High Performance Tensor Network Contraction and Decomposition Algorithms

Edgar Solomonik

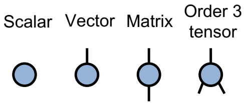
LPNA @CS@Illinois

Department of Computer Science
University of Illinois Urbana-Champaign

ISC High Performance
Hamburg, Germany

May 12, 2024

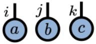
Tensor diagram: a hypergraph representing a tensor network, where tensors are vertices and hyperedges are indices



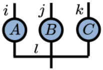
Examples:

Inner product: $\sum_i a_i b_i$

Matrix product : $C_{ik} = \sum_j A_{ij} B_{jk}$



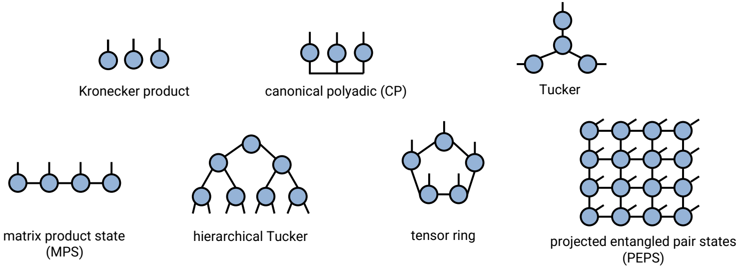
Kronecker/outer product : $T_{ijk} = a_i b_j c_k$



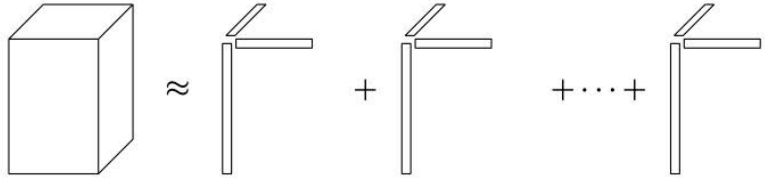
Khatri-Rao product: $T_{ijkl} = A_{il} B_{jl} C_{kl}$

Tensor Decomposition

Tensor decomposition: represent or approximate a tensor as a contraction of smaller tensors



A CP decomposition $\boldsymbol{\mathcal{T}} = [\![A, B, C]\!]$ is a sum of rank one tensors



Applications of Tensor Decompositions in Data Science

- Approximation in modeling of continuous systems
  - quantum chemistry / electrnic structure calculations
  - high-dimensional numerical PDEs
- Data analytics/mining and compression
  - high-order principal component analysis
  - compression of hyperspectral images, neural networks
- Tensor completion
  - given a set of observed entries $\Omega \subset \mathbb{N} \times \mathbb{N} \times \mathbb{N}$, seek

$$\min_{A,B,C} \|(\boldsymbol{\mathcal{T}} - [\![A, B, C]\!])_\Omega\|_F^2 + \lambda(\|A\|_F^2 + \|B\|_F^2 + \|C\|_F^2)$$

  - used for recommender systems, image and video recovery
  - we demonstrate effectiveness for performance modeling[1], e.g.,

$$t_{ijk} = \text{runtime of MatVec of dimension } n_i \text{ and block size } b_j \times b_k$$

[1] Edward Hutter and E.S. ACM/IEEE Supercomputing 2023.

Efficient Tensor Contractions

Cyclops Tensor Framework[1]

- distributed-memory (MPI) library for tensor contractions (C++/OpenMP/CUDA with Python interface)
- finds most communication-efficient distributed layout for contraction
- efficient algorithms for dense tensor redistribution
- extended to support sparsity and general semirings

Sparse tensor times tensor network

- with sparse tensors, fusion of contractions is important
- dynamic programming algorithm to search for optimal loop nest when contracting a single sparse tensor with dense tensors[2]

---

[1] E.S. et al (2014). Journal of Parallel and Distributed Computing, 74(12).

[2] Raghavendra Kanakagiri and E.S., SPAA 2024.

