Intro to Tensor

Outline

What exactly can tensor do?

What is tensor

Notations and Preliminaries, e.g. multiplication

Tensor decomposition, e.g. CP, Tucker

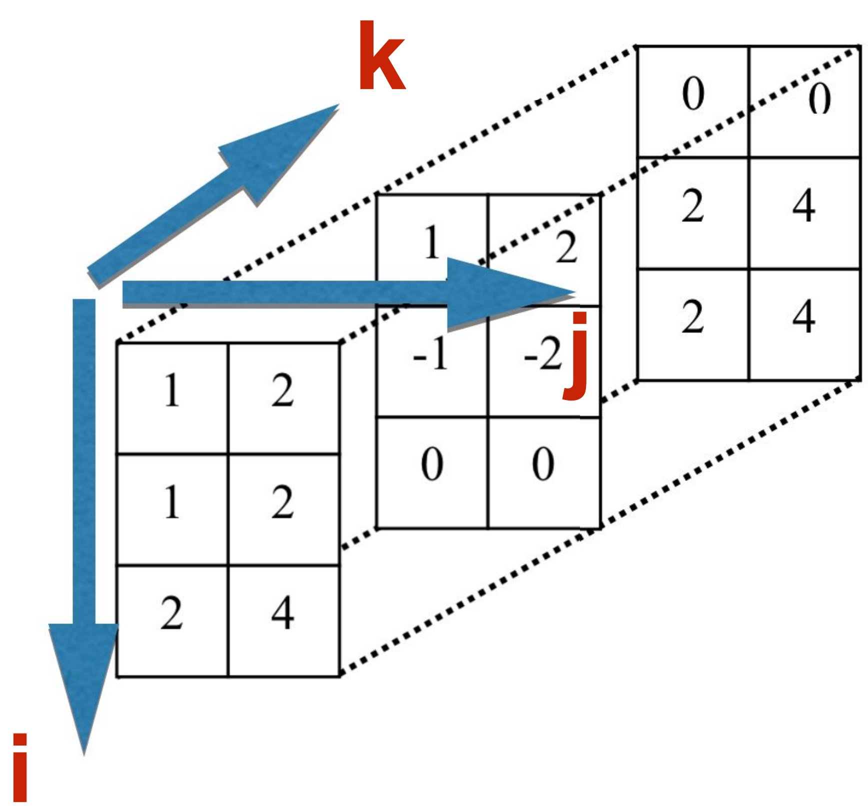
Two applications of tensor decomposition

What is tensor

Tensor is a multidimensional array.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{j1} & a_{j2} & \cdots & a_{jk} \end{bmatrix}$$



Matrix vs Tensor

Why do we need tensor / What can tensor do?

Matrix models relationship between **two** things; sometimes our objects are **more than two** things and they are **entangled** together.

