EE613

Machine Learning for Engineers

LINEAR REGRESSION II

Sylvain Calinon

Robot Learning & Interaction Group

Idiap Research Institute

Nov. 14, 2019

Outline

Linear Regression II (Nov 14)

- Logistic regression
- Tensor-variate regression

HMM: preliminaries (Nov 14)

- Expectation-maximization (EM)
- Covariance structures in HMM

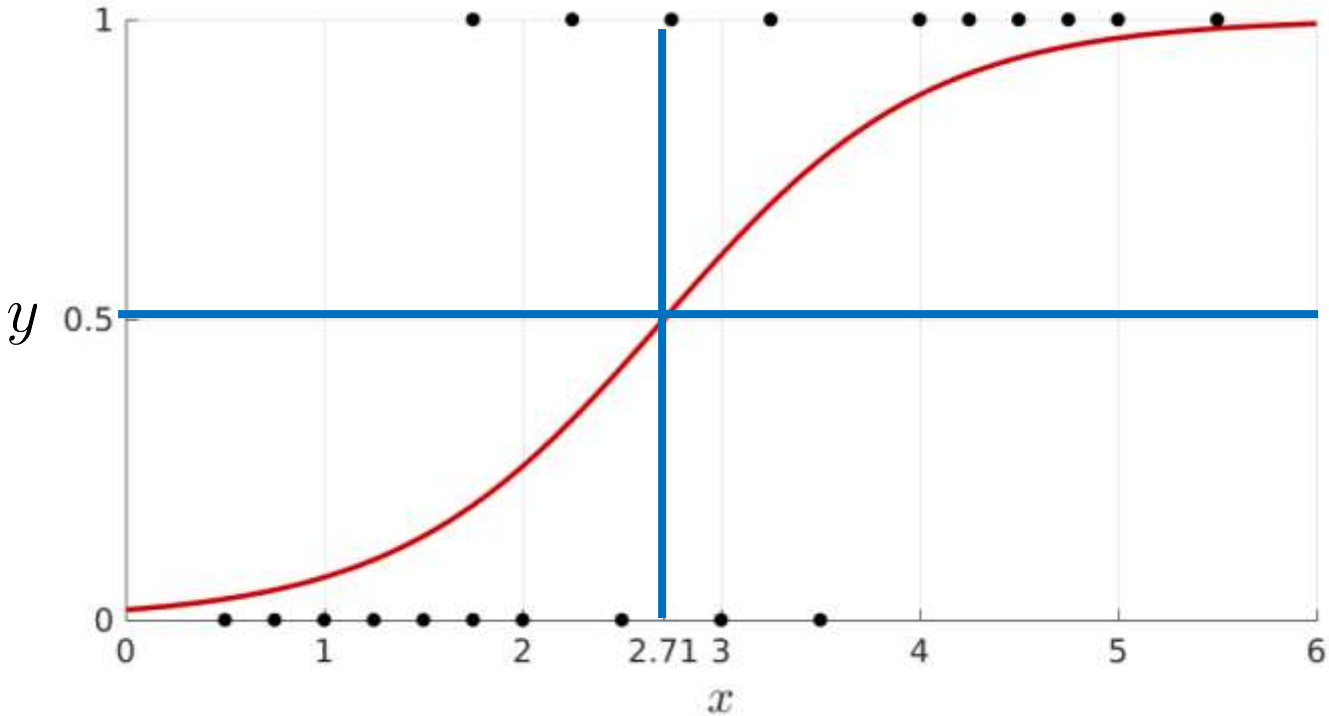
Logistic regression

Python notebook:
demo_LS_IRLS_logRegr.ipynb

Matlab code:
demo_LS_IRLS_logRegr01.m

Logistic regression

Pass/fail in function of the time spent to study at an exam:



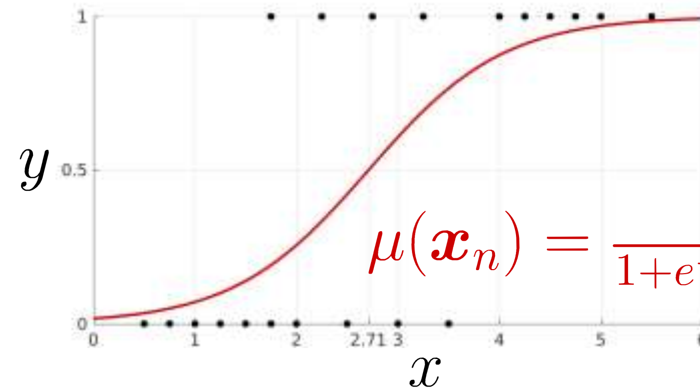
Logistic function:

→ **Classification**

$$\mu(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{a}^\intercal \boldsymbol{x}}}$$

$$\mu(x) = \frac{1}{1+e^{-(a_1+a_2 x)}}$$

Logistic regression



$$\mu(\boldsymbol{x}_n) = \frac{1}{1+e^{-\boldsymbol{a}^\intercal \boldsymbol{x}_n}}$$

Likelihood:

$$\mathcal{L} = \prod_n \mu(\boldsymbol{x}_n)^{y_n} \big(1 - \mu(\boldsymbol{x}_n)\big)^{(1-y_n)}$$

