Tensors in MATLAB

Brett Bader & Tammy Kolda
Sandia National Labs

Outline

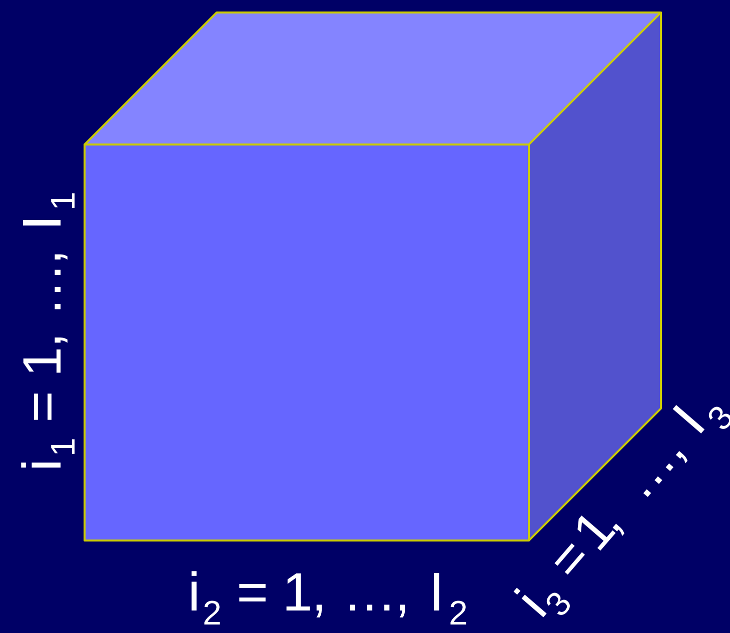
- Introduction & Notation
- Tensor Operations
  - Multiplying times a Matrix
  - Multiplying times a Vector
  - Multiplying times another Tensor
  - Matricization
- Storing Tensors in Factored Form
- Example Algorithms for Generating Factored Tensors

Introduction & Notation

Basic Notation

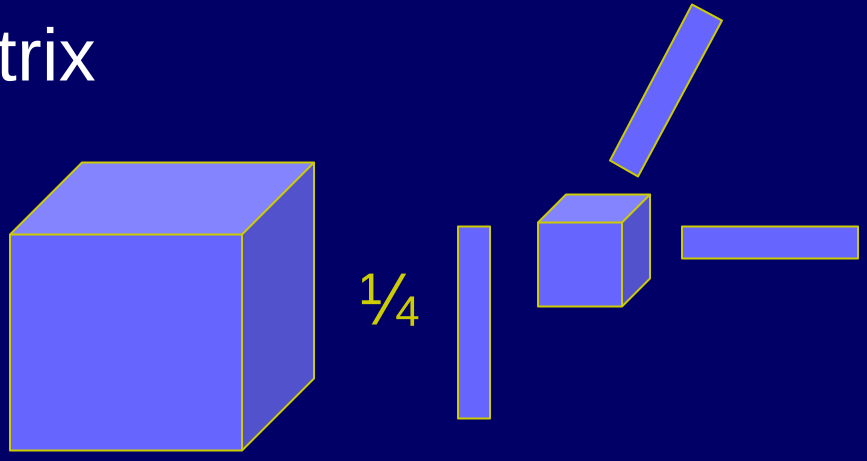
- Indices: n = 1, …, N
- Vector: a of size I_1
- Matrix: A of size I_1 £ I_2
- Tensor: A of size I_1 £ I_2 £ ··· £ I_N
- The *order* of A is N
  - “Higher-order” means N > 2
- The nth *mode* of A is of dimension I_n
  - mode = *dimension* or *way*

Tensor A of size
I_1 £ I_2 £ I_3



Operations on Tensors

- Element-wise: add, subtract, etc.
- Multiply
  - Times a vector or sequence of vectors
  - Times a matrix or sequence of matrices
  - Times another tensor
- Convert to / from a matrix
- Decompose



Tensor Operations

Tensors in MATLAB

- MATLAB is a high-level computing environment
- Higher-order tensors can be stored as multidimensional array (MDA) objects
- But operations on MDAs are limited
  - E.g., no matrix multiplication
- MATLAB’s class functionality enables users to create their own objects
- The `tensor` class extends the MDA capabilities to include multiplication and more
  - Will show examples at the end of the talk

n-Mode Multiplication (with a Matrix)

$$\mathcal{A} \times_n U$$

- Let A be a tensor of size I_1 £ I_2 £ ⋯ £ I_N
- Let U be a matrix of size J_n £ I_n
- Result size: I_1 £ ⋯ £ I_{n-1} £ J_n £ I_{n+1} £ ⋯ £ I_N

$$(\mathcal{A} \times_n U)(i_1, \ldots, i_{n-1}, j_n, i_{n+1}, \ldots, i_N) = \sum_{i_n=1}^{I_n} \mathcal{A}(i_1, i_2, \ldots, i_N)\, U(j_n, i_n).$$