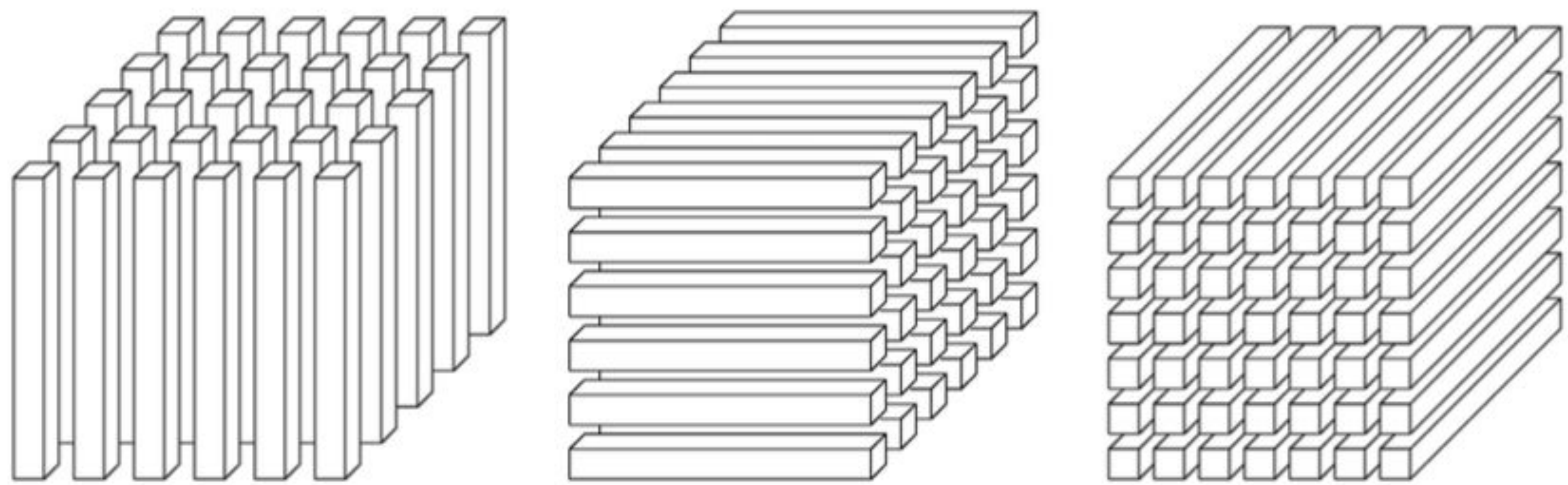
Notations

The **order** of tensor is the number of **dimensions/ways/mo**

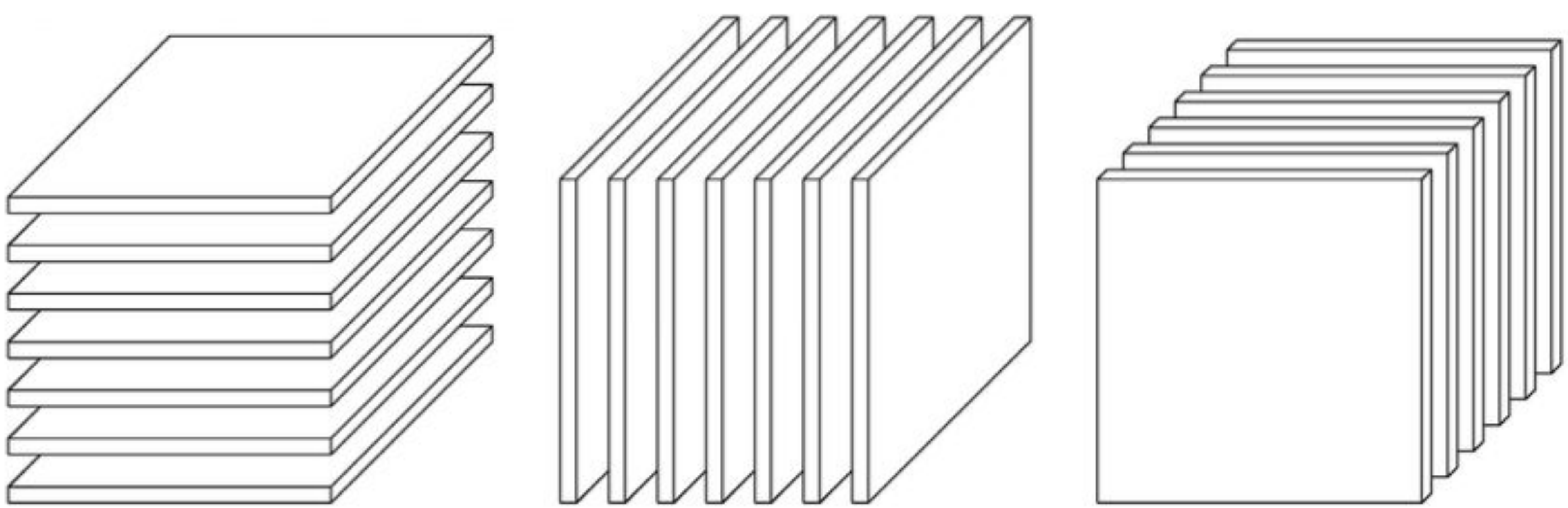
The ith entry of a vector $\mathbf{a}$ is denoted by a_i, element (i, j) of a matrix $\mathbf{A}$ is denoted by a_{ij}, and element (i, j, k) of a third-order tensor $\mathcal{X}$ is denoted by x_{ijk}.

$$\| \mathcal{X} \| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \cdots i_N}^2}. \qquad \langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \cdots i_N} y_{i_1 i_2 \cdots i_N}.$$

Fibers and slices



(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$ (b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$ (c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$



(a) Horizontal slices: $\mathbf{X}_{i::}$ (b) Lateral slices: $\mathbf{X}_{:j:}$ (c) Frontal slices: $\mathbf{X}_{::k}$ (or $\mathbf{X}_k$)

