A Practical Approach to Randomized Tensor decomposition

Kartik Srinivas, Ganesh Bombatkar, Tanmay Garg, Aayush Patel, Ojjas Tyagi

March 2023

Abstract

We summarize our findings and our experimental results on the following paper [BBK17]. We implement the algorithms mentioned in it and explain the motivation for the algorithms in a simple and intuitive way.

1 Notation

Table 1: Notations used

$\boldsymbol{\mathcal{X}}$	$\triangleq$	Tensor to be decomposed
A_i	$\triangleq$	Factor Matrices
$\boldsymbol{\lambda}$	$\triangleq$	Diagonal entries of Core Tensor $\boldsymbol{\Lambda}$
$\mu(A)$	$\triangleq$	Coherence of the Matrix A
$\mathcal{Q}$	$\triangleq$	Unitary Matrix
F_i	$\triangleq$	DFT Matrix, WHT or DCT Matrix
D_i	$\triangleq$	Diagonal Matrix with Rademacher $(+1, -1)$ variables as the diagonals
S	$\triangleq$	Row sampling Matrix for Pre-multiplication
		Product Notations
$\bigotimes_{m \neq n} A_i$	$\triangleq$	$A^{(N)} \otimes A^{(N-1)} \dots A^{(m+1)} \otimes A^{(m-1)} \dots \otimes A^{(1)}$
$\circledast_{m \neq n} A_i$	$\triangleq$	$A^{(N)} \circledast A^{(N-1)} \dots A^{(m+1)} \circledast A^{(m-1)} \dots \circledast A^{(1)}$
$\bigodot_{m \neq n} A_i$	$\triangleq$	$A^{(N)} \odot A^{(N-1)} \dots A^{(m+1)} \odot A^{(m-1)} \dots \odot A^{(1)}$

2 Coherence and Mixing

The motivation is to come up with a randomized algorithm for solving the least squares objective in the CP-ALS decomposition. In general, the least squares problem takes the following form

$$\min_C \|AC^T - B\|$$

Sketching is a technique where a problem is to be solved, and you solve an alternative problem whose solution converges to yours with high probability (convergence in probability).
The idea for CP-rand is to simply sample some rows from both terms inside the norm, i.e , instead solve the following problem:-

$$\min_C \|SAC^T - SB\|$$

where S is a sampling matrix that when premultiplied onto some A returns a matrix with only those corresponding rows. You can read this for further details Mahoney The solution, will not be exact, but the accuracy of the approximation can be analyzed after we define the below two terms.
Leverage l_i: The leverage score is calculated for an overdetermined full-column rank matrix $A \in \mathbb{R}^{n \times d}$. Let the Thin SVD of this matrix be equal to $U_1 \Sigma_1 V_1^T$, where $U_1 \in \mathbb{H}^{n \times d}$. The leverage score l_i is the norm of the i'th row of U_1.

Coherence μ: Coherence is defined as the maximum possible leverage of a particular row in U_1. It is denoted by $\mu(A)$

Consider an example:- Let A have only a single nonzero entry in the (1,1) position, then its rank is simply one. In case I wished to approximate A using some selection of the rows of A, the first row of A would surely be selected. Not choosing this row will lead to a **rank deficiency**. This will make the solution of the sampled least squares problem inaccurate.

If $\mu(A)$ becomes close to 1, then the leverage scores of all the other rows must be lesser than one and all add to the value of $d-1$, (but there are $n-1$ remaining rows!) (Because the Frobenius norm of $U_1 = d$) This means that the *Importance of columns of A is not uniformly distributed across the rows.* Hence, Random row sampling may lead to incorrect results.

2.1 SFDA Mixing

The idea behind this mixing strategy is the follows, First we flip the signs of $A's$ rows using a Diagonal Matrix with ones and minus ones with equal probability. After this we apply the transformation F, which is a supposed "mixing + projection operation". Like a Randomized JLM or a FFT. You can read more about it here [DMMS07].

After this we sample from the rows of FDA that is equivalent to just using one hot vectors as the rows of S and pre-multiplying A with S.

Motivation for FFT is that the matrix multiplication can be done significantly faster using the FFT. The question -Why does the solution remain the same?

