A New Factorization of an Order-p Tensor as a Product of Order-p Tensors

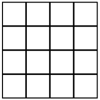
Carla D. Martin
(with M. Kilmer, L. Perrone)

James Madison University

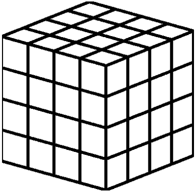
SIAM CS&E Conference 2009
March 4, 2009

What are Tensors?

- Second-order tensor $A = (a_{ij}) \in \mathbb{R}^{n_1 \times n_2}$



- Third-order tensor $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$



- p^{th}-order tensor $\mathcal{A} = (a_{i_1 i_2 \ldots i_p}) \in \mathbb{R}^{n_1 \times \cdots \times n_p}$

Motivation: A two-way decomposition

Suppose $U \in \mathbb{R}^{M\times M}$, $V \in \mathbb{R}^{N\times N}$ are orthogonal, and $\Sigma = U^T A V$, then

$$A = U\Sigma V^T = \sum_{i=1}^{M}\sum_{j=1}^{N} \sigma_{ij} u_i v_j^T$$

$$= \sum_{i=1}^{M}\sum_{j=1}^{N} \sigma_{ij}(u_i \circ v_j)$$

where $u_i = U(:,i)$, $v_j = V(:,j)$

Tensor Decompositions

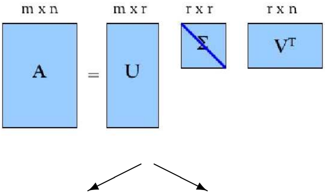
Let $\mathcal{A} \in \mathbb{R}^{M \times N \times P}$

Goal: To find $U \in \mathbb{R}^{M \times M}$, $V \in \mathbb{R}^{N \times N}$, $W \in \mathbb{R}^{P \times P}$, and $\Sigma = (\sigma_{ijk}) \in \mathbb{R}^{M \times N \times P}$ such that

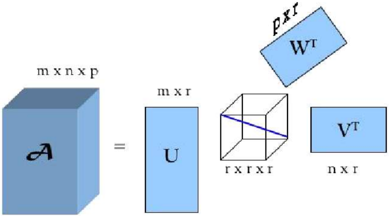
$$\mathcal{A} = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{P} \sigma_{ijk} (u_i \circ v_j \circ w_k)$$

where $u_i = U(:, i)$, $v_j = V(:, j)$, $w_k = W(:, k)$

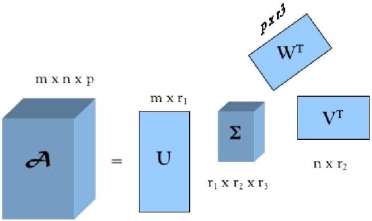
Orthogonal *or* Diagonal for Tensors



Case 1:
Diagonal Σ



Case 2:
Orthogonal U, V, W



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 *first-mode*:

$$\begin{array}{l} \mathcal{A} \in \mathbb{R}^{L \times M_1 \times N_1} \\ \mathcal{B} \in \mathbb{R}^{L \times M_2 \times N_2} \end{array} \quad \Rightarrow \quad \mathcal{AB} \in \mathbb{R}^{M_1 \times N_1 \times M_2 \times N_2}$$

$$\boxed{(\mathcal{AB})_{m_1 n_1 m_2 n_2} = \sum_{\ell=1}^{L} \mathcal{A}_{\ell m_1 n_1} \mathcal{B}_{\ell m_2 n_2}}$$

$$\begin{aligned} m_1 &= 1, \ldots, M_1 & \qquad n_1 &= 1, \ldots, N_1 \\ m_2 &= 1, \ldots, M_2 & \qquad n_2 &= 1, \ldots, N_2 \end{aligned}$$

Tensor-tensor Multiplication

Using contracted product...

