A Practical Randomized CP Tensor Decomposition

Kartik, Tanmay, Aayush, Ganesh, Ojjas

April 2023

Introduction I

What is the Paper About?

- CP/PARAFAC Decomposition is a useful tool in data analytics
- The algorithm to compute the decomposition using the Alternating Least Squares Method (ALS) is slow
- Paper attempts to solve this by reducing computation time and memory

How do they do it?

- Randomized methods have been useful in solving Linear Least Squares Problem

Background I

Important Equations and Results

- Tensor $\mathcal{X} \in \mathbb{R}^{I_1 x \cdots x I_N}$
- mode$-n$ matricization with one-indexing

$$x_{i_i, i_2, \cdots, i_N} = \left[\mathbf{X}_{(n)}\right]_{(i_n, j)} \tag{1}$$

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) J_k \quad \text{where } J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m. \tag{2}$$

Background II

Important Equations and Results

$\mathbf{A} \in \mathbb{R}^{IxJ}$, $\mathbf{B} \in \mathbb{R}^{IxJ}$

$$(\mathbf{A} \odot \mathbf{B})^T(\mathbf{A} \odot \mathbf{B}) = \mathbf{A}^T\mathbf{A} * \mathbf{B}^T\mathbf{B} \tag{3}$$

$$\mathbf{AB} \otimes \mathbf{CD} = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \otimes \mathbf{D}) \tag{4}$$

$$\mathbf{AB} \odot \mathbf{CD} = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \odot \mathbf{D}). \tag{5}$$

Background III

Important Equations and Results

$$\mathcal{Y} = \mathcal{X} \times_n \mathbf{A} \Leftrightarrow \mathbf{Y}_{(n)} = \mathbf{A}\mathbf{X}_{(n)}. \tag{6}$$

$$\mathcal{Y} = \mathcal{X} \times {}_1\mathbf{U}^{(1)} \cdots \times {}_N\mathbf{U}^{(N)} \quad \Leftrightarrow \tag{7}$$

$$\mathbf{Y}^{(n)} = \mathbf{U}^{(n)}\mathbf{X}_{(n)} \left(\mathbf{U}^{(N)} \otimes \cdots \otimes \mathbf{U}^{(n+1)} \otimes \mathbf{U}^{(n-1)} \otimes \cdots \otimes \mathbf{U}^{(1)}\right)^T. \tag{8}$$

Background IV

Important Equations and Results

CP decomposition approximates order-N tensor as a sum of R rank-one tensors

$$\mathcal{X} \approx \tilde{\mathcal{X}} = \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ \cdots \circ a_r^{(N)} \tag{9}$$

$$\mathbf{A}^{(n)} = \left[\begin{array}{cccc} \mathbf{a}_1^{(n)} & \mathbf{a}_2^{(n)} & \cdots & \mathbf{a}_r^{(n)} \end{array} \right] \in \mathbb{R}^{I_n \times R} \tag{10}$$

Background V

Important Equations and Results

Mode-n matricization in terms of factor matrices

$$\tilde{\mathbf{X}}_{(n)} = \mathbf{A}^{(n)}\mathbf{Z}^{(n)\top} \tag{11}$$

$$\mathbf{Z}^{(n)} = \mathbf{A}^{(N)} \odot \cdots \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)} \tag{12}$$

Introducing a normalizing factor λ_r

$$\tilde{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)} \tag{13}$$

Sketching

The idea is to avoid solving a computationally expensive problem, and instead solve another problem, whose solution will converge to yours with high probability.

Sketching

The idea is to avoid solving a computationally expensive problem, and instead solve another problem, whose solution will converge to yours with high probability.

