CS 598 EVS: Tensor Computations
Basics of Tensor Computations

Edgar Solomonik

University of Illinois at Urbana-Champaign
Tensors

A tensor is a collection of elements

A few examples of tensors are
Reshaping Tensors

It's often helpful to use alternative views of the same collection of elements.
Matrices and Tensors as Operators and Multilinear Forms

- What is a matrix?

- What is a tensor?
Tensor Transposition

For tensors of order $\geq 3$, there is more than one way to transpose modes
Tensor Symmetry

We say a tensor is **symmetric** if $\forall j, k \in \{1, \ldots, d\}$

A tensor is **antisymmetric** (skew-symmetric) if $\forall j, k \in \{1, \ldots, d\}$

A tensor is **partially-symmetric** if such index interchanges are restricted to be within disjoint subsets of $\{1, \ldots, d\}$, e.g., if the subsets for $d = 4$ and $\{1, 2\}$ and $\{3, 4\}$, then
Tensor Sparsity

We say a tensor $\mathcal{T}$ is *diagonal* if for some $v$, If most of the tensor entries are zeros, the tensor is *sparse*.
Tensor products can be defined with respect to maps $f : V_f \to W_f$ and $g : V_g \to W_g$.

Tensors can be used to represent multilinear maps and have a corresponding definition for a tensor product.

The Kronecker product between two matrices $A \in \mathbb{R}^{m_1 \times m_2}, B \in \mathbb{R}^{n_1 \times n_2}$ is defined as:

$$A \otimes B = C$$

with

$$b_{ij} a_{kl} = c_{i+k n_1, j+l n_2}$$
$A \otimes B =$

$$
\begin{bmatrix}
  a_{11} B & a_{12} B & \cdots \\
  a_{21} B & & \\
  & & \\
  & & \\
  & & \\
  & & \\
  & & \\
  & & \\
\end{bmatrix}
$$

$T \in \mathbb{R}^{d \times d}$

$t \odot x \rightarrow T \left[ e^{t \langle x \mid \hat{a} \rangle} + \int_{-\infty}^{-\infty} e^{t \langle x \mid \hat{a} \rangle + \frac{1}{2} s \langle \hat{a} \mid \hat{a} \rangle} \right]$
# Tensor Contractions

A *tensor contraction* multiplies elements of two tensors and computes partial sums to produce a third, in a fashion expressible by pairing up modes of different tensors, defining *einsum* (term stems from Einstein’s summation convention).

<table>
<thead>
<tr>
<th>tensor contraction</th>
<th>einsum</th>
<th>diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>inner product</td>
<td>$c_{ij} = a_{ij}b_{ij}$</td>
<td><img src="image" alt="inner_product" /></td>
</tr>
<tr>
<td>outer product</td>
<td>$c_{ijk} = a_{ijk}b_{ijk}$</td>
<td><img src="image" alt="outer_product" /></td>
</tr>
<tr>
<td>pointwise product</td>
<td>$c_{ij} = a_{ij}b_{ij}$</td>
<td><img src="image" alt="pointwise_product" /></td>
</tr>
<tr>
<td>Hadamard product</td>
<td>$c_{ij} = a_{ij}b_{ij}$</td>
<td><img src="image" alt="Hadamard_product" /></td>
</tr>
<tr>
<td>matrix multiplication</td>
<td>$c_{ijk} = a_{ijk}b_{ijk}$</td>
<td><img src="image" alt="matrix_multiplication" /></td>
</tr>
<tr>
<td>batched mat.-mul.</td>
<td>$c_{ijk} = a_{ik}b_{kj}$</td>
<td><img src="image" alt="batched_mat.-mul." /></td>
</tr>
<tr>
<td>tensor times matrix</td>
<td>$c_{ijkl} = a_{ijkl}b_{ijkl}$</td>
<td><img src="image" alt="tensor_times_matrix" /></td>
</tr>
</tbody>
</table>

The terms ‘contraction’ and ‘einsum’ are also often used when more than two operands are involved.
General Tensor Contractions

Given tensor $\mathbf{U}$ of order $s + v$ and $\mathbf{V}$ of order $v + t$, a tensor contraction summing over $v$ modes can be written as