Matrix Interpretation

- A of size m £ n, U of size m £ k, V of size n £ k, Σ of size k £ k

$$A \times_1 U^T = U^T A$$

$$A \times_2 V^T = AV$$

$$\Sigma \times_1 U \times_2 V = U \Sigma V^T$$

Property

$$\mathcal{A} \times_m U \times_n V$$

$$= (\mathcal{A} \times_m U) \times_n V$$

$$= (\mathcal{A} \times_n V) \times_m U$$

Multiplication with a Sequence of Matrices

- Let A be a tensor of size I_1 £ I_2 £ ··· £ I_N
- Let each $U^{(n)}$ be a matrix of size J_n £ I_n

$$\mathcal{B} = \mathcal{A} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)}$$

- B is a tensor of size J_1 £ J_2 £ ··· £ J_N
- New notation

$$\mathcal{B} = \mathcal{A} \times \{U\}$$

Multiplication with all but one of a Sequence of Matrices

- Let A be a tensor of size I_1 £ I_2 £ ⋯ £ I_N
- Let each $U^{(n)}$ be a matrix of size J_n £ I_n

$$\mathcal{B} = \mathcal{A} \times_1 U^{(1)} \cdots \times_{n-1} U^{(n-1)} \times_{n+1} U^{(n+1)} \cdots \times_N U^{(N)}$$

- B of size J_1 £ ⋯ £ J_{n-1} £ I_n £ J_{n+1} £ ⋯ £ J_N
- New notation

$$\mathcal{B} = \mathcal{A} \times_{-n} \{U\}$$

Tensor Multiplication with a Vector

$$\mathcal{A} \,\bar{\times}_n\, u$$

Bar over operator indicates contracted product.

- Let A be a tensor of size I_1 £ I_2 £ ⋯ £ I_N
- Let u be a vector of size I_n
- Result size: I_1 £ ⋯ £ I_{n-1} £ I_{n+1} £ ⋯ £ I_N (order N-1)

$$(\mathcal{A} \,\bar{\times}_n\, u)(i_1, \ldots, i_{n-1}, i_{n+1}, \ldots, i_N)$$

$$= \sum_{i_n=1}^{I_n} \mathcal{A}(i_1, i_2, \ldots, i_N)\, u(i_n).$$

Matrix Interpretation

- A of size m £ n, u of size m, v of size n

$$A \,\bar{\times}_1\, u = A^T u$$

$$A \,\bar{\times}_2\, v = Av$$

Order Matters in Vector Case

$$\mathcal{A} \bar{\times}_m u \bar{\times}_n v$$

$$= (\mathcal{A} \bar{\times}_m u) \bar{\times}_{n-1} v$$

$$= (\mathcal{A} \bar{\times}_n v) \bar{\times}_m u$$

(assuming $m < n$)

Multiplication with a Sequence of Vectors

- Let A be a tensor of size I_1 £ I_2 £ $\cdots$ £ I_N
- Let each $u^{(n)}$ be a vector of size I_n

$$\beta = \mathcal{A} \,\bar{\times}_1\, u^{(1)} \,\bar{\times}_2\, u^{(2)} \cdots \,\bar{\times}_N\, u^{(N)}$$

- β is a scalar
- New notation

$$\beta = \mathcal{A} \,\bar{\times}\, \{u\}$$

Multiplication with all but one of a Sequence of Vectors

- Let A be a tensor of size I_1 £ I_2 £ ··· £ I_N
- Let each $u^{(n)}$ be a matrix of size I_n

$$b = \mathcal{A} \,\bar{\times}_1\, u^{(1)} \cdots \,\bar{\times}_{n-1}\, u^{(n-1)} \,\bar{\times}_{n+1}\, u^{(n+1)} \cdots \,\bar{\times}_N\, u^{(N)}$$

- Result is vector b of size I_n
- New notation

$$b = \mathcal{A} \,\bar{\times}_{-n}\, \{u\}$$

Multiplying two Tensors

- Let A and B be tensors of size I_1 £ I_2 £ $\cdots$ £ I_N

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{A}(i_1, i_2, \ldots, i_N)\, \mathcal{B}(i_1, i_2, \ldots, i_N)$$

