Optimization methods for tensor decomposition

Edgar Solomonik

Department of Computer Science
University of Illinois Urbana-Champaign

ISE Seminar
Lehigh University
January 30th, 2024
1. Tensor Decompositions and Applications

2. Optimization Algorithms for Tensor Decomposition

3. Alternating Mahalanobis Distance Minimization

4. Sketching Methods for Inexact Optimization

5. Conclusion
Tensor diagrams: a hypergraph representing a tensor contraction, 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: represent or approximate a tensor as a contraction of smaller tensors

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

- Compact representation for operators and solutions to PDEs
  - quantum simulation (electronic structure, quantum spin models)
  - plasma physics (Boltzmann equation is a function of position and momentum, resulting in a 6D discretization)
  - high-order methods for fluid dynamics (each element represented by order 3 tensor, ROM results in 3D tensor operators)

- Data analytics/mining and compression
  - high-order principal component analysis
  - completion of multi-dimensional datasets
  - neural networks are composed of tensors

- Bilinear algorithms via CP decomposition

![Diagram showing matrix multiplication and CP decomposition](image)
The minimum rank tree decomposition of a tensor may be obtained via \( n - 1 \) SVDs.

- for Tucker, this is the high-order SVD (HoSVD) algorithm
- tensor train and hierarchical Tucker are similar

Finding the optimal low-rank approximation is NP-hard.

- finding an optimal rank-1 approximation (special case of any tensor decomposition) is NP-hard

Determining the minimum CP (border) rank is NP-hard.

Contracting a 2D lattice tensor network (PEPS) is \#P hard.
Alternating least squares (ALS) is commonly used for tensor decompositions

- minimizing error relative to one tensor (factor) in the decomposition yields a quadratic optimization problem
- monotonic linear convergence to local minima

Classical quadratic optimization in all variables (Gauss-Newton)

- full Jacobian or Hessian matrices are too expensive to form/factorize explicitly
- iterative linear solvers to $J_f^T(x)s = \nabla f(x)$ with implicit Jacobian are competitive with ALS for CP\(^1,2\)

Subgradient methods / SGD are less popular due to slower progress

---

An Effective Distance Metric for CP Decomposition

- CP decomposition algorithms usually minimize the Frobenius norm

\[ \| \mathbf{T} - [A, B, C]\|_F^2 = \| \text{vec}(\mathbf{T}) - \text{vec}([A, B, C])\|_2^2 \]

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

- Ardavan Afshar et al [AAAI 2021] minimize Wasserstein distance, improving robustness for downstream tasks

- We consider Mahalanobis distance based on covariance matrices\(^1\)

\[ \| \text{vec}(\mathbf{T}) - \text{vec}([A, B, C])\|_M^{-1}^2 = \text{vec}(r)^T M^{-1} \text{vec}(r) \]

where \( r = \text{vec}(\mathbf{T}) - \text{vec}([A, B, C]) \)

and \( M = AA^T \otimes BB^T \otimes CC^T \)

\[ + (I - AA^+) \otimes (I - BB^+) \otimes (I - CC^+) \]

\(^1\)Navjot Singh and E.S., Alternating Mahalanobis Distance Minimization for Stable and Accurate CP Decomposition, SISC 2023
Alternating Minimization of Mahalanobis Distance (AMDM)

- Optimizing the new metric

\[
\min_{A,B,C} \| \text{vec}(T) - \text{vec}([A, B, C]) \|_{M^{-1}}^2
\]

in an alternating manner yields ALS-like updates

\[
A = T_{(1)}(C^+ \circ B^+) \quad \text{(1)}
\]

where \(M^+\) denotes the pseudoinverse of matrix \(M\)

- By comparison, the ALS algorithm computes

\[
A = T_{(1)}(C \circ B)^+ \quad \text{(1)}
\]

