol{X}_{(3)}(\boldsymbol{B}\otimes\boldsymbol{A}) \end{aligned}$$

until convergence, with $\boldsymbol{\mathcal{G}}$ finally evaluated as

$$\boldsymbol{\mathcal{G}} \leftarrow \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^\top \times_2 \boldsymbol{B}^\top \times_3 \boldsymbol{C}^\top$$

In contrast to CP, the Tucker decomposition is generally not unique

→ A, B and C constrained to be orthogonal matrices

Tensor-variate regression

Python notebook:
demo_tensorRegr.ipynb

Matlab code:
demo_tensorRegr01.m

Tensor-variate linear regression

y	predicted output
$\boldsymbol{w}$	vector of weights
b	bias
ϵ	Gaussian noise

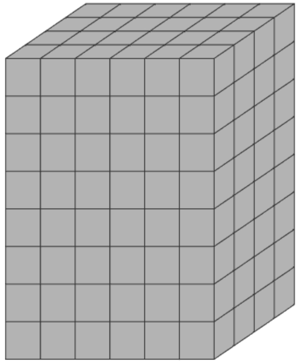
For vector-variate $\boldsymbol{x}$:
$$\begin{aligned} y &= \boldsymbol{x}^\top \boldsymbol{w} + b + \epsilon \\ &= \langle \boldsymbol{x}, \boldsymbol{w} \rangle + b + \epsilon \end{aligned}$$

For matrix-variate $\boldsymbol{X}$:
$$\begin{aligned} y &= {\boldsymbol{w}^{(1)}}^\top \boldsymbol{X} \boldsymbol{w}^{(2)} + b + \epsilon \\ &= \langle \boldsymbol{X}, \boldsymbol{w}^{(1)} \circ \boldsymbol{w}^{(2)} \rangle + b + \epsilon \end{aligned}$$

For tensor-variate $\boldsymbol{\mathcal{X}}$:
$$\begin{aligned} y &= \langle\, \boldsymbol{\mathcal{X}}, \boldsymbol{w}^{(1)} \circ \ldots \circ \boldsymbol{w}^{(M)} \,\rangle + b + \epsilon \\ &= \langle\, \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}} \,\rangle + b + \epsilon \end{aligned}$$

$\Rightarrow$ for $\boldsymbol{\mathcal{W}}$ of rank R:
$$\begin{aligned} y &= \langle\, \boldsymbol{\mathcal{X}}, \textstyle\sum_{r=1}^{R} \boldsymbol{w}_r^{(1)} \circ \ldots \circ \boldsymbol{w}_r^{(M)} \,\rangle + b + \epsilon \\ &= \langle\, \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}} \,\rangle + b + \epsilon \end{aligned}$$

Tensor-variate linear regression: Parameters estimation



$\boldsymbol{\mathcal{X}}_n$

$$y_n = \underbrace{\left\langle \boldsymbol{\mathcal{X}}_n, \sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r \right\rangle} + b$$

$$= \left\langle \boldsymbol{X}_{(1),n} \, , \, \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^\top \right\rangle$$

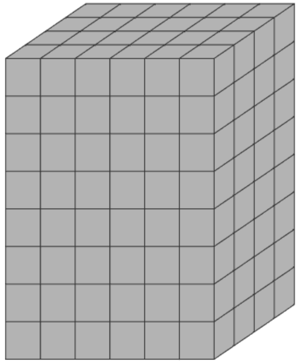
$$= \left\langle \boldsymbol{X}_{(1),n}(\boldsymbol{C} \odot \boldsymbol{B}) \, , \, \boldsymbol{A} \right\rangle$$

$$= \Big\langle \mathrm{vec}\big(\boldsymbol{X}_{(1),n}(\boldsymbol{C} \odot \boldsymbol{B})\big) \, , \, \mathrm{vec}(\boldsymbol{A}) \Big\rangle$$

$$= \underbrace{\mathrm{vec}\big(\boldsymbol{X}_{(1),n}(\boldsymbol{C} \odot \boldsymbol{B})\big)^\top}_{\phi_{1,n}} \mathrm{vec}(\boldsymbol{A})$$

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \boldsymbol{u}^\top \boldsymbol{v}$$