Tensor Decomposition Algorithms

- Rank-1 approximation of high-order tensors is NP-hard
- Alternating least squares (ALS) is commonly used for tensor decompositions
    - For an order 3 tensor $\boldsymbol{\mathcal{X}}$, minimize relative to one factor at a time,

$$\min_A \|\boldsymbol{\mathcal{X}} - [\![A, B, C]\!]\|_F \Rightarrow (C \odot B)A^T \cong X_{(1)}^T$$

    - monotonic linear convergence to local minima
- Classical quadratic optimization in all variables (Gauss-Newton)
    - Jacobian or Hessian matrices are too expensive to form explicitly
    - iterative linear solvers to $J_f^T(x)s = \nabla f(x)$ with implicit Jacobian are competitive with ALS for CP[1,2]

[1] Phan AH, Tichavsky P, Cichocki A. Low complexity damped Gauss-Newton algorithms for CANDECOMP/PARAFAC. SIMAX, 2013.

[2] Singh N, Ma L, Yang H, E.S. Comparison of accuracy and scalability of Gauss–Newton and alternating least squares for CANDECOMC/PARAFAC decomposition. *SIAM Journal on Scientific Computing* (SISC), 2021.

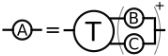
An Effective Distance Metric for CP Decomposition

ALS solves the linear least squares problem

$$\min_{A} \left\| (C \odot B)A^T - T_{(1)}^T \right\|_F$$

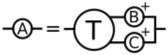
This leads to updates

$$A = T_{(1)}(C \odot B)^{+T}$$



We propose a method that uses a different left inverse of $C \odot B$

$$A = T_{(1)}(C^{+T} \odot B^{+T})$$

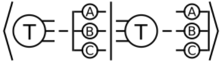


An Effective Distance Metric for CP Decomposition

- CP decomposition algorithms usually minimize the Frobenius norm

$$\|\boldsymbol{\mathcal{T}} - [\![A, B, C]\!]\|_F^2 = \|\operatorname{vec}(\boldsymbol{\mathcal{T}}) - \operatorname{vec}([\![A, B, C]\!])\|_2^2$$

$$= \sum_{i,j,k} \Big(t_{ijk} - \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}\Big)^2$$

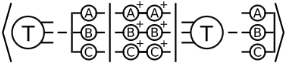


- The new alternating scheme minimizes Mahalanobis distance based on running estimates of covariance matrix inverses[1]

$$\|\operatorname{vec}(\boldsymbol{\mathcal{T}}) - \operatorname{vec}([\![A, B, C]\!])\|_{M^+}^2 = \operatorname{vec}(r)^T M^+ \operatorname{vec}(r)$$

where $\boldsymbol{r} = \operatorname{vec}(\boldsymbol{\mathcal{T}}) - \operatorname{vec}([\![A, B, C]\!])$

and $M = AA^T \otimes BB^T \otimes CC^T$



- Optimizes for most likely decomposition if $\boldsymbol{\mathcal{T}} = \sum_i \boldsymbol{\mathcal{T}}_i$ where $\boldsymbol{\mathcal{T}}_i$ is i.i.d. random rank-1 with Gaussian factors and covariance AA^T,...

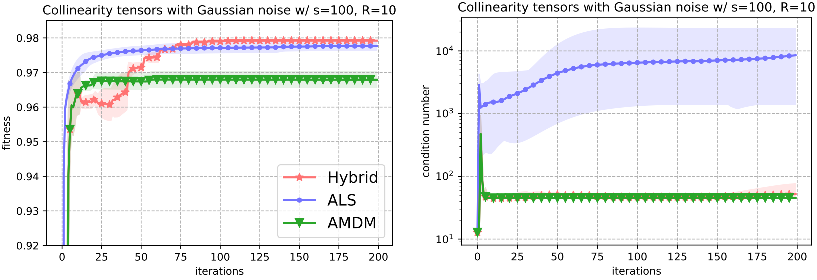
[1] Navjot Singh and E.S., Alternating Mahalanobis Distance Minimization for Stable and Accurate CP Decomposition, *SIAM Journal on Scientific Computing* (SISC), 2023.

