BERKELEY LAB

U.S. DEPARTMENT OF ENERGY
Office of Science

Leverage-Based Sampling Algorithms for Tensor Decomposition Problems

Vivek Bharadwaj [1], Osman Asif Malik [2], Riley Murray [3,2,1], Laura Grigori [4], Aydın Buluç [2,1], James Demmel [1]

[1] Electrical Engineering and Computer Science Department, UC Berkeley

[2] Computational Research Division, Lawrence Berkeley National Lab

[3] International Computer Science Institute

[4] Institute of Mathematics, EPFL & Lab for Simulation and Modelling, Paul Scherrer Institute

Summary

First part of this talk covers two works involving CP decomposition:

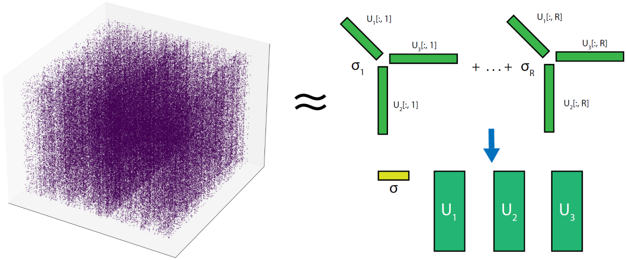
1. Fast Exact Leverage Score Sampling from Khatri-Rao Products with Applications to Tensor Decomposition. To appear at NeurIPS 2023: *https://arxiv.org/abs/2301.12584*

2. Distributed-Memory Randomized Algorithms for Sparse Tensor CP Decomposition. Under review: *https://arxiv.org/abs/2210.05105*

Second part of this talk: emerging extensions of above work to tensor-train decomposition. Collaboration w/ Guillaume Rabusseau, Beheshteh Rakhshan at U. Montreal.

Sparse Tensor Candecomp / PARAFAC Decomposition

Our Goal: Compute an approximate rank-R CP decomposition of an N-dimensional $I \times \ldots \times I$ sparse tensor $\mathcal{T}$:



Focus on large sparse tensors (mode sizes in the millions) and moderate decomposition rank $R \approx 10^2$. Assume $|I_j| = I$ for all j and $I \geq R$.

Alternating Least-Squares CP Decomposition

- ALS procedure: Randomly initialize factors $U_1, ..., U_N$, iteratively optimize one factor at a time while keeping others constant.

- Optimal value for U_j:

$$\mathrm{argmin}_X \|AX - B\|_F$$

  where

  - $A = U_N \odot ... \odot U_{j+1} \odot U_{j-1} \odot ... \odot U_1$ is a **Khatri-Rao product**

  - $B = \mathrm{mat}(\mathcal{T}, j)^\top$

Randomized Linear Least-Squares

- Apply sketching operator S to both A and B, solve reduced problem

$$\min_{\tilde{X}} \left\| SA\tilde{X} - SB \right\|_F$$

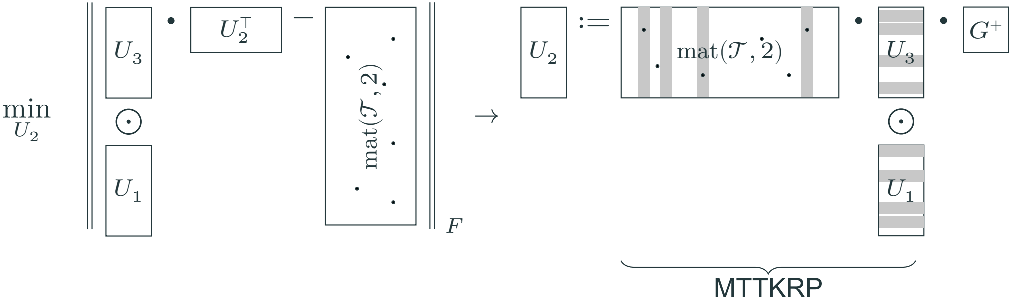
- Want an (ε, δ) guarantee on solution quality: with high probability $(1 - \delta)$,

$$\left\| A\tilde{X} - B \right\|_F \leq (1 + \varepsilon) \min_X \| AX - B \|$$

- Osman talked about Gaussian / TensorSketch operators. Here, restrict S to be a *sampling* matrix: selects and reweights rows from A and B.

Effect of Sampling Operator

$$\min_{U_j} \left\| \left[\bigodot_{k\neq j} U_k \right] \cdot U_j^\top - \mathrm{mat}(\mathcal{T}, j)^\top \right\|_F$$



Prior Work

- (SPALS, D. Cheng et al. 2016): Sample rows according to approximate *leverage score distribution* on A. Worst-case **exponential** in N to achieve (ε, δ) guarantee.

