Topic IV: Tensors

Discrete Topics in Data Mining
Universität des Saarlandes, Saarbrücken
Winter Semester 2012/13

Topic IV: Tensors

1. What is a … tensor?
2. Basic Operations
3. Tensor Decompositions and Rank
    3.1. CP Decomposition
    3.2. Tensor Rank
    3.3. Tucker Decomposition

Kolda & Bader 2009

I admire the elegance of your method of computation; it must be nice to ride through these fields upon the horse of true mathematics while the like of us have to make our way laboriously on foot.

Albert Einstein
in a letter to Tullio Levi-Civita

What is a … tensor?

- A **tensor** is a multi-way extension of a matrix
  - A multi-dimensional array
  - A multi-linear map
- In particular, the following are all tensors:

What is a … tensor?

- A **tensor** is a multi-way extension of a matrix
  - A multi-dimensional array
  - A multi-linear map
- In particular, the following are all tensors:
  - Scalars

$$13$$

What is a … tensor?

- A **tensor** is a multi-way extension of a matrix
  - A multi-dimensional array
  - A multi-linear map
- In particular, the following are all tensors:
  - Scalars
  - Vectors

$$(13, 42, 2011)$$

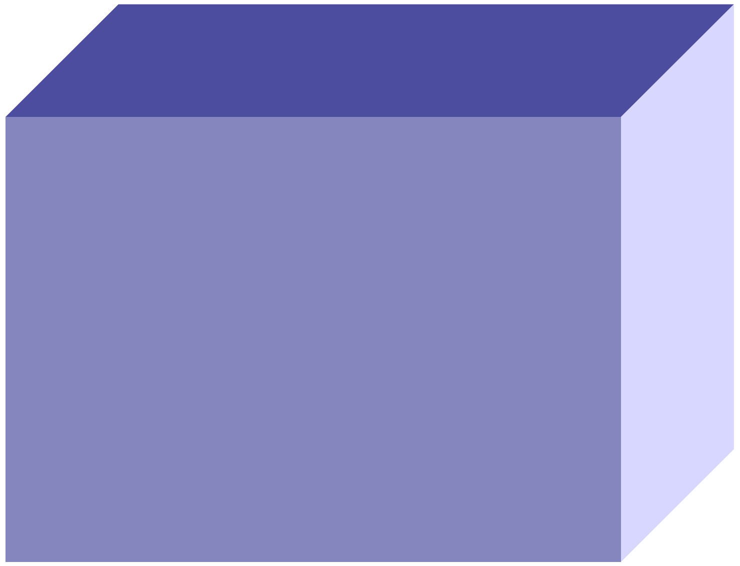
What is a … tensor?

- A **tensor** is a multi-way extension of a matrix
  - A multi-dimensional array
  - A multi-linear map
- In particular, the following are all tensors:
  - Scalars
  - Vectors
  - Matrices

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

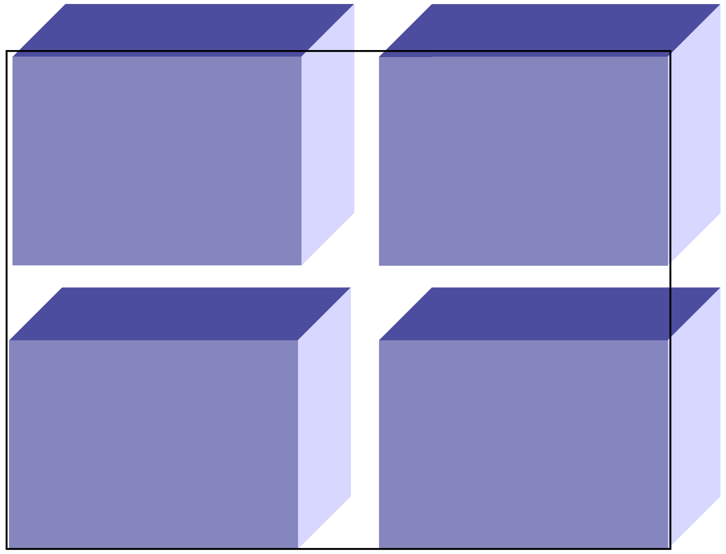
What is a … tensor?

- A **tensor** is a multi-way extension of a matrix
  - A multi-dimensional array
  - A multi-linear map
- In particular, the following are all tensors:
  - Scalars
  - Vectors
  - Matrices



What is a … tensor?