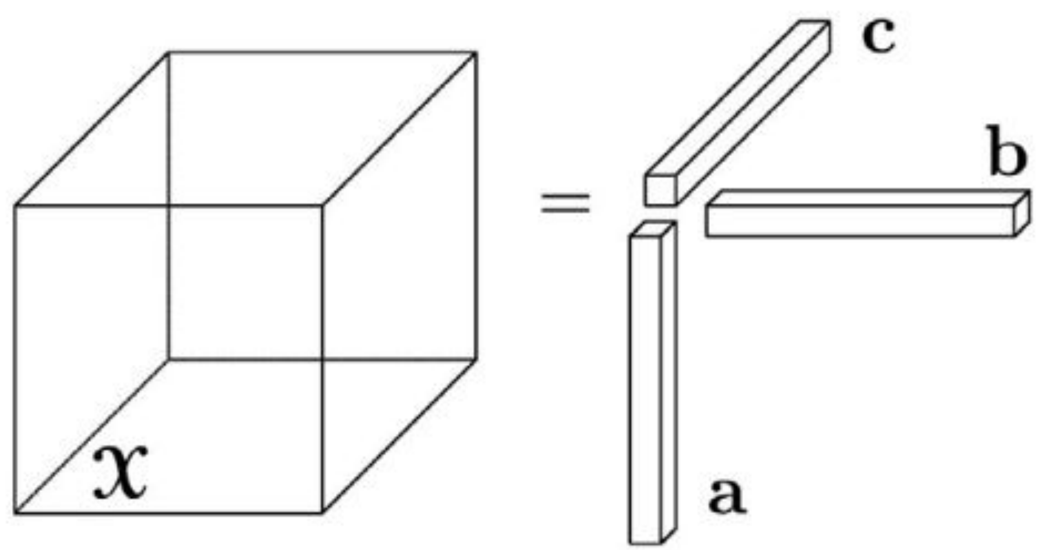
Preliminaries

1. Outer product of N vectors

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}$$

$$x_{i_1 i_2 \cdots i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N} \quad \text{for all } 1 \leq i_n \leq I_n$$

2. Such $\mathcal{X}$ is called rank one tensor



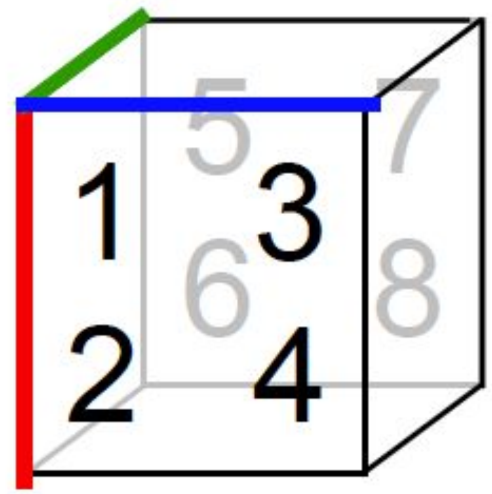
Preliminaries

1.Matricization: transfer tensor into matrix

Mode-n matricization of tensor $\mathcal{X}$ is donated by $\mathbf{X}_{(n)}$ and arranges the mode-n fibers to be columns of the resulting matrix.

Tensor element $(i_1, i_2, \ldots, i_N)$ maps to matrix element (i_n, j), where

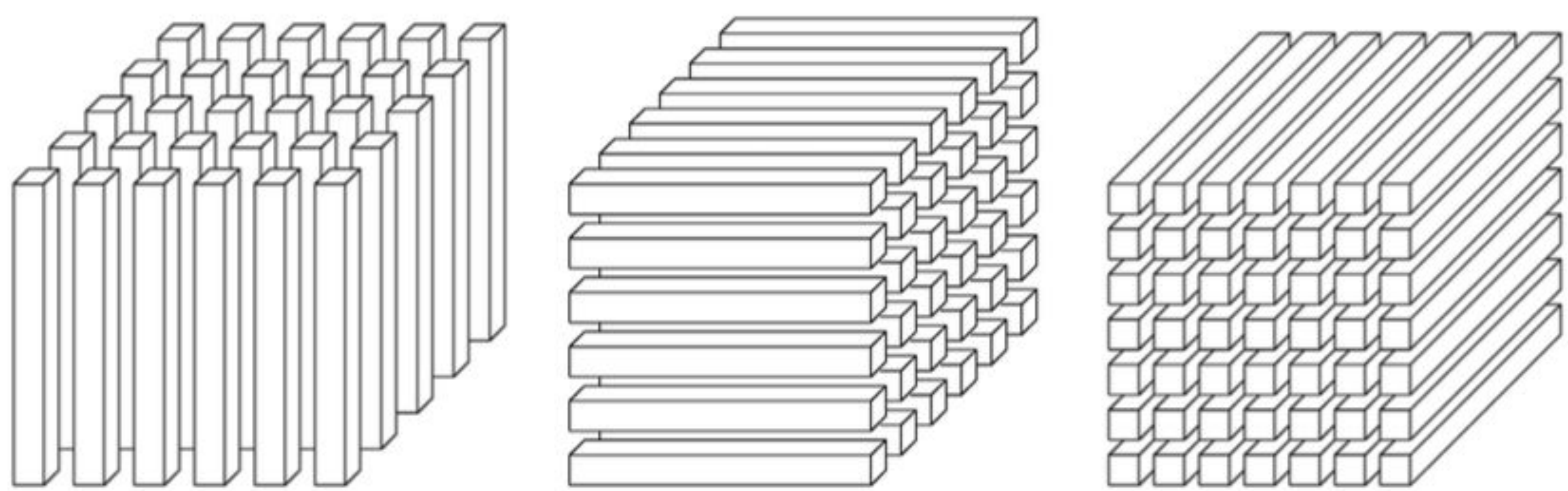
$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k \quad \text{with} \quad J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m.$$