- (CP-ARLS-LEV Larsen & Kolda 2022): Similar approximation, hybrid random-deterministic sampling strategy and practical improvements.

- (TNS-CP, Malik 2022): Samples from exact leverage distribution with **polynomial** complexity to achieve (ε, δ) guarantee, but linear dependence on I for each sample.

Our Contributions

Method	Round Complexity ($\tilde{O}$ notation)
CP-ALS	$N(N+I)I^{N-1}R$
CP-ARLS-LEV (2022)	$N(R+I)R^N/(\varepsilon\delta)$
TNS-CP (2022)	$N^3IR^3/(\varepsilon\delta)$
GTNE (2022)	$N^2(N^{1.5}R^{3.5}/\varepsilon^3+IR^2)/\varepsilon^2$
STS-CP (ours, 2023)	$N(NR^3\log I+IR^2)/(\varepsilon\delta)$

- We build a data structure with runtime **logarithmic** in the height of the KRP and quadratic in R to sample from leverage scores of A.

- Yields the **STS-CP** algorithm: lower asymptotic runtime for randomized CP decomposition than recent SOTA methods (practical too!)

Leverage Score Sampling

We will sample rows i.i.d. from A according to the *leverage score distribution* on its rows. Leverage score ℓ_i of row i is

$$\ell_i = A[i,:](A^\top A)^+ A[i,:]^\top$$

Theorem (Leverage Score Sampling Guarantees)

Suppose $S \in \mathbb{R}^{J \times I}$ is a leverage-score sampling matrix for $A \in \mathbb{R}^{I \times R}$, and define

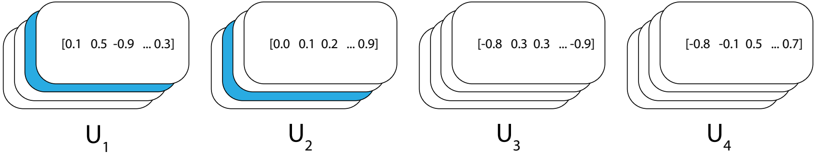
$$\tilde{X} := \arg\min_{\tilde{X}} \left\| SA\tilde{X} - SB \right\|_F$$

If $J \gtrsim R \max(\log(R/\delta), 1/(\varepsilon\delta))$, then with probability at least $1-\delta$,

$$\left\| A\tilde{X} - B \right\|_F \leq (1+\varepsilon) \min_X \|AX - B\|_F$$

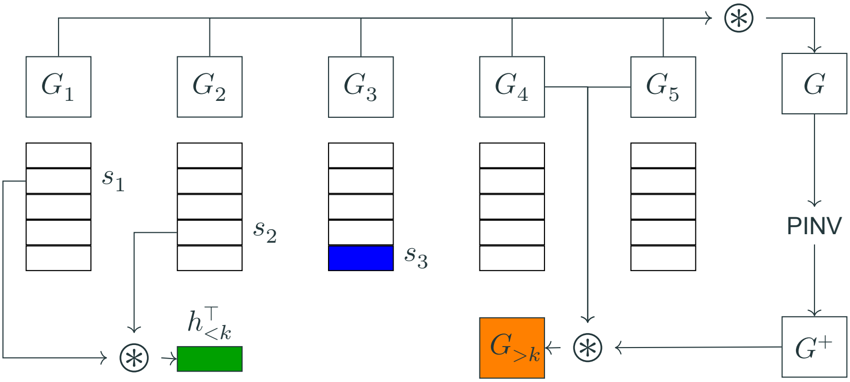
Leverage Score Sampling

- For $I = 10^7, N = 3$, matrix A has 10^{21} rows. Can't even index rows with 64-bit integers.
- Instead: draw a row from each of $U_1, \ldots, U_N$, return their Hadamard product.



- Let $\hat{s}_j$ be a random variable for the row index drawn from U_j. Assume $(\hat{s}_1, \ldots, \hat{s}_N)$ jointly follows the leverage score distribution on $U_1 \odot \ldots \odot U_N$.