Tensor-variate linear regression: Parameters estimation



$\boldsymbol{\mathcal{X}}_n$

$$y_n = \underbrace{\left\langle \boldsymbol{\mathcal{X}}_n, \sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r \right\rangle}\, + b$$

$$= \underbrace{\operatorname{vec}\big(\boldsymbol{X}_{(1),n}(\boldsymbol{C} \odot \boldsymbol{B})\big)^\top}_{\phi_{1,n}} \operatorname{vec}(\boldsymbol{A})$$

$$= \underbrace{\operatorname{vec}\big(\boldsymbol{X}_{(2),n}(\boldsymbol{C} \odot \boldsymbol{A})\big)^\top}_{\phi_{2,n}} \operatorname{vec}(\boldsymbol{B})$$

$$= \underbrace{\operatorname{vec}\big(\boldsymbol{X}_{(3),n}(\boldsymbol{B} \odot \boldsymbol{A})\big)^\top}_{\phi_{3,n}} \operatorname{vec}(\boldsymbol{C})$$

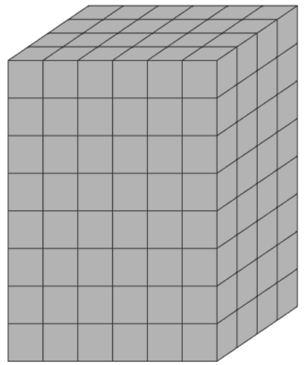
$$\boldsymbol{y} - \mathbf{1}b = \boldsymbol{\Phi}_1 \operatorname{vec}(\boldsymbol{A})$$

$$\boldsymbol{y} - \mathbf{1}b = \boldsymbol{\Phi}_2 \operatorname{vec}(\boldsymbol{B})$$

$$\boldsymbol{y} - \mathbf{1}b = \boldsymbol{\Phi}_3 \operatorname{vec}(\boldsymbol{C})$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ N \end{bmatrix} \qquad \boldsymbol{\Phi}_i = \begin{bmatrix} \phi_{i,1} \\ \phi_{i,2} \\ \vdots \\ \phi_{i,N} \end{bmatrix}$$

