Efficient Leverage Score sampling for Tensor Train Decomposition

Vivek Bharadwaj[*,1]
Behesnteh T. Rakhshan[*,2]
Osman Asif Malik[3]
Guillaume Rabusseau[2,4]

[1] UC Berkeley
[2] Mila, Udem
[3] Lawrence National Lab
[4] CIFAR

Université de Montréal
U.S. DEPARTMENT OF ENERGY | Office of Science
BERKELEY LAB
Mila
CIFAR
IVADO

How to compute Tensor Train Decomposition?

- TT_SVD
- Randomized TT_SVD
- TT_ALS
- Randomized TT_ALS (proposal)

Why Tensor Train Decomposition?

- # of parameters $O(NIR^2)$ instead of $O(I^N)$

  order of a Tensor (N)

  dimension size ($I_1 = I_2 = \dots = I_N = I$)

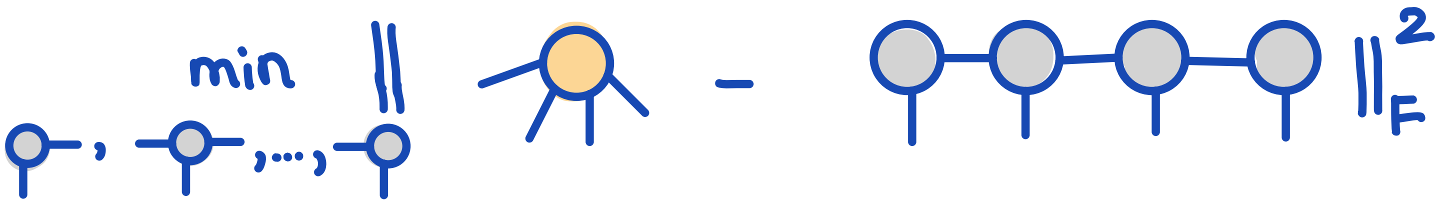
  rank of decomposition (R)

- Finding a good approximation is feasible.
- Numerically Stable.

Why Randomized Alternating Least Squares (ALS)?

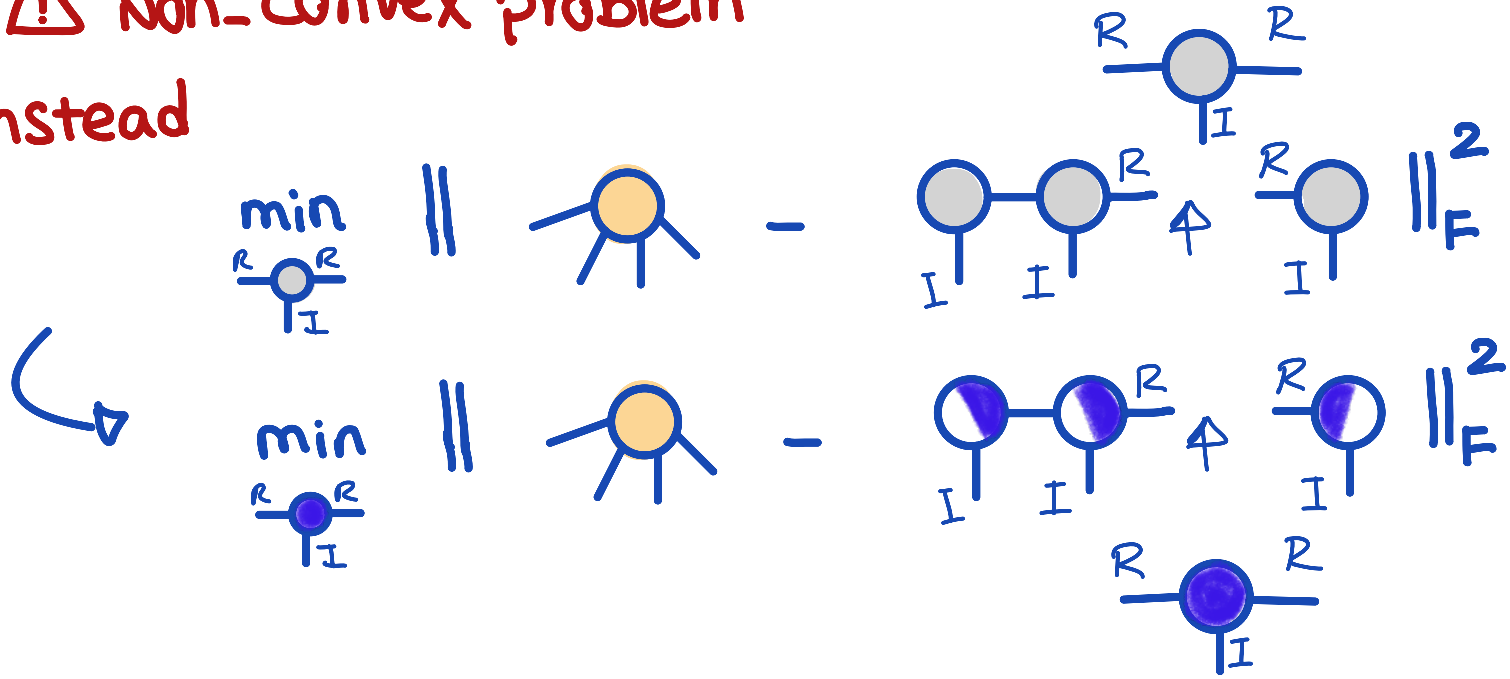
- Lack of randomized ALS methods to find a Tensor Train Decomposition.

