BERKELEY LAB

Sampling-Based Sketches for Tensor Train Core Chains

MS12: Applications of Tensors in Scientific Computing, Part III

May 13, 2024

Vivek Bharadwaj [1, 2]

[1] University of California, Berkeley
[2] Lawrence Berkeley National Laboratory

Collaborators and Published Work

This presentation covers work in progress with **Beheshteh Rakhshan**, **Guillaume Rabusseau** (MILA Quebec), and **Osman Asif Malik** (formerly Lawrence Berkeley, now Encube Technologies).

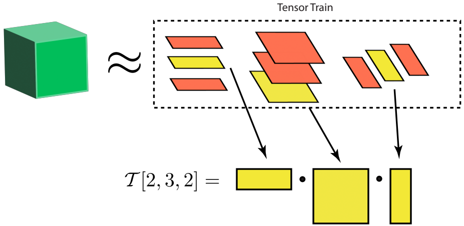
The material is tied closely to two recent papers with **Riley Murray** (Sandia), **Laura Grigori** (EPFL), **Aydın Buluç** (LBNL), and **James Demmel** (UC Berkeley):

- [Bha+23] Fast Exact Leverage Score Sampling from Khatri-Rao Products with Applications to Tensor Decomposition. In NeurIPS 2023.
- [Bha+24] Distributed-Memory Randomized Algorithms for Sparse Tensor CP Decomposition. To appear in SPAA 2024.

Introduction

Tensor Trains / Matrix Product States

A **tensor train (TT)** represents a tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ as a contraction of three-dimensional tensor cores $\mathcal{A}_1, \ldots, \mathcal{A}_N$.



Cores have dimensions $\mathcal{A}_k \in \mathbb{R}^{R_{k-1} \times I_k \times R_k}$, $1 \leq k \leq N$, and we impose $R_0 = R_N = 1$. Cores can represent a tensor with I^N elements using $O(NIR^2)$ space.

Tensor Diagram Notation

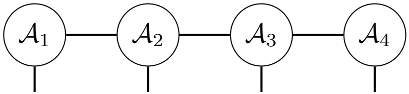


Figure: A 4D tensor train.

Tensor trains compactly represent high-dimensional tensors and even large vectors / matrices (by first folding them up into high-dimensional tensors).

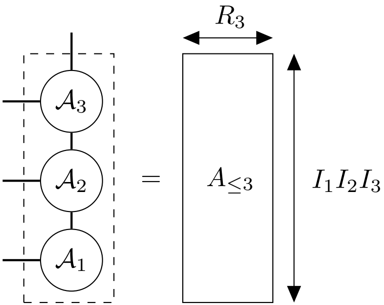
Example applications:

- Ground state calculation for MPO Hamiltonians [Gel17].
- Krylov methods with vectors in TT format [Al +23].
- Vlasov equation, high-dimensional PDE solvers [YL22].

Our Task: Sketching a Core Chain

- Consider cores $\mathcal{A}_1, ..., \mathcal{A}_j$, let $A_{\leq j}$ be the *matricization* of the chain.
- Want a linear map (sketch) that reduces row count of $A_{\leq j}$, preserves column space geometry.
- An (ε, δ)-**subspace embedding** is a distribution over maps $S \in \mathbb{R}^{J \times \prod_{k \leq j} I_k}$. For all $x \in \mathbb{R}^{R_j}$ with high probability $1 - \delta$,

$$(1 - \varepsilon) \|A_{\leq j} x\|_2^2 \leq \|S A_{\leq j} x\|_2^2 \leq (1 + \varepsilon) \|A_{\leq j} x\|_2^2$$



Our Contributions

When each input core has a property called left-orthonormality, we give an algorithm to construct an efficient subspace embedding by sampling rows from $A_{\leq j}$.