- Result is a scalar
- Frobenius norm is just $k\ A\ k^2 = < A, A >$

Multiplying two Tensors

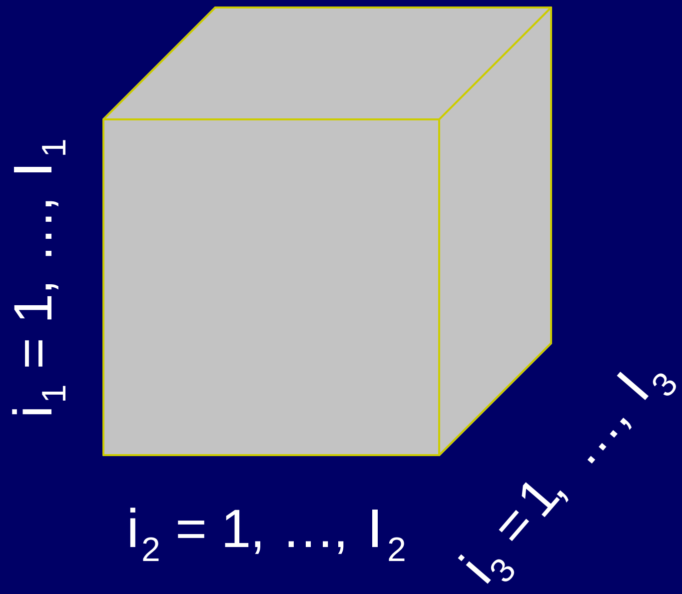
- Let A be of size I_1 £ ⋯ £ I_M £ J_1 £ ⋯ £ J_N
- Let B be of size I_1 £ ⋯ £ I_M £ K_1 £ ⋯ £ K_P

$$\langle \mathcal{A}, \mathcal{B} \rangle_{\{1,\ldots,M;1,\ldots,M\}} (j_1, \ldots j_N, k_1, \ldots, k_P) =$$

$$\sum_{i_1=1}^{I_1} \cdots \sum_{i_M=1}^{I_M} \mathcal{A}(i_1, \ldots, i_M, j_1, \ldots, j_N)\, \mathcal{B}(i_1, \ldots, i_M, k_1, \ldots, k_P).$$

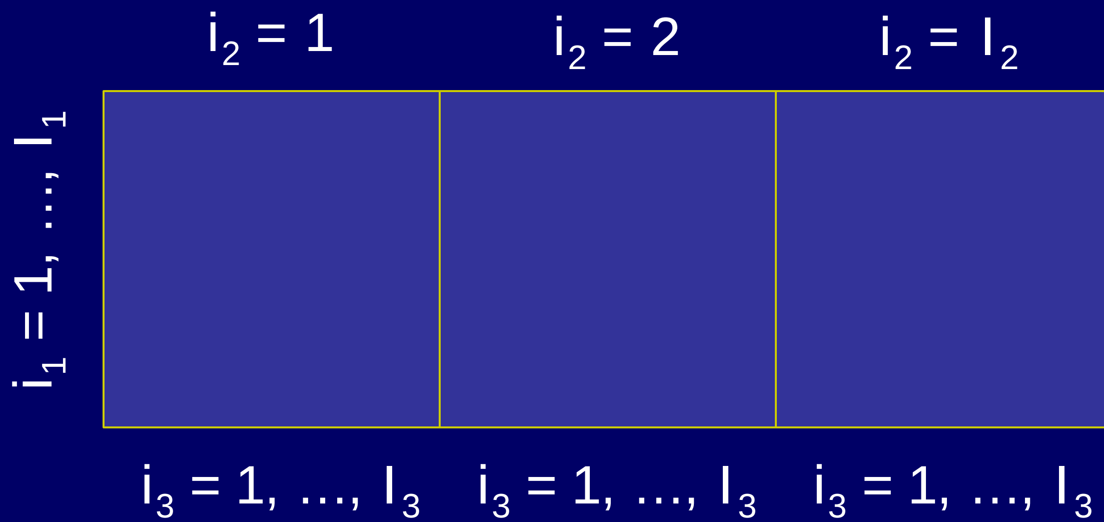
- Result is of size J_1 £ ⋯ £ J_N £ K_1 £ ⋯ £ K_P