Cost function as negative log-likelihood:

$$c = -\sum_n y_n \log\big(\mu(\boldsymbol{x}_n)\big) + (1-y_n)\log\big(1-\mu(\boldsymbol{x}_n)\big)$$

$$\begin{aligned}
\frac{\partial c}{\partial \boldsymbol{a}} &= -\sum_n y_n\, \mu^{-1} \mu\,(1-\mu)\,\boldsymbol{x}_n - (1-y_n)(1-\mu)^{-1}\mu\,(1-\mu)\,\boldsymbol{x}_n \\
&= -\sum_n y_n\,(1-\mu)\,\boldsymbol{x}_n - (1-y_n)\,\mu\,\boldsymbol{x}_n \\
&= \sum_n (\mu - y_n)\,\boldsymbol{x}_n
\end{aligned}$$

$$\mu(t) = \frac{1}{1+e^{-t}}$$

$$\frac{\partial \mu}{\partial t} = \mu(1-\mu)$$

Logistic regression

$$\frac{\partial c}{\partial \boldsymbol{a}} = \sum_n (\mu - y_n)\boldsymbol{x}_n$$

It can for example be solved by a Newton-Raphson iterative optimization scheme $\boldsymbol{a} \leftarrow \boldsymbol{a} - \boldsymbol{H}^{-1}\boldsymbol{g}$,
with gradient $\boldsymbol{g} = \sum_n (\mu(\boldsymbol{x}_n) - y_n)\boldsymbol{x}_n = \boldsymbol{X}^\top(\boldsymbol{\mu} - \boldsymbol{y})$ and Hessian $\boldsymbol{H} = \boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X}$, with diagonal matrix $\boldsymbol{W} = \mathrm{diag}\big(\boldsymbol{\mu} \odot (\boldsymbol{1} - \boldsymbol{\mu})\big)$.
We then obtain

$$\begin{aligned}
\boldsymbol{a} &\leftarrow \boldsymbol{a} - \boldsymbol{H}^{-1}\boldsymbol{g} \\
&\leftarrow \boldsymbol{A} - (\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^\top (\boldsymbol{\mu} - \boldsymbol{y}) \\
&\leftarrow (\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X})^{-1} \big(\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X} \boldsymbol{a} - \boldsymbol{X}^\top (\boldsymbol{\mu} - \boldsymbol{y})\big) \\
&\leftarrow (\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^\top (\boldsymbol{W} \boldsymbol{X} \boldsymbol{a} + \boldsymbol{y} - \boldsymbol{\mu}) \\
&\leftarrow (\boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{W} \boldsymbol{z},
\end{aligned}$$

with *working response* $\boldsymbol{z} = \boldsymbol{X}\boldsymbol{a} + \boldsymbol{W}^{-1}(\boldsymbol{y} - \boldsymbol{\mu})$.