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

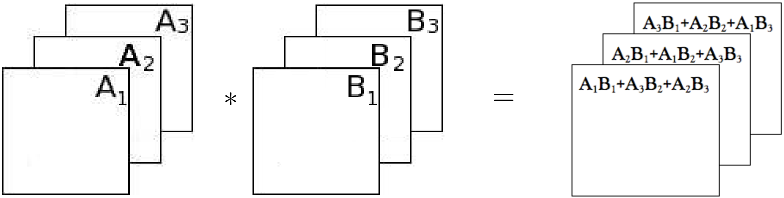
What happens if we create an operation that is closed under "multiplication"?

New tensor-tensor operation

$$\begin{array}{l} \mathcal{A} \in \mathbb{R}^{L\times M\times N} \\ \mathcal{B} \in \mathbb{R}^{M\times P\times N} \end{array} \quad \Rightarrow \quad \mathcal{A} * \mathcal{B} \in \mathbb{R}^{L\times P\times N}$$

- 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



$$\mathcal{A} \quad * \quad \mathcal{B} \quad = \quad \mathcal{C}$$

$$\begin{bmatrix} A_1 & A_3 & A_2 \\ A_2 & A_1 & A_3 \\ A_3 & A_2 & A_1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} A_1 B_1 + A_3 B_2 + A_2 B_3 \\ A_2 B_1 + A_1 B_2 + A_3 B_3 \\ A_3 B_1 + A_2 B_2 + A_1 B_3 \end{bmatrix} = \begin{bmatrix} \mathcal{C}(:,:,1) \\ \mathcal{C}(:,:,2) \\ \mathcal{C}(:,:,3) \end{bmatrix}$$

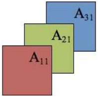
Computation

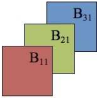
More efficient if performed in the Fourier domain.
For example, if $\mathcal{A} \in \mathbb{R}^{L \times M \times 4}$, $\mathcal{B} \in \mathbb{R}^{M \times P \times 4}$:

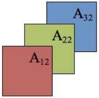
$$
\begin{aligned}
\mathcal{C} &= \mathcal{A} * \mathcal{B} \\
&= (F_4^* \otimes I_L)(F_4 \otimes I_L) \begin{bmatrix} A_1 & A_4 & A_3 & A_2 \\ A_2 & A_1 & A_4 & A_3 \\ A_3 & A_2 & A_1 & A_4 \\ A_4 & A_3 & A_2 & A_1 \end{bmatrix} (F_4^* \otimes I_M)(F_4 \otimes I_M) \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} \\
&= (F_4^* \otimes I_L) \begin{bmatrix} D_1 & & & \\ & D_2 & & \\ & & D_3 & \\ & & & D_4 \end{bmatrix} \begin{bmatrix} \tilde{B}_1 \\ \tilde{B}_2 \\ \tilde{B}_3 \\ \tilde{B}_4 \end{bmatrix}
\end{aligned}
$$

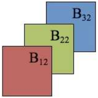
Higher Order Tensor Operations (recursive)

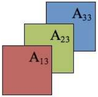
Suppose $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{3x3x3x3}$ $(A_{ij} \in \mathbb{R}^{3x3})$

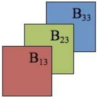
$\mathcal{A}_1 = \mathcal{A}(:,:,:,1) =$ 

$\mathcal{B}_1 = \mathcal{B}(:,:,:,1) =$ 

$\mathcal{A}_2 = \mathcal{A}(:,:,:,2) =$ 

$\mathcal{B}_2 = \mathcal{B}(:,:,:,2) =$ 

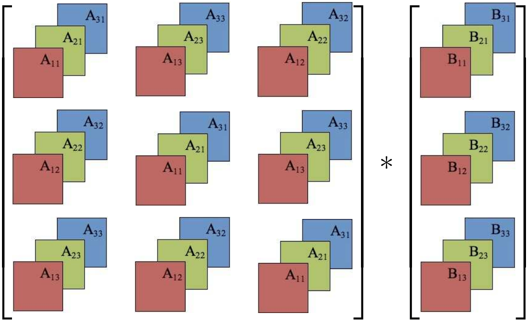
$\mathcal{A}_3 = \mathcal{A}(:,:,:,3) =$ 