Matricize: Converting a Tensor to a Matrix

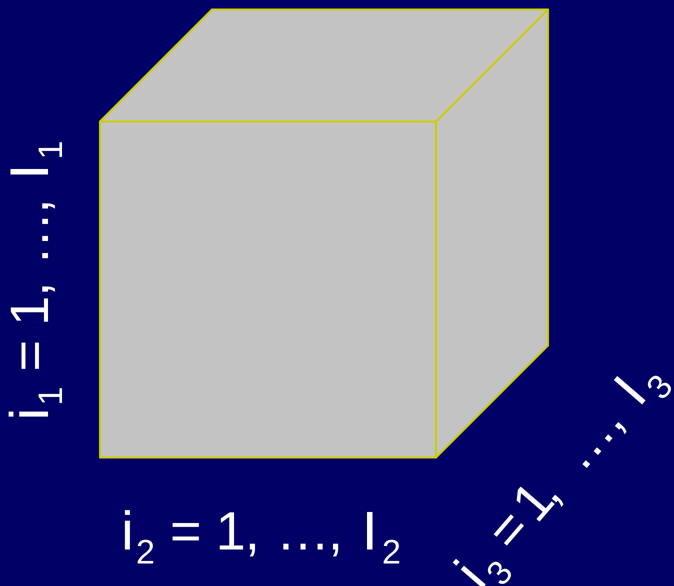


Key Point: Order of the columns doesn't matter so long as it is consistent.

