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 $\boldsymbol{\mathcal{U}}$ of order $s+v$ and $\boldsymbol{\mathcal{V}}$ 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 $\boldsymbol{\mathcal{T}}$, define multilinear function $\boldsymbol{x}^{(1)} = \boldsymbol{f}^{(\boldsymbol{\mathcal{T}})}(\boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(d)})$

Batched Multilinear Operations

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

Tensor Norm and Conditioning of Multilinear Functions

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

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 $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n \times n \times n}$, $\inf_{\boldsymbol{x}, \boldsymbol{y} \in \mathbb{S}^{n-1}} \|\boldsymbol{f}^{(\boldsymbol{\mathcal{T}})}(\boldsymbol{x}, \boldsymbol{y})\|_2 = 0$

Algebras as Tensors

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

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

Homework 1

$$M^{-1} = \gamma I + U(D^{-1} - \gamma I)U^T$$

$$\kappa(M^{-1}A) < \kappa(A)$$

$$A \approx UDU^T \stackrel{?}{=} M$$

$$A = [U \ \bar{U}] \begin{bmatrix} D & \\ & \bar{D} \end{bmatrix} [U \ \bar{U}]^T$$

$$A = UDU^T + \bar{U}\bar{D}\bar{U}^T$$

$$M^{-1}A = \gamma(UDU^T + \bar{U}\bar{D}\bar{U}^T)$$

$$+ UD^{-1}U^T(\quad) = UD^{-1}DU^T = UU^T$$

$$+ \gamma UU^T(\quad \ldots \quad) = \gamma UDU^T$$

$$= UU^T + \gamma \bar{U}\bar{D}\bar{U}^T$$

$$M^{-1}A = \begin{bmatrix} U & \bar{U} \end{bmatrix} \begin{bmatrix} I_{R\times R} & \\ & \gamma \bar{D}_{(n-R)\times(n-R)} \end{bmatrix} \begin{bmatrix} U & \bar{U} \end{bmatrix}^T$$

$$\gamma = R^3$$

$$+ \quad UD^{-1}U^T A = UU^T$$

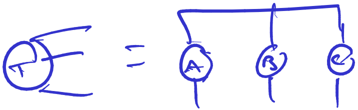
$$\lambda_1 = \frac{1}{\lambda^3}$$

$$\kappa(A) = n^3$$

$$\kappa(M^{-1}A) = \frac{n^3}{R^3} \qquad \text{w. } \gamma = R^3$$

CP Decomposition

- The *canonical polyadic or CANDECOMP/PARAFAC (CP) decomposition* expresses an order d tensor in terms of d factor matrices



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

$$t_{i_1 \ldots i_d} = \sum_r \prod_{j=1}^{d} a^{(j)}_{i_j r}$$

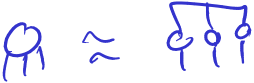
CP Decomposition Basics

- The CP decomposition is useful in a variety of contexts

  · exact rank low or high

  $R \ll n$ $R > n$ $R = O(n^{d-1})$

  · approximate



- Basic properties and methods

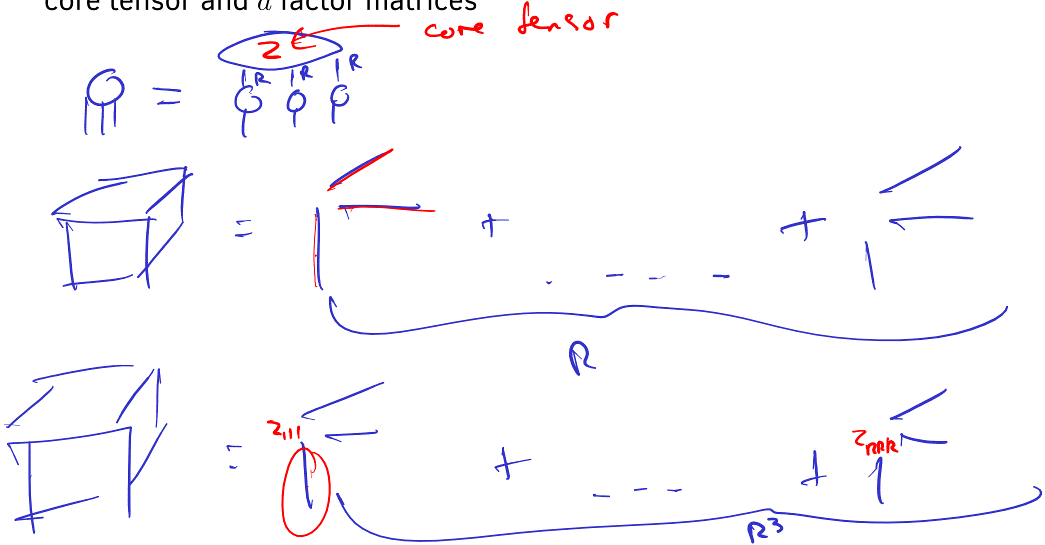
  · approximate is NP-hard   min u, v, w