Goal:

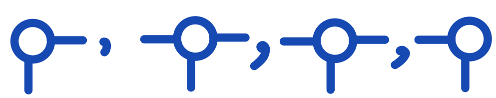


⚠ Non_convex problem

Instead



How?

- ✓ Initialize  randomly.
- ✓ Update one core at a time until convergence.

$$\min_{A_j} \| (A^{<j} \otimes A^{>j^T})(A_j)^T_{(2)} - X^T_{(j)} \|_F^2$$

⚠ cost $O(I^N)$ to solve Least_squares

Randomized Tensor Train ALS (proposal)

- General Randomized Leas_squares

$$\min_X \| AX - b \|^2 \xrightarrow{\text{instead}} \min \| SAX - Sb \|^2$$

- Randomized Tensor Train Least_squares

$$\min_{A_j} \| S(A^{<j} \otimes A^{>j^T})(A_j^T)_{(2)} - SX^T_{(j)} \|_F^2$$

S is a sampling matrix

↳ How to construct S ?

with Leverage scores ; $\mathbb{P}_i \propto A[i,:]\underbrace{(A^TA)^+}_{\text{orthogonal}} A[i,:]^T$

⇝ no cost to compute

In Tensor Train Case:

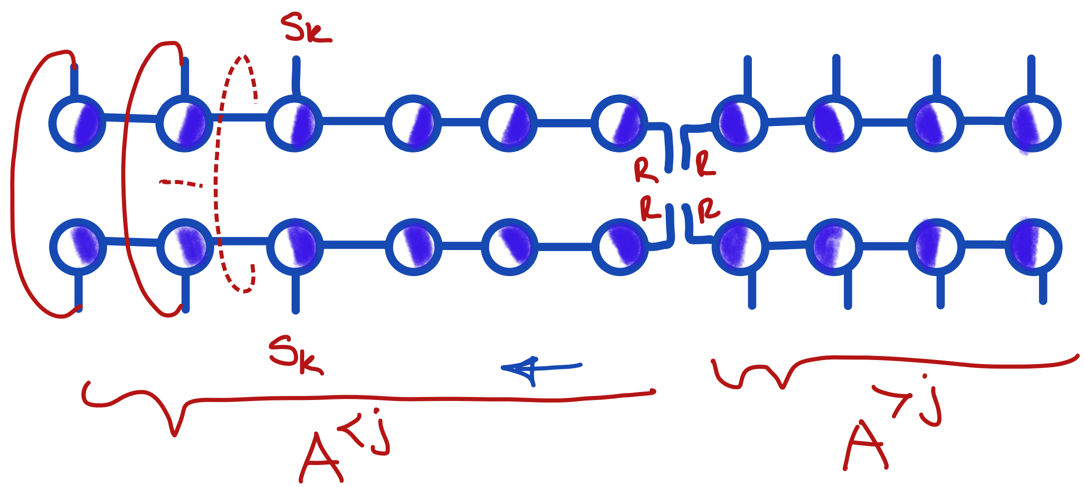
Compute $\mathbb{P}_i \propto A^{\neq j}[i^{\neq j},:]\, A^{\neq j}[i^{\neq j},:]^T$

↳ Still is a challenge.