Sketching

$$\min \|Ax - b\| \to \min \|SAx - Sb\|$$

Leverages

The leverages of a matrix A are the norms of the rows of the left singular Matrix U

$$l_i = \|U(i,:)^2\|$$

Shows the importance of the row in constructing the column space of the Matrix A

Coherence

The coherence of a matrix is the maximum leverage scores amongst its rows. It is a measure of how "spread out" the **importance of the rows is**.

$$\mu(A) = \max_{i \in [n]} l_i$$

Coherence

The coherence of a matrix is the maximum leverage scores amongst its rows. It is a measure of how "spread out" the **importance of the rows is**.

$$\mu(A) = \max_{i \in [n]} l_i$$

The lower the coherence, the better random sketching gets!

Coherence

The coherence of a matrix is the maximum leverage scores amongst its rows. It is a measure of how "spread out" the **importance of the rows is**.

$$\mu(A) = \max_{i \in [n]} l_i$$

The lower the coherence, the better random sketching gets!

$$\mu(A) \geq \frac{d}{n} \quad | A \in \mathbb{R}^{n \times d}$$

SFDA Mixing

- This is done to mix the rows of A such that the coherences can be evenly spread before random sampling.
- Moreover, we use special unitary matrices F (like a DFT Matrix) in order to perform matrix computation fast (using FFT) and a Diagonal Matrix D. (See [DMMS07]) for more details.

SFDA Mixing

- This is done to mix the rows of A such that the coherences can be evenly spread before random sampling.
- Moreover, we use special unitary matrices F (like a DFT Matrix) in order to perform matrix computation fast (using FFT) and a Diagonal Matrix D. (See [DMMS07]) for more details.

$$\min \|Ax - b\| \rightarrow \min \|SFDAx - SFDb\|$$

SFDA Mixing

- This is done to mix the rows of A such that the coherences can be evenly spread before random sampling.
- Moreover, we use special unitary matrices F (like a DFT Matrix) in order to perform matrix computation fast (using FFT) and a Diagonal Matrix D. (See [DMMS07]) for more details.

$$\min \|Ax - b\| \rightarrow \min \|SFDAx - SFDb\|$$

The accuracy of this operation can be provided using FJLT and its variants.

CP ALS I

Objective

- Fitting CP method using Alternating Least Squares Method
- Alternate among modes and fix all factor matrices except $\mathbf{A}^{(n)}$
- From Eq. 13, we get the objective as:

$$\arg\min_{\mathbf{A}^{(n)}} \|\mathbf{X}_{(n)} - \mathbf{A}^{(n)}\mathbf{Z}^{(n)T}\|_F. \tag{14}$$

Normal Equation

In CP-ALS, normal equation of Eq. 14 is used

$$\mathbf{X}_{(n)}\mathbf{Z}^{(n)} = \mathbf{A}^{(n)}\left(\mathbf{Z}^{(n)\top}\mathbf{Z}^{(n)}\right) \tag{15}$$

CP ALS II

Normal Equation

Using Eq. 3, we get:

$$\mathbf{Z}^{(n)\top}\mathbf{Z}^{(n)} = \mathbf{A}^{(N)\top}\mathbf{A}^{(N)} * \cdots * \mathbf{A}^{(n+1)\top}\mathbf{A}^{(n+1)} * \mathbf{A}^{(n-1)\top}\mathbf{A}^{(n-1)} * \cdots * \mathbf{A}^{(1)\top}\mathbf{A}^{(1)} \quad (16)$$

Initialization

- $\mathbf{A}^{(n)}$ can be initialized using HOSVD
- It can also be initialized using a random distribution

CP ALS III

Algorithm 1: CP-ALS Algorithm

```
Input: $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, R
Output: $\boldsymbol{\lambda}$, $\{\mathbf{A}^{(n)}\}$
1  Function CP-ALS($\mathcal{X}$, R):
2      Initialize factor matrices $\mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}$
3      repeat
4          for $n = 1, \dots, N$ do
5              $\mathbf{V} \leftarrow \mathbf{A}^{(N)T}\mathbf{A}^{(N)} * \cdots * \mathbf{A}^{(n+1)T}\mathbf{A}^{(n+1)} * \mathbf{A}^{(n-1)T}\mathbf{A}^{(n-1)} * \cdots * \mathbf{A}^{(1)T}\mathbf{A}^{(1)}$
6              $\mathbf{Z}^{(n)} \leftarrow \mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)}$
7              $\mathbf{W} \leftarrow X^{(n)}\mathbf{Z}^{(n)}$
8              Solve $\mathbf{A}^{(n)}\mathbf{V} = \mathbf{W}$ for $\mathbf{A}^{(n)}$
9              Normalize columns of $\mathbf{A}^{(n)}$ and update $\boldsymbol{\lambda}$
10         end
11     until termination criteria met
12     return $\boldsymbol{\lambda}$, factor matrices $\{\mathbf{A}^{(n)}\}$
```