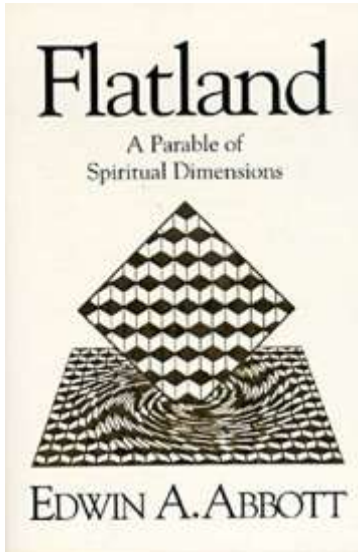
→ IRLS procedure

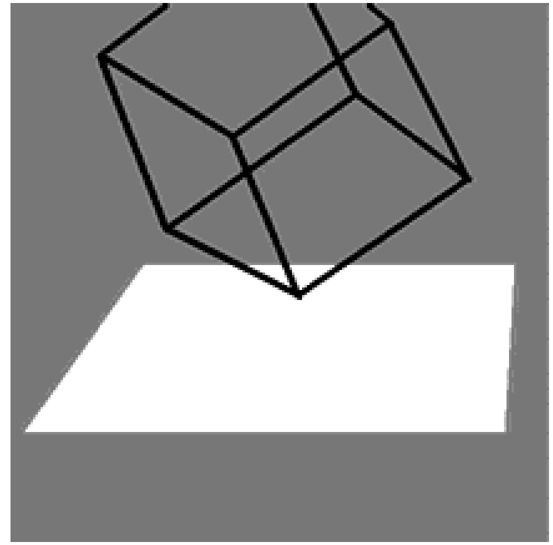
Tensor-variate regression

Python notebook:
demo_tensorRegr.ipynb

Matlab code:
demo_tensorRegr01.m

Tensor methods - Motivation

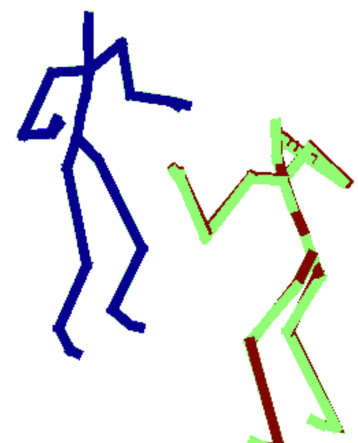




joint
agent
coordinate
sample
time step

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

$$\boldsymbol{X} \in \mathbb{R}^{10 \times 18600}$$

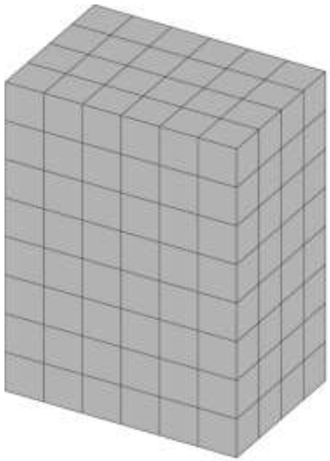


Tensor factorization keeps the structure of the original data

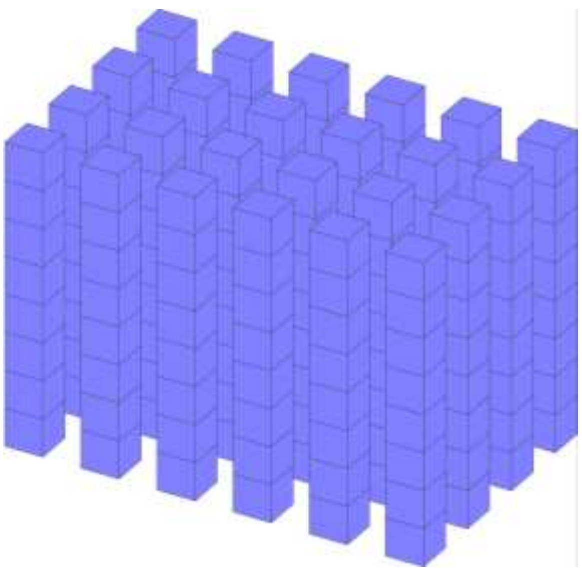
→ Multiway analysis of the data

Tensor indexing - Fibers

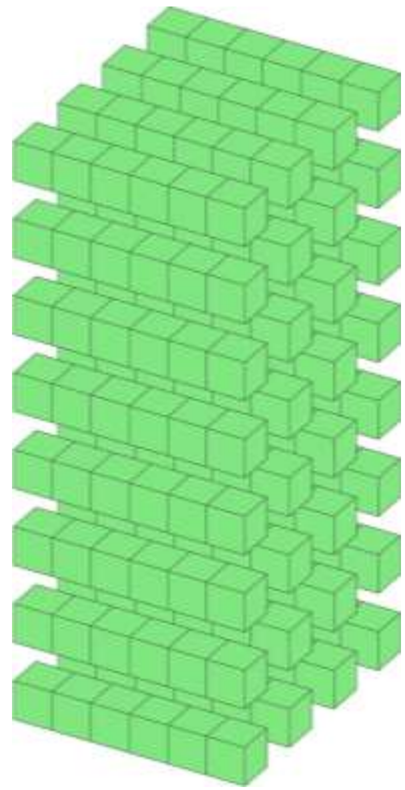
$\boldsymbol{\mathcal{X}}$	tensor
$\boldsymbol{X}$	matrix
$\boldsymbol{x}$	vector
x	scalar



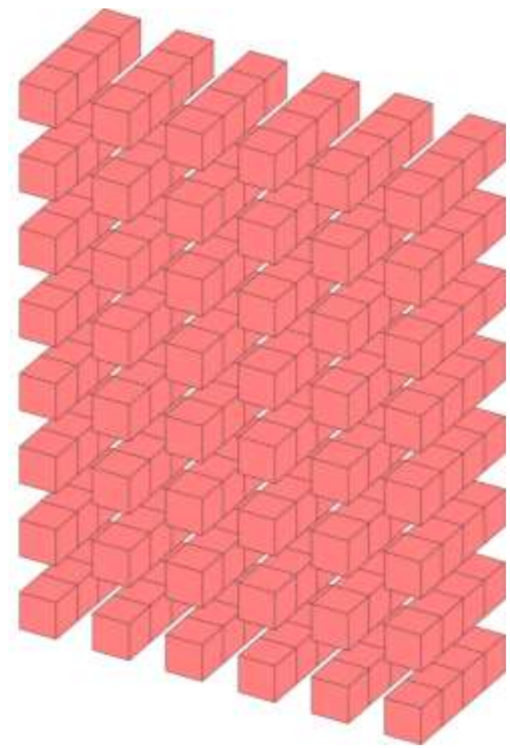
$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{8\times 6\times 4}$



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



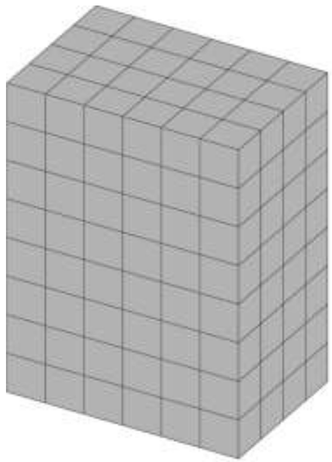
$\boldsymbol{x}_{i,:,k}$ (row)