$\mathcal{B}_3 = \mathcal{B}(:,:,:,3) =$ 

Higher Order Tensor Operations (recursive)

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

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



9x9x3 9x3x3

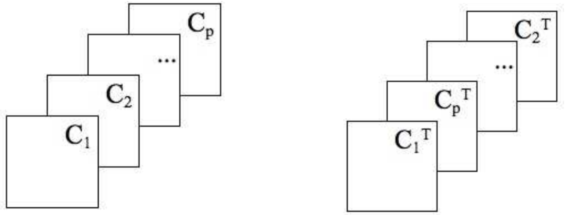
Higher Order Tensor Operations (recursive)

$$
\left[\begin{array}{ccc|ccc|ccc}
A_{11} & A_{13} & A_{12} & A_{31} & A_{33} & A_{32} & A_{21} & A_{23} & A_{22} \\
A_{12} & A_{11} & A_{13} & A_{32} & A_{31} & A_{33} & A_{22} & A_{21} & A_{23} \\
A_{13} & A_{12} & A_{11} & A_{33} & A_{32} & A_{31} & A_{23} & A_{22} & A_{21} \\
\hline
A_{21} & A_{23} & A_{22} & A_{11} & A_{13} & A_{12} & A_{31} & A_{33} & A_{32} \\
A_{22} & A_{21} & A_{23} & A_{12} & A_{11} & A_{13} & A_{32} & A_{31} & A_{33} \\
A_{23} & A_{22} & A_{21} & A_{13} & A_{12} & A_{11} & A_{33} & A_{32} & A_{31} \\
\hline
A_{31} & A_{33} & A_{32} & A_{21} & A_{23} & A_{22} & A_{11} & A_{13} & A_{12} \\
A_{32} & A_{31} & A_{33} & A_{22} & A_{21} & A_{23} & A_{12} & A_{11} & A_{13} \\
A_{33} & A_{32} & A_{31} & A_{23} & A_{22} & A_{21} & A_{13} & A_{12} & A_{11}
\end{array}\right]
\left[\begin{array}{c}
B_{11} \\ B_{21} \\ B_{31} \\ \hline B_{12} \\ B_{22} \\ B_{32} \\ \hline B_{13} \\ B_{23} \\ B_{33}
\end{array}\right]
$$

Matrix multiply $\longrightarrow$ Leads to a recursive algorithm

Transpose

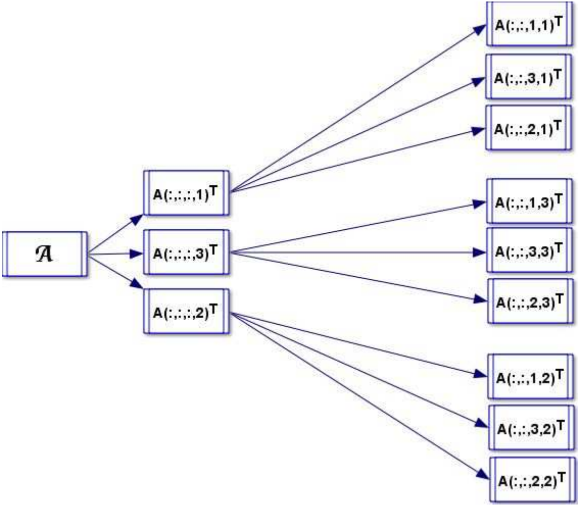
Let $\mathcal{C} \in \mathbb{R}^{L \times M \times P}$ with faces $C_1, \ldots, C_P \in \mathbb{R}^{L \times M}$. Then



It follows that $(\mathcal{B} * \mathcal{C})^T = \mathcal{C}^T * \mathcal{B}^T$

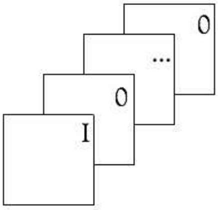
Higher Order Tensor Transpose (recursive)

The higher order tensor transpose follows a recursive process.