$\mathcal{X} =$ 

$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}$$

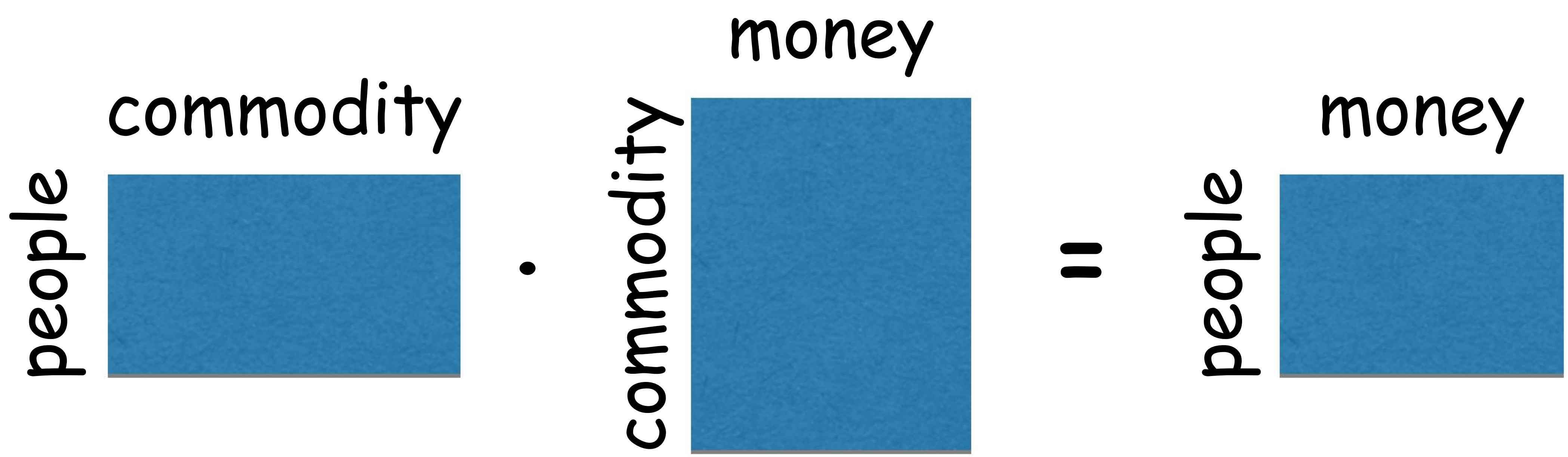
$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$



(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$ (b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$ (c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$

Tensor Multiplication

Matrix



Tensor multiplication is very similar to that of matrix

Tensor Mode-n Multiplication

The *n-mode (matrix) product* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n \mathbf{U}$ and is of size $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$. Elementwise, we have

$$(\mathcal{X} \times_n \mathbf{U})_{i_1 \cdots i_{n-1} j \, i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} \, u_{j i_n}.$$

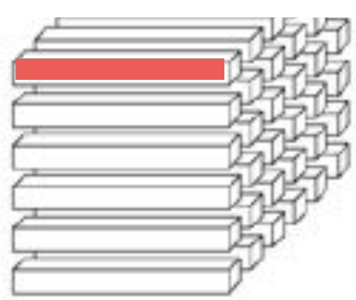
$$\mathcal{Y} = \mathcal{X} \times_2 \mathbf{B} \in \mathbb{R}^{I \times M \times K}$$

$$\mathbf{Y} = \mathcal{X} \,\bar{\times}_1\, \mathbf{a} \in \mathbb{R}^{J \times K}$$