CP ALS IV

Computation Cost

- *Line 5*: $O(R^2 \sum_{m \neq n} I_m) + O(R^2 N)$
- *Line 6*: $O(R \prod m \neq n I_m)$
- *Line 7:* $O(R \prod_m I_m)$
- *Line 8:* $O(2R^2 I_n)$
- Outer Iteration: $O(NR \prod_n I_n)$
- Initialization: $(\sum_{n>1} I_n) \prod_m I_m + \sum_{n>1} I_n^3$

CP RAND I

Modifying Equations

We will rewrite Eq. 14 as :

$$\arg\min_{\mathbf{A}^{(n)}} \|\mathbf{A}^{(n)}\mathbf{Z}^{(n)T} - \mathbf{X}_{(n)}\|_F. \tag{17}$$

Random Sampling

- Uniformly sample rows from $\mathbf{Z}^{(n)}$ and the corresponding rows from $\mathbf{X}_{(n)}^T$
- Let S denote the number of desired number of samples
- Let $\mathcal{S}$ denote the samples from $\{1, \cdots, \prod_{m\neq n} I_m\}$ and $|\mathcal{S}| = S$
- Rows of $\mathbf{S}$ of $\prod_{m\neq n} I_m \times \prod_{m\neq n} I_m$ identity matrix

CP RAND II

How is it Better?

- Forming the (full) Khatri-Rao product $\mathbf{Z}^{(n)}$ is expensive
- Compute $\mathbf{S}\mathbf{Z}^{(n)}$ without computing $\mathbf{Z}^{(n)}$
- Using matricization mapping, the j^{th} row of $\mathbf{Z}^{(n)}$ of Hadamard Product

$$\begin{aligned}\mathbf{Z}^{(n)}(j,:) = \mathbf{A}^{(1)}(i_1,:) * \cdots * \mathbf{A}^{(n-1)}(i_{n-1},:) \\ * \mathbf{A}^{(n+1)}(i_{n+1},:) * \cdots * \mathbf{A}^{(N)}(i_N,:)\end{aligned} \quad (18)$$

CP RAND III

Algorithm 2: Sampled Khatri-Rao Product

```
Input: $\mathbf{S}$, $\mathbf{A}^{(N)}, \dots, \mathbf{A}^{(n+1)}, \mathbf{A}^{(n-1)}, \dots, \mathbf{A}^{(1)}$
Output: $\mathbf{Z}_S$
Function SKR($\mathbf{S}$, $\mathbf{A}^{(N)}, \dots, \mathbf{A}^{(n+1)}, \mathbf{A}^{(n-1)}, \dots, \mathbf{A}^{(1)}$):
    Retrieve idxs from $\mathbf{S}$
    $\mathbf{Z}_S \leftarrow 1$                                          ▷ $1 \in \mathbb{R}^{S \times R}$
    for $m = 1, \dots, n-1, n+1, \dots, N$ do
        $\mathbf{A}_S^{(m)} \leftarrow \mathbf{A}^{(m)}(\mathbf{idxs}(:, m), :)$            ▷ MATLAB-style indexing
        $\mathbf{Z}_S \leftarrow \mathbf{Z}_S * \mathbf{A}_S^{(m)}$
    end
    return $\mathbf{Z}_S$
```

