Developing Tensor Operations with an Underlying Group Structure

Carla D. Martin
(with M. Kilmer, Tufts University)

James Madison University

NSF Workshop:
Future Directions in Tensor-Based Computation and Modeling
February 21, 2009
Tensor Decompositions (Tucker/HOSVD, PARAFAC)

Let \( A \in \mathbb{R}^{M \times N \times P} \)

\[
A = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{P} \sigma_{ijk} (u_i \circ v_j \circ w_k)
\]

\[
= \Sigma_1 U \times_2 V \times_3 W
\]

\[
A = \sum_{i=1}^{r} (u_i \circ v_i \circ w_i)
\]
What other factorizations are possible?

- Develop different notions of factorizations and projections based on different tensor operations
- Tie factorizations to fundamental concepts in linear algebra such as group structure, invertibility, existence, uniqueness
- New compression strategies that may be modified for tensors with special structure
- Investigate computational efficiencies with regard to sparse and dense tensors
Tensor-tensor Multiplication (contracted product)

Contracted product in the \textit{first-mode}:

\[ \mathcal{A} \in \mathbb{R}^{L \times M_1 \times N_1} \]
\[ \mathcal{B} \in \mathbb{R}^{L \times M_2 \times N_2} \]

\[ \Rightarrow \quad \mathcal{A}\mathcal{B} \in \mathbb{R}^{M_1 \times N_1 \times M_2 \times N_2} \]

\[ (\mathcal{A}\mathcal{B})_{m_1 n_1 m_2 n_2} = \sum_{\ell=1}^{L} A_{\ell m_1 n_1} B_{\ell m_2 n_2} \]

\[ m_1 = 1, \ldots, M_1 \quad n_1 = 1, \ldots, N_1 \]
\[ m_2 = 1, \ldots, M_2 \quad n_2 = 1, \ldots, N_2 \]
Tensor-tensor Multiplication

Using contracted product...

- Set of all third-order tensors is not closed
- No notion of inverse possible

What happens if we create an operation that is closed under “multiplication”?
New tensor-tensor operation

\[ \mathcal{A} \in \mathbb{R}^{L \times M \times N} \quad \mathcal{B} \in \mathbb{R}^{M \times P \times N} \quad \Rightarrow \quad \mathcal{A} \ast \mathcal{B} \in \mathbb{R}^{L \times P \times N} \]

- Operation defined in terms of the tensor “slices”
- Circulant matrices play a role
- Operation is associative
- Can define an inverse
- Set of \( N \times N \times N \) invertible tensors form a group under this operation
New tensor-tensor operation

\[ A \ast B = C \]

\[
\begin{bmatrix}
A_1 & A_3 & A_2 \\
A_2 & A_1 & A_3 \\
A_3 & A_2 & A_1 \\
\end{bmatrix}
\ast
\begin{bmatrix}
B_1 \\
B_2 \\
B_3 \\
\end{bmatrix}
= 
\begin{bmatrix}
A_1 B_1 + A_3 B_2 + A_2 B_3 \\
A_2 B_1 + A_1 B_2 + A_3 B_3 \\
A_3 B_1 + A_2 B_2 + A_1 B_3 \\
\end{bmatrix}
= 
\begin{bmatrix}
C(\cdot, \cdot, 1) \\
C(\cdot, \cdot, 2) \\
C(\cdot, \cdot, 3) \\
\end{bmatrix}
\]
More efficient if performed in the Fourier domain. For example, if $A \in \mathbb{R}^{L \times M \times 4}$, $B \in \mathbb{R}^{M \times P \times 4}$:

$$C = A \ast B$$

$$= (F_4^* \otimes I_L)(F_4 \otimes I_L) \begin{bmatrix} A_1 & A_4 & A_3 & A_2 \\ A_2 & A_1 & A_4 & A_3 \\ A_3 & A_2 & A_1 & A_4 \\ A_4 & A_3 & A_2 & A_1 \end{bmatrix} (F_4^* \otimes I_M)(F_4 \otimes I_M) \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}$$

$$= (F_4^* \otimes I_L) \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \end{bmatrix} \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \\ \tilde{B}_3 \\ \tilde{B}_4 \end{bmatrix}$$
Higher Order Tensor Operations (recursive)

Suppose $A, B \in \mathbb{R}^{3 \times 3 \times 3 \times 3}$ ($A_{ij} \in \mathbb{R}^{3 \times 3}$)

$A_1 = A(:, :, :, 1) = \begin{bmatrix} A_{11} \\ A_{12} \\ A_{13} \end{bmatrix}$

$B_1 = B(:, :, :, 1) = \begin{bmatrix} B_{11} \\ B_{12} \\ B_{13} \end{bmatrix}$

$A_2 = A(:, :, :, 2) = \begin{bmatrix} A_{21} \\ A_{22} \\ A_{23} \end{bmatrix}$

$B_2 = B(:, :, :, 2) = \begin{bmatrix} B_{21} \\ B_{22} \\ B_{23} \end{bmatrix}$

$A_3 = A(:, :, :, 3) = \begin{bmatrix} A_{31} \\ A_{32} \\ A_{33} \end{bmatrix}$

$B_3 = B(:, :, :, 3) = \begin{bmatrix} B_{31} \\ B_{32} \\ B_{33} \end{bmatrix}$
Higher Order Tensor Operations (recursive)

