BERKELEY LAB

Fast Exact Leverage Score Sampling from Khatri-Rao Products with Applications to Tensor Decomposition

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

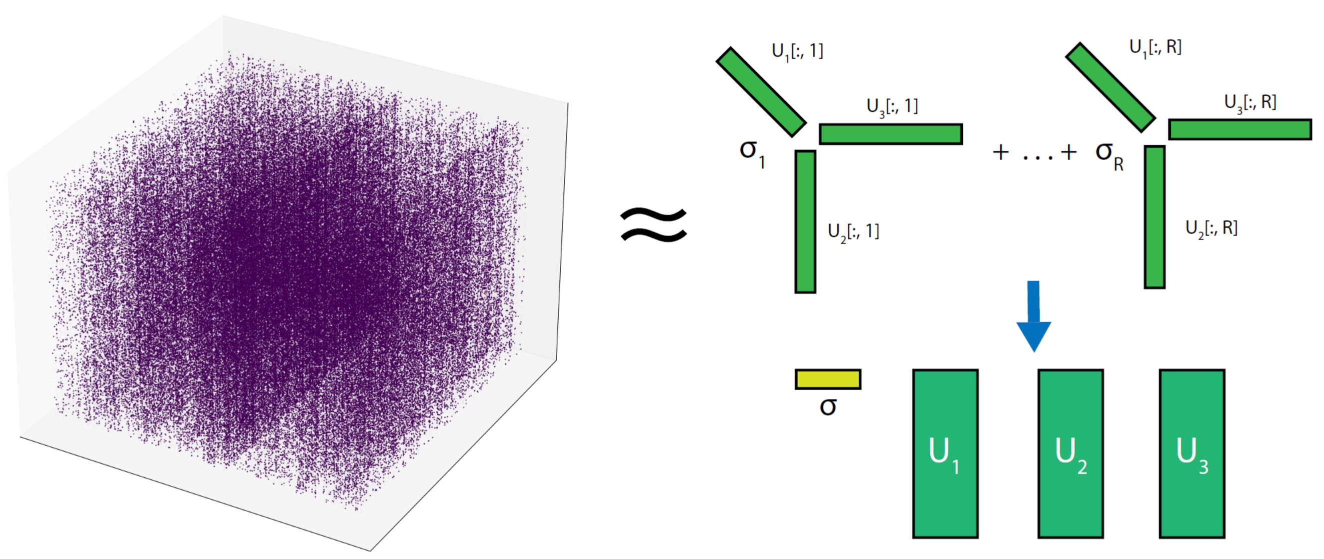
[1] UC Berkeley, [2] Lawrence Berkeley National Lab, [3] International Computer Science Institute, [4] Institute of Mathematics, EPFL & Lab for Simulation and Modelling, Paul Scherrer Institute

NEURAL INFORMATION PROCESSING SYSTEMS

U.S. DEPARTMENT OF ENERGY Office of Science

Sparse Tensor Decomposition and Khatri-Rao Products

Motivating Application: Given an $(N+1)$-dimensional sparse tensor $\mathcal{T}$, compute an approximate Candecomp / PARAFAC decomposition:



Decomposition of rank R consists of factor matrices $U_j \in \mathbb{R}^{I_j \times R}$, $1 \leq j \leq N+1$ to store the outer product components, vector $\sigma \in \mathbb{R}^R$ to store generalized singular values. Want to capture values of nonzero entries AND locations of zero entries. Problem is non-convex and NP-hard.

Alternating Least-Squares (ALS): Iteratively optimize one factor at a time while keeping the others constant (also called block coordinate descent). Optimization problem for U_{N+1} is an overdetermined linear least-squares problem

$$\min_X \|AX - B\|_F \qquad (1)$$

where $A = U_N \odot \ldots \odot U_1$, B is a sparse matrix. $\odot$ denotes a Khatri-Rao Product (KRP), a column-wise Kronecker Product of two matrices:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \odot \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} aw & bx \\ cw & dx \\ ay & bz \\ cy & dz \end{bmatrix}$$

Least-squares problems of this form also arise in PDE-inverse problems, signal processing, and compressed sensing.

Leverage Score Sampling

Problem: Design matrix A from Equation (1) has $\prod_{j=1}^N I_j$ rows. May not even fit in memory!