The Conditional Distribution of $\hat{s}_k$



Theorem

$$p(\hat{s}_k = s_k \mid \hat{s}_{<k} = s_{<k}) \propto \langle h_{<k} h_{<k}^\top, U_k [s_k, :]^\top U_k [s_k, :], G_{>k} \rangle$$

Key Sampling Primitive

$$q[i] := C^{-1} \langle h_{<k} h_{<k}^{\top}, U_k[i,:]^{\top} U_k[i,:], G_{>k} \rangle$$

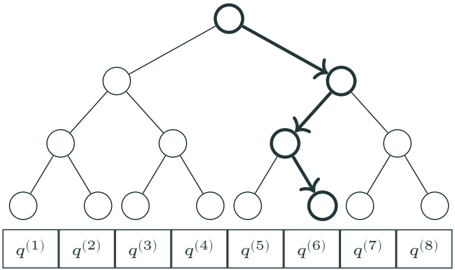
- Can't compute q entirely - would cost $O(IR^2)$ per sample per mode.
- Imagine we magically had all entries of q - how to sample? Initialize I bins, j'th has width $q[j]$.
- Choose random real r in $[0, 1]$, find "containing bin" i such that

$$\sum_{j=0}^{i-1} q[j] < r < \sum_{j=0}^{i} q[j]$$

Binary Tree Inversion Sampling

- Locate bin via binary search (truncated to $\log(I/R)$ levels)
- Root: branch right iff $\sum_{j=0}^{I/2} q[j] < r$
- Level 2: branch right iff

$$\sum_{j=0}^{I/2} q[j] + \sum_{j=I/2}^{3I/4} q[j] < r$$



Key: Can compute summations quickly if we cache information at each node!

Caching Partial Gram Matrices

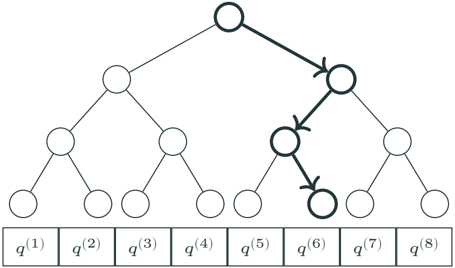
Let an internal node v correspond to an interval of rows $[S(v)...E(v)]$.

$$
\begin{aligned}
\sum_{j=S(v)}^{E(v)} q[j] &= \sum_{j=S(v)}^{E(v)} C^{-1} \langle h_{<k} h_{<k}^\top, U_k[j,:]^\top U_k[j,:], G_{>k} \rangle \\
&= C^{-1} \langle h_{<k} h_{<k}^\top, \sum_{j=S(v)}^{E(v)} U_k[j,:]^\top U_k[j,:], G_{>k} \rangle \\
&= C^{-1} \langle h_{<k} h_{<k}^\top, U_k[S(v):E(v),:]^\top U_k[S(v):E(v),:], G_{>k} \rangle \\
&:= C^{-1} \langle h_{<k} h_{<k}^\top, G^v, G_{>k} \rangle
\end{aligned} \tag{1}
$$

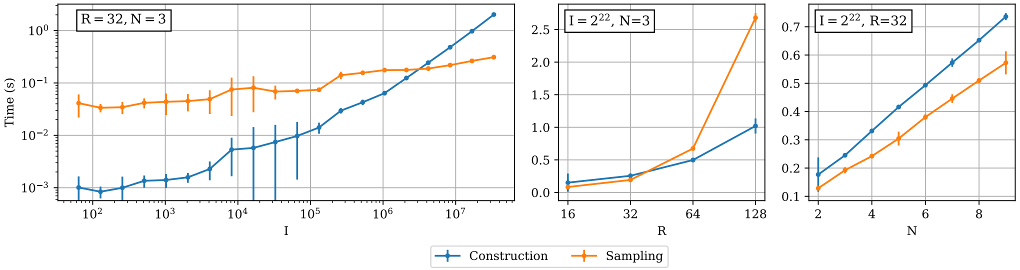
Can compute and store G^v for ALL nodes v in time $O(IR^2)$, storage space $O(IR)$.
Only have to recompute once per ALS round.

Efficient Sampling after Caching

- At internal nodes, compute $C^{-1}\langle h_{<k} h_{<k}^{\top}, G^{v}, G_{>k}\rangle$ in $O(R^2)$ time (read normalization constant from root)
- At leaves, spend $O(R^3)$ time to compute remaining values of q. Can reduce to $O(R^2 \log R)$, see paper.
- Complexity per sample: $O(NR^2 \log I)$ (summed over all tensor modes).