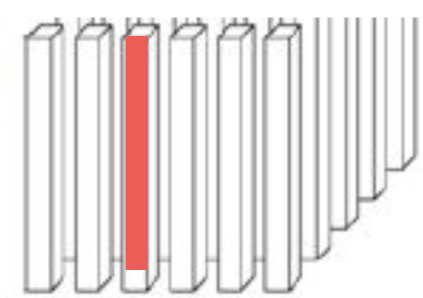
The *n-mode (vector) product* of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a vector $\mathbf{v} \in \mathbb{R}^{I_n}$ is denoted by $\mathcal{X} \,\bar{\times}_n\, \mathbf{v}$. The result is of order $N-1$, i.e., the size is $I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N$. Elementwise,

$$(\mathcal{X} \,\bar{\times}_n\, \mathbf{v})_{i_1 \cdots i_{n-1} i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} \, v_{i_n}.$$

Multiply each row (mode-2) fiber by **B**



product of $\boldsymbol{a}$ and each column (mode-1) fiber



$$\mathbf{X}_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}$$

$$\mathbf{U} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

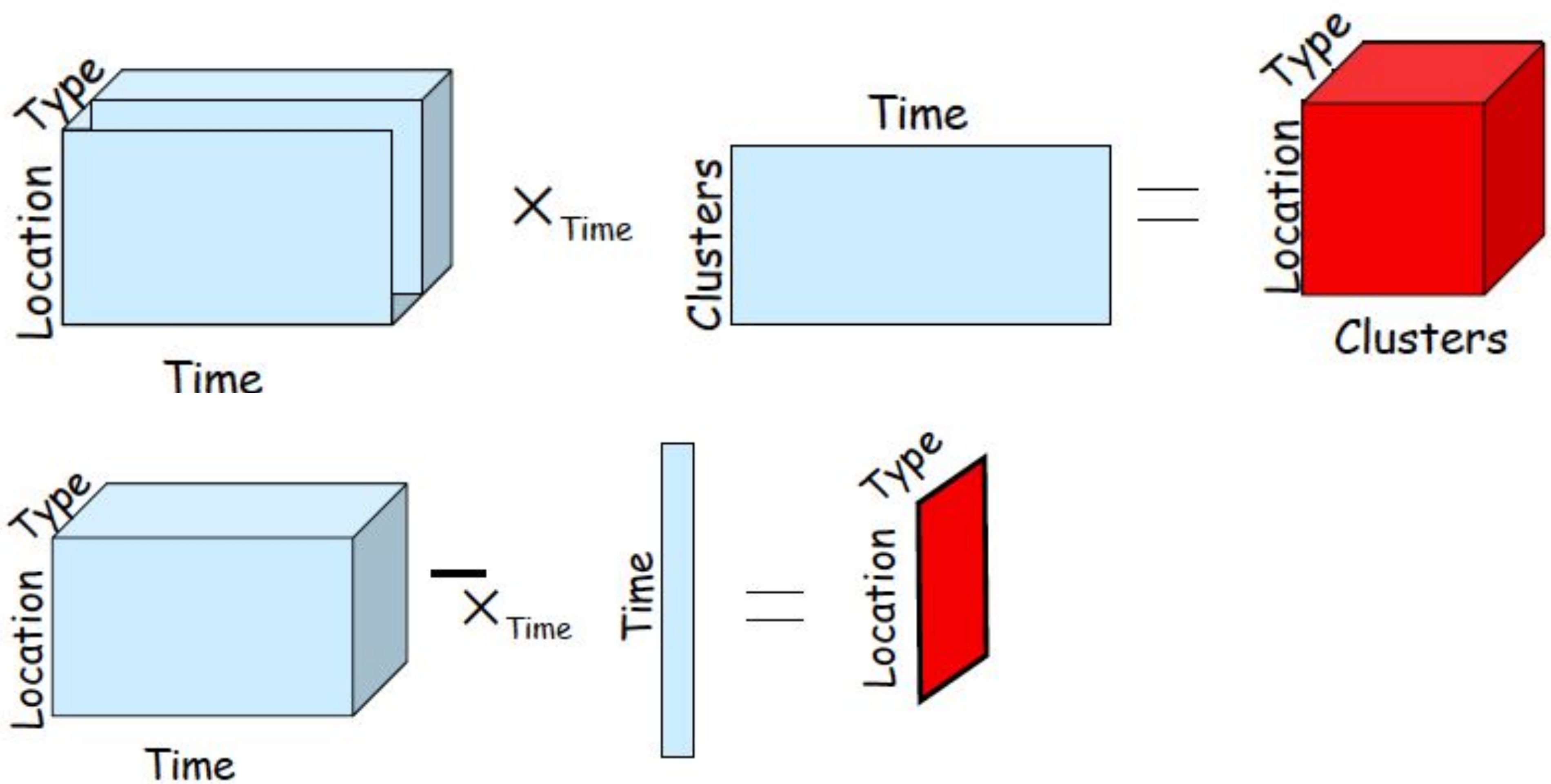
$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_1 \mathbf{U} \in \mathbb{R}^{2 \times 4 \times 2}$$