CP RAND IV

Algorithm 3: CPRAND Algorithm

```
Input: $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, $\mathbf{R}$, $\mathbf{S}$
Output: $\boldsymbol{\lambda}$, $\{\mathbf{A}^{(n)}\}$
1  Function CPRAND($\mathcal{X}$, $\mathbf{R}$, $\mathbf{S}$):
2      Initialize factor matrices $\mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
3      repeat
4          for $n = 1, \ldots, N$ do
5              Define sampling operator $\mathbf{S} \in \mathbb{R}^{S \times \prod_{m \neq n} I_m}$
6              $\mathbf{Z}_S \leftarrow \mathrm{SKR}(\mathbf{S}, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(n-1)}, \mathbf{A}^{(n+1)}, \ldots, \mathbf{A}^{(N)})$
7              $\mathbf{X}_S^T \leftarrow \mathbf{S}\mathbf{X}_{(n)}^T$
8              $\mathbf{A}^{(n)} \leftarrow \arg\min_{\mathbf{A}} \|\mathbf{Z}_S \mathbf{A}^T - \mathbf{X}_S^T\|_F$
9              Normalize columns of $\mathbf{A}^{(n)}$ and update $\boldsymbol{\lambda}$
10         end
11     until termination criteria met
12     return $\boldsymbol{\lambda}$, factor matrices $\{\mathbf{A}^{(n)}\}$
```

CP RAND V

Computation Cost

- Generating S random multi-indices: $O(SN)$
- *Line 6:* $O(SR(N-1))$
- *Line 8:* $O(2SR^2 + 2SRI_n + R^2 I_n)$. Assuming $I_n > R$ and $S > R$, we get $O(2SRI_n)$
- Outer Iteration: $O(SR \sum_n I_n)$

CP RAND VI

Coherence

- Effectiveness of CPRAND depends on the coherence of the coefficient matrix $\mathbf{Z}^{(n)}$

Lemma 3

Given $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$, then $\mu(\mathbf{A} \otimes \mathbf{B}) = \mu(\mathbf{A})\mu(\mathbf{B})$

Lemma 4

Given $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$, then $\mu(\mathbf{A} \odot \mathbf{B}) \leq \mu(\mathbf{A})\mu(\mathbf{B})$

CP RANDMIX I

Failure of CPRAND

- If the individual factor matrices happen to be highly coherent, CPRAND may fail to converge
- This can be avoided by mixing terms before sampling
- Apply FJLT transformation to inner iteration before sampling

$$\arg\min_{\mathbf{A}^{(n)}} \|\mathcal{F}\mathbf{D}\mathbf{Z}^{(n)}\mathbf{A}^{(n)T} - \mathcal{F}\mathbf{D}\mathbf{X}_{(n)}\|_F \tag{19}$$

$$\hat{\mathbf{Z}}^{(n)} = \bigodot_{\substack{m=N\\ m\neq n}}^{1} \mathcal{F}_m\mathbf{D}_m\mathbf{A}^{(m)} = \left(\bigotimes_{\substack{m=N\\ m\neq n}}^{1} \mathcal{F}_m\mathbf{D}_m\right)\mathbf{Z}^{(n)} = \left(\bigotimes_{\substack{m=N\\ m\neq n}}^{1} \mathcal{F}_m\right)\left(\bigotimes_{\substack{m=N\\ m\neq n}}^{1} \mathbf{D}_m\right)\mathbf{Z}^{(n)} \tag{20}$$

CP RANDMIX II

Equation - Before Sampling

$$\underset{\mathbf{A}^{(n)}}{\arg\min} \left\| \left(\bigotimes_{\substack{m=N \\ m\neq n}}^{1} \mathcal{F}_m \mathbf{D}_m \right) \mathbf{Z}^{(n)} \mathbf{A}^{(n)T} - \left(\bigotimes_{\substack{m=N \\ m\neq n}}^{1} \mathcal{F}_m \mathbf{D}_m \right) \mathbf{X}^{(n)T} \right\|_F \quad (21)$$