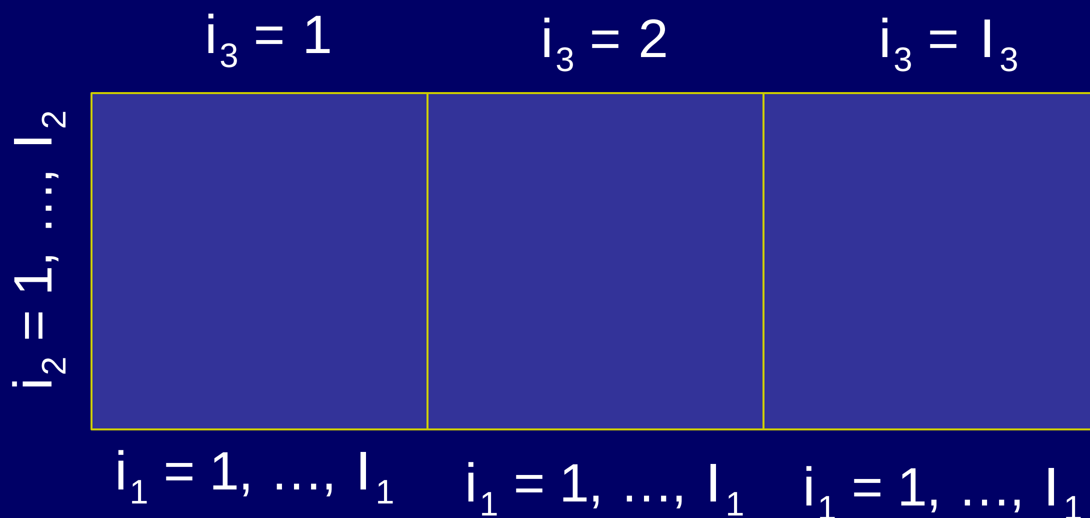
$$A_{(1)} =$$



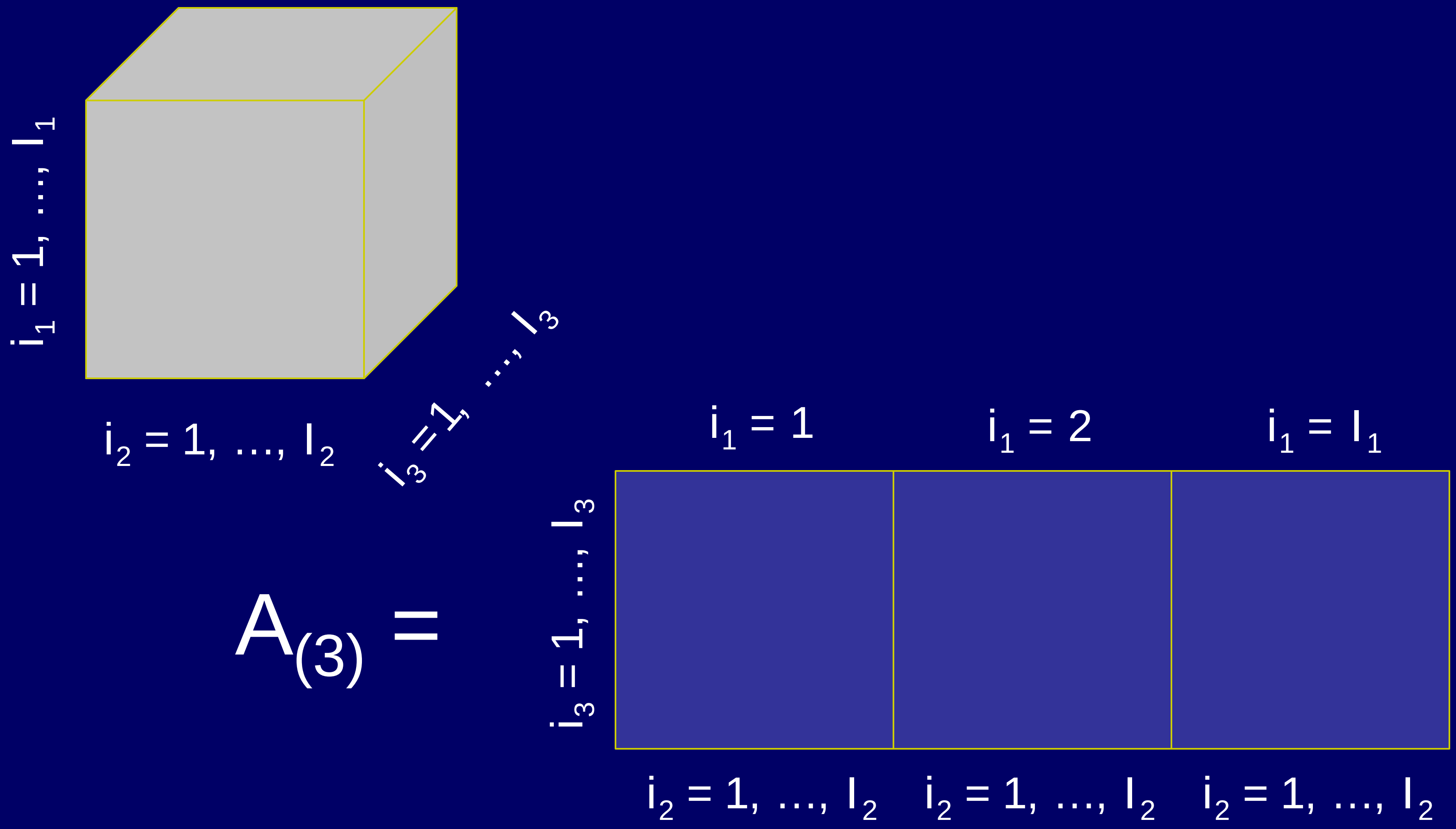
Matricize: Converting a Tensor to a Matrix



$A_{(2)} =$

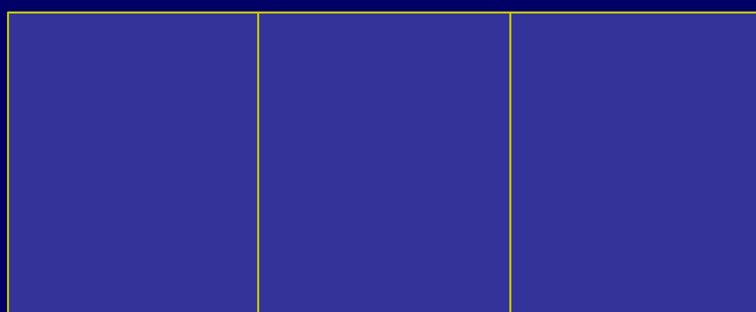


Matricize: Converting a Tensor to a Matrix

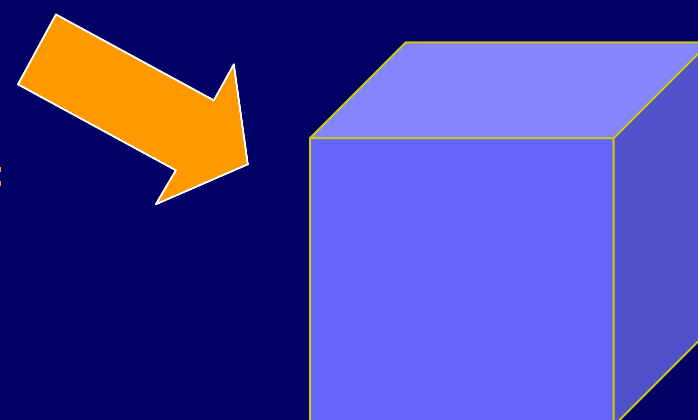


Inverse Matricize

- One may also take a matrix and convert it into a tensor



Need to know the size of the tensor as well as the mode (and type) of matricization



