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: $n = 1, \ldots, N$
- Vector: $a$ of size $I_1$
- Matrix: $A$ of size $I_1 \times I_2$
- Tensor: $A$ of size $I_1 \times I_2 \times \cdots \times I_N$
- The order of $A$ is $N$
  - “Higher-order” means $N > 2$
- The $n$th mode of $A$ is of dimension $I_n$
  - mode = *dimension* or *way*
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)

Let $A$ be a tensor of size $I_1 \mathbf{\times} I_2 \mathbf{\times} \cdots \mathbf{\times} I_N$

Let $U$ be a matrix of size $J_n \mathbf{\times} I_n$

Result size: $I_1 \mathbf{\times} I_{n-1} \mathbf{\times} J_n \mathbf{\times} I_{n+1} \mathbf{\times} \cdots \mathbf{\times} I_N$

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

- A of size $m \times n$, $U$ of size $m \times k$, $V$ of size $n \times k$, $\Sigma$ of size $k \times k$

\[
A \times_1 U^T = U^T A
\]

\[
A \times_2 V^T = AV
\]

\[
\Sigma \times_1 U \times_2 V = U \Sigma V^T
\]
\[ A \times_m U \times_n V \]

\[ = (A \times_m U) \times_n V \]

\[ = (A \times_n V) \times_m U \]
Multiplication with a Sequence of Matrices

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

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

- $B$ is a tensor of size $J_1 \times J_2 \times \cdots \times J_N$
- New notation

\[
B = A \times \{U\}
\]
Multiplication with **all but one** of a Sequence of Matrices

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

$$
B = A \times_1 U^{(1)} \times_2 U^{(n-1)} \times_{n+1} U^{(n+1)} \cdots \times_N U^{(N)}
$$

- $B$ of size $J_1 \times \cdots \times J_{n-1} \times I_n \times J_{n+1} \times \cdots \times J_N$
- New notation

$$
B = A \times_{-n} \{U\}
$$
Tensor Multiplication with a Vector

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

$$(A \times_n u)(i_1, \ldots, i_{n-1}, i_{n+1}, \ldots, i_N)$$

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

- $A$ of size $m \times n$, $u$ of size $m$, $v$ of size $n$

\[ A \times_1 u = A^T u \]

\[ A \times_2 v = Av \]
Order Matters in Vector Case

\[ A \times_m u \times_n v \]

\[ = (A \times_m u) \times_{n-1} v \]

\[ = (A \times_n v) \times_m u \]

(assuming \( m < n \))
Multiplication with a Sequence of Vectors

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

\[
\beta = \mathbf{A} \times_1 u^{(1)} \times_2 u^{(2)} \cdots \times_N u^{(N)}
\]

- $\beta$ is a scalar
- New notation

\[
\beta = \mathbf{A} \times \{\mathbf{u}\}
\]
Multiplication with all but one of a Sequence of Vectors

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

Result is vector $b$ of size $I_n$

New notation

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

$$b = A \bar{X}_{-n} \{u\}$$
Let \( A \) and \( B \) be tensors of size \( I_1 \times I_2 \times \ldots \times I_N \).

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

- Result is a scalar
- Frobenius norm is just \( kA \ k^2 = \langle A, A \rangle \)
Multiplying two Tensors

- Let $A$ be of size $I_1 \leq I_M \leq I_{1:M} \leq I_N$
- Let $B$ be of size $I_1 \leq I_M \leq I_{1:M} \leq K_P$

Result is of size $J_1 \leq J_N \leq K_1 \leq K_P$
Matricize: Converting a Tensor to a Matrix