Equation - After Sampling

$$\underset{\mathbf{A}^{(n)}}{\arg\min} \left\| \mathbf{S}\hat{\mathbf{Z}}^{(n)} \mathbf{A}^{(n)T} - \mathbf{S}(\mathbf{D}_n \mathcal{F}_n \mathbf{X}^{(n)})^T \right\|_F \quad (22)$$

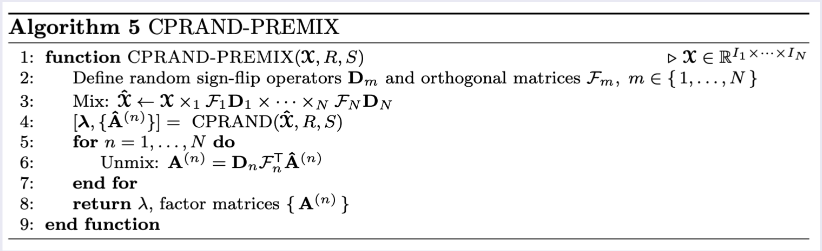
CP RANDMIX III

Algorithm 4: CPRAND-MIX Algorithm

```
Input: $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, $\mathbf{R}$, $\mathbf{S}$
Output: $\boldsymbol{\lambda}$, $\{\mathbf{A}^{(n)}\}$
1  Procedure CPRAND-MIX($\mathcal{X}$, $\mathbf{R}$, $\mathbf{S}$):
2      Initialize factor matrices $\mathbf{A}^{(m)}$, $m \in \{2, \ldots, N\}$
3      Define random sign-flip operators $\mathbf{D}_m$ and unitary matrices $\mathcal{F}_m$,
         $m \in \{1, \ldots, N\}$
4      Mix factor matrices: $\hat{\mathbf{A}}^{(m)} \leftarrow \mathcal{F}_m \mathbf{D}_m \mathbf{A}^{(m)}$, $m \in \{2, \ldots, N\}$
5      Mix tensor: $\hat{\mathcal{X}} \leftarrow \mathcal{X} \times_1 \mathcal{F}_1 \mathbf{D}_1 \cdots \times_N \mathcal{F}_N \mathbf{D}_N$
6      repeat
7          for $n = 1, \ldots, N$ do
8              Define sampling operator $\mathbf{S} \in \mathbb{R}^{S \times \prod_{m \neq n} I_m}$
9              $\hat{\mathbf{Z}}_S \leftarrow \mathrm{SKR}(\mathbf{S}, \hat{\mathbf{A}}^{(N)}, \ldots, \hat{\mathbf{A}}^{(n+1)}, \hat{\mathbf{A}}^{(n-1)}, \ldots, \hat{\mathbf{A}}^{(1)})$
10             $\hat{\mathbf{X}}_S^T \leftarrow \mathbf{D}_n \mathbf{F}_n^* (\mathbf{S} \hat{\mathcal{X}}_{(n)}^T)^T$
11             $\mathbf{A}^{(n)} \leftarrow \arg\min_{\mathbf{A}} \|\hat{\mathbf{Z}}_S \mathbf{A}^T - \mathbf{X}_S^T\|_F$, subject to $\mathbf{A}$ being real-valued
12             Normalize columns of $\mathbf{A}^{(n)}$ and update $\boldsymbol{\lambda}$
13             $\hat{\mathbf{A}}^{(n)} \leftarrow \mathcal{F}_n \mathbf{D}_n \mathbf{A}^{(n)}$
14         end
15     until termination criteria met
16     return $\boldsymbol{\lambda}$, factor matrices $\{\mathbf{A}^{(n)}\}$
```