- Both \(C^+ \circ B^+\) and \((C \circ B)^+\) are left inverses of \(C \circ B\), suitable for minimizing

\[
\min_A \| (C \circ B)A^T - T_{(1)}^T \|
\]
Convergence to Exact Decomposition

When seeking an exact decomposition for a rank $R \leq s$ tensor

- 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 at least quartic order local convergence per sweep of alternating updates
  - error from true solution after solving for one factor scales with product of errors of other factors
- cost per iteration is roughly the same as ALS (dominated by single matricized tensor times Khatri-Rao product (MTTKRP))

---

\(^1\) 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.
AMDM achieves high-order convergence for exact decomposition of synthetic random low-rank problems
Properties of Fixed Points of AMDM

- When \( \text{rank}(\mathcal{T}) > R \), consider an AMDM fixed point, \( A, B, C \)
- \( X = A^+T \), \( Y = B^+T \), \( Z = C^+T \) yield a critical point of

\[
f(X, Y, Z) = \langle \mathcal{T}, [X, Y, Z] \rangle - \log(\det(X^TXY^TYZ^TZ))
\]

and satisfy tensor-eigenvector-like equations:

\[
A = X^+T = T_{(1)}(Z \odot Y) \quad A = -X^+ = -T(1)(Z \odot Y)
\]
\[
B = Y^+T = T_{(2)}(Z \odot X) \quad B = -Y^+ = -T(2)(Z \odot X)
\]
\[
C = Z^+T = T_{(3)}(Y \odot X) \quad C = -Z^+ = -T(3)(Y \odot X)
\]

- The reconstructed tensor \( \tilde{T} = [A, B, C] \) exactly represents the action of the original tensor on vectors in the span of the factors

\[
T_{(1)} \text{vec}(u) = \tilde{T}_{(1)} \text{vec}(u), \quad \forall u \in \text{span}(C \odot B)
\]
\[
T_{(2)} \text{vec}(v) = \tilde{T}_{(2)} \text{vec}(v), \quad \forall v \in \text{span}(C \odot A)
\]
\[
T_{(3)} \text{vec}(w) = \tilde{T}_{(3)} \text{vec}(w), \quad \forall w \in \text{span}(B \odot A)
\]
Approximate Decomposition Results with AMDM

- AMDM finds decomposition with lower CP condition number\(^1\)
- Hybrid version gradually transitions from basic AMDM to ALS

---

\(^1\)P. Breiding and N. Vannieuwenhoven, SIMAX 2018.
Statistical Interpretation of AMDM

Consider a random rank-1 tensor

\[ X = u \circ v \circ w, \]

where \( u, v, \) and \( w \) are Gaussian random vectors with zero mean and covariance matrices:

\[ \mathbb{M}[u] = AA^T, \mathbb{M}[v] = BB^T, \text{ and } \mathbb{M}[w] = CC^T. \]

Let \( T \) be a sum of \( R \) samples of \( X, \)

\[ T = N + \sum_{i=1}^{R} X_i. \]

AMDM performs covariance matrix estimation for \( X, \) while simultaneously minimizing Mahalanobis distance derived from the covariance matrix,

\[ \mathbb{M}[u \otimes v \otimes w] = AA^T \otimes BB^T \otimes CC^T. \]
Minimize for each factor in an alternating manner,

$$\text{vec}(T)^T M[u \otimes v \otimes w]^+ \text{vec}(T), \text{ s.t. } \det(M[u \otimes v \otimes w]) = 1$$

[likelihood of covariance matrix given $T$]

$$\text{vec}(T - [A, B, C])^T M[u \otimes v \otimes w]^+ \text{vec}(T - [A, B, C])$$

[fit under metric].