$$\mathbf{Y}_1 = \begin{bmatrix} 22 & 49 & 76 & 103 \\ 28 & 64 & 100 & 136 \end{bmatrix}, \quad \mathbf{Y}_2 = \begin{bmatrix} 130 & 157 & 184 & 211 \\ 172 & 208 & 244 & 280 \end{bmatrix}$$

Tensor Mode-n Multiplication

$$\mathcal{Y} = \mathcal{X} \times_n \mathbf{U} \quad \Leftrightarrow \quad \mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}$$

$$\mathcal{X} \times_m \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n \mathbf{B} \times_m \mathbf{A} \quad (m \neq n)$$



Tensor Decomposition

CP

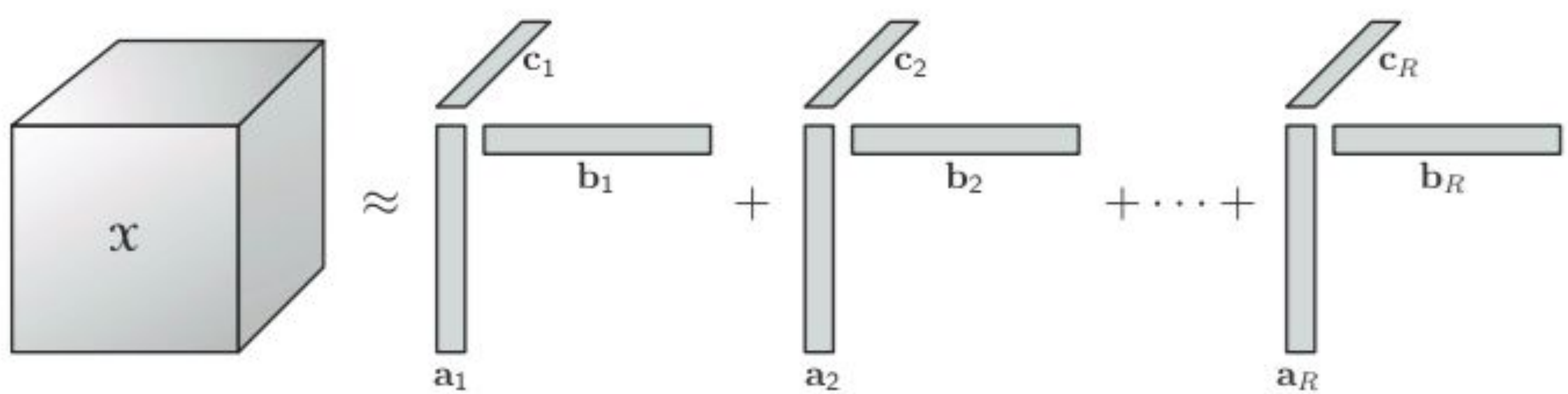
Tucker

CP Decomposition

CP decomposition: express a tensor as the sum of finite nur

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

$$x_{ijk} \approx \sum_{r=1}^{R} a_{ir}\, b_{jr}\, c_{kr} \text{ for } i = 1, \ldots, I,\ j = 1, \ldots, J,\ k = 1, \ldots, K$$



Tensor Rank

rank($\mathcal{X}$): the smallest number of rank-one tensors that generate $\mathcal{X}$

$$\underset{R}{\text{minimize}} \quad \mathcal{X} = \sum_{r=1}^{\mathcal{R}} \mathbf{a_r} \circ \mathbf{b_r} \circ \mathbf{c_r}$$

Rank decomposition: exact CP decomposition with R = rank(X) components is called the rank decomposition

Compute CP

Rank decomposition is NP hard but unique.

Assuming the number of components is fixed, there are many algorithms to compute a CP decomposition, such as ALS.