CP RANDMIX IV

Pre-Mixing



Algorithm 5 CPRAND-PREMIX

```
1: function CPRAND-PREMIX($\mathfrak{X}, R, S$)                                    ▷ $\mathfrak{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$
2:     Define random sign-flip operators $\mathbf{D}_m$ and orthogonal matrices $\mathcal{F}_m$, $m \in \{1, \ldots, N\}$
3:     Mix: $\hat{\mathfrak{X}} \leftarrow \mathfrak{X} \times_1 \mathcal{F}_1 \mathbf{D}_1 \times \cdots \times_N \mathcal{F}_N \mathbf{D}_N$
4:     $[\boldsymbol{\lambda}, \{\hat{\mathbf{A}}^{(n)}\}] =$ CPRAND($\hat{\mathfrak{X}}, R, S$)
5:     for $n = 1, \ldots, N$ do
6:         Unmix: $\mathbf{A}^{(n)} = \mathbf{D}_n \mathcal{F}_n^{\mathsf{T}} \hat{\mathbf{A}}^{(n)}$
7:     end for
8:     return $\lambda$, factor matrices $\{ \mathbf{A}^{(n)} \}$
9: end function
```

Figure: CP-Rand Premix

CP RANDMIX V

Cost

- Extra Cost compared to Algorithm 3 on *Line 3*: $O\left(\sum_{k=1}^{N}\prod_m [I_m \log I_k]\right) = O\left(\left(\prod_m I_m\right)\log\left(\prod_m I_m\right)\right)$

Stopping Criteria I

Problem with Residual Norm

- Sampled least squares computations are so inexpensive that checking this stopping condition can take longer than the iteration
- Authors propose a sampling based for computing approximate stopping criterion

Notation

Let $[N]$ denote the set $\{1, \cdots, N\}$, Let $\hat{P}$ denote natural number. Let us take

$$\hat{\mathcal{I}} \subset \mathcal{I} \equiv [I_1] \otimes [I_2] \otimes \cdots \otimes [I_N]$$

as a random subset of $\hat{P}$ indices of $\mathcal{X}$

Stopping Criteria II

Notation

Let $\mathcal{E} = \mathcal{X} - \tilde{\mathcal{X}}$, we can see that

$$\|\mathcal{E}\|^2 = \sum_{\mathbf{i} \in \mathcal{I}} e_{\mathbf{i}}^2 = P\mu$$

$$P = \prod_n I_n$$

$$\mu = \text{mean}\left\{e_{\mathbf{i}}^2 \mid \mathbf{i} \in \mathcal{I}\right\}$$

Stopping Criteria III

Approximation

The next step is to approximate μ with $\hat{\mu}$, which is on the subset $\hat{\mathcal{I}}$

$$\mu = \hat{\mu} \quad \text{where} \quad \hat{\mu} = \text{mean}\left\{ e_{\mathbf{i}}^2 \mid \mathbf{i} \in \hat{\mathcal{I}} \right\}$$

The new residual norm can be estimated as

$$\frac{\|\mathcal{E}\|}{\|\mathcal{X}\|} = \frac{(P\mu)^{1/2}}{\|\mathcal{X}\|} \approx \frac{(P\hat{\mu})^{1/2}}{\|\mathcal{X}\|}$$

Stopping Criteria IV

Multiplicative Chernoff-Hoeffding Bounds

Let $\mu_{\text{max}} = \max_i(e_i^2)$ be be the maximum value. For any $\gamma \in (0, 1)$, we can write the upper and lower tail bound as:

$$\begin{aligned}
\Pr\{\hat{\mu} \geq (1+\gamma)\mu\} &\leq \exp\left(-\frac{2\gamma^2\mu^2\hat{P}}{\mu_{\text{max}}^2}\right) \\
\Pr\{\hat{\mu} \leq (1-\gamma)\mu\} &\leq \exp\left(-\frac{\gamma^2\mu^2\hat{P}}{\mu_{\text{max}}^2}\right)
\end{aligned} \tag{23}$$

Stopping Criteria V

Lemma

For any $\gamma \in (0, 1)$, we can bound the relative difference in the approximated and true error as:

$$\Pr\left\{\sqrt{1-\gamma} \leq \frac{(P\hat{\mu})^{1/2}}{\|\mathcal{E}\|} \leq \sqrt{1+\gamma}\right\} \leq \exp\left(-2\frac{\gamma^2\mu^2\hat{P}}{\mu_{\max}^2}\right) \quad (24)$$