Solution

Suppose we have drawn $\hat{s}_{j-1}=s_{j-1}, \dots, \hat{s}_{k+1}=s_{k+1}$ and now want to draw $\hat{s}_k = s_k$

$$\mathbb{P}(\hat{s}_k = s_k \mid \hat{s}_{j-1}=s_{j-1}, \dots, \hat{s}_{k+1}=s_{k+1}) = \frac{\mathbb{P}(\hat{s}_k = s_k, \dots, \hat{s}_{j-1}=s_{j-1})}{\mathbb{P}(\hat{s}_{k+1}=s_{k+1}, \dots, \hat{s}_{j-1}=s_{j-1})}$$



$$\rightsquigarrow \quad \mathbb{P}(\hat{S}_k = s_k \mid \hat{S}_{>k} = s_{>k}) \propto$$

$$\mathrm{Tr}\left(H_{>k}^T A_k[:, s_k, :]^T A[:, s_k, :] H_{>k} \right)$$

where

$$H_{>k} = A_{k+1}[:, s_{k+1}, :] \ldots A_{j-1}[:, s_{j-1}, :]$$

⚠ Updating $H_{>k}$ cost $O(R^3)$

Define

$$q = \frac{1}{R}\left(L[:,1]^2 + \ldots + L[:,R]^2 \right) \quad \text{where} \quad L = A^{<j}$$

✓ Sample a column uniformly, $\hat{t} = t$

✓ Sample a row from $L[:, t]^2$

↳ Reduce the cost to $O(R^2)$

For any $\varepsilon_p, \delta \in (0,1)$ the sampling procedure above guarantees that with $J = \tilde{O}(R^2/\varepsilon_p \delta)$ sample per Least_Square problem

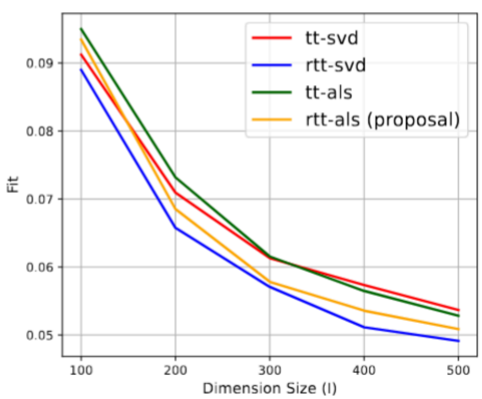
$$\| A^{\neq j} (\tilde{A}_j)^T_{(2)} - X^T_{(j)} \| \leq (1+\varepsilon_p) \min \| A^{\neq j} (A_j)^T_{(2)} - X^T_{(j)} \|$$

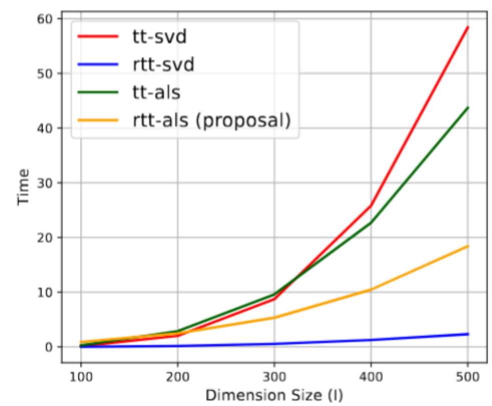
The overall complexity

$$O\left(\frac{\#\text{iter}}{\varepsilon_p \delta} R^4 \sum_{j=1}^{N} N \log I_j + I_j \right)$$

Experiments :

✓ Synthetic Dense Tensors (J=5000, 5 trials)