Then, $\mathcal{A} \ast \mathcal{B}$:

$$
\begin{bmatrix}
\mathcal{A}_1 & \mathcal{A}_3 & \mathcal{A}_2 \\
\mathcal{A}_2 & \mathcal{A}_1 & \mathcal{A}_3 \\
\mathcal{A}_3 & \mathcal{A}_2 & \mathcal{A}_1
\end{bmatrix} \ast
\begin{bmatrix}
\mathcal{B}_1 \\
\mathcal{B}_2 \\
\mathcal{B}_3
\end{bmatrix}
$$

9x9x3 9x3x3

Carla Martin - martincd@jmu.edu (JMU)
Higher Order Tensor Operations (recursive)

Matrix multiply → Leads to a recursive algorithm
Let $C \in \mathbb{R}^{L \times M \times P}$ with faces $C_1, \ldots, C_P \in \mathbb{R}^{L \times M}$. Then

\[ (B \ast C)^T = C^T \ast B^T \]

It follows that $(B \ast C)^T = C^T \ast B^T$
Higher Order Tensor Transpose (recursive)

The higher order tensor transpose follows a recursive process.
The $N \times N \times P$ identity tensor, $\mathcal{I}$, is the tensor whose frontal face is the $N \times N$ identity matrix and whose other faces are zeros.

In general, $\mathcal{A} \ast \mathcal{I} = \mathcal{I} \ast \mathcal{A} = \mathcal{A}$
Inverse

Let $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$. Then the tensor inverse of $\mathcal{A}$ is any tensor $\mathcal{B} \in \mathbb{R}^{N \times N \times N}$ such that

$$\mathcal{A} \star \mathcal{B} = \mathcal{B} \star \mathcal{A} = \mathcal{I}$$

We denote the inverse of $\mathcal{A}$ as $\mathcal{A}^{-1}$.

It follows that $(\mathcal{A} \star \mathcal{B})^{-1} = \mathcal{B}^{-1} \star \mathcal{A}^{-1}$
Frobenius Norm and Orthogonality

Let $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{L \times M \times N}$. Then the Frobenius norm of $\mathcal{A}$ is

$$
\|\mathcal{A}\|_F = \sqrt{\sum_{i=1}^{L} \sum_{j=1}^{M} \sum_{k=1}^{N} a_{ijk}^2}
$$

Let $Q \in \mathbb{R}^{N \times N \times P}$. $Q$ is orthogonal if

$$
Q^T \ast Q = Q \ast Q^T = I
$$

If $\mathcal{A}$ is a tensor, then it follows that

$$
\|Q \ast \mathcal{A}\|_F = \|\mathcal{A}\|_F
$$
Tensor SVD

Let $\mathcal{A} \in \mathbb{R}^{L \times M \times N}$. Then $\mathcal{A}$ can be factored as

$$\mathcal{A} = \mathcal{U} \ast \mathcal{S} \ast \mathcal{V}^T$$

where $\mathcal{U} \in \mathbb{R}^{L \times L \times N}$ and $\mathcal{V} \in \mathbb{R}^{M \times M \times N}$ are orthogonal tensors and $\mathcal{S} \in \mathbb{R}^{L \times M \times N}$ has diagonal matrix faces.

If $\mathcal{A} \in \mathbb{R}^{N \times N \times N}$,

$$\mathcal{A} = \sum_{i=1}^{N} \mathcal{U}(; , i , :) \ast \mathcal{S}(i , i , :) \ast \mathcal{V}(; , i , :)^T$$
Tensor SVD: computation

- Computation of the tensor SVD involves SVDs of block diagonal elements obtained from block diagonalizing the circulant matrix generated by $A$.

- Using the SVDs of the blocks leads to algorithms for compression.

- Decomposition extends recursively to order-$p$ tensors when $p > 3$. 

\[ A = U \ast S \ast V^T \]
Tensor SVD: computation

\[
\begin{bmatrix}
A_1 & A_4 & A_3 & A_2 \\
A_2 & A_1 & A_4 & A_3 \\
A_3 & A_2 & A_1 & A_4 \\
A_4 & A_3 & A_2 & A_1 \\
\end{bmatrix}
= (F \otimes I)
\begin{bmatrix}
D_1 & & & \\
& D_2 & & \\
& & D_3 & \\
& & & D_4 \\
\end{bmatrix}
(F^* \otimes I)
\]

\[
= (F \otimes I)
\begin{bmatrix}
U_1 \Sigma_1 V_1^T & & & \\
& U_2 \Sigma_2 V_2^T & & \\
& & U_3 \Sigma_3 V_3^T & \\
& & & U_4 \Sigma_4 V_4^T \\
\end{bmatrix}
(F^* \otimes I)
\]