Let $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ be a third-order tensor. The goal is to compute a CP decomposition with R components that best approximates $\mathcal{X}$, i.e., to find

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\| \quad \text{with} \quad \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [\![\boldsymbol{\lambda} \, ; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]. \tag{3.7}$$

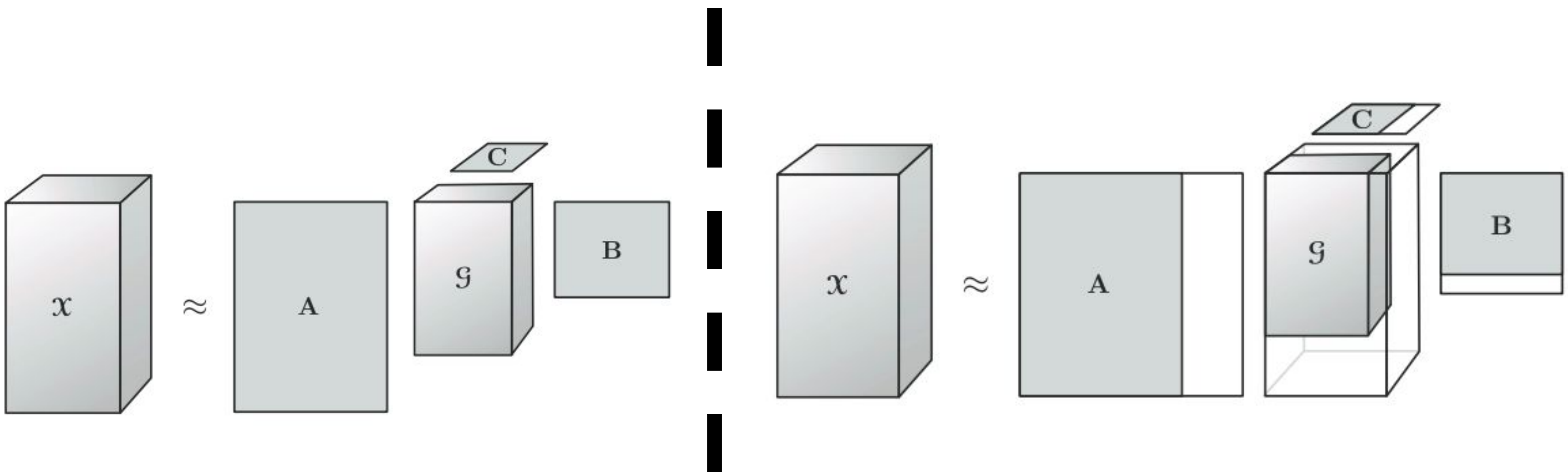
CP Computing — ALS

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\| \quad \text{with} \quad \hat{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [\![\boldsymbol{\lambda} \,; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!].$$

- The ALS approach fixes B and C to solve for A, then fixes A and C to solve for B, then fixes A and B to solve for C, and continues to repeat the entire procedure until some convergence criterion is satisfied.

procedure CP-ALS($\mathcal{X}$,R)
    initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$
    repeat
        for $n = 1, \ldots, N$ **do**
            $\mathbf{V} \leftarrow \mathbf{A}^{(1)\mathsf{T}}\mathbf{A}^{(1)} * \cdots * \mathbf{A}^{(n-1)\mathsf{T}}\mathbf{A}^{(n-1)} * \mathbf{A}^{(n+1)\mathsf{T}}\mathbf{A}^{(n+1)} * \cdots * \mathbf{A}^{(N)\mathsf{T}}\mathbf{A}^{(N)}$
            $\mathbf{A}^{(n)} \leftarrow \mathbf{X}_{(n)}(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)})\mathbf{V}^{\dagger}$
            normalize columns of $\mathbf{A}^{(n)}$ (storing norms as $\boldsymbol{\lambda}$)
        end for
    until fit ceases to improve or maximum iterations exhausted
    return $\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

Tucker Decomposition



A, **B** and **C** are columnwise orthonormal (in most cases).

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

$$x_{ijk} \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr}\, a_{ip}\, b_{jq}\, c_{kr} \quad \text{for} \quad i = 1, \ldots, I, \;\; j = 1, \ldots, J, \;\; k = 1, \ldots, K.$$

Tucker Computing

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

HOSVD method: The basic idea is to find those components that best capture the variation in mode n, independent of the other modes.

procedure $\texttt{HOSVD}(\mathcal{X}, R_1, R_2, \ldots, R_N)$
    for $n = 1, \ldots, N$ **do**
        $\mathbf{A}^{(n)} \leftarrow R_n$ leading left singular vectors of $\mathbf{X}_{(n)}$
    end for
    $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
    return $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

Tucker Computing

$$\begin{aligned} \min_{\mathcal{G}, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}} \quad & \left\| \mathcal{X} - [\![\mathcal{G} \, ; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!] \right\| \\ \text{subject to} \quad & \mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}, \\ & \mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n} \text{ and columnwise orthogonal for } n = 1, \ldots, N \end{aligned}$$