Theorem (Core Chain Subspace Embedding)

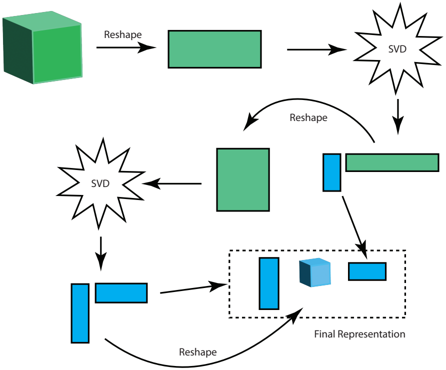
Given left-orthonormal tensor cores $\mathcal{A}_1, ..., \mathcal{A}_j$, *assume for simplicity* $I_1 = ... = I_j = I$ *and* $R_1 = ... = R_j = R$. *There exists a data structure with the following properties:*

1. *The DS has construction time* $O(IR^3)$ *with space overhead linear in the input core sizes.*
2. *The DS randomly draws a single row from* $A_{\leq j}$ *proportional to its squared row norm in time* $O(jR^2 \log I)$.

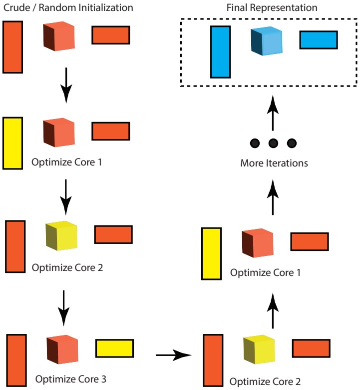
With this data structure, only $J = O\left(\frac{R}{\varepsilon^2} \log\left(\frac{R}{\delta}\right)\right)$ samples are need for an (ε, δ)-SE.

Context and Prior Work

Tensor Train Decomposition

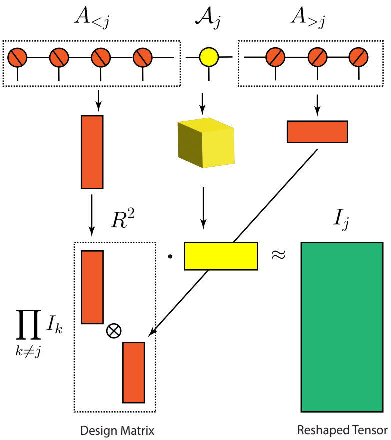


SVD-Based Algorithm



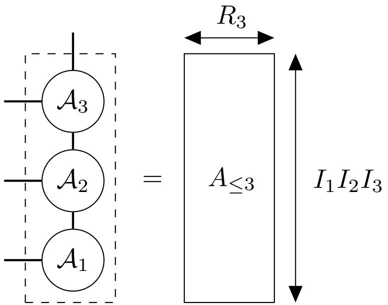
Iterative Algorithm

Sketching Application 1: ALS Fitting



Sketching Application 2: TT Rounding*

- Many operations on TTs (addition, multiplication by matrix-product operator) inflate the rank R.
- Want an operation to recompress the TT to some lower rank r. Randomized algorithms are particularly effective!
- **Main operation:** Gram matrix estimation of $A_{\leq j}$ for $1 \leq j < N$. Key ingredient is a structured sketch S to reduce row count of $A_{\leq j}$

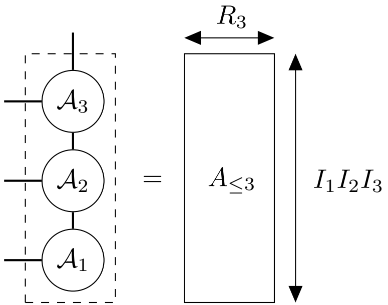


Sketching Application 2: TT Rounding*

- Two excellent papers provide upper / lower bounds on complexity of random TT-rounding:

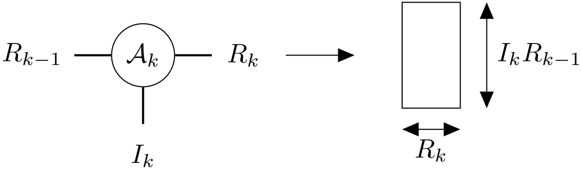
    - **Algorithm:** Randomized Algorithms for Rounding in the Tensor-Train Format. Al Daas et. al. [Al +23].
    - **Lower Bound:** Cost-efficient Gaussian tensor network embeddings for tensor-structured inputs . Ma and Solomonik [MS22].