Matricization & Mode-n Multiplication

$$\mathcal{C} = \mathcal{A} \times_n B$$

$$C_{(n)} = BA_{(n)}$$

Summary on Tensor Operations

Tensor times Matrix

$$\mathcal{B} = \mathcal{A} \times_n U$$

$$\mathcal{B} = \mathcal{A} \times \{U\}$$

$$\mathcal{B} = \mathcal{A} \times_{-n} \{U\}$$

Tensor times Vector

$$\mathcal{B} = \mathcal{A} \bar{\times}_n u$$

$$\beta = \mathcal{A} \bar{\times} \{u\}$$

$$b = \mathcal{A} \bar{\times}_{-n} \{u\}$$

Tensor times Tensor

$$\langle \mathcal{B}, \mathcal{A} \rangle$$

Matricization

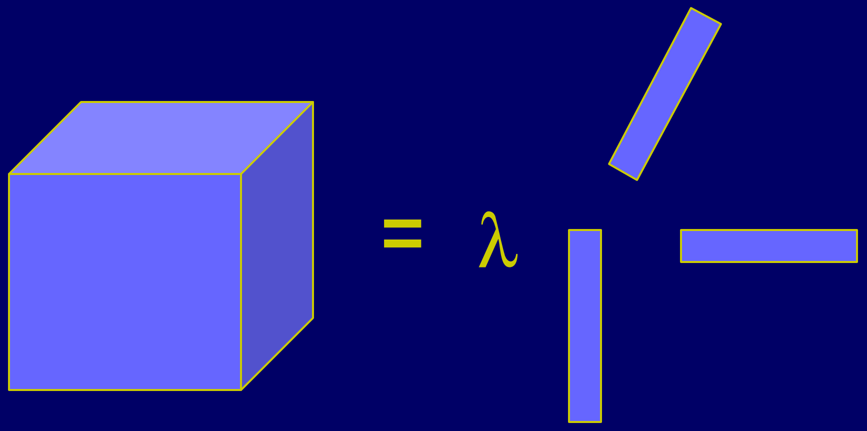
$$\mathcal{A} \Rightarrow A_{(n)}$$

Factored Tensors

Rank-1 Tensor

$$\mathcal{A} = \lambda\, u^{(1)} \circ u^{(2)} \circ \ldots \circ u^{(N)}$$

$$\mathcal{A}(i_1, i_2, \ldots, i_N) = \lambda\, u^{(1)}_{i_1}\, u^{(2)}_{i_2} \ldots u^{(N)}_{i_N}$$