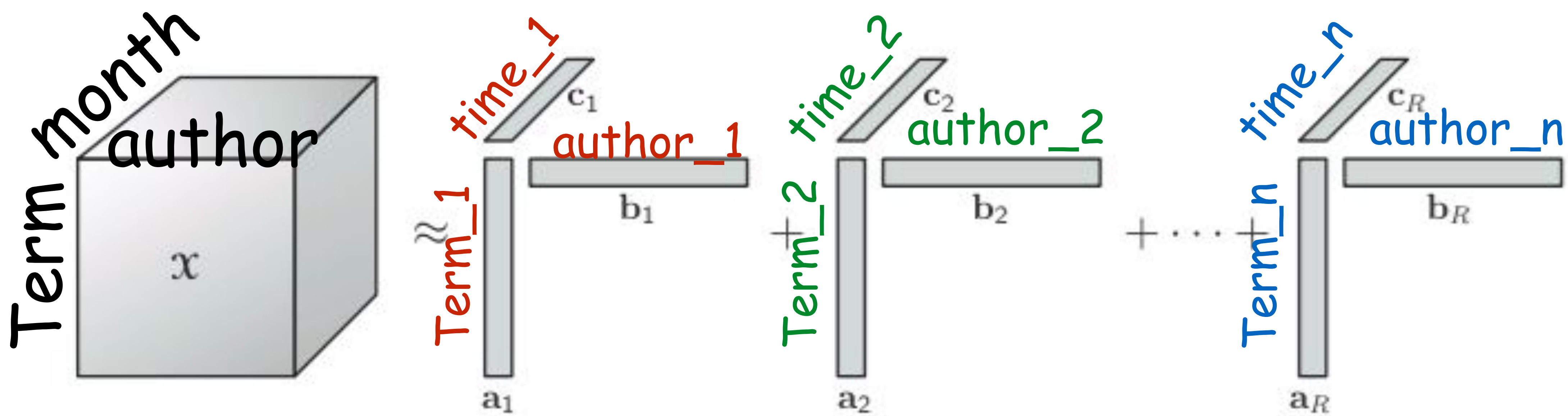
---

procedure HOOI($\mathcal{X}, R_1, R_2, \ldots, R_N$)
  initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$ using HOSVD
  repeat
    for $n = 1, \ldots, N$ **do**
      $\mathcal{Y} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \cdots \times_{n-1} \mathbf{A}^{(n-1)\mathsf{T}} \times_{n+1} \mathbf{A}^{(n+1)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
      $\mathbf{A}^{(n)} \leftarrow R_n$ leading left singular vectors of $\mathbf{Y}_{(n)}$
    end for
  until fit ceases to improve or maximum iterations exhausted
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\mathsf{T}} \times_2 \mathbf{A}^{(2)\mathsf{T}} \cdots \times_N \mathbf{A}^{(N)\mathsf{T}}$
  return $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
end procedure

One CP Example

- Extract and detect meaningful discussions from Enron email
- term-author-month array $\mathcal{X}$

$$\mathcal{X} \approx \sum_{l=1}^{r} \lambda_l (A_l \circ B_l \circ C_l)$$



Brett W. Bader et. al, Discussion Tracking in Enron Email Using PARAFAC

One Tucker Example

- Suppose we have a dataset. Each picture contains: people with different poses and expressions under different illuminations.
- Construct people-view-illumimation-expression-pixels tensor
- D is 28 × 5 × 3 × 3 × 7943

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{\text{people}} \times_2 \mathbf{U}_{\text{views}} \times_3 \mathbf{U}_{\text{illums}} \times_4 \mathbf{U}_{\text{expres}} \times_5 \mathbf{U}_{\text{pixels}}$$

where the $28 \times 5 \times 3 \times 3 \times 7943$ core tensor $\mathcal{Z}$ governs the interaction between the factors represented in the 5 mode matrices: The 28×28 mode matrix $\mathbf{U}_{\text{people}}$ spans the space of people parameters, the 5×5 mode matrix $\mathbf{U}_{\text{views}}$ spans the space of viewpoint parameters, the 3×3 mode matrix $\mathbf{U}_{\text{illums}}$ spans the space of illumination parameters and the 3×3 mode matrix $\mathbf{U}_{\text{expres}}$ spans the space of expression parameters. The 7943×7943 mode matrix $\mathbf{U}_{\text{pixels}}$ orthonormally spans the space of images.

M. Alex O. Vasilescu and Demetri Terzopoulos, Multilinear Analysis of Image Ensembles: TensorFaces (ECCV 02)

Supplemental Materials

M. Alex O. Vasilescu (http://alumni.media.mit.edu/~maov/research_index.html)