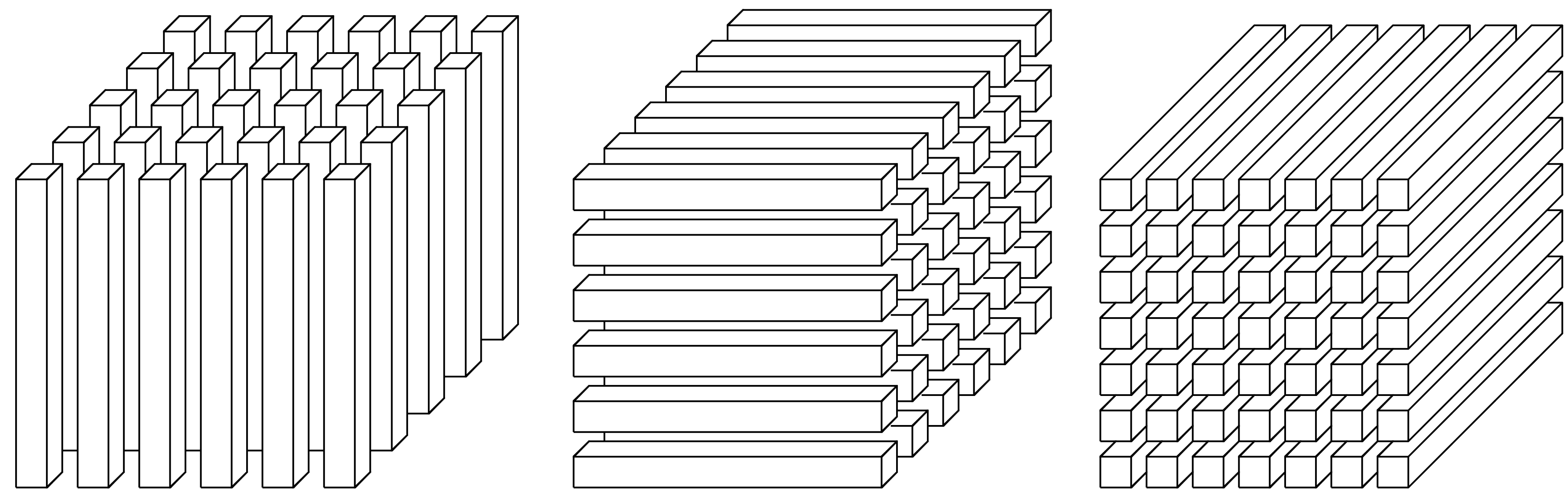
- A **tensor** is a multi-way extension of a matrix
  - A multi-dimensional array
  - A multi-linear map
- In particular, the following are all tensors:
  - Scalars
  - Vectors
  - Matrices



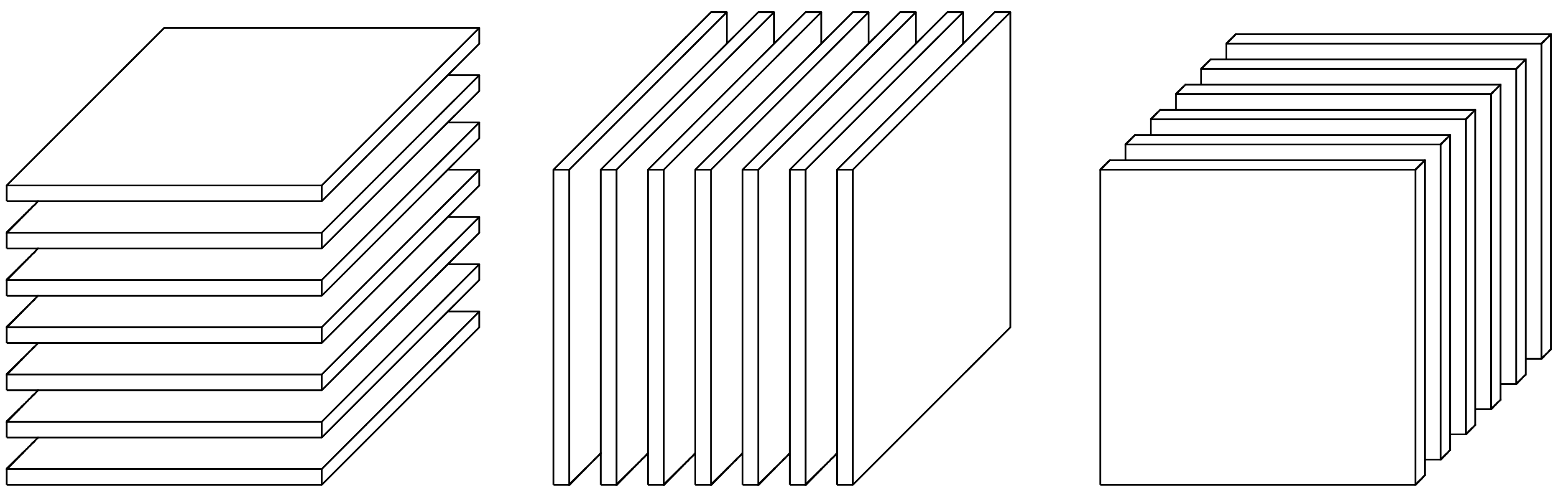
Why Tensors?