CP Model

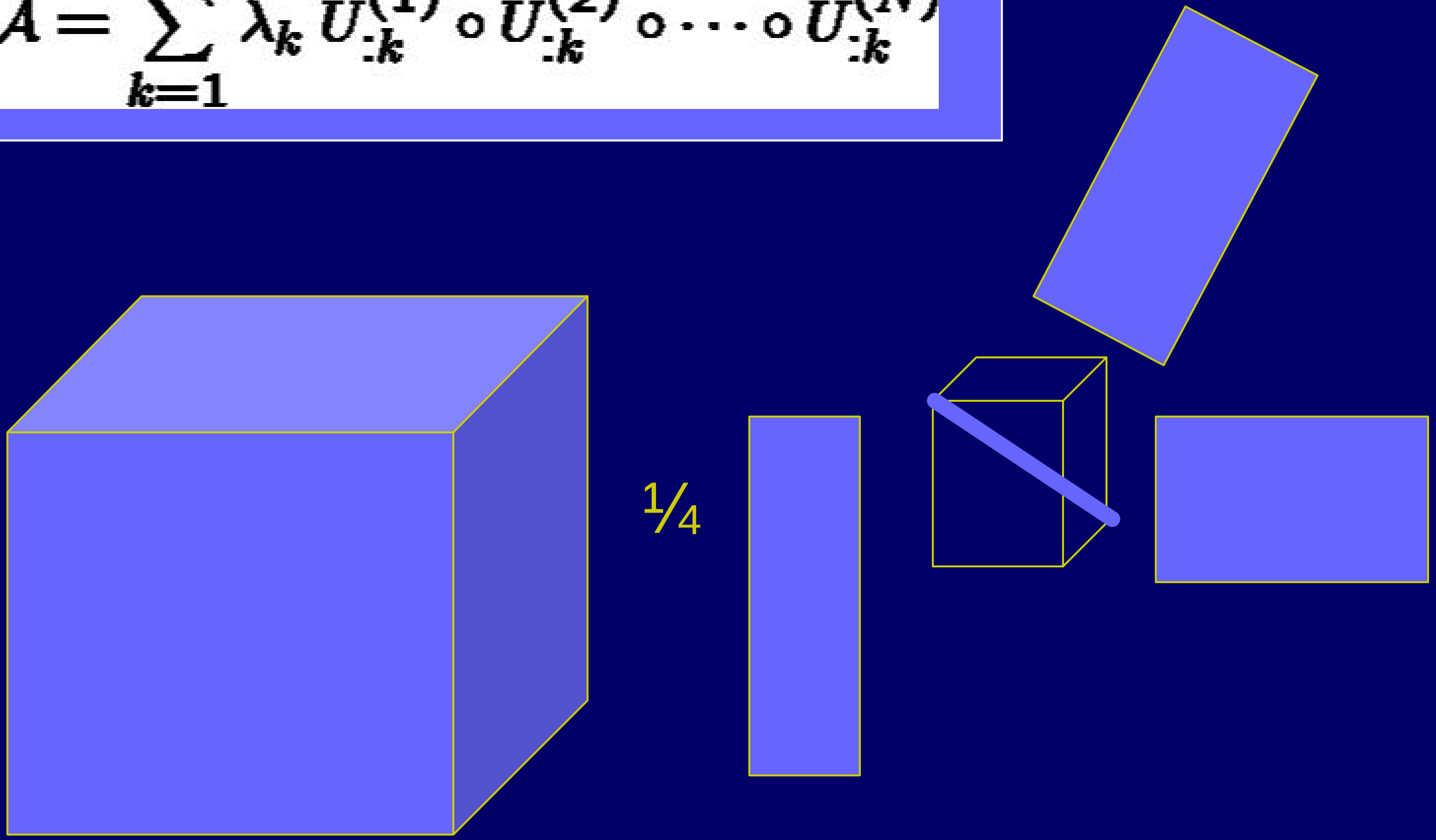
- “CP” is shorthand for CANDECOMP (Carrol and Chang, 1970) and PARAFAC (Harshman, 1970) – identical models that were developed independently

$$\mathcal{A} = \sum_{k=1}^{K} \lambda_k \, U_{:k}^{(1)} \circ U_{:k}^{(2)} \circ \cdots \circ U_{:k}^{(N)}$$

- λ is a K-vector
- Each $U^{(n)}$ is an I_n £ K matrix
- Tensor A is size I_1 £ I_2 £ $\cdots$ £ I_N

CP Model

$$\mathcal{A} = \sum_{k=1}^{K} \lambda_k \, U^{(1)}_{:k} \circ U^{(2)}_{:k} \circ \cdots \circ U^{(N)}_{:k}$$