Convergence to Exact Decomposition

- ALS achieves a linear convergence rate[1]
- High-order convergence possible by optimizing all variables via Gauss-Newton,[2,3,4] but is costly per iteration relative to ALS
- AMDM achieves superlinear convergance for small R
- AMDM cost per iteration is almost the same as ALS

[1] A. Uschmajew, SIMAX 2012
[2] P. Paatero, Chemometrics and Intelligent Laboratory Systems 1997.
[3] A.H. Phan, P. Tichavsky, A. Cichocki, SIMAX 2013.
[4] N. Singh, L. Ma, H. Yang, E.S., SISC 2021.

Experimental Results for Approximate Decomposition



- for approximate decomposition, AMDM achieves good conditioning
- hybrid ALS/AMDM achieves low residual

Inexact Optimization for Tensor Decompositions

We now return to approximation in the standard Frobenius norm, and consider fast inexact algorithms for various decompositions

- ALS for tensor decompositions yields highly over-constrained linear least squares problems with tensor product structure
- for CP, the factor A is determined from Khatri-Rao product $B \odot C$
- for the HOOI algorithm for Tucker, the equations are given by a Kronekecer product $B \otimes C$ with orthogonal B and C
- the number of right-hand sizes is often large (for CP each row of A is independent in a step of ALS) and they are expensive to construct

Sketching for Alternating Least Squares

Radomized subspace embeddings provide a powerful tool for fast approximation

- for $A \in \mathbb{R}^{m\times n}$ seek random $S \in \mathbb{R}^{k\times m}$ such that, $\forall x \in \mathbb{R}^n$,

$$\|S^T SAx - Ax\| \leq \epsilon\|Ax\| \text{ w.h.p.}$$

- compute $SA\hat{x} \cong Sb$, then if $Ax \cong b$, $\|Ax - A\hat{x}\| \leq \epsilon\|b\|$, w.h.p.

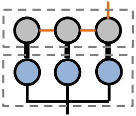
A variety of distributions can be chosen for the random sketch matrices

- sampling (each row of S has one nonzero) is effective especially for sparse A or b, leverage scores provide optimal sampling distribution, requires $k = O(n\log(n)/\epsilon^2)$
- count sketch (each column of S has one nonzero) avoids need to know leverage score distribution at increased complexity of applying S

Efficient Sketching Matrices

If A or b have tensor product structure, choosing S to also have matching structure enables fast computation of SA and Sb, e.g., if

$$A = B \otimes C, \quad S = S_1 \otimes S_2, \quad SA = (S_1 B) \otimes (S_2 C).$$



- We have developed efficient sketching algorithms for (sparse) CP and Tucker[1] and general dense tensor networks[2]
- Tensor decompositions with sketching substantially improve efficiency for large scale tensor decomposition problems[3]

[1] Linjian Ma and E.S. Fast and accurate randomized algorithms for low-rank tensor decompositions, NeurIPS'21

[2] Linjian Ma and E.S. Cost-efficient Gaussian tensor network embeddings for tensor-structured inputs, NeurIPS'22

[3] Bharadwaj V, Malik OA, Murray R, Buluç A, Demmel J. Distributed-Memory Randomized Algorithms for Sparse Tensor CP Decomposition, arXiv:2210.05105.

Further References and Recent Work by LPNA

- **Cyclops for tensor completion** *Navjot Singh, et al. Distributed-memory tensor completion for generalized loss functions in python using new sparse tensor kernels, JPDC 2022.*
- **AMDM:** *Navjot Singh and E.S. Alternating Mahalanobis Distance Minimization for Stable and Accurate CP Decomposition, SISC 2023.*
- **Sketching Tucker:** *Linjian Ma and ES., Fast and accurate randomized algorithms for low-rank tensor decompositions, NeurIPS'21.*
- **Sketching general tensor networks:** *Linjian Ma and E.S. Cost-efficient Gaussian tensor network embeddings for tensor-structured inputs, NeurIPS 2022.*
- **CP for perf. modeling:** *Edward Hutter and E.S. High-dimensional performance modeling via tensor completion, SC 2023.*
- **Efficient sparse tensor contraction:** *Raghavendra Kanakagiri and E.S. Minimum cost loop nests for contraction of a sparse tensor with a tensor network, SPAA 2024.*