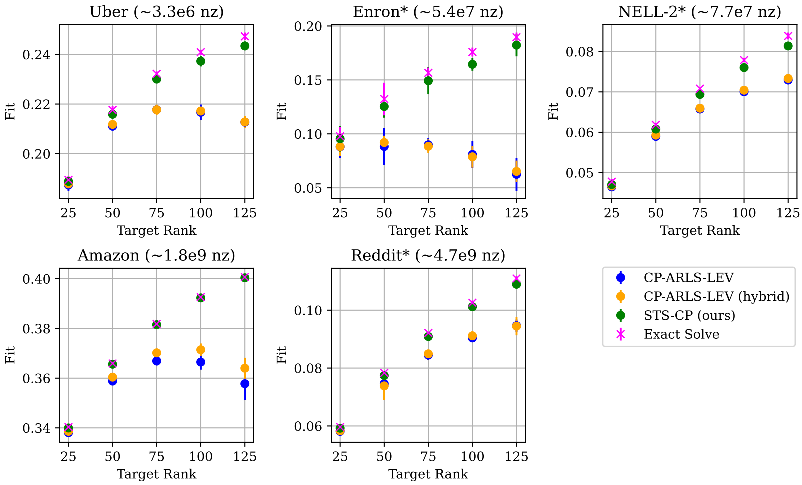


Runtime Benchmarks (LBNL Perlmutter CPU)



C++ Implementation Linked to OpenBLAS. 1 Node, 128 OpenMP Threads, BLAS3 Construction, BLAS2 Sampling, $J = 65,536$ samples.

Accuracy Comparison for Fixed Sample Count



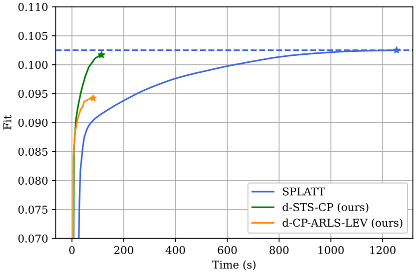
ALS Accuracy Comparison for $J = 2^{16}$ samples.

Distributed-Memory High-Performance Implementation

- We give high-performance implementations of STS-CP and CP-ARLS-LEV scaling to thousands of CPU cores.

- Up to 11x speedup over SPLATT

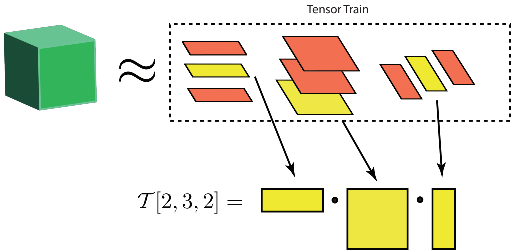
- Several communication / computation optimizations unique to randomized CP decomposition.



Accuracy vs. time, Reddit tensor, $R = 100$, 512 cores / 4 Perlmutter CPU nodes, 4.7 billion nonzeros.

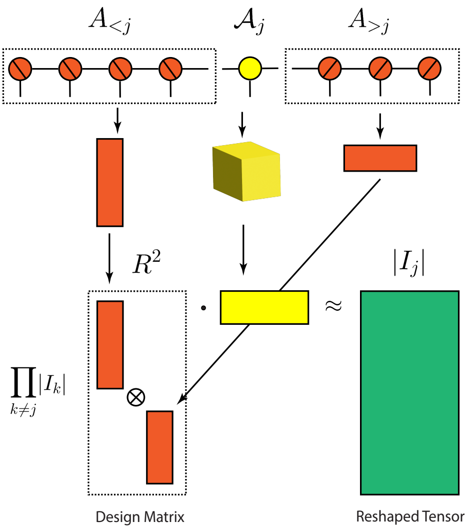
Tensor-Train Decomposition

The tensor-train decomposition represents a tensor $\mathcal{T}$ as a contraction between order-3 “tensor-cores”.



j'th core has dimensions $R_j \times |I_j| \times R_{j+1}$. Represents a tensor with I^N elements using $O(NIR^2)$ space when all rank are equal.

Iterative TT Optimization Problems



Sampling from $A_{<j}$

Theorem (Orthonormal Subchain Leverage Sampling)

There exists a data structure that costs $O(IR^3)$ per tensor train core to build / update. For any $1 < j \leq N$, the structure can sample a row from $A_{<j}$ proportional to it squared row norm in time

$$O((j-1)R^2 \log I)$$

Apply same binary tree trick to the left matricizations of each core $\mathcal{A}_j$, exploit orthonormality to reduce complexity. Accelerates TT-ALS.

Ongoing Work

- Looking for further applications of orthonormal tensor train sketch.
- Extension to non-orthonormal case challenging, but potentially rewarding.
- If you have an application involving contraction of an unstructured operator with a tensor-train / MPS, let’s talk!

Thank you! Read the work on Arxiv:

https://arxiv.org/abs/2301.12584

https://arxiv.org/abs/2210.05105