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

Its 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 $\geqslant 3$, 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 and Kronecker Products

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 $\boldsymbol{A} \in \mathbb{R}^{m_1 \times m_2}$, $\boldsymbol{B} \in \mathbb{R}^{n_1 \times n_2}$

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)

tensor contraction	*einsum*	*diagram*
inner product		
outer product		
pointwise product		
Hadamard product		
matrix multiplication		
batched mat.-mul.		
tensor times matrix		

The terms 'contraction' and 'einsum' are also often used when more than two operands are involved

General Tensor Contractions

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

Unfolding the tensors reduces the tensor contraction to matrix multiplication

Properties of Einsums

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

A contraction can be succinctly described by a *tensor diagram*

Matrix-style Notation for Tensor Contractions

The *tensor times matrix* contraction along the mth mode of $\boldsymbol{\mathcal{U}}$ to produce $\boldsymbol{\mathcal{V}}$ is expressed as follows

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

Identities with Kronecker and Khatri-Rao Products

- Matrix multiplication is distributive over the Kronecker product

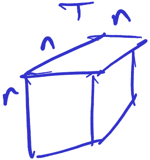
- For the Khatri-Rao product a similar distributive identity is

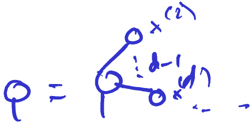
Multilinear Tensor Operations

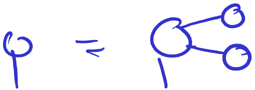
Given an order d tensor $\mathcal{T}$, define multilinear function $\boldsymbol{x}^{(1)} = \boldsymbol{f}^{(\mathcal{T})}(\boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(d)})$

order N

N-1





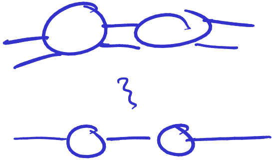


$$x_i = \sum_{jk} t_{ijk} y_j z_k$$

$$x_j = \sum_{ik} t_{ijk} y_i z_k$$

$$x = f^{(t)}(y, z)$$

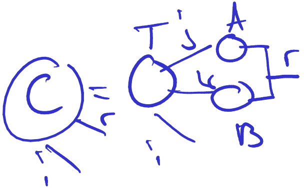
$$f^{(t)}: R^n \times R^n \to R^n$$



Batched Multilinear Operations

The multilinear map $f^{(\mathcal{T})}$ is frequently used in tensor computations

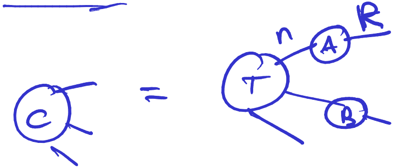
MTTKRP → CP decomp.



$$c_{ir} = \sum_{jk} t_{ijk} a_{jr} b_{kr}$$

$$c^{(r)} = f^{(T)}(a^{(r)}, b^{(r)})$$

TTMc → Tucker decomp.



$$c_{irs} = \sum_{jk} t_{ijk} a_{jr} b_{ks}$$

$$c^{(r,s)} = f^{(T)}(a^{(r)}, b^{(s)})$$

Tensor Norm and Conditioning of Multilinear Functions

We can define elementwise and operator norms for a tensor $\mathcal{T}$

$$\|T\|_F = \Big(\sum_{ijk} t_{ijk}^2\Big)^{1/2}$$

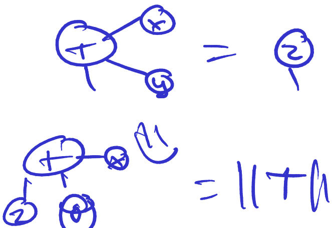
$$\|M\|_2 = \max_{x \in S^{n-1}} \|Ax\|_2$$

$$\{x : \|x\|_2 = 1, x \in \mathbb{R}^n\}$$

$$\|T_2\| = \max_{x,y \in S^{n-1}} \|f_T(x,y)\|_2$$

$$\max_{x,y,z \in S^{n-1}} \sum_{ijk} t_{ijk} x_i y_j z_k$$

z_k



$= \|T\|_2$

$$\max_{x \in S^{n-1}} \|Ax\|_2 = \underbrace{\max_{x,y \in S^{n-1}} y^T A x}_{\text{field of values}} \le \underbrace{\|y\|}_{1} \|Ax\|$$