Tensor-variate linear regression: Parameters estimation



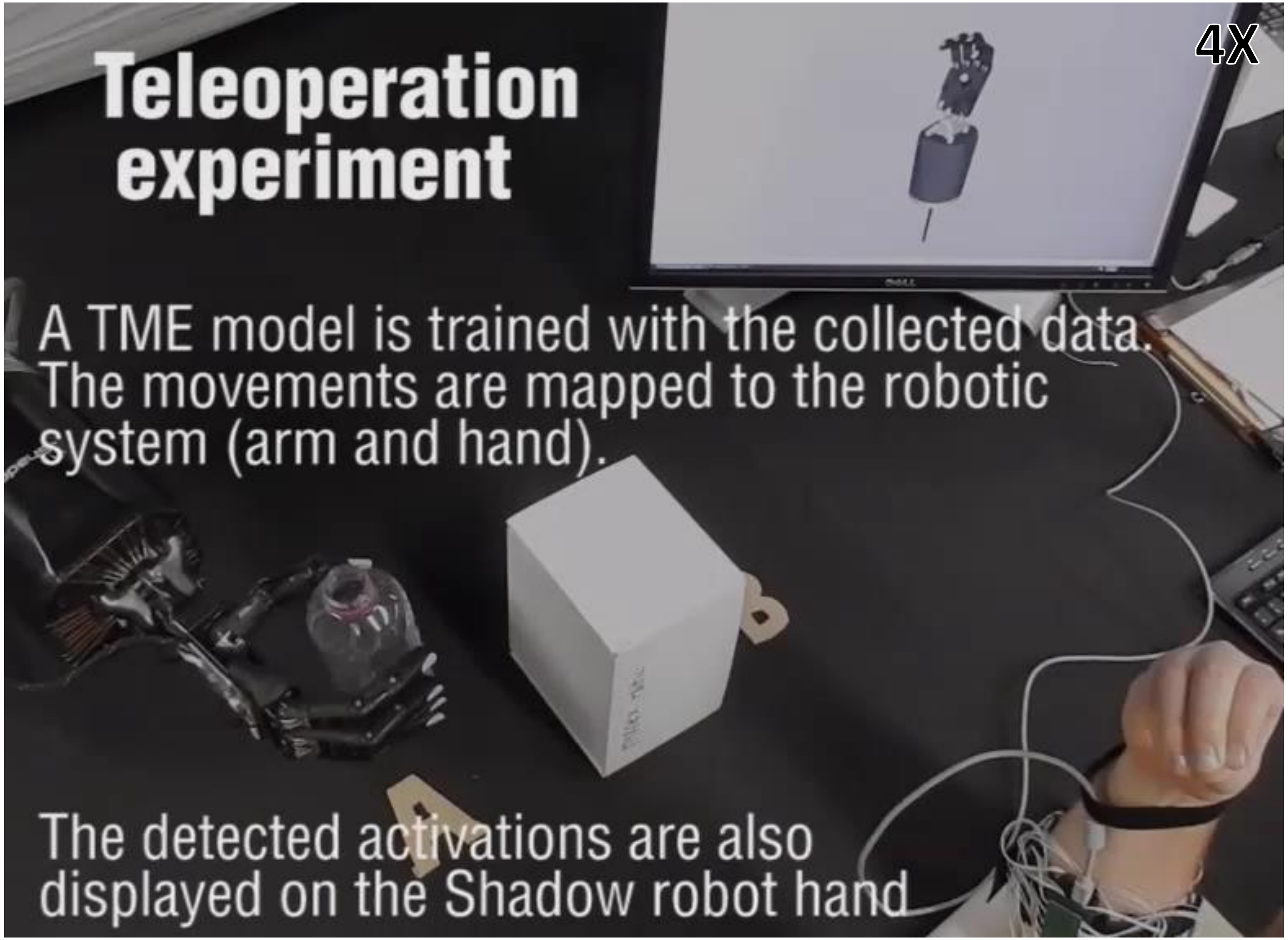
$\boldsymbol{\mathcal{X}}_n$

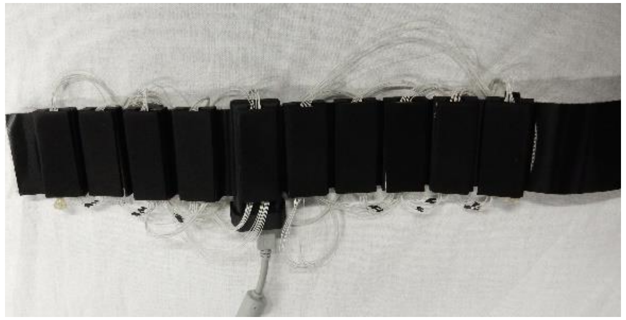
Alternating least squares (ALS) update rules:

$$\boldsymbol{y} - \mathbf{1}b = \boldsymbol{\Phi}_1 \operatorname{vec}(\boldsymbol{A})$$
$$\boldsymbol{y} - \mathbf{1}b = \boldsymbol{\Phi}_2 \operatorname{vec}(\boldsymbol{B})$$
$$\boldsymbol{y} - \mathbf{1}b = \boldsymbol{\Phi}_3 \operatorname{vec}(\boldsymbol{C})$$

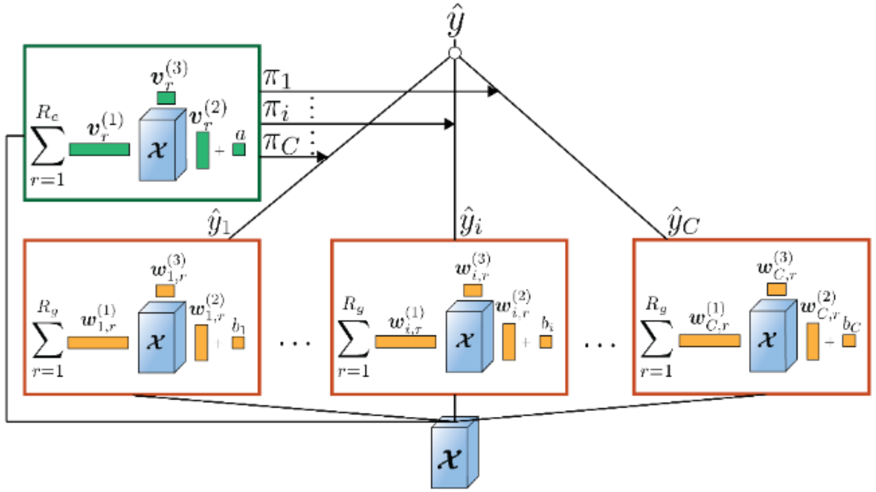
$$\begin{aligned}
\operatorname{vec}(\boldsymbol{A}) &\leftarrow \boldsymbol{\Phi}_1^{\dagger}(\boldsymbol{y} - \mathbf{1}b) \\
\operatorname{vec}(\boldsymbol{B}) &\leftarrow \boldsymbol{\Phi}_2^{\dagger}(\boldsymbol{y} - \mathbf{1}b) \\
\operatorname{vec}(\boldsymbol{C}) &\leftarrow \boldsymbol{\Phi}_3^{\dagger}(\boldsymbol{y} - \mathbf{1}b) \\
b &\leftarrow \frac{1}{N}\sum_{n=1}^{N}\Big(y_n - \big\langle\, \boldsymbol{X}_{(1),n}\,,\; \boldsymbol{A}(\boldsymbol{C}\odot\boldsymbol{B})^{\top}\,\big\rangle\Big)
\end{aligned}$$