- Tensors can be used when matrices are not enough
- A matrix can represent a binary relation
  - A tensor can represent an *n*-ary relation
    - E.g. subject–predicate–object data
  - A tensor can represent a set of binary relations
    - Or other matrices
- A matrix can represent a matrix
  - A tensor can represent a series/set of matrices
  - But using tensors for time series should be approached with care

Fibres 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 $\mathbf{X}_k$)

Kolda & Bader 2009

Basic Operations

- Tensors require extensions to the standard linear algebra operations for matrices
- A **multi-way vector outer product** is a tensor where each element is the product of corresponding elements in vectors: $\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$, $(\mathcal{X})_{ijk} = a_i b_j c_k$
- A **tensor inner product** of two same-sized tensors is the sum of the element-wise products of their values: $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i=1}^{I} \sum_{j=1}^{J} \cdots \sum_{z=1}^{Z} x_{ij\cdots z} y_{ij\cdots z}$

Tensor Matricization

- Tensor **matricization** unfolds an N-way tensor into a matrix
  - **Mode-*n* matricization** arranges the mode-n fibers as columns of a matrix
    - Denoted $\mathbf{X}_{(n)}$
  - As many rows as is the dimensionality of the nth mode
  - As many columns as is the product of the dimensions of the other modes
- If $\mathcal{X}$ is an N-way tensor of size $I_1 \times I_2 \times \ldots \times I_N$, then $\mathbf{X}_{(n)}$ maps element $x_{i_1,i_2,\ldots,i_N}$ into (i_n, j) where

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

Matricization Example

$$\boldsymbol{\mathcal{X}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

Matricization Example

$$\boldsymbol{\mathcal{X}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

Matricization Example

$$\mathcal{X} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

$$\mathbf{X}_{(1)} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

Matricization Example

$$\mathcal{X} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

$$\mathbf{X}_{(1)} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

$$\mathbf{X}_{(2)} = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

$$\mathbf{X}_{(3)} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 \end{pmatrix}$$

Tensor Multiplication

- Let $\mathcal{X}$ be an N-way tensor of size $I_1 \times I_2 \times \ldots \times I_N$, and let $\mathbf{U}$ be a matrix of size $J \times I_n$
  - The ***n*-mode matrix product** of $\mathcal{X}$ with $\mathbf{U}$, $\mathcal{X} \times_n \mathbf{U}$ is of size $I_1 \times I_2 \times \ldots \times I_{n-1} \times J \times I_{n+1} \times \ldots \times I_N$
  - $(\mathcal{X} \times_n \mathbf{U})_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} u_{j i_n}$
    - Each mode-n fibre is multiplied by the matrix $\mathbf{U}$
  - In terms of unfold tensors: $\mathcal{Y} = \mathcal{X} \times_n \mathbf{U} \iff \mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}$
- The ***n*-mode vector product** is denoted $\mathcal{X} \bar{\times}_n \mathbf{v}$
  - The result is of order N–1
  - $(\mathcal{X} \bar{\times}_n \mathbf{v})_{i_1 \cdots i_{n-1} i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_N} v_{i_n}$
    - Inner product between mode-n fibres and vector $\mathbf{v}$

Kronecker Matrix Product

- Element-per-matrix product
- *n*-by-*m* and *j*-by-*k* matrices give *nj*-by-*mk* matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \cdots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \cdots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \cdots & a_{n,m}\mathbf{B} \end{pmatrix}$$

Khatri–Rao Matrix Product

- Element-per-column product
  - Number of columns must match
- *n*-by-*m* and *k*-by-*m* matrices give *nk*-by-*m* matrix

$$\mathbf{A} \odot \mathbf{B} = \begin{pmatrix} a_{1,1}\mathbf{b}_1 & a_{1,2}\mathbf{b}_2 & \cdots & a_{1,m}\mathbf{b}_m \\ a_{2,1}\mathbf{b}_1 & a_{2,2}\mathbf{b}_2 & \cdots & a_{2,m}\mathbf{b}_m \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{b}_1 & a_{n,2}\mathbf{b}_2 & \cdots & a_{n,m}\mathbf{b}_m \end{pmatrix}$$

Hadamard Matrix Product

- The element-wise matrix product
- Two matrices of size *n*-by-*m*, resulting matrix of size *n*-by-*m*

$$\mathbf{A} * \mathbf{B} = \begin{pmatrix} a_{1,1}b_{1,1} & a_{1,2}b_{1,2} & \cdots & a_{1,m}b_{1,m} \\ a_{2,1}b_{2,1} & a_{2,2}b_{2,2} & \cdots & a_{2,m}b_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}b_{n,1} & a_{n,2}b_{n,2} & \cdots & a_{n,m}b_{n,m} \end{pmatrix}$$