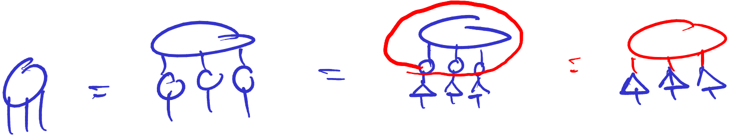
  $R=1$ approximation is NP-hard $\|T - u \circ v \circ w\|_F$

  · exact decomposition (CP) is NP-hard

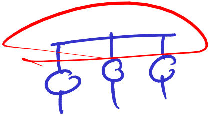
Tucker Decomposition

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



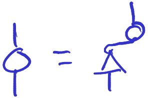


$$t_{ijk} = \sum_{pqr} z_{pqr}\, a_{ip}\, b_{jq}\, c_{kr}$$



$$A, B, C \in \mathbb{R}^{n \times R} \qquad \underline{R \le n}$$

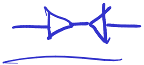
$$A^T A = I \qquad B^T B = I$$



$A \quad Q R$

$Q^T Q = I$

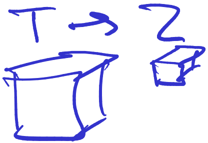
$QQ^T =$



Tucker Decomposition Basics

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

  - compression



- Basic properties and methods

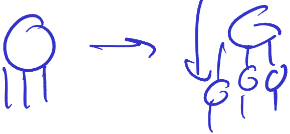
  - approx. is NP hard, for R=1

  - exact factorization can be efficiently

HOSVD

higher-order singular value decomposition

- alg. to compute Tucker
- one-shot



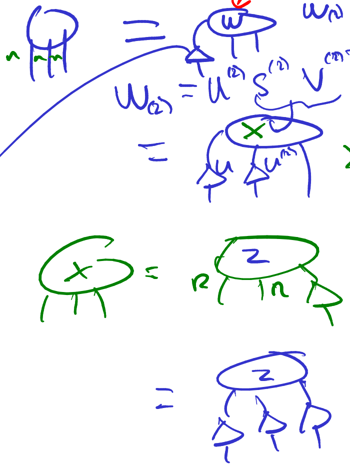
low-rank approx. & fitting

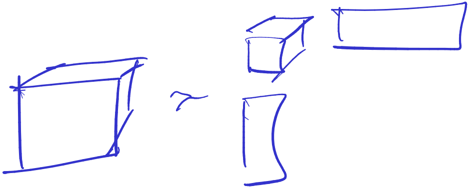
$T_{(1)} = U S V^T$ $\quad T_{(1)} \in \mathbb{R}^{n \times n^2}$

$W_{(1)} \in \mathbb{R}^{n \times nR}$

$W_{(2)} = U^{(2)} S^{(2)} V^{(2)T}$

$X_{(3)} \in \mathbb{R}^{nR \times R^2}$





exact Tucker ranks (dims. of core tensor)

are

(rank($T_{(1)}$), rank($T_{(2)}$), rank($T_{(3)}$))