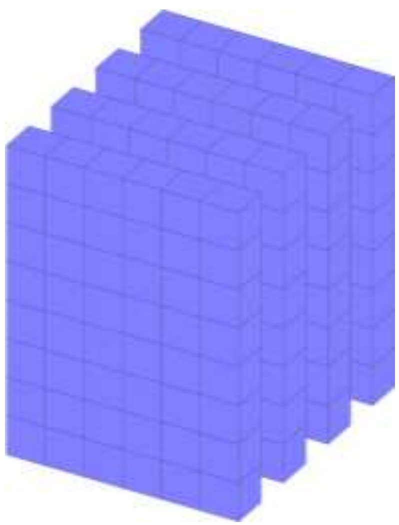
$\boldsymbol{x}_{i,j,:}$ (tube)

Tensor indexing - Slices

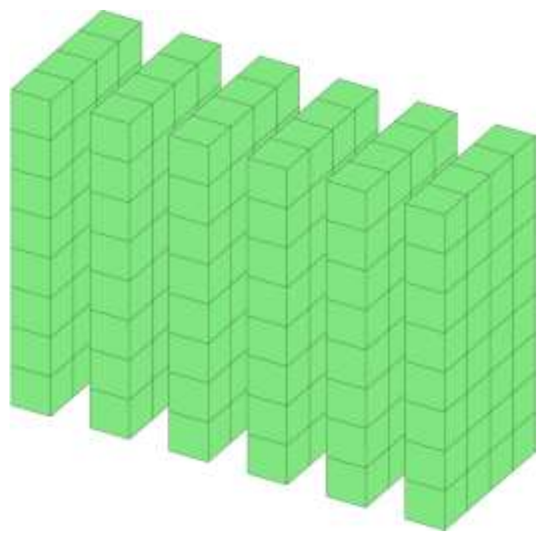
$\boldsymbol{\mathcal{X}}$ tensor
$\boldsymbol{X}$ matrix
$\boldsymbol{x}$ vector
x scalar



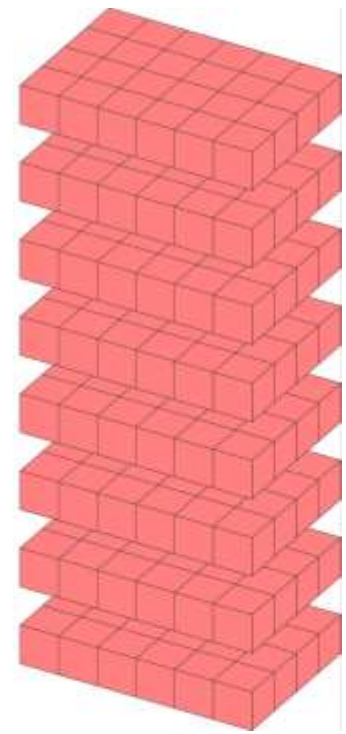
$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{8\times 6\times 4}$



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



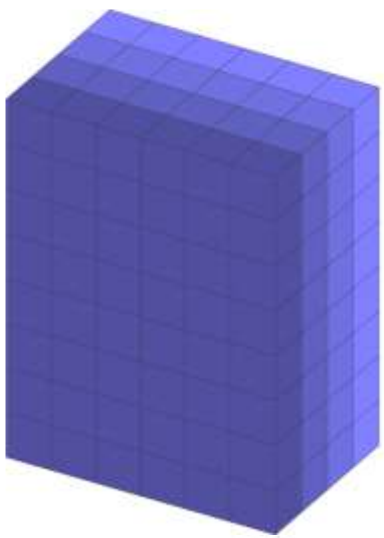
$\boldsymbol{X}_{:,j,:}$ (lateral)



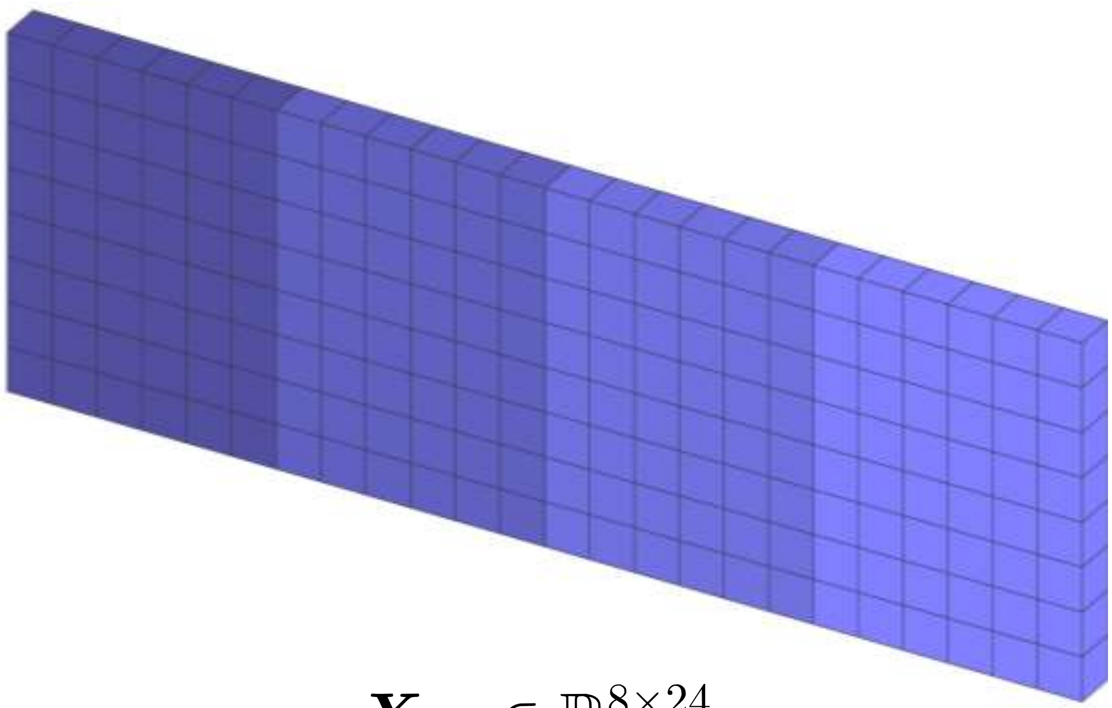
$\boldsymbol{X}_{i,:,:}$ (horizontal)