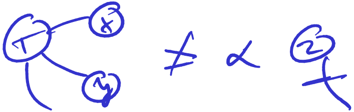
if $y = Ax / \|Ax\|_2$

then

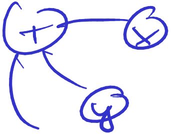
$$y^T A x = \frac{x^T A^T A x}{\|Ax\|_2}$$

$$= \|Ax\|_2$$

$x, y, z \in S^{n-1}$

if  $\neq \alpha$ z

$\exists\ z' \in S^{n-1}$ s.t.

unless  $= \alpha_1$ z

and  $= \alpha_2$ y

it cannot be

.... that $\sum_{ijk} \alpha_{ijk} x_i y_j z_k$

Conditioning of Multilinear Functions

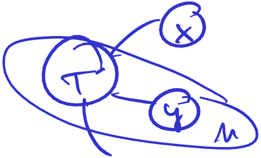
Evaluation of the multilinear map is typically ill-posed for worst case inputs

$x, y \in S^{n-1}$

$$\kappa(f^{(T)}, x, y) = \| J_{f^{(T)},x}(x,y) \|_2 \, / \, \| f^{(T)}(x,y) \|_2$$

$$f^{(T)} \in \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n \qquad J_{f^{(T)},x}(x,y) \in \mathbb{R}^{n \times n}$$

$f^{(T,y)}(x) =$



$= Mx$

$$\| J_{f^{(T)},x} \|_2 = \| M \|_2 \leq \| T \|_2$$

Well-conditioned Tensors

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

$$\|f^{(T)}(x,y)\|_2 \overset{x,y \in S^{n-1}}{=} 1 \qquad f^{(T)}: S^{n-1} \times S^{n-1} \to S^{n-1}$$

$$\begin{bmatrix} 1 & \\ & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \\ & \end{bmatrix} \qquad \inf_{x,y \in S^{n-1}} \|f^{(T)}(x,y)\|_2$$

m.n

Ill-conditioned Tensors

For $n \notin \{2,4,8\}$ given any $\mathcal{T} \in \mathbb{R}^{n\times n\times n}$, $\inf_{\boldsymbol{x},\boldsymbol{y}\in\mathbb{S}^{n-1}} \|\boldsymbol{f}^{(\mathcal{T})}(\boldsymbol{x},\boldsymbol{y})\|_2 = 0$

$\exists\, x, y \quad f^{(T)}(x,y) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

if $\|f^{(T)}(x,y)\|_2^2 = \|x\|_2^2 \|y\|_2^2$

$\forall\, x, y \quad \kappa(T) = 1$

Hurwitz problem

seeks quadratic forms in x, y $z_1 \ldots z_n$, s.t.

e.g. $z_1 = 2x_1 y_2 + 3x_4 y_1$

$\exists\, x, y$

$(x_1^2 + \ldots + x_n^2)(y_1^2 + \ldots y_n^2) = (z_1^2 + \ldots + z_n^2)$

$z_i = \sum_{jk} t_{ijk} x_j y_k$

Algebras as Tensors

A third order tensor can be used to describe an algebra. The Hurwitz problem also

An algebra is a vector space ($\mathbb{R}^n$) equipped with a bilinear function $f: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$