Some identities

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$$

$$(\mathbf{A} \otimes \mathbf{B})^\dagger = \mathbf{A}^\dagger \otimes \mathbf{B}^\dagger$$

$$\mathbf{A} \odot \mathbf{B} \odot \mathbf{C} = (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C} = \mathbf{A} \odot (\mathbf{B} \odot \mathbf{C})$$

$$(\mathbf{A} \odot \mathbf{B})^T (\mathbf{A} \odot \mathbf{B}) = \mathbf{A}^T \mathbf{A} * \mathbf{B}^T \mathbf{B}$$

$$(\mathbf{A} \odot \mathbf{B})^\dagger = \left((\mathbf{A}^T \mathbf{A}) * (\mathbf{B}^T \mathbf{B})\right)^\dagger (\mathbf{A} \odot \mathbf{B})^T$$

$\mathbf{A}^\dagger$ is the Moore–Penrose pseudo-inverse

Tensor Decompositions and Rank

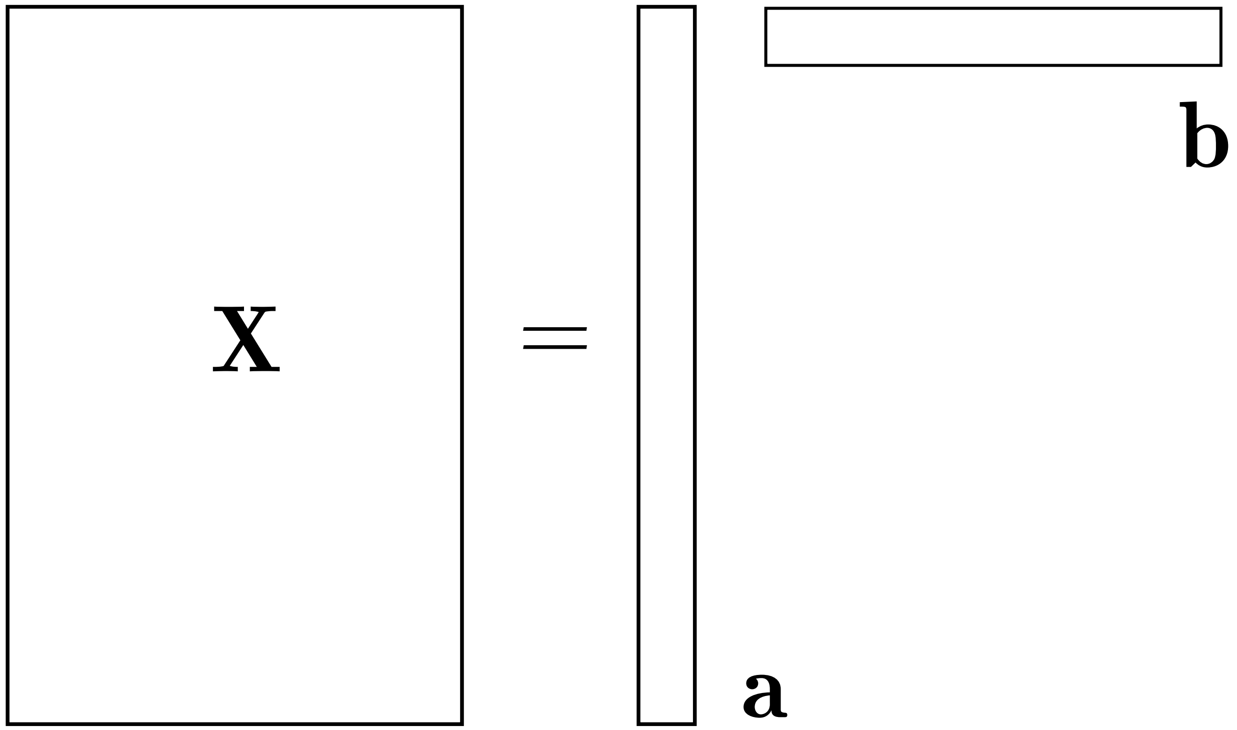
- A matrix decomposition represents the given matrix as a product of two (or more) **factor matrices**
- The **rank** of a matrix **M** is the
  - Number of linearly independent rows (*row rank*)
  - Number of linearly independent columns (*column rank*)
  - Number of rank-1 matrices needed to be summed to get **M** (*Schein rank*)
    - Rank-1 matrix is an outer product of two vectors
  - They all are equivalent

Tensor Decompositions and Rank

- A matrix decomposition represents the given matrix as a product of two (or more) **factor matrices**
- The **rank** of a matrix **M** is the
  - Number of linearly independent rows (*row rank*)
  - Number of linearly independent columns (*column rank*)
  - Number of rank-1 matrices needed to be summed to get **M** (*Schein rank*)
    - Rank-1 matrix is an outer product of two vectors
  - They all are equivalent