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

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

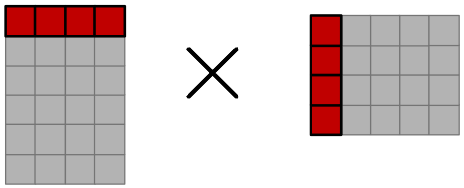
Kronecker $$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{1,1}\boldsymbol{B} & a_{1,2}\boldsymbol{B} & \cdots & a_{1,J}\boldsymbol{B} \\ a_{2,1}\boldsymbol{B} & a_{2,2}\boldsymbol{B} & \cdots & a_{2,J}\boldsymbol{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I,1}\boldsymbol{B} & a_{I,2}\boldsymbol{B}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

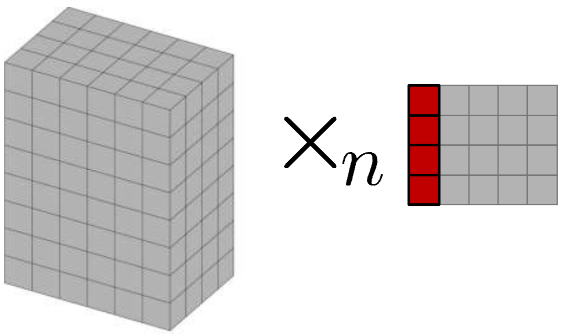
$$\boldsymbol{A} \in \mathbb{R}^{I \times J} \quad \boldsymbol{B} \in \mathbb{R}^{K \times L} \quad \boldsymbol{A} \otimes \boldsymbol{B} \in \mathbb{R}^{IK \times JL}$$

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

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

Mode-n product





$$\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$$

$$\boldsymbol{M} \in \mathbb{R}^{J \times I_n}$$

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

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

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

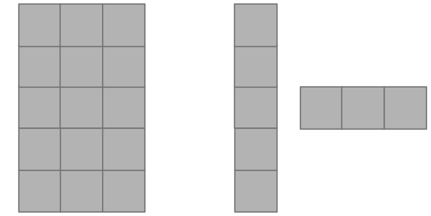
$$y_{i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} x_{i_1,\ldots,i_N}\, m_{j,i_n} \quad \text{(elementwise)}$$

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

Outer product and inner 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 $x_{i,j,k} = a_i\, b_j\, c_k$.



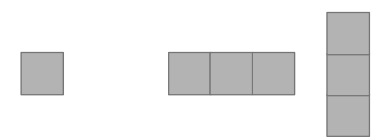
$$\boldsymbol{X} = \boldsymbol{a}\ \boldsymbol{b}^\top = \boldsymbol{a} \circ \boldsymbol{b}$$

(outer 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

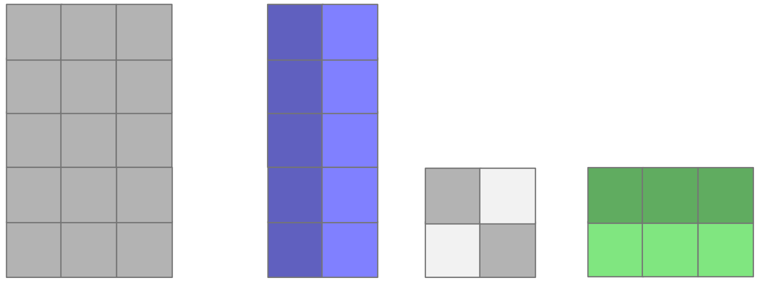
$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle = \langle \boldsymbol{A}_{(n)}, \boldsymbol{B}_{(n)} \rangle = \langle \mathrm{vec}(\boldsymbol{\mathcal{A}}), \mathrm{vec}(\boldsymbol{\mathcal{B}}) \rangle.$$