Example: Tensor-variate mixture of experts





Tactile myography (TMG) dataset organized as sets of 8x40 matrices

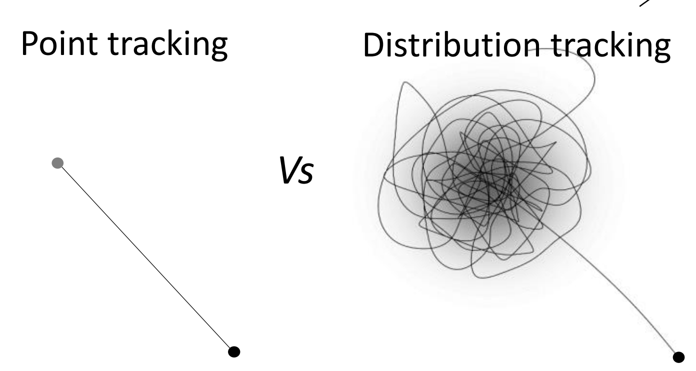


Tensor-variate mixture of experts, with tensor regression as experts and tensor logistic regression as gating functions

[Jaquier, Haschke and Calinon, Robotics and Autonomous Systems, 2021]

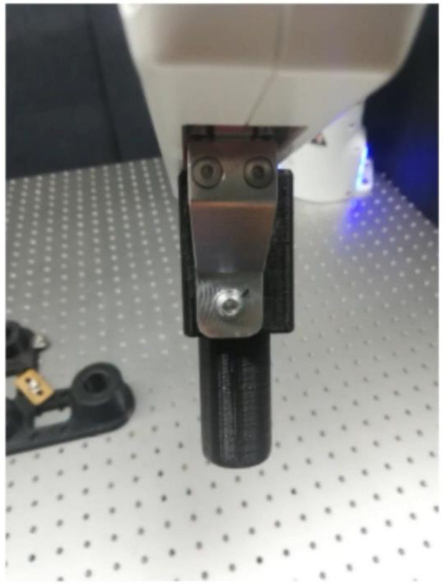
Example: Ergodic control

Ergodic control as search behavior



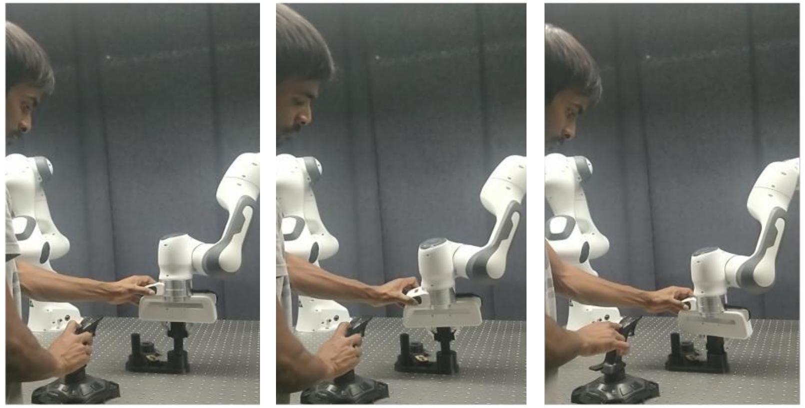
The approach relies on **Fourier basis functions** expansion
→ low-rank tensor structure

We evaluate the proposed approach using two different peg grasps:


Grasp #1


Grasp #2

Insertion task (Siemens gears benchmark)



Demonstration of insertion pose variations to provide a spatial reference distribution

[Shetty, Silvério and Calinon, IEEE Trans. On Robotics, 2021]

Recommended material

Tensor methods

Kolda T, Bader B (2009) Tensor decompositions and applications. SIAM Review 51(3):455-500

Comon P (2014) Tensors: A brief introduction. IEEE Signal Processing Magazine 31(3):44-53

Rabanser S, Shchur O, Günnemann S (2017) Introduction to tensor decompositions and their applications in machine learning. arXiv:171110781 pp 1-13

Sorber L, Van Barel M, De Lathauwer L (2015) Structured data fusion. IEEE Journal of Selected Topics in Signal Processing 9(4):586-600

Tensor methods - Softwares

http://tensorly.org (Python)

https://www.tensorlab.net (Matlab)

TensorLy
Tensor Learning in Python

Tensorlab
A MATLAB package for tensor computations.