- **Caveat*:** Our result cannot accelerate this application, since we rely on the left-orthonormality property.



The Left-Orthonormality Condition

The operation $\mathrm{mat}(\mathcal{A}_k, 3)$ is a flattening of $\mathcal{A}_k$ into a matrix:



Core $\mathcal{A}_k$ is **left-orthonormal** if $A_k^L = \mathrm{mat}(\mathcal{A}_k, 3)$ is orthonormal, i.e. $A_k^{L\top} A_k^L = I$.

Proposition (Left-Orthonormal Core Chain)

If cores $\mathcal{A}_1, ..., \mathcal{A}_j$ are left-orthonormal, the matrix $A_{\leq j}$ is orthonormal.

Row-Norm Squared Sampling

We will sample rows i.i.d. from matrix $A_{\leq j}$. The i-th row is sampled with probability

$$p_i = \frac{1}{R} \left\| A_{\leq j}\left[i, :\right] \right\|^2$$

Theorem ([Woo14], Adapted)

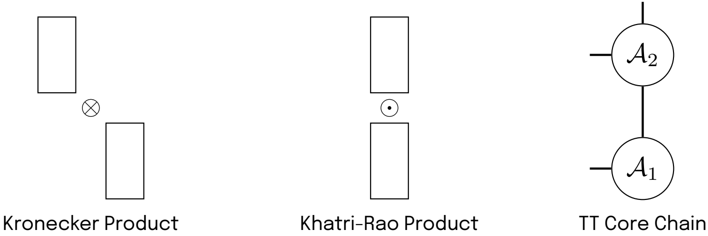
Let $S \in \mathbb{R}^{J \times \prod_{k \leq j} I_k}$ be a sampling matrix for orthonormal matrix $A_{\leq j}$ that selects rows i.i.d. according to their squared row norms (reweighting them appropriately). There exists constant C so if

$$J \geq CR \frac{\log(2R/\delta)}{\varepsilon^2},$$

then S is an (ε, δ)-subspace embedding for $A_{\leq j}$.

Sampling from Other Tensor Products

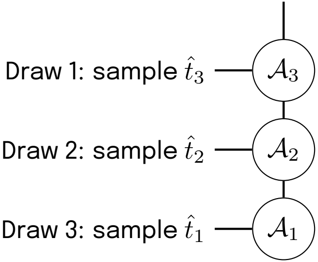
- Row-norm-squared sampling from an Kronecker product is trivial (sample independently from each matrix).
- Slightly more complicated for Khatri-Rao product, but doable (use ideas from [DYH19]).
- We are first to demonstrate efficient sampling from left-orthonormal TT core chains.



Efficient Core Chain Sketching

Conditional Sampling from $A_{\leq j}$

- To draw one row from $A_{\leq j}$, we sample one slice from each core $\mathcal{A}_j, \mathcal{A}_{j-1}, ...\mathcal{A}_1$. Let $\hat{t}_j, ..., \hat{t}_1$ be RVs for each index.
- The product of these slices forms a row from $A_{\leq j}$. Sample each index $\hat{t}_k$ conditioned on $\hat{t}_{k+1}, ..., \hat{t}_j$.
- The order of sampling is counterintuitive; the most efficient matrix multiplication order to materialize a row from $A_{\leq j}$ starts by slicing $\mathcal{A}_1$.



Step 1: Sample Column Uniformly from $A_{\leq j}$

- Let the target distribution be

$$q := \frac{1}{R}\left(A_{\leq j}\left[:, 1\right]^2 + \ldots + A_{\leq j}\left[:, R\right]^2\right)$$

- q has the form of a **mixture distribution**. Can sample a column uniformly at random, then restrict ourselves to sampling from the squared entry distribution on that column.

- We reap a **computational advantage** through this restriction.

Step 2: Form the Conditional Distribution

Suppose that we have selected column $\hat{r} = r$ and $\hat{t}_{k+1} = t_{k+1}, ..., \hat{t}_j = t_j$ for index $k \leq j$. Define "history vector" $h_{>k} \in \mathbb{R}^R$ as

$$h_{>k} := \mathcal{A}_{k+1}\left[:, t_{k+1}, :\right] \cdot ... \cdot \mathcal{A}_j\left[:, t_j, :\right] \cdot e_r$$

where e_r is the r-th standard basis vector.