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

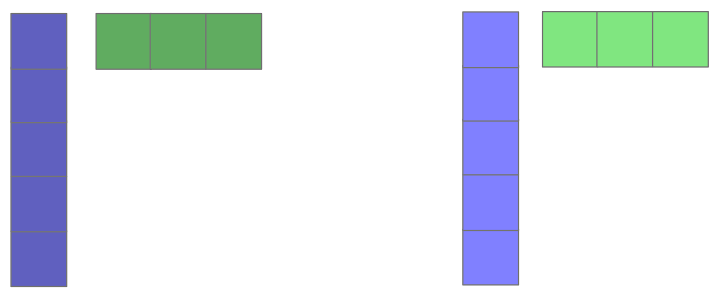
(inner product)

Singular value decomposition (SVD)

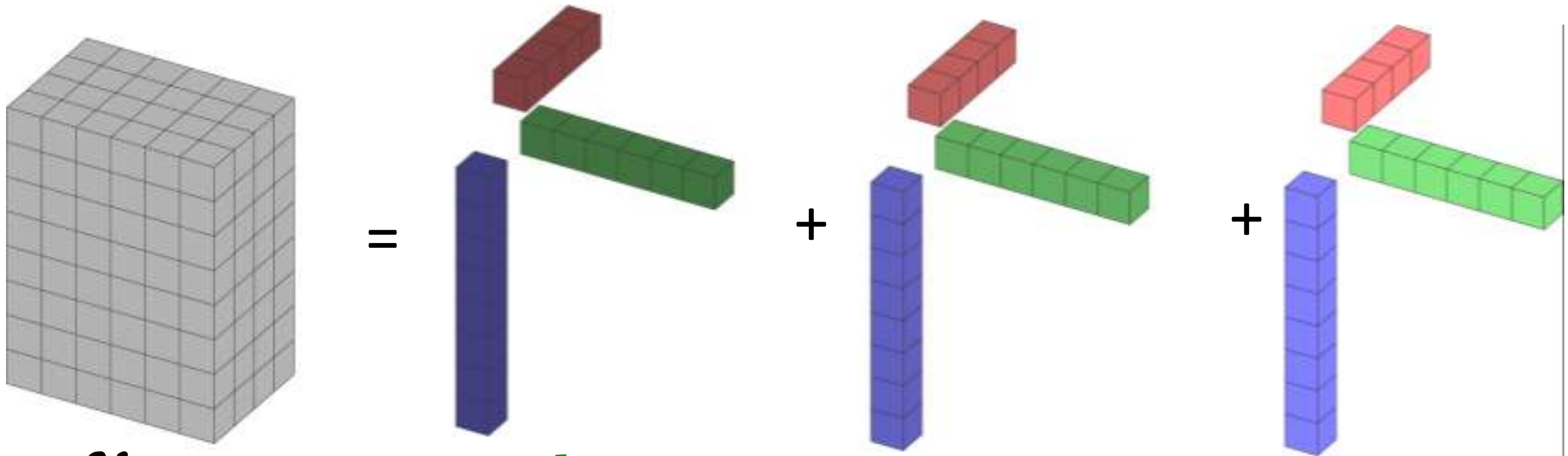


$$\boldsymbol{X} = \boldsymbol{U} \; \boldsymbol{\Sigma} \; \boldsymbol{V}^\top$$

$$= \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^\top$$

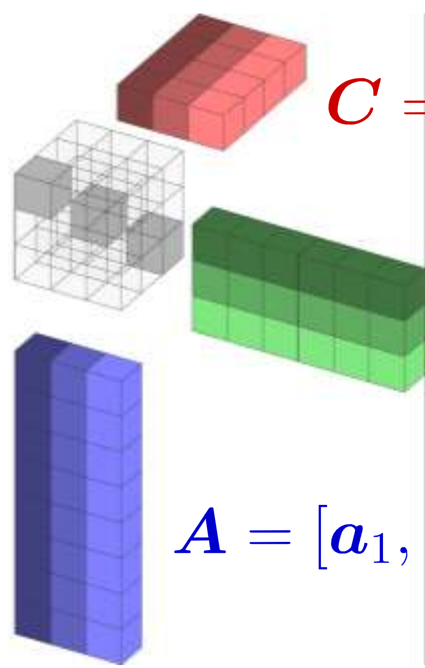


CP decomposition



$$\mathcal{X} = \mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \mathbf{a}_3 \circ \mathbf{b}_3 \circ \mathbf{c}_3$$

=



$$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3]$$

$$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$$

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$$

CP decomposition

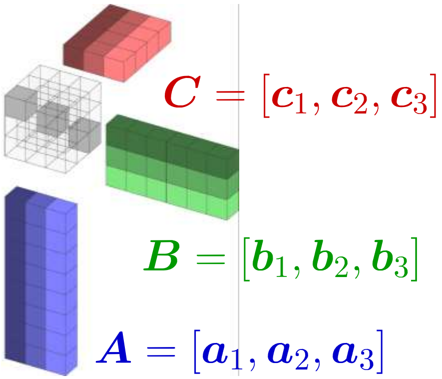
$$\boldsymbol{A} \in \mathbb{R}^{I \times R}$$
$$\boldsymbol{B} \in \mathbb{R}^{J \times R}$$
$$\boldsymbol{C} \in \mathbb{R}^{K \times R}$$

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

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



$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_R]$ 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