3 Algorithms

3.1 CP-ALS

The goal is to solve the least squares problem described by

$$\min_{A_n} \|\mathcal{X}_n - [[\mathbf{\Lambda}; A_1, A_2, \dots A_N]]\|$$

iteratively. We will matricize the term (since it does not change the objective)

$$\min_{A_n} \|\mathcal{X}_n - A_n \underbrace{(\bigodot_{m \neq n} A_m)^T}_{Z_n^T} \|$$

Then we apply a trick to form the MTTKRP(matricized $\mathcal{X}$ multiplied by the Khatri rao product $= \mathcal{X}_n Z_n$)

$$\min_{A_n} \|\text{MTTKRP} - A_n(\circledast_{m \neq n} A_m^T A_m)\|$$

Now we solve the least squares problem.

$$\boxed{A_n \approx \text{MTTKRP}(\circledast_{m \neq n} A_m^T A_m)^+}$$

3.2 CP-RAND

No mixing; only random rows are selected from the Matrix that is represented by supposed $Z_n = \bigodot_{m \neq n} A_m$. The main idea lies in the following two facts:-

- Sampling from the rows of Z_n can be seen as sampling from the Rows of each A_i and taking their Hadamard product.

$$Z_n(j,:) = \circledast A_{k_i}^{(i)}$$

Where the indices k_i are formed in a reverse lexicographic ordering (index of A_n varies the slowest) and $A^{(1)}$ index varies the fastest. The same indexing technique can be used for calculating the appropriate indices of the mode-n matricized version of $\boldsymbol{\mathcal{X}}$.

The indexing , will be the same to get a particular row, because of the following result:-

$$X^{(n)\ T}(j,:) = \mathbf{X}_{k_1,k_2,\ldots:,k_{n+1},\ldots k_n}$$

Using these tricks, I sample from the rows and solve the following problem:-

$$\min_{A_n} \|SZ^{(n)}A_n^T - SX_{(n)}^T\|$$

The time to solve this problem is much smaller and can be computed like this:-

3.3 Coherence of MTTKRP

Theorem 3.1.

$$\mu(A \otimes B) = \mu(A) \times \mu(B)$$
$$\mu(A \odot B) \leq \mu(A) \times \mu(B)$$

Now that we have established that these operations affect the net coherence of the product, we can see that since the matrix $Z^{(n)}$ is an MTTKRP, the value of the coherence of the matrix that we are sampling from **either drastically drops or drastically increases**(more with the order of the tensor). If it drops, then it would means that we are reaching uniformity in the leverage scores amongst the rows.

3.4 CP-RAND-MIX

, The algorithm is almost similar except that some important steps, such as a Fourier transform, is added to mixup the rows and spread the row leverages of the matrix. How does it spread? The supposed spreading of the leverages of the rows can be achieved by simply reducing the coherence. The following facts form the core reasoning behind the algorithm

- In case the factors A_i turn out to be highly coherent, then you must **mix** the rows *of each individual factor* A_i using a DFT Matrix F_i and a diagonal matrix D_i
- The product of these two matrices is unitary.
- Multiplying the least squares problem with a unitary matrix leaves it unchanged.
- Theorems that relate products of KRP and Kronecker product will help us aggregate and transform **per-term multiplications** into a net multiplication of the KRP(Khatri rao product) with a unitary matrix $\mathcal{Q}$. See 3.4

We will mutliply the mixing coefficients on each of the A_m so that we can reduce the coherence individually. Once this is done, we can take the Khatri rao product of each of the terms , which hopefully by our earlier theorems will lead to good drops of coherence in the MTTKRP. The per term mixing can be seen as an overall mixing, using the product rules of KRP and KP.

$$\bigodot_m F_m D_m A_m = (\bigotimes_m F_m D_m)(\bigodot A_m) = (\bigotimes_m F_m)(\bigotimes_m D_m) \bigodot A_m = \mathcal{Q}\mathbf{Z_{(n)}} \tag{1}$$

The problem now reduces to

$$\min_{A_n} \|\mathcal{Q}Z^{(n)}A_n^T - \mathcal{Q}X_{(n)}^T\|$$

After this we use the matricization trick for Kronecker products and just multiply by the inverse of the mixing coefficient for mode n to continue we let the matrix $\hat{\boldsymbol{\mathcal{X}}} = [[\boldsymbol{\mathcal{X}}, F_iD_i]]$, then note that the mode n matricization of $\hat{\boldsymbol{\mathcal{X}}}$ is precisely $= F_nD_n\mathcal{X}_{(n)}\mathcal{Q}^T$. From this we get the following result

$$\boxed{(D_n^* F_n^T \hat{\mathcal{X}}_{(n)})^T = \mathcal{Q}X_{(n)}^T}$$

Now we apply S for sampling and then we implement it using the same indexing trick that we applied in CP-RAND The final optimization problem then reduces to

$$\min_{A_n} \|S\mathcal{Q}Z^{(n)}\hat{A}_n^T - D_nF_n^*(S\hat{\mathcal{X}}_n)^T\|$$

4 Experiments

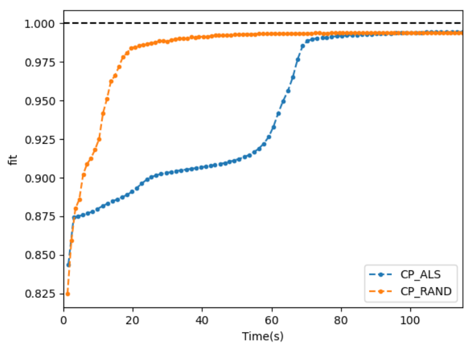
The code for the experiments can be found on Github. We have done the following experiments