In the first objective, we fix the generalized variance of the distribution, $\det(M[x \otimes y \otimes z])$. 
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

Randomized 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, $\forall x \in \mathbb{R}^n$,  
  \[ \| S^T S A x - A x \| \leq \epsilon \| A x \| \text{ w.h.p.} \]

- compute $S A \hat{x} \approx S b$, then if $A x \approx b$, $\| A x - A \hat{x} \| \leq \epsilon \| b \|$, 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 $S$
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

$$A = B \otimes C, S = S_1 \otimes S_2, SA = (S_1 B) \otimes (S_2 C).$$
Leverage score sampling

- Since $Q = C \otimes B$, leverage scores satisfy

$$l_{(i-1)n+j}(Q) = \|q_{(i-1)n+j}\|^2_2 = \|c_i\|^2_2\|b_j\|^2_2 = l_i(C)l_j(B)$$

hence we can take products of independent samples of rows of $A$ and $B$ to obtain the leverage-score based distribution of columns of $Q$

- Since $A$, $B$, $C$ are changing, we must sample the tensor (right-hand side) differently in each optimization step

 TensorSketch\(^1\) reduces the amount of necessary sampling to 1 round

\(^1\)Malik and Becker, NeurIPS 2018.
ALS + TensorSketch (Malik and Becker, NeurIPS 2018)

- Solving for each factor matrix or the core tensor at a time

\[ \min_A \frac{1}{2} \left\| \left( C \otimes B \right) X^{(1)}_T A^T - T^{(1)}_T \right\|^2_F \]

or

\[ \min_X \frac{1}{2} \left\| \left( C \otimes B \otimes A \right) \text{vec}(X) - \text{vec}(T) \right\|^2_F \]

<table>
<thead>
<tr>
<th>Algorithm for Tucker</th>
<th>LS subproblem cost</th>
<th>Sketch size ((k))</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOOI</td>
<td>(\Omega(\text{nnz}(\mathcal{T})R))</td>
<td>/</td>
</tr>
<tr>
<td>ALS + TensorSketch</td>
<td>(\tilde{O}(knR + kR^3))</td>
<td>(O((R^2/\delta) \cdot (R^2 + 1/\epsilon)))</td>
</tr>
<tr>
<td>HOOI + TensorSketch</td>
<td>(O(knR + kR^4))</td>
<td>(O((R^2/\delta) \cdot (R^2 + 1/\epsilon^2)))</td>
</tr>
<tr>
<td>HOOI + leverage scores</td>
<td>(O(knR + kR^4))</td>
<td>(O(R^2/(\epsilon^2 \delta)))</td>
</tr>
</tbody>
</table>
Experiments: Tensors with Spiked Signal

(a) 5 sweeps, sample size $16R^2$

(b) 5 sweeps, sample size $KR^2$

(c) sample size $16R^2$

- $\mathbf{T} = \mathbf{T}_0 + \sum_{i=1}^{5} \lambda_i a_i \circ b_i \circ c_i$, each $a_i, b_i, c_i$ has unit 2-norm, $\lambda_i = 3 \frac{\|\mathbf{T}_0\|_F}{i^{1.5}}$

- Leading low-rank components obey the power-law distribution

- Tensor size $200 \times 200 \times 200$, $R = 5$

- TS-ref: (Malik and Becker, NeurIPS 2018)
Experiments: CP decomposition

\[ \mathbf{T} = \sum_{i=1}^{R_{\text{true}}} a_i \circ b_i \circ c_i, \quad R_{\text{true}} / R = 1.2 \]

- Tensor size \( 2000 \times 2000 \times 2000 \), \( R = 10 \), sample size \( 16R^2 \)


- Tucker+CP: Run Tucker HOOI first, then run CP-ALS on the Tucker core
  - Run Tucker HOOI with 5 sweeps, CP-ALS with 25 sweeps

- Recent work (V Bharadwaj et al, Larsen and Kolda, arXiv:2301.12584) implicitly samples the leverage score distribution for CP exactly
Problem: Given a tensor network input data, $x$, find a **Gaussian** tensor network embedding, $S$, such that the embedding is $(\epsilon, \delta)$-accurate and

- The number of rows of $S$ (sketch size $m$) is low
- Asymptotic cost to compute $Sx$ is minimized