Lemma (Conditional distribution for $\hat{t}_k$)

Suppose we impose a conditional distribution on $\hat{t}_k$ given by

$$p(\hat{t}_k = t_k \mid \hat{t}_{>k} = t_{>k} \wedge \hat{r} = r) = \left\| \mathcal{A}_k\left[:, t_k, :\right] \cdot h_{>k} \right\|_2^2 .$$

Then the joint RV $(\hat{t}_1, ..., \hat{t}_j)$ follows the squared row norm distribution on $A_{\leq j}$.

Note: without step 1, $h_{>k}$ would be a matrix.

Step 3: Sample the Conditional Distribution

We have a data structure to efficiently sample from the prior distribution! Flatten core $\mathcal{A}_k$ into its left-matricization, apply the following lemma:

Lemma ([Bha+23], Adapted)

Given a matrix $A \in \mathbb{R}^{IR \times R}$, there exists a data structure with the following properties:

- *Its construction time is $O(IR^3)$ with space overhead $O(IR^2)$.*
- *Given any vector $h \in \mathbb{R}^R$, it can draw a single sample from the un-normalized distribution of weights $(A \cdot h)^2$ in time $O(R^2 \log I)$.*

Experiments and Further Work

Verifying Sampler Correctness

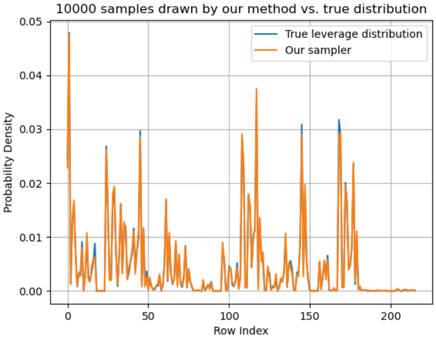


Figure: Sampling from the left subchain of an $16 \times 16 \times 16$ TT-tensor with rank 4.

FROSTT Sparse Tensor Train ALS Fitting

Tensor	Dimensions	NNZ	Prep.
Uber	183 x 24 x 1.1K x 1.7K	3.3M	-
Enron	6K x 5.7K x 244K x 1.2K	54M	log
NELL-2	12K x 9.1K $\times$ 29K	77M	-

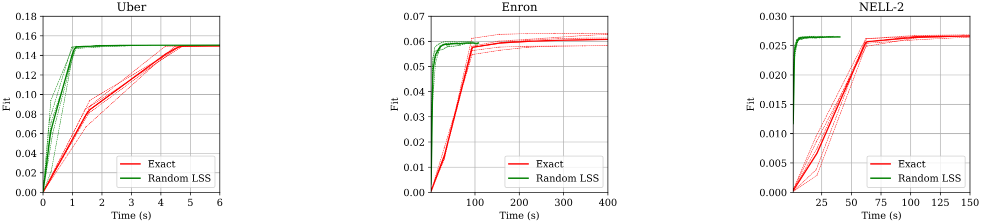


Figure: Accuracy vs. time for three FROSTT tensors, $R = 6$, $J = 2^{16}$ for our randomized ALS algorithm.

Accuracy and Per-Iteration Speedup

Tensor	R	RS Fit	Exact ALS Fit	Avg. Speedup of RS over Exact
	4	0.1332	0.1334	4.0x
	8	0.1646	0.1654	3.0x
Uber	12	0.1828	0.1846	1.5x
	4	0.0498	0.0507	17.8x
	8	0.0669	0.0711	10.5x
Enron	12	0.0810	0.0856	7.4x
	4	0.0213	0.0214	26.0x
	8	0.0311	0.0317	22.2x
NELL-2	12	0.0382	0.0394	15.8x

Table: Average Fits and speedup, $J = 2^{16}$ for randomized algorithms, 40 ALS iterations.