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

\[ A_{(1)} = \]

\[ \begin{array}{ccc}
  i_3 = 1, \ldots, l_3 & i_3 = 1, \ldots, l_3 & i_3 = 1, \ldots, l_3 \\
  i_2 = 2 & i_2 = 2 & i_2 = l_2 \\
  i_1 = 1, \ldots, l_1 & i_1 = 1, \ldots, l_1 & i_1 = 1, \ldots, l_1 \\
\end{array} \]
Matricize: Converting a Tensor to a Matrix

$$i_1 = 1, \ldots, l_1$$

$$i_2 = 1, \ldots, l_2$$

$$i_3 = 1, \ldots, l_3$$

$$A_{(2)} =$$

$$i_1 = 1, \ldots, l_1$$

$$i_2 = 1, \ldots, l_2$$

$$i_3 = 1$$

$$i_3 = 2$$

$$i_3 = l_3$$
Matricize: Converting a Tensor to a Matrix

\[ A^{(3)} = \]

\[ i_1 = 1, \ldots, l_1 \]
\[ i_2 = 1, \ldots, l_2 \]
\[ i_3 = 1, \ldots, l_3 \]
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

\[ C = A \times_n B \]

\[ C_{(n)} = BA_{(n)} \]
Summary on Tensor Operations

Tensor times Matrix

\[ \mathbf{B} = \mathbf{A} \times_n \mathbf{U} \]
\[ \mathbf{B} = \mathbf{A} \times \{\mathbf{U}\} \]
\[ \mathbf{B} = \mathbf{A} \times_{-n} \{\mathbf{U}\} \]

Tensor times Vector

\[ \mathbf{B} = \mathbf{A} \times_n \mathbf{u} \]
\[ \mathbf{\beta} = \mathbf{A} \times \{\mathbf{u}\} \]
\[ \mathbf{b} = \mathbf{A} \times_{-n} \{\mathbf{u}\} \]

Tensor times Tensor

\[ \langle \mathbf{B}, \mathbf{A} \rangle \]

Matricization

\[ \mathbf{A} \Rightarrow \mathbf{A}(n) \]
Factored Tensors
Rank-1 Tensor

\[ A = \lambda \ u^{(1)} \circ u^{(2)} \circ \ldots \circ u^{(N)} \]

\[ A(i_1, i_2, \ldots, i_N) = \lambda \ u_{i_1}^{(1)} u_{i_2}^{(2)} \ldots u_{i_N}^{(N)} \]
“CP” is shorthand for CANDECOMP (Carrol and Chang, 1970) and PARAFAC (Harshman, 1970) – identical models that were developed independently.

\[ A = \sum_{k=1}^{K} \lambda_k U^{(1)} \circ U^{(2)} \circ \cdots \circ U^{(N)} \]

- \( \lambda \) is a \( K \)-vector.
- Each \( U^{(n)} \) is an \( I_n \times K \) matrix.
- Tensor \( A \) is size \( I_1 \times I_2 \times \cdots \times I_N \).
CP Model

\[ A = \sum_{k=1}^{K} \lambda_k U_{:k}^{(1)} \circ U_{:k}^{(2)} \circ \ldots \circ U_{:k}^{(N)} \]
Tucker Model

- Tucker, 1966

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

- \( \lambda \) is a tensor of size \( K_1 \times K_2 \times \cdots \times K_N \)
  - “Core Tensor” or “Core Array”
- Each \( U^{(n)} \) is an \( I_n \times K_n \) matrix
- Tensor \( A \) is size \( I_1 \times I_2 \times \cdots \times I_N \)
Tucker Model

\[ A = \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \cdots \sum_{k_N=1}^{K_N} \lambda(\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_N) \circ U^{(1)}_{k_1} \circ U^{(2)}_{k_2} \circ \cdots \circ U^{(N)}_{k_N} \]
Higher Order Power Method
De Lathauwer, De Moor, Vandewalle

- Compute a rank-1 approximation to a given tensor

**In:** $A$ of size $I_1 \times I_2 \times \ldots \times I_N$

**Out:** $B = \lambda u^{(1)} \otimes u^{(2)} \otimes \ldots \otimes u^{(N)}$ is a rank-one tensor of size $I_1 \times I_2 \times \ldots \times I_N$ that estimates $A$
HO Power Method

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

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

$$\tilde{u}_{k+1}^{(n)} = A \tilde{x}_n \{ u_k \}.$$ 

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

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

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