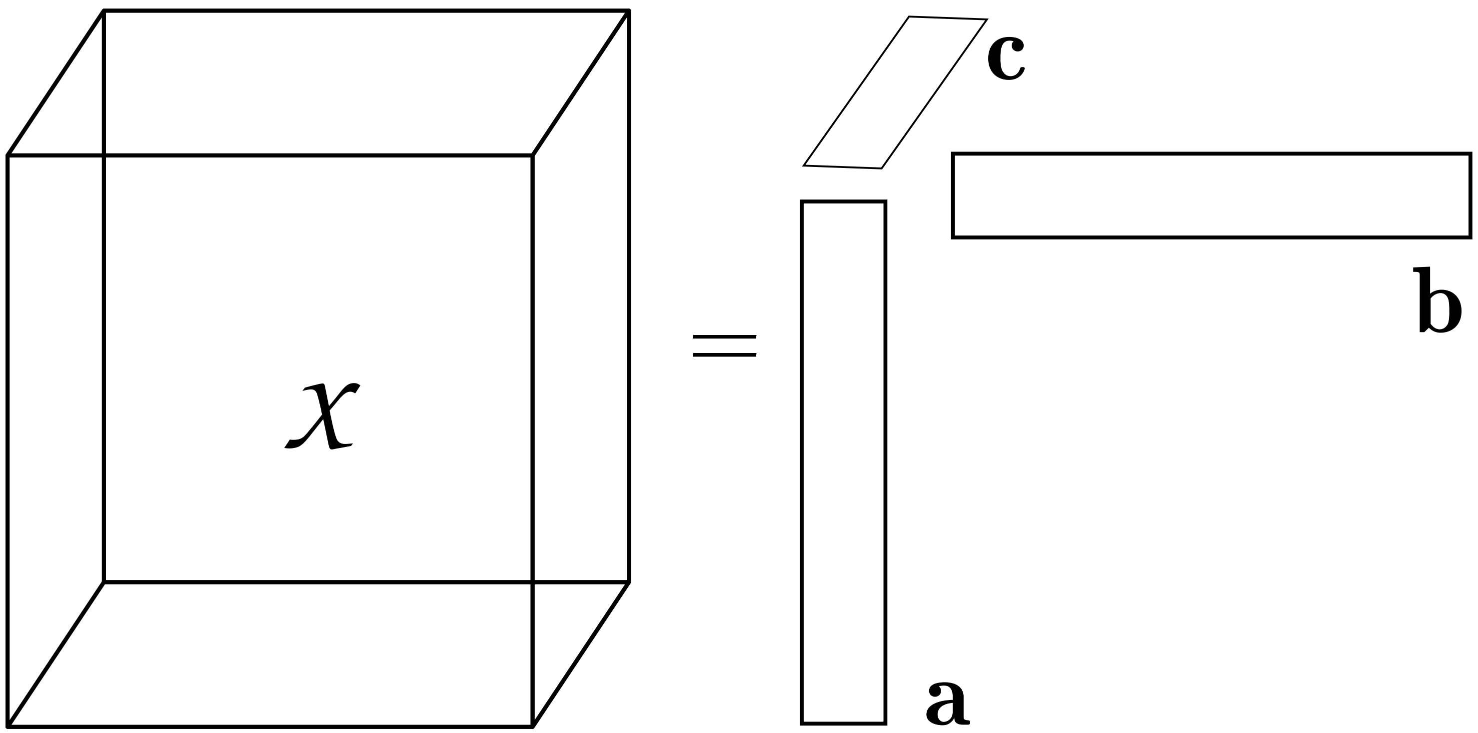
This we generalize

Rank-1 Tensors



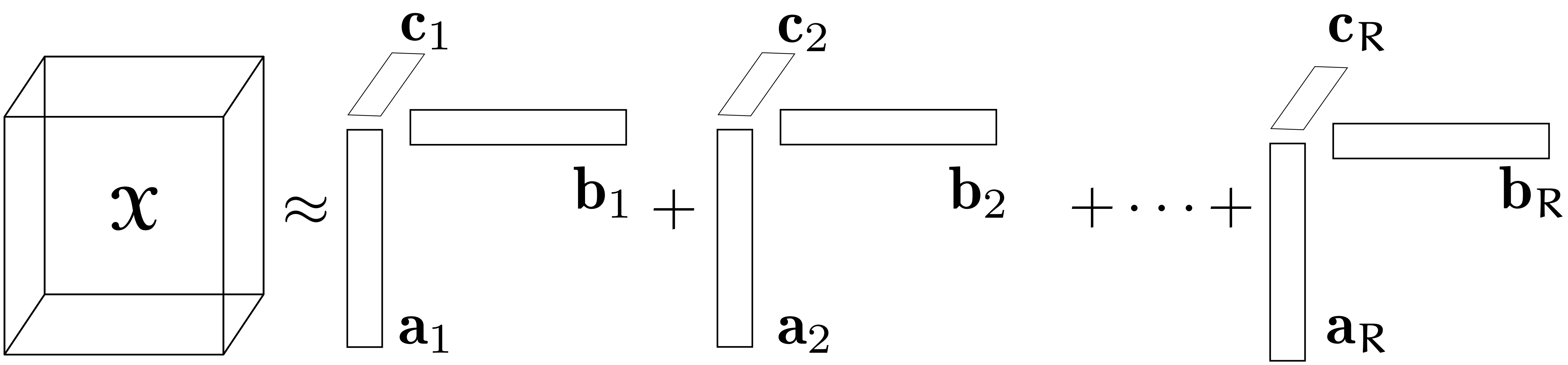
$$\mathbf{X} = \mathbf{a} \circ \mathbf{b}$$

Rank-1 Tensors



$$\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$$

The CP Tensor Decomposition



$$x_{ijk} \approx \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$$

More on CP

- The *size* of the CP factorization is the number of rank-1 tensors involved
- The factorization can also be written using *N* factor matrix (for order-*N* tensor)
  - All column vectors are collected in one matrix, all row vectors in other, all tube vectors in third, etc.
  - These matrices are typically called **A, B,** and **C** for 3rd order tensors

CANDECOM, PARAFAC, ...

Name	Proposed by
Polyadic Form of a Tensor	Hitchcock, 1927 [105]
PARAFAC (Parallel Factors)	Harshman, 1970 [90]
CANDECOMP or CAND (Canonical decomposition)	Carroll and Chang, 1970 [38]
Topographic Components Model	Möcks, 1988 [166]
CP (CANDECOMP/PARAFAC)	Kiers, 2000 [122]

Table 3.1: Some of the many names for the CP decomposition.

Kolda & Bader 2009

Another View on the CP

- Using matricization, we can re-write the CP decomposition
  - One equation per mode

$$\mathbf{X}_{(1)} = \mathbf{A}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}$$
$$\mathbf{X}_{(2)} = \mathbf{B}(\mathbf{C} \odot \mathbf{A})^{\mathsf{T}}$$
$$\mathbf{X}_{(3)} = \mathbf{C}(\mathbf{B} \odot \mathbf{A})^{\mathsf{T}}$$