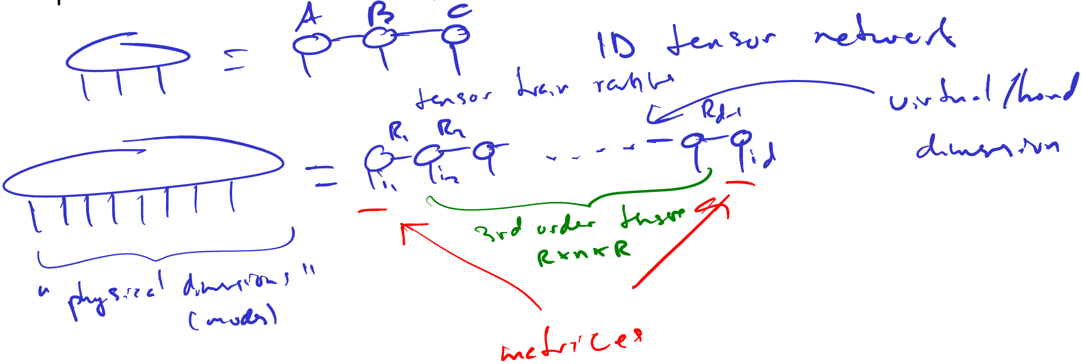
$$A = \big|\,\overline{} + \cdots \big|\,\overline{}$$

$$A = \begin{bmatrix} u_1 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}^T$$

$$T = \underbrace{\big|\,\overline{} + \big|\,\overline{} \cdots \big|\,\overline{}}$$

Tensor Train Decomposition

- The *tensor train decomposition* expresses an order d tensor as a chain of products of order 2 or order 3 tensors



$$t_{ijk} = \sum_{rs} a_{ir} b_{rjs} c_{sk}$$

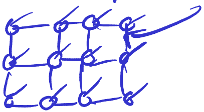
matrix-product-state $\quad t_{i_1 \dots i_d} = \langle a^{(i_1)}, B^{(i_2)} C^{(i_3)} D^{(i_d)} e^{(i_5)} \rangle$

Tensor Train Decomposition Basics

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

· model 1D systems physics

↳ 2D



· Poisson matrix/operator has low tensor train rank after folding (QTT)

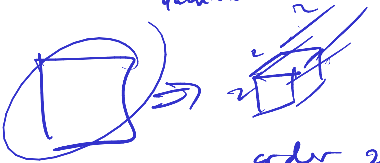
quantized

- Basic properties and methods

R=1 = CP = Tucker

↳ NP hardness

if exact ∃, can compute efficiently



order 2

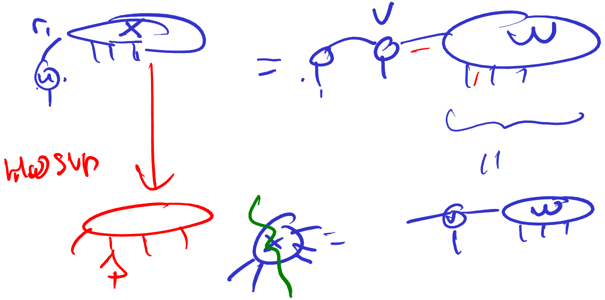
n × n

order 2 log n

2 × 2 × ... × 2

TTSVD

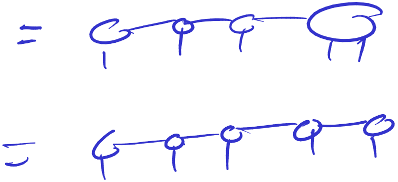
TTSVD



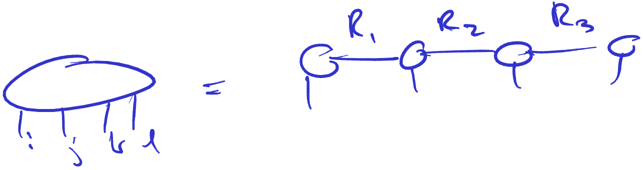
N-1 matrix low-rank factorization

$$T_{i_1 \ldots i_4} = \sum_{r_1} U_{i_1 r_1} X_{r_1 i_2 i_3 i_4}$$

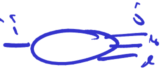
$$X_{r_1 i_2 i_3 i_4} = \sum_{r_2} V_{r_1 i_2 r_2} W_{r_2 i_3 i_4}$$



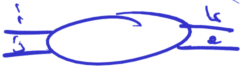
TT ranks



$R_1 = \text{rank}\left(T_{(1)} \right)$



$R_2 = \text{rank}\Big($



$\Big)$

$R_3 = \text{rank}\left(T_{(3)} \right)$

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	dnR	$dnR + R^d$	$2nR + (d-2)nR^2$
uniqueness	unique if rank low	no	no
orthogonalizability	none	partial	partial (canonical form)
exact decomposition	NP hard	HOSVD	TTSVD
approximation	NP hard	NP hard	NP hard
typical method	ALS	HOSVD	TT-ALS (DMRG)