\[
= (F \otimes I)
\begin{bmatrix}
U_1 & U_2 & U_3 & U_4 \\
& & & \\
& & & \\
& & & \\
\end{bmatrix}
\begin{bmatrix}
\Sigma_1 & & & \\
& \Sigma_2 & & \\
& & \Sigma_3 & \\
& & & \Sigma_4 \\
\end{bmatrix}
\begin{bmatrix}
V_1^T & & & \\
& V_2^T & & \\
& & V_3^T & \\
& & & V_4^T \\
\end{bmatrix}
(F^* \otimes I)
\]
Tensor SVD: computation

\[
= (F^* \otimes I_m) \begin{bmatrix}
    U_1 & \cdots & U_p
\end{bmatrix}\]
\[
\otimes
\]
\[
= (F \otimes I_m) (F^* \otimes I_n) \begin{bmatrix}
    \Sigma_1 & \cdots & \Sigma_p
\end{bmatrix}\]
\[
\otimes
\]
\[
= (F \otimes I_n) \begin{bmatrix}
    V_1^T & \cdots & V_p^T
\end{bmatrix}
\]

\[
= \hat{U}_1 \quad \hat{U}_p \quad \cdots \quad \hat{U}_2
\]
\[
\hat{U}_2 \quad \hat{U}_1 \quad \cdots \quad \hat{U}_3
\]
\[
\vdots \quad \vdots \quad \vdots \quad \vdots
\]
\[
\hat{U}_p \quad \hat{U}_{p-1} \quad \cdots \quad \hat{U}_1
\]
\[
\hat{S}_1 \quad \hat{S}_p \quad \cdots \quad \hat{S}_2
\]
\[
\hat{S}_2 \quad \hat{S}_1 \quad \cdots \quad \hat{S}_3
\]
\[
\vdots \quad \vdots \quad \vdots \quad \vdots
\]
\[
\hat{S}_p \quad \hat{S}_{p-1} \quad \cdots \quad \hat{S}_1
\]
\[
\hat{V}_1^T \quad \hat{V}_p^T \quad \cdots \quad \hat{V}_2^T
\]
\[
\hat{V}_2^T \quad \hat{V}_1^T \quad \cdots \quad \hat{V}_3^T
\]
\[
\vdots \quad \vdots \quad \vdots \quad \vdots
\]
\[
\hat{V}_p^T \quad \hat{V}_{p-1}^T \quad \cdots \quad \hat{V}_1^T
\]

\[
= U \ast S \ast V^T
\]
One Compression Strategy

Suppose $A \in \mathbb{R}^{L \times M \times N}$

Can prove that if $A = U \ast S \ast V^T$ then

$$
\sum_{i=1}^{N} U(:, :, i), \quad \sum_{i=1}^{N} V(:, :, i)
$$

are orthogonal.

Therefore

$$
\sum_{i=1}^{N} A(:, :, i) = \left( \sum_{i=1}^{N} U(:, :, i) \right) \left( \sum_{i=1}^{N} S(:, :, i) \right) \left( \sum_{i=1}^{N} V(:, :, i) \right)^T
$$
One Compression Strategy

\[ \sum_{i=1}^{N} A(\cdot, \cdot, i) = \left( \sum_{i=1}^{N} U(\cdot, \cdot, i) \right) \left( \sum_{i=1}^{N} S(\cdot, \cdot, i) \right) \left( \sum_{i=1}^{N} V(\cdot, \cdot, i) \right)^T \]

- Choose \( k_1 << L \), \( k_2 << M \) and compute truncated SVD, \( \tilde{U} \tilde{S} \tilde{V}^T \)
- Set \( T(\cdot, \cdot, i) = \tilde{U}^T A(\cdot, \cdot, i) \tilde{V} \) for \( i = 1, \ldots, N \)
- Can rewrite “compressed” tensor \( A_c \) as sum of outer products:

\[ A \approx A_c = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \tilde{U}(\cdot, i) \circ \tilde{V}(\cdot, j) \circ T(i, j, :) \]

- Computationally, do not need to compute Tensor SVD to obtain representation above
New Tensor SVD Factorization

Advantages:

- Orientation Specific
- Allows for weighting of the tensor slices according to data
- Emits a factorization with an underlying group structure that easily extends other matrix factorizations to tensors

Disadvantages:

- Orientation Specific
- Does not seem to work for many applications in chemometrics where specific orientation is not required
- Randomly generated tensors, compression (measured in norm) is similar to HOSVD
Suppose $A \in \mathbb{R}^{N \times N \times N}$. Then $A$ can be factored as

\[ A = Q \ast R \]

$Q$ orthogonal, $R$ with upper triangular faces

\[ A = Q \ast B \ast Q^T \]

$Q$ orthogonal, $B$ with upper triangular faces
Open Questions

- Does a group structure give us an advantage in terms of applications or theoretical multilinear algebra?
- Are non-SVD extensions another area in which to investigate?
- Does this have a use when tensors are sparse or otherwise structured?

Thank you!