An (oblivious) embedding $S \in \mathbb{R}^{m \times s}$ is $(\epsilon, \delta)$-accurate if\(^1\)

$$\Pr \left[ \frac{\|Sx\|_2 - \|x\|_2}{\|x\|_2} > \epsilon \right] \leq \delta \text{ for any } x$$

\(^1\)Woodruff, Sketching as a tool for numerical linear algebra, 2014
Sketching Tensor Network Data

Previous work:
- Kronecker product embedding\(^1\): inefficient in computational cost
- Tree embedding (e.g. MPS)\(^2\): efficient for specific data (Kronecker product, MPS), but efficiency unclear for general tensor network data

Assumptions throughout our analysis:
- Classical \(O(n^3)\) matmul cost
- Consider embeddings defined on graphs with no hyperedges
- Each dimension to be sketched
  - has a size lower bounded by the sketch size
  - is only adjacent to one data tensor

---
\(^1\)Ahle et al, Oblivious sketching of high-degree polynomial kernels, SODA 2020
\(^2\)Rakhshan and Rabusseau, Tensorized random projections, AISTATS 2020
Sufficient condition for \((\epsilon, \delta)\)-accurate embedding

The embedding \(G = (V, E, w)\) is accurate if there exists a linear ordering of \(V\) such that in its induced DAG, the weighted sum of out-going edges adjacent to each \(v \in V\) is \(\Omega(m)\), where \(m = N \log(1/\delta)/\epsilon^2\)

Proof of accuracy leverages two key prior results\(^1\)

1. If \(S\) is \((\epsilon, \delta)\)-accurate, so is \(I \otimes S \otimes I\)
2. If \(S_1, \ldots, S_N\) are \(O(\epsilon/\sqrt{N}), \delta)\)-accurate, \(S_1 \cdots S_N\) is \((\epsilon, \delta)\)-accurate

\(^1\)Ahle et al, Oblivious sketching of high-degree polynomial kernels, SODA 2020
Efficient General Sketching

- Tensor network sketch contains
  1. Kronecker product embedding
  2. binary tree of small tensor network gadgets

- Each gadget sketches product of two tensors
  - chosen to minimize cost depending on connectivity
  - may or may not be a tree

- Can reduce cost by up to $O(\sqrt{m})$ relative to binary tree
  - near-optimal under assumptions
Applications of Tensor Network Sketching

- If input data is Khatri-Rao product or tensor product
  - new gadgets reduce cost by $O(\sqrt{m})$ relative to Gaussian binary tree embedding
  - this allows acceleration of sketching for CP decomposition
  - tree-like sketch structure also allows intermediate reuse during optimization (dimension trees)

- When data is an MPS (tensor train)
  - plain tree sketch is efficient (sketch can be binary tree or MPS-like)
  - shows optimality (subject to our sufficient condition) of prior work\(^1\)

---

\(^1\) Al Daas, Hussam, et al. Randomized algorithms for rounding in the tensor-train format, SISC 2023.
Summary and Conclusions

- Sketching for Tucker decomposition
  - Sketching HOOI gives accurate decomposition with enough sketch size
  - TensorSketch permits 1-pass (streaming) Tucker and CP
  - High polynomial scaling in rank; for CP addressable by indirect leverage score sampling

- Gaussian tensor network sketching
  - achieves linear cost relative to number of input tensors
  - limited analysis to Gaussian tensors, classical matrix multiplication cost
  - not considering hyperedges in sketch, e.g., Khatri-Rao product in TensorSketch

---

Further References and Recent Work by LPNA

- **AMDM:** Navjot Singh and E.S. Alternating Mahalanobis Distance Minimization for Stable and Accurate CP Decomposition, SISC 2023.

- **Sketching Tucker:** Linjian Ma and E.S., 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, arXiv:2307.05740.

- **Inexact solvers for interior point:** Samah Karim and E.S. Efficient preconditioners for interior point methods via a new Schur-complement-based strategy, SIMAX 2022.