Solving CP: The ALS Approach

1. Fix **B** and **C** and solve **A**
2. Solve **B** and **C** similarly
3. Repeat until convergence

Solving CP: The ALS Approach

1. Fix **B** and **C** and solve **A**
2. Solve **B** and **C** similarly
3. Repeat until convergence

$$\min_{\mathbf{A}} \left\| \mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}} \right\|_{\mathsf{F}}$$

Solving CP: The ALS Approach

1. Fix **B** and **C** and solve **A**
2. Solve **B** and **C** similarly
3. Repeat until convergence

$$\min_{\mathbf{A}} \left\| \mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}} \right\|_{\mathsf{F}}$$

$$\mathbf{A} = \mathbf{X}_{(1)} \left((\mathbf{C} \odot \mathbf{B})^{\mathsf{T}} \right)^{\dagger}$$

Solving CP: The ALS Approach

1. Fix **B** and **C** and solve **A**
2. Solve **B** and **C** similarly
3. Repeat until convergence

$$\min_{\mathbf{A}} \left\| \mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}} \right\|_{\mathsf{F}}$$

$$\mathbf{A} = \mathbf{X}_{(1)} \left((\mathbf{C} \odot \mathbf{B})^{\mathsf{T}} \right)^{\dagger}$$

$$\mathbf{A} = \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B}) (\mathbf{C}^{\mathsf{T}}\mathbf{C} * \mathbf{B}^{\mathsf{T}}\mathbf{B})^{\dagger}$$

Solving CP: The ALS Approach

1. Fix **B** and **C** and solve **A**
2. Solve **B** and **C** similarly
3. Repeat until convergence

$$\min_{\mathbf{A}} \left\| \mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}} \right\|_F$$