Solution: Generate random sampling matrix S with $J \ll \prod_{j=1}^N I_j$ rows, solve

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

Choosing S as a sampling matrix preserves sparsity of B. To guarantee residual within $(1+\varepsilon)$ of true minimum w.h.p. $(1-\delta)$, sample $\tilde{O}(R/(\varepsilon\delta))$ rows proportional to leverage scores:

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

Central Challenge: How do we sample rows according to the leverage score distribution of A when even materializing A is too expensive?

Our Contributions

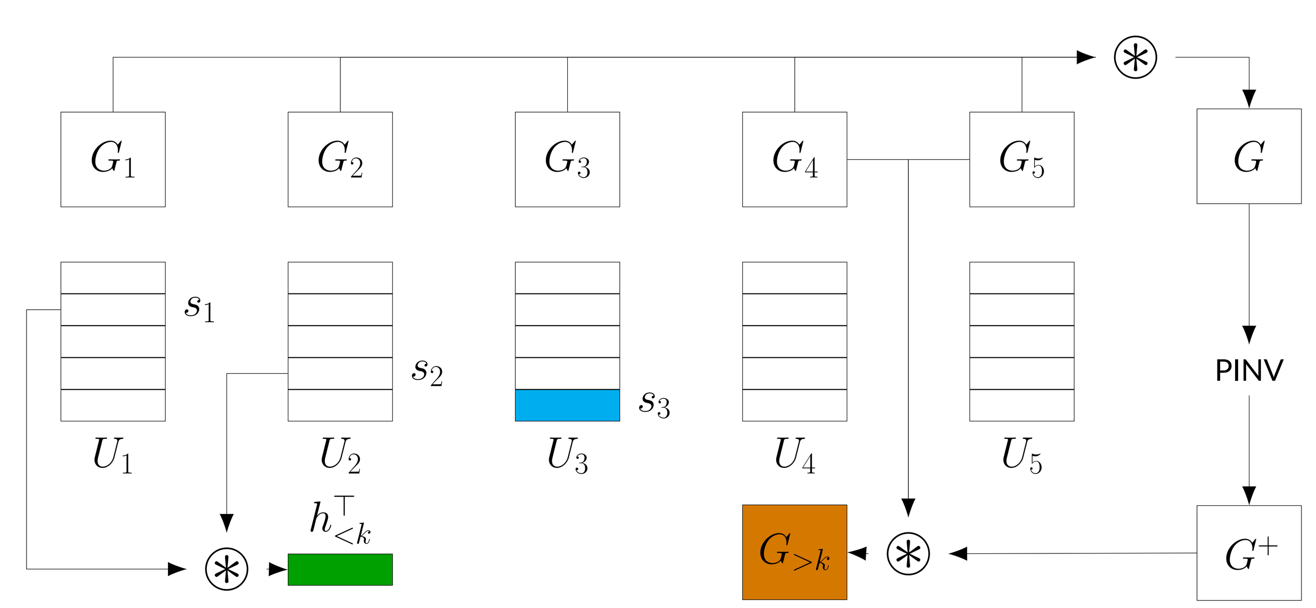
We build a data structure to sample rows from the exponentially tall matrix A in time logarithmic in its row count and quadratic in its column count from the exact leverage score distribution. To decompose an $(N+1)$-dimensional tensor, our method achieves the lowest asymptotic runtime for sketched CP decomposition compared to recent SOTA methods.

Method	Complexity per ALS Round
CP-ALS [1]	$N(N+I)I^N R$
CP-ARLS-LEV [2]	$N(R+I)R^{N+1}/(\varepsilon\delta)$
TNS-CP [3]	$N^3 I R^3/(\varepsilon\delta)$
GTNE [4]	$N^2(N^{1.5}R^{3.5}/\varepsilon^3 + IR^2)/\varepsilon^2$
STS-CP (ours)	$N(NR^3 \log I + IR^2)/(\varepsilon\delta)$

Our method accelerates decomposition of sparse tensors with billions of nonzero entries.

Methodology

Sample rows from $U_1, \ldots, U_N$ in sequence, each conditioned on the last.