Tucker Model

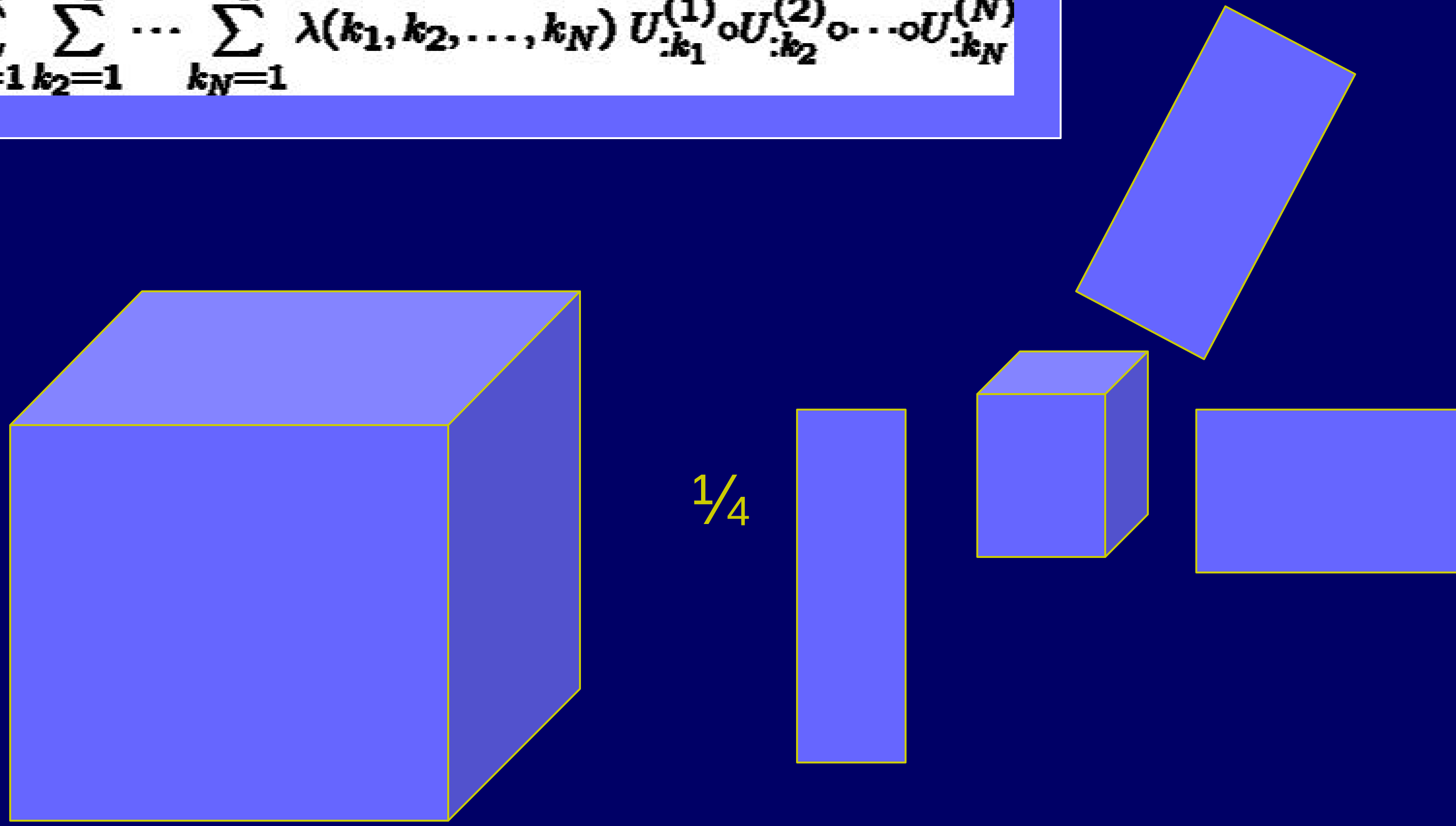
- Tucker, 1966

$$\mathcal{A} = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \cdots \sum_{k_N=1}^{K_N} \lambda(k_1, k_2, \ldots, k_N)\, U^{(1)}_{:k_1} \circ U^{(2)}_{:k_2} \circ \cdots \circ U^{(N)}_{:k_N}$$

- λ is a tensor of size K_1 £ K_2 £ ⋯ £ K_N
  - "Core Tensor" or "Core Array"
- Each $U^{(n)}$ is an I_n £ K_n matrix
- Tensor A is size I_1 £ I_2 £ ⋯ £ I_N

Tucker Model

$$\mathcal{A} = \sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}\cdots\sum_{k_N=1}^{K_N}\lambda(k_1,k_2,\ldots,k_N)\, U^{(1)}_{:k_1}\circ U^{(2)}_{:k_2}\circ\cdots\circ U^{(N)}_{:k_N}$$



Higher Order Power Method
De Lathauwer, De Moor, Vandewalle

- Compute a rank-1 approximation to a given tensor
- In: A of size I_1 £ I_2 £ ··· £ I_N
- Out: B = λ $u^{(1)}$ **?** $u^{(2)}$ **?** ··· **?** $u^{(N)}$ is a rank-one tensor of size I_1 £ I_2 £ ··· £ I_N that estimates A

HO Power Method

For $k = 1, 2, \ldots$ (until converged), do:

For $n = 1, \ldots, N$, do:

$$\tilde{u}_{k+1}^{(n)} = \mathcal{A} \,\bar{\times}_{-n} \{u_k\}.$$

$$\lambda_{k+1}^{(n)} = \left\| \tilde{u}_{k+1}^{(n)} \right\|$$

$$u_{k+1}^{(n)} = \tilde{u}_{k+1}^{(n)} / \lambda_{k+1}^{(n)}$$

Let $\lambda = \lambda_K$ and $\{u\} = \{u_K\}$ where K is the index of the final result of the iterations.

MATLAB Classes Examples

Note: MATLAB class does not replace Bro's N-Way Toolbox