$\mathbf{U} = \sum_{i} \mathbf{U}^{i} \otimes \mathbf{V}^{i}$

Unfolding the tensors reduces the tensor contraction to matrix multiplication

$\mathbf{W} = \mathbf{U} \mathbf{V}$
Properties of Einsums

Given an elementwise expression containing a product of tensors, the operands commute

\[ AB = BA \]

A contraction can be succinctly described by a tensor diagram

for \( i \) for \( k \) for \( i' \) for \( j' \)

\[ d_{ij} = d_{ij} + a_{ik}b_{kj} \]
Matrix-style Notation for Tensor Contractions

The **tensor times matrix** contraction along the $m$th mode of $\mathbf{U}$ to produce $\mathbf{V}$ is expressed as follows

\[
\mathbf{V} = \mathbf{U} \times_k \mathbf{M}
\]

The **Khatri-Rao product** of two matrices $\mathbf{U} \in \mathbb{R}^{m \times k}$ and $\mathbf{V} \in \mathbb{R}^{n \times k}$ products $\mathbf{W} \in \mathbb{R}^{mn \times k}$ so that

\[
\begin{bmatrix} u_1 & \cdots & u_k \end{bmatrix} \odot \begin{bmatrix} v_1 & \cdots & v_k \end{bmatrix} = \begin{bmatrix} u_1 \odot v_1 & u_2 \odot v_2 & \cdots & u_k \odot v_k \end{bmatrix}
\]
Identities with Kronecker and Khatri-Rao Products

- Matrix multiplication is distributive over the Kronecker product

\[
(A \otimes B)(C \otimes D) = (AC) \otimes (BD)
\]

\[
O(n^6)
\]

\[
O(n^4)
\]

- For the Khatri-Rao product a similar distributive identity is

\[
(A \odot B)^T(C \odot D) = (A^T C) \otimes (B^T D)
\]

\[
\begin{align*}
M &= A^T C \\
N &= B^T D
\end{align*}
\]

\[
M \odot N
\]
\[ \sum \sum_{i,j} \sum_{k,l} \sum_{m,n} a_{i,k} b_{j,l} c_{k,m} d_{l,n} e_{m,n} = w_{k,l} \]

\[
\left( \sum_i a_{i,k} c_{i,k} \right) \left( \sum_j b_{j,l} d_{j,l} \right) = (A^T C) \cdot (B^T D) \]
Multilinear Tensor Operations

Given an order $d$ tensor $\mathcal{T}$, define multilinear function $x^{(1)} = f(\mathcal{T})(x^{(2)}, \ldots, x^{(d)})$

$$x^{(1)} = f(\mathcal{T})(x^{(2)}, \ldots, x^{(d)})$$

$$f^{(m)} \in \mathbb{R}^n \times \ldots \times \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$x_i^{(1)} = \sum_{i_2, \ldots, i_d} T_{i, i_2, \ldots, i_d} x_i^{(2)} \ldots x_i^{(d)}$$
Batched Multilinear Operations

The multilinear map $f(T)$ is frequently used in tensor computations.

$$M^{T}KRP \text{ - matricized tensor Khatri-Rao product}$$

$$A^{(n)} = T^{(n)} (A^{(1)} \odot \ldots \odot A^{(n-1)} \odot A^{(n+1)} \odot \ldots \odot A^{(r)})$$

$$\Phi = \begin{bmatrix} C \end{bmatrix}$$

$$a_{ir} = T_{ijk} b_{jr} c_{kr}$$

$$A = T^{(n)} (B \odot C)$$

$$\begin{bmatrix} a^{(1)} & \ldots & a^{(r)} \end{bmatrix} = \begin{bmatrix} f_{T}(b^{(1)} c^{(1)}) & \ldots & f_{T}(b^{(2)} c^{(2)}) \end{bmatrix}$$
TTMC

tensor - linear - matrix chain

\[ Z = T x_1 A^{(1)} x_2 A^{(2)} \cdots x_{d-1} A^{(d-1)} \]

\[ A^{(i)} = \sum_{a^{(i)}} a^{(i)} \]

\[ Z^{(i_1 \cdots i_d)} = \sum_{a^{(i_1 \cdots i_d)}} \alpha^{(i_1 \cdots i_d)} \]

\[ 2 \left< l_1, \ldots, l_{d-1} \right> = f^T (a^{(i_1 \cdots i_d)}) \]
Tensor Norm and Conditioning of Multilinear Functions