: structure tensor : $f := f^{(T)}$ $z = f^{(T)}(x,y) \Rightarrow z_i = \sum_{jk} t_{ijk} x_j y_k$

implies a result for division algebras, for which the bilinear product is invertible

inverse element | $n = 2, 4, 8$

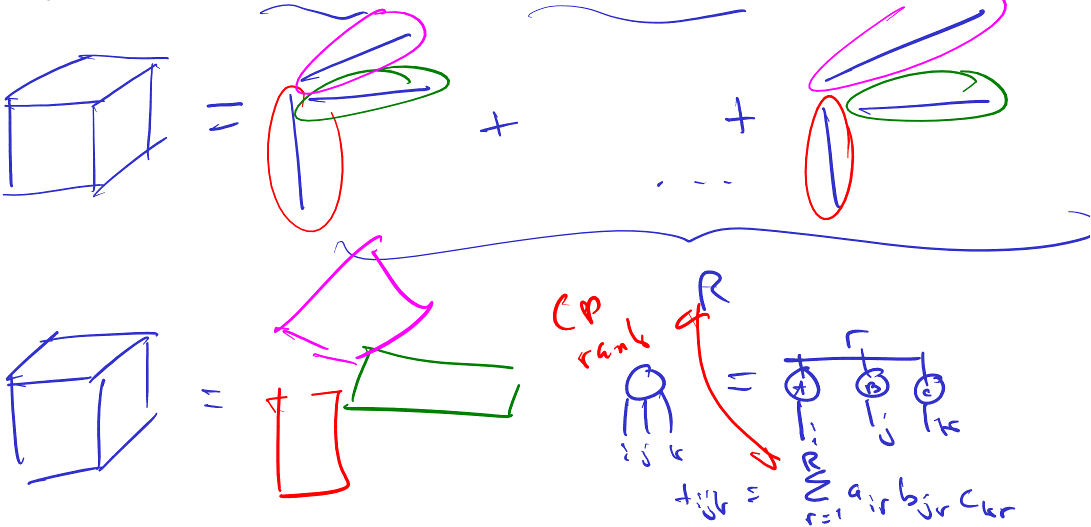
imaginary numbers, quaternions, octonions

$T: S^{n-1} \times S^{n-1} \to S^{n-1}$

$E(T) = 1$

CP Decomposition

- 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

  · exact low rank CPD

  · reduced-size approximation of T, $\min_{A,B,C} \| T - [[A,B,C]] \|_F$

  deduce properties of T

  · $R = O(1)$

  · $R = O(\text{poly}(n))$ $R \le n^{d-1}$

  $\tilde{T}_{ijk} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$ $d \cdot nR$

  $\tilde{T}_{ijk} = \sum_{r=1}^{R} \alpha_r a_{ir} b_{jr} c_{kr}$

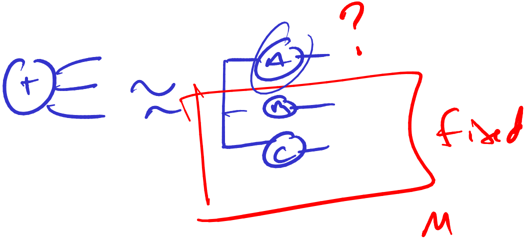
- Basic properties and methods

  · CPD is unique if A, B, C are full rank and $R \le 3n/2$

  uniqueness

  · max rank n^{d-1} $\sim R \le n$

  'typical rank', for $n=2$, 3 and 4 are typical rank 20% 80%



$$T \approx M A$$

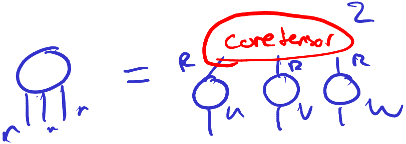
$$\min \| T - M A \|_F$$

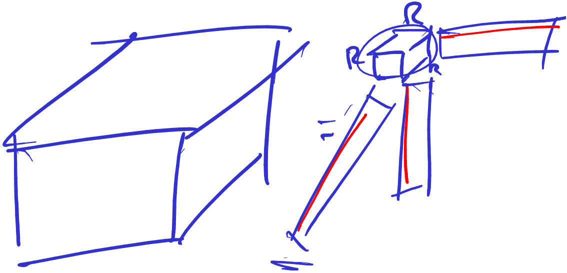
$$M^T M A = \underbrace{M^T T}_{\text{MTTKRP}}$$

$$M A \cong T$$

Tucker Decomposition

- The *Tucker decomposition* expresses an order d tensor via a smaller order d core tensor and d factor matrices