Synthetic Data Generation

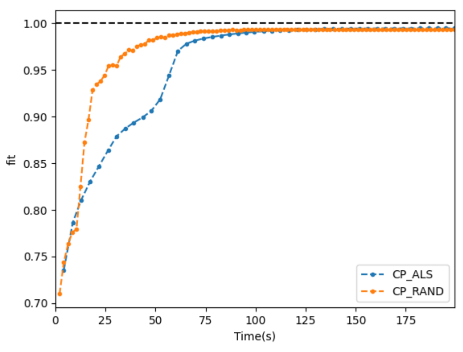
We have generated synthetic data for large Tensor using the following methods

1. Generated factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$, $n \in \{1, \ldots, N\}$
2. Computed
$$\mathcal{X}_{\text{true}} = \sum_{r=1}^{R_{\text{true}}} \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(N)}$$
3. Added noise $\mathcal{N} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ to $\mathcal{X}_{\text{true}}$ to get $\mathcal{X}$
$$\mathcal{X} = \mathcal{X}_{\text{true}} + \eta \left(\frac{\|\mathcal{X}_{\text{true}}\|}{\|\mathcal{N}\|} \right) \mathcal{N}$$

4.1 Fit Time of CPALS vs CP-RAND



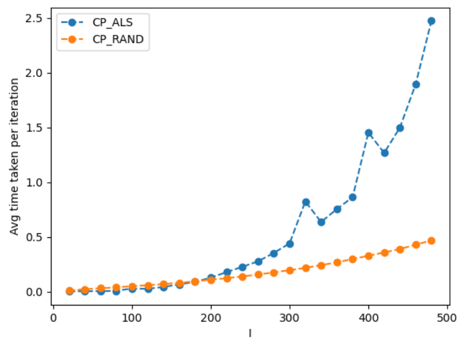
(a) Random $300 \times 300 \times 300$



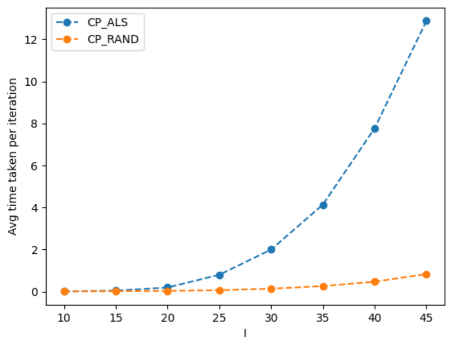
(b) Random $80 \times 80 \times 80 \times 80$

Figure 1: Runtime comparison for fitting the CP tensor decomposition on random synthetic tensors generated to have rank 5 and 1% noise. We compare a single run of three methods using a target rank of 5

4.2 Computation Time of CPALS vs CP-RAND



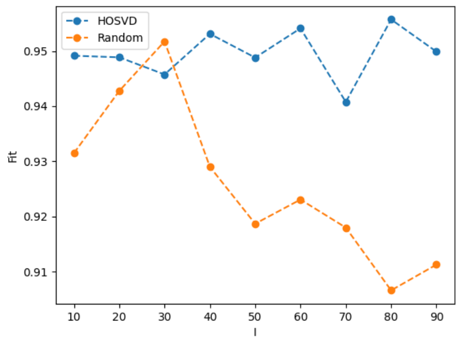
(a) order 3: $I \times I \times I$



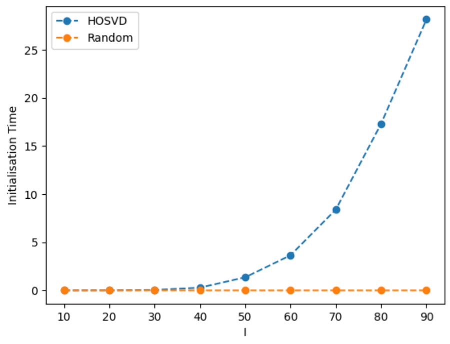
(b) order 5: $I \times I \times I \times I \times I$

Figure 2: Mean time per iteration of CP-ALS, CPRAND and CPRAND-MIX for 3rd- and 5thorder tensors. The target rank is R = 5

4.3 HOSVD vs Random Initialisation of factor matrices



(a) order 3: $I \times I \times I$



(b) order 3: $I \times I \times I$

Figure 3: Fig (a) shows the fit value for HOSVD and Random Initialisation for mode 3 tensor, Fig (b) initialisation time for HOSVD and random initialisation

References

[BBK17] Casey Battaglino, Grey Ballard, and Tamara G. Kolda. A practical randomized CP tensor decomposition. *CoRR*, abs/1701.06600, 2017.

[DMMS07] Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *CoRR*, abs/0710.1435, 2007.