We can define elementwise and operator norms for a tensor $\mathbf{T}$.
Conditioning of Multilinear Functions

Evaluation of the multilinear map is typically ill-posed for worst case inputs
Well-conditioned Tensors

For equidimensional tensors (all modes of same size), some small ideally conditioned tensors exist.
Ill-conditioned Tensors

For \( n \notin \{2, 4, 8\} \) given any \( T \in \mathbb{R}^{n \times n \times n} \), \( \inf_{x, y \in S^{n-1}} \| f(T)(x, y) \|_2 = 0 \)
The canonical polyadic or CANDECOMP/PARAFAC (CP) decomposition expresses an order $d$ tensor in terms of $d$ factor matrices.
CP Decomposition Basics

- The CP decomposition is useful in a variety of contexts

- Basic properties and methods
The *Tucker decomposition* expresses an order $d$ tensor via a smaller order $d$ core tensor and $d$ factor matrices.
Tucker Decomposition Basics

- The Tucker decomposition is used in many of the same contexts as CP

- Basic properties and methods
Tensor Train Decomposition

- The *tensor train decomposition* expresses an order $d$ tensor as a chain of products of order 2 or order 3 tensors
Tensor Train Decomposition Basics

- Tensor train has applications in quantum simulation and in numerical PDEs

- Basic properties and methods
We can compare the aforementioned decomposition for an order $d$ tensor with all dimensions equal to $n$ and all decomposition ranks equal to $R$

<table>
<thead>
<tr>
<th>Decomposition</th>
<th>CP</th>
<th>Tucker</th>
<th>Tensor Train</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Uniqueness</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Orthogonalizability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exact Decomposition</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Approximation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Typical Method</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bilinear Algorithms

A bilinear algorithm (V. Pan, 1984) $\Lambda = (F^{(A)}, F^{(B)}, F^{(C)})$ computes

$$c = F^{(C)}[(F^{(A)^T} a) \odot (F^{(B)^T} b)],$$

where $a$ and $b$ are inputs and $\odot$ is the Hadamard (pointwise) product.
Bilinear Algorithms as Tensor Factorizations

- A bilinear algorithm corresponds to a CP tensor decomposition

- For multiplication of $n \times n$ matrices, we can define a matrix multiplication tensor and consider algorithms with various bilinear rank
Strassen’s Algorithm

Strassen’s algorithm \[
\begin{bmatrix}
C_{11} & C_{12} \\
C_{21} & C_{22}
\end{bmatrix} = \begin{bmatrix}
A_{11} & A_{12} \\
A_{21} & A_{22}
\end{bmatrix} \cdot \begin{bmatrix}
B_{11} & B_{12} \\
B_{21} & B_{22}
\end{bmatrix}
\]

\[
M_1 = (A_{11} + A_{22}) \cdot (B_{11} + B_{22})
\]
\[
M_2 = (A_{21} + A_{22}) \cdot B_{11}
\]
\[
M_3 = A_{11} \cdot (B_{12} - B_{22})
\]
\[
M_4 = A_{22} \cdot (B_{21} - B_{11})
\]
\[
M_5 = (A_{11} + A_{12}) \cdot B_{22}
\]
\[
M_6 = (A_{21} - A_{11}) \cdot (B_{11} + B_{12})
\]
\[
M_7 = (A_{12} - A_{22}) \cdot (B_{21} + B_{22})
\]

\[
C_{11} = M_1 + M_4 - M_5 + M_7
\]
\[
C_{21} = M_2 + M_4
\]
\[
C_{12} = M_3 + M_5
\]
\[
C_{22} = M_1 - M_2 + M_3 + M_6
\]

By performing the nested calls recursively, Strassen’s algorithm achieves cost,