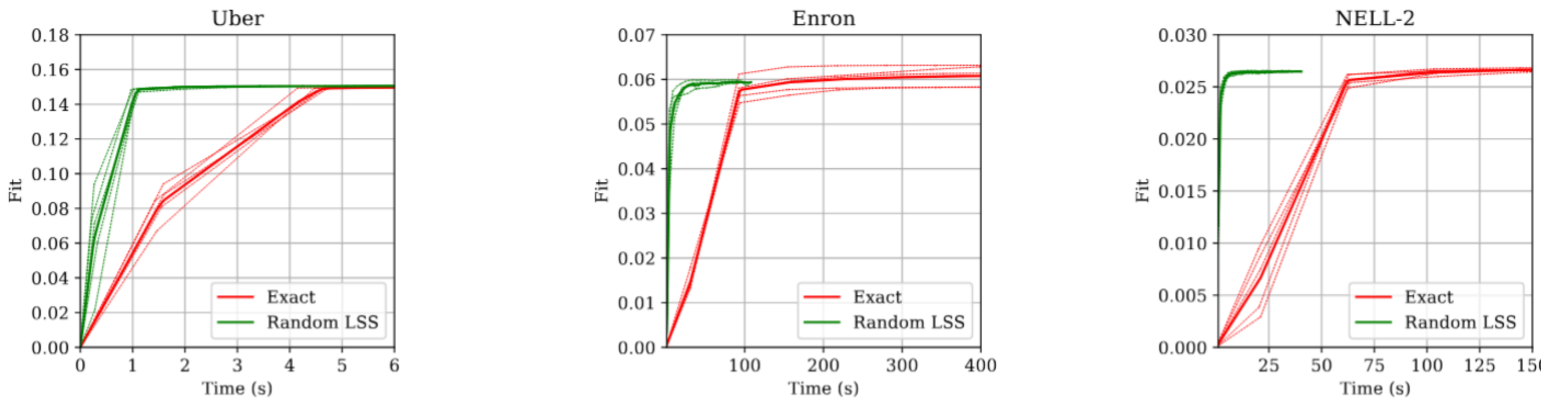


✓ Real Data ($J=2000$, target rank $\tilde{R}=5$)

Method	Pavia Uni. Fit	Pavia Uni. Time	Tabby Cat Fit	Tabby Cat Time	MNIST Fit	MNIST Time	DC Mall Fit	DC Mall Time
TT-ALS	0.61	4.16	0.65	44.570	0.46	8.29	0.59	21.86
rTT-ALS (proposal)	0.60	0.82	0.65	7.360	0.45	2.20	0.59	2.81
TT-SVD	0.61	6.65	0.65	136.189	0.46	17.19	0.59	41.45
rTT-SVD	0.61	0.33	0.65	4.285	0.46	0.65	0.59	0.46

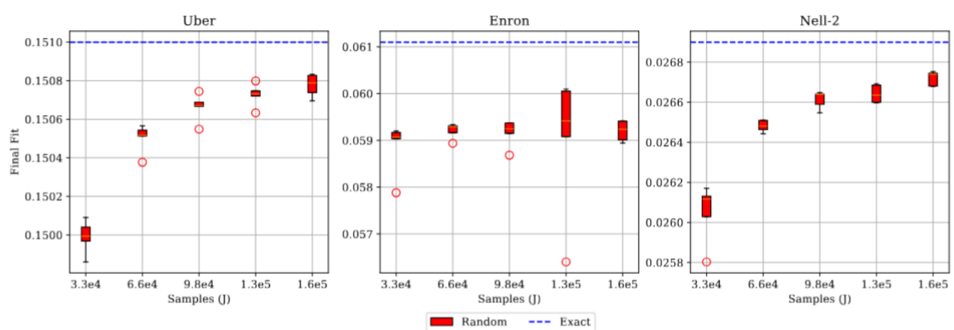
✓ Sparse Tensors

Fit vs time R=6, J=16



R	Uber rTT-ALS	Uber TT-ALS	Uber Speedup	Enron rTT-ALS	Enron TT-ALS	Enron Speedup	NELL-2 rTT-ALS	NELL-2 TT-ALS	NELL-2 Speedup
4	0.1332	0.1334	4.0x	0.0498	0.0507	17.8x	0.0213	0.0214	26.0x
6	0.1505	0.1510	3.5x	0.0594	0.0611	12.4x	0.0265	0.0269	22.8x
8	0.1646	0.1654	3.0x	0.0669	0.0711	10.5x	0.0311	0.0317	22.2x
10	0.1747	0.1760	2.4x	0.0728	0.0771	8.5x	0.0350	0.0359	20.5x
12	0.1828	0.1846	1.5x	0.0810	0.0856	7.4x	0.0382	0.0394	15.8x

J=16
40 iters



Fit vs # of samples

Thank you !