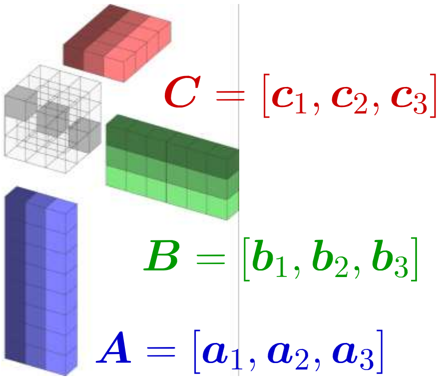
The CP decomposition can be solved by alternating least squares (ALS), by repeating

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

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

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

until convergence, yielding the update rules

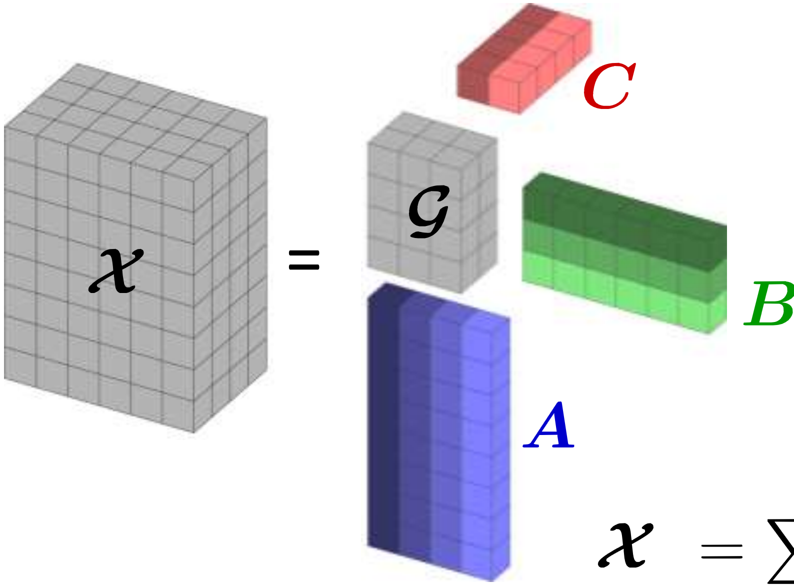


$$\boldsymbol{A} \leftarrow \boldsymbol{X}_{(1)} \left((\boldsymbol{C} \odot \boldsymbol{B})^\top \right)^\dagger$$

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

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

Tucker decomposition



Core tensor

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

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

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

Parameters estimation:
Higher-order orthogonal iteration (HOOI)

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

which can be solved by repeating

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

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

→ A, B and C constrained to be orthogonal matrices

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

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

Parameters estimation: Higher-order orthogonal iteration (HOOI)

The problem can be recast as a series of maximization subproblems

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

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

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

Tensor-variate linear regression

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

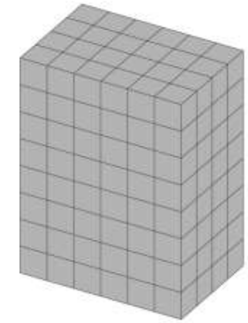
For vector-variate $\boldsymbol{x}$:
$$\begin{aligned} y &= \boldsymbol{x}^\top \boldsymbol{w} + b + \epsilon \\ &= \langle \boldsymbol{x}, \boldsymbol{w} \rangle + b + \epsilon \end{aligned}$$

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



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

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

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

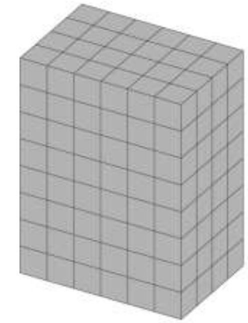
$$= \left\langle\, \boldsymbol{X}_{(1),n}(\boldsymbol{C} \odot \boldsymbol{B})\ ,\ \boldsymbol{A} \right\rangle$$

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

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

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

Tensor-variate linear regression: Parameters estimation



$\mathcal{X}_n$

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

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

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

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

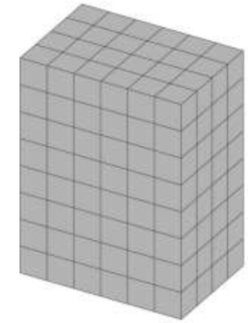
$$\boldsymbol{y} - \mathbf{1}b = \boldsymbol{\Phi}_1 \operatorname{vec}(\boldsymbol{A})$$

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

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

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_N \end{bmatrix} \qquad \boldsymbol{\Phi}_i = \begin{bmatrix} \boldsymbol{\Phi}_{i,1} \\ \boldsymbol{\Phi}_{i,2} \\ \vdots \\ \boldsymbol{\Phi}_{i,N} \end{bmatrix}$$

Tensor-variate linear regression: Parameters estimation



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

ALS update rules:

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

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

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

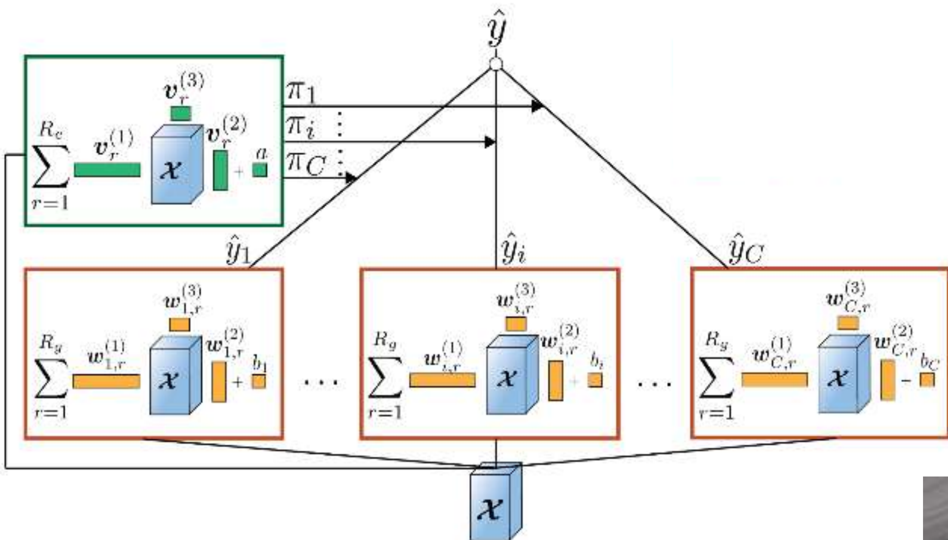
$$\operatorname{vec}(\boldsymbol{A}) \leftarrow \boldsymbol{\Phi}_1^{\dagger}(\boldsymbol{y} - \mathbf{1}b)$$

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

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

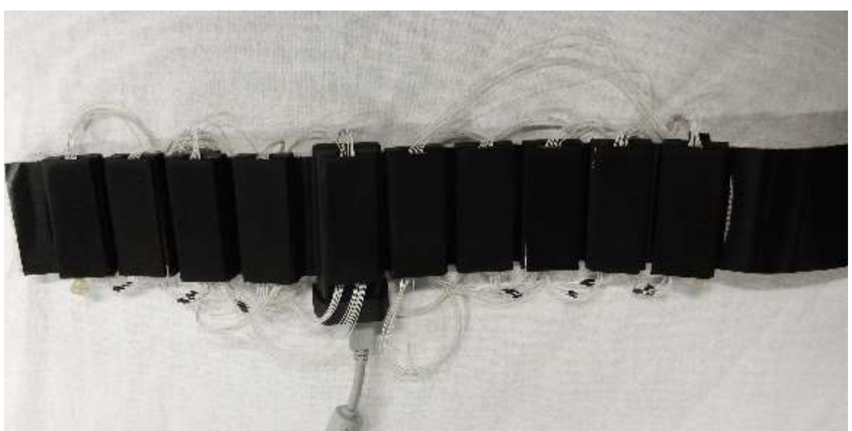
$$b \leftarrow \frac{1}{N}\sum_{n=1}^{N}\Big(y_n - \big\langle\, \boldsymbol{X}_{(1),n}\,,\ \boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{B})^{\top}\,\big\rangle\Big)$$

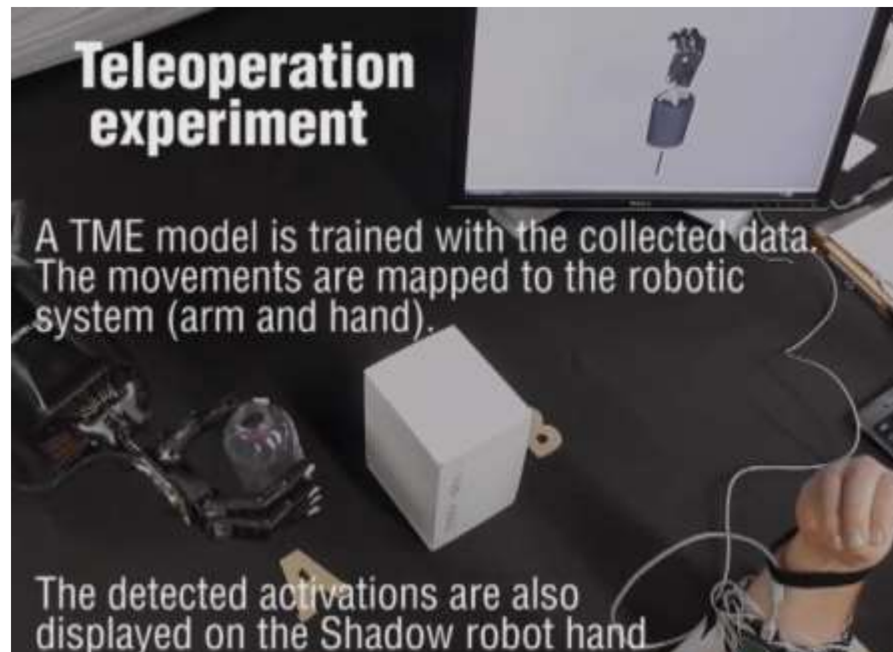
Application: Tensor-variate mixture of experts



logistic regression (gates)

ridge regression (experts)





[Jaquier, Haschke and Calinon (2019), arXiv:1902.11104]

References

Logistic regression

Walker, SH, Duncan, DB (1967) Estimation of the probability of an event as a function of several independent variables. Biometrika 54 (1/2): 167–178.

Tensor-variate regression

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)