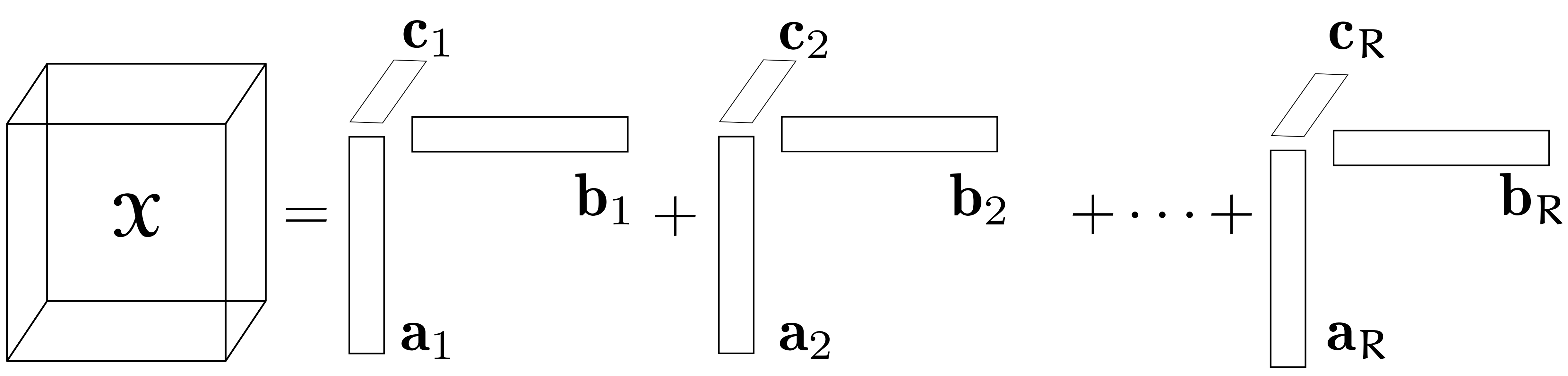
$$\mathbf{A} = \mathbf{X}_{(1)} \left((\mathbf{C} \odot \mathbf{B})^{\mathsf{T}} \right)^{\dagger}$$

$$\mathbf{A} = \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B}) (\mathbf{C}^{\mathsf{T}}\mathbf{C} * \mathbf{B}^{\mathsf{T}}\mathbf{B})^{\dagger}$$

R-by-R matrix

Tensor Rank

- The **rank** of a tensor is the minimum number of rank-1 tensors needed to represent the tensor exactly
  - The CP decomposition of size *R*
  - Generalizes the matrix Schein rank



Tensor Rank Oddities #1

- The rank of a (real-valued) tensor is different over reals and over complex numbers.
  - With reals, the rank can be *larger* than the largest dimension
    - rank($\mathcal{X}$) $\leq \min\{IJ, IK, JK\}$ for I-by-J-by-K tensor

$$\mathcal{X} = \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \right)$$

$$\mathbf{A} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -\mathfrak{i} & \mathfrak{i} \end{pmatrix},$$

$$\mathbf{B} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ \mathfrak{i} & -\mathfrak{i} \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 1 & 1 \\ \mathfrak{i} & -1 \end{pmatrix}$$

Tensor Rank Oddities #1

- The rank of a (real-valued) tensor is different over reals and over complex numbers.
  - With reals, the rank can be *larger* than the largest dimension
    - $\mathrm{rank}(\mathcal{X}) \leq \min\{IJ, IK, JK\}$ for I-by-J-by-K tensor

$$\mathcal{X} = \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \right)$$

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$$

Tensor Rank Oddities #1

- The rank of a (real-valued) tensor is different over reals and over complex numbers.
  - With reals, the rank can be *larger* than the largest dimension

$$\mathcal{X} = \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \right)$$

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$$

Tensor Rank Oddities #1

- The rank of a (real-valued) tensor is different over reals and over complex numbers.
  - With reals, the rank can be *larger* than the largest dimension
    - $\mathrm{rank}(\mathcal{X}) \leq \min\{IJ, IK, JK\}$ for I-by-J-by-K tensor