Computation Cost

- Cost of Computing $\hat{P}\hat{\mu}$: $O(\hat{P}RN)$
- Cost of Computing exact error: $O(R\prod_n I_n)$

Computational Time

Data Generation

Data Generation

- **Synthetic Data**
  1. Generated factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$, $n \in \{1, \dots, N\}$
  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 $\mathcal{X}$
  $$\mathcal{X} = \mathcal{X}_{\text{true}} + \eta \left(\frac{\|\mathcal{X}_{\text{true}}\|}{\|\mathcal{N}\|} \right) \mathcal{N}$$

Data Generation

Data Generation

- **Synthetic Data**
- **Coil Data Set**
  1. Used the *Coil-100* data set for the experiments.
  2. Contains images of size $128 \times 128 \times 3$ of 100 different objects, with 72 different angles for each object.
  3. Gives a tensor $\mathcal{X}$ from non Gaussian distribution of the data. $\mathcal{X} \in \mathbb{R}^{128 \times 128 \times 3 \times 7200}$

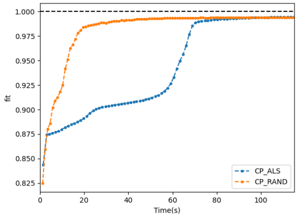
Experiments

Experiments Conducted

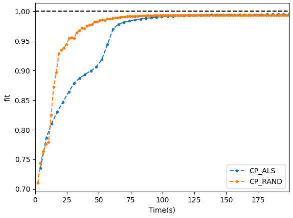
1. **Fit Time**: Measure the accuracy as a function of time for different models.
2. **Computational Time:** Measure the average time taken per iteration, ignoring the convergence check.
3. **Initialization Time:** Time required to initialize the factor matrices by *Random* and *HOSVD* method.

Experiments

Fit Time



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

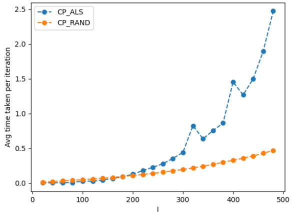


(b) Random $80 \times 80 \times 80 \times 80$

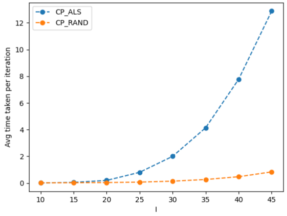
Figure: Runtime comparison for fitting the CP tensor decomposition on random synthetic tensors generated to have rank 5 and 1% noise. We compare a single run of three methods using a target rank of 5

Experiments

Computational Time



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

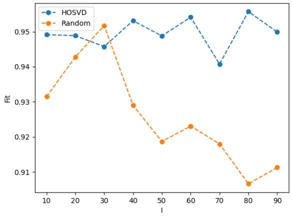


(b) order 5: $I \times I \times I \times I \times I$

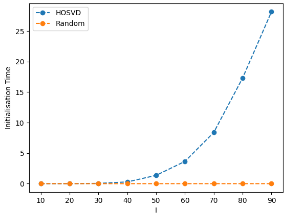
Figure: Mean time per iteration of CP-ALS, CPRAND and CPRAND-MIX for 3rd- and 5thorder tensors. The target rank is $R = 5$

Experiments

HOSVD vs Random Initialisation



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



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

Figure: Fig (a) shows the fit value for HOSVD and Random Initialisation for mode 3 tensor, Fig (b) initialisation time for HOSVD and random initialisation, Here data is synthetically generated with actual rank 7, and Target rank 5

Conclusion

Contributions of Paper [BBK17]

- Randomized algorithm prefers incoherent matrices
- Prove that the coherence of the Khatri-Rao product is bounded above by the product of the coherence of its factors
- CPRAND algorithm that uses a randomized least squares solver for the subproblems in CP-ALS
- CPRAND-MIX algorithm that employs efficient mixing to promote incoherence
- Stopping condition that estimates the model fit error

References I

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

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