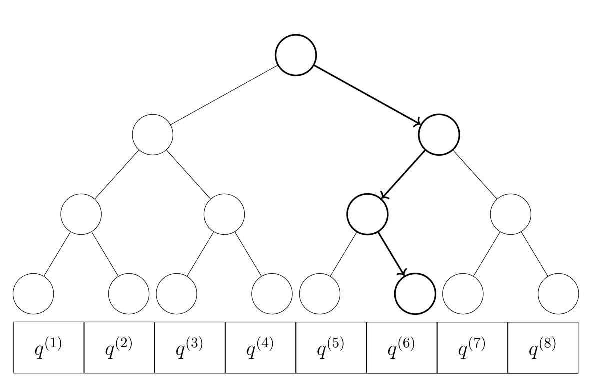


Conditional Distribution: $q[s_k] = p(\hat{s}_k = s_k \mid \hat{s}_{<k} = s_{<k}) \propto \langle h_{<k} h_{<k}^\top, U_k[s_k,:]^\top U_k[s_k,:], G_{>k} \rangle$

Sampling Procedure: Sample $r \sim \mathrm{Unif}[0,1]$, find "containing bin" of width q_j via **binary search.**

Root: branch right iff $\sum_{j=0}^{I_k/2} q[j] < r$

Level 2: branch right iff $\sum_{j=0}^{I_k/2} q[j] + \sum_{j=I_k/2}^{3I_k/4} q[j] < r$...



Key: For nodes v in search tree corresponding to row interval $[S_0(v), S_1(v)]$ (up to level $\log(I_k/R)$), compute and store "partial gram matrix":

$$G^v = \sum_{i=S_0(v)}^{S_1(v)} U_k[i,:]^\top U_k[i,:]$$

Construction runtime is $O(I_k R^2)$ with storage requirement $O(I_k R)$. During sampling, computing the branch decision at each internal node costs $O(R^2)$ with cached partial gram matrices. $O(R^3)$ work required below level $\log(I_k/R)$, but can improve to $O(R^2 \log R)$ (see paper). Total time per sample is $O(R^2 \log I_k)$.

Runtime Benchmarks

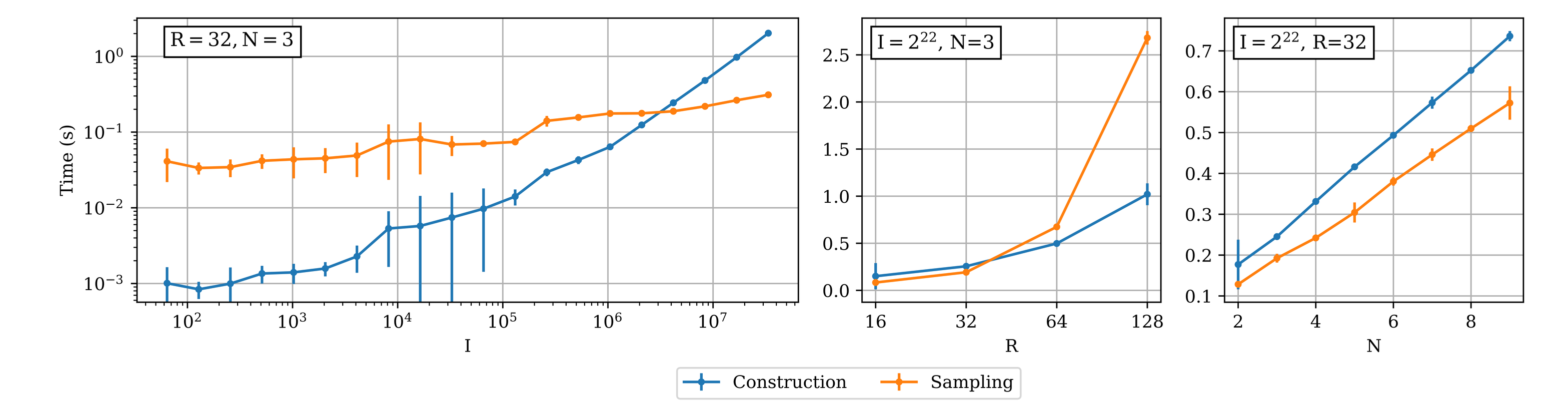


Figure 1. Average Time to Construct Data Structure and Draw 50,000 Samples from Khatri-Rao Product.