$$\mathcal{X} = \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \right)$$

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix},$$

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$$

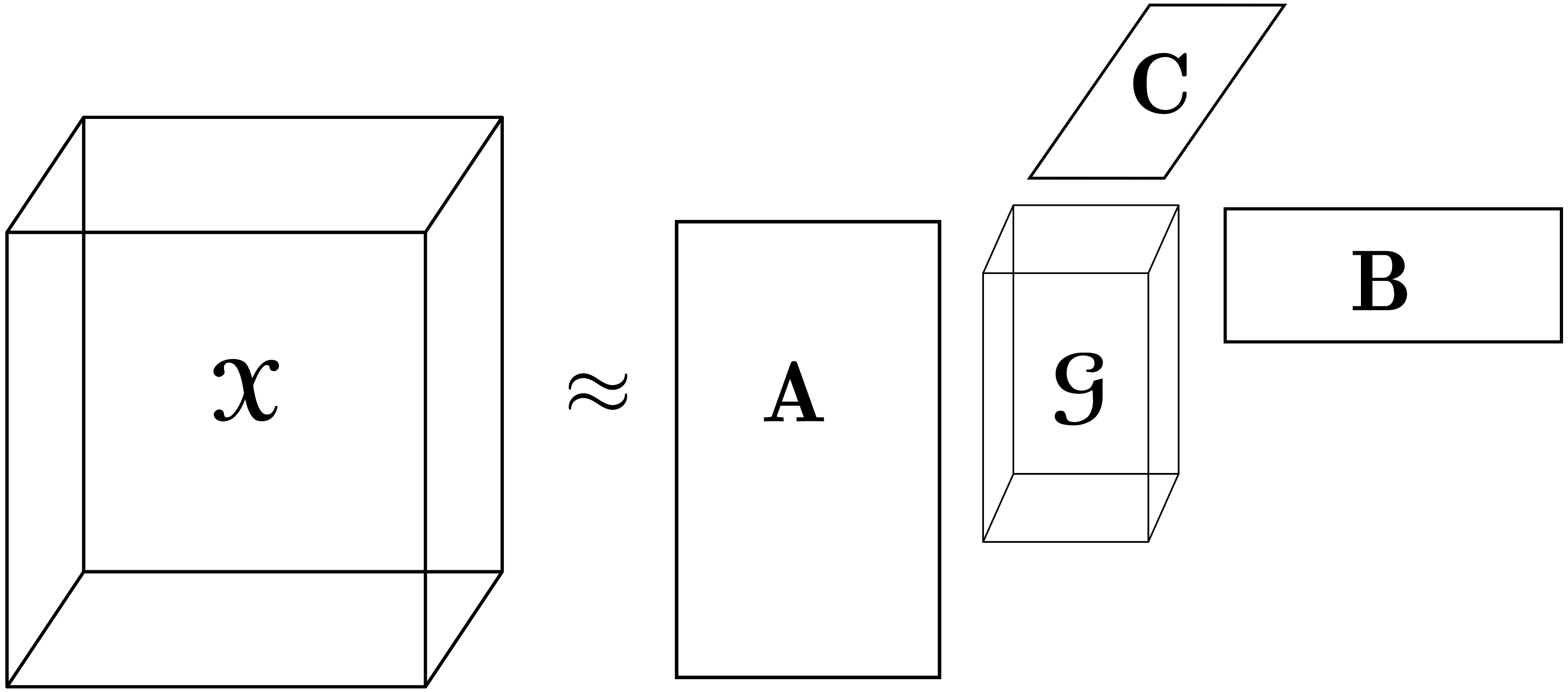
Tensor Rank Oddities #2

- There are tensors of rank R that can be approximated arbitrarily well with tensors of rank R' for some $R' < R$.
  - That is, there are no *best* low-rank approximation for such tensors.
  - Eckart–Young-theorem shows this is impossible with matrices.
  - The smallest such R' is called the **border rank** of the tensor.

Tensor Rank Oddities #3

- The rank-R CP decomposition of a rank-R tensor is *essentially unique* under mild conditions.
  - Essentially unique = only scaling and permuting are allowed.
  - Does not contradict #2, as this is the rank decomposition, not low-rank decomposition.
  - Again, not true for matrices (unless orthogonality etc. is required).

The Tucker Tensor Decomposition



$$x_{ijk} \approx \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} a_{ip} b_{jq} c_{kr}$$

Tucker Decomposition

- Many degrees of freedom: often **A**, **B**, and **C** are required to be orthogonal
- If P=Q=R and core tensor $\mathcal{G}$ is hyper-diagonal, then Tucker decomposition reduces to CP decomposition
- ALS-style methods are typically used
  - The matricized forms are

$$\mathbf{X}_{(1)} = \mathbf{A}\mathbf{G}_{(1)}(\mathbf{C} \otimes \mathbf{B})^T$$

$$\mathbf{X}_{(2)} = \mathbf{B}\mathbf{G}_{(2)}(\mathbf{C} \otimes \mathbf{A})^T$$

$$\mathbf{X}_{(3)} = \mathbf{C}\mathbf{G}_{(3)}(\mathbf{B} \otimes \mathbf{A})^T$$

Higher-Order SVD (HOSVD)

- One method to compute the Tucker decomposition
  - Set **A** as the leading P left singular vectors of $\mathbf{X}_{(1)}$
  - Set **B** as the leading Q left singular vectors of $\mathbf{X}_{(2)}$
  - Set **C** as the leading R left singular vectors of $\mathbf{X}_{(3)}$
  - Set tensor $\mathcal{G}$ as $\mathcal{X} \times_1 \mathbf{A}^T \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T$

Comments

- Tensors generalize matrices
- Many matrix concepts generalize as well
  - But some don't
  - And some behave very differently
- Compared to matrix decomposition methods, tensor algorithms are in their youth
  - Notwithstanding that Tucker did his work in 60's