Work in Progress

- Row-norm squared sampling is simple for Kronecker and Khatri-Rao products. Our work shows that it is also efficient for TT chains.
- We are actively searching for other (low-error) sparse tensors and other applications for our subspace embedding algorithm.
- Want to develop related tools for non-orthogonal chains, if possible. Could accelerate tensor train rounding.
- If any of these techniques / results interest you, please come talk to me!

Thank you, questions welcome.

References I

[Al +23] Hussam Al Daas, Grey Ballard, Paul Cazeaux, Eric Hallman, Agnieszka Międlar, Mirjeta Pasha, Tim W. Reid, and Arvind K. Saibaba. “Randomized Algorithms for Rounding in the Tensor-Train Format”. In: *SIAM Journal on Scientific Computing* 45.1 (2023), A74–A95. DOI: 10.1137/21M1451191.

[Bha+23] Vivek Bharadwaj, Osman Asif Malik, Riley Murray, Laura Grigori, Aydın Buluç, and James Demmel. “Fast Exact Leverage Score Sampling from Khatri-Rao Products with Applications to Tensor Decomposition”. In: *Thirty-seventh Conference on Neural Information Processing Systems*. Dec. 2023.

[Bha+24] Vivek Bharadwaj, Osman Asif Malik, Riley Murray, Laura Grigori, Aydın Buluç, and James Demmel. “Distributed-Memory Randomized Algorithms for Sparse Tensor CP Decomposition”. In: *Proceedings of the 36th ACM Symposium on Parallelism in Algorithms and Architectures*. SPAA ’24. Nantes, France: Association for Computing Machinery, 2024.

[DYH19] QIN DING, Hsiang-Fu Yu, and Cho-Jui Hsieh. “A Fast Sampling Algorithm for Maximum Inner Product Search”. In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Ed. by Kamalika Chaudhuri and Masashi Sugiyama. Vol. 89. Proceedings of Machine Learning Research. PMLR, Apr. 2019, pp. 3004–3012.

References II

[Gel17] Patrick Gelß. “The tensor-train format and its applications”. PhD thesis. Freien Universität Berlin, 2017.

[Mal22] Osman Asif Malik. “More Efficient Sampling for Tensor Decomposition With Worst-Case Guarantees”. In: *Proceedings of the 39th International Conference on Machine Learning*. Vol. 162. Proceedings of Machine Learning Research. PMLR, July 2022, pp. 14887–14917.

[MS22] Linjian Ma and Edgar Solomonik. “Cost-efficient Gaussian tensor network embeddings for tensor-structured inputs”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 38980–38993.

[Woo14] David P. Woodruff. “Sketching as a Tool for Numerical Linear Algebra”. In: *Foundations and Trends® in Theoretical Computer Science* 10.1 (2014), pp. 1–157. ISSN: 1551-305X, 1551-3068. DOI: `10.1561/0400000060`. URL: `http://arxiv.org/abs/1411.4357`.

[YL22] Erika Ye and Nuno F. G. Loureiro. “Quantum-inspired method for solving the Vlasov-Poisson equations”. In: *Phys. Rev. E* 106 (3 Sept. 2022), p. 035208. DOI: `10.1103/PhysRevE.106.035208`.

Backup Slides

Oblivious Subspace Embeddings

- An **oblivious** subspace embedding doesn't require any prior information about A.
- Choose S as:
  - An i.i.d. Gaussian / Rademacher random matrix
  - A Countsketch / Sparse Sign Embedding (fixed nnz per column)
  - A composition of a random diagonal, FFT-like operator, and uniform sparse sampler

i.i.d. Gaussian

$$\begin{bmatrix} -0.01 & -0.39 & 0.37 \\ -0.47 & 0.74 & -0.10 \end{bmatrix}$$

Countsketch

$$\begin{bmatrix} +1 & 0 & +1 \\ 0 & -1 & 0 \end{bmatrix}$$

Leverage Scores and Linear Least-Squares

When $A_{\leq j}$ is orthonormal, the squared norm of each row is equal to its **leverage score**. Leverage score sampling can accelerate linear least squares:

Theorem (Leverage Score Sampling Guarantees, [Mal22])

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* $1 - \delta$,

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