Sparse Tensor Experiments

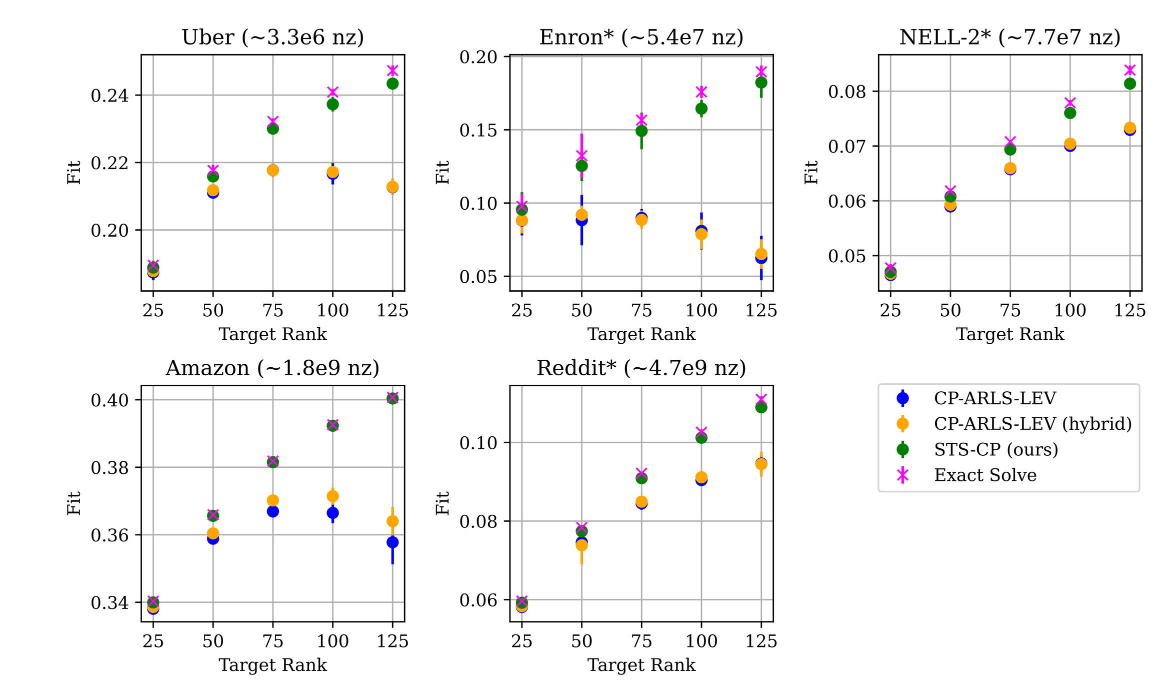


Figure 2. Accuracy Achieved by CP-ARLS-LEV, STS-CP, and Exact ALS on Sparse Tensor Decomposition, $J = 2^{16}$ samples for randomized algorithms.

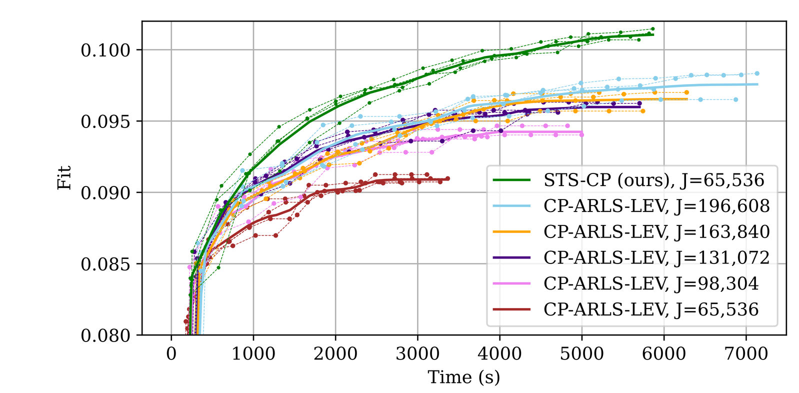


Figure 3. Fit vs. Time, Reddit Tensor (4.8 billion nonzeros) for CP-ARLS-LEV and STS-CP (ours). Thick lines are averages of individual traces.

Acknowledgements and References

We thank the referees for valuable feedback. The authors of this work were funded by the Department of Energy Office of Science and the National Science Foundation.

[1] T. Kolda, B. Bader. Tensor decompositions and applications. SIAM Review, Aug. 2009.

[2] B. Larsen, T. Kolda. Practical leverage-based sampling for low-rank tensor decomposition. SIAM J. Matrix Analysis and Applications, Aug. 2022.

[3] O. A. Malik. More efficient sampling for tensor decomposition with worst-case guarantees. ICML, July 2022.

[4] L. Ma and E. Solomonik. Cost-efficient Gaussian tensor network embeddings for tensor-structured inputs. NeurIPS, Nov. 2022.