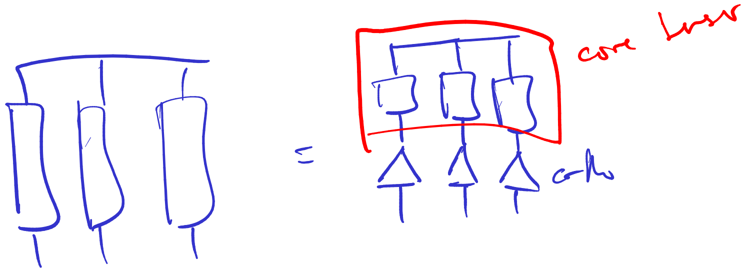




$$t_{ijk} = \sum_{pqr} z_{pqr} u_{ip} v_{jq} w_{kr}$$

$$U^T U = I \qquad V^T V = I \qquad W^T W = I$$

u, v, w orthogonal


core

Tucker Decomposition Basics

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

$R \leq S$

- 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

Summary of Tensor Decomposition Basics

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

decomposition	CP	Tucker	tensor train
size			
uniqueness			
orthogonalizability			
exact decomposition			
approximation			
typical method			

Bilinear Algorithms

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

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

where $\boldsymbol{a}$ and $\boldsymbol{b}$ 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 $\begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{B}_{11} & \boldsymbol{B}_{12} \\ \boldsymbol{B}_{21} & \boldsymbol{B}_{22} \end{bmatrix}$

$$\begin{aligned}
\boldsymbol{M}_1 &= (\boldsymbol{A}_{11} + \boldsymbol{A}_{22}) \cdot (\boldsymbol{B}_{11} + \boldsymbol{B}_{22}) \\
\boldsymbol{M}_2 &= (\boldsymbol{A}_{21} + \boldsymbol{A}_{22}) \cdot \boldsymbol{B}_{11} \\
\boldsymbol{M}_3 &= \boldsymbol{A}_{11} \cdot (\boldsymbol{B}_{12} - \boldsymbol{B}_{22}) \\
\boldsymbol{M}_4 &= \boldsymbol{A}_{22} \cdot (\boldsymbol{B}_{21} - \boldsymbol{B}_{11}) \\
\boldsymbol{M}_5 &= (\boldsymbol{A}_{11} + \boldsymbol{A}_{12}) \cdot \boldsymbol{B}_{22} \\
\boldsymbol{M}_6 &= (\boldsymbol{A}_{21} - \boldsymbol{A}_{11}) \cdot (\boldsymbol{B}_{11} + \boldsymbol{B}_{12}) \\
\boldsymbol{M}_7 &= (\boldsymbol{A}_{12} - \boldsymbol{A}_{22}) \cdot (\boldsymbol{B}_{21} + \boldsymbol{B}_{22})
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{C}_{11} &= \boldsymbol{M}_1 + \boldsymbol{M}_4 - \boldsymbol{M}_5 + \boldsymbol{M}_7 \\
\boldsymbol{C}_{21} &= \boldsymbol{M}_2 + \boldsymbol{M}_4 \\
\boldsymbol{C}_{12} &= \boldsymbol{M}_3 + \boldsymbol{M}_5 \\
\boldsymbol{C}_{22} &= \boldsymbol{M}_1 - \boldsymbol{M}_2 + \boldsymbol{M}_3 + \boldsymbol{M}_6
\end{aligned}$$

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