Identity

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, $\mathcal{A} * \mathcal{I} = \mathcal{I} * \mathcal{A} = \mathcal{A}$

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

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

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

It follows that $(\mathcal{A} * \mathcal{B})^{-1} = \mathcal{B}^{-1} * \mathcal{A}^{-1}$

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

$$\mathcal{Q}^T * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^T = \mathcal{I}$$

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

$$||\mathcal{Q} * \mathcal{A}||_F = ||\mathcal{A}||_F$$

Tensor SVD

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

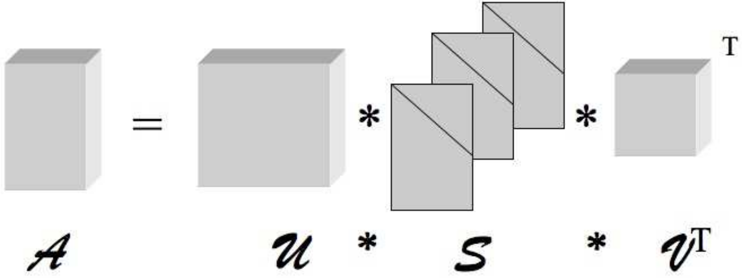
$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$$

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

$$\mathcal{A} = \sum_{i=1}^{N} \mathcal{U}(:, i, :) * \mathcal{S}(i, i, :) * \mathcal{V}(:, i, :)^T$$

Tensor SVD: computation



- Computation of the tensor SVD involves SVDs of block diagonal elements obtained from block diagonalizing the circulant matrix generated by $\mathcal{A}$
- Using the SVDs of the blocks leads to algorithms for compression
- Decomposition extends recursively to order-p tensors when $p > 3$

Tensor SVD: computation

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

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

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

Compression Strategy

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

Can prove that if $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ then

$$\sum_{i=1}^{N} \mathcal{U}(:,:,i), \quad \sum_{i=1}^{N} \mathcal{V}(:,:,i) \qquad \text{are orthogonal.}$$

Therefore

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

Compression Strategy

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

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

$$\mathcal{A} \approx \mathcal{A}_c = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \tilde{U}(:,i) \circ \tilde{V}(:,j) \circ \mathcal{T}(i,j,:)$$

- 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
- Randomly generated tensors, compression (measured in norm) is similar to existing factorizations

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
- Randomly generated tensors, compression (measured in norm) is similar to existing factorizations

Weighting

Suppose $\mathcal{A}$ is an $L \times M \times N$ tensor

Standard Compression Strategy:

$$A = \sum_{i=1}^{N} \mathcal{A}(:,:,i)$$

Weighted Compression Strategy:

$$A = \sum_{i=1}^{N} w(i)\mathcal{A}(:,:,i)$$

where w is a weighting vector

Used when there is a known dependence on the tensor slices

QR and Eigenvalue Extensions

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

$\swarrow \qquad \searrow$

$$\mathcal{A} = \mathcal{Q} * \mathcal{R}$$

$\mathcal{Q}$ orthogonal, $\mathcal{R}$ with upper triangular faces

$$\mathcal{A} = \mathcal{Q} * \mathcal{B} * \mathcal{Q}^T$$

$\mathcal{Q}$ orthogonal, $\mathcal{B}$ with upper triangular faces

Summary

- New definitions allow for greater flexibility in design of compression algorithms for tensors
- Can be extended to other matrix factorizations (e.g., QR)
- Computationally efficient
- Allows for weighting of tensor faces
- Underlying algebraic structure within the factorization

Thank you!